Driving Behavior

Andres Eduardo Nowak de Anda A01638430,
Isaac Emanuel García González A01566697,
Samuel Alejandro Diaz del Guante Ochoa A01637592, and
Ernesto Lopez Villarreal A01552124

Instituto Tecnologico de Monterrey, Guadalajara Jalisco, Mexico
Gildardo Sánchez Ante,
Javier Mauricio Antelis Ortíz,
Alberto De Obeso Orendain,
Brenda Ivette García Maya, and
Gilberto Ochoa Ruiz

Abstract. En la clase de Inteligencia artificial avanzada para la ciencia de datos I, se realizó un proyecto donde se hizo un análisis del comportamiento de conductores de vehículos a motor, estos se dividieron en tres categorías, lento, normal y agresivo. Se elaboro una base de datos con variables obtenidas mediante un celular de la marca Samsung Galaxy S21, estas se registraron por medio del giroscopio midiendo grados de rotación por segundo, el acelerómetro midiendo la aceleración, estos en las tres coordenadas cartesianas y el tiempo cuando se guardaron estas medidas registrando cada medio segundo. En el documento se mostrarán diferentes modelos de clasificación, estos en su mayoría supervisados con solamente un modelo no supervisado, además, se analizarán los resultados mediante gráficas y distintos métodos de análisis de variables, usando las bases de datos divididas en prueba y entrenamiento para su uso en los diferentes modelos y obtener mejores resultados de predicción.

Keywords: Comportamiento de manejo, Machine Learning, Entrenamiento no supervisado, algoritmo, ANOVA Test, Information Gain, correlación, datos atípicos, matriz de confusión

1 Introducción

De acuerdo con Oracle la inteligencia artificial (IA) se refiere a aquellos sistemas o máquinas que puedan replicar la inteligencia humana y puedan mejorar de manera iterativa a partir de la información que recopilan. La aplicación de la IA es cada vez más común en nuestra vida, esto porque cada vez hay más aplicaciones útiles. De acuerdo con la Harvard Business Review, las empresas utilizan la IA principalmente para:

- Detectar y disuadir intrusiones de seguridad (44%)
- Resolver problemas tecnológicos de los usuarios (41%)

- Reducir el trabajo de la gestión de producción (34%)
- Medir el cumplimiento interno en el uso de proveedores aprobados (34%)

Dentro de la IA existe una disciplina conocida como aprendizaje automático,cuyos algoritmos revisan los datos y se vuelven capaces de predecir comportamientos futuros. Automáticamente, también en este contexto, implica que estos sistemas se mejoran de forma autónoma con el tiempo, sin intervención humana.

En este documento se describe el desarrollo y uso de diferentes modelos de predicción con ayuda de algoritmos de aprendizaje automático en un problema real. Este problema es reconocido por la Fundación de Seguridad Vial, quienes identifican que el 55.7% de todos los accidentes viales son causados por conductores agresivos. La conducción agresiva incluye el exceso de velocidad, las pausas repentinas y los giros bruscos a la izquierda o a la derecha. Todos estos eventos se reflejan en los datos del acelerómetro y el giroscopio.

A lo largo del documento se presentarán diferentes modelos de aprendizaje automático que nos ayudarán a generar predicciones para saber si un conductor es lento, normal o agresivo.

2 Analisis de los datos

Para la realización de los modelos de clasificación lo primero que se hizo fue identificar las variables existentes. Son tres de aceleración (m/s^2) y rotación ($^\circ/s$) sobre los tres ejes, clases de compartamiento y tiempo en el que se registraron los datos (*segundos*). El conjuto de datos esta dividido en dos partes, uno para entrenamiento y otro para pruebas.

Table 1. Resumen descriptivo de las variables numericas en el conjunto de datos de entrenemiento.

	AccX	AccY	AccZ	GyroX	GyroY	GyroZ	Timestamp
count	3644.000000	3644.000000	3644.000000	3644.000000	3644.000000	3644.000000	3.644000e+03
mean	0.040467	-0.073418	0.008271	0.001593	-0.001273	0.007949	3.582707e+06
std	0.985653	0.903408	0.985061	0.066918	0.126205	0.115687	6.421479e+02
min	-4.636523	-4.699795	-7.143998	-0.751822	-1.587028	-1.236468	3.581629e+06
25%	-0.550695	-0.592540	-0.558464	-0.028558	-0.053756	-0.029398	3.582121e+06
50%	0.003931	-0.080833	0.002262	0.001985	-0.001833	0.002978	3.582702e+06
75%	0.595987	0.452401	0.556157	0.031918	0.051313	0.040852	3.583270e+06
max	4.985548	4.245151	5.171739	0.849255	1.679879	1.190500	3.583791e+06

Se puede observar en la tabla la distribución de datos minimos y maximos junto a su distribución dada por cuartiles. Tambien se obtiene el promedio y la desviación

estandar. En el caso de aceleración el rango de los datos se encuentran entre -7 a 5 m/s^2 y la rotación de -1.5 a 1.5 $°/s$.

Lo siguiente fue realizar el balance de las clases. La clase "Slow" cuenta con 1331 (36.5%) registros, 1200 (32.9%) registros para la clase "Normal" y 1113 (30.5%) registros para la clase "Aggressive". En general, los datos cuentan con una distribución cuasi-equivalente.

