UADE

Facultad de Ingeniería y Ciencias Exactas

Inteligencia Artificial

Trabajo Práctico 1:

Machine Learning

Profesor:

Christian Parkinson

Integrantes - Grupo 6		
Apellido y Nombre	**Legajo**	**Carrera**
Breuer, Andrés	1120248	Ing. en Informatica
Cabezas, Alexander	1086279	Ing. en Informatica
Cappellano, María	1056140	Ing. en Informatica
Caneva, Matheo	1129004	Ing. en Informatica
Facón, Nicolas	1133988	Ing. en Informatica

Índice

1. Introducción

El siguiente trabajo está enfocado en interactuar con un conjunto de datos sobre accidentes viales con el fin de comprender el flujo de trabajo que consta de interpretar y entender un problema, para abordar una solución con técnicas de Machine Learning y así afianzar los conceptos vistos en clase sobre estos temas.

2. Problema

Se cuenta con un dataset compuesto de 2.974.335 accidentes automovilísticos, cada uno con 49 atributos. Estos accidentes ocurrieron en Estados Unidos entre el año 2016 y 2020. Cada accidente está clasificado por severidad, es decir, que cada accidente tiene una severidad asignada. La severidad se clasifica en 4 categorías (1, 2, 3, 4).

Este caso se puede identificar como un problema aprendizaje supervisado, concretamente un problema de clasificación donde cada punto de dato tiene la clase a predecir. Por ende el objetivo es predecir la severidad de un nuevo accidente en base a estos datos.

3. Variables y características

Variable	Tipo de Variable	Descripción
TMC	Numérico discreto	Código "Traffic Message Channel"
Severity	Numérico discreto / Categórica Ordinal	Severidad del accidente. Tiene los siguientes valores: 1,2,3,4
Start_Time	Fecha y hora	Hora de inicio
End_Time	Fecha y hora	Hora fin
Start_Lat	Numérico continuo	Latitud GPS inicial
Start_Lng	Numérico continuo	Longitud GPS final
Distance(mi)	Numérico continuo	Longitud del camino afectado por el accidente
Side	Categórica Nominal	Dirección de la calle
City	Categórica Nominal	Ciudad
County	Categórica Nominal	Condado
State	Categórica Nominal	Estado
Airport_Code	Categórica Nominal	Código aeroportuario más cercano al accidente
Weather_Timestamp	Fecha y hora	time-stamp del clima
Temperature(F)	Numérico continuo	Temperatura (Fahrenheit)
Wind_Chill(F)	Numérico continuo	Viento (Fahrenheit)
Humidity(%)	Numérico discreto	Humedad relativa (porcentaje)
Pressure(in)	Numérico continuo	Presión atmosférica (pulgadas)
Visibility(mi)	Numérico discreto	Visibilidad (millas)
Wind_Speed	Numérico continuo	Velocidad del viento (milla x Hora)
Wind_Direction	Categórica Nominal	Dirección del viento
Precipitation(in)	Numérico continuo	Cantidad de precipitaciones, si las hay
Weather_Condition	Categórica Nominal	Condición atmosférica (rain, snow, thunderstorm, fog, etc.).

Amenity	Categórica Nominal	Punto de interés (PdI) que indica la presencia de una amenity cerca
Bump	Categórica Nominal	Punto de interés (PdI) que indica la presencia de un reductor de velocidad cerca
Crossing	Categórica Nominal	Punto de interés (PdI) que indica la presencia de una senda peatonal cerca
Give_Way	Categórica Nominal	Punto de interés (PdI) que indica la presencia de un signo de alejarse
Junction	Categórica Nominal	Punto de interés (PdI) que indica la presencia de una intersección cerca
No_Exit	Categórica Nominal	Punto de interés (PdI) que indica la presencia de una calle sin salida cerca
Railway	Categórica Nominal	Punto de interés (PdI) que indica la presencia de un cruce de tren cerca
Roundabout	Categórica Nominal	Punto de interés (PdI) que indica la presencia de una ruta provincial cerca
Station	Categórica Nominal	Punto de interés (PdI) que indica la presencia de una estación (tren, autobús, etc) cerca
Stop	Categórica Nominal	Punto de interés (PdI) que indica la presencia de un signo de alto cerca
Traffic_Calming	Categórica Nominal	Punto de interés (PdI) que indica la presencia de una señal de tráfico calmado cerca
Traffic_Signal	Categórica Nominal	Punto de interés (PdI) que indica la presencia de una señal de mucho tráfico cerca
Turning_Loop	Categórica Nominal	Punto de interés (PdI) que indica la presencia de una señal de giro en u cerca
Sunrise_Sunset	Categórica Nominal	Período del día (day, night) según la salida o caída del sol
Civil_Twilight	Categórica Nominal	Período del día (day, night) según el crepúsculo local

Nautical_Twilight	Categórica Nominal	Período del día (day, night) según el crepúsculo náutico.
Astronomical_Twilight	Categórica Nominal	Período del día (day, night) según el crepúsculo astronómico.

