ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

GRADO EN INGENIERÍA AEROESPACIAL EN AERONAVEGACIÓN.

TRABAJO FIN DE GRADO

ESTUDIO DE CLASIFICADORES BASADO EN MÉTODOS DE APRENDIZAJE MÁQUINA PARA LA DETECCIÓN DE EVENTOS DE NIEBLA EN LA AUTOVÍA A8

Autor: Javier Heras García

Tutor: David Casillas Pérez

Cotutora: Laura Cornejo Bueno

Curso académico: 2021 / 2022

Resumen

En el presente proyecto se van a diseñar una serie de clasificadores capaces de predecir eventos de visibilidad a partir de una base de datos real de 2018 y 2019, centrada en torno a la autovía A-8 a su paso por el municipio de Mondoñedo.
A partir de esta obtenemos los valores de visibilidad en tres sensores meteorológicos en nuestra área de estudio. Después de ello accedemos a la página web del Centro Europeo de Previsiones Meteorológicas a Plazo Medio (European Centre for Medium-Range Weather Forecasts, ECMWF) para descargar los datos de una serie de variables meteorológicas para nuestra área de estudio. Sin embargo, al estudiar nuestros datos descargados en Panoply 5.0.0, observamos que obtenemos mediciones meteorológicas para cada hora, frente a los cinco minutos de la visibilidad. Para solucionar esto realizaremos en Matlab R2021B promedios de visibilidad para obtener mediciones horarias también. Una vez obtenemos los datos, realizamos una tabla para cada sensor con las variables meteorológicas y su valor de visibilidad asociado. A continuación, establecemos unos filtros de visibilidad para obtener el tipo de niebla asociada a esta y pasamos nuestros datos por ellos, añadiendo los resultados a las tablas.
Una vez tenemos las tablas, dividimos los datos en un 80 % para entrenamiento y un 20 % para pruebas. Los datos de pruebas los reservamos para usar los mismos en todos los experimentos. En cuanto a los de entrenamiento, al estar los datos desbalanceados, usaremos las técnicas de sobremuestreo y submuestreo para obtener el mismo número de muestras de cada evento de visibilidad. Tras obtener los datos de entrenamiento, los usamos como entradas para generar y entrenar nuestros distintos tipos de clasificadores. Después de ello pasamos nuestros datos de prueba para que predigan los eventos de niebla y los comparamos con las salidas reales. A partir de estas comparaciones obtenemos las diferentes métricas de salida para estudiar y evaluar nuestros clasificadores.
Obtenemos que el mejor para predecir los tres eventos mediante las dos técnicas de balanceo es el Bagged Trees, seguido por el Wide NN y el Fine Gaussian SVM con resultados parejos entre sí y destacando por su resistencia al menor número de muestras, el Fine KNN con unos resultados algo peores y, por último, el Fine Tree con los peores resultados de este grupo.

Abstract

The aim of the present proyect is designing several clasifiers capable of predicting visibility events based on a real database of 2018 and 2019, located in the motorway A-8 in Mondañedo. From this database we obtain the values of visibility in three meteorological sensors in our area. After this, we access to the webpage ECMWF, where we download data of a series of meteorological variables for our area.

However, when studying our variables in Panoply 5.0.0, we discover we get information for meteorological variables every hour, instead of five minutes, how for visibility. Because of this, we calculate visibility averages. After this, we create tables with our meteorological and visibility data for each sensor, and filter them with our visibility filter to obtain the fog associated to each measure, and add the type of fog associated to the table.

Next, we split our data up into a 80 % for training and a 20 % for testing all our classificators with the same samples. Due to our training data are umbalanced, we use oversampling and undersampling techniques to get the same number of samples of all every event. Finally, after this, we use our training data to design and train our classifiers of all the types, and test them with our test data. After this, we compare the results of predictions with the real events and get our metric of study.

From these metrics we get Bagged Tree is the best classifiers to predict all the events for both techniques of balance for 2018 and 2019, followed by Wide NN and Fine Gaussian SVM with similar results and resistance to a lower number of samples, Fine KNN with a bit worse results, and lastly, Fine Tree, with the worst results of this group.

Agradecimientos

Transmitir mi más sincero agradecimiento a todos aquellos que me han ayudado a lo largo de esta etapa y han colaborado en esta investigación. En primer lugar, a mi tutor, el profesor ayudante doctor, David Casillas Pérez y a mi cotutora, la profesora ayudante doctora, Laura Cornejo Bueno por su ayuda en la planificación, información y organización en este Trabajo de Fin de Grado.

Índice general

Índice de figuras

Índice de tablas

Lista de acrónimos

TFG Trabajo de Fin de Grado.

TP True Positive, o Verdaderos Positivos.

FP False Positive, o Falsos Positivos.

TN True negative, o Verdaderos Negativos.

FN False Negative, o Falsos Negativos.

ROC Receiver Operating Characteristic, o Característica Operativa del Receptor.

AUC Area under the curve, o Área bajo la curva.

KNN K-Nearest Neighbor, o K-vecinos más cercanos.

NN K-Neural Network, o Red Neuronal.

MLP Multilayer Perceptron, o Perceptrón Multicapa.

SVM Support-vector machine, o Máquinas de vectores de soporte.

ECMWF European Centre for Medium-Range Weather Forecasts, o Centro Europeo de Previsiones Meteorológicas a Plazo Medio.

IA Inteligencia Artificial.

FIS Fuzzy Interference System,o sistema de interferencia difusa.

SIRTA Instrumented Site for Atmospheric Remote Sensing Research.

WRF Weather Research and Forecasting.

SIRTA Instrumented Site for Atmospheric Remote Sensing Research.

SVR Support Vector Regression, o Regresión de Vectores de Soporte

RVR Runway Visual Range,o alcance visual en pista.

POM Proportional Odds Model,o modelo de probabilidades proporcionales.

SVOREX Support Vector Ordinal Regression considering Explicit Constraints.

SVORIM Support Vector Ordinal Regression but with Implicit Constraints.

KDLOR Kernel Discriminant Learning for Ordinal Regression, o Aprendizaje Discriminante de Kernel para Regresión Ordinal

LSTM Long short-term memory, o Red de memoria a largo y corto plazo.

CNN Convolutional Neural Net- works, o Método de redes neuronales convolucionales.

CoNN Condensed Nearest Neighbours, o vecinos más cercanos condensados.

NCR Neighbourhood Cleaning Rule, o regla de limpieza del vecindario.

RUS Random UnderSampler, o submuestreo aleatorio.

RF Random Forest, o bosque aleatorio.

ANN Artificial Neural Network-based methods, o métodos basados en redes neuronales artificiales.

ELM Extreme Learning Machine, o aprendizaje automático extremo.

LREG Linear Regression, o regresión lineal.

EREG ElasticNet Regression, o regresión red elástica.

GLM Generalized Linear Model, o modelo lineal generalizado.

RBF radial basis function, o función base radial.

Capítulo 1

Introducción

En este trabajo se van a implementar una serie de clasificadores de distinto tipo con el objetivo de evaluar por medio de cuál se obtienen los mejores resultados en la predicción de eventos niebla. Debido a la mayor disponibilidad de datos en la A-8 a la altura de Mondoñedo, se toma esta ubicación como referencia, aunque su aplicación final es la predicción en aeropuertos.

1.1. Contexto

En la actualidad, los eventos de niebla constituyen un auténtico quebradero de cabeza para los responsables de muchos aeropuertos, no sólo de España, sino de todo el mundo [1]. Es conocido que actualmente las aerolíneas y aeropuertos están invirtiendo mucho dinero en tener los mejores equipos meteorológicos con el fin de predecir el tiempo que va a acontecer de forma muy precisa, cada hora o incluso cada minuto. Es importante conocer lo mejor posible las condiciones de despegue, aterrizaje y vuelo de nuestras aeronaves, incluido el desplazamiento en pista debidas a situaciones meteorológicas adversas, como por ejemplo, tormenta de rayos, nieve, hielo, fuertes vientos y precipitaciones, o incluso un intenso calor y altas temperaturas, y especialmente y la que más nos compete en nuestro proyecto, una situación de niebla. Es muy importante conocer al minuto el tiempo que vamos a tener para poder anticiparnos y tomar las medidas necesarias para poder seguir operando de forma eficiente y segura y sufrir el menor número de contratiempos posible, tales como, demoras o cancelaciones en nuestros vuelos [2].
Hoy en día, la gran mayoría de los aviones disponen de unos instrumentos a bordo gracias a los cuales pueden volar con seguridad en la gran mayoría de situaciones meteorológicas.
Sin embargo, los momentos más complicados son el aterrizaje, el movimiento en pista o el despegue.
En caso de que haya niebla, por ejemplo, habría que tomar una serie de medidas a la hora del aterrizaje, como llevar más combustible, o incluso aterrizar en otro aeropuerto cercano. Asimismo, a la hora del aterrizaje es importante conocer si se va a producir un evento niebla, para conocer

la visibilidad que se encontrarán y decidir si llevar a cabo un aterrizaje visual o un aterrizaje instrumental. Hoy en día como se ha mencionado antes, casi todas las aeronaves cuentan con instrumentos de vuelo, así como los aeropuertos con equipamiento de aterrizaje instrumental como el Sistema de aterrizaje instrumental o ILS (en inglés Instrument Landing System); por lo que pueden aterrizar en situaciones de baja visibilidad, aunque algunas aeronaves necesiten mayor visibilidad y por tanto no puedan aterrizar en dicho aeropuerto en esas condiciones.
Los mayores contratiempos en caso de niebla se producen al movilizar una aeronave en pista. Al tener que realizar esta tarea de forma manual, es donde se producen las mayores demoras en caso de niebla. Es por ello muy importante poder anticiparse a estos sucesos con el fin de minimizar sus consecuencias.
En caso de poder predecir los sucesos niebla, esto nos podría permitir una mejor organización en los aeropuertos. Contaríamos con tiempo suficiente para poder ajustar nuestros vuelos y preparar los despegues y aterrizajes de nuestros aviones para que no se afecten unos a otros. Todo ello nos permitiría no sufrir los retrasos o incluso cancelaciones debido a estos sucesos, junto con los perjuicios económicos y riesgos sobre la seguridad asociados.
Es por ello, con el fin de anticiparnos a estos sucesos, que se van a desarrollar una serie de clasificadores de distinto tipo, y se estudiarán los resultados de los distintos clasificadores para descubrir por medio de cuál se obtienen las mejores predicciones de los tipos de visibilidad.
En nuestro caso vamos a predecir los sucesos niebla en la Autovía A-8 a la altura del municipio de Mondoñedo. La elección de esta ubicación no es casualidad, esta zona, conocida como el O Alto do Fiouco, es famosa por sus frecuentes nieblas intensas, que apenas te permiten ver a unos metros de distancia y en la cual ya se han sufrido numerosos problemas a consecuencia de ello. Por lo tanto, es una buena situación para desarrollar el proyecto en torno a este área. Posteriormente se podrían extrapolar nuestros clasificadores para poder aplicarse a los aeropuertos para evitar las consecuencias anteriormente mencionadas. La Figura 1.1 muestra una imagen de nuestro área de estudio, la autovía A-8 a su paso por Mondoñedo con sus coordenadas:

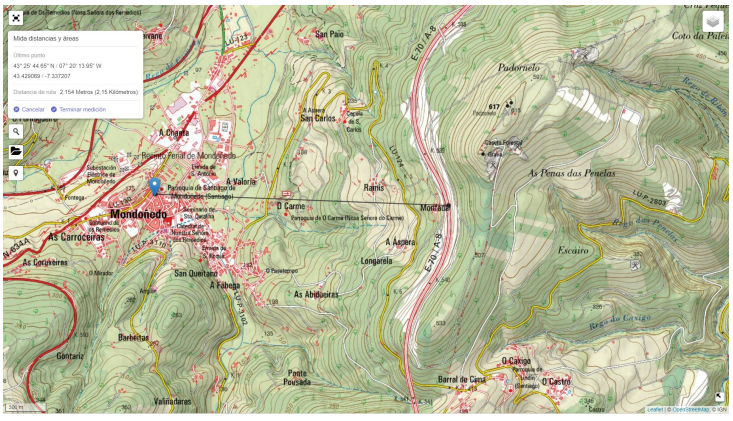


Figura 1.1: Mapa Modoñedo

1.2. Objetivos

Nuestro objetivo principal es diseñar, a partir de una base de datos real, unos clasificadores con los que predecir los eventos niebla. Una vez obtenidos los diferentes clasificadores evaluaremos sus resultados. De esta forma, podremos conocer las cualidades y capacidades de cada uno y decidir cual es el óptimo a la hora de aplicarlos en un caso real. Esto nos va a permitir contar con el predictor más capaz para cada situación y poder anticiparnos de la mejor manera posible, consiguiendo reducir al mínimo los perjuicios que acabamos de ver debido a estos fenómenos.

1.3. Metodología

Para llevar a cabo nuestro proyecto hemos seguido los siguientes pasos que se describen a continuación de forma detallada:

1. Para empezar, se nos proporcionó una base de datos para trabajar. Esta base de datos fue obtenida por medio de tres sensores meteorológicos localizados entorno al área ya mencionada de la A-8 a la altura de Mondoñedo. Estos sensores han calculado una serie de variables meteorológicas en este área cada 5 minutos durante los años 2018 y 2019. Además, en esta base de datos se nos da la visibilidad de salida asociada a estas variables meteorológicas.

2. Una vez tenemos los datos obtenidos por nuestros tres sensores en Mondoñedo, accedemos a la página del ECMWF, a ERA5, donde descargamos datos meteorológicos de forma horaria en nuestra área de estudio, entre ellas: *contenido de agua de lluvia, nubosidad, humedad específica y relativa, contenido líquido, presión en la superficie, temperatura de superficie y de rocío, radiación solar neta y la precipitación total*. Tras descargarlas, abrimos los archivos en el programa Panoply, versión 5.0.0, donde los podemos visualizar y analizar.

3. Al comprobar que los datos descargados han sido medidos de forma horaria y los datos de la visibilidad cada 5 minutos, realizamos promedios de los valores de visibilidad. Cada doce valores de visibilidad realizamos un promedio de estas, consiguiendo la sincronización.
   Para la realización de los promedios se ha hecho uso de Matlab R2021b. Durante este trabajo ha habido intervalos de 12 muestras, en los sensores de visibilidad, que tenían valores nulos. Para solucionar este problema, en los casos en los que hubiera un intervalo de 12 muestras sin valor y que el programa a posteriori no lo tomara como 0 al hacer los modelos, recorremos la base de datos hasta obtener el valor anterior y posterior más próximo y realizamos una regresión lineal con estos valores para rellenar los intervalos de muestras vacías. Una vez rellenadas las muestras vacías realizamos los promedios con normalidad.

4. Una vez tenemos las muestras de visibilidad y nuestras variables meteorológicas de forma horaria, creamos tres tablas para los años 2018 y 2019. Cada una de estas tablas tiene las variables meteorológicas horaria en 2018 y 2019, con su visibilidad asociada, de cada uno de los tres sensores distintos.

5. A continuación, creamos unos filtros por medio de unos umbrales de visibilidad. Para una visibilidad inferior a un determinado valor se considera que hay mucha niebla (clase 3) y para una superior a otro que hay poca niebla (clase 1). Una visibilidad intermedia entre ambos valores la consideraremos media niebla (clase 2). En nuestro caso hemos establecido el umbral de mucha niebla en 400 metros y de media niebla en 1500 metros.
   Tras establecer los filtros desarrollamos un bucle con el que recorremos nuestras tablas y según sea el valor de la visibilidad en cada intervalo, añadimos a nuestras tablas una columna con la categoría del tipo de niebla que tenemos en cada instante.

6. Ahora dividimos nuestras tablas de los tres sensores en 2018 y 2019 en un 80 % para datos de entrenamiento y un 20 % de esos datos para testear nuestros clasificadores de eventos niebla.
   Para seleccionar de forma aleatoria los índices de las muestras que van a datos para entrenamientos y las que van para pruebas, generamos una serie de índices enteros aleatorios, que no se repiten, de la longitud de nuestras tablas. Una vez tenemos estos índices, con un bucle los recorremos hasta el 80 % de su longitud, asignando los valores de nuestras tablas con esos índices a las tablas para entrenamiento. Una vez terminamos el 80 % de los índices, empezamos otro bucle que recorra el 20 % restante de los valores, asignando las muestras de nuestras tablas con esos índices asociados para probar nuestros modelos.

7. Una vez tenemos separados nuestros datos de prueba y de entrenamiento, guardamos los de entrenamiento para usar siempre los mismos en cada tipo de clasificador. Además, los datos de entrenamiento se encuentran desbalanceados, ya que contamos con más sucesos poca niebla que del resto. Necesitamos tener aproximadamente el mismo número de cada tipo de niebla para entrenar correctamente nuestro clasificadores y que pueda predecir de forma eficiente los tres tipos de sucesos. Con este fin, tenemos dos opciones, realizar un sobremuestreo (SMOTE) o submuestreo (Undersampling). Por lo tanto, pasamos nuestras tablas de entrenamiento para nuestros 3 puntos en 2018 y 2019 por nuestros programas en Matlab de sobremuestreo y submuestreo.
   En sobremuestreo, dividimos el número de muestras de la clase más numerosa entre el número de las otras dos. Con ese número hacemos un bucle hasta ese valor para las otras dos clases, en el cual generaríamos ese número de muestras sintéticas de cada una de las que teníamos, multiplicando nuestro número de muestras originales por ese valor obtenido de la división. Para generar una muestra sintética, tomamos la muestra original, y le sumamos la diferencia entre nuestro valor y uno de sus 5 vecinos más próximos (seleccionado de forma aleatoria), multiplicado por un decimal entre 0 y 1 obtenido de forma aleatoria

mediante una fórmula en Matlab. Como en sobremuestreo, la división entre nuestra clase más numerosa y las otras no es exacto, para cubrir el resto y obtener el mismo número de las tres clases, las muestras restantes las generamos por sobremuestreo o submuestreo, según sea el resultado de la división y del sobremuestreo original, necesitando generar más muestras sintéticas o eliminarlas para obtener el mismo número que la clase mayoritaria.
En cuanto al submuestreo, es muy simple, obtenemos el número de muestras de la clase con menos muestras de las tres y obtenemos el número de muestras que hay que eliminar de las otras dos clases para obtener ese número.
Una vez hecho esto ya tenemos las dos tablas de sobremuestreo y submuestreo para cada uno de los tres puntos de los dos años.

8. Una vez ya tenemos todas las tablas de entrenamiento y prueba balanceadas, empezamos con los modelos de los clasificadores. En este apartado lo que hacemos es ir generando una serie de modelos en Matlab de distintos tipos de clasificadores, los cuales se explicarán más adelante. Para ello, le pasamos como entrada las distintas tablas de entrenamiento que hemos ido generando, tanto para sobremuestreo como para submuestreo, entrenando al modelo con los datos meteorológicos y el tipo de niebla que generan. Una vez entrenado usamos el modelo creado para que nos prediga un tipo de niebla y comparamos la que nos predice el modelo con la real de nuestras tablas.

9. Una vez creados los distintos tipos de modelos para todos nuestros puntos por sobremuestreo y submuestreo en 2018 y 2019, obtenemos una serie de métricas para los resultados de la fase de prueba de nuestros modelos, con las que estudiaremos los distintos tipos de clasificadores más adelante.
Entre esas métricas encontramos, las matrices de confusión, la precisión y exhaustividad o el valor F.

10. Por último, a partir de las métricas obtenidas para cada uno de los casos, las representamos y estudiamos, elaborando un estudio de los distintos clasificadores en la memoria.

1.4. Estructura de la memoria

Una vez obtenemos los resultados de la parte práctica de nuestro proyecto, realizamos la memoria del mismo.

- Capítulo 1, redactamos la introducción, el apartado donde desarrollamos en profundidad el contexto en el que se ha desarrollado nuestro proyecto, la importancia de él y los objetivos y resultados que esperamos obtener. Después de la introducción, desarrollamos la metodología, en la que describimos en profundidad todos los pasos que se han seguido en la realización del proyecto.

- Capítulo 2, desarrollamos el Estado del Arte, en el que explicamos algunos ejemplos de trabajos realizados por otras personas en este ámbito relacionado con nuestro proyecto, una descripción de su trabajo, las técnicas que se han usado para realizarlos y los resultados obtenidos en ellos.

- Capítulo 3, realizamos un estudio teórico y un análisis de los distintos métodos y técnicas llevados a cabo en nuestro trabajo, un desarrollo teórico y explicación de los distintos tipos de clasificadores desarrollados, y los subtipos que se pueden encontrar en ellos.

- Capítulo 4, exponemos los experimentos llevados a cabo y los resultados obtenidos para los distintos tipos de clasificadores.

- Finalmente, Capítulo 5, mostramos las conclusiones que sacamos de los resultados obtenidos en el capítulo anterior y las analizamos, y en el capítulo seis desarrollamos la líneas futuras, exponiendo mejoras que implementaríamos en nuestros modelos para mejorarlo si pudiéramos.

Capítulo 2

Estado del Arte

Debido a la importancia de desarrollar Inteligencias Artificiales (IAs) capaces de realizar tareas complejas, desde hace décadas se ha estado trabajando en este campo con ese fin. A continuación, vamos a desarrollar un estudio de diversos trabajos a lo largo de las últimas décadas sobre el desarrollo de máquinas artificiales; en el ámbito de los eventos niebla:

Empezamos nuestro estudio en la década de 1980 con un estudio de la escuela de posgrado de la armada de Estados Unidos [3]. Este estudio se desarrolló con el objetivo de predecir la niebla, reducir los riesgos en transporte marítimo y vuelos a baja altura asociados a esta y facilitar localizar las fuerzas enemigas.
En este estudio se realizó una investigación sobre la distribución estadística de seis parámetros de salida del modelo, en función de la ocurrencia de los eventos niebla y no niebla, para un área climatológicamente homogénea del Atlántico Norte en verano. Tras examinar y comparar los histogramas de las distribuciones de unos predictores, se decidió usar: flujo de humedad superficial, la diferencia entre la temperatura del aire en 925mb y la temperatura de la superficie del mar, el arrastre, la radiación de onda larga, el flujo de calor sensible y, por último, la frecuencia de estratos.
Mediante este estudio se confirmó la importancia del conocimiento de las distribuciones de los eventos para una exitosa predicción. Asimismo, se ha comprobado que los predictores con distribuciones significativamente sesgadas serán mejor descritos por distribución beta o gamma que por normal. Por el contrario, se ha demostrado que para predictores con distribución en campana, los resultados son menos claros, su interpretación depende del sistema de puntuaje usado y muestran diferencias menos significativas entre las tres distribuciones.
A partir de todo esto, salvo para el predictor frecuencia de estratos, no se llegan a conclusiones definitivas de como influye la bondad de ajuste de una distribución en la predicción. Sin embargo, no se puede rechazar que Beta o Gamma puedan servir como proxy de la Normal.

El primer estudio que encontramos en la década de 1990 es sobre la clasificación de los tipos de nubes mediante casos basados en selección de algoritmos [4]. Este estudio tiene el objetivo de

obtener una predicción precisa del tiempo, debido a su importancia en las diversas actividades de la marina.
Para llevar a cabo esta tarea, se emplearán unos algoritmos de selección de características para mejorar la precisión de nuestros clasificadores, al reducir el número de características usadas para caracterizar una base de datos.
En este estudio se usaron cuatro tipos de algoritmos en función de su estrategia de control y su algoritmo de búsqueda: el forward sequential selection (FSS), el Backwark sequential selection (BSS), la estrategia de control de envoltura (en inglés, wrapped control) y el algoritmo FSS y, por último, el control de envoltura y el algoritmo BSS.
Durante el experimento, se decidió usar algoritmos secuenciales debido a su mejor actuación frente a otros como los exponenciales y aleatorios. En cuanto a los clasificadores, se usó el IB1, un no paramétrico basados en modelos, por sus buenas precisiones en selección de características y al poder usarse como función de evaluación.
Finalmente, los resultados más precisos se obtuvieron con la búsqueda FSS combinada con el IB1 como función de evaluación. En concreto, se observó cómo al usar el IB1 como función separada se mejora la actuación al usar el FSS o el BSS, especialmente la actuación del IB1 con el FSS. También se puede apreciar la actuación comparativamente pobre en cuanto a rendimiento de los dos últimos algoritmos, siendo significativamente más bajas que las otras dos técnicas.

En 1995 encontramos un nuevo intento de usar el aprendizaje automático para predecir eventos climatológicos [5]. En esta ocasión, el objetivo de la investigación era el desarrollo de dos técnicas de aprendizaje automático con las que predecir la niebla en Cuba y posteriormente compararlos entre sí para comprobar cuál de ambos es más eficiente. Los dos algoritmos elegidos fueron la función discriminante Fisher clásica y el LVQ desarrollado por la universidad tecnológica de Helsinki.
Como resultado de la investigación, se obtuvo que el porcentaje de muestras correctas por ambas técnicas era superior al 70 % , por lo que se consideró que ambas realizaron una buena actuación. En aquellos casos en los que la muestra de entrenamiento es lo suficientemente grande y casi equiprobabilística, el algoritmo LVQ provee un mayor número de predicciones correctas. Sin embargo, los resultados obtenidos mediante el LVQ no son estables cuando la muestra de entrenamiento está lejos de ser equiprobilística, dado que el número de casos niebla es reducido.
A partir de los resultados obtenidos, se obtiene la hipótesis de que hasta que estén disponibles muestras más grandes para algunas regiones, será necesario emplear ambos métodos para la predicción de la niebla en Cuba.

Al comienzo de la década del 2000, podemos encontrar un estudio sobre la predicción de hielo mediante técnicas de aprendizaje automático [6] para minimizar la formación de hielo y los graves perjuicios económicos a consecuencia de este, afectando seriamente a las cosechas y

dañándolas.
En este trabajo se explora la posibilidad del desarrollo de un sistema de predicción empírico para la protección de frutas y vegetales contra el hielo en el sur de la provincia de Santa Fe en Argentina. Para este fin se consideran una serie de técnicas de aprendizaje automático en problemas de regresión y clasificación: las redes neuronales artificiales, clasificadores de Bayes simples y los K vecinos más cercanos.
Como resultado del estudio, se obtiene una estructura muy ruidosa de los datos, lo cual apunta solamente a una ligera mejora en la actuación de algunas de las sofisticadas técnicas no lineales empleadas en el estudio, frente a las ecuaciones de regresión multivariable estándar (lineales).
Asimismo, declaran entre las posibles mejoras en futuras investigaciones del problema, usar datos más recientes que no estaban disponibles en el momento del estudio y, lo más importante de todo, nuevas variables predictoras que ayuden en la extracción de información no lineal más precisa. Además, están estudiando modificar el conjunto de datos de entrenamiento para redes neuronales artificiales excluyendo las muestras obvias de no hielo. Por último, también han decidido proponer como mejora considerar la red neuronal artificial con un número mucho mayor de unidades ocultas, reduciendo por tanto el error al maximizar la varianza del conjunto.

En mayo de 2008 se llevó a cabo un estudio para evitar desastres como consecuencia de la reducción de la visibilidad [7]. Para ello, desarrollan unos índices para la predicción de eventos niebla, basados en el post-procesado de variables meteorológicas del modelo ECMWF recogidas entre 1990 y 2002. Para ello se utilizará minería de datos, un proceso de descubrir automáticamente información útil en grandes repositorios de datos.
Como resultado de la minería de datos, se obtiene un modelo de clasificación de dos clases, niebla o no niebla. Para la realización de los modelos se han utilizado los algoritmos de árboles de decisión y las redes bayesianas. Durante el experimento, también cabe destacar el uso de la técnica de submuestreo para balancear las bases de datos.
Los cinco mejores modelos de árboles obtuvieron el 80 % de eventos niebla clasificados correctamente y el 30 % incorrectamente, frente al 80 % y el 25 % obtenidos por los dos mejores mediante Bayes.
Estos resultados se analizaron mediante la curva característica operativa del receptor (Receiver Operating Characteristic, ROC), y el área bajo la curva (Area Under Curve, AUC), y matrices de confusión. Estas últimas permiten observar el elevado número de falsos positivos, el cuál sugiere reducir mediante alternativas el desbalanceo en los datos de entrenamiento. Por último, también se aprecia que seis modelos superan el umbral del AUC de 0.7 y en cinco de ellos, el 0.8.

En 2008 se llevó a cabo también una investigación para predecir la niebla usando un sistema de interferencia difusa (FIS) basado en reglas, con el objetivo de introducir el concepto FIS en predicción de niebla [8].

Este enfoque usa el concepto de sistema de lógica difusa pura para mapear a partir de conjuntos borrosos. La base de este método es construir el dominio base de las reglas difusas a partir de las observaciones actuales del tiempo diario en invierno en Nueva Delhi.
A partir de los resultados se observa como la dispersión del rocío y su tasa de cambio son los parámetros más importantes en la predicción. Asimismo, se apreció como la formación de niebla es dominante para el punto de rocío superior a 7ºC, junto con su difusión entre 1 y 3 ºC. A parte de esto, tiene que darse una tasa de cambio de la dispersión del punto de rocío negativa y una velocidad del viento inferior a 4 nudos.
Con este estudio se nos presenta una técnica capaz de predecir la niebla con un horizonte temporal de entre cinco y seis horas de adelanto, la cual ha probado su eficiencia en el estudio.

En el año 2012 encontramos una investigación en el observatorio SIRTA, en París [9]. En este estudio se utilizaron un conjunto de instrumentos de teledetección activos y pasivos, junto con unos sensores in situ de este observatorio durante seis meses. Esto se realizó para documentar los procesos radiactivos, microfísicos y dinámicos de la niebla continental. En esta investigación se presentan varios modelos y diferentes escenarios posibles para explicar la formación, desarrollo y disipación de los tres principales eventos de niebla y cuantificar el impacto de cada proceso. De este estudio se observa cómo las gotas de nubes que caen, junto con la llovizna afectan a la formación y disipación de niebla. También se puede apreciar cómo la microfísica y la dinámica de las nubes se ven afectadas por un enfriamiento radiactivo en nubes a mayor altitud. Por último, también se aprecia como la velocidad de viento horizontal y la energía cinética turbulenta afectan a los procesos en la nube, como la mezcla turbulenta, la velocidad vertical de las partículas y la evaporación de la base de la nube. Aparte, se destaca como la turbulencia en la nube afecta a la distribución del tamaño de las gotas y su distribución vertical, y por ello a la visibilidad cercana a la superficie.

Al año siguiente se realizó una investigación sobre el uso de los árboles de decisión para la predicción del tiempo [10]. Se decidieron estos modelos dado que la evaluación del árbol de decisión puede ser cuantificada y su uso es simple. Este modelo se usará para predecir eventos niebla, lluvia y truenos, ingresando en la entrada datos de promedios de temperatura, humedad y presión.
Los resultados del experimento muestran que de 72 pruebas instanciadas, 46 se clasificaron correctamente, lo que proporciona un Coeficiente kappa de Cohen de 0.584. A raíz de estos resultados se sugiere mejorarlos tomando más atributos en el modelo e incrementando los datos de entrenamiento. Estos resultados nos muestran además que para aprovechar el potencial de una gran cantidad de datos, el árbol de decisión puede ser utilizado en la predicción de una variable dependiente como la niebla y la lluvia.

En 2015 se llevó a cabo un estudio para desarrollar un modelo de pronóstico inmediato de eventos niebla para la seguridad del tráfico en carreteras en una región desértica costera, Dubai [11]. Para lograr este objetivo se hizo uso del algoritmo árboles de decisión y se realizaron observaciones de alta frecuencia tomadas en estaciones meteorológicas automáticas como base de datos para el análisis de patrones útiles.
A partir de los resultados, se observa cómo los árboles de decisión aumentaron las habilidades de predicción en comparación al modelo WRF o el PAFOG. Asimismo, estos también mejoraron aún más la integración de la salida de los modelos numéricos acoplados de predicción de niebla en la base de datos de entrenamiento. Con esta técnica se obtuvieron una probabilidad de detección, el índice de falsas alarmas y la puntuación de habilidad de Gilbert de 0.88, 0.19 y 0.69 respectivamente.
A partir de estos resultados se observa que la mejor predicción se obtiene durante las primeras seis horas de predicción, mientras que para tiempos superiores, los modelos numéricos acoplados son superiores.

En el año 2017 encontramos una investigación de predicción de eventos de baja visibilidad en aeropuertos mediante técnicas de regresión [12].
En el estudio se examina el desempeño de varios regresores para el aeropuerto de Valladolid, como el SVR, las máquinas de aprendizaje automático y procesos gaussianos. También se estudiará hasta que punto pueden ser preprocesadas las variables meteorológicas de entrada con la transformada Wavelet.
Los resultados muestran una alta eficiencia para predecir eventos de baja visibilidad. Destaca el proceso graussiano como el mejor con un 98 % de predicciones acertadas cuando el rango de visibilidad es superior a 100 metros, frente al 80 % cuando es inferior. También se observa que todos los algoritmos se ven afectados por condiciones meteorológicas extremas, aunque muestran mejores resultados si se incluyen datos meteorológicos de una torre vecina y si se preprocesa usando una transformada Wavalet.

Acercándonos a la actualidad, encontramos en 2018 una investigación sobre la predicción de eventos de baja visibilidad debido a tres clases de niebla (niebla, neblina y despejado) mediante clasificación ordinal, debido a esta naturaleza de los eventos de baja visibilidad [13].
En este trabajo se propone un modelo híbrido de predicción para eventos diarios de baja visibilidad, que combina ventanas de tamaño fijo y dinámicas, y adapta su tamaño de acuerdo con la dinámica de la serie temporal. En este estudio se toman como variables meteorológicas en el aeropuerto de Valladolid: temperatura, humedad relativa, velocidad y dirección del viento, QNH, y como variable objetivo, el RVR, que nos permite caracterizar los eventos niebla en aeropuertos. En este estudio se van a utilizar la ventana fija y dos dinámicas, la ventana dinámica basada en el cambio de etiqueta y la ventana dinámica basada en el cambio de varianza.

Asimismo, se van a utilizar el modelo POM, dos métodos de máquinas de vectores de soporte, el método de regresión ordinal y el SVORIM y el KDLOR.
A partir de los resultados obtenidos, el KDLOR es la máquina de aprendizaje con un mejor rendimiento, especialmente para las clases minoritarias, mostrando un comportamiento razonable. Por los resultados de este modelo, se considera que el modelo híbrido propuesto podría mejorar la seguridad y rentabilidad de las operaciones en aeropuertos afectados por eventos de baja visibilidad.

En el año 2020 se realizó un estudio sobre la aplicación de la LSTM, un tipo de red neuronal que utiliza la puerta de salida y de memoria a corto plazo, para predecir eventos de niebla a partir de datos meteorológicos [14].
En este estudio, se propone un marco con una red LSTM y una capa totalmente conectada para predecir eventos niebla a corto plazo. Para lograr esto se utilizaron cuatro conjuntos de datos meteorológicos obtenidos mediante observaciones horarias.
Para evaluar la eficacia del modelo, se compararán sus resultados obtenidos en las cuatro bases de datos con los de los modelos AdaBoost, el KNN y el CNN.
A partir de los resultados observados, el LSTM logra una puntuación superior a las de KNN, AdaBoost y CNN, especialmente en Puntuación de Amenazas, superando al mejor algoritmo de aprendizaje automático tradicional.
Por lo tanto, se demuestra que con este modelo se mejora la actuación prediciendo eventos niebla a corto plazo.

