**MODELADO**

Después de la preparación de datos, tenemos que desarrollar el modelado.

Un modelo tiene como objetivo predecir una variable objetivo a partir de un conjunto de variables explicativas y lo vamos a desarrollar mediante técnicas de Machine Learning siguiendo los siguientes pasos:

1. EDA
2. División de nuestro dataset
3. Modelos
4. Mejora de modelos
5. Selección del modelo

Para esta parte del trabajo, vamos a utilizar lenguaje Python a través de Júpiter Notebook.

1. **EDA - Análisis exploratorio de datos**

Partimos de nuestra base de datos limpia de R. Un dataset con un total de 12976 líneas y 29 columnas

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Las columnas son variables que debemos saber qué tipo de datos tienen y cómo se distribuyen; esto lo haremos viendo el tipo de datos, su histograma y el número de observaciones únicas de cada variable.

Tabla

Descripción generada automáticamente con confianza media

Gráfico de cajas y bigotes

Descripción generada automáticamente

Tabla

Descripción generada automáticamente con confianza mediaInterfaz de usuario gráfica, Tabla

Descripción generada automáticamente con confianza media

A simple vista, lo que vemos, es que las variables Unnamed e id tienen tantos valores únicos como observaciones contienen, por lo que decidimos eliminar ambas variables.

Por otro lado, las variables se dividen entre variables numéricas y variables categóricas. En Python no podemos trabajar con variables categóricas, por lo que tendremos que hacer un tratamiento especial con ellas y por ello, vamos a dividir el dataset en dos:

Texto

Descripción generada automáticamente con confianza bajaTexto

Descripción generada automáticamente

En primer lugar, eliminaremos aquellas que tengan demasiadas categorías, como Idmunicipio y fecha, que tienen un total de 268 y 2272 valores únicos respectivamente. Al tener tantas categorías y estar representada de una manera menos fraccionada en idprovincia para el caso de idmunicipio y Mes, Trimestre y Año para fecha, decidimos prescindir de ellas.

Las formas más comunes de tratar este tipo de variables suelen ser transformarlas a variables ordinales o realizar la codificación One-Hot.

Transformarlas en orden numérico en ocasiones dificulta la predicción ya que da diferentes pesos a las distintas categorías de una variable, por ello, esta transformación la vamos a utilizar para aquellas variables que sí sigan un orden en sus categorías; sin embargo, para aquellas que no sigan un orden, utilizaremos la codificación One-Hot. Este método consiste en crear una nueva variable binaria por cada categoría existente en la variable inicial, donde 1 serán las observaciones que pertenezcan a esa categoría y 0 las demás.

Para saber qué variable será transformada con cada método, vemos qué categorías son las que componen nuestras variables.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamenteImagen que contiene Texto

Descripción generada automáticamente

Tabla

Descripción generada automáticamente con confianza media

Texto

Descripción generada automáticamente con confianza baja

* **variables con codificación One-Hot**: idprovincia, Trimestre, Mes, DIR\_VIENTO
* **variables orden numérico**: gastos, ALTITUD, VELMEDIA

Tras realizar estas transformaciones y previa comprobación de que no se han traspasado los datos con ningún valor faltante, nuestro dataset tiene un total de 12976 líneas y 45 columnas y está listo para comenzar con el modelo.

1. **DIVISIÓN DE NUESTRO DATASET**

En primer lugar, antes de empezar a generar nuestros modelos, debemos definir nuestra variable objetivo, que será causa. El resto de variables serán nuestras variables explicativas.

Recordemos cómo se distribuye nuestra variable, ya que lo deberemos tener en cuenta a la hora de seleccionar el tipo de modelo que queremos realizar.

Escala de tiempo

Descripción generada automáticamente con confianza media

La variable causa se compone de cinco categorías, las cuales están desbalanceadas y debere- mos tenerlo en cuenta.

En segundo lugar, tenemos que dividir de manera aleatoria nuestro dataset en dos, una parte que usaremos para entrenar nuestro modelo que corresponderá con el 80% a la que llamaremos Train y otra para evaluarlo, con el 20% restante que será Test. En esta parte es en la primera en la que deberemos tener en cuenta el desbalanceo de nuestra variable objetivo, distribuyendo de manera proporcional las categorías.