4. Ingeniería de Características y Preprocesamiento

4.1 Transformaciones

Se realizaron diferentes transformaciones en todo los registros del conjunto de datos con el objetivo de preparar los datos para los modelos de Machine Learning que aplicarán y también proporcionar una perspectiva más general del problema a estos modelos. A continuación se describen las transformaciones realizadas.

Se realiza una categorización a los atributos State que consiste en realizar un cambio de los valores de State por regiones de Estados Unidos. Esto se hace debido por la alta cardinalidad de estos atributos, es decir, dichos atributos presentan una alta cantidad de valores únicos, lo cual no permite considerarlos para aplicar la técnica de One-Hot-Encoding. La cantidad de valores únicos son: 11.639 para City, 1.692 para County y 49 para State. El conjunto de regiones que se utilizó tiene una cardinalidad muy baja, igual a 5. En consecuencia, los atributos State, City y County son removidos del conjunto de datos, al momento de realizar el entrenamiento de los modelos de Machine Learning. Ver tabla 4.1.

Tabla 4.1. Reglas para realizar el cambio de membresía de State a Region.

State	Region
al	southeast
ar	southeast
az	southwest
ca	west
co	west
ct	northeast
dc	southeast
de	southeast
fl	southeast
ga	southeast
ia	midwest
id	west

il	midwest
in	midwest
ks	midwest
ky	southeast
la	southeast
ma	northeast
md	northeast
me	northeast
mi	midwest
mn	midwest
mo	midwest
ms	southeast
mt	west
nc	southeast
nd	midwest
ne	midwest
nh	northeast
nj	northeast
nm	southwest
nv	west
ny	northeast
oh	midwest
ok	southwest
or	west
pa	northeast
ri	northeast
sc	southeast
sd	midwest
tn	southeast
tx	southwest

ut	west
va	southeast
vt	northeast
wa	west
wi	midwest
wv	southeast
wy	west

Se realiza una categorización del atributo Weather_Condition a Weather_Type, debido a su alta cardinalidad, lo cuál dificulta el uso de la técnica One-Hot-Encoding. El conjunto de tipo de climas que se utiliza presenta una cardinalidad muy baja, igual a 7. En consecuencia, el atributo Weather_Condition son removidos del conjunto de datos, al momento de realizar el entrenamiento de los modelos. Ver tabla 4.2.

Tabla 4.2. Reglas para realizar el cambio de membresía de Weather_Conditiion a Weather_Type.

Weather_Condition	**Weather_Type**
blowing snow	snowy
ice pellets	sleet
clear	sunny
fog	restricted_visibility
heavy ice pellets	sleet
mist	restricted_visibility
funnel cloud	windy
heavy drizzle	rainy
rain shower	rainy
a precipitation	rainy
squalls	windy
light sleet	sleet
heavy sleet	sleet
snow and thunder	snowy

haze	restricted_visibility
light haze	restricted_visibility
patches of fog	restricted_visibility
light snow showers	snowy
light blowing snow	snowy
thunder and hail	snowy
widespread dust	restricted_visibility
windy	windy
light freezing drizzle	rainy
thunderstorms and snow	snowy
heavy blowing snow	snowy
heavy freezing drizzle	snowy
light freezing rain	rainy
heavy rain showers	rainy
sleet	sleet
thunderstorms and rain	rainy
light freezing fog	restricted_visibility
sand	restricted_visibility
showers in the vicinity	rainy
light rain shower	rainy
rain	rainy
snow and sleet	snowy
light rain	rainy
overcast	cloudy
light thunderstorms and rain	rainy
partial fog	restricted_visibility
mostly cloudy	cloudy
tornado	windy
light thunderstorm	windy

light rain with thunder	windy
drizzle	rainy
heavy smoke	restricted_visibility
shallow fog	restricted_visibility
light snow	snowy
heavy t-storm	rainy
freezing rain	rainy
heavy snow	snowy
t-storm	rainy
thunder	cloudy
light rain showers	rainy
snow	snowy
light snow and sleet	sleet
snow showers	snowy
heavy snow with thunder	snowy
light snow with thunder	snowy
hail	snowy
partly cloudy	cloudy
smoke	restricted_visibility
heavy thunderstorms and rain	rainy
heavy freezing rain	rainy
dust whirlwinds	restricted_visibility
heavy thunderstorms and snow	snowy
heavy rain	rainy
light snow shower	snowy
rain showers	rainy
light thunderstorms and snow	snowy
thunder in the vicinity	cloudy

low drifting snow	snowy
scattered clouds	cloudy
snow grains	snowy
light drizzle	rainy
wintry mix	rainy
light snow grains	snowy
heavy thunderstorms with small hail	windy
fair	sunny
light hail	snowy
volcanic ash	restricted_visibility
light ice pellets	sleet
dust whirls	restricted_visibility
light fog	restricted_visibility
cloudy	cloudy
thunderstorm	windy
blowing dust	restricted_visibility
blowing sand	restricted_visibility
small hail	snowy
drizzle and fog	rainy

Se realiza una categorización del atributo Wind_Speed utilizando un conjunto de reglas utilizadas para clasificar los vientos según su velocidad. Se agrega un nuevo atributo, al conjunto de datos, con la escala del viento según su velocidad, se llama Wind_Scale. Esto se realiza para proporcionar una perspectiva del entorno y también debido a la baja cardinalidad de la clasificación de los vientos. Ver tabla 4.3.