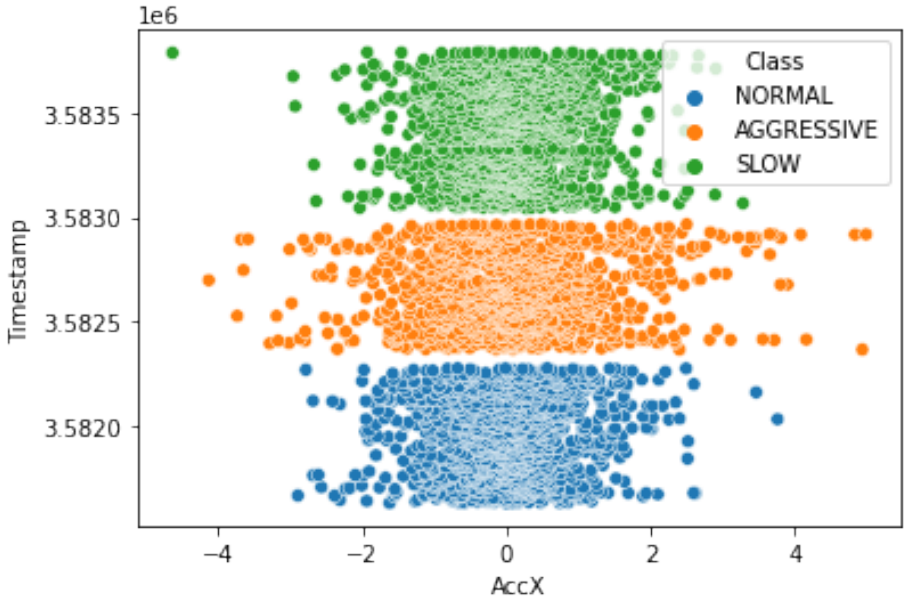
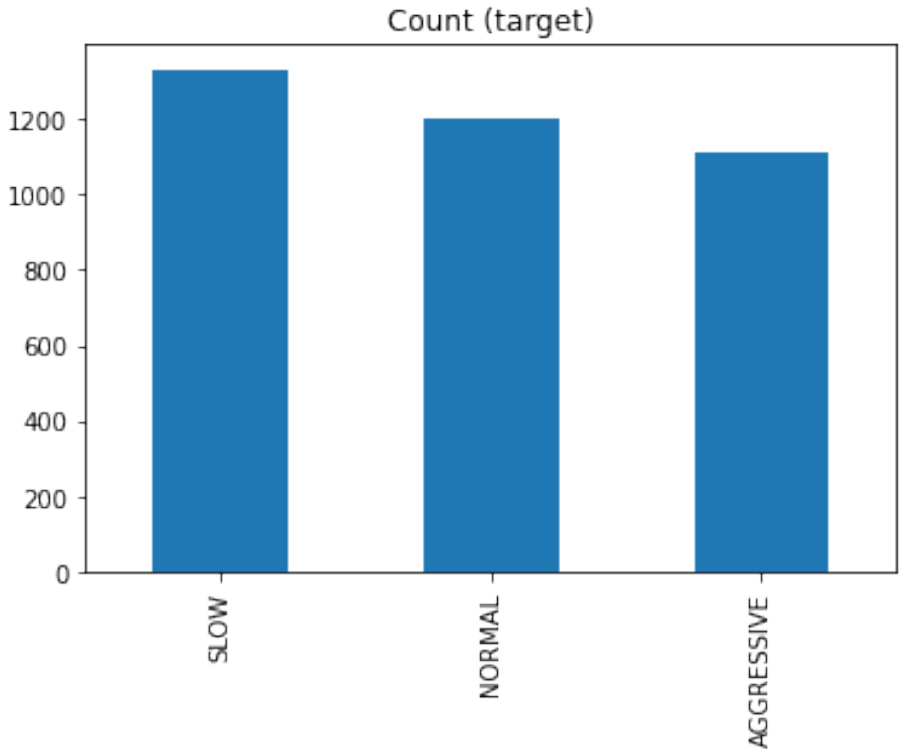


Fig. 1. Distribución total de registros por clase.

2.1 Balance de datos

Al comparar las variables, de la colección de datos para entrenamiento, de aceleración y giroscopio agrupadas por clase contra tiempo se presentan un par de situaciones peculiares. Para empezar, existen rangos largos de tiempo en donde el comportamiento del conductor se mantiene igual. Añadiendo a esto, la dispersión de datos entre cada una de las clases es bastante similar y no existe un patrón claro de clasificación en ninguna de las variables de aceleración o giroscopio. Esto va a presentar un reto significativo para predecir el tipo de comportamiento.

Fig. 2. Comparación de variables entre aceleración en el eje x contra tiempo de registro agrupado por clase de conductor.



2.2 ANOVA

Adicionalmente, se realizó un análisis de varianza conocido como ANOVA. La prueba de ANOVA es una herramienta estadística principalmente utilizada para determinar la influencia que tienen las variables independientes en la variable dependiente en un estudio de regresión. Si no existe una varianza real entre los grupos, la razón F del ANOVA debería ser cercana a 1. La fórmula de ANOVA está dada por $F = MSE/MST$.

La Hipótesis Nula del test ANOVA es $H_0 : \mu_1 = \mu_2 = \ldots \mu_i$ Ha al menos que una es diferente si p-valor ≤ 0.05 entonces se rechaza la H_0.

En este caso la mayoría de las variables aceptan la hipótesis nula por lo que se puede concluir que no tienen una relación significativa con la variable a predecir con excepción de aceleración en X y Y que se encuentran por debajo del p-valor

Table 2. Test anova entre las variables de aceleración y rotación con respecto la variable de clasificación.

	F-statistic
AccX	0.0201
AccY	9.36e-05
AccZ	0.399
GyroX	0.703
GyroY	0.870
GyroZ	0.261

establecido y por lo que se rechaza la hipótesis nula por lo que son mejores candidatos para crear un buen modelo de predicción.

2.3 Information Gain

Para machine learning existe algo conocido como "information gain" que es la cantidad de información obtenida sobre una variable aleatoria o señal a partir de la observación de otra variable aleatoria. Esta información es relevante para construir un árbol de decisión, elemento relevante para la implementación de modelos como un "random forest", uno de los algoritmos utilizados en este estudio.

Table 3. Resultados de information gain ordenados de mayor a menor

	Coefficient
Timestamp	1.096179
AccY	0.042545
AccX	0.021876
GyroZ	0.013497
AccZ	0.006019
GyroX	0.000000
GyroY	0.000000

Como se puede observar en la tabla anterior, la variables "Timestamp" presenta un resultado superior al resto de las variables. Es tentador utilizar esta variable como la única variable para predecir pero esto solo son los resultados para los datos de entrenamiento y regresando a la sección de "Balance de datos", estos cuentan con la peculiaridad de estar ordenados por rangos mientras que este no es el caso en los datos de prueba en donde las clases no presentan este patrón. Intuitivamente, "Timestamp" no debería ser una variable que se debería de tomar en cuenta ya que no describe cambios bruscos de velocidad o rotación como el

resto de las variables. Es muy probable que se necesite remover esta categoría del conjunto de datos para reducir la major cantidad de ruido al crear los modelos. Sin embargo, variables como aceleración en X y Y junto a giroscopio en Z son las siguientes mejores opciones para higienizar el cúmulo de datos ya que aceleración en Z y giroscopio en X y Y dan resultados por debajo del resto.

2.4 Creacion de nuevos datos (Feature training)

Con los datos que se tenían de aceleración pensamos en crear nuevos features que serían los de diferencia de aceleración, ya que tu no quieres saber ne si que aceleracion tenía en ese punto si no que tanto cambia de punto a punto, para saber que tipo de movimiento la persona hace, como se comporta, va rapido y luego frena fuerte y así.

2.5 Normalización de datos

La normalización es una parte importante para la ciencia de datos y la etapa exploratoria de una investigación. La normalización permite manipular y escalar el rango de datos a una escala fija, en la mayoría de los casos de 0.0 a 0.1. Para este proyecto es necesario aplicar esta técnica ya que los datos dados por el acelerómetro y giroscopio están en diferentes unidades. Realizar la normalización de estas variables va ha permitir obtener otro tipo de resultados analíticos y mejores modelos. En adición a calcular los datos normalizados se decidió multiplicar por mil para hacer crecer la escala y reducir la pérdida de datos en posteriores operaciones y modificaciones a la colección de datos.