Finalmente, este mismo 2022 se ha llevado a cabo un estudio sobre la aplicación de métodos de regresión y clasificación para predecir eventos niebla [15].
Para lograrlo toman datos reales de la estación meteorológica de Mondoñedo y establecen umbrales de clasificación en función de la visibilidad. En este estudio se han considerado las variables meteorológicas: Precipitación acumulada, temperatura del aire, presión atmosférica, temperatura de rocío y del suelo, radiación solar global, humedad relativa, salinidad, visibilidad y dirección y velocidad del viento.
Debido a que esta base de datos se encuentra desbalanceada se aplica la técnica de sobremuestreo y las de submuestreo CoNN, NCR, Tomek Links TL y el RUS para solucionarlo. Aparte de estas, se utilizará una tercera técnica híbrida entre ambas para solucionarlo.
En este estudio se han utilizado los métodos de conjunto: AdaBoost, gradient boosting, Bagging y el RF. En ANN se han utilizado el MLP y el ELM. En métodos lineales se han usado el LREG, el ERG y el GLM. Por último, se han utilizado otros como el proceso gaussiano, el KNN, el árbol de decisión y el Gaussian Naive Bayes.
Estos modelos serán evaluados mediante las métricas raíz del error cuadrático medio, el error medio absoluto y el coeficiente de determinación.

A partir de estos resultados, se obtiene que, para regresión, los métodos ANN son los más apropiados, especialmente el MLP. Asimismo, se revela que los métodos están influenciados de forma distinta con respecto a la estandarización y normalización de las variables de entrada. Por último, hay que destacar los peores resultados de los métodos lineales, sobre todo el EREG.
En cuanto al problema de clasificación, destacan los métodos de conjunto como los más apropiados, especialmente el GB y el RF. Además, se observa que la técnica de balanceo no condiciona los resultados (para la tarea de regresión no fue necesario balancear dado que no se ve afectada por el desbalanceo de las muestras).
Por último, cabe señalar que los clasificadores nominales obtienen mejores resultados frente a los ordinales, no obteniendo buenos resultados con los ordinales para clasificar eventos de baja visibilidad.

Como se ha podido comprobar a partir de todos los estudios analizados en este capítulo, podemos observar que desde hace décadas, ya en los años ochenta, eran conscientes de la importancia de poder predecir los eventos meteorológicos y de visibilidad. Desde entonces hemos podido analizar las diferentes estrategias que han ido descubriendo y estudiando nuestros antecesores para lograr este objetivo, y el que continuamos en este proyecto.

Capítulo 3

Estudio teórico de las técnicas de clasificación, Análisis exploratorio e Implementación

El objetivo de este proyecto es desarrollar mediante diversas técnicas de aprendizaje automático una serie de clasificadores para la predicción de eventos niebla.
El concepto de aprendizaje automático (Machine Learning, ML) surgió en 1943 a partir de un modelo de redes neuronales, al intentar mapear de forma matemática la cognición humana. En 1950 Alan Turing propuso el Test Turing para comprobar si una máquina es inteligente o no, en función de si tenía la capacidad de convencer a un humano de que ella lo es también. Finalmente, poco después en Darthmouth College surgió de forma oficial la inteligencia artificial. A partir de este momento comenzaron a surgir multitud de programas y algoritmos capaces de realizar infinitud de tareas complejas, desde planificar rutas, hasta reconocimiento de voz, etcétera [16].
El aprendizaje automático es un subcampo de la inteligencia artificial, que se define como la capacidad de una máquina de imitar un comportamiento humano inteligente, que da a los ordenadores la capacidad de aprender de la experiencia y realizar tareas complejas de un modo similar al humano, sin estar explícitamente programados [17].
El aprendizaje automático comienza preparando los datos sobre los que entrenará el modelo. Una vez hecho esto se elige el modelo deseado, se le entrena y se deja que aprenda. Después, se le proporcionan los datos de evaluación y se evalúa lo preciso que es el modelo. El resultado de este proceso es un modelo capaz de realizar predicciones con diferentes conjuntos de datos.
Estos modelos pueden diseñarse para muchas funciones:

- Descriptiva: El sistema usa los datos para explicar que ocurrió.
- Predictiva: El sistema usa los datos con el fin de predecir que ocurrirá.
- Prescriptiva: El sistema usará los datos para hacer sugerencias sobre que acción tomar.

Por último, podemos clasificar nuestro sistema de aprendizaje en las siguientes subcategorías:

- Modelos de aprendizaje automático supervisado, entrenados con conjuntos de datos etiquetados, los cuales permiten que los modelos aprendan y se vuelvan más precisos con el tiempo.

- El aprendizaje automático no supervisado, un programa busca patrones en datos no etiquetados.

- El aprendizaje automático de refuerzo entrena a las máquinas a través de prueba y error para tomar la mejor acción mediante el establecimiento de un sistema de recompensas.

3.1. Redes Neuronales (Neural Network, NN)

En este proyecto vamos a diseñar clasificadores de diversa naturaleza, como redes neuronales [18]. Las NN son un método de aprendizaje profundo y consisten en una serie de capas de neuronas artificiales interconectadas, y alimentadas por funciones de activación. Como se ve en la Figura 3.1, cada una de estas neuronas recibe un producto de entradas y pesos aleatorios a los que añade un sesgo estático único y lo pasa a través de la función de activación para decidir el valor final de salida de la neurona. Una vez se genera la salida de la última capa, se calcula la función de pérdida y se realiza una retroprogación, en la que se ajustan los pesos para obtener un valor óptimo de estos y una mínima pérdida. Por lo tanto, estos pesos son aprendidos por ML y se autoajustan en función de la diferencia entre salidas predichas y valores reales.

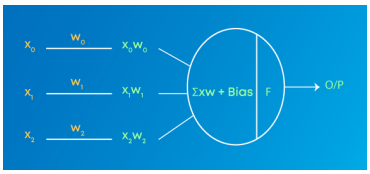


Figura 3.1: Propagación a través de una red neuronal.

Como se ve en la Figura 3.2, una red neuronal está compuesta por una capa de entrada con las dimensiones del vector de entrada, una capa oculta con todos los nodos intermedios, que toma el conjunto de pesos de entrada y produce una salida mediante una función de activación, y la capa de salida de la red neuronal.

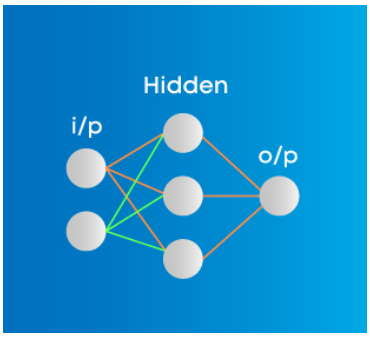


Figura 3.2: Capas de una red neuronal.

En función de una serie de parámetros como su estructura, el flujo de datos, el número de neuronas o sus densidades, se distinguen diferentes tipos de redes neuronales. Empezamos hablando del perceptrón, el modelo más simple y antiguo de neurona capaz de realizar ciertos cálculos a partir de unos datos de entrada, aplicar una función de activación y producir una salida final. Como se puede ver en la Figura 3.3, se trata de un algoritmo de aprendizaje binario supervisado que separa el espacio de entrada en dos categorías mediante el hiperplano (3.1):

$$W^T \cdot X + b_i = 0 \tag{3.1}$$

Presenta las ventaja de ser capaz de implementar puertas lógicas como ”AND”, ”OR”, o ”NAND”. Sin embargo, sólo puede aprender problemas separados linealmente.

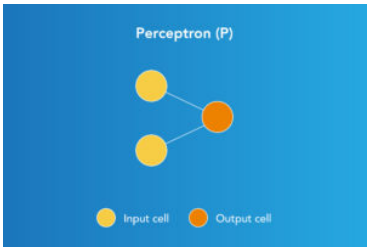


Figura 3.3: Modelo Perceptrón.

También podemos encontrar las NN prealimentadas con una o dos capas, en las que como se ve en la Figura 3.4, los datos viajan en una dirección a través de los nodos artificiales a los nodos de salida. El número de capas depende de la complejidad de la función de activación alimentada por el producto de entradas por pesos y se propaga únicamente hacia adelante.

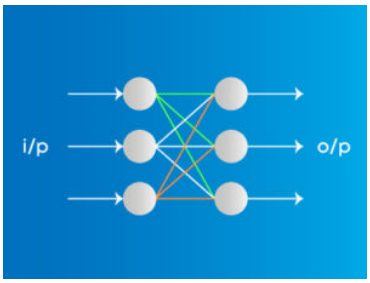


Figura 3.4: Red Neuronal Prealimentada.

Estas redes presentan ventajas como su menor complejidad, y ser fáciles de diseñar y mantener. Sin embargo, al carecer de capas densas y retropropagación, su capacidad de generalización no es elevada.

Asimismo, tenemos el perceptrón multicapa, MLP, un algoritmo de aprendizaje supervisado, que al ser entrenado por una base de datos aprende una función. A partir de un conjunto de características y un objetivo, nuestro clasificador puede aprender una función no lineal con la que realizar una clasificación o regresión. Como se puede ver en la Figura 3.5, en un MLP podemos encontrar las tres capas, una de entrada, una o más no lineales capas ocultas y por último, una capa de salida. Estas redes presentan ventajas como su capacidad para aprender modelos no lineales y en tiempo real. Sin embargo, presentan desventajas como que las capas ocultas tienen una función de pérdida no convexa donde existe más de un mínimo local. Por ello, diferentes inicializaciones conducen a precisiones de validación distintas. Además, el MLP requiere ajustar una serie de parámetros como el número de neuronas, capas e iteraciones, y es sensible al escalado de datos.

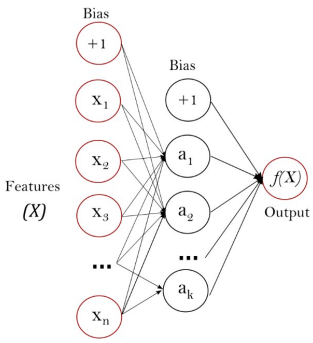


Figura 3.5: MLP Una Capa Oculta

También podemos encontrar las NN Radial Basis Function (RBF). Como se puede ver en la Figura 3.6, estas constan de un vector entrada, una capa de neuronas de base radial y una salida con un nodo por categoría. La clasificación se realiza midiendo la similitud entre la entrada y datos de entrenamiento. Si son iguales las muestras se genera un valor entre 0 y 1, en función de la distancia euclídea entre ambos, cayendo a 0 a medida que la distancia crece.

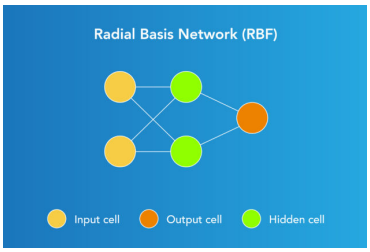


Figura 3.6: RBF Neuronal Network

Por último, encontramos las NN recurrentes. Estas son capaces de guardar la salida de una capa, retroalimentarla a la entrada y ayudar en la predicción. Estas redes cuentan con una primera capa prealimentada, seguida por una recurrente que recuerda parte de la información con una función memoria. Si la predicción es incorrecta, se emplea la tasa de aprendizaje para realizar pequeños cambios para realizar correctamente la predicción en la retropropagación.
Entre sus ventajas destacan modelar datos secuenciales, donde se supone que cada muestra depende de las históricas. En cuanto a sus desventajas destacan los problemas de desaparición o repentino crecimiento del gradiente.

Estas han sido las NN más importantes, sin embargo, como podemos ver en la Figura 3.7 podemos encontrar otras como los modelos secuencia a secuencia, con dos NN recurrentes trabajando a la vez, un codificador procesando la entrada y un decodificador obteniendo la salida. También, como se ve en la Figura 3.8, tenemos la red neuronal modular, que contiene un número de redes diferentes independientes, realizando subtareas para lograr la salida.

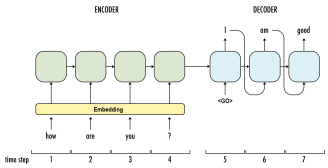


Figura 3.7: Sequence To sequence Neuronal Network

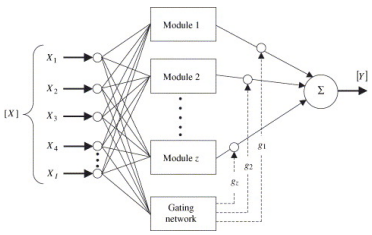


Figura 3.8: Red Neuronal Modular

En cuanto a nuestro proyecto, se van a utilizar las NN Narrow, Medium, Wide, Bilayared y Trilayared. Cada una de estas consisten en una NN prealimentada completamente conectada para clasificación. Esto significa que cada neurona de una capa está conectada a todas las de

otra. Cada capa multiplica la entrada por los pesos, añade un sesgo y termina con una función de activación. La capa final o función de activación softmax produce la salida de la red, o lo que es lo mismo, los resultados de clasificación o etiquetas predichas.
Una vez ya conocemos más sobre las redes neuronales, vamos a explicar las ecuaciones matemáticas que hay detrás de una red neuronal [19]:
Empezamos explicando el proceso de envío de la información a través de la red hacía adelante: En este proceso, las neuronas de dos capas completamente conectadas, obtienen la salida mediante la suma ponderada de las entradas a esta.

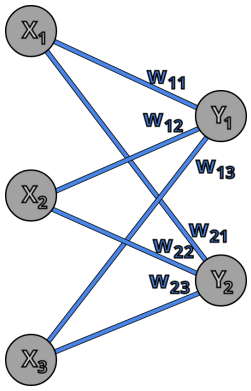


Figura 3.9: Salida capa Red Neuronal.

Por ejemplo, como se aprecia en la Figura 3.9, las ecuaciones de las neuronas de la segunda capa serían las Ecuaciones (3.2) y (3.3):

$$Y_1 = W_{11} \cdot X_1 + W_{12} \cdot X_3 + W_{13} \cdot X_3 \tag{3.2}$$

$$Y_2 = W_{21} \cdot X_1 + W_{22} \cdot X_3 + W_{23} \cdot X_3 \tag{3.3}$$

Las Ecuaciones (3.2), y (3.3) se juntan y se expresan en la matriz (3.4):

$$\begin{pmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{22} & W_{23} \end{pmatrix} \cdot \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} = \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \tag{3.4}$$

Por último, solo falta aplicar la función de activación de la capa de salida. Si consideramos una capa de salida n, la salida final sería la siguiente Expresión (3.5):

$$Salida_n = activación(Pesos_{n-1} \cdot Salida_{n-1}) \tag{3.5}$$

Ahora vamos a explicar la retropropagación. Esta se usa para enviar el error a las capas bajas de la red para que aprendan.

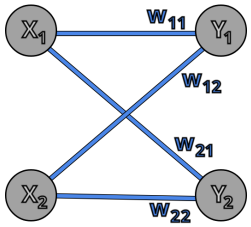


Figura 3.10: Capas Retropropagación.

Si consideramos el ejemplo de la Figura 3.10, vemos que cada neurona contribuye al error de todas las neuronas de la capa de salida, obteniendo las Ecuaciones de error, (3.6) y (3.7):

$$E_1 = \frac{W_{11}}{W_{11} + W_{12}} \cdot EY_1 + \frac{W_{21}}{W_{21} + W_{21}} \cdot EY_2 \tag{3.6}$$

$$E_2 = \frac{W_{12}}{W_{11} + W_{12}} \cdot EY_1 + \frac{W_{22}}{W_{21} + W_{21}} \cdot EY_2 \tag{3.7}$$

Ahora, al igual que hicimos en la Ecuación (3.4) vamos a representar estas ecuaciones juntas en forma matricial. Cabe resaltar que el denominador actúa como un factor normalizador, por lo que no nos tenemos que preocupar por él sin perder información importante:

$$E_X = \begin{pmatrix} W_{11} & W_{21} \\ W_{12} & W_{22} \end{pmatrix} \cdot \begin{pmatrix} E_{Y_1} \\ E_{Y_2} \end{pmatrix} \tag{3.8}$$

Al observar la Ecuación (3.8) podemos observar como las Ecuaciones (3.8) y (3.4) se relacionan mediante la siguiente forma:

$$\begin{pmatrix} W_{11} & W_{21} \\ W_{12} & W_{22} \end{pmatrix} == \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix}^T \tag{3.9}$$

Finalmente, a partir de la relación dada por la Ecuación (3.9) obtenemos el algoritmo de retropropagación:

$$E_{n-1} = W_n^T \cdot E_n \tag{3.10}$$

Por último, una vez obtenemos los errores de la red, la enseñamos a actualizar los pesos y con ello minimizar la función de error. Para ello, vamos generando valores aleatorios y revisamos si la pendiente (3.11) de la función va colina abajo como vemos en la Ecuación (3.11)

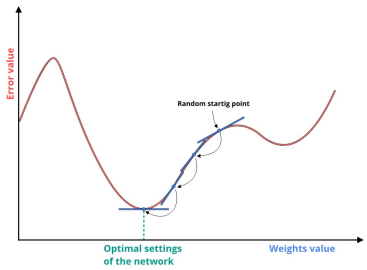


Figura 3.11: Imagen de como obtener la pendiente de la función de error.

$$W_n = W_n - \frac{\partial E_{n+1}}{\partial W_n} \tag{3.11}$$

Además de la derivada necesitamos conocer el ratio de aprendizaje, el cuál es un número entre 0 y 1 que regula cómo de grandes son los pasos que se toman yendo colina abajo. Este lo podemos calcular mediante la Ecuación (3.12):

$$W_n = W_n - LearningRate \cdot \frac{\partial E_{n+1}}{\partial W_n} \tag{3.12}$$

3.2. Clasificador de Conjunto (Ensemble)

El siguiente clasificador usado en el estudio es el de conjunto [20]. Estos modelos buscan el mejor rendimiento mediante la combinación de predicciones de múltiples modelos. Podemos encontrar diferentes tipos de métodos de conjunto. El Bagging se usa con el objetivo de reducir la varianza, realizando ajustes en las muestras de los datos y promediando las predicciones. Se trata de métodos muy simples sin necesidad de adaptar el algoritmo base. Por ello, proporcionan un método de reducir el sobreajuste y funcionan mejor con modelos fuertes y complejos, en contraste con los métodos Boosting que suelen funcionar mejor con modelos débiles. Un modelo débil es aquel que se desempeña ligeramente mejor que un modelo ingenuo [21], es decir, que no usa ningún conocimiento o aprendizaje para hacer una predicción [22]. Por lo tanto, el modelo débil logra una exactitud de predicción ligeramente superior al 50 %. En este modelo, como se puede ver en la Figura 3.12, se crea una serie de subconjuntos a partir de la base de datos original, seleccionando observaciones con reemplazo. En cada uno de ellos se crea un modelo de cada conjunto de entrenamiento aprendido en paralelo de forma independiente. Finalmente, se determinan las predicciones de salida combinando las de los modelos, reduciendo la varianza al introducir aleatorización.

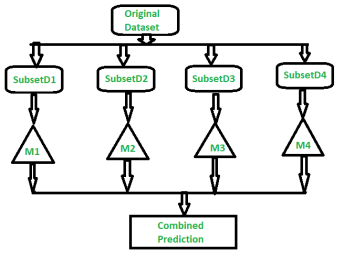


Figura 3.12: Ejemplo de Clasificador de Conjunto de tipo Bagging

El clasificador Boosting agrega conjuntos de forma secuencial corrigiendo las predicciones hechas por modelos anteriores y generando predicciones promedio. La clave de este modelo es corregir los errores de predicción, agregando al modelo conjuntos de forma secuencial, que intentan corregir las predicciones del modelo anterior y así sucesivamente, como podemos ver en la Figura 3.13. Esto suele implicar usar árboles de decisión muy simples, cuyas predicciones se promedian proporcionalmente a su desempeño. El objetivo es crear un modelo fuerte a partir de débiles.

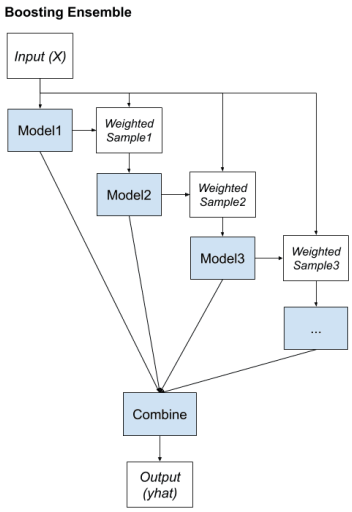


Figura 3.13: Ejemplo de Clasificador de Conjunto de tipo Bagging

El siguiente algoritmo es el RUS Boost, cuyo objetivo es aliviar el problema del desbalanceo. Para ello combina muestras de datos y Boosting, proporcionando un método simple y eficiente para datos de entrenamiento desbalanceados. Este método destaca por tener un desempeño favorable, mayor rendimiento, menos costoso y más rápido frente a otros modelos, como el híbrido SMOTE Boost. Sin embargo, presenta el inconveniente de pérdida de información, aunque se supera en gran medida con el Boosting. En este algoritmo, como vemos en la Figura 3.14, primero se inicializan las muestras. Entonces, se entrenan T modelos débiles y se les aplica submuestreo. Se provee de muestras y pesos a la clase débil y nos devuelve una hipótesis débil. Por último, calculamos las pseudo pérdidas, que nos definen como de lejos está la predicción del

modelo del valor objetivo [23]. Una vez las conocemos, actualizamos los pesos y la distribución de estos, la normalizamos, y después de T iteraciones obtenemos la hipótesis final.

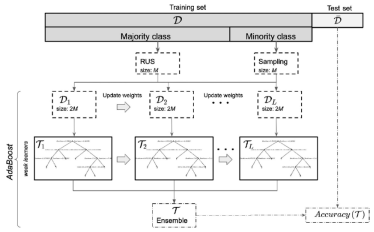


Figura 3.14: Ejemplo de Clasificador de Conjunto de tipo RUSBoosting

Aparte de estos, podemos encontrar el subspace discriminant, un algoritmo de aprendizaje que combina predicciones de múltiples árboles de decisión entrenados con subconjuntos de datos de entrenamiento diferentes. La selección aleatoria de características en los subespacios es la principal deficiencia, pudiendo darse una selección pobre de características, volviéndose pobre la decisión final del conjunto.
Con el fin de disminuir este inconveniente se utiliza el método de votación mayoritaria. Esta técnica utiliza cada clasificador para predecir por separado la variable objetivo y posteriormente se combinan las salidas para adoptar la decisión final.
Por último, encontramos el método subspace KNN. Se trata de un método con solo dos parámetros, la distancia entre una muestra de prueba y entrenamiento, y el número de vecinos k. Sin embargo, tiene desventajas como no considerar el peso de diferentes atributos, la perturbación del espacio, su sensibilidad al espacio de atributos de entrada y el gran coste de cálculo.
Estos clasificadores empiezan obteniendo muestras aleatorias de las características. Después obtienen unas muestras aleatorias de tamaño *n* y construyen el clasificador KNN y se calcula la precisión de este utilizando las características utilizadas en la construcción del modelo para comparar las predichas con las objetivo.

3.3. Árboles de Decisión (Decision Trees)

El siguiente tipo de clasificador es el árbol de decisión [24]. Es una técnica de aprendizaje supervisado, mayormente usada para clasificación, aunque también para regresión. Se trata de un clasificador con estructura en forma de árbol, donde los nodos representan las características de la base de datos, las ramas representan las reglas de decisión y cada nodo hoja representa la salida. Estos nodos pueden ser de decisión, usados para tomar decisiones y contienen más ramas y hojas que no contienen más. El árbol de decisión se considera una representación gráfica para obtener todas las posibles soluciones del problema. Para predecir una clase a partir de una base de datos, como se puede ver en la Figura 3.15, se parte del nodo raíz del árbol y se compara el valor del atributo raíz con el de la base de datos real. A partir de esta sigue la rama y salta al siguiente nodo. Este proceso sigue repitiéndose hasta llegar al nodo hoja final.

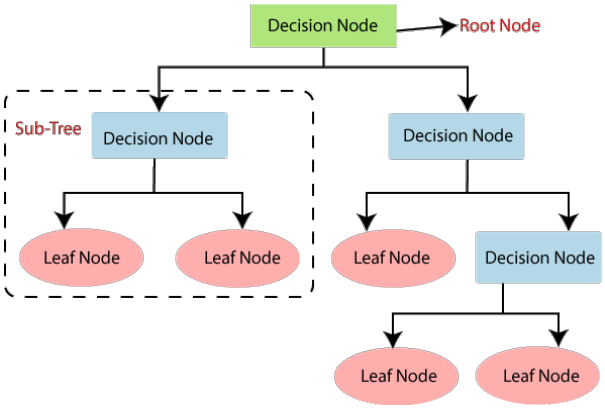


Figura 3.15: Ejemplo de Clasificador Árbol de Decisión

Este método presenta las ventajas como ser simple de entender, su utilidad para la resolución de problemas de toma de decisiones, ayudar a pensar todas las posibles salidas del problema, además de requerir una menor limpieza de datos comparado con otros métodos. Sin embargo, contiene muchas capas, volviéndose complejo. Además, tiene un problema de sobreajuste, y la complejidad se incrementa con el número de clases.
En función del tipo de variable objetivo podemos encontrar los árboles de variable decisión categorical, con variable objetivo categórica, y continua, con variable objetivo continua.
En cuanto a nuestro proyecto, se van a desarrollar los clasificadores árboles de decisión: Coarse, Medium y Fine. Como se puede ver en la Figura 3.16, para realizar una predicción comienzan en el nodo superior y en cada decisión los predictores deciden la rama que tomar. Una vez se alcanza el nodo hoja, se clasifica el dato como 0 o 1.

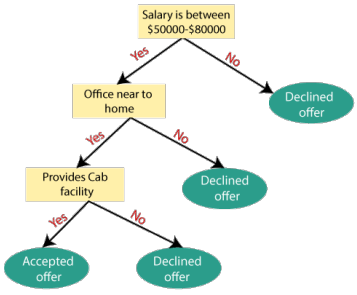


Figura 3.16: Ejemplo 2 de Clasificador Árbol de Decisión

Finalmente, vamos a desarrollar un ejemplo matemático de estos modelos [25]. En general, los nodos raíz o primeros atributos de prueba dependen de medidas como la entropía, ganancia de información, coeficiente de Gini, el ratio y reducción de ganancia y la prueba χ^2. Estos criterios calculan sus valores para cada atributo, los ordenan y los colocan en el árbol siguiendo un orden. La entropía es una medida de la aleatoriedad de la información procesada, cuanto mayor es esta, más difícil de llegar a una conclusión con dicha información. Esta se puede

calcular para un atributo mediante la Ecuación (3.13):

$$E(S) = \sum_{i=1}^{c} p_i \cdot \log_2(p_i) \tag{3.13}$$

Donde calculamos la probabilidad de un evento i, p_i de un estado como:

$$p_i = \frac{x_i}{|D|} \tag{3.14}$$

La entropía para múltiples atributos la calcularíamos mediante la Ecuación (3.15):

$$E(T, X) = \sum_{c \in X} P(c) \cdot E(c) \tag{3.15}$$

Donde T es el estado actual y X es el atributo seleccionado.
La ganancia de información es una propiedad estadística que mide como de bien separa un atributo las muestras de entrenamiento de acuerdo a su meta de clasificación. El objetivo es construir un árbol de decisión con la ganancia de información más alta y la menor entropía. Podemos calcular la ganancia de información con la siguiente Fórmula (3.16):

$$Ganancia\ de\ Información(T, X) = Entropía(T) - Entropía(T, X). \tag{3.16}$$

El coeficiente de Gini es la función de coste usada para evaluar las divisiones en la base de datos. Se calcula substrayendo el sumatorio de las probabilidades al cuadrado de cada clase mediante la Ecuación (3.17):

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2 \tag{3.17}$$

El ratio de ganancia es la relación entre la ganancia de información y la información intrínseca y se obtiene mediante la Ecuación (3.18):

$$Ratio\ de\ Ganancia = \frac{Información\ de\ Ganancia}{Información\ de\ división} \tag{3.18}$$

La información de división es un número entero positivo que describe el potencial valor de dividir una rama de un nodo. Tras sustituir en la Ecuación (3.18) obtenemos (3.19):

$$Ratio\ de\ Ganancia = \frac{Entropía(Antes\ de\ División) - \sum_{K}^{j=1} Entropía(j, Tras\ División)}{\sum_{j=1}^{K} W_j \cdot \log_2 \cdot w_j} \tag{3.19}$$

La reducción en la varianza es un algoritmo usado para variables objetivo continuas (problemas

de regresión). El algoritmo usa una fórmula estándar para elegir la mejor división (3.20):

$$Varianza = \frac{\sum(X - \bar{X})^2}{n} \tag{3.20}$$

Dónde $\bar{X}$ es la media de los valores, X es el valor actual y n es el número de valores.
Para calcular la varianza, primero calculamos la varianza en cada nodo y después la varianza para cada división como la media ponderada de la varianza de cada nodo.
Por último, la prueba χ^2 se encarga de descubrir la significancia estadística entre las diferencias entre los sub-nodos y el nodo padre. Se calcula como el sumatorio del cuadrado de las diferencias entre las frecuencias observadas y esperadas de la variable objetivo mediante la Ecuación (3.21):

$$\chi = \sum \frac{(O - E)^2}{E} \tag{3.21}$$

Donde, O es el resultado obtenido y E el esperado. Para calcular esta métrica, primero la calculamos para un nodo individual mediante el cálculo de la desviación para todas las clases. A continuación calculamos χ usando un sumatorio de todos los χ de todas las clases de cada nodo de la división.

3.4. K-Vecinos más cercanos (K-Nearest Neighbor, KNN)

También vamos a diseñar clasificadores KNN, un algoritmo de ML supervisado fácil de implementar, tanto para clasificación como regresión [26]. Este algoritmo funciona asumiendo la similitud entre un nuevo dato y los casos disponibles, clasificando al nuevo dato en la categoría más similar a él de entre las disponibles. Asimismo, este es un algoritmo no paramétrico, es decir, no realiza suposición alguna sobre los datos subyacentes. Además, no aprende del conjunto de entrenamiento de forma inmediata, sino que almacena la base de datos y lleva a cabo una acción en ella a la hora de clasificar. Por lo tanto, estos modelos solamente almacenan la base de datos y al llegar nuevos valores la usa para clasificar, como podemos ver en la Figura 3.17:

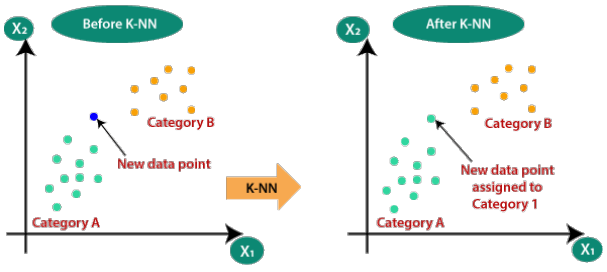


Figura 3.17: Ejemplo 2 de Clasificador KNN

La importancia de este algoritmo recae en que es uno de los más fáciles de implementar, permitiendo identificar fácilmente la clase de una base de datos particular. Primero se selecciona el número k de vecinos y se calcula su distancia euclídea. A continuación, tomamos los k vecinos más cercanos y asignamos los puntos de dato a la categoría que tiene más de ellos. Esta técnica destaca por ventajas como su simpleza, robustez al ruido y más efectivo ante bases de entrenamiento grandes. Por otro lado, este algoritmo puede ser complejo y de alto coste computacional en el cálculo de las distancias. En este proyecto se van a usar los modelos: Fine, Medium, Coarse, Cubic y Weighted KNN.
Estos clasifican nuestros puntos a partir de unas métricas que determinan la distancia a los vecinos más próximos, y en función de la clase de los vecinos más próximos los clasifican como una clase u otra [27]. Estos modelos determinan la distancia a los vecinos más cercanos mediante unas métricas y calculan la posibilidad de que sean similares. La clasificación se basa en cuáles comparten posibilidades más altas. La función distancia puede ser euclídea, Minkowski o de Hamming. Como se ve en la Figura 3.18, la euclídea es la distancia más corta entre dos puntos.

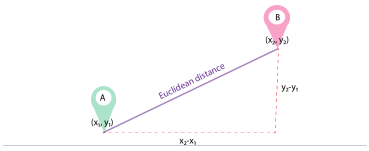


Figura 3.18: Distancia Euclidiana en clasificador KNN

Este es el modo más común de obtener la distancia. Si tenemos dos puntos (X, Y) y (a, b), obtenemos la distancia euclídea como la Ecuación (3.22):

$$distancia((X, Y), (a, b)) = \sqrt{(X - a)^2 + (Y - b)^2} \tag{3.22}$$

Una vez calculada la distancia, la entrada X se asigna a la clase con una probabilidad mayor mediante la Fórmula (3.23):

$$P(Y = j|X = x) = \frac{1}{K} \sum_{i \in A} I(Y^{(i)} = j) \tag{3.23}$$

La distancia Minkowski es una generalización de la distancia euclídea. Se define como la distancia entre dos puntos en el espacio vectorial normalizado (espacio real de dimensión N). Si consideramos dos puntos $P_1 : (X_1, X_2, ..., X_n$ y $P_2 : (Y_1, Y_2, ..., Y_n)$, la distancia Minkowski vendría dada por la Ecuación (3.24):

$$Distancia\ Minkowski\ = \sqrt[p]{(x_1)^p + (x_2 - y_2)^2 + ... + (x_n - y_n)^p}, \tag{3.24}$$

donde p es la dimensión.
Esta fórmula se puede representar de forma gráfica como en la Figura 3.19:

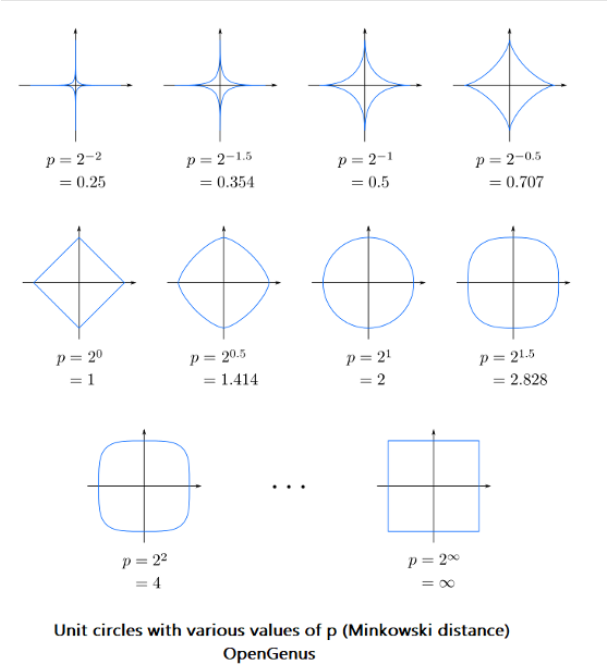


Figura 3.19: Representación visualización fórmula Minkowski.

Por último, la distancia Hamming, $d(a, b)$, es una métrica para comparar dos cadenas de datos binarios. Se calcula realizando la operación XOR (disyunción exclusiva), $a \oplus b$, entre las dos cadenas a y b, y contando el número total de unos en la cadena resultante (número de bits diferentes).

3.5. Máquinas de vectores de soporte (Support Vector Machines, SVM)

El último tipo de clasificadores de este estudio es la SVM, un algoritmo de ML supervisado usado para regresión, detección de valores atípicos y especialmente para clasificación [28]. Como se puede ver en la Figura 3.20, el objetivo de este algoritmo es encontrar un hiperplano que clasifique los puntos de datos. La dimensión de este hiperplano depende del número de características.