Tabla 4.3. Reglas para realizar la clasificación de los vientos según su velocidad.

Wind_Speed (km/h)	**Wind_Scale**
<1	Calm
1-5	Light Air

6-11	Light Breeze
12-19	Gentle Breeze
20-28	Moderate Breeze
29-38	Fresh Breeze
39-49	Strong gale
50-61	Fresh Breeze
62-74	Fresh gale
75-88	Strong gale
89-102	Whole gale
103-117	Storm
>117	Hurricane

Se realiza una transformación sobre el atributo Start_Time. De este modo, se obtiene el año, el mes, el día y la hora del accidente. Estos valores se agregan al conjunto de datos con los siguientes nombres: Start_Time__year, Start_Time__month, Start_Time__day y Start_Time__hour. En consecuencia, el atributo Start_Time es removido del conjunto de datos.

Se usa el atributo Start_Time para obtener más información como la temporada del año, el trimestre del año y el momento del día. La temporada del año se agrega al conjunto de datos como Season, que puede ser: winter, spring, summer y fall. El trimestre del año se agrega al conjunto de datos como Quarter, con valores posibles: q1, q2, q3, q4. Finalmente, el momento del día se agrega al conjunto de datos como Day_Part, que puede ser: night, morning, afternoon y evening.

Se realiza una transformación de los atributos Start_Lat y Start_Lng a coordenadas tridimensionales. Esto se realiza debido a que estos atributos no son fáciles de estandarizar y pueden afectar el cálculo de distancia por las diferencias de alturas de dichas coordenadas. Las coordenadas tridimensionales son agregadas al dataset como Start_x, Start_y y Start_z.

Se realiza una transformación de los siguientes atributos: Amenity, Bump, Crossing, Give_Way, Junction, No_Exit, Railway, Roundabout, Station, Stop, Traffic_Calming, Traffic_Signal, Turning_Loop. Los valores de estos atributos se utilizan para determinar el grado de influencia de la zona según su complejidad. El cálculo que se realiza consiste en atribuirle a todos un peso equivalente sobre este valor final, un peso igual a 1, luego se suman todos los valores iguales a verdadero y se divide por el número total de estos atributos, son 13 atributos en total. De esta forma el grado de influencia de

la zona en el accidente varía entre 0 y 1. Este grado de influencia se agrega al conjunto de datos con el nombre de Environment_Influence. En consecuencia, los 13 atributos utilizados para calcular este valor son removidos del conjunto de datos.

Se aplica la técnica One-Hot-Encoding a las variables categóricas como: Wind_Direction, Region, Weather_Type, Wind_Scale, Season, Quarter y Day_Part. Al aplicar esta técnica se crean nuevos atributos que corresponden a los valores únicos de cada uno de estos atributos. Es por eso que es importante reducir la cardinalidad de algunos atributos como se mencionó anteriormente.

Se estandarizan los atributos: Start_x, Start_y, Start_z, Distance, Temperature, Wind_Chill, Humidity, Pressure, Visibility y Environment_Influence.

4.2 Imputación de Pérdidos

Al analizar el conjunto de datos para determinar el porcentaje de valores faltantes o valores pérdidos por atributo, ver Tabla 4.4. Se decide eliminar del conjunto de datos los atributos con un porcentaje alto de valores faltantes, mayor a 70% y se mantienen los atributos con un porcentaje menor a 70%. Se eliminan los registros para los atributos con un porcentaje de faltantes menor al 5%.

Los atributos que se eliminan son los siguientes: Source, TMC, End_Time, End_Lat, End_Lng, Number, Street, Side, Descripcion, Zipcode, TimeZone, Airport_Code, Weather_Timestamp, Sunrise_Sunset, Civil_Twilight, Nautical_Twilight y Astronomical_Twilight. Hay algunos campos que son removidos porque consideramos que no son relevantes para el problema y notamos que hay algunos que no aportan nueva información, como sucede con el atributo Weather_Timestamp que casi tiene los mismos valores que el atributo Start_Time. Se considera al atributo Airport_Code como no relevante debido a que aporta la misma información que el State, City y County. también se debe a que para darle un mayor uso a este atributo se tendría que combinar con otros conjuntos de datos para obtener la latitud y longitud del aeropuerto, lo cuál va más allá del alcance del trabajo práctico.

Tabla 4.4. Porción de faltantes por atributo en el conjunto de datos.