Fórmula para normalización:

$$df_{norm} = [(df - min(df))/(max(df) - min(df))] * 1000$$

2.6 Correlación de datos

Fig. 3. Mapa de calor que representa la correlación entre cada una de las variables.

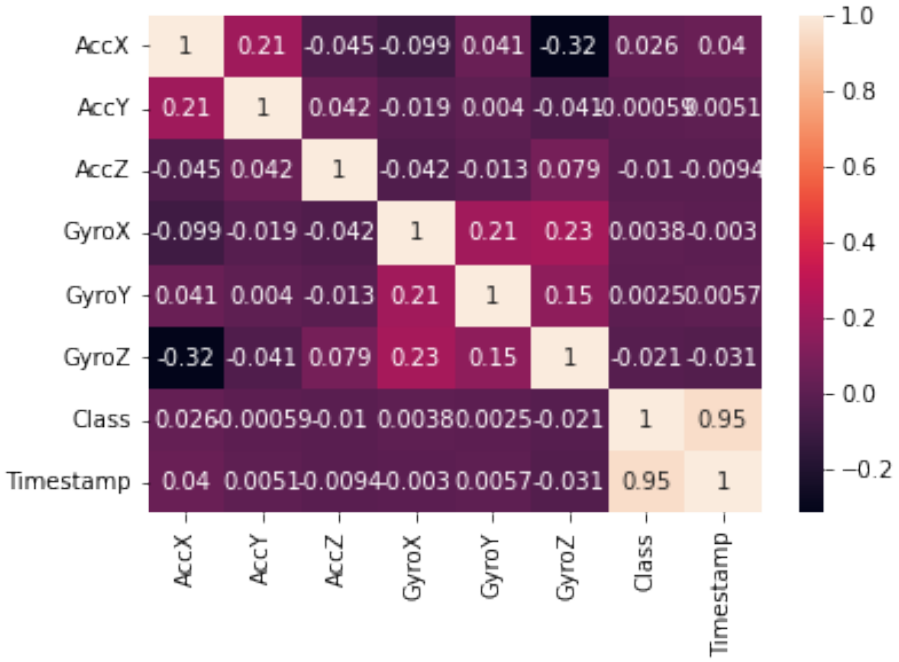


Fig. 4. Pairplot (dispersión e histograma) de variables de aceleración normalizados

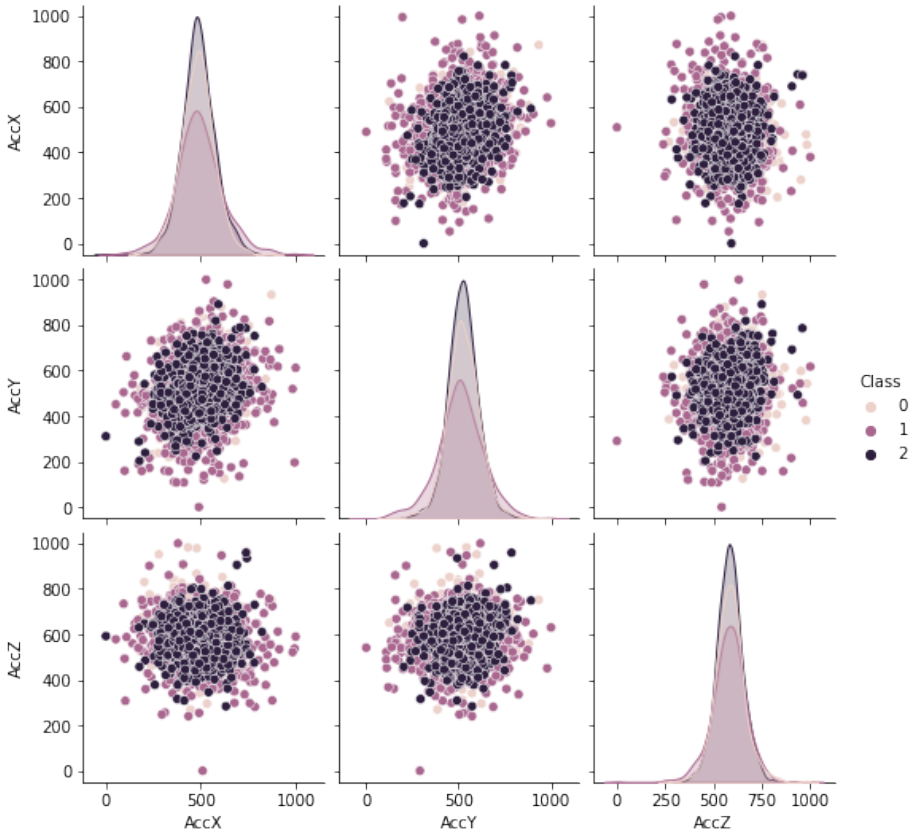
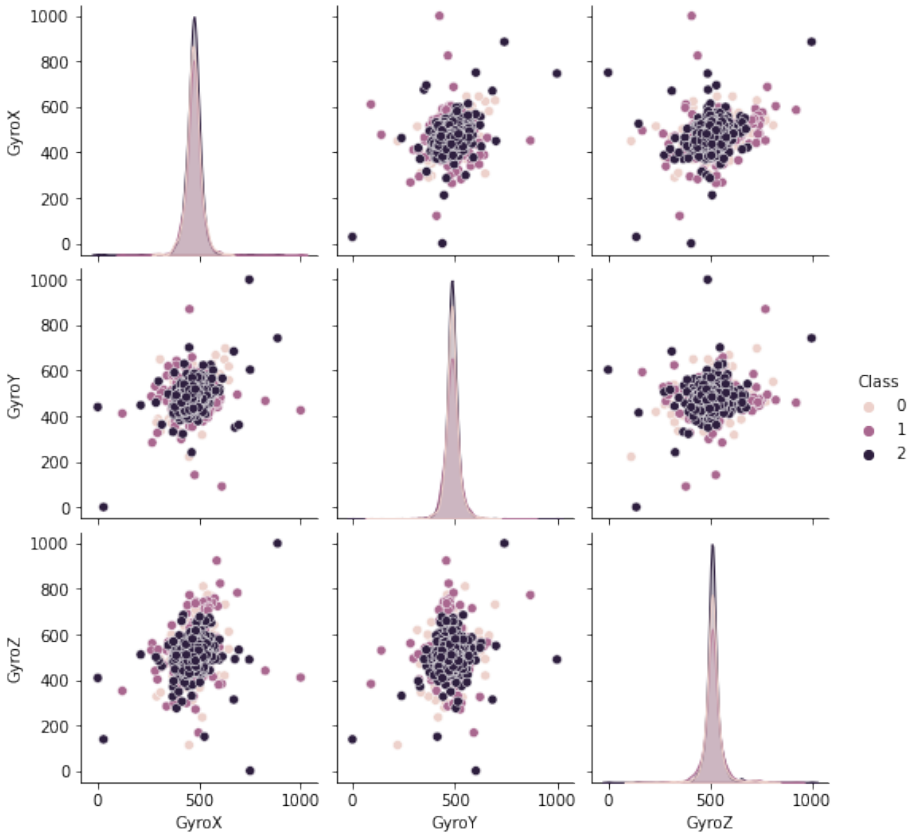