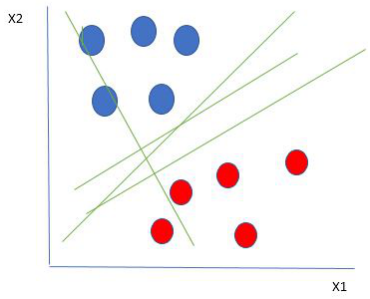


Figura 3.20: Ejemplo 2 de Clasificador KNN

A partir de la Figura 3.20 queda claro que existen múltiples líneas que segregan nuestros puntos. Para seleccionar la mejor hay que elegir la que permita una mayor separación entre clases, como podemos apreciar en la Figura 3.21:

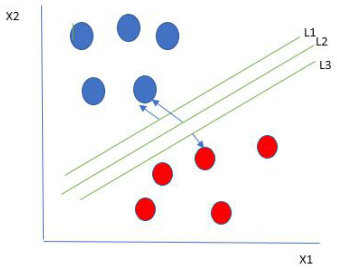


Figura 3.21: Ejemplo de selección del mejor hiperplano en SVM.

En cuanto a las ventajas de este modelos, destaca su efectividad para casos de alta dimensionalidad, su eficiente memoria ante un subconjunto de datos de entrenamiento y cuando el número de dimensiones es mayor que el de muestras. Sin embargo, si el número de características es mayor que el de muestras es mejor evitar sobreajustes o que el SVM provee estimaciones de probabilidad directamente.
Podemos diferenciar dos tipos de clasificadores SVM. Primero tenemos los lineales, si los datos están ordenados linealmente y se pueden separar mediante un hiperplano con una línea recta. Por el contrario, si estos no se pueden, tenemos los no lineales. Para estos casos, necesitamos añadir una tercera dimensión z y utilizar kernels para convertir los datos en separables. El SVM Kernel es una función que toma un espacio de entrada de baja dimensión y lo transforma en uno de más alta, convirtiendo un problema no separable en uno que lo es. Es especialmente útil para problemas no lineales. Podemos diferenciar los Kernel lineal, si los datos son linealmente separables, estos son los más comunes. Además, tenemos los Kernel no lineales, en los que no pueden ser clasificados con una función lineal. Podemos diferenciar el polinómico, exponencial, laplaciano, hiperbólico, base radial, etcétera.
En nuestro proyecto, vamos a utilizar el modelo lineal, SVM lineal y los no lineales, Quadratic, Cubic, Fine, Medium y Coarse Gaussian SVM. Estos modelos van a clasificar nuestras muestras

encontrando el mejor hiperplano. Al tener en nuestro estudio más de dos clases, nuestros modelos van a reducir el problema en tres subproblemas binarios, aprendiendo cada uno a distinguir su clase positiva del resto.

Por último, vamos a desarrollar la parte matemática que hay detrás de las máquinas de vectores de soporte [29].

Lo primero que tenemos que tener en cuenta al tratar con los clasificadores de máquinas de vectores de soporte es el producto escalar de vectores, como podemos ver en la Figura 3.22. Este se utiliza para obtener un valor escalar a partir de la operación de dos vectores. El producto escalar se puede definir como la proyección de un vector sobre otro, multiplicado por el producto de otro vector.

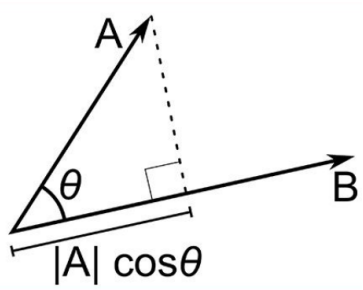


Figura 3.22: Producto Escalar SVM.

Matemáticamente se obtiene mediante la siguiente Fórmula (3.25):

$$A \cdot B = |A| \cdot \cos\theta \cdot |B| \tag{3.25}$$

Como se puede visualizar en la Figura 3.23, cuando estamos en un clasificador SVM y queremos saber si un punto aleatorio X yace al lado izquierdo o derecho del hiperplano, asumimos este punto como un vector y hacemos un vector W perpendicular al hiperplano. Consideramos que C es la distancia del vector W desde el origen hasta la frontera de decisión. Ahora tomamos una proyección de X sobre W o producto escalar. Si este es mayor que C decimos que el punto X está ubicado en el lado derecho, si es menor en el izquierdo y si es igual a cero está en la frontera de decisión.

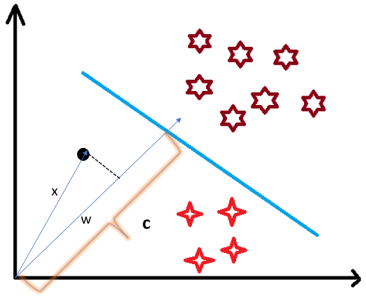


Figura 3.23: Clasificación Punto en SVM.

Podemos resumir los criterios de clasificación de un punto en un clasificador SVM en función del producto escalar en las siguientes Ecuaciones (3.26):

$$\vec{X} \cdot \vec{w} = C(\text{El punto se ubica en la frontera de decisión.}) \tag{3.26}$$

$$\vec{X} \cdot \vec{w} > C(\text{Muestras positivas.}) \tag{3.27}$$

$$\vec{X} \cdot \vec{w} < C(\text{Muestras negativas.}) \tag{3.28}$$

Ahora vamos a hablar del margen en SVM. Como ya sabemos, la ecuación de un hiperplano es (3.29):

$$w \cdot X + b, \tag{3.29}$$

donde w es un vector normal al hiperplano y b es un ajuste. Como acabamos de explicar, tenemos unas ecuaciones para clasificar un punto como positivo o negativo, a partir de ellas definimos la regla de decisión mediante la Ecuación (3.5):

$$y = \begin{cases} +1 & \text{si } \vec{X} \cdot \vec{w} + b >= 0, \\ -1 & \text{si } \vec{X} \cdot \vec{w} + b < 0. \end{cases}$$

Una vez tenemos las ecuaciones para decidir si un punto es de clase positiva o negativa, necesitamos una w y b, tales que como se visualiza en la Figura3.24, el margen tenga una distancia máxima, d.

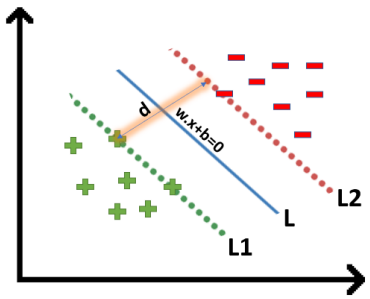


Figura 3.24: Calcular el Margen en SVM.

Para calcular d necesitamos las ecuaciones de L_1 y L_2, las cuáles vienen dadas por las Ecuaciones $L_1 = w \cdot X + b = 1$ y $L_2 = w \cdot X + b = -1$, que tienen que estar a la misma distancia de la recta L.

A continuación vamos a hablar de la función de optimización. Primero, para poder obtener esta, necesitamos considerar algunas restricciones. La restricción es, calcular la distancia d de tal forma que ningún punto positivo o negativo pueda cruzar la línea margen. Estas restricciones escritas de forma matemática quedan mediante la Ecuación (3.30):

$$\vec{w} \cdot \vec{X} + b \leq -1(\text{Para todos los puntos de las clases negativas.}) \tag{3.30}$$

$$\vec{w} \cdot \vec{X} + b \geq 1(\text{Para todos los puntos de las clases positivas.}) \tag{3.31}$$

Para simplificar estas dos restricciones en una sola (3.32), podemos decir que para que un punto sea clasificado correctamente, esta condición debería ser siempre cierta:

$$y_i(\vec{w} \cdot \vec{X} + b) \geq 1 \tag{3.32}$$

Ahora, vamos a tomar dos vectores, el primero de la clase negativa y el segundo de la positiva. Necesitamos que la distancia entre ambos $X_2 - X_1$ sea lo más corta posible. Para ello tomamos un vector w perpendicular al hiperplano y con el producto escalar calculamos la proyección del vector distancia entre los vectores sobre él, mediante la Ecuación (3.33).

$$(X_2 - X_1) \cdot \frac{\vec{W}}{||W||} \tag{3.33}$$

Dado que X_1 y X_2 son vectores soporte y descansan sobre el hiperplano, $y_i \cdot (2 \cdot X + b) = 1$. Tras sustituir esta expresión para las restricciones negativa y positiva, juntarlas en una sola y operar obtenemos la Ecuación (3.34):

$$d = \frac{2}{||W||} \tag{3.34}$$

Por lo tanto, la función que tenemos que maximizar es (3.35):

$$|arg|_{max}(W^*, b^*)\frac{2}{||W||}, tal\ que\ y_i(\vec{W} \cdot \vec{X} + b) \geq 1 \tag{3.35}$$

Una vez que hemos estudiado el problema SVM Margen duro (hard margin SVM), vamos a estudiarlo en márgenes SVM blandos:
En la vida real no encontramos bases de datos que son linealmente separables, encontraremos bases de datos casi linealmente separables o no linealmente separables. En estos casos tenemos que modificar la ecuación de arriba para que permita algunas clasificaciones erróneas mediante la Fórmula (3.36):

$$|arg|_{min}(W^*, b^*)\ \frac{||W||}{2} + c \cdot \sum_{i=1}^{n} \zeta_i \tag{3.36}$$

Para esta ecuación, como podemos visualizar en la (3.25), para los puntos clasificados correctamente, nuestra zeta será igual a 0 y para los clasificados de forma incorrecta zeta es simplemente la distancia de ese punto en particular desde su hiperplano correcto.

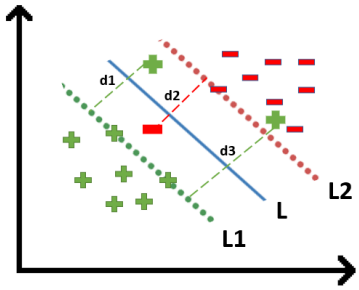


Figura 3.25: Ejemplo de Clasificación SVM margen blando.

Por lo tanto, podemos afirmar la Relación (3.37):

$$Error\ SVM = Error\ Margen + Error\ Clasificación \tag{3.37}$$

Por último, vamos a exponer los diferentes tipos de funciones Kernel:
El Kernel polinómico (3.38):

$$f(X_1, X_2) = (X_1^T \cdot X_2 + 1)^d \tag{3.38}$$

Suponiendo que tenemos dos características X_1 y X_2 e Y como variable de salida, obtenemos (3.39):

$$X_1^T \cdot X_2 = \begin{pmatrix} X_1^2 & X_1X_2 \\ X_1X_2 & X_2^2 \end{pmatrix} \tag{3.39}$$

La función Kernel sigmoidal (3.40):

$$f(X_1, X_2) = \tanh \alpha \cdot x^T \cdot y + x \tag{3.40}$$

En tercer lugar tenemos la función Kernel RBF (3.41):

$$f(X_1, X_2) = e^{\frac{-||(x_1 - x_2||^2}{2 \cdot \sigma^2}} \tag{3.41}$$

En cuarto lugar encontramos nuestra función Kernel de Bessel (3.42):

$$k(x, y) = \frac{J_{v+1} \cdot (\sigma \cdot ||x - y||)}{||x - y||^{-n \cdot (v+1)}} \tag{3.42}$$

La función de la última función Kernel que estamos desarrollando es la función Kernel Anova (3.43):

$$k(x, y) = \sum_{k=1}^{n} exp(-\sigma \cdot (x^k - y^k)^2)^d \tag{3.43}$$

Capítulo 4

Experimentos y resultados

4.1. Experimentos

En este apartado vamos a desarrollar una explicación de todos los pasos que se han realizado durante el proyecto para realizar una tarea de clasificación. Para ello, se ha procedido a la implementación de algoritmos de ML capaces de predecir los eventos de niebla y con esto la visibilidad en nuestra ubicación de estudio.
En primer lugar, partimos de una base de datos real obtenida de unos sensores instalados en tres puntos entorno al área de Mondoñedo, Galicia, España. Esta zona es famosa por la alta frecuencia de eventos de baja visibilidad, lo que la convierte en una opción ideal para que nuestros clasificadores predigan la visibilidad en la A-8 en su paso por ella. A continuación, descargamos los datos meteorológicos de modelos ECMWF, intentando tomar las mismas variables meteorológicas que las proporcionadas en la base de datos. En nuestro caso, se van a usar diez variables predictoras: *el contenido de agua de lluvia, la nubosidad, la humedad específica, la humedad relativa, el contenido líquido, la presión de superficie, la precipitación total, la temperatura de superficie y de rocío y la radiación solar neta.*
Una vez descargamos las lecturas de todas estas variables para los años 2018 y 2019, abrimos los archivos con la aplicación Panoply y analizamos los datos descargados. Podemos observar que los modelos ECMWF han tomado las medidas cada hora, mientras que los sensores de Mondoñedo cada cinco minutos. Por ello, tenemos que tomar las medidas de visibilidad de la base de datos proporcionada y realizar el promedio con estos valores, cada doce valores de visibilidad de entrada tenemos que obtener un valor promedio de visibilidad de salida, pasando así de doce valores de visibilidad cada hora a uno. Para lograr esto, como ya se explicó antes, usaremos el software Matlab.
Una vez tenemos los promedios de visibilidad para cada medición meteorológica, categorizamos estos valores en función del tipo de evento niebla que se está produciendo: poca niebla o ninguna, neblina o media niebla o mucha niebla. Con el fin de lograr esta meta, establecemos unos umbrales. Si tenemos una medición de visibilidad menor a 400 metros lo clasificamos

como mucha niebla, si hay una visibilidad entre 400 y 1500 metros como media niebla y para una visibilidad superior a 1500 metros como clase poca niebla.
Al analizar la clase de niebla en cada medición, observamos que el número de eventos poca niebla es muy superior al de las otras clases y por lo tanto nuestras 3 bases de datos se encuentran desbalanceadas. Como ya se ha analizado en la Sección 2, la base de datos desbalanceada supone un problema para la tarea de clasificación. Si no solucionáramos este problema en la etapa de entrenamiento de nuestros clasificadores, pasaríamos como entrada muchas más muestras de la clase poca niebla que del resto, y en consecuencia nuestros clasificadores no serían capaces de predecir estas otras clases, sino solamente los eventos poca niebla de forma precisa.
En este estudio para poder resolver el problema de desbalanceo se han utilizado las técnica de sobremuestreo (SMOTE) y de submuestreo (UnderSampling), las cuáles serán posteriormente analizada sobre cuál es más eficiente generando predicciones de eventos niebla. Sin embargo, antes de realizar estas dos técnicas dividimos nuestras muestras en dos, un 80 % de los datos para usarse en la fase de entrenamiento de nuestros clasificadores y el 20 % restante para la fase de prueba o validación.
La técnica de SMOTE es una técnica de sobremuestreo para generar muestras sintéticas de las clases mayoritarias en una base de datos desbalanceada [30]. Este algoritmo toma como entrada las muestras de las clases minoritarias, y a partir de las características de estas y sus vecinos más cercanos produce las muestras de salida. Además de esto, SMOTE resuelve el problema de desbalanceo aumentando las muestras de las clases minoritarias sin afectar al número de la mayoritaria.
En cuanto a la técnica de submuestreo, esta en cambio resuelve el problema de desbalanceo de la forma contraria [31]. En este algoritmo se reduce el número de muestras de las clases mayoritaria de la base de datos para obtener el mismo número de todas las clases. Además, al igual que para SMOTE con la clase mayoritaria, en submuestreo la minoritaria no se ve afectada en su número de muestras.
Una vez hemos terminado de balancear las muestras de entrenamiento mediante las dos técnicas diferentes, vamos a crear los clasificadores para ambos métodos.
Para este proyecto se van a desarrollar como se explicó en el capítulo anterior clasificadores de diferentes tipos como: los métodos de conjunto, las máquinas de vectores soporte, las redes neuronales, los k vecinos más cercanos y por último, los árboles de decisión.
A su vez, dentro de cada tipo de clasificadores se han empleado diferentes métodos, los cuáles vamos a explicar a continuación:

Métodos de Conjunto: En este estudio se han utilizado las siguientes técnicas de conjunto: Bagged Tree, Boosted Tree, RUS Boosted Tree, Subspace Discriminant y el Subspace KNN.

Máquinas de Vectores de Soporte: Para este tipo de clasificadores se ha decidido usar las variantes: Coarse Gaussian SVM, Quadratic SVM, Cubic SVM, Linear SVM y Medium Gaussian

SVM.

Redes Neuronales: En cuanto a los clasificadores neuronales, se optó por desarrollar nuestro experimento con los siguientes métodos: Bilayered NN, Medium NN, Narrow NN, Trilayered NN y la Wide NN.

K Vecinos más Cercanos: De igual manera, se han decidido desarrollar los siguientes clasificadores KNN: Coarse KNN, Cosine KNN, Cubic KNN, Fine KNN, Medium KNN y el Weighted KNN.

Árboles de Decisión: Por último, se han desarrollado los: Coarse Tree, Fine Tree y el Medium Tree.

Para esta etapa del proyecto, primero implementamos en Matlab el código de los diferentes clasificadores que hemos elegido para nuestro problema. Una vez hecho esto pasaremos como entrada las diferentes tablas de entrenamiento que hemos ido obteniendo para los diferentes puntos y métodos de balanceo. Una vez tenemos entrenados nuestros clasificadores empezamos la fase de prueba. En ésta usaremos las tablas de prueba separadas antes de iniciar el balanceo para comparar los tipos de evento que predicen nuestros clasificadores. Una vez tenemos las salidas predichas obtenemos a partir de ellos las diferentes métricas, tablas y gráficas que mostraremos y analizaremos a continuación. Sin embargo, es necesario resaltar antes de ello, que debido al elevado número de figuras y tablas, no es posible incluir todas en este capítulo respetando el límite máximo de extensión del proyecto. Por esta razón, se van a incluir en este capítulo aquellas obtenidas con los clasificadores con mejores resultados en el estudio. En cuanto al resto, se van a incluir todas las tablas e imágenes (incluidas las que se mostrarán en este capítulo) en los anexos de tablas y de imágenes.

4.2. Resultados

El estudio de los resultados obtenidos por los distintos clasificadores va a consistir en el análisis de una serie de métricas obtenidas a partir de sus predicciones, como el valor de las medias de precisión de los clasificadores, del verdadero positivo, verdadero negativo, falso positivo, falso negativo, la precisión, la exhaustividad (recall), el valor F (F1 score), la curva ROC y el AUC, las medias de las eficiencias y las desviaciones típicas, y por último, las matrices de confusión. Las gráficas (las matrices de confusión y las curvas ROC/AUC), las obtenemos en Matlab a partir de las predicciones de nuestros clasificadores y las bases de datos de prueba con los eventos reales. En cuanto a las métricas, a partir de los resultados calculamos en Matlab las siguientes [37]:

La exactitud, fracción de predicciones acertadas por nuestro modelo:

$$Exactitud = \frac{Número\ de\ predicciones\ correctas}{Número\ Total\ de\ Predicciones} \tag{4.1}$$

La precisión, la cual nos muestra la fracción de muestras predichas como positivas que realmente lo eran:

$$Precisión = \frac{TP}{TP + FP} \tag{4.2}$$

La exhaustividad, fracción de las muestras verdaderamente predichas como positivas entre todas las muestras positivas:

$$Exhaustividad = \frac{TP}{TP + FN} \tag{4.3}$$

Valor-F, métrica que combina la precisión y la exhaustividad en una única medida. Matemáticamente es la media armónica de la precisión y la exhaustividad:

$$Valor - F = 2 \cdot \frac{Precisión \cdot Exhaustividad}{Precisión + Exhaustividad} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{4.4}$$

La exactitud (Accuracy), es la fracción de todas las muestras clasificadas correctamente por el clasificador:

$$Exactitud = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.5}$$

A partir de los resultados obtenidos mostrados en los anexos, vamos a reducir todos nuestros clasificadores a cinco, el mejor de cada tipo de clasificador (Conjunto, KNN, SVM, Redes Neuronales y Árboles), y los analizaremos para obtener los mejores de entre ellos.

4.2.1. Clasificadores de Conjunto

En este apartado vamos a analizar los resultados obtenidos por medio de los distintos clasificadores de conjunto, y elegirá al más eficiente de entre ellos. Como criterio selector del mejor clasificador vamos a utilizar la exactitud como vemos en la Tabla 4.1 (o accuracy[1]). Como se puede ver en la Tabla 4.1 que muestra la media de la exactitud de los años 2018 y 2019, en los tres sensores para cada uno de los clasificadores, el Bagged Trees es el que muestra una mayor exactitud en sus predicciones, tanto para sobremuestreo como para submuestreo y es el clasificador elegido para este tipo.

Medias Exactitud (%)	Bagged Trees	Boosted Trees	RUS Boosted Trees	Subspace Discriminant	Subspace KNN
Sobremuestreo	**82.46666**	71.63333	70	59.56666	75.03333
Submuestreo	**73.23333**	55.83333	57.26666	59.4	45.51777

Tabla 4.1: Tabla Medias Exactitudes Clasificadores de Conjunto.

Una vez seleccionado el mejor clasificador, vamos a estudiar los resultados obtenidos para

[1]No confundir con Precisión o Precision.

las diferentes técnicas de balanceo de datos, recogidos en las Tablas 4.2a, 4.2b, 4.3a, 4.3b . Estas tablas están formadas por tres filas, cada una para los resultados de predicción de un evento de niebla (poca, media y mucha niebla), y una serie de columnas, cada una para una métricas de estudio, que a su vez se dividen en tres subcolumnas. Cada una de éstas muestra los resultados en cada uno de los tres puntos de análisis; la primera subcolumna muestra los datos del sensor uno, la segunda del dos y la tercera del tres. Cada clasificador cuenta con cuatro tablas, mostrando los resultados obtenidos mediante sobremuestro y submuestreo tanto para el año 2018 como para el año 2019.

Clase	**Verdaderos Positivos**			**Verdaderos Negativos**			**Falsos Positivos**			**Falsos Negativos**		
Poca Niebla	890	1048	1407	255	220	41	408	211	138	50	124	17
Media Niebla	7	3	7	1414	1476	1497	27	30	10	155	94	89
Mucha Niebla	213	175	22	1044	1133	1501	58	136	19	288	159	61

Clase	**Precisión**			**Exhaustividad**			**Valor-F**		
Poca Niebla	0.68567	0.83241	0.91068	0.94681	0.89420	0.98806	0.79535	0.86220	0.94779
Media Niebla	0.20588	0.09091	0.41176	0.04321	0.03093	0.07292	0.07143	0.04615	0.12389
Mucha Niebla	0.78598	0.56270	0.53659	0.42515	0.52395	0.26506	0.55181	0.54264	0.35484

(a) Métricas Bagged Trees mediante sobremuestreo en 2018.

Clase	**Verdaderos Positivos**			**Verdaderos Negativos**			**Falsos Positivos**			**Falsos Negativos**		
Poca Niebla	517	602	941	544	370	157	119	61	22	423	570	483
Media Niebla	76	25	49	1037	1106	1177	404	400	330	86	72	47
Mucha Niebla	309	244	53	924	998	1312	178	271	208	192	90	30

Clase	**Precisión**			**Exhautividad**			**Valor F**		
Poca Niebla	0.81289	0.90799	0.97715	0.55000	0.51365	0.66081	0.65609	0.65613	0.78844
Media Niebla	0.15833	0.05882	0.12929	0.46914	0.25773	0.51042	0.23676	0.09579	0.20632
Mucha Niebla	0.63450	0.47379	0.20307	0.61677	0.73054	0.63855	0.62551	0.57479	0.30814

(b) Métricas Bagged Trees mediante submuestreo en 2018.

Tabla 4.2: Métricas Bagged Trees en 2018.

Clase	**Verdaderos Positivos**			**Verdaderos Negativos**			**Falsos Positivos**			**Falsos Negativos**		
Poca Niebla	1065	1183	1450	265	210	62	225	71	75	48	139	16
Media Niebla	12	5	14	1439	1487	1518	21	36	15	131	75	56
Mucha Niebla	213	162	37	1189	1256	1524	67	146	12	134	39	30

Clase	**Precisión**			**Exhaustividad**			**Valor F**		
Poca Niebla	0.82558	0.94338	0.95082	0.95687	0.89486	0.98909	0.88639	0.91848	0.96958
Media Niebla	0.36364	0.12195	0.48276	0.08392	0.06250	0.20000	0.13636	0.08264	0.28283
Mucha Niebla	0.76071	0.52597	0.75510	0.61383	0.80597	0.55224	0.67943	0.63654	0.63793

(a) Métricas Bagged Trees mediante sobremuestreo en 2019.

Clase	**Verdaderos Positivos**			**Verdaderos Negativos**			**Falsos Positivos**			**Falsos Negativos**		
Poca Niebla	741	747	1171	431	267	125	59	14	12	372	575	295
Media Niebla	69	27	36	1105	1095	1320	355	428	213	74	53	34
Mucha Niebla	249	173	51	1126	1188	1416	130	214	120	98	28	16

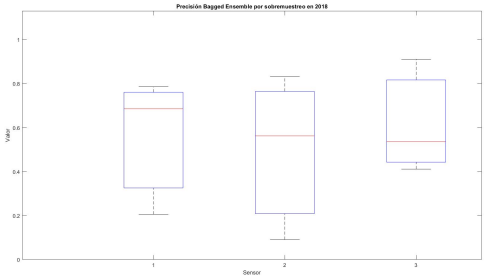
Clase	**Precisión**			**Exhaustividad**			**Valor F**		
Poca Niebla	0.92625	0.98160	0.98986	0.66577	0.56505	0.79877	0.77470	0.71723	0.88411
Media Niebla	0.16274	0.05934	0.14458	0.48252	0.33750	0.51429	0.24339	0.10093	0.22571
Mucha Niebla	0.65699	0.44703	0.29825	0.71758	0.86070	0.76119	0.68595	0.58844	0.42857

(b) Métricas Bagged Trees mediante submuestreo en 2019.

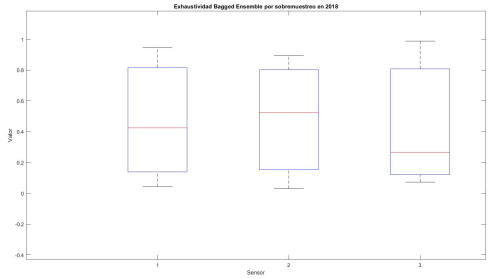
Tabla 4.3: Métricas Bagged Trees en 2019.

Observando por ejemplo la primera Tabla 4.2a, podemos destacar el elevado número de verdaderos positivos y negativos que posee, ocurriendo lo mismo en el resto de tablas. Además,

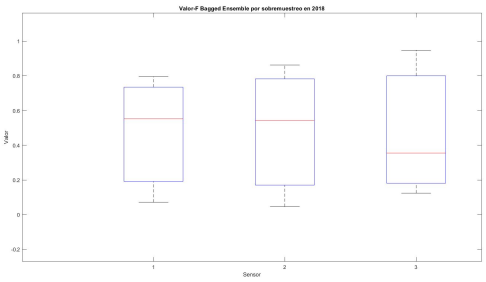
como podemos apreciar en la tercera Tabla 4.3a, el Bagged Trees destaca especialmente sobre el resto de modelos del anexo, en su menor número de falsos positivos y negativos. Además, también muestra unos mayores valores de precisión, exhaustividad y de valor-F, siendo por ello el que presenta una mayor capacidad de predecir de forma correcta los eventos de estudio.
En cuanto al resto de los clasificadores bajo estudio, una vez elegido el Bagged Tree como el que presenta mejores resultados, los clasificadores Boosted Trees y RUS Boosted Trees son los siguientes. Estas dos máquinas predictivas muestran unos valores buenos, aunque algo inferiores de verdaderos positivos y negativos, y de métricas de precisión, exhaustividad y valor-F. Como se ha mencionado, es en las métricas de falsos positivos y negativos donde el resto de clasificadores, como estos dos, muestran especialmente su baja capacidad de clasificación con respecto al mejor modelo.
Después de él, encontramos el clasificador Subspace KNN. Este clasificador muestra unos resultados inferiores en las métricas de verdaderos positivos y negativos. Aunque cabe destacar que es el que está mostrando mejores resultados en falsos positivos y negativos. Sin embargo, también muestra unos resultados inferiores en las métricas de precisión, exhaustividad y valor-F.
Por último, a partir de las métricas podemos resaltar que aunque posee unos resultados razonables comparados con los dos anteriores en el punto uno, en los puntos dos y tres, es el clasificador que más empeora sus resultados, especialmente con los sucesos media y mucha niebla, tanto para sobremuestreo como para submuestreo.
Por último, tenemos el predictor Subspace Discriminant. Este es claramente el peor predictor de este tipo de clasificadores. Muestra peores números de verdaderos positivos y negativos, mayores falsos positivos y negativos, y una peor precisión, exhaustividad y valor-F, tanto en sobremuestreo como en submuestreo. Destacan especialmente sus resultados pobres en los puntos dos y tres, sobre todo en cuanto a los eventos media y mucha niebla.
A partir de todo lo observado, se aprecia la superioridad de las métricas del clasificador Bagged Trees, como también podremos apreciar a continuación de forma visual en las Figuras 4.1 y 4.2:



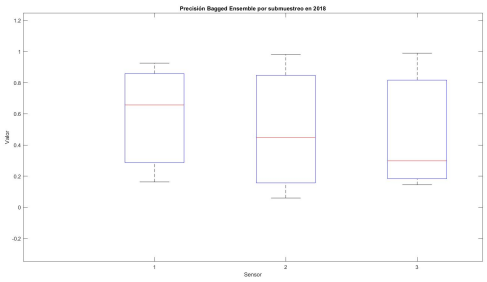
(a) Precisión para Sobremuestreo en 2018



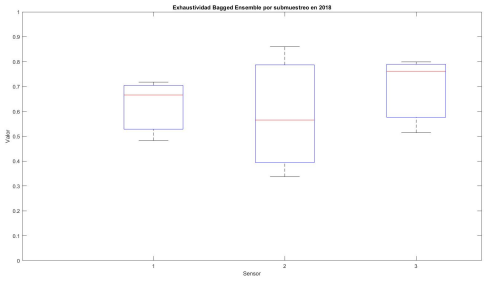
(b) Exhaustividad para Sobremuestreo en 2018



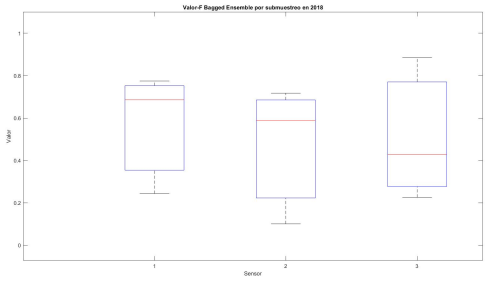
(c) Valor-F para Sobremuestreo en 2018



(d) Precisión para Submuestreo en 2018

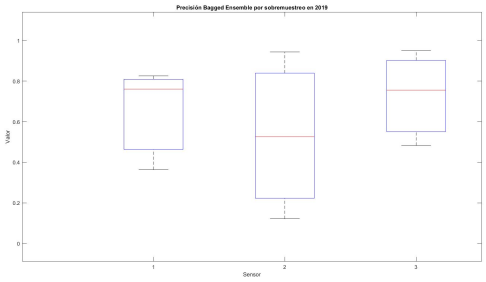


(e) Exhaustividad para Submuestreo en 2018

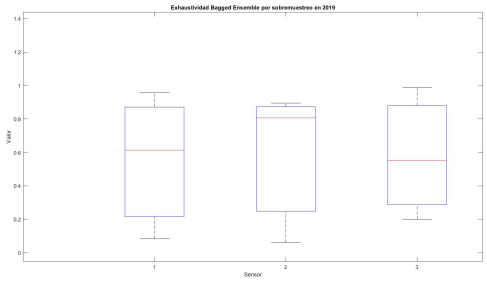


(f) Valor-F para Submuestreo en 2018

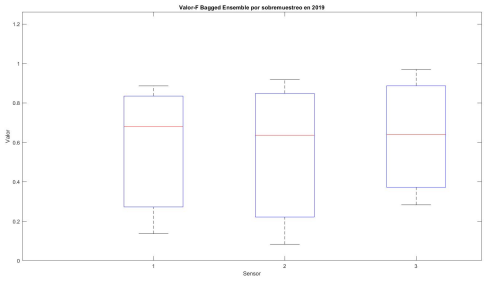
Figura 4.1: Métricas Bagged Trees en 2018



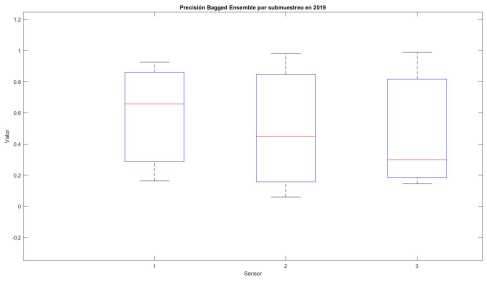
(a) Precisión para Sobremuestreo en 2019



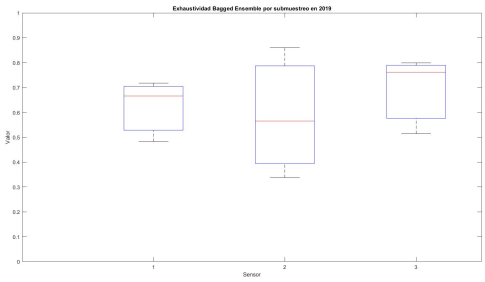
(b) Exhaustividad para Sobremuestreo en 2019



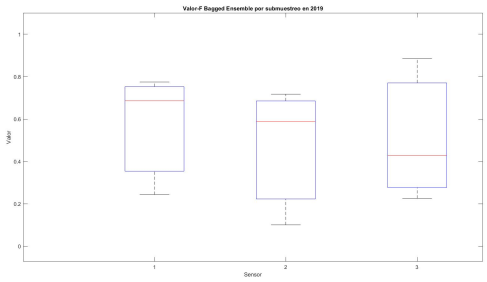
(c) Valor-F para Sobremuestreo en 2019



(d) Precisión para Submuestreo en 2019



(e) Exhaustividad para Submuestreo en 2019



(f) Valor-F para Submuestreo en 2019

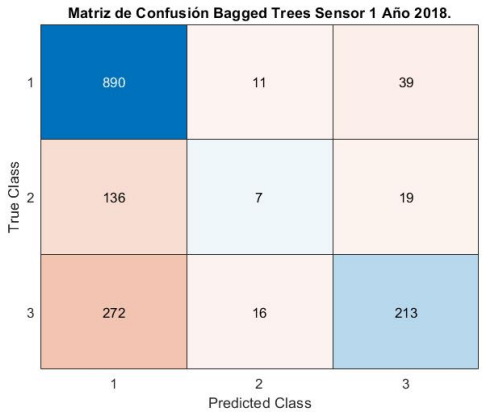
Figura 4.2: Métricas Bagged Trees en 2019

En las Figuras 4.1 y 4.2 podemos visualizar las métricas del clasificador Bagged Trees mediante unos diagramas de cajas, el eje X representa el punto o sensor donde se obtiene la métrica, y el eje Y su valor. En estos diagramas la línea roja representa la mediana, extremos inferior y superior de la caja indican los percentiles 25 y 75, respectivamente, y por último, los bigotes se extienden hasta los puntos de datos más extremos que no se consideran valores atípicos, los cuáles se indican con el marcador +.

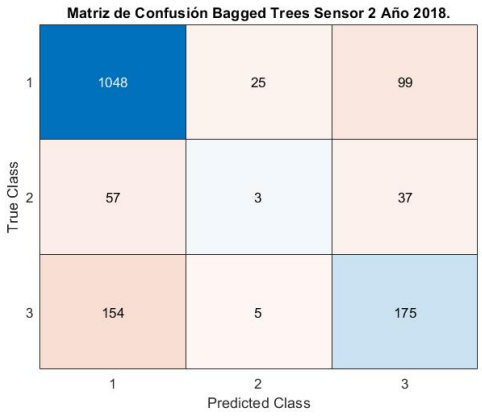
A partir de estas gráficas, y de las del resto de clasificadores en el anexo, puede parecer que el clasificador presenta una variación en el valor de las métricas muy elevados. Sin embargo, estas pueden resultar engañosas, ya que como se ha observado en las tablas el clasificador presenta peores resultados para los eventos de media niebla y mucha niebla, especialmente en el punto dos y sobre todo en el tres. Por lo tanto, en el intervalo de la caja, las muestras de los eventos poca niebla se encontrarán en la parte superior de ésta y la mayoría de los de los otros eventos en la parte inferior.