Atributo	**Porción de Faltantes**
Source	0

TMC	24.4784
Severity	0
Start_Time	0
End_Time	0
Start_Lat	0
Start_Lng	0
End_Lat	75.5216
End_Lng	75.5216
Distance(mi)	0
Description	0
Number	64.4717
Street	0
Side	0
City	0.0028
County	0
State	0
Zipcode	0.0296
Country	0
Timezone	0.1063
Airport_Code	0.1913
Weather_Timestamp	1.2341
Temperature(F)	1.8849
Wind_Chill(F)	62.287
Humidity(%)	1.9895
Pressure(in)	1.6186
Visibility(mi)	2.2086
Wind_Direction	1.5163
Wind_Speed(mph)	14.8215
Precipitation(in)	67.1867
Weather_Condition	2.2167

Amenity	0
Bump	0
Crossing	0
Give_Way	0
Junction	0
No_Exit	0
Railway	0
Roundabout	0
Station	0
Stop	0
Traffic_Calming	0
Traffic_Signal	0
Turning_Loop	0
Sunrise_Sunset	0.0031
Civil_Twilight	0.0031
Nautical_Twilight	0.0031
Astronomical_Twilight	0.0031

Se mantienen los atributos Wind_Chill, Precipitation y Wind_Speed, aún cuando sus porciones de faltantes en el conjunto de datos es de ~62%, ~67%, ~15% respectivamente. Para estos atributos se decide utilizar el valor más probable para llenar los faltantes. Para determinar el valor más probable se utiliza un modelo de Machine Learning, el **KNeighborsRegressor** que permite predecir números continuos. Para cada caso se obtiene el mejor modelo, el modelo con la puntuación más alta, a partir de un rango de número de vecinos. El rango que se utiliza es 2, 3, 5, 7 y 9. Ver tabla 4.5.

Tabla 4.5. Puntuación más alta obtenida y el número de vecinos utilizado que permite obtener el mejor modelo para determinar el valor más probable por atributo.

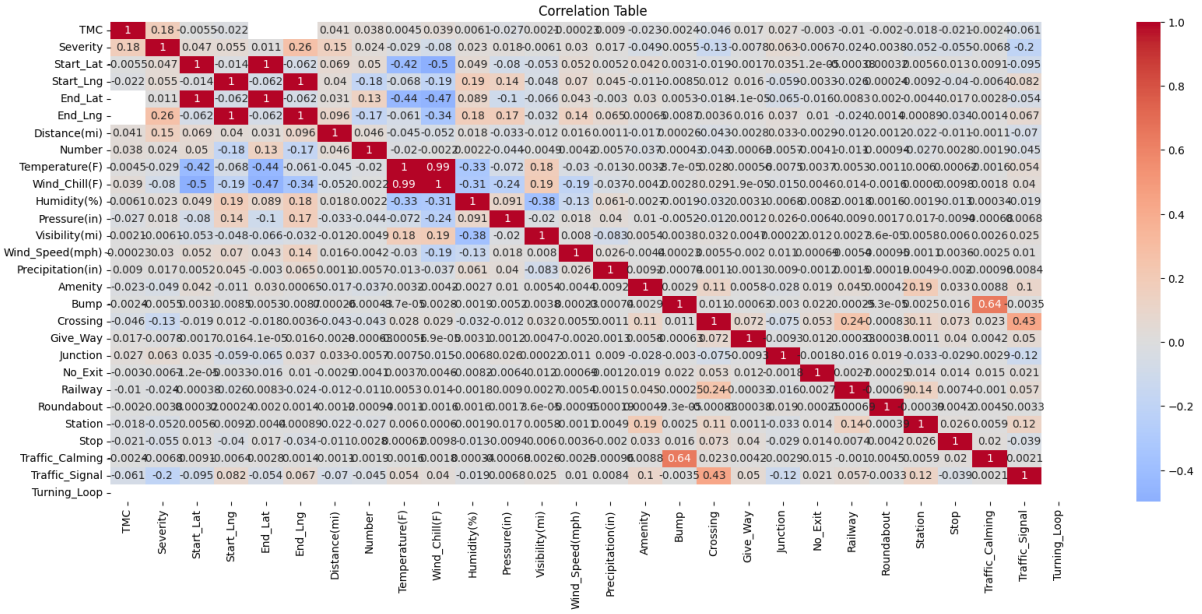
Atributo	Número de Vecinos	Puntuación
Wind_Chill	5	0.9743

Precipitation	2	0.9029
Wind_Speed	7	0.9913

4.3 Selección de Características

Se determina la correlación entre los atributos del conjunto de datos inicial para identificar qué atributos se pueden descartar. La correlación es calculada mediante el coeficiente de Pearson que mide la correlación lineal entre dos series de valores; este coeficiente es la relación entre la covarianza de dos variables y el producto de sus desviaciones estándar. Sus valores varían entre -1 y 1, 1 implica que hay una estrecha relación entre las dos variables y 0 implica que no hay relación alguna entre ellas. Con los resultados de aplicar esta técnica se decide remover el atributo Wind_Chill, porque se puede observar que hay una estrecha relación con el atributo Temperature. La correlación se plotea en un mapa de calor, ver figura 4.6.

Figura 4.6. Mapa de calor que muestra la correlación entre los atributos del conjunto de datos.



Para realizar la selección de las características más relevantes, que se pueden utilizar para el entrenamiento de los modelos de Machine Learning, se utilizó el modelo ExtraTrees, ExtraTreesClassifier para ser más precisos. El modelo ExtraTrees es un modelo similar al Random Forest con la única diferencia que este usa aleatoriedad para dividir las características en vez de utilizar un algoritmo greedy como lo hace el Random Forest lo cuál lo hace menos costoso computacionalmente.