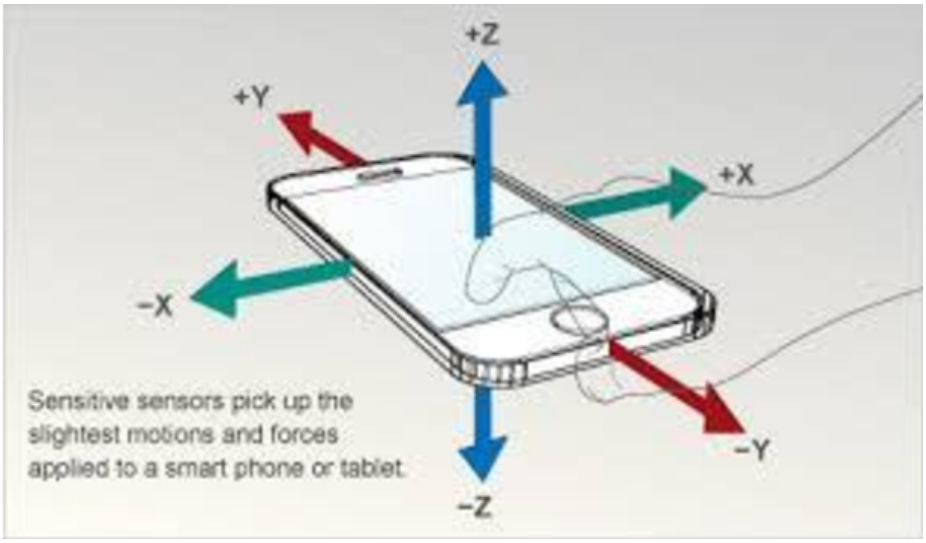
Fig. 5. Pairplot (dispersión e histograma) de variables de giroscopio normalizados



Una vez ya normalizados los datos, el siguiente paso es calcular la correlación entre las variables para escoger las que tengan una cantidad mayor para construir los modelos. Idealmente y realísticamente, se busca recibir una correlación alta positiva o negativa (mayor a 0.7) entre las variables de aceleración y giroscopio con respecto a la variable clase. No obstante, ninguna cumple con lo esperado, de hecho, la correlación entre estas variables es casi nula. Observando los diagramas de dispersión anteriores, los datos a través de todas las variables seleccionadas muestran estar empalmados con excepción de un par de datos atípicos que pertenecen principalmente a las clases de comportamiento normal y agresivo. Es necesario realizar más operaciones de preparación para reducir esta tendencia y establecer agrupaciones de datos más claros para la etapa de implementación de modelos.

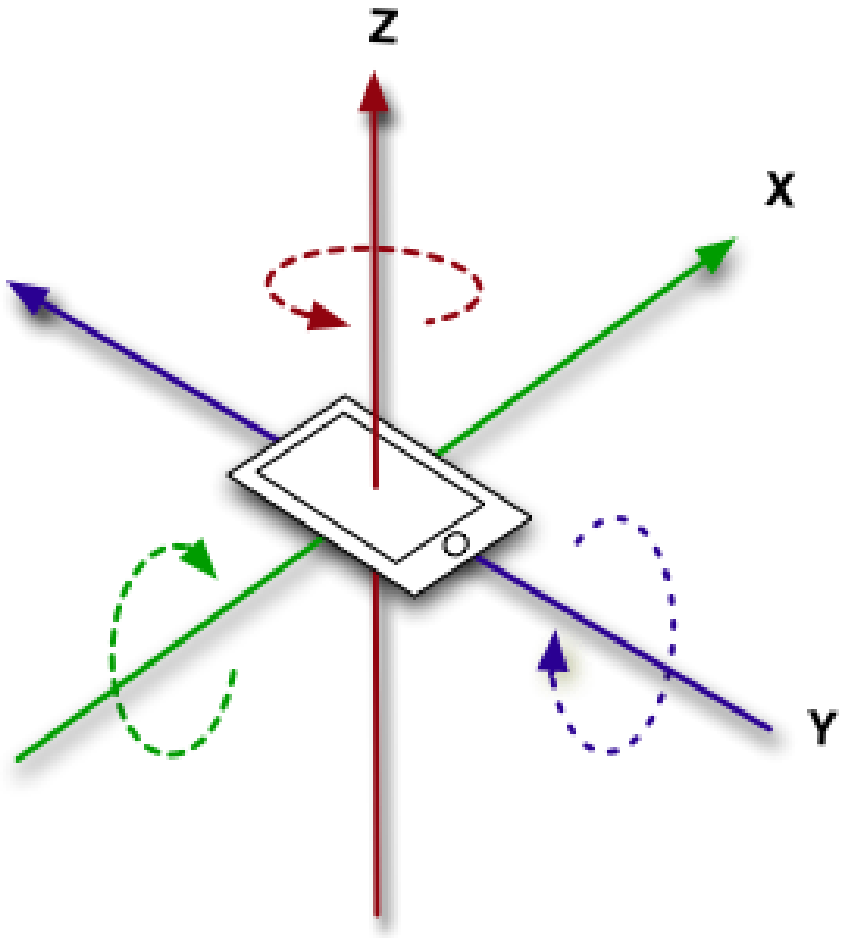
2.7 Información sobre acelerómetro y giroscopio del dispositivo

Fig. 6. Diagrama de los ejes del dispositivo para medir aceleración



Revisando el mapa de calor de correlación, la prueba ANOVA e information gain, es posible llegar a conclusión que la aceleración en el eje Z es una variable desechable que se puede remover para crear los modelos de clasificación. La aceleración en ese eje es muy probable que represente movimiento vertical de arriba hacia abajo (asumiendo que se encuentra como se observa en la figura anterior). Se especula que lo único en lo que probablemente ayudaría es en comprobar cómo se comporta una persona al interactuar con obstáculos en la calle como baches y topes.