Al analizar por ejemplo la primera Figura 4.1 se puede confirmar la superioridad del clasificador Bagged Trees sobre el resto, presentando valores superiores en los tres parámentros, tanto para sobre como submuestreo. Esto ocurre también para 2019, en la Figura 4.2 .

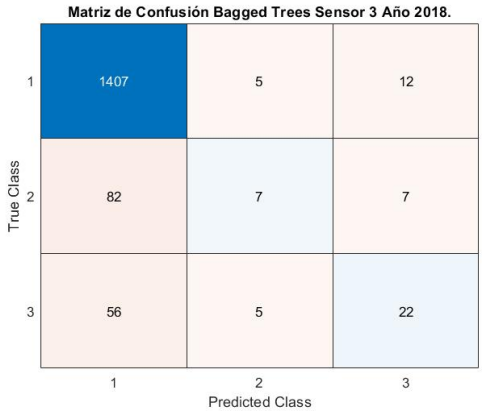
Una vez analizadas las métricas pasamos a las matrices de confusión de los distintos clasificadores:



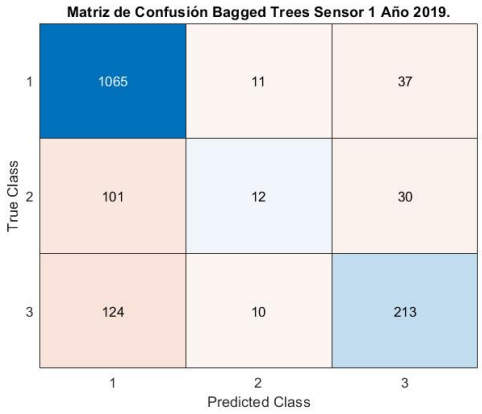
(a) Punto 1 en 2018



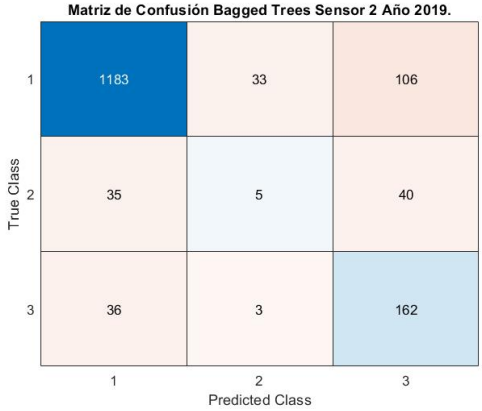
(b) Punto 2 en 2018



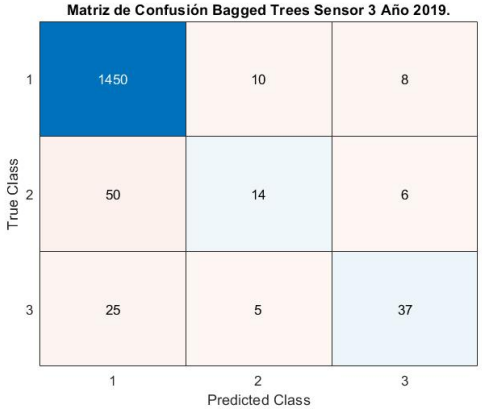
(c) Punto 3 en 2018



(d) Punto 1 en 2019

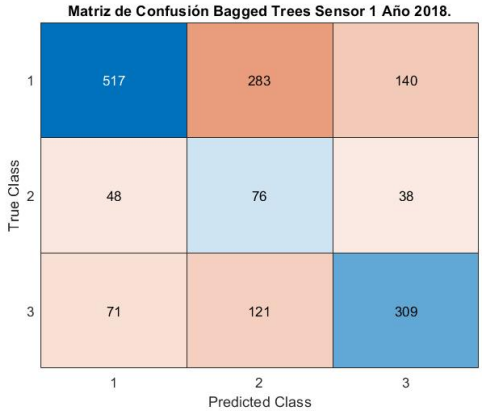


(e) Punto 2 en 2019

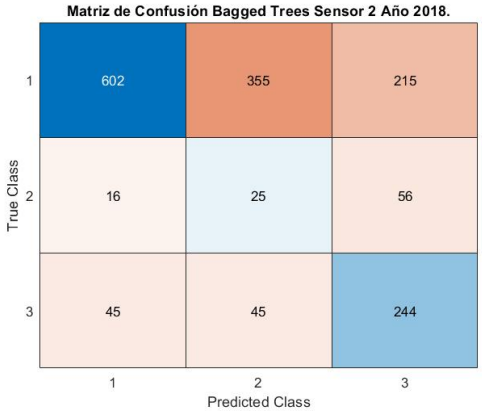


(f) Punto 3 en 2019

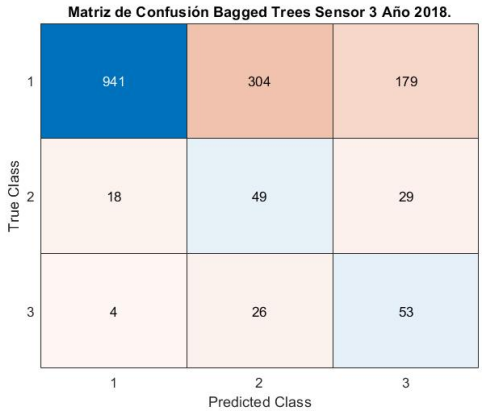
Figura 4.3: Matrices de Confusión del Bagged Trees de Conjunto mediante Sobremuestreo



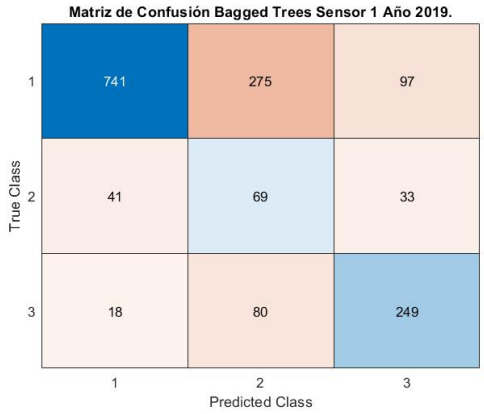
(a) Punto 1 en 2018



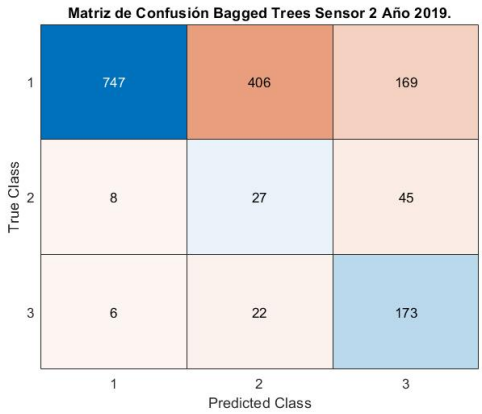
(b) Punto 2 en 2018



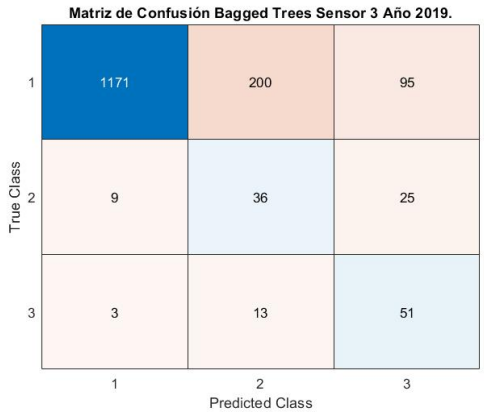
(c) Punto 3 en 2018



(d) Punto 1 en 2019



(e) Punto 2 en 2019



(f) Punto 3 en 2019

Figura 4.4: Matrices de Confusión del Bagged Trees de Conjunto mediante Submuestreo

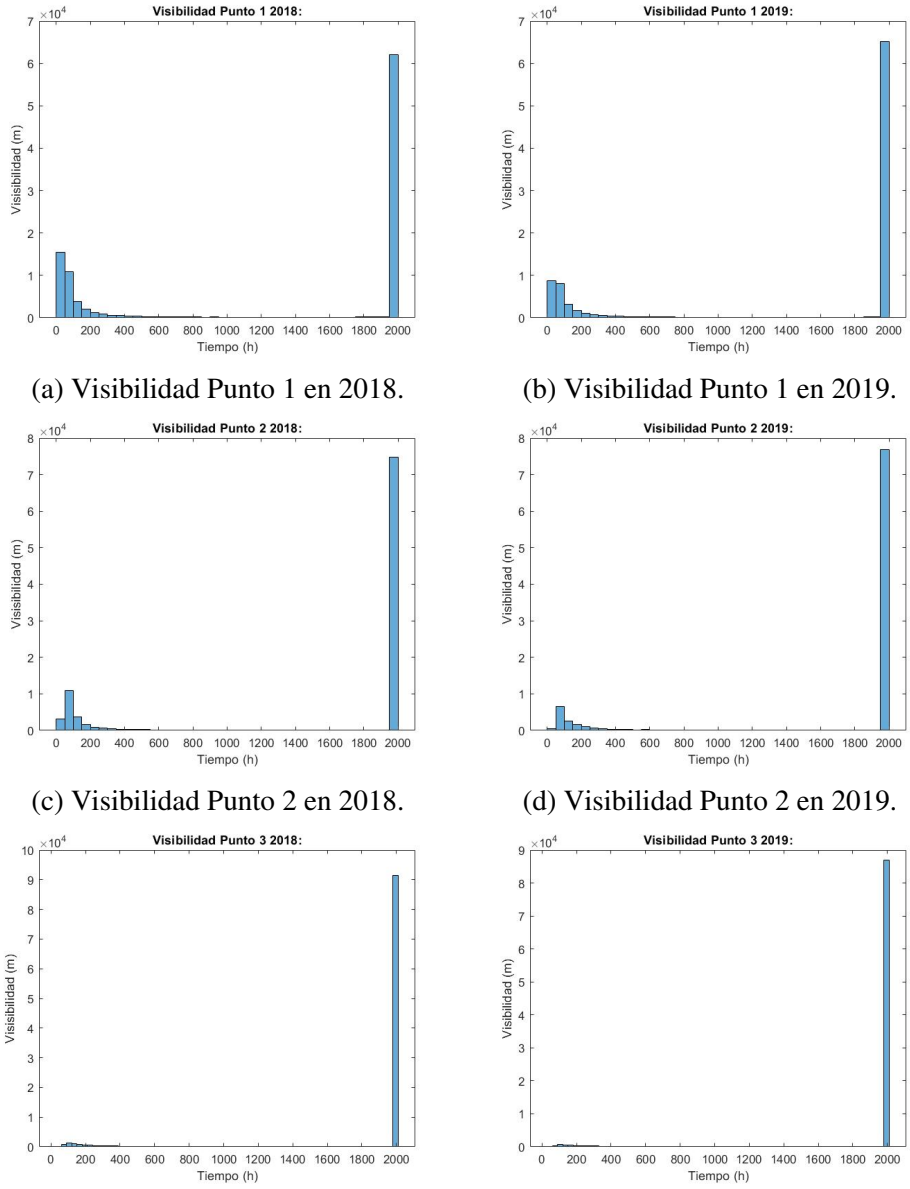
A partir de las matrices de confusión del Bagged Trees mostradas arriba, y de las del resto de predictores en el anexo, podemos apreciar una serie de aspectos que nos confirman los datos representados en las tablas de métricas. Primero de todo, como se puede observar en las Figuras 4.3a y 4.3c de sobremuestreo, al igual que en las tablas, los clasificadores mejoran sus resultados para el suceso poca niebla en los puntos dos y tres con respecto al punto uno, al igual que empeora los de media niebla y mucha niebla, tanto en verdaderos positivos como en falsos negativos. Este hecho se confirma para submuestreo en la Figura 4.4 analizando las mismas Figuras 4.4a y 4.4c. Asimismo, podemos apreciar notablemente un empeoramiento en los resultados en las matrices obtenidas por submuestreo en la Figura 4.4c, con respecto a las de sobremuestreo de la 4.3. Esto tiene sentido dado que al obtener las muestras de entrenamiento balanceando por submuestreo, reducimos el número de muestras de entrenamiento. Sin embargo, al reducir el número de muestras en submuestreo, el principal afectado es el evento poca niebla, el que mayor número de pérdidas sufre; por el contrario, de las otras dos clases, la que resulte tener menos número de muestras, apenas sufre una reducción en este número al tener muchas menos comparadas con Poca Niebla. Debido a esto, observamos como al sufrir estas pérdidas en el balaceo, el evento poca niebla en las matrices de confusión de submuestreo tiene unos peores resultados: menores verdaderos positivos y mayores falsos positivos. Por el contrario, las estimaciones de

los eventos media y mucha niebla son mejores. Esto puede ser debido, como ya se mencionó, a que la clase de estas dos que tenga más muestras, apenas sufre una reducción. Además, en el caso de sobremuestreo, al tener que crear tantas muestras sintéticas de media y mucha niebla a partir de las pocas, en comparación con poca niebla, puede haber afectado en la precisión de estos sucesos.

Por último, al observar las matrices de sobremuestreo y submuestreo, por ejemplo las Figuras 4.3b y 4.4b, en las que este hecho es más visible, los principales errores al predecir son con el evento de mayor número de muestras: el evento poca niebla con el mucha niebla (el siguiente con mayor número de estas) y los media niebla y mucha niebla con el suceso poca niebla, apoyando esta teoría. Además, hay que destacar en estas imágenes el aumento considerable en el número de muestras poca niebla predichas de forma errónea como media y mucha niebla en submuestreo. Esto es debido al muy inferior número de muestras de entrenamiento de sucesos poca niebla con respecto a sobremuestreo (el evento poca niebla es el que más muestras pierde en submuestreo al ser la clase mayoritaria). Debido a esto se produce un aumento considerable en el número de predicciones erróneas como media y mucha niebla en vez de como poca niebla.

En comparación al resto de clasificadores, en las matrices de confusión se visualiza igualmente que los mejores resultados los produce el clasificador Bagged Trees. Se observa que este tiene el mayor número de verdaderos positivos y el menor número de falsos positivos y negativos con respecto al resto. Además, se trata del clasificador con mejores resultados en todas las situaciones, mediante sobremuestreo y submuestreo, para los tres puntos y para los tres tipos de eventos. En cuanto al resto de clasificadores, el Boosted Trees, RUS Boosted Trees y Subspace KNN presentan peores resultados, aunque aún relativamente buenos. Sin embargo, subspace discriminant presenta unos resultados más pobres de forma apreciable. El Bagged Trees es el único que muestra buenos resultados en todas las matrices, como se puede ver en las Figuras 4.3 y 4.4. En cuanto al resto, estos empeoran con respecto a éste en submuestreo (a excepción de RUS Boosted Trees en el punto 1 mediante submuestreo) y especialmente en los puntos dos y tres por ambos métodos, mostrando por lo tanto una menor capacidad para predecir eventos cuando hay un menor número de muestras de prueba.

Este empeoramiento en los resultados en los sensores dos y tres tanto en las gráficas como en las tablas para todos los clasificadores tiene su motivo. Esto es debido a, como se aprecia en las siguientes imágenes, al menor número de muestras de media y mucha niebla en los sensores dos en las Figuras 4.5c, 4.5d y tres en las 4.5e, 4.5f, con respecto al sensor uno en las 4.5a, 4.5b. Debido a ello, en los puntos dos y tres empeoran los resultados de las predicciones de estos eventos y mejoran los de poca niebla (cuyas muestras aumentan en estos sensores). Como se ha visto en las matrices 4.3 y 4.4 esta situación aparece en ambas técnicas de balanceo durante 2018 y 2019.



(a) Visibilidad Punto 1 en 2018.

(b) Visibilidad Punto 1 en 2019.

(c) Visibilidad Punto 2 en 2018.

(d) Visibilidad Punto 2 en 2019.

(e) Visibilidad Punto 3 en 2018.

(f) Visibilidad Punto 3 en 2019.

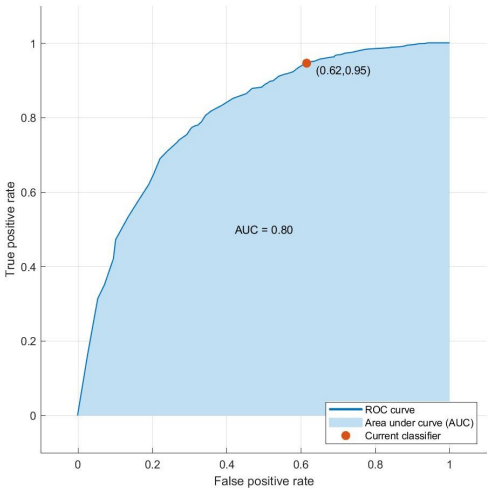
Figura 4.5: Número de ocurrencias de los diferentes eventos de visibilidad.

Finalmente, analizamos las curvas ROC y el AUC de nuestros clasificadores de Conjunto. En estas Figuras 4.6 podemos observar la ROC, es decir, curva de probabilidad y el AUC, que nos indica la medida de separabilidad del modelo, cómo de capaz es de distinguir entre las distintas clases, y por lo tanto a mayor AUC mejor puede el modelo identificar cada clase correctamente, [32].

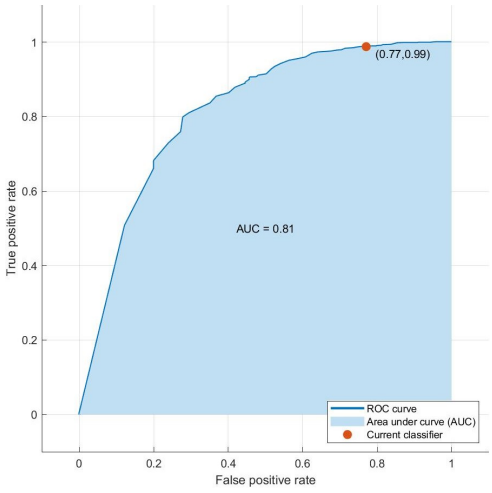
A partir de estas gráficas podemos observar los resultados vistos hasta ahora. El Bagged Trees es el clasificador que mejores resultados presenta, en los tres tipos de gráficas (considerando positiva cada una de las clases) en los tres sensores por los dos métodos de balanceo.

Para empezar, en estas gráficas destaca que los eventos poca niebla y mucha niebla tengan los AUC más elevados, como se observa por ejemplo en las de Bagged Trees en 4.6a y 4.6c y 4.6e.

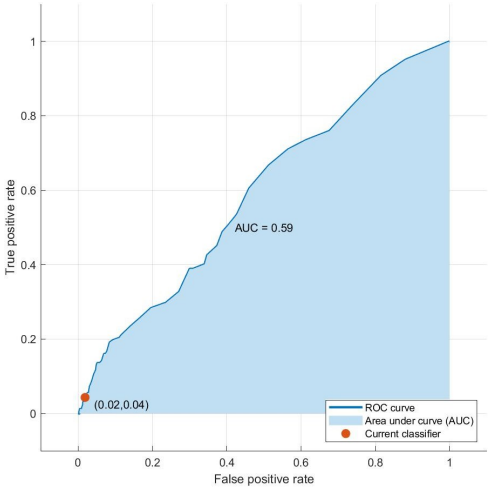
Esto puede ser debido a que son los que poseen una características meteorológicas más claras y fáciles de identificar, sumado a que son los que poseen un mayor número de muestras de prueba. Además de esto, en estas gráficas se aprecia un AUC superior para el evento mucha niebla como clase positiva, como se ve en 4.6e y 4.6f. Esto puede ser consecuencia del muy elevado número muestras de poca niebla con respecto al de mucha, pudiendo provocar que tenga muchas más muestras a predecir y con las que equivocarse. Esto concuerda con las gráficas de submuestreo del anexo donde se obtienen unas AUC más elevadas que en sobremuestreo en los tres puntos. Esto como se ha dicho puede deberse al que al tener un inferior número de muestras de cada evento hay menor muestras que predecir y con las que se pueden equivocar. A todo esto hay que añadir como se muestra en las Figuras 4.6b, 4.6d y 4.6f , el aumento producido en las AUC de los puntos dos y tres, especialmente notable para submuestreo, en todos los clasificadores. Esto parece respaldar la hipótesis ya mencionada de que, al haber menos muestras de prueba de una clase se produce un mayor AUC debido a las menos oportunidades para equivocarse.



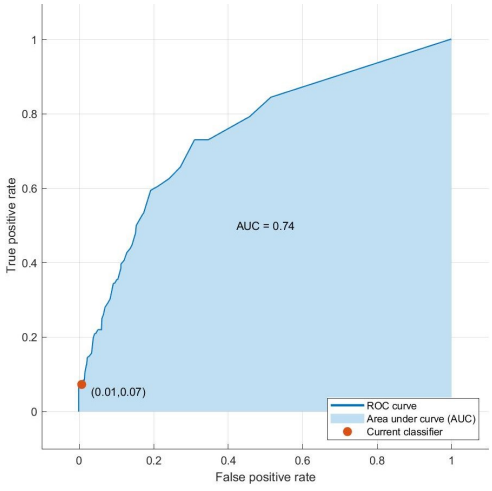
(a) Clase Positiva Evento Poca Niebla en el Punto 1.



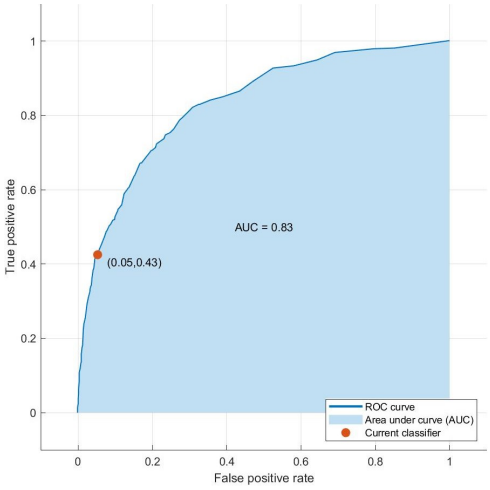
(b) Clase Positiva Evento Poca Niebla en el Punto 3.



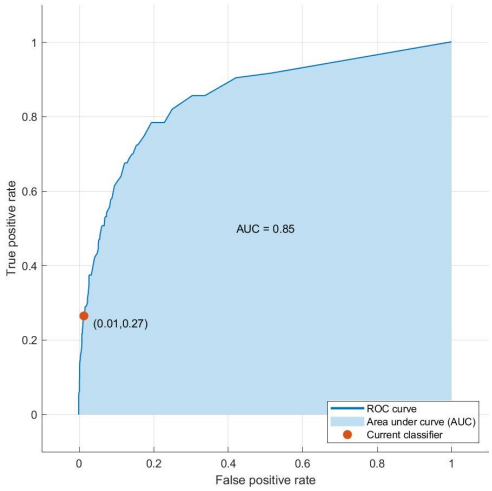
(c) Clase Positiva Evento Media Niebla en el Punto 1.



(d) Clase Positiva Evento Media Niebla en el Punto 3.



(e) Clase Positiva Evento Mucha Niebla en el Punto 1.



(f) Clase Positiva Evento Mucha Niebla en el Punto 3.

Figura 4.6: ROC y AUC de Bagged Tress en el punto 1 y 3 por sobremuestreo en 2018.

A partir de todos los resultados analizados, queda demostrado en ellos la superioridad del clasificador Bagged Trees, y sobre todo la resistencia de predecir de forma correcta los eventos media y poca niebla en los puntos dos y tres, a pesar de las muchas menos muestras de estas dos clases de eventos, tanto para sobremuestreo como para submuestreo.

4.2.2. Clasificadores KNN

A continuación, vamos a analizar los tipos de clasificadores KNN. En cuanto a estos clasificadores, observando las medias de exactitud de la Tabla 4.4, detectamos que el Fine KNN es el más eficiente. A pesar de que no tiene la exactitud más elevada para submuestreo, debido a su superioridad en sobremuestreo, por medio de la media de ambas es la mejor elección.

Medias Exactitud	Coarse KNN	Cosine KNN	Cubic KNN	Fine KNN	Medium KNN	Weighted KNN
Sobremuestreo	60.95	65.01666	63.96666	**77.73333**	75.03	69.55
Submuestreo	57.18333	62.5	60.41666	**61.65**	61.48	63.6

Tabla 4.4: Tabla Medias Exactitudes Clasificadores de KNN

Una vez elegido el clasificador con el mayor número de predicciones acertadas, comenzamos, como en el apartado anterior analizando las métricas de los diferentes clasificadores:

Clase	Verdaderos Positivos			Verdaderos Negativos			Falsos Positivos			Falsos Negativos		
Poca Niebla	749	843	1306	459	321	104	199	88	75	196	351	118
Media Niebla	26	14	23	1302	1339	1424	155	177	83	120	73	73
Mucha Niebla	313	229	36	930	1029	1440	161	252	80	199	93	47

Clase	Precisión			Exhaustividad			Valor F		
Poca Niebla	0.79008	0.90548	0.94569	0.79259	0.70603	0.91713	0.79134	0.79341	0.93119
Media Niebla	0.14365	0.07330	0.21698	0.17808	0.16092	0.23958	0.15902	0.10072	0.22772
Mucha Niebla	0.66034	0.47609	0.31034	0.61133	0.71118	0.43373	0.63489	0.57036	0.36181

(a) Métricas Fine KNN mediante sobremuestreo en 2018.

Clase	Verdaderos Positivos			Verdaderos Negativos			Falsos Positivos			Falsos Negativos		
Poca Niebla	470	520	832	544	362	156	119	69	23	470	652	592
Media Niebla	67	30	42	1035	1063	1125	406	443	382	95	67	54
Mucha Niebla	289	213	51	850	941	1247	252	328	273	212	121	32

Clase	Precisión			Exhaustividad			Valor F		
Poca Niebla	0.79796	0.88285	0.97310	0.50000	0.44369	0.58427	0.61478	0.59057	0.73014
Media Niebla	0.14165	0.06342	0.09906	0.41358	0.30928	0.43750	0.21102	0.10526	0.16154
Mucha Niebla	0.53420	0.39372	0.15741	0.57685	0.63772	0.61446	0.55470	0.48686	0.25061

(b) Métricas Fine KNN mediante submuestreo en 2018.

Tabla 4.5: Métricas Fine KNN en 2018.

Clase	Verdaderos Positivos			Verdaderos Negativos			Falsos Positivos			Falsos Negativos		
Poca Niebla	960	1047	1414	358	227	69	132	54	68	153	275	52
Media Niebla	25	10	20	1347	1382	1492	113	141	41	118	70	50
Mucha Niebla	246	156	33	1129	1207	1509	127	195	27	101	45	34

Clase	Precisión			Exhaustividad			Valor F		
Poca Niebla	0.87912	0.95095	0.95412	0.86253	0.79198	0.96453	0.87075	0.86422	0.95929
Media Niebla	0.18116	0.06623	0.32787	0.17483	0.12500	0.28571	0.17794	0.08658	0.30534
Mucha Niebla	0.65952	0.44444	0.55000	0.70893	0.77612	0.49254	0.68333	0.56522	0.51969

(a) Métricas Fine KNN mediante sobremuestreo en 2019.

Clase	Verdaderos Positivos			Verdaderos Negativos			Falsos Positivos			Falsos Negativos		
Poca Niebla	647	698	1092	426	253	118	64	28	19	466	624	374
Media Niebla	59	26	38	1048	1063	1244	412	460	289	84	54	32
Mucha Niebla	229	136	47	1064	1147	1418	192	255	118	118	65	20

Clase	Precisión			Exhaustividad			Valor F		
Poca Niebla	0.90999	0.96143	0.98290	0.58131	0.52799	0.74488	0.70943	0.68164	0.84750
Media Niebla	0.12527	0.05350	0.11621	0.41259	0.32500	0.54286	0.19218	0.09187	0.19144
Mucha Niebla	0.54394	0.34783	0.28485	0.65994	0.67662	0.70149	0.59635	0.45946	0.40517

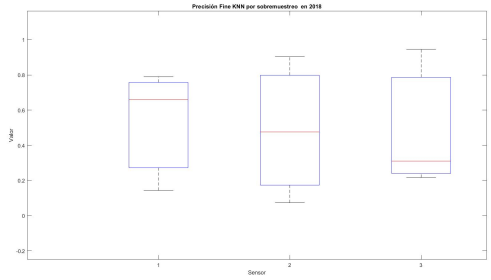
(b) Métricas Fine KNN mediante submuestreo en 2019.

Tabla 4.6: Métricas Fine KNN en 2019.

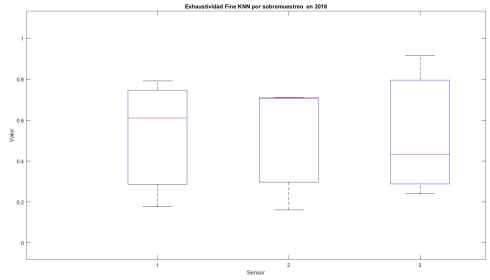
A partir de las métricas mostradas en las tablas, podemos apreciar que el Fine KNN es el predictor más eficiente de este tipo, aunque en este caso los resultados no sean tan buenos como los del Bagged Trees. Por ejemplo, si analizamos la Tabla 4.5a y la comparamos con las del resto para sobremuestreo en 2018, se aprecia como al igual que sucedía con el Bagged Trees, el Fine KNN posee los mejores resultados de verdaderos positivos y negativos. Sin embargo, en esta tabla se aprecia que los otros clasificadores le superan en algunas de las métricas, especialmente para el evento media niebla. Fine Tree destaca principalmente para poca niebla y mucha niebla. En submuestreo, en las Tablas 4.5b, 4.6b, apreciamos una menor precisión del Fine KNN en cuando a los verdaderos positivos y negativos. Sin embargo, donde destaca especialmente este clasificador es en los falsos positivos y negativos. El resto de clasificadores poseen una mayor cantidad de falsos positivos y negativos, para los tres eventos, incrementándose esta diferencia para sobremuestreo. El Fine KNN en las métricas en las que destaca son especialmente para

los eventos de media y mucha niebla, superando en general en estos eventos al resto, tanto para sobremuestreo como submuestreo.

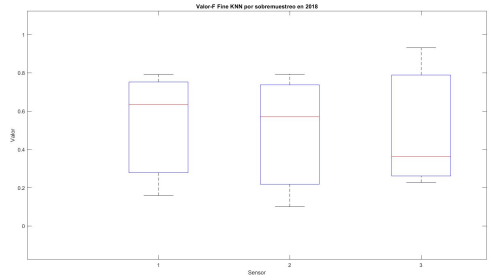
En cuanto al resto de los clasificadores, el Cosine KNN, Coarse KNN, Cubic KNN, Weighted KNN y el Medium KNN, siguen todos la misma tendencia. Todos ellos muestran unos números de verdaderos positivos y negativos más o menos similar, pero sin embargo, como ya se mencionó, la proporción de estos con los falsos positivos y negativos es mucho peor que en el caso del Fine KNN, especialmente para los eventos de media niebla y mucha niebla. Sin embargo, de entre ellos podemos destacar al Weighted KNN. Este clasificador, a diferencia del resto, mantiene una mejor proporción de falsos positivos y negativos para los verdaderos positivos, siendo el siguiente mejor por detrás de Fine KNN. Por último, en cuanto a las métricas de clasificación, exhaustividad y de valor-F, el Weighted KNN, se postula claramente como el siguiente mejor algoritmo, siendo el que mejores resultados muestra frente a los otros cuatro en las tres métricas. Para estos clasificadores, al no destacar numéricamente, es especialmente de ayuda las representaciones gráficas de los resultados, como las siguientes que se muestran a continuación y que se corresponden con Fine KNN, en las Figuras 4.7 y 4.8:



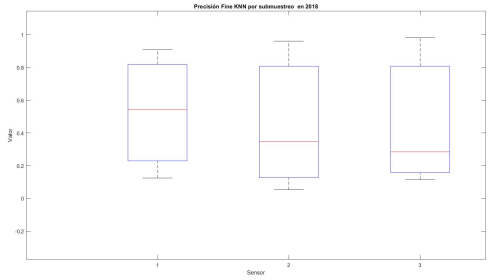
(a) Precisión para Sobremuestreo en 2018



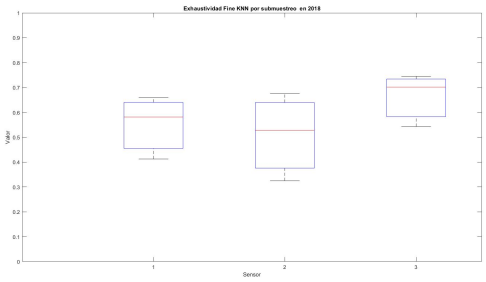
(b) Exhaustividad para Sobremuestreo en 2018



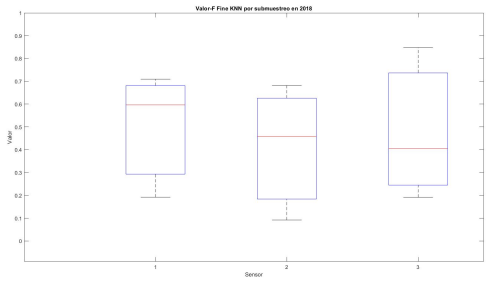
(c) Valor-F para Sobremuestreo en 2018



(d) Precisión para Submuestreo en 2018

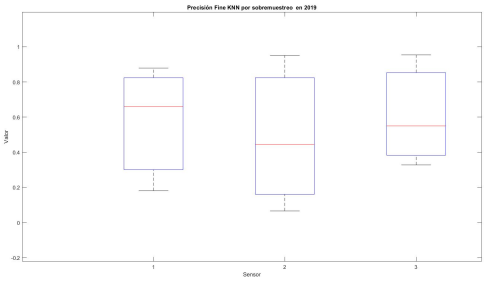


(e) Exhaustividad para Submuestreo en 2018

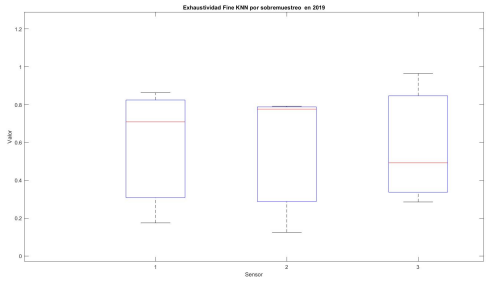


(f) Valor-F para Submuestreo en 2018

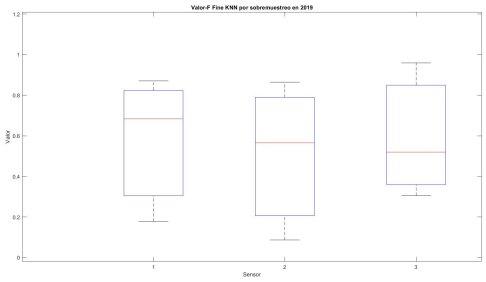
Figura 4.7: Métricas Fine KNN en 2018



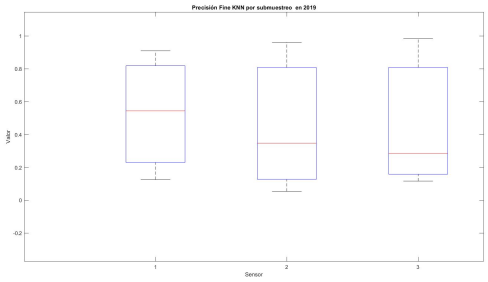
(a) Precisión para Sobremuestreo en 2019