Al utilizar dicho modelo junto a SelectFromModel podemos determinar las características más relevantes, ya que selecciona las características donde su importancia absoluta sea superior o igual a la media de las importancias. Las características relevantes son: Distance, Temperature, Humidity, Pressure, Visibility, Wind_Speed, Precipitation, Start_x, Start_y, Start_z, Environment_Influence, start_time__year, start_time__month, start_time__day, start_time__hour, region__west, region__northeast, region__southwest, region__southeast, region__midwest, season__winter, season__fall, quarter__q4, day_part__morning, day_part__afternoon, day_part__evening, day_part__night, wind_scale__strong_breeze, wind_scale__gentle_breeze, wind_scale__light_breeze, wind_scale__fresh_breeze, wind_scale__moderate_breeze, weather_type__cloudy, weather_type__sunny, weather_type__rainy.

4.4 Distribución de los Datos

Se analizan los datos para determinar cómo se distribuyen los puntos de datos por clase (atributo Severity) y por año (atributo Start_Time) considerando que la clase está relacionada al tiempo en el que ocurre el accidente. Se puede observar que los datos están muy desbalanceados, la proporción de accidentes Severity=2 y Severity=4 es muy baja. Esto puede llegar a ser un problema para el entrenamiento de ciertos modelos, representa un dato a considerar cuando se observan las métricas de cada modelo. Ver tabla 4.7.

Tabla 4.7. Distribución de los accidentes por severidad y año.

start_time__year	Severity	count	total_count	fraction
2016	1	219	397.771	0
2016	2	261.987	397.771	0,66
2016	3	121.901	397.771	0,31
2016	4	13.664	397.771	0,03
2017	1	270	694.396	0
2017	2	448.322	694.396	0,65
2017	3	222.688	694.396	0,32
2017	4	23.116	694.396	0,03
2018	1	252	864.910	0
2018	2	558.329	864.910	0,65

2018	3	281.790	864.910	0,33
2018	4	24.539	864.910	0,03
2019	1	196	924.416	0
2019	2	666.595	924.416	0,72
2019	3	230.686	924.416	0,25
2019	4	26.939	924.416	0,03

4.5 División de Datos

Para el entrenamiento y verificación de los modelos de Machine Learning se divide el conjunto de datos por año y mes (atributo Start_Time), de esta forma se obtiene 3 subconjuntos de datos, un conjunto para entrenamiento, un conjunto para validación y otro conjunto para prueba (test). El conjunto para entrenamiento representa el 60% de los datos, los conjuntos de validación y prueba representan un 20% cada uno. El conjunto de entrenamiento abarca los años 2016, 2017 y parte del 2018, el conjunto de validación abarca los años 2018 y 2019 y el conjunto de prueba comprende el restante del año 2019. Ver tabla 4.8.

Tabla 4.8. División de Datos: subconjuntos de entrenamiento, validación y prueba (test).

Conjunto de Entrenamiento

start_time__year	2016	2017	2018
start_time__month			
1	0	53325	72419
2	973	49605	69291
3	6149	55660	72435
4	17635	46856	70982
5	17137	40164	74456
6	29600	44968	62361
7	44487	42046	64006
8	54809	78286	74475
9	53070	73581	71202
10	53771	72629	0
11	62887	67893	0

12	57253	69383	0

Conjunto de Validación

start_time__year	2018	2019
start_time__month		
1	0	77139
2	0	72664
3	0	67662
4	0	71494
10	79832	0
11	80111	0
12	68238	0

Conjunto de Prueba (test)

start_time__year	2019
start_time__month	
5	12922
6	63742
7	64246
8	72896
9	84761
10	103159
11	79597
12	94975

5. Algoritmos aplicados

Para la predicción de resultados elegimos usar técnicas de aprendizaje supervisado de clasificación: KNN (K-Near Neighbors), Random Forest y SVM (Support Vector Machine) con el kernel lineal. Utilizamos las librerías de Sci-Kit Learn tanto para los modelos (*sklearn*), skopt (scikit-optimize) para la optimización bayesiana dado que hasta la fecha es el método más eficiente.

Se realizaron algunos experimentos para balancear las severidades 1 y 4 con la librería imblearn pero trajo como consecuencia que el conjunto de datos aumentó de tamaño lo cuál incrementa la complejidad computacional del problema. Se decidió realizar la optimización bayesiana sin balancear los datos.

Para cada modelo que se va a explicar a continuación se realizó un ajuste de hiper parámetros con el método de optimización bayesiana:

Tabla 5.1. Parámetros elegidos para Random Forest:

Random Forest			
parámetros	mínimo	Máximo	mejor modelo
max_depth	2	22	19
min_samples_split	10	200	157
n_estimators	50	500	162

Tabla 5.2. Parámetros elegidos para Knn:

Knn			
parámetros	mínimo	Máximo	mejor modelo
n_neighbors	2	22	18

Tabla 5.3. Parámetros elegidos para Knn:

Linear SVM			
parámetros	mínimo	Máximo	mejor modelo
C	0,001	1.000,00	129

5.1. StratifiedKFold and cross validation:

La validación cruzada o cross validation es un método que consiste en evaluar y probar el rendimiento de un modelo de machine learning, con el fin de encontrar un mejor modelo rápidamente. Esta técnica ayuda a la comprensión y aplicación de este modelado predictivo, siendo fácil y sencilla de aplicar. Además, la validación cruzada tiene menor sesgo al estimar las habilidades del modelo.