Texto

Descripción generada automáticamente

Una vez realizados estos dos pasos, ya podemos comenzar a generar nuestros modelos.

1. **MODELOS**

Para los modelos vamos a utilizar métodos Backward (Eliminación hacia atrás), partimos de modelos iniciales muy complejos que incluyen todas las variables y después, vamos reduciéndolas.

Al querer predecir una variable multicategoría, utilizaremos los modelos que consideramos pueden ser más idóneos para ella.

* Support Vector Machines = SVM : Consiste en un conjunto de métodos de aprendizaje supervisados que realiza un método one-against-one . Utiliza una técnica de iguala los dintintos tipos de categorías, por lo tanto, es una buena opción para predecir una variable categórica desbalanceada.
* Árbol de decisión: Combinación y subdivisión en ramas de las variables de una forma binaria para la toma de la decisión con mayor probabilidad de que ocurra un suceso
* Random Forest: Conjunto de árboles de decisión
* Gradient Boosting: Está formado por un conjunto de árboles de decisión individuales, entrenados de forma secuencial, de forma que cada nuevo árbol trata de mejorar los errores de los árboles anteriores. La diferencia con Random Forest es que utiliza árboles más débiles, con menos profundidad.

Estos modelos no serán normalizados ya que hemos comprobado que no afecta al resultado final.

Las métricas de rendimiento que vamos a utilizar es el Acurracy, utilizando una escala del 0 al

1, en la que 0 indica una capacidad nula de predicción y 1 que el modelo tendría la capacidad

de predecir correctamente el 100% de las causas de los incendios.

Los resultados de los modelos más complejos con todas nuestras variables iniciales, son los siguientes:

'MODELO 1 :Score modelo SVM : 0.8701848998459168 con todas las variables: 44'

'MODELO 2 : Score modelo Árbol de decisión : 0.7619414483821263 con todas las variables: 44'

'MODELO 3 : Score modelo Random Forest : 0.8701848998459168 con todas las variables: 44'

'MODELO 4 : Score modelo Gradient Boosting : 0.8701848998459168 con todas las variables : 44 '

Para empezar, tenemos unos resultados muy buenos, parece que los modelos se ajustan muy

bien; ahora, nuestro siguiente paso, es reducir las variables para poder conseguir un modelo más sencillo y un mejor rendimiento.

Al ser el Árbol de decisión el modelo que peores resultados nos da en una primera valoración

vamos a prescindir de él.

1. **MEJORA DE MODELOS**

Para mejorar nuestro modelo vamos a realizar los siguientes ajustes:

* Selección de Parámetros
* Selección de variables

**4.1 SELECCIÓN DE PARÁMETROS**

Para ello, hemos utilizado Gridsearch,una técnica que selecciona los parámetros que debemos utilizar para sacar un mayor rendimiento a nuestro modelo indicándole el modelo que queremos utilizar y los parámetros a probar.

**4.2 SELECCIÓN DE VARIABLES**

Iremos reduciendo variables a nuestro dataset a través de tres métodos de filtro:

4.2.1.Teniendo en cuenta la correlación entre variables predictoras y la correlación de esas variables con la variable objetivo

Mostramos la correlación entre las variables que superen el 50% de correlación entre ellas y las variables que tengan una correlación superior al 5% con la variable objetivo.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteGráfico

Descripción generada automáticamente

Eliminamos en primer lugar las variables más correlacionadas entre si (TMAX,TMIN y time\_ext) una vez eliminadas, nos quedamos con las variables que estén por encima del 5% de correlación con la variable objetivo.