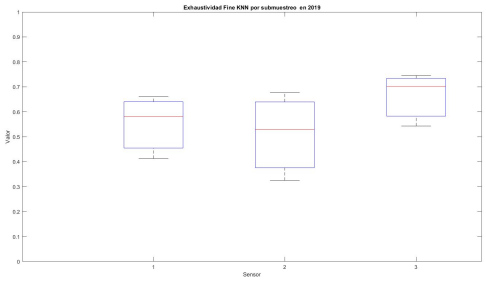
(b) Exhaustividad para Sobremuestreo en 2019



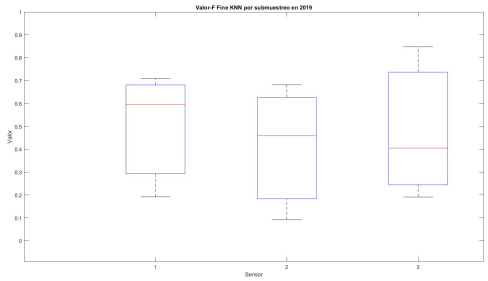
(c) Valor-F para Sobremuestreo en 2019



(d) Precisión para Submuestreo en 2019



(e) Exhaustividad para Submuestreo en 2019

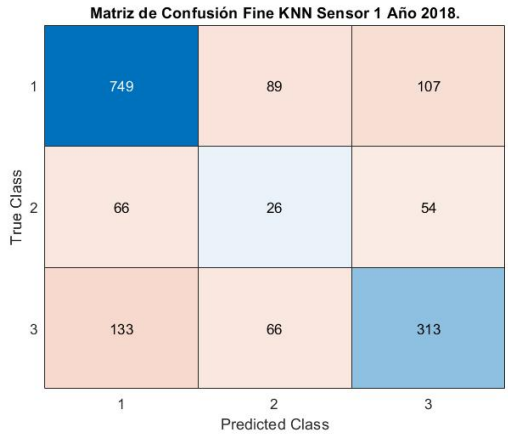


(f) Valor-F para Submuestreo en 2019

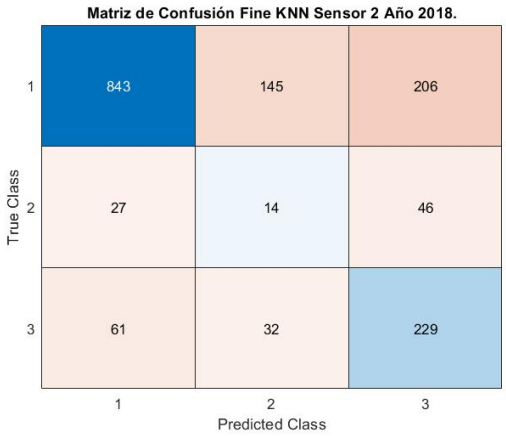
Figura 4.8: Métricas Fine KNN en 2019

A partir de estas métricas se puede apreciar que el Fine KNN destaca sobre el resto por tener una precisión, exhaustividad y valor-F ligeramente superiores a las del resto de clasificadores;

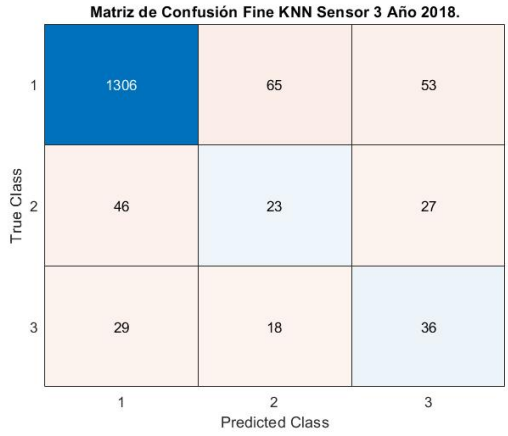
con unas medianas que suelen ser algo superiores, y unos bigotes alcanzando mayores valores, aunque no hay una diferencia tan clara como en el Bagged Trees anterior. Esto tiene sentido, dado que estas gráficas muestran las medianas de los tres puntos, y por lo tanto, como ya dijimos, este clasificador era el que destacaba por ser el que posee mejores resultados de media.
Además, estos diagramas de cajas abarcan un amplio rango de valores, debido a las muestras de media y mucha niebla, que cuentan con unos resultados apreciablemente peores como se ha ha ido observando en todos los clasificadores.
A continuación, vamos a proceder a analizar las matrices de confusión obtenidas a partir de los distintos clasificadores:



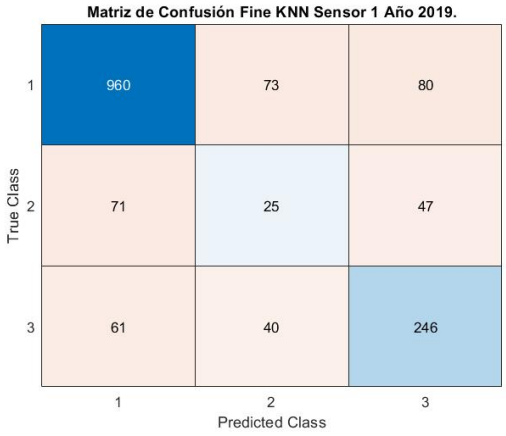
(a) Punto 1 en 2018



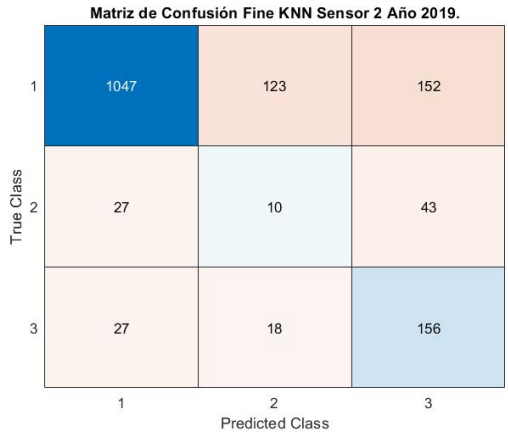
(b) Punto 2 en 2018



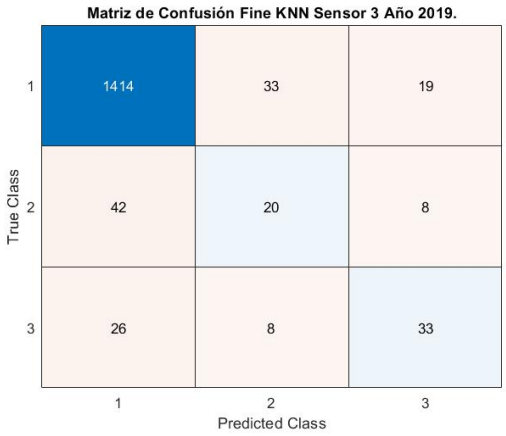
(c) Punto 3 en 2018



(d) Punto 1 en 2019

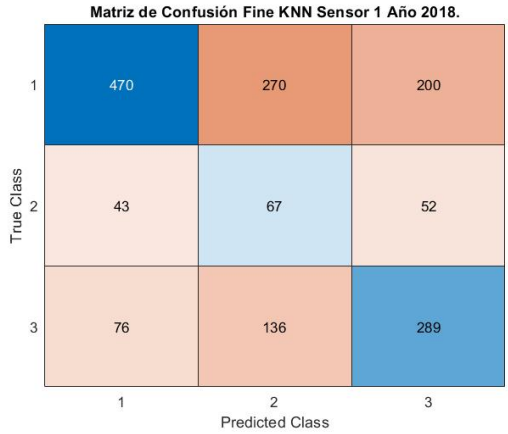


(e) Punto 2 en 2019

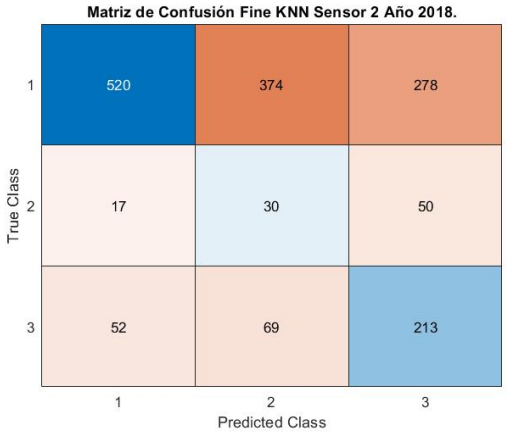


(f) Punto 3 en 2019

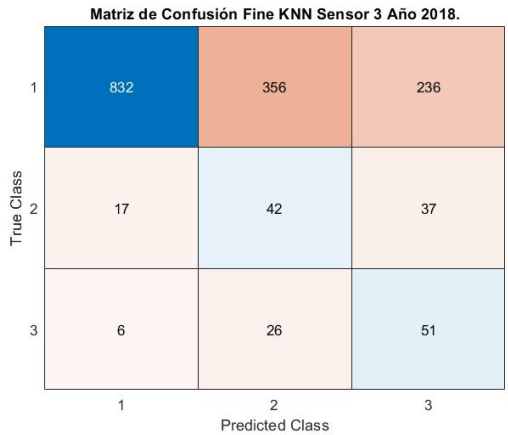
Figura 4.9: Matrices de Confusión del Fine KNN mediante Sobremuestreo



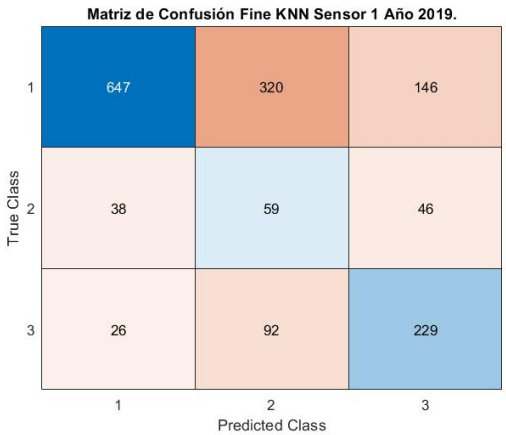
(a) Punto 1 en 2018



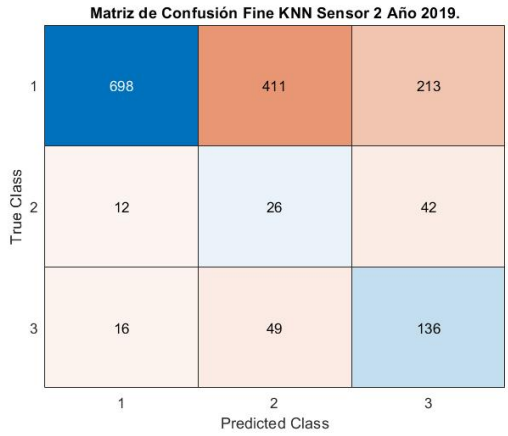
(b) Punto 2 en 2018



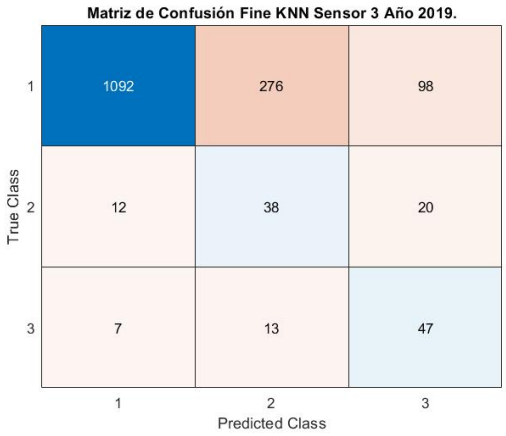
(c) Punto 3 en 2018



(d) Punto 1 en 2019



(e) Punto 2 en 2019



(f) Punto 3 en 2019

Figura 4.10: Matrices de Confusión del Fine KNN mediante Submuestreo

A partir de las matrices de confusión del Fine KNN podemos destacar algunas de las observaciones hechas a partir de las métricas de las tablas. Primero de todo, mediante sobremuestreo, como se aprecia en la Figura 4.9, se aprecia una elevada eficiencia, ya observada en los datos numéricos, para los eventos de poca y de mucha niebla. En cuanto a media niebla, no destaca al igual que el resto de clasificadores observados hasta ahora. Sin embargo, a diferencia del Bagged Tree, el Fine KNN muestra una menor precisión en los puntos dos y tres, con peores resultados para media y mucha niebla. Por lo tanto, el Fine KNN parece mostrar una mayor sensibilidad a la escasez de datos de este tipo de sucesos en estos sensores.

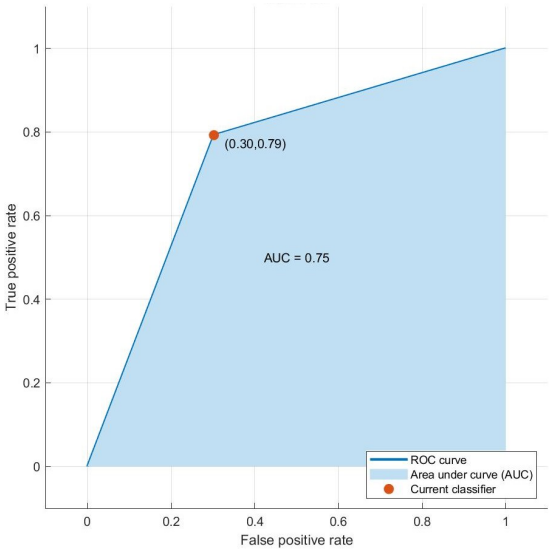
Al igual que en los clasificadores de conjunto, los principales errores de predicción suelen darse entre las clases mayoritarias, tanto para sobremuestreo como para submuestreo, como se puede observar en las Figuras 4.9b y 4.10e respectivamente. Sin embargo, en el punto tres, el evento poca niebla se confunde más con media que con mucha niebla. Esto podría deberse a que las matrices KNN pueden ser menos sensibles en sobremuestreo a este efecto y por ello se confunde más con la clase media niebla que tiene características meteorológicas más parecidas a las suyas. Además, al igual que sucedía en los clasificadores de conjunto, en los KNN también empeoran los resultados de los eventos media y mucha niebla, y mejoran los de poca, en los puntos dos y tres, debido a las menores y mayores muestras de estos respectivamente en estos puntos. En esta línea, como se puede ver al comparar la Figura 4.9b con 4.9b, se aprecia un empeoramiento de las predicciones de los eventos poca niebla en submuestreo, a la vez que mejoran los de media y mucha con respecto a sobremuestreo. Esto parece respaldar la hipótesis de que empeoran las predicciones niebla de los eventos poca niebla debido al menor número de muestras de entrenamiento, y mejoran las clasificaciones de media y mucha niebla, debido a no tener muestras sintéticas en el entrenamiento.

Por último, en cuanto a las matrices de confusión obtenidas por los distintos clasificadores KNN queda palpable la superioridad del clasificador Fine KNN sobre el resto. Se trata del predictor que cuenta con unos verdaderos positivos y negativos más elevados, y especialmente destaca

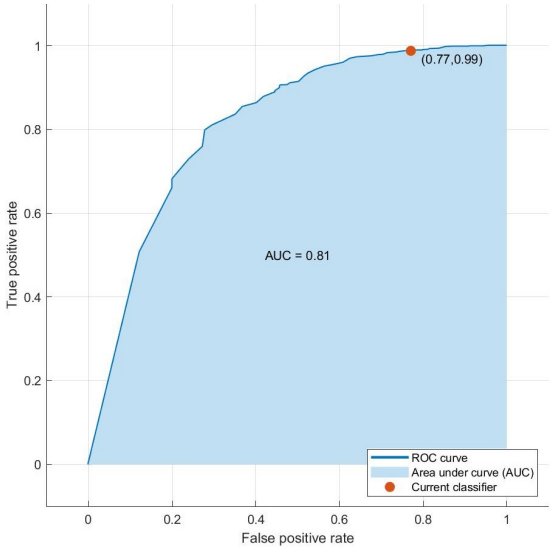
por tener unos falsos positivos y negativos más reducidos que el resto. Cabe destacar que es el que mejores resultados muestra para los eventos de media y mucha niebla, convirtiéndose en un buen clasificador pese al reducido número de muestras de estos eventos. En cuanto al resto de clasificadores, después del Fine KNN, tanto para sobremuestreo como en submuestreo destaca el Weighted KNN como el siguiente clasificador con mejores resultados. Con mayores verdaderos positivos y negativos y menores falsos positivos y negativos en los tres puntos, destaca especialmente en Media y Mucha niebla sobre el resto de los clasificadores. Después de estos, presentan resultados parejos el Medium, Cubic y Cosine KNN y por último el Coarse KNN que es el que muestra unos resultados más pobres.

Después de analizar las matrices de confusión, al igual que en el apartado anterior, vamos a analizar las gráficas ROC y las AUC de estos clasificadores. En cuanto a estas gráficas, hay que destacar que al igual que sucedía con las de los clasificadores de conjunto, para éstas también aumentan los AUC al pasar del punto uno al dos y tres, como en las Figuras 4.11a y 4.11b, donde hay menos muestras, apoyando la teoría de que al tener menos muestras de prueba se da menos oportunidad para cometer errores. Sin embargo, a diferencia de lo que sucedía antes, en esta ocasión, al observar la Figura 4.11a frente a la 4.11e, vemos que en estos clasificadores no poca niebla en la mayoría de casos. Esto puede ser debido a que en estos clasificadores las predicciones hechas para los eventos mucha niebla muestran más fallos de los que lograban las predicciones hechas para mucha niebla en los clasificadores de conjunto.

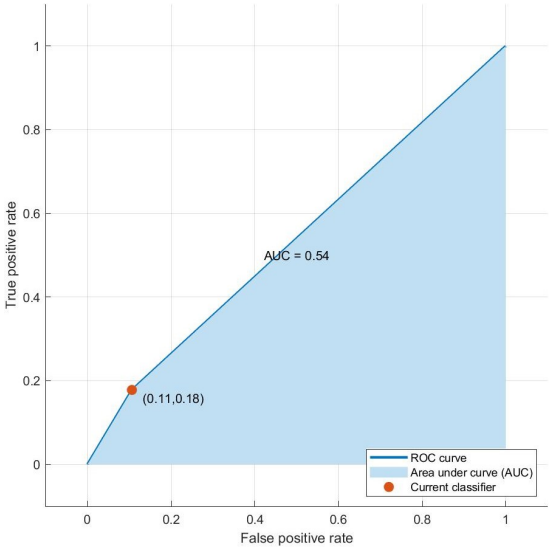
Finalmente, cabe destacar que el Fine KNN cuenta con menores AUC que el resto de clasificadores. Esto puede ser debido a que, como se vio en las tablas de las métricas, en algunas de las métricas de verdaderos positivos y negativos eran superiores otros clasificadores, por lo que podrían tener unas AUC mayores; sin embargo como se vio en las matrices de confusión, cuentan con falsos positivos y negativos bastante superiores de los que posee el Fine KNN, convirtiéndole en el mejor clasificador.



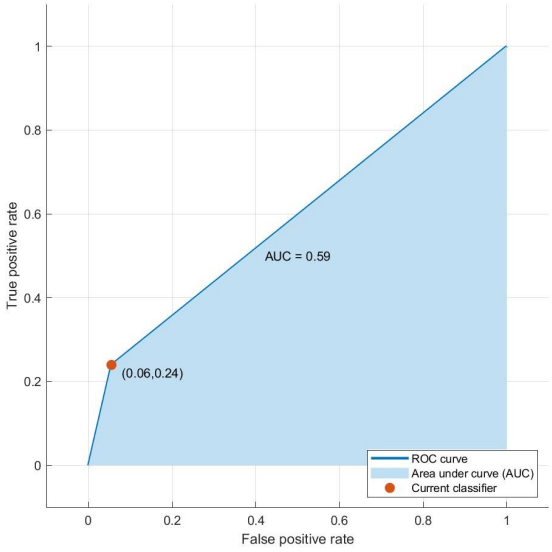
(a) Clase Positiva Evento Poca Niebla en el Punto 1.



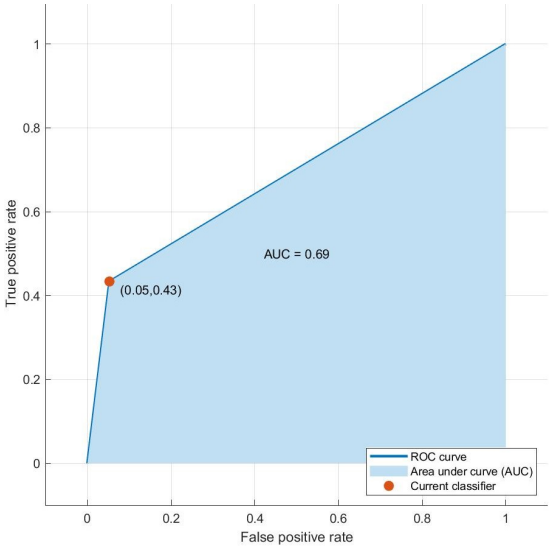
(b) Clase Positiva Evento Poca Niebla en el Punto 3.



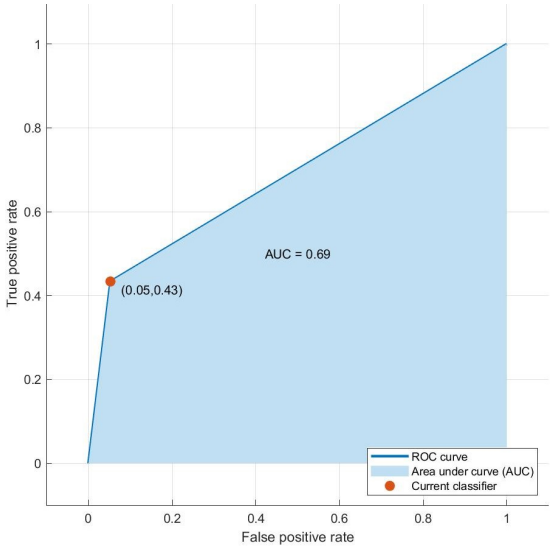
(c) Clase Positiva Evento Media Niebla en el Punto 1.



(d) Clase Positiva Evento Media Niebla en el Punto 3.



(e) Clase Positiva Evento Mucha Niebla en el Punto 1.



(f) Clase Positiva Evento Mucha Niebla en el Punto 3.

Figura 4.11: ROC y AUC de Fine KNN en el punto 1 y 3 por sobremuestreo durante 2018.

4.2.3. Redes Neuronales

A continuación, vamos a analizar los clasificadores creados mediante redes neuronales. La Tabla 4.7 muestra las medias de la exactitud de los clasificadores basados en redes neuronales. Elegimos el Wide NN como el mejor de este tipo, teniendo una mayor exactitud, tanto en sobremuestreo como en submuestreo.

Medias Exactitud	**Bilayared NN**	**Medium NN**	**Narrow NN**	**Trilayared NN**	**Wide NN**
Sobremuestreo	61.1	67.45	63.76666	61.7	**71.23888**
Submuestreo	59.8	62.11666	61.13333	60.315	**63.23333**

Tabla 4.7: Tabla Medias Exactitudes Clasificadores de Redes Neuronales

Una vez seleccionado el clasificador pasamos a analizar los resultados obtenidos por las distintas redes neuronales creadas en el proyecto, comenzando como siempre con las tablas de las métricas.

Clase	**Verdaderos Positivos**			**Verdaderos Negativos**			**Falsos Positivos**			**Falsos Negativos**		
Poca Niebla	648	745	1164	465	345	115	198	86	64	292	427	260
Media Niebla	43	24	34	1205	1230	1334	236	276	173	119	73	62
Mucha Niebla	284	226	44	908	1023	1396	194	246	124	217	108	39

Clase	**Precisión**			**Exhaustividad**			**Valor-F**		
Poca Niebla	0.76596	0.89651	0.94788	0.68936	0.63567	0.81742	0.72564	0.74388	0.87783
Media Niebla	0.15412	0.08000	0.16425	0.26543	0.24742	0.35417	0.19501	0.12091	0.22442
Mucha Niebla	0.59414	0.47881	0.26190	0.56687	0.67665	0.53012	0.58018	0.56079	0.35060

(a) Métricas Wide Neural Network mediante sobremuestreo en 2018.

Clase	**Verdaderos Positivos**			**Verdaderos Negativos**			**Falsos Positivos**			**Falsos Negativos**		
Poca Niebla	452	524	835	521	359	145	142	72	34	488	648	589
Media Niebla	70	34	37	986	1033	1130	455	473	377	92	63	59
Mucha Niebla	245	197	49	863	966	1249	239	303	271	256	137	34

Clase	**Precisión**			**Exhaustividad**			**Valor F**		
Poca Niebla	0.76094	0.87919	0.96087	0.48085	0.44710	0.58638	0.58931	0.59276	0.72830
Media Niebla	0.13333	0.06706	0.08937	0.43210	0.35052	0.38542	0.20378	0.11258	0.14510
Mucha Niebla	0.50620	0.39400	0.15313	0.48902	0.58982	0.59036	0.49746	0.47242	0.24318

(b) Métricas Wide Neural mediante submuestreo en 2018.

Tabla 4.8: Métricas Wide Neural en 2018.

Clase	Verdaderos Positivos			Verdaderos Negativos			Falsos Positivos			Falsos Negativos		
Poca Niebla	831	877	1387	392	259	82	98	22	55	282	445	79
Media Niebla	44	17	24	1239	1243	1466	221	280	67	99	63	46
Mucha Niebla	252	170	37	1099	1165	1503	157	237	33	95	31	30

Clase	Precisión			Exhaustividad			Valor F		
Poca Niebla	0.89451	0.97553	0.96186	0.74663	0.66339	0.94611	0.81391	0.78973	0.95392
Media Niebla	0.16604	0.05724	0.26374	0.30769	0.21250	0.34286	0.21569	0.09019	0.29814
Mucha Niebla	0.61614	0.41769	0.52857	0.72622	0.84577	0.55224	0.66667	0.55921	0.54015

(a) Métricas Wide Neural Network mediante sobremuestreo en 2019.

Clase	Verdaderos Positivos			Verdaderos Negativos			Falsos Positivos			Falsos Negativos		
Poca Niebla	684	735	1098	410	256	119	80	25	18	429	587	368
Media Niebla	57	25	34	1071	1118	1229	389	405	304	86	55	36
Mucha Niebla	219	149	45	1082	1138	1432	174	264	104	128	52	22

Clase	Precisión			Exhaustividad			Valor F		
Poca Niebla	0.89529	0.96711	0.98387	0.61456	0.55598	0.74898	0.72882	0.70605	0.85050
Media Niebla	0.12780	0.05814	0.10059	0.39860	0.31250	0.48571	0.19355	0.09804	0.16667
Mucha Niebla	0.55725	0.36077	0.30201	0.63112	0.74129	0.67164	0.59189	0.48534	0.41667

(b) Métricas Wide Neural mediante submuestreo en 2019.

Tabla 4.9: Métricas Wide Neural en 2019.

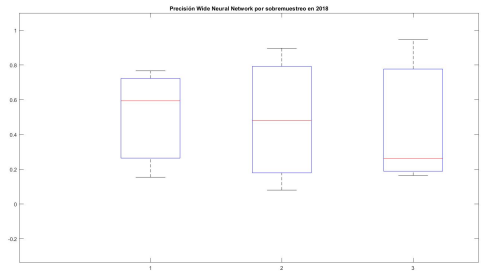
Mediante el análisis de la Tabla 4.8a, podemos observar que al igual que sucedía con el clasificador Fine KNN, no destaca tan claramente. Este clasificador presenta buen rendimiento en los verdaderos positivos y negativos del evento poca niebla. Sin embargo, en los verdaderos positivos y negativos de media y mucha niebla y en los verdaderos negativos, está más repartido, mostrándose superior en algunos valores e inferior en otros. En esta tabla además, podemos observar, al igual que sucedía con las de Conjunto y KNN, que en el punto dos y tres, al reducirse el número de muestras de media y mucha niebla, y aumentar las de poca niebla, empeoran los resultados de los primeros y mejoran los del segundo, fortaleciendo la hipótesis hecha previamente, sobre la relación del número de muestras de entrenamiento con este hecho, que parece afectar a todos los tipos de clasificadores. El punto fuerte de este clasificador van a ser los falsos positivos y negativos, pudiendo observar en esta tabla como éstos se presentan como inferiores en la mayoría de valores. Por último, en cuanto a los valores de precisión, exhaustividad y valor-F, el Wide NN se muestra también superior al resto en la mayoría de valores. Esto puede deberse a que a pesar de no destacar especialmente en los verdaderos positivos y negativos, es más eficiente al ser el que menores falsos positivos y negativos presenta.

En cuanto a las tablas obtenidas mediante submuestreo, si observamos la Tabla 4.2b, podemos descubrir que a pesar de que en éstas el Wide NN no destaca tanto en los verdaderos y los falsos como para sobremuestreo, en las métricas de precisión, exhaustividad y valor-F, sigue presentando los mejores valores de los clasificadores redes neuronales, destacando en muchas de las medidas.

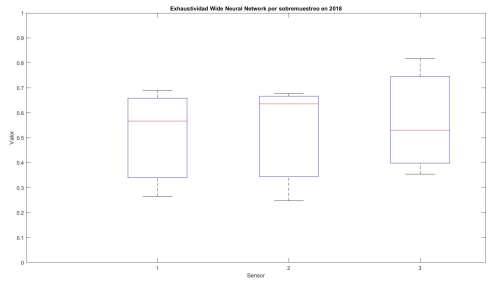
Por último, como se puede ver en las Tablas 4.8a y 4.9a, este clasificador se muestra como el mejor en precisión, exhaustividad y valor-F, cuando se van reduciendo las muestras de prueba, como en el punto tres, especialmente para media y mucha niebla. Por lo tanto, se trata del clasificador más preciso para trabajar en tareas de clasificación con escasez de muestras para estos eventos.

En cuanto al resto de los clasificadores, el Medium NN se proclama como el que presenta unas

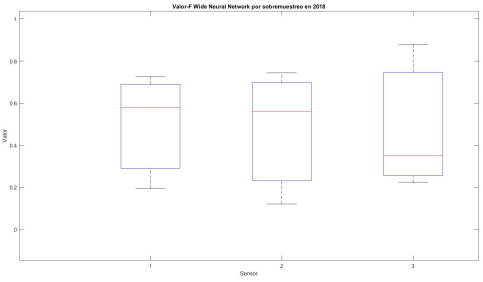
mejores métricas, seguido algo por debajo por el Narrow NN. El Medium NN cuenta con los mejores resultados, especialmente en sobremuestreo, teniendo los mayores verdaderos positivos y los menores falsos positivos y negativos. Además de esto, la diferencia del Medium NN frente al Narrow NN, se va ampliando como se ve en sus tablas, en los puntos 2 y aún más en el 3. Por ello, cabe deducir, que el Medium NN presenta una mayor precisión frente a la reducción del número de muestras que se producen para los eventos de media y mucha niebla.
Finalmente, en relación a los dos clasificadores restantes, el Trilayered se muestra superior al Bilayared en todas las métricas, en la mayoría de valores, tanto para sobremuestreo como submuestreo. Cuenta en general con mayores verdaderos positivos y negativos, y valores de precisión, exhaustividad y valor-F, así como menores falsos positivos y negativos, además de mostrarse superior en los puntos dos y tres y teniendo por consiguiente una mayor resistencia.
Tras haber analizado las tablas numéricas, vamos a estudiar su representación en diagramas de cajas:



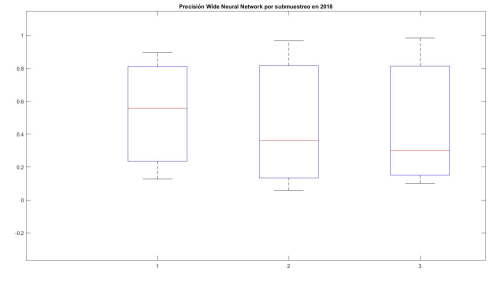
(a) Precisión para Sobremuestreo en 2018



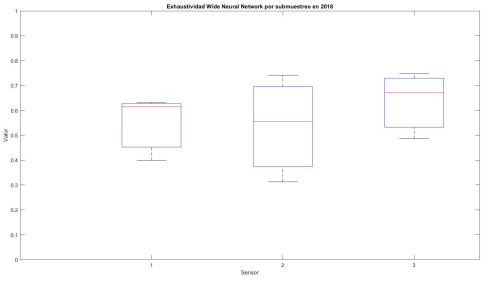
(b) Exhaustividad para Sobremuestreo en 2018



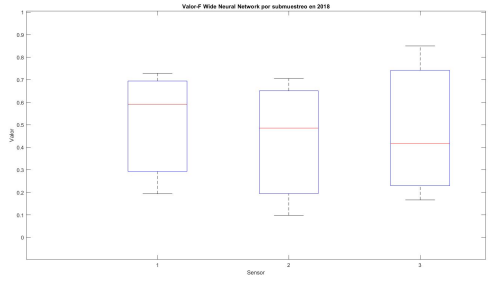
(c) Valor-F para Sobremuestreo en 2018



(d) Precisión para Submuestreo en 2018

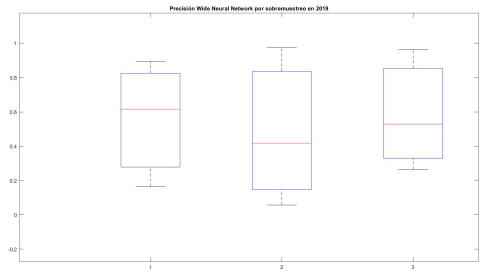


(e) Exhaustividad para Submuestreo en 2018

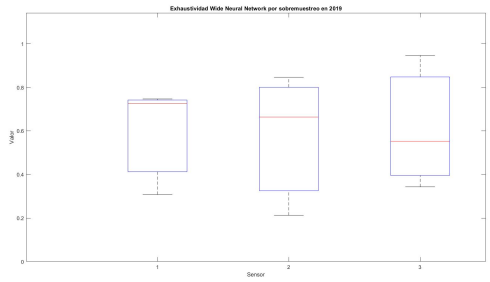


(f) Valor-F para Submuestreo en 2018

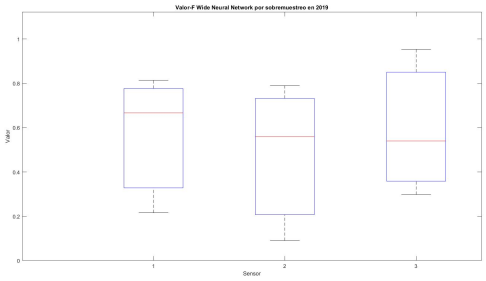
Figura 4.12: Métricas Wide NN en 2018



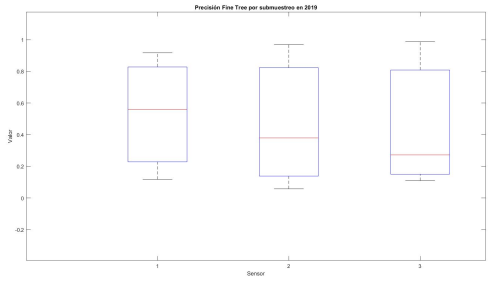
(a) Precisión para Sobremuestreo en 2019



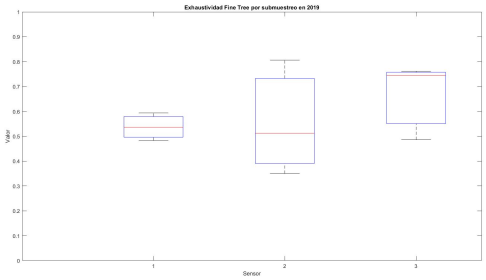
(b) Exhaustividad para Sobremuestreo en 2019



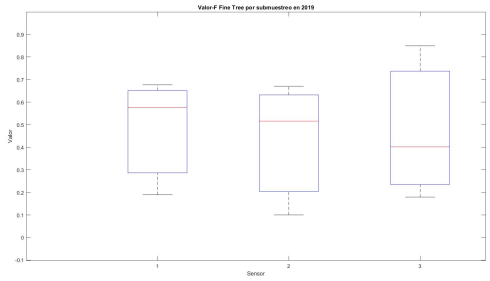
(c) Valor-F para Sobremuestreo en 2019



(d) Precisión para Submuestreo en 2019



(e) Exhaustividad para Submuestreo en 2019



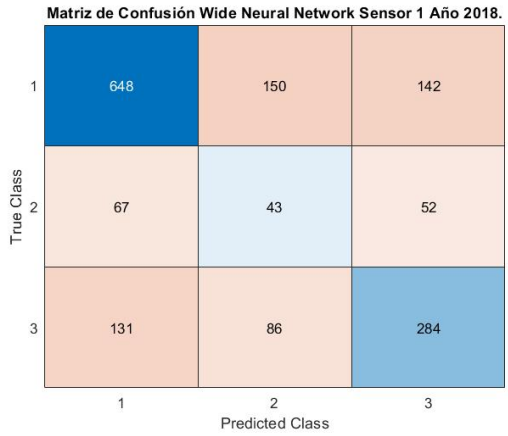
(f) Valor-F para Submuestreo en 2019

Figura 4.13: Métricas Wide NN en 2019

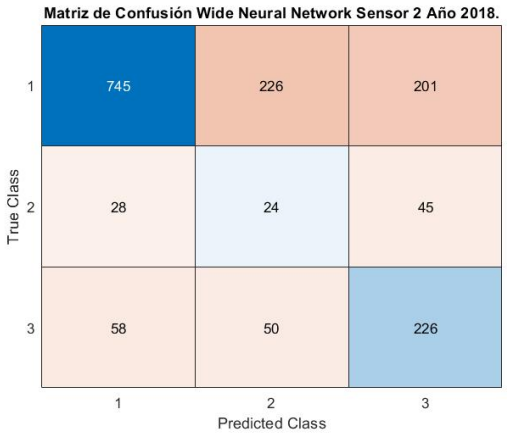
A partir de las Figuras 4.12 y 4.13, se confirma lo que ya habíamos apreciado por medio de las tablas. Si se observan las Figuras, por ejemplo, para sobremuestreo en 2018, 4.7a, 4.12b y 4.12c, se aprecia claramente la superioridad de las métricas de este clasificador con respecto al resto, superando sus valores a los del resto en casi todas las medianas. En cambio, para submuestreo, si analizamos los Figuras 4.7d, 4.12e y 4.12f, se observa que está mucho más igualado. Éste tiene sentido dado que como se observó de forma numérica, para submuestreo el Wide NN no destacaba claramente sobre el resto, estando igualado entre los mejores predictores de redes neuronales para submuestreo, mostrándose superior en algunas medidas y no tanto en otras. Por último, en cuanto al resto de los clasificadores, tras el Wide NN se encuentran el Medium NN y Narrow NN, como los que presentan las siguientes mejores gráficas de las métricas; y algo por debajo se encuentran el Trilayered NN y el Bilayered NN para cerrar la lista de los clasificadores de redes neuronales.