Hay varias formas de realizar la validación cruzada una de ella es el StratifiedKFold donde la distribución de las clases se conserva para la división de entrenamiento y test. Hay estudios que muestran que este tipo de validación cruzada produce mejores estimaciones de sesgo y varianza, especialmente en casos de proporciones de clase desiguales.

Stratified k-Fold realiza los siguientes pasos:

1) Elige una cantidad de pliegues – k
2) Divide el conjunto de datos en k pliegues. Cada pliegue debe contener aproximadamente el mismo porcentaje de muestras de cada clase objetivo que el conjunto completo.
3) Elige k – 1 pliegues que será el conjunto de entrenamiento. El pliegue restante será el conjunto de datos de prueba (test).
4) Entrena el modelo con el conjunto de entrenamiento. En cada iteración se debe entrenar un nuevo modelo.
5) Válida en el conjunto de prueba (test)
6) Guarda el resultado de la validación.
7) Repite los pasos 3 – 6 k veces. Cada vez usa el restante pliegue como un conjunto de datos de prueba. Al final se habrá validado el modelo en cada pliegue que se tenga, es decir k veces.
8) Para obtener la puntuación final promedia los resultados obtenidos en el paso 6.

5.2. Optimización Bayesiana:

La optimización bayesiana es una estrategia de diseño secuencial para la optimización global de funciones de caja negra que no asume ninguna forma.

La optimización global es un problema desafiante de hallar una entrada de como resultados el costo mínimo o máximo de una función objetivo dada.

Típicamente la forma de la función objetivo es compleja e intratable de analizar y, a menudo, no convexo, no lineal, de alta dimensión, ruidoso y computacionalmente costoso de evaluar.

La optimización bayesiana provee una técnica basada en el teorema de bayes para dirigir la búsqueda de un problema de optimización global que sea eficiente y eficaz. Funciona construyendo un modelo probabilístico de la función objetivo llamado función sustituta que luego se busca de manera eficiente con una función de adquisición antes de elegir varias muestras candidatas para su evaluación en la función objetivo real.

La optimización bayesiana se utiliza a menudo en el aprendizaje automático aplicado para ajustar los hiper parámetros de un modelo de buen rendimiento determinado en un conjunto de datos de validación.

Se utiliza esta estrategia por una investigación realizada y se dice que la optimización bayesiana es la estrategia más eficiente hasta la fecha. Si se compara con el grid search, el grid search tiene el inconveniente de no escalar bien y un incremento en el tamaño del espacio de búsqueda de los hiper parámetros puede resultar un aumento exponencial de tiempo y recursos.

5.3. k Nearest Neighbors

K-Nearest Neighbors (kNN) es un algoritmo de aprendizaje automático que destaca por su simplicidad y efectividad en ciertos tipos de problemas de clasificación y regresión. Una de sus principales ventajas es su naturaleza no paramétrica, lo que significa que no hace suposiciones explícitas sobre la forma del modelo subyacente, lo que le permite adaptarse a diversas estructuras de datos. Además, su implementación es directa, lo que facilita su comprensión y uso, especialmente en conjuntos de datos pequeños y con un número moderado de características relevantes.

Sin embargo, KNN tiene sus limitaciones. Su rendimiento puede disminuir significativamente en conjuntos de datos grandes debido a la necesidad de calcular la distancia a cada punto de datos, lo que también lo hace computacionalmente intensivo. La presencia de características irrelevantes o redundantes puede afectar negativamente su precisión, y sufre en situaciones de alta dimensionalidad, un fenómeno conocido como la "maldición de la dimensionalidad". Además, kNN puede ser sensible a datos desbalanceados, lo que podría llevar a un sesgo en la clasificación. Finalmente, elegir el número adecuado de vecinos, 'k', es crucial para su buen funcionamiento, pero no existe una regla fija para determinar este valor, lo que puede ser un desafío en la práctica.

Sus ventajas y limitaciones se pueden resumir de la siguiente manera:

Ventajas:

- El algoritmo es simple y sencillo de implementar.
- No se necesita construir un modelo y hacer un fine tuning con muchos parámetros o hacer supuestos adicionales.
- El algoritmo es versátil. Este puede ser usado para clasificación,regresión o búsqueda.

Limitaciones:

- Este algoritmo se vuelve más lento al incrementar la cantidad de ejemplos y variables.

5.4. Random Forest

Random Forest es conocido por su robustez y precisión, especialmente en problemas de clasificación y regresión. Una de sus principales fortalezas es su capacidad para manejar conjuntos de datos grandes y complejos, proporcionando resultados precisos incluso en presencia de un gran número de características y datos faltantes. Es menos propenso al sobreajuste gracias a su enfoque de ensamble, que combina múltiples árboles de decisión. Además, Random Forest puede manejar tanto variables numéricas como categóricas y proporciona una medida útil de la importancia de las características. Sin embargo, puede ser computacionalmente intensivo, especialmente con un gran número de árboles, y su modelo puede ser difícil de interpretar debido a su naturaleza de ensamble. Además, aunque es menos sensible al ruido y a las características irrelevantes que los árboles de decisión individuales, aún puede verse afectado por estos factores.