Fig. 7. Diagrama de los ejes del dispositivo para medir rotación

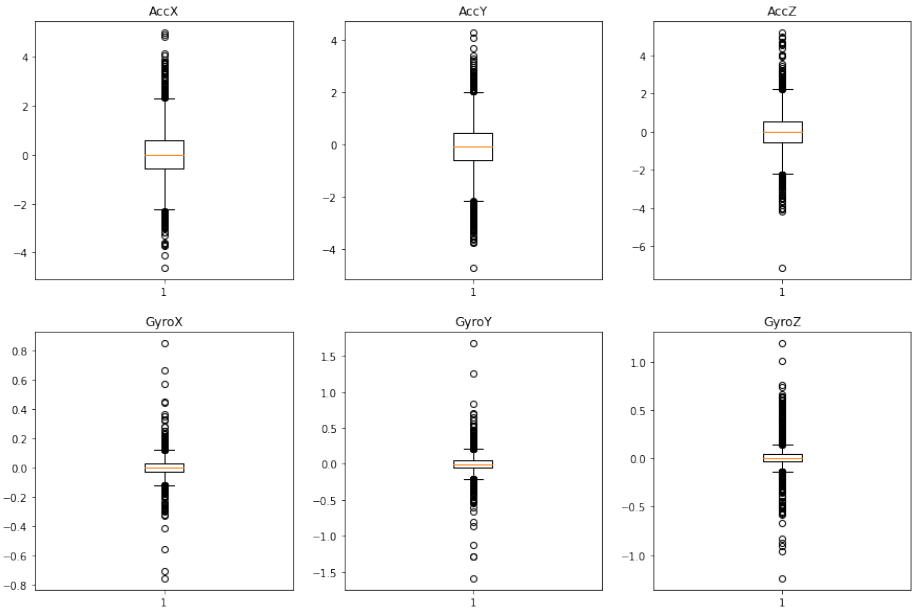


De igual manera, el mapa de calor, test de ANOVA e information gain reveló información importante sobre la influencia que posiblemente van a tener las variables de giroscopio sobre la efectividad y precisión de los modelos. En particular, los ejes X y Y del giroscopio son los que tienen menos correlación con la variable clase y los que describen muy poco del comportamiento del conductor al revisar el diagrama del giroscopio. Por lo contrario, la variable de giroscopio en el eje Z semeja ser una buena opción para construir los modelos ya que parece indicar vueltas del volante y no cambios pequeños de inclinación como las otras dos variables.

2.8 Datos atípicos

Otro aspecto considerable a tener en cuenta son los datos atípicos de las variables numéricas. Si bien, a través de todas las variables parece que existe hegemonía entre ellas, una cantidad considerable de datos fuera de una zona periférica que se muestra en los diagramas de dispersión. Para revisar esto se realizaron varios diagramas de caja y bigote para visualizar datos atípicos y un conteo de la cantidad de datos atípicos por clase. El diagrama de caja y bigotes muestra la distribución de datos por cuartiles, mínimos, máximos y valores atípicos.

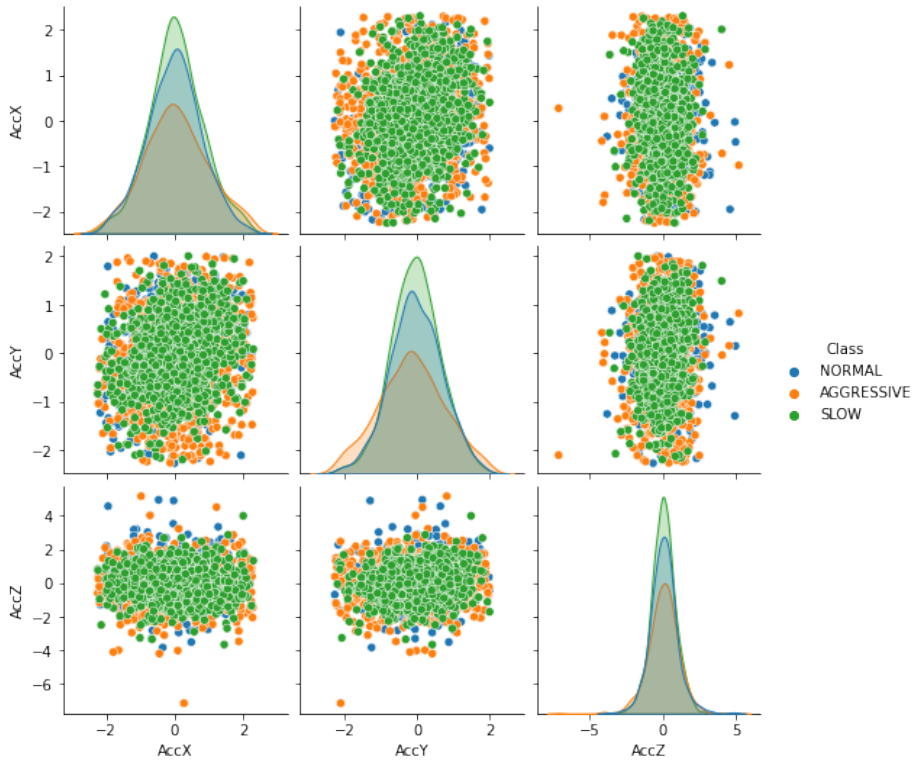
Fig. 8. Datos atípicos por variable de aceleración y giroscopio



Para sacar los datos atípicos es necesario ordenar los datos de forma ascendente y calcular los cuartiles. Su posición está dada por Q = k(n+1)/4 para conjuntos de datos impares donde k es el número del cuartil y n es el número de observaciones. Por último, obtener el límite superior $L_i = Q_1 + 1.5 * (Q_3 - Q_1)$ y limite inferior $L_s = Q_1 - 1.5 * (Q_3 - Q_1)$ para a partir de esos límites contar el total de valores sobre o debajo de ellos.

No obstante, intentar quitar valores atípicos solo empeora la colección de datos. Tan solo remover los datos atípicos en base de las variables X y Y en aceleración reduce la cantidad de observaciones de la clase “Slow” de 1331 a 1301 registros, de 1200 a 1154 registros para la clase “Normal” y de 1113 a 972 para la clase “Aggressive”. Además, si se observa el pairplot a continuación en base a las variables de aceleración, la mayoría de los datos son aún más homogéneos que antes.

Fig. 9. Pairplot (dispersión e histograma) de variables del acelerómetro sin valores atípicos en ejes X y Y



Está claro que los datos atípicos son clave para el entrenamiento de los modelos ya que en su mayoría representan comportamientos principalmente agresivos y ayudan a diferenciar un poco más las clases entre sí.

2.9 Modificaciones a la colección de datos original (ETL)

Como parte de uno de los entregables del proyecto y para facilitar el flujo de trabajo, se realizó un ETL (Extract, Transform, Load). Con este script es posible automatizar el proceso de modificar un dataset de entrada en el caso de que nuevas observaciones lleguen para ser analizadas, entrenadas y probadas por los modelos con el menor número de acciones adicionales. El resultado de este proceso regresa las dos colecciones de datos (de entrenamiento y prueba) con la composición adecuada para los modelos.