Una vez terminamos de analizar las métricas, vamos a pasar a analizar las matrices de confusión, que nos permiten observar de una manera muy clara los resultados del experimento y comparar

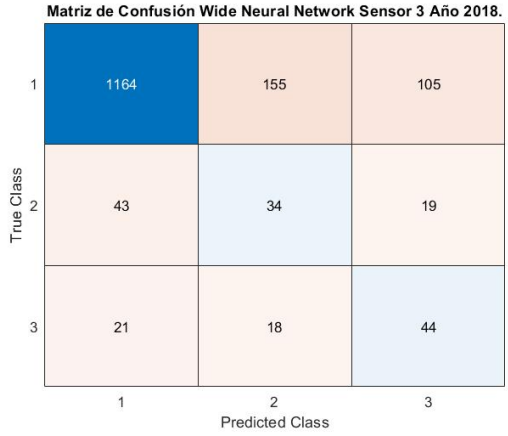
clasificadores entre sí.



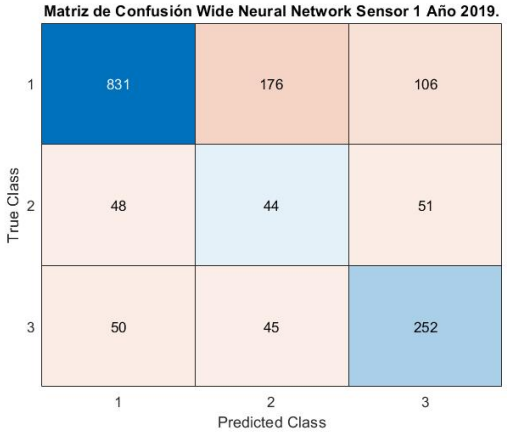
(a) Punto 1 en 2018



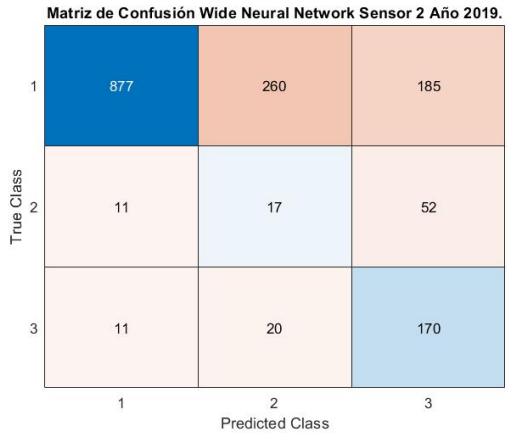
(b) Punto 2 en 2018



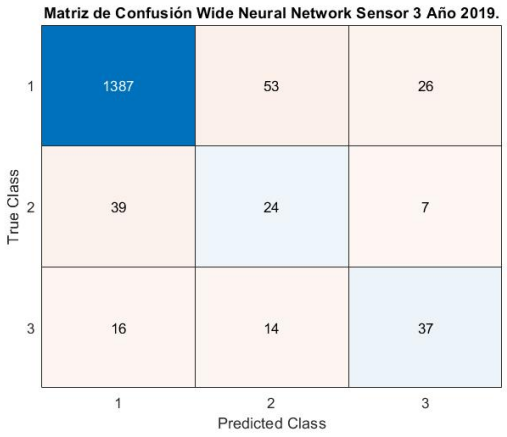
(c) Punto 3 en 2018



(d) Punto 1 en 2019

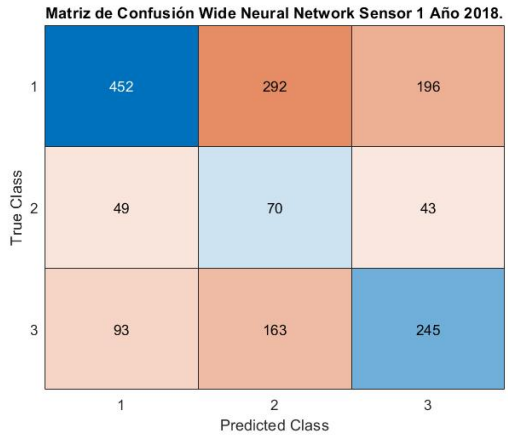


(e) Punto 2 en 2019

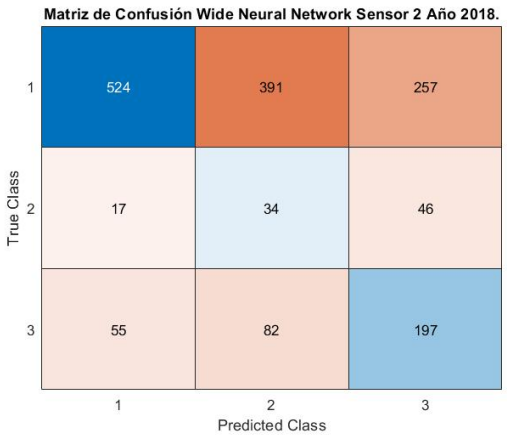


(f) Punto 3 en 2019

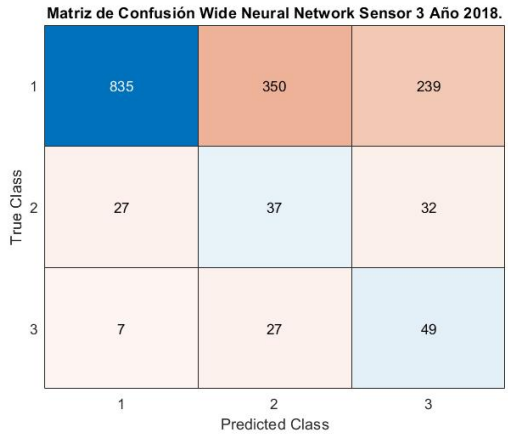
Figura 4.14: Matrices de Confusión del Wide NN mediante Sobremuestreo



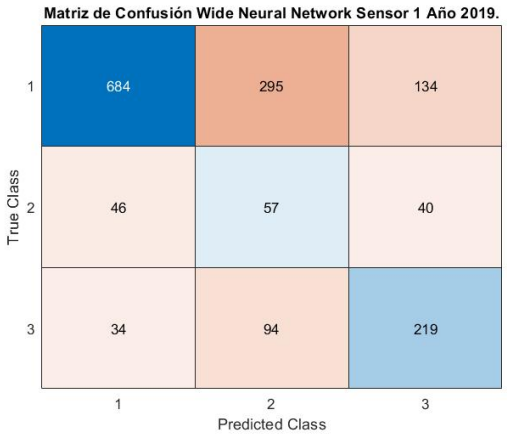
(a) Punto 1 en 2018



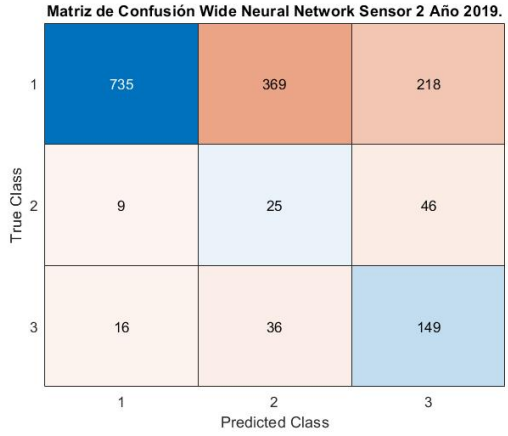
(b) Punto 2 en 2018



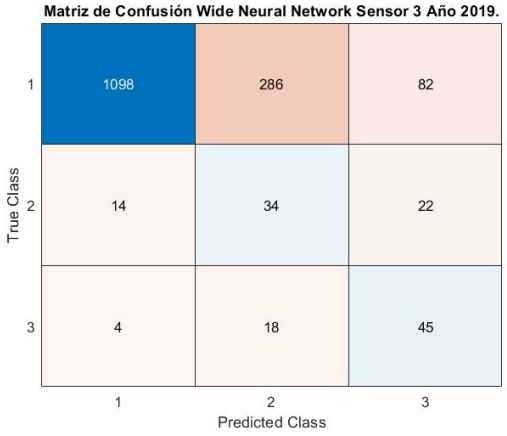
(c) Punto 3 en 2018



(d) Punto 1 en 2019



(e) Punto 2 en 2019



(f) Punto 3 en 2019

Figura 4.15: Matrices de Confusión del Wide NN mediante Submuestreo

Si observamos las matrices de sobremuestreo del Wide NN 4.14, por ejemplo, en el punto uno 4.14a, y lo comparamos con las del resto de clasificadores, se puede apreciar claramente la superioridad que presenta este clasificador sobre el resto, obteniendo éste el menor número de falsos en la mayoría de medidas, con respecto al resto de clasificadores. El Wide NN no obtiene, como ya se mencionó, los verdaderos positivos y negativos más altos, pero obtiene la

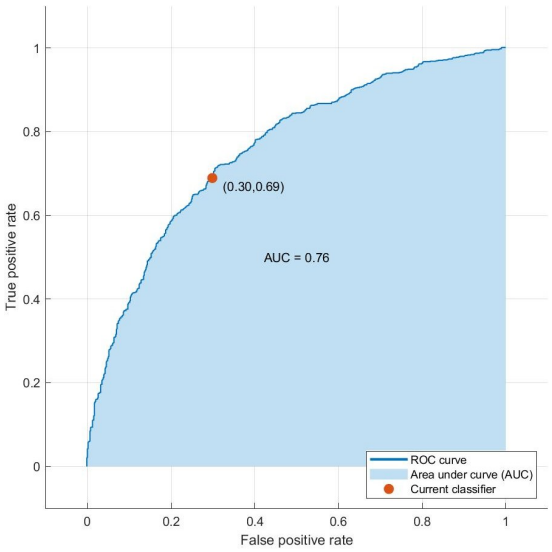
mejor relación verdaderos/falsos de los clasificadores, beneficiándose de su menor números de falsos positivos y negativos, especialmente en los eventos de media y mucha niebla. Además, es de destacar, que el Wide NN sobresale como el mejor clasificador para media y mucha niebla para sobremuestreo, como se puede apreciar claramente por ejemplo, en los puntos uno o tres en 2018 en las Figuras 4.14a y 4.14c, en los tres sensores. Por último, si se observan las matrices de confusión en 2018 de los puntos uno y tres tanto para sobremuestreo en las Figuras 4.14a y 4.14c, como para submuestreo en las Figuras 4.15a y 4.15c, se aprecia como el Wide NN es el clasificador que mejor evoluciona del punto uno al tres. En sobremuestreo se mantiene como el mejor clasificador tanto para el punto uno como el tres, y en submuestreo, a pesar de no mostrar una actuación tan destacable en el punto uno, es de los que mejor evoluciona también al punto tres, mostrando los mejores resultados en este punto. Por lo tanto, el Wide NN es el clasificador que mejor gestiona la reducción de muestras en los puntos dos y tres.

En cuanto al resto de clasificadores, el Medium NN es el siguiente mejor, obteniendo los verdaderos positivos más altos en la mayoría de clases frente a los otros clasificadores restantes; así como menores falsos positivos y negativos. También destaca por ser el siguiente clasificador con una mejor evolución a los puntos dos y tres después del Wide NN, tanto para sobremuestreo como para submuestreo.

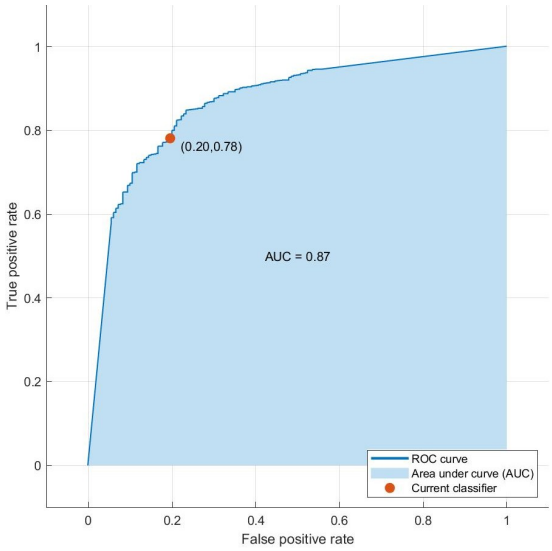
Después de éste destaca el Narrow NN, presentando en general los siguientes resultados más eficientes, mayores verdaderos positivos y negativos y unos falsos positivos y negativos en torno a los obtenidos por el resto. El algoritmo Bilayared es el siguiente con mejores valores, aunque suele mostrar unos verdaderos positivos y negativos bajos comparado con los del resto. Destaca por su evolución, siendo también de los que muestra una mejor evolución, obteniendo en el punto tres unos falsos positivos y negativos más bajos y unos verdaderos positivos y negativos para media y mucha niebla comparables a los de Medium y Narrow. Este clasificador destaca además para submuestreo. Por último tenemos el Trilayared NN, que es el que presenta unos resultados más pobres, teniendo unos falsos positivos y negativos muy elevados, destacando por sus relativamente altos valores para los verdaderos positivos para media y mucha niebla.

Finalmente, al igual que se ha venido apreciando en los otros tipos de clasificadores, los principales errores de predicción en las matrices NN se producen entre las clases mayoritarias, poca niebla con mucha niebla y ésta y media niebla con poca, como se puede ver por ejemplo en la Figura 4.14b. Además, también se produce un empeoramiento de las predicciones del punto uno al dos y tres para media y mucha niebla, como se puede observar al evolucionar de la Figura 4.14a a la 4.14c, mejorando para poca niebla; así como un aumento en el número de errores en submuestro con respecto a sobremuestreo del evento poca niebla, debido a como ya se dijo por la gran reducción en su número de muestras de entrenamiento.

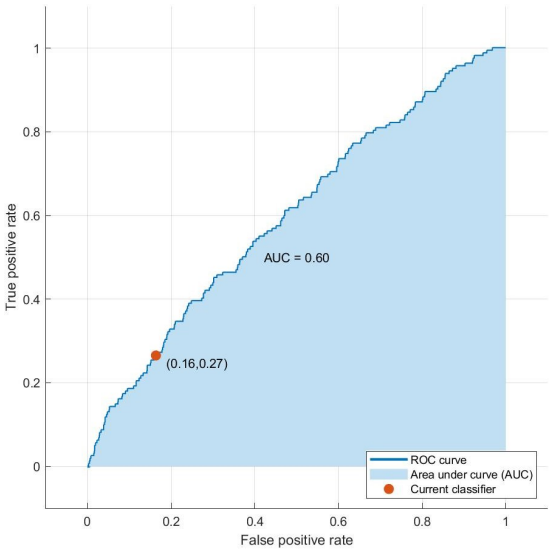
Por último, vamos a analizar las gráficas de las curvas ROC y las áreas AUC, para el Wide NN 4.16:



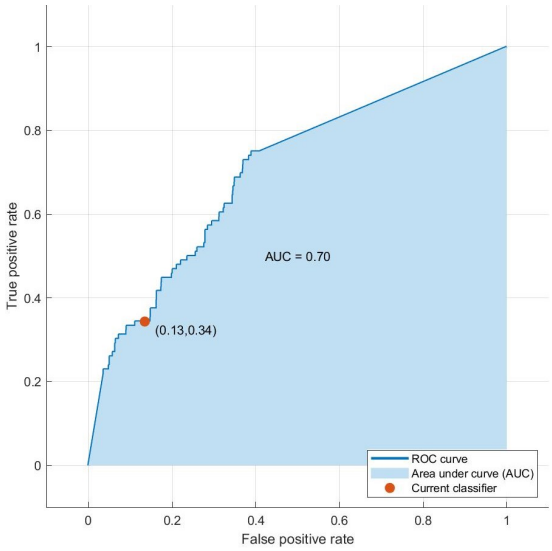
(a) Clase Positiva Evento Poca Niebla en el Punto 1.



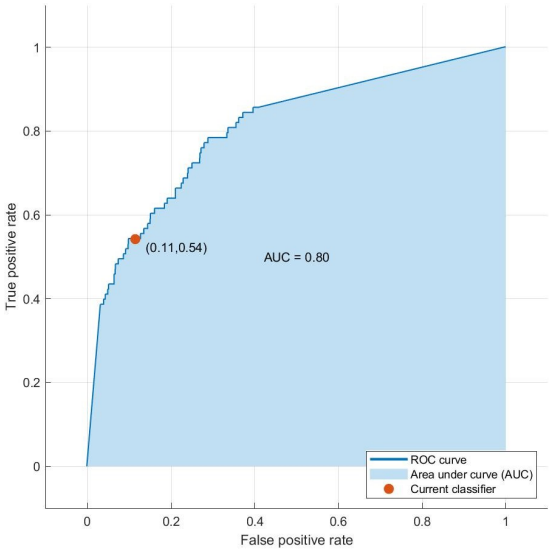
(b) Clase Positiva Evento Poca Niebla en el Punto 3.



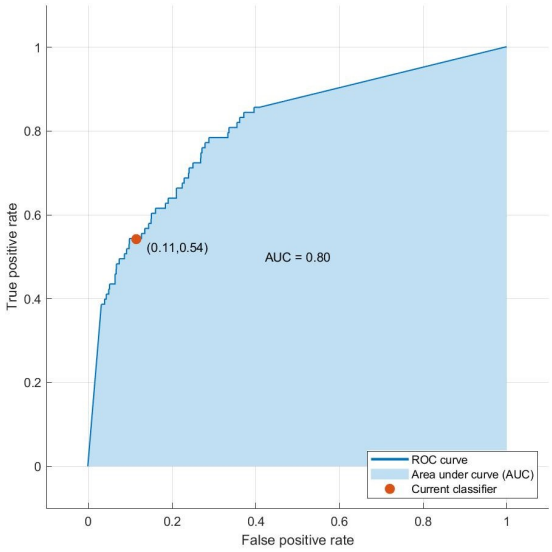
(c) Clase Positiva Evento Media Niebla en el Punto 1.



(d) Clase Positiva Evento Media Niebla en el Punto 3.



(e) Clase Positiva Evento Mucha Niebla en el Punto 1.



(f) Clase Positiva Evento Mucha Niebla en el Punto 3.

Figura 4.16: ROC y AUC de Wide NN en el punto 1 y 3 por sobremuestreo durante 2018.

Si nos paramos a observar las gráficas de los clasificadores, podemos resaltar varios detalles. Por ejemplo, al igual que sucedía en los clasificadores de Conjunto, el AUC del suceso mucha niebla, como en la Figuras 4.16e y 4.16f, es superior a los de poca niebla, como en las 4.16a y 4.16b, ocurriendo este hecho en la mayoría de gráficas de todos los clasificadores NN. Esto podría respaldar la hipótesis hecha para los clasificadores de Conjunto de tener un mayor AUC de mucha niebla debido al menor número de muestras. Por lo tanto, con un menor número de muestras, si no tienes una tasa de fallo de predicciones muy elevada cometerás menos errores al predecir, que con una tasa de aciertos más alta pero con muchas más muestras, como sucede aquí. Además, se produce igualmente un aumento del AUC del punto uno al tres, como se puede ver en las Figuras 4.16, en la que se muestra un ejemplo del punto uno frente al tres. Ésto puede ser, como se acaba de señalar, debido al menor número de muestras en este punto frente al del uno. Sin embargo, también tiene que haber un mínimo de precisión al predecir, razón por la cual el evento media niebla es el peor. Debido a sus muy bajas muestras en comparación al resto, no logra la precisión necesaria para poder lograr un AUC superior.
Por último, en comparación a los otros clasificadores, en sobremuestreo destaca el Wide NN como el que presenta las AUC más elevadas; esto es debido a que, como vimos tanto en las tablas como en las matrices de confusión, es el que menores fallos de predicción presenta. En cuanto a submuestreo, aquí el Wide NN no destaca tanto como para sobremuestreo y por ello sus AUC no son tan elevados, habiendo clasificadores con algunas AUC mayores.

A partir de todo lo visto en este apartado podemos concluir que el Wide NN es el mejor clasificador de las redes neuronales creadas. Se trata del que comete menores fallos de predicción, presentando unos aciertos de los más elevados, y el mejor a la hora de predecir eventos media y mucha niebla. Por el contrario, hay que tener en cuenta que para submuestreo pierde mucha eficacia, aunque se mantiene con algunas de las mejores métricas obtenidas.

4.2.4. SVM

Ahora vamos a pasar a examinar los clasificadores SVM. Analizando la Tabla 4.10 de medias de estos clasificadores, se aprecia claramente que el Fine Gaussian SVM destaca sobre el resto. Este posee claramente las exactitudes más altas tanto para sobremuestreo como submuestreo, especialmente para el primero. Por lo tanto, elegimos este clasificador como el mejor basado en máquinas de vectores soporte.

Medias Exactitud	Coarse Gaussian SVM	Fine Gaussian SVM	Quadratic SVM	Cubic SVM	Linear SVM	Medium Gaussian SVM
Sobremuestreo	57.56666	**73.76666**	58.93333	54.78333	56.96666	61.833
Submuestreo	53.63333	**61.633333**	56.465	57.01666	56.15	57.25

Tabla 4.10: Tabla Medias Exactitudes Clasificadores de SVM

A continuación, vamos a pasar a estudiar los resultados obtenidos por los clasificadores SVM creados, empezando como siempre con las tablas de las métricas.

Clase	Verdaderos Positivos			Verdaderos Negativos			Falsos Positivos			Falsos Negativos		
Poca Niebla	694	744	1240	498	366	115	138	65	64	273	428	184
Media Niebla	33	23	28	1271	1279	1371	203	227	136	96	74	68
Mucha Niebla	347	254	38	908	979	1423	188	290	97	160	80	45

Clase	Precisión			Exhaustividad			Valor F		
Poca Niebla	0.83413	0.91965	0.95092	0.71768	0.63481	0.87079	0.77154	0.75114	0.90909
Media Niebla	0.13983	0.09200	0.17073	0.25581	0.23711	0.29167	0.18082	0.13256	0.21538
Mucha Niebla	0.64860	0.46691	0.28148	0.68442	0.76048	0.45783	0.66603	0.57859	0.34862

(a) Métricas Fine Gaussian SVM mediante sobremuestreo en 2018.

Clase	Verdaderos Positivos			Verdaderos Negativos			Falsos Positivos			Falsos Negativos		
Poca Niebla	470	518	1011	545	361	149	118	70	30	470	654	413
Media Niebla	80	26	44	998	1009	1198	443	497	309	82	71	52
Mucha Niebla	285	220	47	895	997	1358	207	272	162	216	114	36

Clase	Precisión			Exhaustividad			Valor F		
Poca Niebla	0.79932	0.88095	0.97118	0.50000	0.44198	0.70997	0.61518	0.58864	0.82028
Media Niebla	0.15296	0.04971	0.12465	0.49383	0.26804	0.45833	0.23358	0.08387	0.19599
Mucha Niebla	0.57927	0.44715	0.22488	0.56886	0.65868	0.56627	0.57402	0.53269	0.32192

(b) Métricas Fine Gaussian SVM mediante submuestreo en 2018.

Tabla 4.11: Métricas Fine Gaussian SVM en 2018.

Clase	Verdaderos Positivos			Verdaderos Negativos			Falsos Positivos			Falsos Negativos		
Poca Niebla	884	918	1373	411	237	88	79	44	49	229	404	93
Media Niebla	40	9	25	1271	1301	1449	189	222	84	103	71	45
Mucha Niebla	260	164	38	1105	1156	1502	151	246	34	87	37	29

Precisión			Exhaustividad			Valor F			
Poca Niebla	0.91796	0.95426	0.96554	0.79425	0.69440	0.93656	0.85164	0.80385	0.95083
Media Niebla	0.17467	0.03896	0.22936	0.27972	0.11250	0.35714	0.21505	0.05788	0.27933
Mucha Niebla	0.63260	0.40000	0.52778	0.74928	0.81592	0.56716	0.68602	0.53682	0.54676

(a) Métricas Fine Gaussian SVM mediante sobremuestreo en 2019.

Clase	Verdaderos Positivos			Verdaderos Negativos			Falsos Positivos			Falsos Negativos		
Poca Niebla	687	739	1243	425	248	114	65	33	23	426	583	223
Media Niebla	68	19	33	1066	1078	1369	394	445	164	75	61	37
Mucha Niebla	236	153	50	1103	1188	1446	153	214	90	111	48	17

Clase	Precisión			Exhaustividad			Valor F		
Poca Niebla	0.91356	0.95725	0.98183	0.61725	0.55900	0.84789	0.73673	0.70583	0.90996
Media Niebla	0.14719	0.04095	0.16751	0.47552	0.23750	0.47143	0.22479	0.06985	0.24719
Mucha Niebla	0.60668	0.41689	0.35714	0.68012	0.76119	0.74627	0.64130	0.53873	0.48309

(b) Métricas Fine Gaussian SVM mediante submuestreo en 2019.

Tabla 4.12: Métricas Fine Gaussian SVM en 2019.

Después de analizar detenidamente las tablas de las métricas podemos destacar una serie de observaciones sobre ellas. El Fine Gaussian demuestra ser el mejor clasificador del grupo. Es superior al resto tanto en que, presenta los más elevados verdaderos positivos y negativos y especialmente importante, menores falsos positivos y negativos. Además de esto, posee mayores valores de precisión, exhaustividad y valor-F. Como podemos ver, al comparar las Tablas 4.11a y 4.11b, con las de sobremuestreo y submuestreo del resto de clasificadores, esta superioridad del Fine Gaussian se produce tanto para sobremuestreo como submuestreo, aunque se reduce la diferencia para este último. También hay que destacar que se trata del que posee mejores resultados al reducirse las muestras de los eventos de media y mucha niebla en los puntos dos y

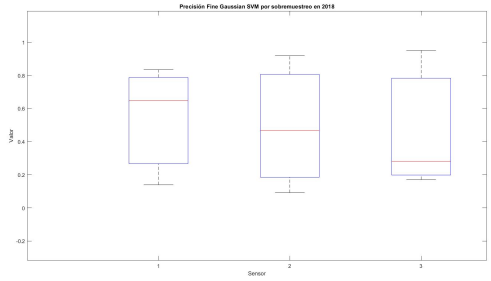
tres.

En cuanto al resto de los clasificadores, podemos destacar al Medium Gaussian como el siguiente mejor clasificador, tanto en sobremuestreo como submuestreo. Este se postula por sus mejores resultados especialmente para poca y mucha niebla, poseyendo para estos eventos la mayor parte de mejores verdaderos positivos y negativos, menores falsos positivos y negativos y también en precisión, exhaustividad y valor-F. También destaca como el mejor al ir evolucionando a los puntos dos y tres, especialmente en poca y mucha niebla. Sin embargo, cabe destacar su debilidad en los eventos de media niebla.

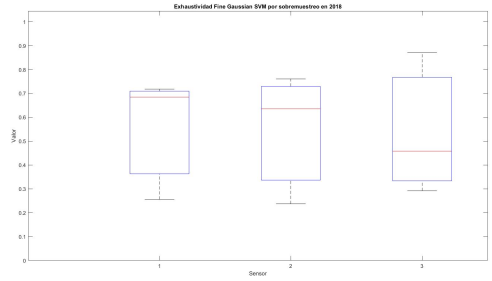
Tras el Medium Gaussian podemos destacar al Quadratic, que destaca sobre los otros tres restantes teniendo mejores verdaderos positivos y negativos, especialmente para poca y media niebla, aunque menores falsos positivos y negativos en media y mucha niebla. También destaca por sus mejores valores de precisión, exhaustividad y de valor-F, sobre todo en los eventos poca y mucha niebla.

Por último, tenemos a los clasificadores Coarse Gaussian, Cubic y Linear que muestran unos resultados parejos como se observa en la Tabla 4.10. Por ejemplo, Coarse y Cubic destacan especialmente para los eventos poca niebla, mientras que Linear destaca para los de media y mucha niebla, pudiendo ser el mejor dentro de este grupo de tres. También cabe destacar los resultados obtenidos por Cubic SVM para submuestreo, donde como se observa en la Tabla 4.10 se postula como el mejor de los tres restantes, Coarse Gaussian y Linear, demostrándose en las tablas de métricas de submuestreo, que posee mayores verdaderos positivos y negativos y menores falsos positivos y negativos, obteniendo especialmente mejores resultados para poca y media niebla.

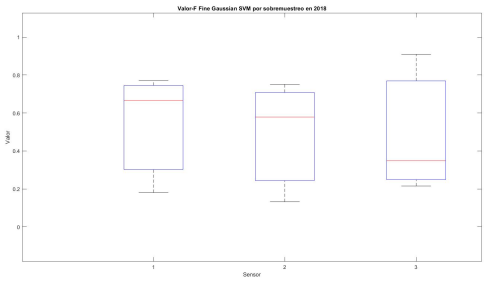
Una vez hemos analizado los datos numéricos de los clasificadores SVM, vamos a observar y analizar sus representaciones en diagramas de cajas:



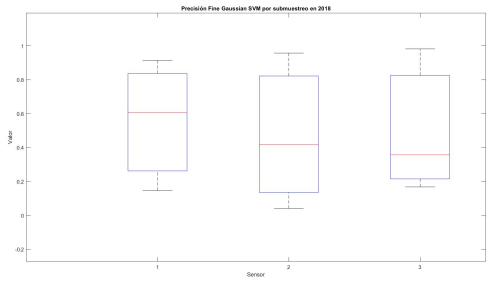
(a) Precisión para Sobremuestreo en 2018



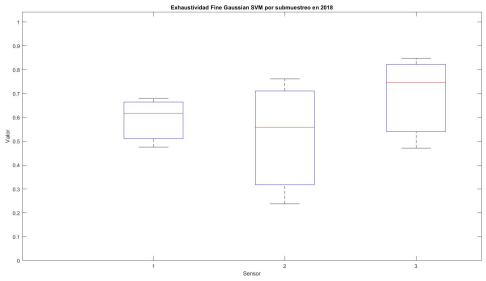
(b) Exhaustividad para Sobremuestreo en 2018



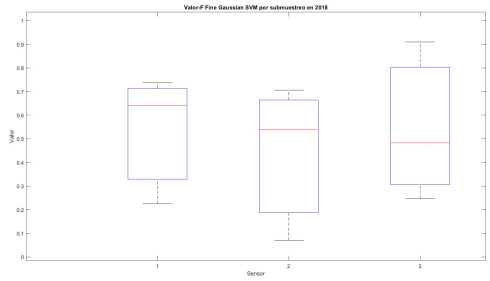
(c) Valor-F para Sobremuestreo en 2018



(d) Precisión para Submuestreo en 2018

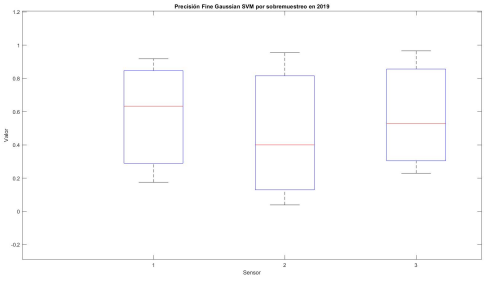


(e) Exhaustividad para Submuestreo en 2018

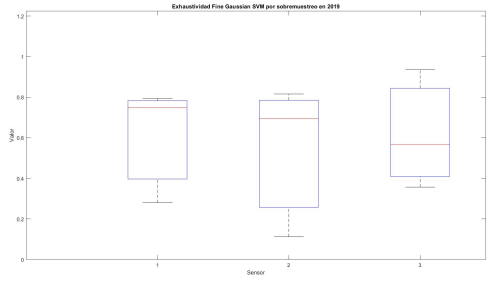


(f) Valor-F para Submuestreo en 2018

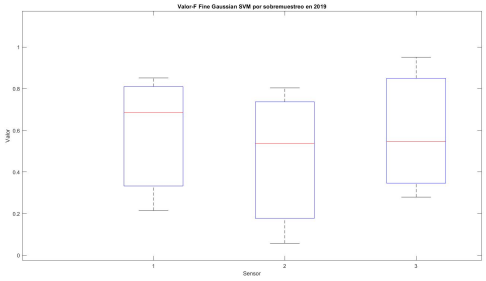
Figura 4.17: Métricas Fine Gaussian SVM en 2018



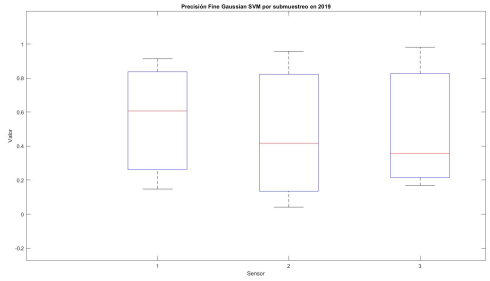
(a) Precisión para Sobremuestreo en 2019



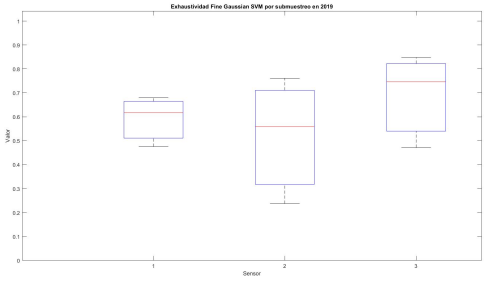
(b) Exhaustividad para Sobremuestreo en 2019



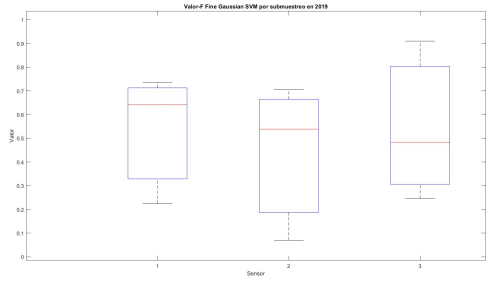
(c) Valor-F para Sobremuestreo en 2019



(d) Precisión para Submuestreo en 2019



(e) Exhaustividad para Submuestreo en 2019



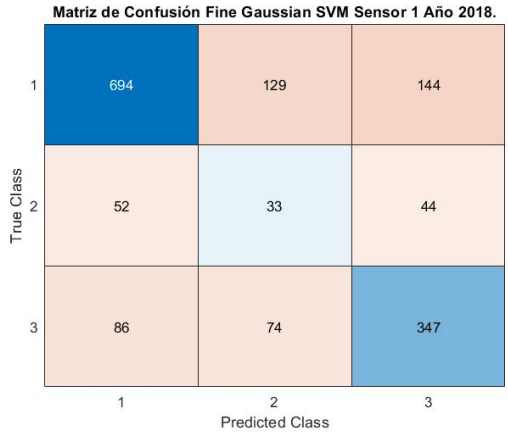
(f) Valor-F para Submuestreo en 2019

Figura 4.18: Métricas Fine Gaussian SVM en 2019

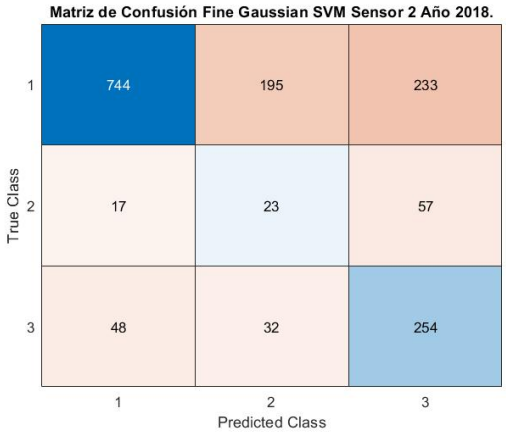
A partir de estas gráficas, podemos corroborar los datos ya observados en las tablas, declarándose el Fine Gaussian como el mejor clasificador. Si tomamos por ejemplo, las Figuras de sobremuestreo 4.17a, 4.17b y 4.17a, podemos comprobar cómo este clasificador destaca sobre el resto en la gran mayoría de las medianas obtenidas. Sin embargo, al observar en las Figuras 4.17d, 4.17e y 4.17f, vemos que para submuestreo, no destaca tanto, siendo superior en algunas medidas e inferior en otras.