Sus ventajas y limitaciones se pueden resumir de la siguiente manera:

Ventajas:

- Puede manejar gran volumen de variables y ejemplos.
- Es versátil, puede usarse tanto para tareas de regresión como clasificación. Puede manejar atributos binarios, atributos categóricos y continuos. Es necesario realizar muy poco preprocesamiento. No es necesario cambiar la escala ni transformar los datos.
- Es paralelizable lo que significa que se puede dividir el proceso en muchas máquinas. Esto hace que el tiempo computacional sea rápido.
- Es más rápido de entrenar que un árbol de decisión por que trabaja sólo con un subconjunto de atributos por lo que se puede trabajar con cientos de atributos. La velocidad de predicción es

significativamente más rápida que la velocidad de entrenamiento porque podemos guardar modelos generados previamente para usos futuros.
- Maneja los valores atípicos agrupándolos. También es indiferente a las características no lineales.
- Intenta minimizar la tasa de error general, por lo que cuando tenemos un conjunto de datos desequilibrado, la clase más grande obtendrá una tasa de error baja mientras que la clase más pequeña tendrá una tasa de error mayor.
- Cada árbol de decisión tiene una gran varianza, pero un sesgo bajo. Pero como promediamos todos los árboles en un bosque aleatorio, también promediamos la varianza para tener un modelo de sesgo bajo y varianza moderada.

Limitaciones:

- Los modelos de Random Forest no son interpretables son como cajas negras.
- Para conjuntos de datos muy grandes, el tamaño de los árboles puede consumir mucha memoria.
- Puede tender a sobre ajustarse, por lo que debes ajustar los hiper parámetros.

5.5. Linear Support Vector Machine

SVM es altamente efectivo en espacios de alta dimensionalidad y cuando existe una clara marginación entre las clases. Es particularmente eficaz en problemas de clasificación complejos, donde puede construir hiperplanos óptimos para separar diferentes clases. Una ventaja clave de SVM es su capacidad para utilizar diferentes funciones del kernel, lo que le permite adaptarse a varias estructuras de datos. Sin embargo, SVM puede ser menos efectivo en conjuntos de datos con un gran volumen de muestras, ya que su tiempo de entrenamiento escala de manera no favorable con el tamaño del conjunto de datos. Además, la elección y el ajuste de los parámetros del kernel y los valores de regularización pueden ser complicados y requieren un cuidadoso ajuste para obtener un rendimiento óptimo. Además, al igual que Random Forest, SVM puede ser desafiante de interpretar, especialmente con kernels no lineales.

El Linear SVM es una variante específica del algoritmo SVM (Support Vector Machine) que se utiliza especialmente para clasificar datos que son linealmente separables. Esto significa que puede trazar una línea (en dos dimensiones) o un plano (en más dimensiones) para separar diferentes clases de puntos.

Los beneficios principales de utilizar Linear SVM en comparación con SVM tradicionales incluyen mayor simplicidad y velocidad, siendo especialmente eficaz para datos linealmente separables. Es altamente eficiente en espacios de alta dimensión y los resultados son más fáciles de interpretar debido a su naturaleza lineal. Además, ofrece un menor riesgo de sobreajuste, particularmente en contextos con datos de alta dimensionalidad. Estas características hacen de Linear SVM una opción atractiva para aplicaciones como clasificación de textos y análisis de sentimientos.

Sus ventajas y limitaciones se pueden resumir de la siguiente manera:

Ventajas:

- Ser efectivos en espacios de alta dimensionalidad, aún cuando el número de dimensiones supera el número de muestras.
- Eficiente gestión de la memoria, al usar solo un subconjunto de puntos en la función de decisión.
- Si el kernel es lineal puede trabajar con gran volumen de datos.

Limitaciones:

- Si el número de características es mucho mayor que el número de muestras, resulta crucial evitar el sobreentrenamiento escogiendo el kernel y el término de regularización adecuados.
- SVM no proporciona estimaciones de probabilidad.
- La frontera de decisión depende directamente de los valores más próximos, aunque sean erróneos.
- SVM es muy dependiente de la escala de los datos, por lo que conviene escalar adecuadamente.

6. Resultados

A continuación se presentan los resultados obtenidos de las métricas de Precisión, Sensitividad y F1 score de acuerdo a la optimización bayesiana.

Tabla 6.1: Valores de la métrica de Precisión

	Precisión		
Conjunto de datos	Random Forest	KNN	SVM (lineal)
Entrenamiento	77,76	68,48	**60,12**
Validación	71,91	60,77	**62,84**
Test	72,46	65,4	61,62

Revisando la métrica anterior, el algoritmo de Support Vector Machine tiene menor diferencia si se compara los conjuntos de datos de entrenamiento y validación y entrenamiento y test. Así mismo se mantiene poca variación en validación y test por lo que este algoritmo es estable mientras que Random forest y KNN no lo son. Es importante destacar que el modelo de Support vector Machine es regular ya que se encuentra un poco por encima su valor a si se estimara con el lanzamiento de una moneda.