El ETL tiene la opción para remover columnas seleccionadas, valores negativos, calcular diferencia entre la entrada anterior y ajustar esas diferencias por cada

segundo ya que cada entrada en el dataset corresponde a medio segundo. En concreto se removieron variables como la aceleración en Z, giroscopio en X y Y y timestamp debido a que no como se discutió en partes anteriores tienen poca correlación con la variable predictora, un information gain bajo y en el caso del timestamp, es irrelevante al pronosticar comportamiento de manejo. Además, se calculó la diferencia entre cada observación de aceleración y giroscopio restante para obtener un conjunto de datos más robusto y que pudiera explicar el comportamiento de mejor manera en relación a cambios drásticos. Al final este último cambio solo mejoró de manera diminuta las calificaciones de los modelos.

3 Modelos

3.1 Implementación de modelos con Grid Search CV y Randomized Search CV

En el proceso de implementación de los modelos una parte importante a tomar en cuenta es el ajuste de los hiper parámetros que ofrece cada implementación del algoritmo. Los hiper parámetros son valores del modelo que necesitan ser ajustados para controlar la salida o el comportamiento del algoritmo utilizado para el modelado. Existen varios pasos para el ajuste de los hiper parámetros como escoger el método de validación cruzada y las métricas para evaluar el modelo.

La validación cruzada es un procedimiento de remuestreo utilizado para evaluar los modelos de aprendizaje automático. Este método tiene un único parámetro que se refiere al número de particiones en las que se va a dividir la muestra de datos dada. Los datos se dividen en conjuntos de entrenamiento, validación y prueba para evitar sobreajuste de datos. Cada vez que el modelo se ajusta a los datos de entrenamiento se evalúa con los datos de prueba y la media de la puntuación de la evaluación se utiliza para analizar el modelo global.

Existen dos técnicas bastante populares como Grid Search y Randomized Search para realizar el ajuste de hiper parámetros. Grid search es el más sencillo de los dos ya que prueba todas las permutaciones de los hiper parámetros sin embargo es computacionalmente costoso. Mientras tanto Randomized Search utiliza una porción aleatoria de todas las permutaciones, es menos costosa que el Grid Search pero no te garantiza los mejores hiper parámetros.

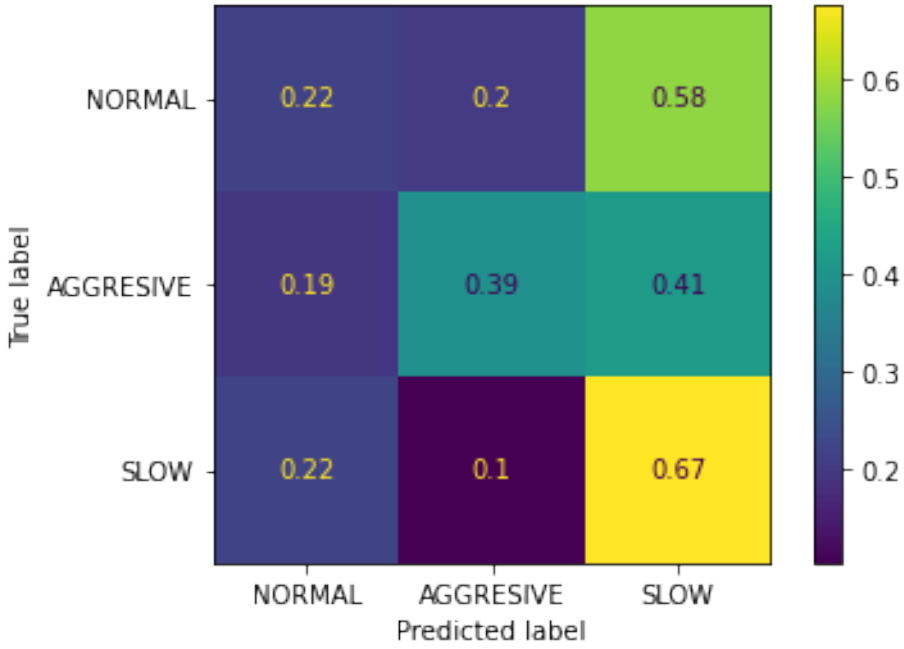
En los siguientes modelos se hizo uso de las dos técnicas. Grid Search para los algoritmos menos intensivos como regresión logística y knn. Random Forest y XG Boost fueron implementados utilizando Random Search ya que el tamaño de la colección de datos y la cantidad de hiper parámetros sobrepasan las capacidades del hardware donde fue probado.

3.2 XG Boost

El XG Boost es uno de los algoritmos supervisados de Machine Learning que más se usan en la actualidad. Esto se debe por su facilidad de implementación y sus buenos resultados. Este es una implementación del algoritmo de gradiente de árboles reforzado, cuyo algoritmo intenta predecir con precisión una variable objetivo combinando las estimaciones de un conjunto de modelos más simples y débiles. La base de este algoritmo es generar múltiples modelos secuenciales de predicción "sencilla" y cada nuevo modelo toma el resultado del modelo anterior, haciendo que los resultados obtenidos en cada iteración sean más robustos y por ende más exactos.

Resultados Calificación con conjunto de datos de prueba 45.2010%

Fig. 10. Matriz de confusión de XG Boost



La calificación de este modelo dio 45.2010% de precisión para el conjunto de datos de prueba. En la matriz de confusión se observa que principalmente clasifica correctamente la clase "Slow" ya que el modelo al crear agrupaciones, en estas predomina el tipo lento y el algoritmo no logra clasificar correctamente el resto de las clases dando a su vez una gran cantidad de falsos positivos.