En cuanto al resto de clasificadores, el Medium Gaussian obtiene los mejores diagramas después del Fine Gaussian, destacando sobre los restantes tanto para sobre como para submuestreo. Tras él iría el Quaadratic SVM, aunque de forma más igualada y menos destacada sobre los tres restantes, y por último, con unos resultados bastante parejos el Coarse Gaussian SVM, Cubic SVM y Linear SVM.

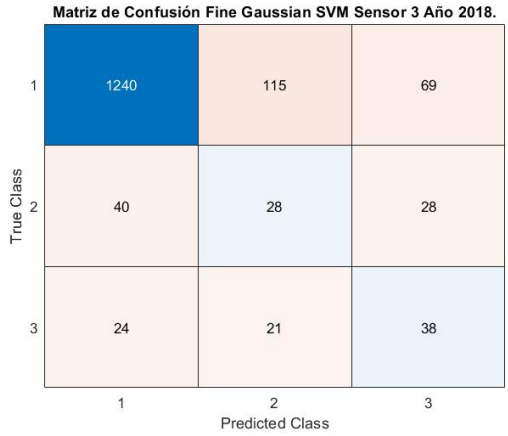
Una vez analizadas las gráficas de las metricas, vamos a analizar las matrices de confusión de los clasificadores SVM:



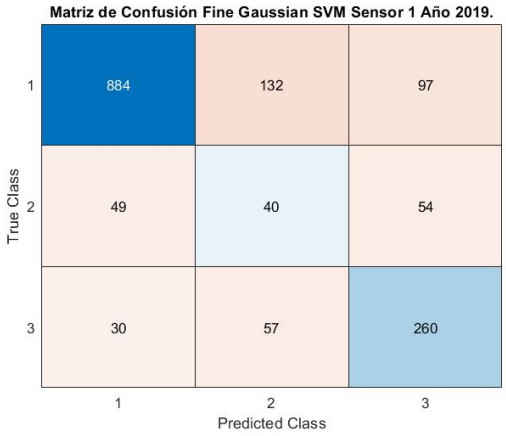
(a) Punto 1 en 2018



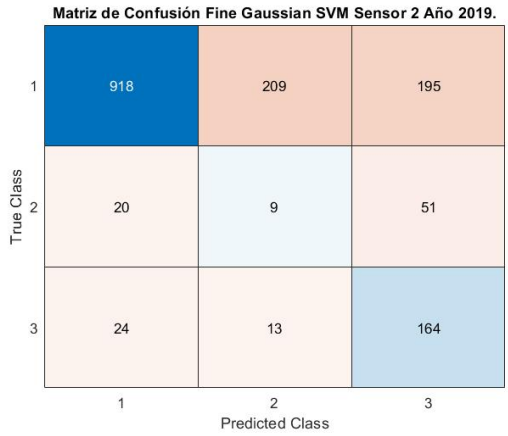
(b) Punto 2 en 2018



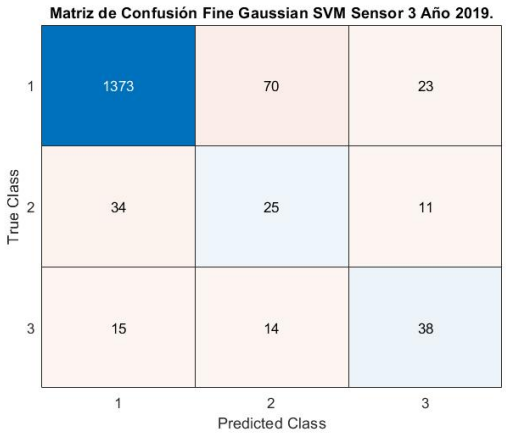
(c) Punto 3 en 2018



(d) Punto 1 en 2019

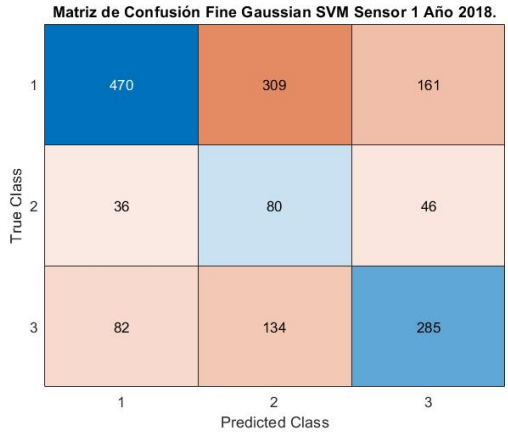


(e) Punto 2 en 2019

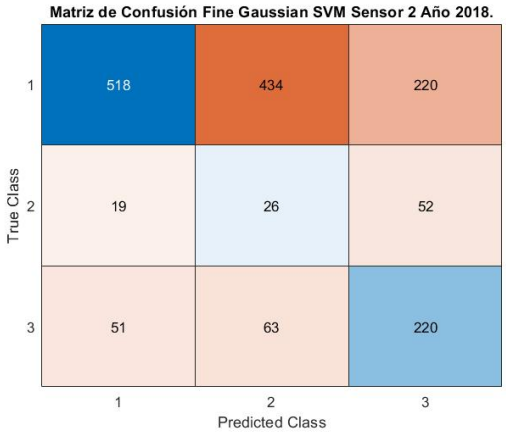


(f) Punto 3 en 2019

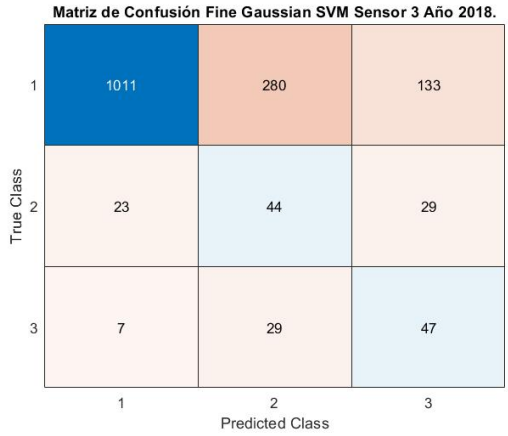
Figura 4.19: Matrices de Confusión del Fine Gaussian SVM mediante Sobremuestreo



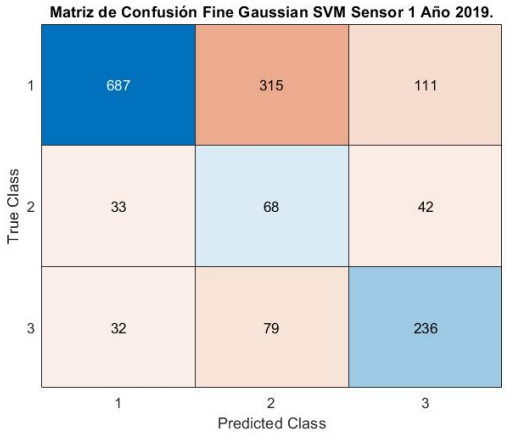
(a) Punto 1 en 2018



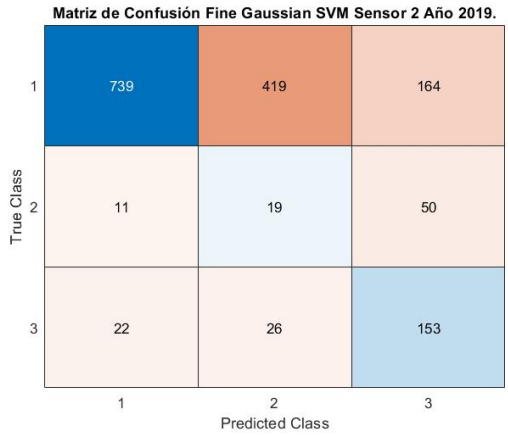
(b) Punto 2 en 2018



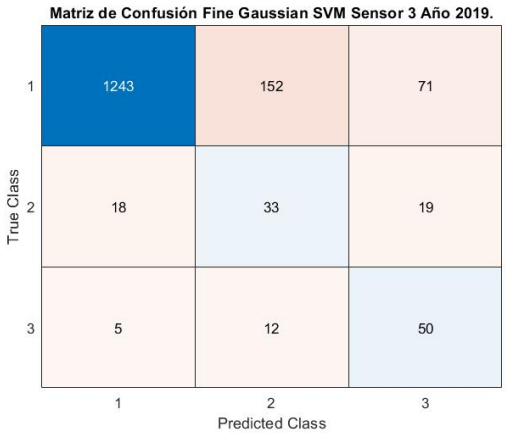
(c) Punto 3 en 2018



(d) Punto 1 en 2019



(e) Punto 2 en 2019



(f) Punto 3 en 2019

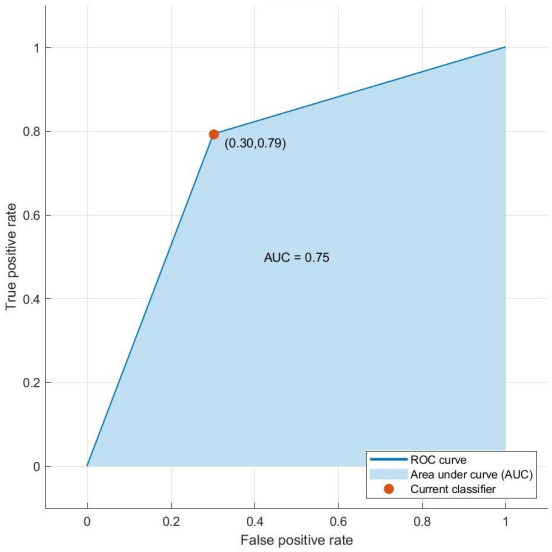
Figura 4.20: Matrices de Confusión del Fine Gaussian SVM mediante Submuestreo

A partir de estas matrices de confusión podemos realizar una serie de observaciones. Para empezar, si examinamos las matrices de sobremuestreo de Fine Gaussian, como las de la Figura 4.19, y las comparamos con las del resto de los clasificadores, podemos observar como éste es el que posee unos mayores verdaderos positivos y negativos, especialmente para poca y mucha niebla; así como los menores falsos positivos y negativos, destacando para los eventos media y mucha niebla. En cuanto a submuestreo en la Figura 4.20, el Fine Gaussian no es tan eficiente, destacando en algunos verdaderos y falsos y siendo superado en otros, especialmente para los eventos poca y mucha niebla. También cabe destacar sobre el Fine Gaussian, que este es al que menos le afecta la pérdida de muestras en los puntos dos y tres, tanto para sobremuestreo como para submuestreo, siendo por lo tanto muy superior en cuanto al escaso número de errores de predicción cometidos, aunque es superado en bastantes medidas de verdaderos positivos.

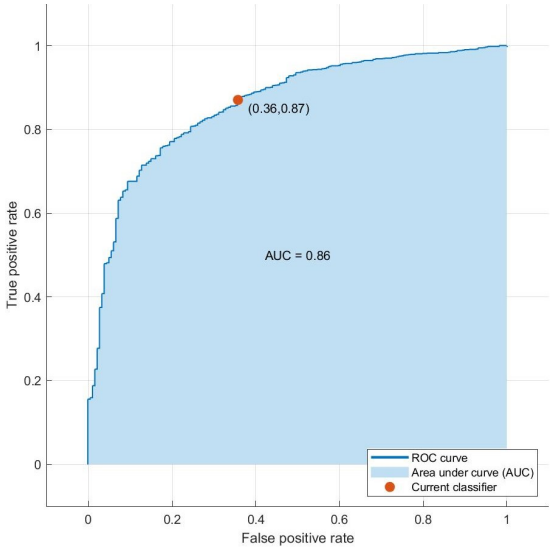
Con respecto a las matrices de confusión SVM, se produce una reducción en los puntos dos y tres en el número de predicciones correctas para los eventos de media y mucha niebla y un aumento en los aciertos para poca niebla. Esto es debido a la reducción en el número de muestras en los sensores dos y tres de estos eventos y a un aumento de las de poca niebla, teniendo menos y más muestras reales respectivamente de estos eventos para entrenar y para la fase de pruebas. Por último, cabe destacar también el empeoramiento notable de los resultados para submuestreo de los eventos de poca niebla y la apenas variación de los de media y mucha niebla. Esto es debido a que el principal afectado por la reducción de muestras es el evento poca niebla, y al sólo tener muestras reales en submuestreo de los eventos de media y mucha niebla.

En cuanto al resto de clasificadores, el siguiente que presenta mejores resultados es el Medium Gaussian, destacando especialmente sobre el resto por su menor número de fallos para los eventos de media y mucha niebla, y por ser el que mejor evoluciona hasta el sensor tres con la reducción de muestras, tanto para sobremuestreo como submuestreo. Después de éste iría el Quadratic, que posee unos mayores aciertos y menores fallos, especialmente para media y mucha niebla, y por último, Coarse Gaussian, Cubic y Linear, entre los cuales parece destacar

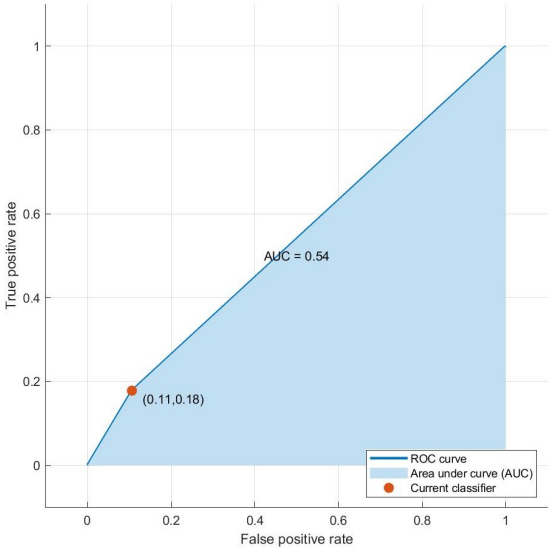
Coarse Gaussian, con sus mayores aciertos y menores fallos, especialmente en sobremuestreo. Finalmente, una vez analizadas las matrices de confusión, vamos a terminar el análisis con las gráficas ROC y el AUC:



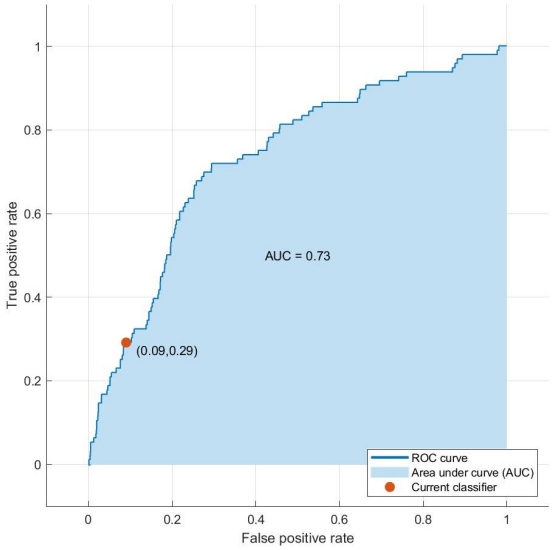
(a) Clase Positiva Evento Poca Niebla en el Punto 1.



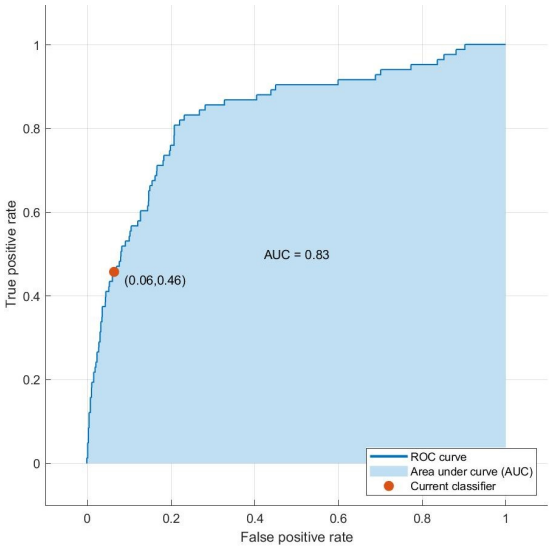
(b) Clase Positiva Evento Poca Niebla en el Punto 3.



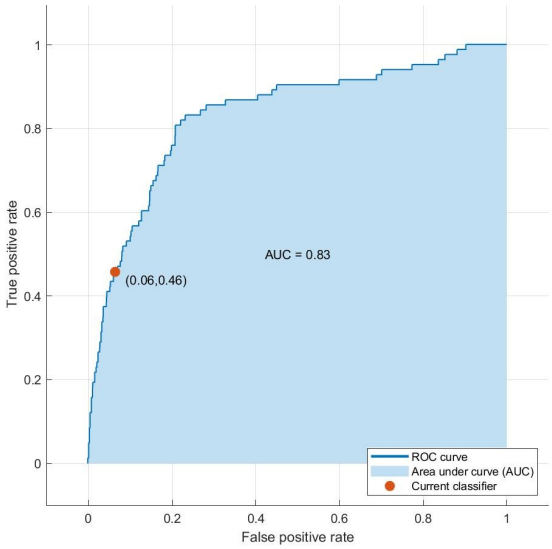
(c) Clase Positiva Evento Media Niebla en el Punto 1.



(d) Clase Positiva Evento Media Niebla en el Punto 3.



(e) Clase Positiva Evento Mucha Niebla en el Punto 1.



(f) Clase Positiva Evento Mucha Niebla en el Punto 3.

Figura 4.21: ROC y AUC de Fine Gaussian SVM en el punto 1 y 3 por sobremuestreo en 2018.

A partir del análisis de estas gráficas, podemos destacar, al igual que ha ocurrido en la mayoría de los clasificadores observados hasta el momento, como se se puede apreciar en la Figura 4.21a, que el AUC de los eventos de mucha niebla suele ser superior al de poca niebla, así como el de media niebla es el peor como se ve en la Figura 4.21c, apoyando la hipótesis formulada sobre que el AUC depende del número de muestras de cada evento. En relación a este hecho, el menor número de muestras también influye en que al pasar del punto uno al tres aumenten las AUC de los tres eventos, como se aprecia en la Figura 4.21. Por último, cabe resaltar el empeoramiento de los resultados en submuestreo. Este deterioro puede ser debido a que, a pesar de tener menos muestras, este tipo de clasificador es menos eficiente en submuestreo y por lo tanto tiene una menor AUC.
Finalmente, en cuanto a la comparación entre clasificadores, se confirman los resultados obtenidos hasta ahora, al comparar la Figura 4.21 con el resto de clasificadores, se obtiene que éste es el que posee unas mayores AUC, después del cual destacan el Medium, el Cuadratic y por último, Coarse Gaussian, Linear y Cubic, entre los que parece destacar Coarse Gaussian ligeramente en sobremuestreo y Cubic para submuestreo.

A partir de todos los datos analizados, podemos destacar al Fine Gaussian como el mejor clasificador de los SVM, destacando entre ellos por ser el clasificador con menores fallos de predicción, y por ser el más capaz de predecir los eventos de media y mucha niebla, al soportar mejor la reducción en el número de muestras de estos eventos. Sin embargo, hay que tener en cuenta la reducción en la eficiencia que sufre este clasificador en submuestreo, aunque obteniendo de los mejores resultados entre estos clasificadores.

4.2.5. Árboles de Decisión

Finalmente, llegamos al último tipo de clasificadores, los árboles de decisión. Al igual que hemos ido haciendo hasta ahora, vamos a utilizar el criterio de las medias de exactitud para elegir el mejor clasificador de este tipo. Como se puede ver en la Tabla 4.13, el clasificador que presenta unos mejores resultados es el Fine Tree, el cual a pesar de tener una media inferior en submuestreo frente al Medium Tree, al hacer la media con sobremuestreo es el que posee un mejor rendimiento, siendo por lo tanto el elegido en este tipo.

Medias Exactitud	Coarse Tree	Fine Tree	Medium Tree
Sobremuestreo	52.73333	**65.5**	60.51666
Submuestreo	55.63333	**56.65**	57.9

Tabla 4.13: Tabla Medias Exactitudes Clasificadores de Árboles

Una vez elegido el Fine Tree como el mejor clasificador vamos a empezar a analizar los resultados obtenidos por este tipo de clasificadores, empezando con el análisis de las tablas de métricas:

Clase	Verdaderos Positivos			Verdaderos Negativos			Falsos Positivos			Falsos Negativos		
Poca Niebla	790	952	1149	264	229	98	399	180	81	150	242	275
Media Niebla	9	8	17	1350	1402	1395	91	114	112	153	79	79
Mucha Niebla	188	185	37	976	1117	1313	126	164	207	313	137	46

Clase	Precisión			Exhaustividad			Valor F		
Poca Niebla	0.66442	0.84099	0.93415	0.84043	0.79732	0.80688	0.74213	0.81857	0.86586
Media Niebla	0.09000	0.06557	0.13178	0.05556	0.09195	0.17708	0.06870	0.07656	0.15111
Mucha Niebla	0.59873	0.53009	0.15164	0.37525	0.57453	0.44578	0.46135	0.55142	0.22630

(a) Métricas Fine Tree mediante sobremuestreo en 2018.

Clase	Verdaderos Positivos			Verdaderos Negativos			Falsos Positivos			Falsos Negativos		
Poca Niebla	480	552	888	498	338	142	165	93	37	460	620	536
Media Niebla	70	35	43	984	1014	1141	457	492	366	92	62	53
Mucha Niebla	241	185	37	912	1023	1288	190	246	232	260	149	46

Clase	Precisión			Exhaustividad			Valor F		
Poca Niebla	0.74419	0.85581	0.96000	0.51064	0.47099	0.62360	0.60568	0.60759	0.75607
Media Niebla	0.13283	0.06641	0.10513	0.43210	0.36082	0.44792	0.20319	0.11218	0.17030
Mucha Niebla	0.55916	0.42923	0.13755	0.48104	0.55389	0.44578	0.51717	0.48366	0.21023

(b) Métricas Fine Tree mediante submuestreo en 2018.

Tabla 4.14: Métricas Fine Tree en 2018.

Clase	Verdaderos Positivos			Verdaderos Negativos			Falsos Positivos			Falsos Negativos		
Poca Niebla	797	925	1215	341	218	115	149	63	22	316	397	251
Media Niebla	52	22	30	1143	1254	1380	317	269	153	91	58	40
Mucha Niebla	180	134	47	1148	1212	1400	108	190	136	167	67	20

Clase	Precisión			Exhaustividad			Valor F		
Poca Niebla	0.84249	0.93623	0.98222	0.71608	0.69970	0.82879	0.77416	0.80087	0.89900
Media Niebla	0.14092	0.07560	0.16393	0.36364	0.27500	0.42857	0.20313	0.11860	0.23715
Mucha Niebla	0.62500	0.41358	0.25683	0.51873	0.66667	0.70149	0.56693	0.51048	0.37600

(a) Métricas Fine Tree mediante sobremuestreo en 2019.

Clase	Verdaderos Positivos			Verdaderos Negativos			Falsos Positivos			Falsos Negativos		
Poca Niebla	597	677	1093	436	261	123	54	20	14	516	645	373
Media Niebla	69	28	34	944	1072	1258	516	451	275	74	52	36
Mucha Niebla	206	162	51	1095	1137	1400	161	265	136	141	39	16

Clase	Precisión			Exhaustividad			Valor F		
Poca Niebla	0.91705	0.97131	0.98735	0.53639	0.51210	0.74557	0.67687	0.67063	0.84959
Media Niebla	0.11795	0.05846	0.11003	0.48252	0.35000	0.48571	0.18956	0.10018	0.17942
Mucha Niebla	0.56131	0.37939	0.27273	0.59366	0.80597	0.76119	0.57703	0.51592	0.40157

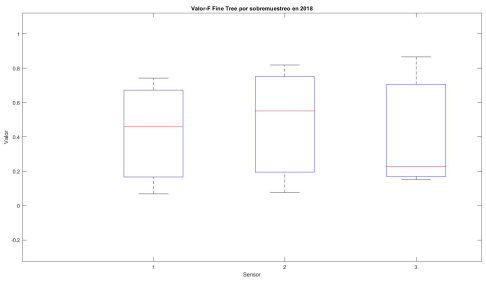
(b) Métricas Fine Tree mediante submuestreo en 2019.

Tabla 4.15: Métricas Fine Tree en 2019.

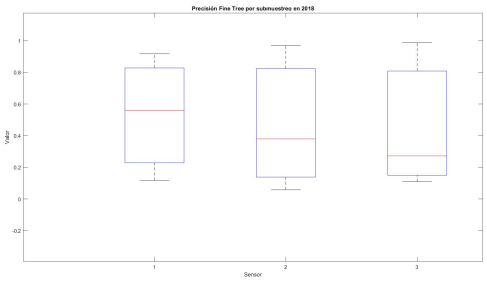
Al analizar estas tablas, podemos destacar algunas observaciones. Por ejemplo, al comparar la Tabla 4.14a, podemos notar cómo es superior al resto en cuanto al número de verdaderos positivos y negativos y falsos positivos, especialmente para los eventos de poca y mucha niebla. Por el contrario, al analizar la Tabla 4.14b se observa que en submuestro posee los mejores verdaderos positivos y negativos para el evento de media niebla. Además de esto, si comparamos las Tablas 4.14a y 4.14b con sus equivalentes de los otros clasificadores, se observa como el Fine Tree destaca por ser el que menores falsos positivos y negativos presenta, tanto para sobremuestreo como submuestreo. Por último, al comparar estas mismas tablas, se puede advertir como el Fine Tree es el que mejor se adapta a la reducción en el número de muestras, destacando los valores de sus métricas de precisión, exhaustividad y valor-F sobre los de los otros clasificadores; de forma especial para los puntos dos y tres donde se reduce el número de muestras.

En cuanto al resto de clasificadores, después del Fine Tree destaca el Medium Tree como el siguiente mejor clasificador, especialmente para sobremuestreo, donde sobresale en la gran mayoría de medidas, tanto en verdaderos positivos y negativos como en falsos positivos y negativos y precisión, exhaustividad y valor-F. En cuanto a submuestreo, en estas medidas la diferencia se reduce, sobresaliendo el Coarse Tree en algunas medidas. Sin embargo, el Medium en los valores de precisión, exhaustividad y valor-F, acaba destacando en la mayoría de las medidas, siendo por tanto el Coarse Tree el peor de los de este tipo.

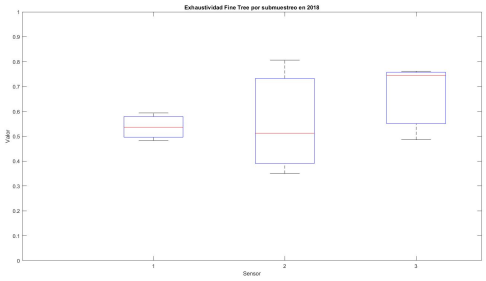
En cuanto a los resultados obtenidos mediante submuestreo, aunque no destaca tan claramente un clasificador sobre el resto como para sobremuestreo, el Medium Tree acaba destacando en la mayoría de las métricas de precisión, exhaustividad y valor-F. Por lo tanto, el Coarse Tree es el peor de los de este tipo de clasificadores. Una vez analizadas las tablas de las métricas, pasamos a estudiar como estas se representan en forma de diagramas de cajas.



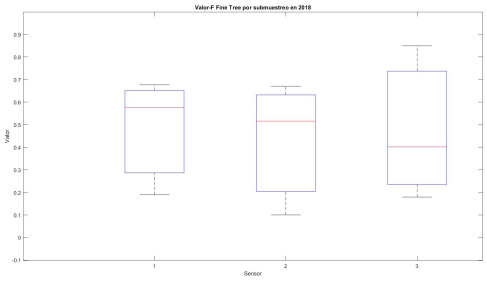
(c) Valor-F para Sobremuestreo en 2018



(d) Precisión para Submuestreo en 2018

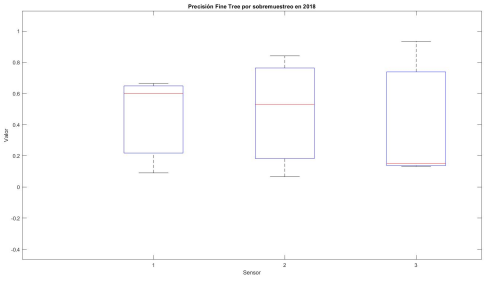


(e) Exhaustividad para Submuestreo en 2018

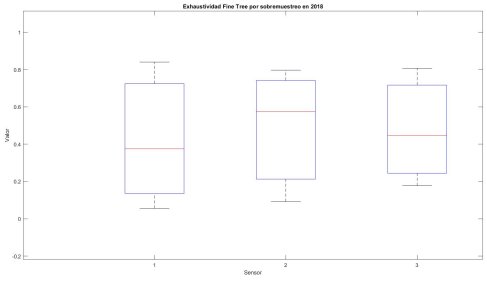


(f) Valor-F para Submuestreo en 2018

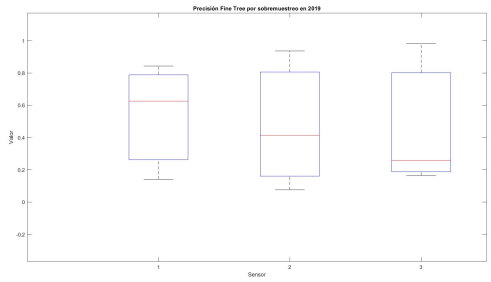
Figura 4.22: Métricas Fine Tree en 2018



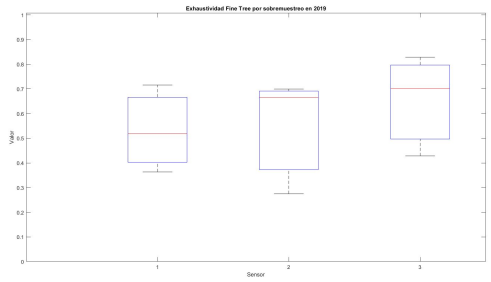
(a) Precisión para Sobremuestreo en 2018



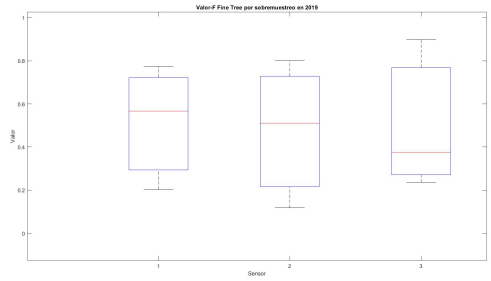
(b) Exhaustividad para Sobremuestreo en 2018



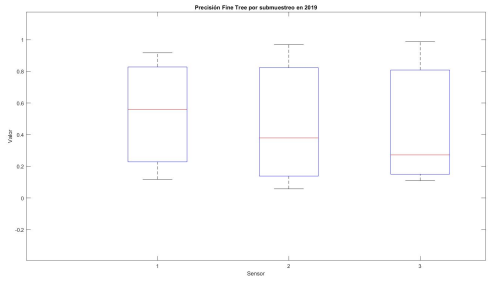
(a) Precisión para Sobremuestreo en 2019



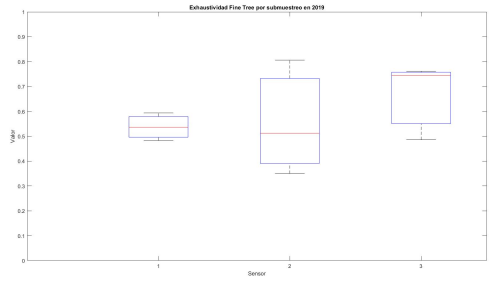
(b) Exhaustividad para Sobremuestreo en 2019



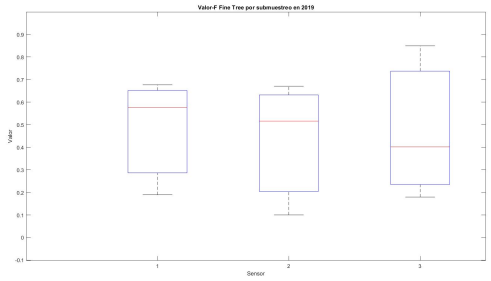
(c) Valor-F para Sobremuestreo en 2019



(d) Precisión para Submuestreo en 2019



(e) Exhaustividad para Submuestreo en 2019

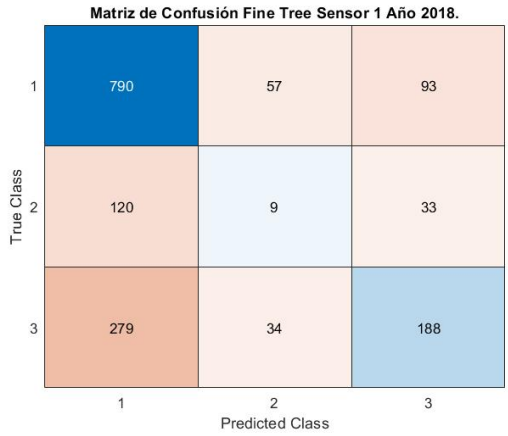


(f) Valor-F para Submuestreo en 2019

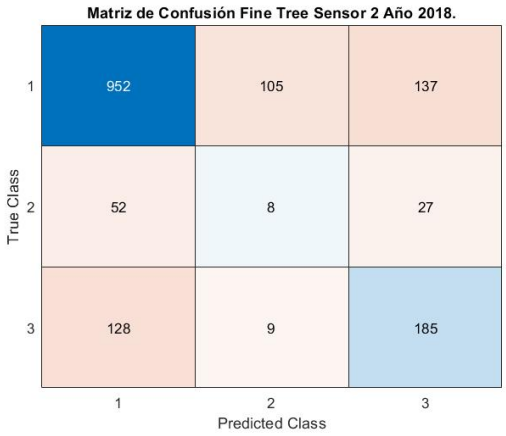
Figura 4.23: Métricas Fine Tree en 2019

Una vez estudiadas estos diagramas, podemos confirmar de forma visual los resultados obtenidos en las tablas. Por ejemplo, si comparamos las imágenes para sobremuestreo en 2018, mediante las Figuras 4.22a, 4.22b y 4.22c, podemos observar como el Fine Tree se confirma como el mejor clasificador de este tipo, presentando una medianas superiores a las del resto de clasificadores para sobremuestreo. Después del Fine Tree sobresale el Medium Tree,que comparándolo con el Fine Tree, se muestra que para submuestreo está más o menos al mismo nivel, sobresaliendo incluso en algunas medidas. Finalmente, se confirma el Coarse Tree como el que presenta las medias más pobres.