Tabla 6.2: Valores de la métrica de la Sensitividad

	Recall (Sensitividad)		
Conjunto de datos	Random Forest	KNN	SVM (linear)
Entrenamiento	77,68	69,84	**56**
Validación	73,74	65,39	**57,09**
Test	73,56	68,36	61,42

La sensitividad es la proporción de casos pertenecientes a la clase positiva que son correctamente clasificados. De acuerdo a los valores anteriores se observa que el modelo de Support Vector Machine es estable si se compara el valor de la métrica para entrenamiento y validación pero si se hace el mismo ejercicio para test la diferencia es de 4 puntos. Por otro lado, la métrica de la Sensitividad muestra que los modelos de KNN y Random Forest son inestables. Dado que la sensitividad toma el porcentaje de cada clase positiva que se clasifican correctamente, es claro que hay muchos ejemplos que se están clasificando mal en validación y test así mismo el hecho de que el dataset se encuentre desbalanceado hace que esta métrica sea difícil de mejorar.

Tabla 6.3: Valores de la métrica de la F1 score

	F1 score		
Conjunto de datos	Random Forest	KNN	SVM (linear)
Entrenamiento	75,53	65,83	**49,56**
Validación	71,14	61,46	**52,58**
Test	72,33	66,25	59,62

Dado que el F1 score es la media armónica entre la especificidad y sensitividad vemos que los algoritmos mantienen gran diferencia entre entrenamiento, validación y test comparándolos uno a uno y esto está afectado por que el dataset tiene clases muy desbalanceadas.

Tabla 6.4: Valores de las métricas de la Precisión, Sensibilidad y F1 score del mejor modelo de Support Vector Machine sobre el conjunto de datos de entrenamiento:

Classification Report - Entrenamiento: (Support Vector Machine):				
métrica	precisión	sensitividad	F1 score	support
severity=1	0,00	0,01	0,00	699
severity=2	0,69	0,84	0,76	1102076
severity=3	0,47	0,00	0,01	553891
severity=4	0,09	0,56	0,15	52615
accuracy			0,56	1709281
macro avg	0,31	0,35	0,23	1709281
weighted avg	0,6	0,56	0,5	1709281

Del anterior cuadro se puede afirmar que la severidad 1, 3 y 4 no fueron clasificados correctamente por el modelo dado que representan 0.4 por mil, 32,4% y 3% respectivamente y mucha variación de las métricas para cada clase dado al desbalance de la misma.

Se observa que la exactitud(accuracy) se ubicó en 56 lo cual indica que el modelo no es muy bueno clasificando y dado que la exactitud es una medida que puede dar como resultado un clasificador que esté sesgado a la clase más frecuente **no** debe considerarse con conjuntos de datos desbalanceados. Se observa que el macro average F1 score es muy bajo así como el weighted lo cual indica que el modelo no está clasificando bien cada una de las clases-

Tabla 6.5: Valores de las métricas de la Precisión, Sensibilidad y F1 score del mejor modelo de Support Vector Machine sobre el conjunto de datos de validación:

Classification Report - Validación (Support Vector Machine):				
métrica	precisión	sensitividad	F1 score	support
severity=1	0,00	0,00	0	102
severity=2	0,72	0,81	0,77	347234
severity=3	0,47	0,00	0,01	148822
severity=4	0,08	0,56	0,14	16953
accuracy			0,57	513111
macro avg	0,32	0,34	0,23	513111
weighted avg	0,63	0,57	0,53	513111

Del anterior cuadro se puede afirmar que la severidad 1, 3 y 4 no fueron clasificados correctamente por el modelo dado que representan 0.2 por mil, 29% y 3,3% respectivamente, teniendo diferentes magnitudes para las tres métricas. Se mantiene el mismo comportamiento donde la precisión, la sensitividad y F1 score para cada clase tienen mucha variación. Por otro lado, las métricas del macro average F1 score es muy bajo así como el weighted lo cual indica que el modelo no está clasificando bien cada una de las clases.

Tabla 6.6: Valores de las métricas de la Precisión, Sensibilidad y F1 score del mejor modelo de Support Vector Machine sobre el conjunto de datos de test:

Classification Report - Test (Support Vector Machine):				
métrica	precisión	sensitividad	F1 score	support
severity=1	0,00	0,02	0	123
severity=2	0,79	0,80	0,79	426333
severity=3	0,12	0,00	0	129986
severity=4	0,07	0,57	0,12	14730
accuracy			0,61	571172
macro avg	0,24	0,35	0,23	571172
weighted avg	0,62	0,61	0,60	571172

Del anterior cuadro se puede afirmar que la severidad 1, 3 y 4 no fueron clasificados correctamente por el modelo dado que representan 0.2 por mil, 22,75% y 2,6% respectivamente, teniendo diferentes magnitudes para las tres métricas.

Las clases para las métricas de: precisión, la sensitividad y F1 score tienen una gran variación y esto es ocasionado por el desbalance de las clases presentes. Por otro lado, las métricas del macro average y F1 score es muy bajo así como el weighted lo cual indica que el modelo no está clasificando bien cada una de las clases.