Los modelos con esta selección de variables dan los siguientes resultados:

'MODELO 5 : Score modelo SVM reducido teniendo en cuenta correlaciones : 0.8701848998459168 con 13 variables'

'MODELO 6 :Score modelo Random Forest reducido teniendo en cuenta correlaciones : 0.8713405238828967 con 13 variables'

'MODELO 7 : Score modelo Gradient Boosting reducido teniendo en cuenta correlaciones : 0.8701848998459168 con 13 variables'

4.2.2. Mejores variables por el método de chi-cuadrado: Evalúa la probabilidad de correlación entre las variables.

Este método selecciona las siguientes variables : ['superficie', 'lat','lng','muertos','heridos','time\_ctrl','time\_ext','personal','medios', 'PRECIPITACION']

Los resultados de los modelos serían los siguientes:

'MODELO 8 : Score modelo SMV por Chi-cuadrado : 0.8701848998459168 con 10 variables'

'MODELO 9: Score modelo SMV por Chi-cuadrado : 0.8713405238828967 con 10 variables'

'MODELO 10 : Score modelo Gradient Boosting por Chi-cuadrado : 0.8701848998459168 con 10 variables'

### 4.2.3. Selección de las mejores variables a través la importancia de las variables predictoras

Este método solo se ha realizado con Random Forest y Gradient Boosting ,ya que el modelo SVM no lo permite.

Las variables seleccionadas y los resultados para cada uno de ellos serían los siguientes:

**['lat', 'lng', 'personal', 'medios', 'TMAX', 'Año']**

'MODELO 11 : Score modelo Gradient Boosting respecto a las variables con mayor importancia : 0.8701848998459168 con 7 variables'

**['superficie', 'lat', 'lng', 'time\_ctrl', 'personal', 'medios', 'TMEDIA','RACHA', 'SOL', 'Año', 'PRES\_RANGE']**

'MODELO 12 : Score modelo Random Forest respecto a las variables con mayor importancia : 0.8713405238828967 con 12 variables'

1. **SELECCIÓN DEL MODELO**

Cuando realizamos las predicciones en Train, podemos contrastarlas con los datos reales en Test. Si los valores son cercanos, quiere decir que nuestro modelo lo podemos dar por bueno y para evaluarlo utilizaremos la validación cruzada.

La validación cruzada selecciona distintos cortes (en este caso hemos seleccionado 5) para realizar el entrenamiento y la validación del modelo comprobando que los resultados obtenidos son iguales independientes a la muestra.

Tras realizar la validación cruzada a los modelos que consideramos más óptimos, vamos a representar este método mediante boxplot para los tres modelos que hemos ido utilizando.

Gráfico, Gráfico de cajas y bigotes

Descripción generada automáticamente

Los tres modelos tienen una capacidad predictiva por encima del 85% y varianzas muy bajas, lo que demuestra que sus predicciones no son aleatorias. El grafico junto a los resultados de la validación cruzada de los modelos concretos nos lleva a seleccionar el modelo que consideramos que llega a obtener los mejores resultados, así como su sencillez. Por tanto, el modelo seleccionado es Random Forest con la selección de variables del MODELO 12, cuyos resultados de la validación cruzada son los siguientes:

**Texto

Descripción generada automáticamente**

Además, podemos ver también la matriz de confusión de nuestro modelo, así como las variables que lo componen y su importancia.

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

En la matriz de confusión vemos que en la primera clase de la variable predice prácticamente el 100% de los valores, sin embargo, en el resto de categorías sucede todo lo contrario, lo cual seguramente sea causado por el excesivo desbalanceo con el que contábamos. Vamos a comprobarlo mediante un reporte de clasificaciones donde nos indica los resultados de las predicciones de manera desglosada:

Tabla

Descripción generada automáticamente

Efectivamente, las predicciones de la 3º,4º y 5º clase son muy bajas.

Tanto la matriz de confusión como los resultados de nuestras clases indican que el desbalanceo afecta a los resultados, ya que el Accuracy se consigue debido a que la mayoría de la precisión viene dada por nuestra clase mayoritaria y por lo tanto, en este caso , quizás no sea la mejor manera de medir nuestro modelo; para poder realizar una predicción más precisa realizamos un método por el cual se generan observaciones adicionales en las clasificaciones minoritarias, con ello, nuestros resultados serían:

Tabla

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

Se reduce la precisión de la primera categoría, pero mejora la precisión del resto.

\*Futuras mejoras: realizar un modelo dicotómico, separando la variable únicamente entre incendios intencionados y no intencionados. Visión de negocio: aseguradoras.