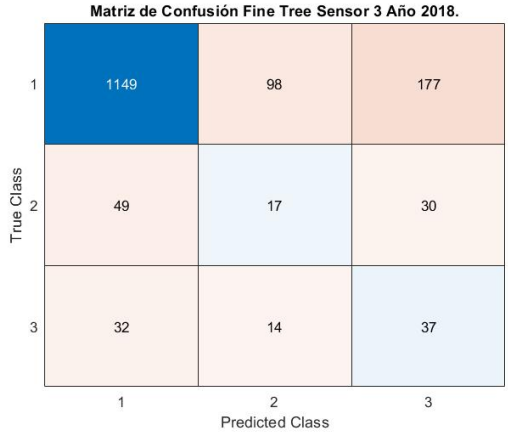
Una vez terminamos de analizar las métricas, tanto sus valores numéricos como sus diagramas, vamos a proceder con las matrices de confusión de estos clasificadores.



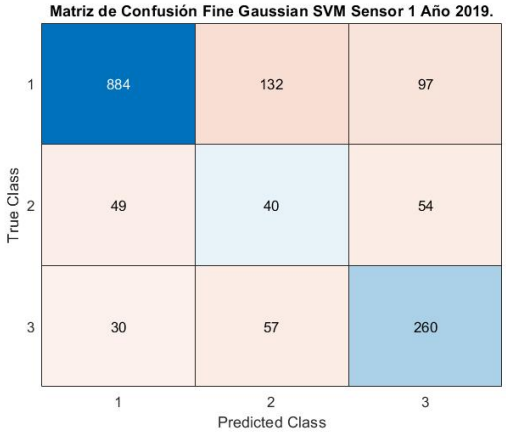
(a) Punto 1 en 2018



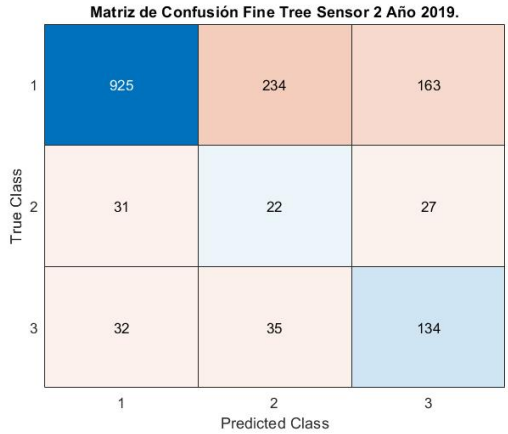
(b) Punto 2 en 2018



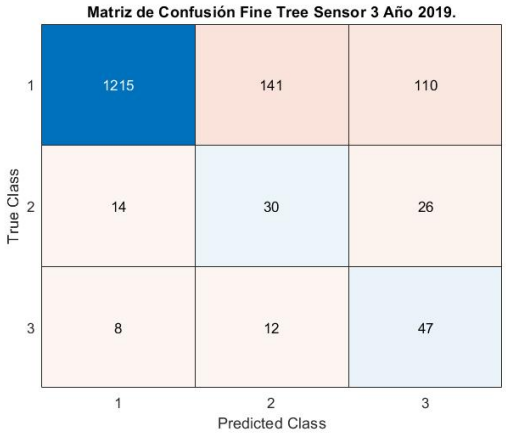
(c) Punto 3 en 2018



(d) Punto 1 en 2019

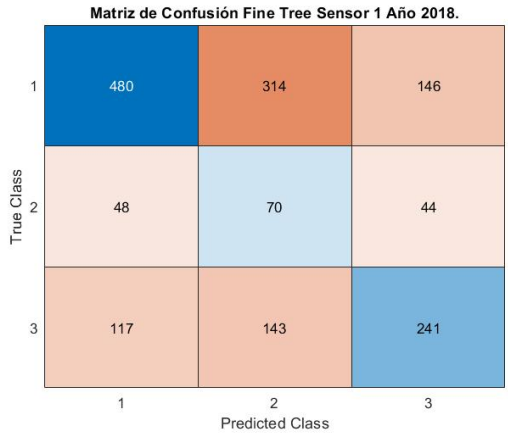


(e) Punto 2 en 2019

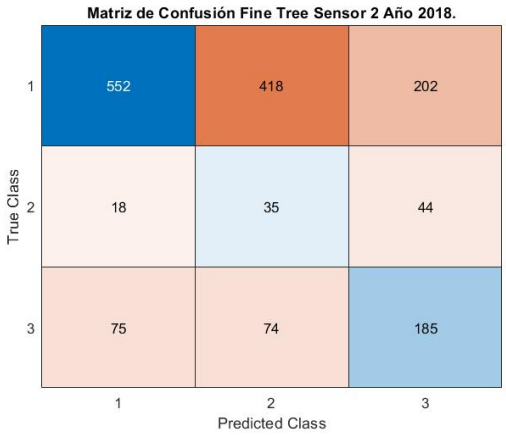


(f) Punto 3 en 2019

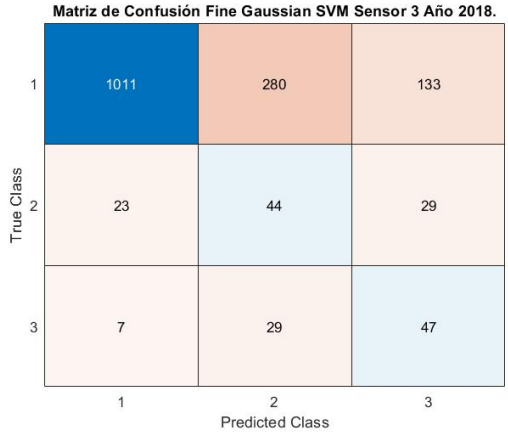
Figura 4.24: Matrices de Confusión del Fine Tree mediante Sobremuestreo



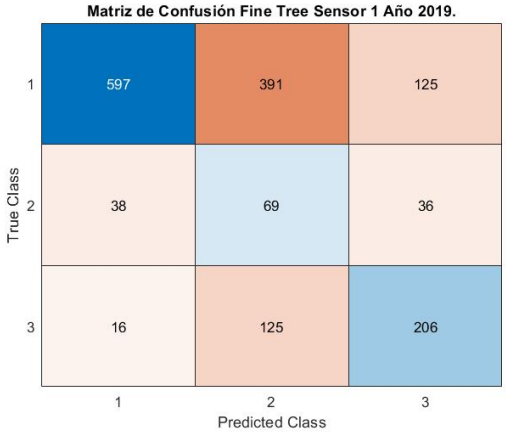
(a) Punto 1 en 2018



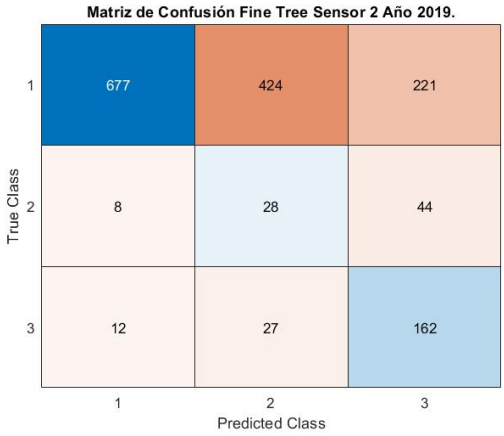
(b) Punto 2 en 2018



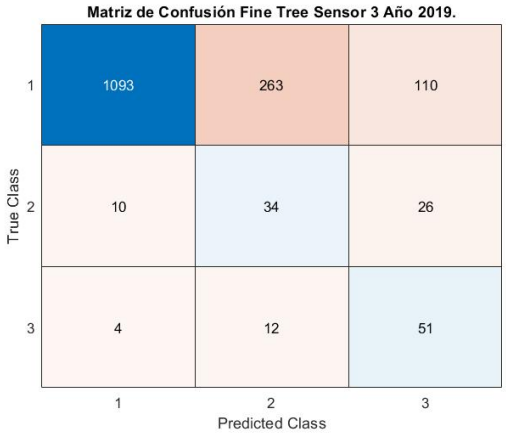
(c) Punto 3 en 2018



(d) Punto 1 en 2019



(e) Punto 2 en 2019



(f) Punto 3 en 2019

Figura 4.25: Matrices de Confusión del Fine Tree mediante Submuestreo

Al analizar estas matrices de confusión podemos destacar una serie de observaciones. En primer lugar, el Fine Tree destaca especialmente por mostrar mejores resultados que Coarse Tree, y aunque está más igualado con los obtenidos por Medium Tree, al tener éste menos fallos para poca niebla, el Fine Tree destaca especialmente en sobremuestreo por tener más aciertos

para poca niebla e igualarle en media niebla. En cuanto a submuestreo, como se puede ver en la Figura 4.25a, podemos apreciar como destaca el Fine Tree sobre el Medium Tree, en el punto uno por mostrar unos menores fallos de predicción en la mayoría de medidas, así como superarle en aciertos en media niebla, además de superar al clasificador Coarse Tree en la mayoría de fallos. A pesar de no destacar especialmente en el punto uno, tanto para sobremuestreo como para submuestreo, es el que muestra mejores resultados de media, mostrando ser el que mayor resistencia posee en los puntos dos y tres, como se puede apreciar en las Figuras 4.24c y 4.25c respectivamente.

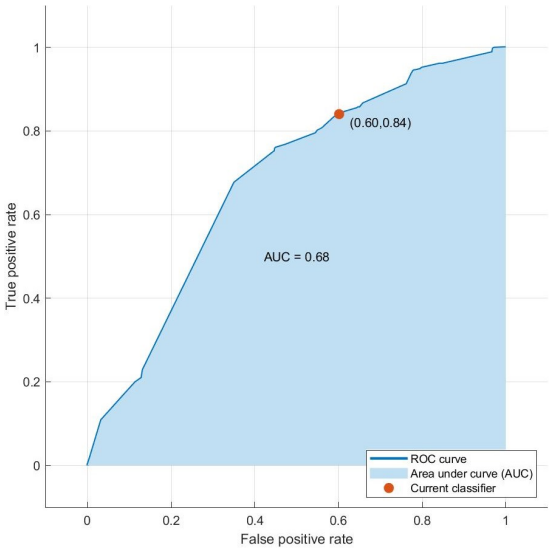
En cuanto a los clasificadores Medium y Coarse Tree, el Medium Tree se muestra superior al Coarse Tree, tanto en mejores valores de verdaderos positivos y negativos, como menores de falsos positivos y negativos, en sobremuestreo y submuestreo. Se muestra superior además, en los puntos dos y tres, con mayor resistencia a tener menor número de muestras. Sin embargo, cabe destacar que el Coarse Tree muestra la capacidad de poseer un mayor número de aciertos en media niebla que el medium tree, aunque a la vez muestra un mayor número de falsos positivos en esta clase.

Finalmente, se puede puede observar como se confunde también en estos clasificadores un evento con el más numeroso de los otros dos restantes, como se puede apreciar en la mayoría de matrices de confusión, como se ve en la Figura 4.24. Cabe resaltar que en algunas matrices de confusión de 2019, como en la Figura 4.24f, se confunde más el evento de poca niebla con el de media niebla. Esto puede ser debido a que hay menor número de eventos de baja visibilidad, lo que puede provocar que en 2019, al haber un número de eventos de mucha niebla más próximo al de media niebla, y pueda confundirse más veces los eventos de poca niebla con los de media niebla, al tener características meteorológicas más parecidas. Además de esto, se puede apreciar como al pasar del punto uno al dos y tres se reduce la eficiencia al predecir los eventos media y poca niebla, y aumenta al predecir poca niebla, como se puede apreciar por ejemplo al examinar la Figura 4.24a frente a la 4.24c. Este hecho es debido a la reducción en el número de muestras de estos sucesos en los sensores dos y tres, y al aumento del número de sucesos de poca niebla. Asimismo, si estudiamos las matrices de sobremuestreo de la Figura 4.24, frente a las de submuestreo de la 4.25, se puede apreciar como aumentan de forma considerable el número de errores al intentar predecir los eventos de poca niebla; esto es debido al menor número de muestras de entrenamiento de estos sucesos. Además de esto, hay que destacar el aumento en el número de aciertos para los eventos media y mucha niebla, esto podría tener relación con que para submuestreo, los eventos media y mucha niebla únicamente poseen muestras reales y no sintéticas, como para sobremuestreo, siendo más precisas sus predicciones.

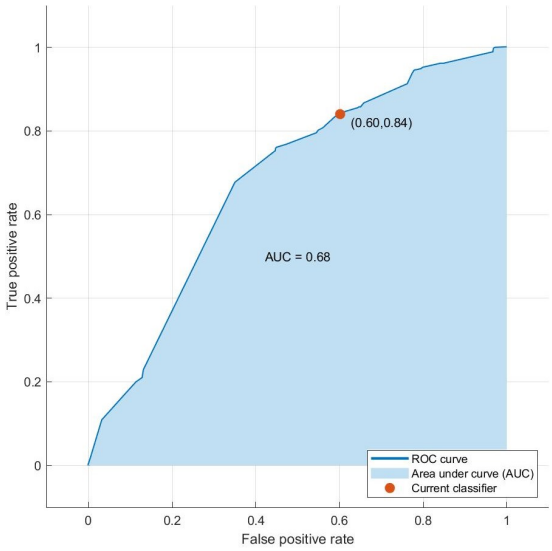
Por último, terminamos el análisis de estos clasificadores con las gráficas ROC y las AUC. Como se puede observar en estas gráficas, al comparar la Figura 4.26a con la 4.26e, el AUC del evento mucha niebla es superior, para la mayoría de casos, al de poca niebla, reforzando la hipótesis de que el AUC está relacionado con el número de muestras; siendo menor para poca niebla porque a pesar de predecir mejor los eventos de poca niebla, al tener muchas más muestras, comete más

errores y esto puede provocar tener una menor AUC. También se confirma que el AUC para el evento media niebla es el inferior de los tres debido a que es el que menos muestras posee, y esto pesa más en este caso que tener mayor número de muestras para fallar. Además, se produce un aumento en las AUC en submuestreo con respecto a los que poseían en sobremuestreo. Este hecho vendría a respaldar que, al tener menor número de muestras en submuestreo con respecto a sobremuestreo y, debido a que no hay muestras sintéticas, serían más precisas las predicciones. En esta línea destaca que en el punto tres frente al uno, como se aprecia en la Figura 4.26, los resultados se mantienen estables, y de hecho en algunos casos mejoran los AUC en los puntos dos y tres, especialmente en submuestreo, a pesar de tener muchas muestras menos; esto es debido a la relación entre el AUC y el número de muestras.

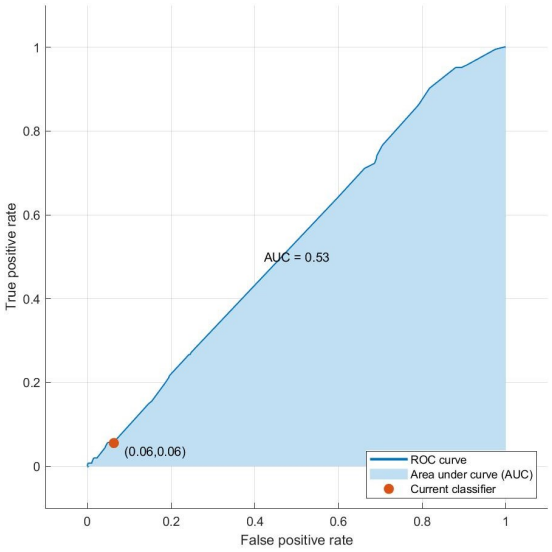
En cuanto a las gráficas entre clasificadores, a partir de éstas se confirma que el Fine tree muestra los mejores resultados, aunque no destaque mucho en el punto uno frente al resto. Especialmente en submuestreo, muestra su superioridad frente a los otros clasificadores en el punto dos y tres. A partir de estas gráficas también podemos confirmar como el Medium Tree posee los mejores resultados después del Fine Tree, mostrando mejores resultados que el Coarse Tree en la gran mayoría de mediciones, en todos los puntos, tanto en sobremuestreo como submuestreo.



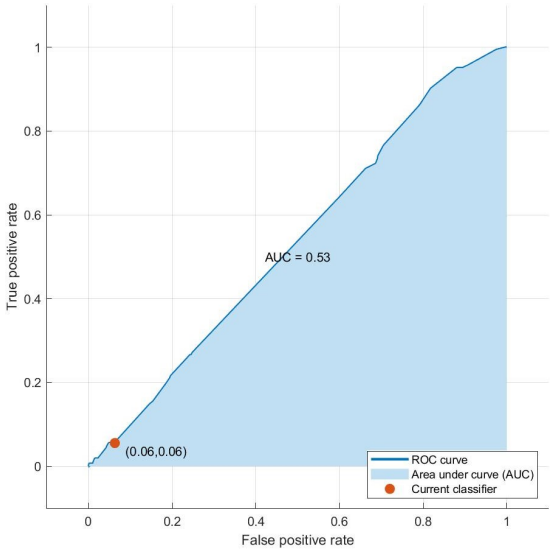
(a) Clase Positiva Evento Poca Niebla en el Punto 1.



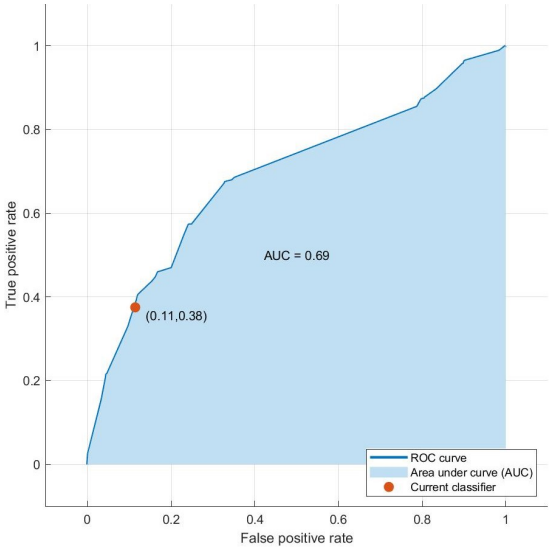
(b) Clase Positiva Evento Poca Niebla en el Punto 3.



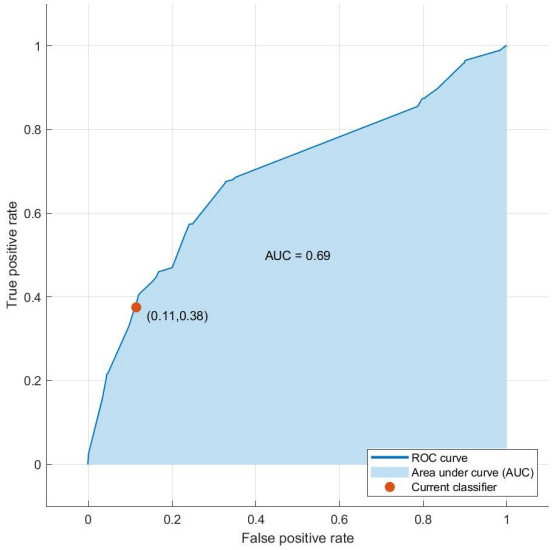
(c) Clase Positiva Evento Media Niebla en el Punto 1.



(d) Clase Positiva Evento Media Niebla en el Punto 3.



(e) Clase Positiva Evento Mucha Niebla en el Punto 1.



(f) Clase Positiva Evento Mucha Niebla en el Punto 3.

Figura 4.26: ROC y AUC de Fine Tree en el punto 1 y 3 por sobremuestreo durante 2018.

A partir de todos los resultados analizados en este apartado, podemos destacar como el mejor clasificador al Fine Tree, con más aciertos para los puntos dos y tres para media y mucha niebla, poseyendo por tanto la mejor exactitud general de estos clasificadores, a pesar de encontrarse seguido muy de cerca por el Medium Tree. Por último, podemos encontrar al Coarse Tree, como el clasificador con los resultados más pobres de este grupo.

4.2.6. Análisis Final de los cinco mejores clasificadores

Finalmente, vamos a terminar analizando el mejor clasificador obtenido y estudiar cuáles son los mejores clasificadores de todos los diseñados en el estudio. Para poder compararlos entre sí, vamos a utilizar como criterio selector las medias de las exactitudes de las predicciones hechas por cada clasificador, como podemos ver en la Tabla 4.16:

Medias Exactitud	Bagged Trees Ensemble	Fine KNN	Wide NN	Fine Gaussian SVM	Fine Tree
Sobremuestreo	**82.46666**	69.55	71.23888	73.76666	65.5
Submuestreo	**73.23333**	63.6	63.23333	61.63333	56.65

Tabla 4.16: Tabla Medias Exactitudes Mejores Clasificadores

Como se puede apreciar en la Tabla 4.16, el clasificador de conjunto Bagged Trees es el que presenta unas mejores exactitudes de los cinco, tanto para sobremuestro como submuestreo. Se trata sin duda del mejor, ya que es el único de todos que ha ganado, en prácticamente todos los parámetros de las métricas de estudio, al resto de clasificadores de su tipo, tanto para sobremuestreo como submuestreo; obteniendo mayores predicciones acertadas de todos los eventos, y los más altos valores de las métricas de exactitud, precisión, exhaustividad y valor-F. Además destaca como el clasificador que mejor soporta pasar del sensor uno al dos y tres, mostrando los mejores resultados en este último sensor.

Después del Bagged Trees podemos encontrar al Wide NN y al Fine Gaussian como los siguientes mejores clasificadores. Estos dos muestran, como podemos ver en la Tabla 4.16, unas medias de exactitud muy parecidas al hacer la media entre la de sobremuestreo y submuestreo. Se convierten por ello en los siguientes mejores clasificadores, superando ligeramente al Fine KNN. El Wide NN y el Fine Gaussian SVM se muestran superiores al Fine KNN especialmente en Sobremuestreo. A pesar de que en el sensor uno el Fine KNN se muestre mejor en la mayoría de las mediciones siendo el que presenta mayores aciertos y menores fallos en la mayoría de las clases, el Wide NN y el Fine Gaussian le superan al pasar al sensor dos y tres, donde se dispone de menor número de muestras de los eventos media niebla y mucha niebla, teniendo mayores aciertos y menores fallos en la mayoría de clases. Por lo tanto, estos dos clasificadores demuestran tener una mayor resistencia ante la escasez de muestras. Este hecho se justifica al mostrarse especialmente superiores al Fine KNN en las clases media niebla y mucha niebla, las

dos con menor número de muestras. En cuanto a submuestreo, como se ve en la Tabla 4.16, los resultados de estas tres se encuentran ajustados entre ellos, no destacando especialmente ninguno de los tres, y por lo tanto, en general entre sobremuestreo y submuestreo, destacan el Wide NN y el Fine Gaussian SVM.
Finalmente, aunque ambos no destacan ampliamente entre las demás redes neuronales, siendo superados en algunas métricas de estudio, especialmente para los verdaderos positivos, destacan por ser los que menores errores cometen, especialmente como hemos visto al compararlos con el Fine KNN, para los eventos de media y mucha niebla, además de ser de los que mejor se comportan al ir cambiando del sensor uno al dos y al tres. Sin embargo, como vemos al comparar la exactitud de sobremuestreo con la de submuestreo, ambos sufren una caída en la eficiencia de sus predicciones en submuestreo.

Después del Wide NN y el Fine Gaussian SVM, podemos destacar al clasificador Fine KNN. Este muestra unas medidas de exactitudes ligeramente inferiores a las del Wide NN y el Fine Gaussian SVM. A pesar de no destacar tanto como el Bagged Trees, se trataba igualmente del mejor clasificador KNN, con mayores predicciones acertadas en los tres tipos de eventos, valores de exactitud, precisión, exhaustividad y valor-F, y menores fallos, tanto para sobremuestreo como submuestreo. También destaca al igual que Bagged Trees, por ser el mejor prediciendo los eventos de media y mucha niebla, y ser el mejor en los sensores dos y tres ante la escasez de muestras. Sin embargo, como acabamos de ver, a pesar de ser el que mejor predice los eventos media niebla y mucha niebla, y resiste la escasez de muestras para los clasificadores KNN, se tratan de sus puntos débiles, siendo superado por el Bagged Trees, Wide NN y el Fine Gaussian SVM debido a ello.

Por último, podemos destacar al clasificador Fine Tree. Se trata claramente del peor clasificador del grupo, teniendo los peores valores de exactitud tanto para sobremuestreo como para submuestreo, como podemos ver en la Tabla 4.16. Sin embargo, no solamente es el que muestra unas exactitudes más bajas, sino que como se ha podido ir notando al analizar los clasificadores de su grupo, se ha tratado claramente del clasificador que menos ha sobresalido entre los de su clase. Este clasificador es superado en muchos valores de los resultados, especialmente por el Medium Tree, aunque también superando a estos en otros como ya se ha mencionado en su análisis, y que justificaban que fuera el mejor de ellos.

Capítulo 5

Conclusiones y Líneas Futuras

5.1. Conclusiones

El objetivo del presente trabajo era diseñar una serie de clasificadores con los que predecir los eventos de niebla en el área de Mondoñedo, y con ello su visibilidad, a partir de una base de datos desbalanceada de variables meteorológicas. A partir del análisis de los resultados obtenidos en el desarrollo de este proyecto hemos podido llegar a una serie de conclusiones.

Primero de todo, a partir de las diferencias generales apreciadas en todos los clasificadores, entre los resultados obtenidos por el sensor uno con respecto a los de de los sensores dos y tres, podemos llegar a la conclusión de que la eficiencia de los clasificadores está relacionada con el número de ocurrencias de los eventos. A partir de estas diferencias, podemos deducir que los sensores dos y tres deben de encontrarse más alejados de la zona de más ocurrencia de la niebla de Mondoñedo. En relación a esto hemos podido observar igualmente como se han mostrado más eficientes las predicciones del evento poca niebla que de mucha y especialmente de media niebla, al ser el evento poca niebla el que menos muestras posee.

Asimismo, a partir de los principales errores observados en las predicciones podemos afirmar que los principales errores de predicción se cometen al confundir una evento niebla con el que evento con más muestras de los dos restantes. Por ejemplo, el evento poca niebla con el mucha niebla, y el mucha y media niebla con el poca niebla.

Además de esto, a partir de las diferencias observadas entre los resultados obtenidos en sobremuestreo con respecto a los de submuestreo, podemos afirmar que el método de balanceo de sobremuestreo ha demostrado ser capaz de obtener unas mejores predicciones que aquellas obtenidas por medio de submuestreo.

Por último, de todos los clasificadores diseñados examinados, podemos destacar el clasificador Bagged Trees, el cual ha demostrado ser el mejor para predecir los tres tipos de eventos, tanto para sobremuestreo como submuestro, además de ser el que mejor robustez presenta para predecir eventos con escasez de muestras. Después de éste, encontramos al Wide NN y al Fine SVM que han mostrado unos resultados similares, no destacando especialmente por tener el mayor

número de aciertos, pero sí sobresaliendo por el menor número de fallos, y siendo los mejores en predecir los eventos media y mucha niebla y en adaptarse al menor número de muestras. Aunque como hemos observado, su eficiencia desciende considerablemente ante submuestreo. Después de estos clasificadores, encontramos al Fine KNN, quién a pesar de destacar en los mismos aspectos que el primero no alcanza sus resultados. Además, presenta una menor resistencia a la escasez de muestras que el Wide NN y el Fine Gaussian SVM. Por último, encontramos el Fine Tree, el cual ha mostrado ser el que obtiene unos resultados más pobres de este grupo, no destacando especialmente entre ellos.

5.2. Líneas futuras

En el presente estudio hemos avanzado en el desarrollo de clasificadores capaces de predecir eventos a partir de unas bases de datos, así como examinar y conocer los puntos fuertes y débiles de cada tipo de clasificador.

En cuanto a posibles mejoras y líneas de estudio futuras, sería interesante poder estudiar qué resultados obtendrían nuestros clasificadores, si les pusiéramos a prueba con las bases de datos de los otros sensores y analizar como se adaptan a las bases de datos recogidas en otros puntos. Por ejemplo los clasificadores entrenados con los datos de entrenamiento del sensor uno, evaluarlo con los datos de prueba del dos y tres.

Asimismo, como hemos observado, el principal problema que han enfrentado nuestros clasificadores ha sido la diferencia en el número de ocurrencias de los distintos eventos, por ello se podría estudiar qué resultados se obtendrían realizando el balanceo antes de dividir la base de datos para entrenamiento y prueba; teniendo por lo tanto ambos subconjuntos balanceados después. Por último, como pudimos ver en el capítulo del estado del arte, en el capítulo 2, en el estudio [15] pudimos observar cómo para regresión no era necesario aplicar técnicas de balanceo, como si lo era para clasificación. Es por ello que podría resultar de interés probar nuestras bases de datos en algunos modelos de regresión a los que no les afecta la diferencia en el número de muestras de los diferentes eventos, y comparar resultados con los modelos obtenidos por clasificación.

Bibliografía

[1] PRANJAL PANDE. Fog and planes: How low visibility can impact operations. JAN 17, 2021.

[2] STEVE ARBOGAST. Aviation weather issues – fog – part 2: Impact of fog. APRIL 5, 2016.

[3] Oliver J. Muldoon. Relating the distributional character of numerical model output parameters to the occurence of fog over the north atlantic ocean. 1986.

[4] David W. Aha and Richard L. Bankert. Feature selection for case-based classification of cloud types: An empirical comparison. 1994.

[5] Lino R. Naranjo-Díaz and Arnaldo Alfonso. Fog forecasting in cuba. neural networks versus discriminant analysis. 1995.

[6] H. D. Navone P. F. Verdes, P. M. Granitto and H. A. Ceccatto. Frost prediction with machine learning techniques. 2000.

[7] Gaetano Zazzaro. An index for local fog forecast by applying data mining techniques. 2008.

[8] Sankar Nath y A.K. Sharma. A.K. Mitra. Fog forecasting using rule-based fuzzy inference system. 2008.

[9] Alain Protat Dominique Bouniol Neda Boyouk Jean-Charles Dupont, Martial Haeffelin and Yohann Morille. Stratus–fog formation and dissipation: A 6-day case study. 2012.

[10] Rajesh Kumar. Decision tree for the weather forecasting. 2013.

[11] Juraj Bartok1 Ivana Bartoková1, Andreas Bott2 and Martin Gera. Fog prediction for road traffic safety in a coastal desert region: Improvement of nowcasting skills by the machine-learning approach. 2015.

[12] 3 J. Sanz-Justo2 E. Cerro-Prada3 L. Cornejo-Bueno1, C. Casanova-Mateo2 and S. Salcedo-Sanz. Efficient prediction of low-visibility events at airports using machine-learning regression. 2017.

[13] C. Casanova-Mateo J. Sanz-Justob S. Salcedo-Sanzd C. Hervás-Martínez. D. Guijo-Rubioa, P.A. Gutiérrez. Prediction of low-visibility events due to fog using ordinal classification. 2018.

[14] Ye-qing Yao Hui Lua Peng Chenc-Bing Wang Kai-chao Mia, Ting-ting Hana and Jun Zhang. Application of lstm for short term fog forecasting based on meteorological elements. 2020.

[15] C. Casanova-Mateo S. Ghimire E. Cerro-Prada P.A. Gutierrez-R.C. Deo S. Salcedo-Sanz. C. Castillo-Boton, D. Casillas-Pérez. Machine learning regression and classification methods for fog events prediction. 2022.

[16] Wesley Chai. A timeline of machine learning history. 20 Oct 2020.

[17] Sara Brown. Machine learning, explained. Apr 21, 2021.

[18] Great Learning Team. Types of neural networks and definition of neural network. Sep 25, 2021.

[19] Aleksander Obuchowski. Understanding neural networks 2: The math of neural networks in 3 equations. Apr 16, 2020.

[20] Jason Brownlee. A gentle introduction to ensemble learning algorithms. April 27, 2021.

[21] Jason Brownlee. Strong learners vs. weak learners in ensemble learning. May 3, 2021.

[22] Jason Brownlee. How to develop and evaluate naive classifier strategies using probability. September 25, 2019.

[23] Maciej Balawejder. Loss functions in machine learning. Feb 24, 2022.

[24] Decision tree classification algorithm.

[25] Nagesh Singh Chauhan. Decision tree algorithm, explained. february 9,2022.

[26] K-nearest neighbor(knn) algorithm for machine learning.

[27] Ashwin Pandey. The math behind knn. exploring the metric functions used in k-nearest neighbors (knn) model. january 7,2021.

[28] Support vector machine algorithm.

[29] Anshul Saini. Support vector machine (svm): A complete guide for beginners. October 12,2021.

[30] Peter Lu Blanca Li and v chmccl. Smote. 27/04/2022.

[31] O*NET OnLine. What is undersampling?. April 2022.

[32] Sarang Narkhede. Understanding auc - roc curve. Jun 26, 2018.

[33] Paul M.Tag and James E.Peak. Machine learning of maritime fog forecast rules. *Applied Meteorology*, 35:714–724, 1995.

[34] Rashida048. Simple explanation on how decision tree algorithm makes decisions. Apr 7, 2027.

[35] P. Chitra. S. Abirami. The digital twin paradigm for smarter systems and environments: The industry use cases. 2020.

[36] Boaz Shmueli. Multi-class metrics made simple, part i: Precision and recall. Jul 2, 2019.

[37] Jose Martinez Heras. Precision, recall, f1, accuracy en clasificación. 09/10/2020.

Anexos

Debido al elevado número de figuras y tablas obtenidas como resultados de nuestros clasificadores, en la memoria solamente se recogen las del mejor clasificador de cada tipo. Se pueden encontrar las de todos los clasificadores en los siguientes enlaces a los anexos disponibles en GitHub. Aquí podemos encontrar los anexos de las tablas de métricas, diagramas de cajas de las métricas, matrices de confusión de los años 2018 y 2019 y los de las gráficas ROC y AUC mediante sobremuestreo y submuestreo en los años 2018 y 2019:

Anexo Tablas

Anexo Imágenes de las Métricas

Anexo Imágenes de las Matrices de Confusión en 2018

Anexo Imágenes de las Matrices de Confusión en 2019

Anexo Gráficas ROC y los AUC mediante sobremuestreo en 2018

Anexo Gráficas ROC y los AUC mediante sobremuestreo en 2019

Anexo Gráficas ROC y los AUC mediante submuestreo en 2018

Anexo Gráficas ROC y los AUC mediante submuestreo en 2019