3.3 KNN

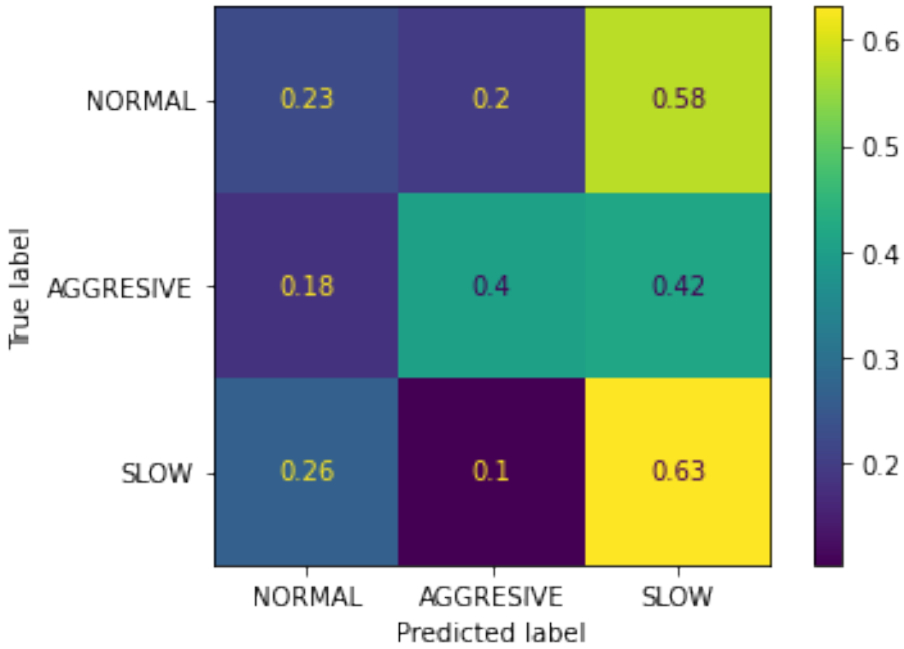
El algoritmo de k vecinos más cercanos, también conocido como KNN o k-NN, es un modelo de clasificación supervisado no paramétrico, utilizando la proximidad de los objetos o datos para realizar la predicción del grupo al que puede pertenecer un dato, partiendo de la suposición de encontrar puntos similares cerca uno del otro. Este algoritmo clasifica los grupos con etiquetas de las clases que se tienen en la base de datos y el dato a clasificar toma en cuenta la mayor cantidad de una misma clase cercanas a la ubicación de este, normalmente basándose en mayor a 50% pero variando dependiendo de la cantidad de clases y la cercanía de los conjuntos entre sí.

Para medir la distancia entre los puntos, existen diferentes fórmulas que usan vectores para calcular las distancias: Distancia euclidiana, Distancia Manhattan, Distancia minkowski.

Este algoritmo a su vez usa la constante K para determinar el rango en el que se hará la la comparación de un punto de consulta específico, siendo esta un hiper parámetro ya que es el único dato que él se debe definir antes de realizar el modelo y así pudiendo variar las constantes y puede llevar a un ajuste excesivo o insuficiente para poder predecir la clase

Resultados Calificación con conjunto de datos de prueba 44.0012%

Fig. 11. Matriz de confusión de KNN



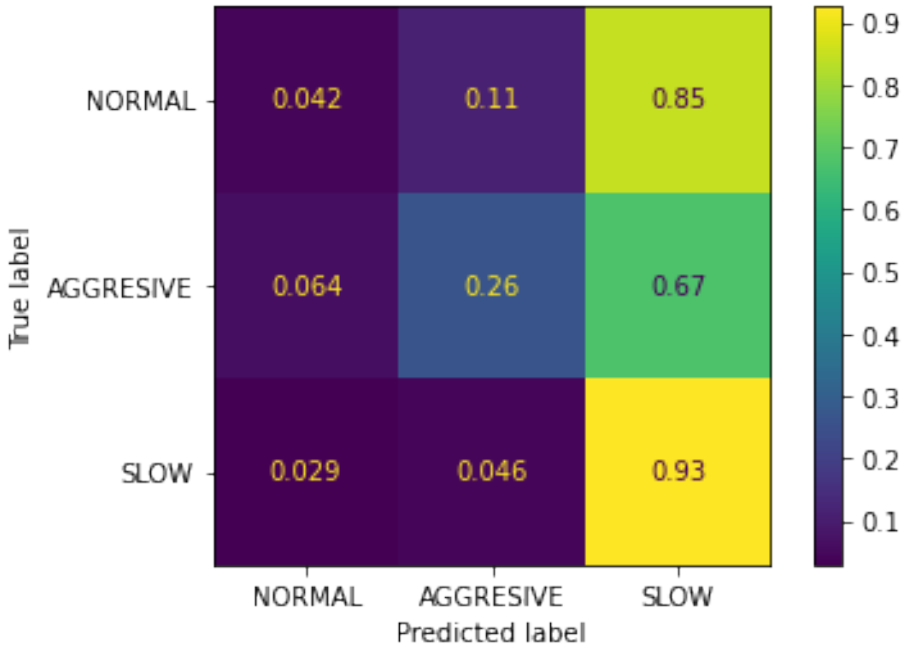
La calificación de este modelo dio 44.0012% de precisión para el conjunto de datos de prueba. Por como funciona el algoritmo, se obtiene mayor precisión en la clase "Slow" ya que al generar un área circular en el conjunto de datos, al haber mayor número de datos de la clase lento, predice con mayor consistencia esta clase y a su vez se obtienen una cantidad alta de falsos negativos.

3.4 Logistic Regression

El modelo de regresión logística es similar a un modelo de regresión lineal, donde se intente predecir un valor, la diferencia está que este modelo predice la clasificación de los datos basado en diferentes clases que se le da al modelo, los coeficientes que se obtienen de la regresión logística pueden utilizarse para estimar la probabilidad de cada variable independiente del modelo

Resultados Calificación con conjunto de datos de prueba 46.4656

Fig. 12. Matriz de confusión de Regresión logística



La calificación de este modelo dio 46.4656% de precisión para el conjunto de datos de prueba. Este modelo al igual que los anteriores, predice con mayor exactitud la clase lento, esto por haber más datos de esta clase y el modelo genera una línea de separación en los datos, al estar todos aglomerados no detecta correctamente las demás clases.

3.5 Random Forest

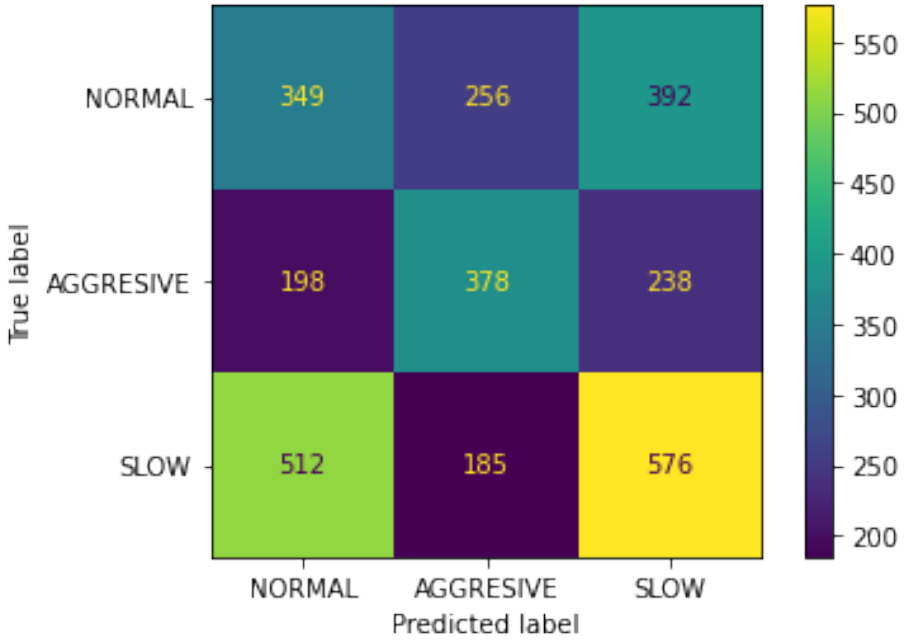
El algoritmo Bosque aleatorio utiliza árboles de decisión que se crean a partir de diferentes secciones de la base de datos para crear diferentes árboles de decisión, sin llegar a ver todos los datos de entrenamiento, para que estos puedan predecir el valor que se está buscando, al combinar sus resultados, unos errores se compensan con otros y se obtiene una predicción que generaliza mejor.

El algoritmo Bosque aleatorio utiliza árboles de decisión que se crean a partir de diferentes secciones de la base de datos para crear diferentes árboles de decisión, sin llegar a ver todos los datos de entrenamiento, para que estos puedan predecir el valor que se está buscando, al combinar sus resultados, unos errores se compensan con otros y se obtiene una predicción que generaliza mejor.

- Nodos de decisión: Tienen una condición al principio y tienen más nodos debajo de ellos.
- Nodos de predicción: No tienen ninguna condición ni nodos debajo de ellos.

Fig. 13. Matriz de confusión de Random Forest

Resultados Calificación con conjunto de datos de prueba 42.2503



La calificación de este modelo dio 42.2503% de precisión para el conjunto de datos de prueba. Observando la matriz de confusión es claro que existe un pronóstico que favorece las clases de "Slow" y "Aggressive" a diferencia de modelos anteriores que solo predicen "Slow" de forma efectiva. No obstante, reduce la precisión de predicción de la clase "Slow" considerablemente comparado con los demás modelos ya que el algoritmo al crear agrupaciones y comparar el resultado de todos los árboles, encuentra la mejor permutación de hiper parámetros para el resultado óptimo.

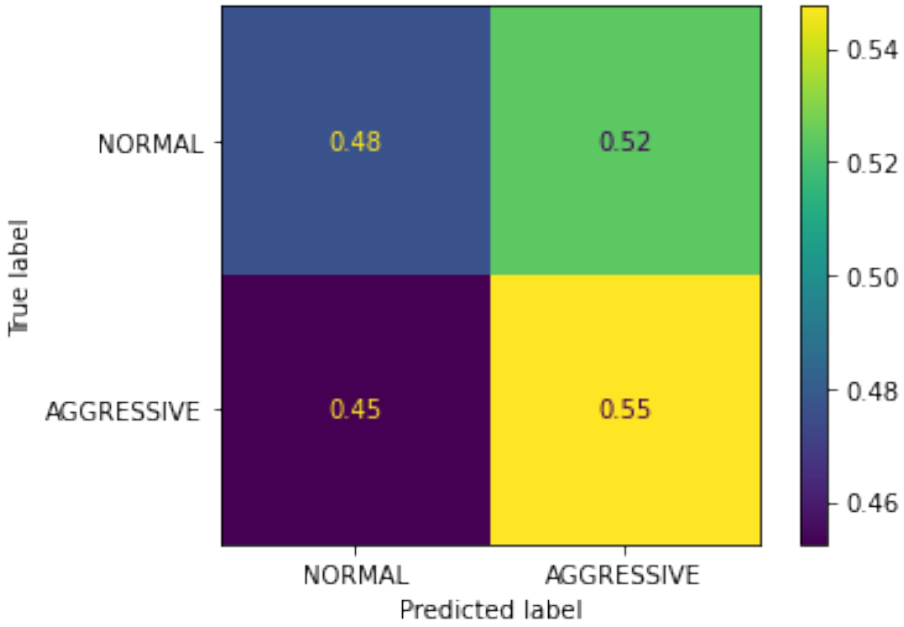
3.6 Modelos Binarios

Los modelos de clasificación binarios utilizan los mismo algoritmos que los modelos regulares, más sin embargo, sólo maneja dos clases de las que se desea predecir, para que resulte en un error menor en la predicción de estas dos. La clase que se decidió remover fue la clase "Slow" ya que una gran parte de los modelos parece favorecer este tipo de comportamiento sobre los otros. También se decidió quitar esta clase con el objetivo de buscar obtener mejores resultados para predecir "Aggressive" ya que es la clase más importante a estimar con mayor precisión.

3.7 Binary XG Boost

Resultados Calificación con conjunto de datos de prueba 51.6299%

Fig. 14. Matriz de confusión de Binary XG Boost

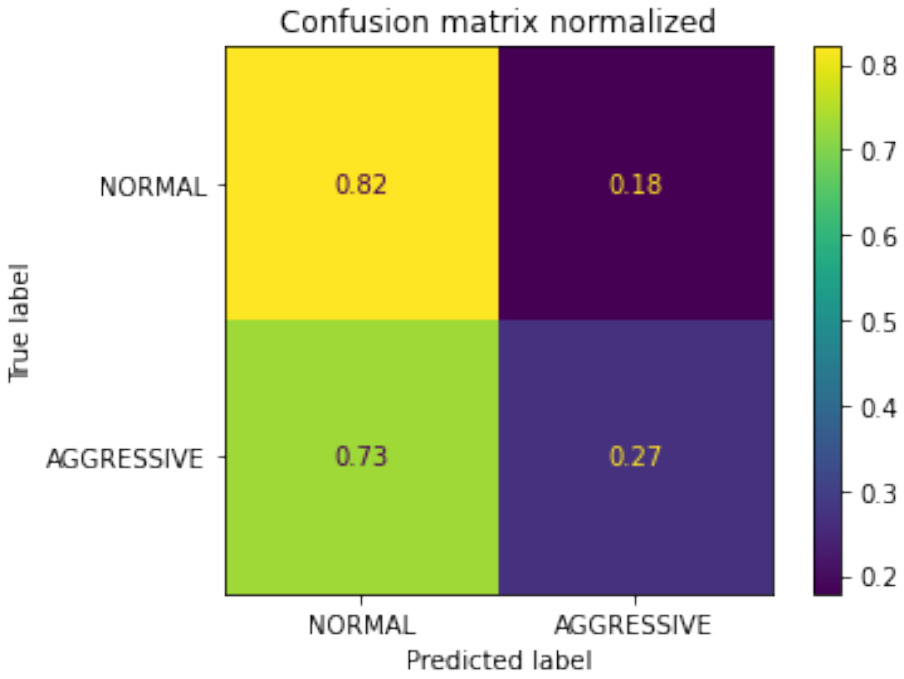


La calificación de este modelo dio 51.6299% de precisión para el conjunto de datos de prueba. Naturalmente este modelo tiene puntuación sobre su variante de tres clases, ya que existe menos ruido que el algoritmo puede interpretar de forma errónea. No obstante, el incremento de precisión al clasificar valores sigue siendo bastante bajo.

3.8 Binary KNN

Resultados Calificación con conjunto de datos de prueba 57.3163%

Fig. 15. Matriz de confusión de Binary KNN

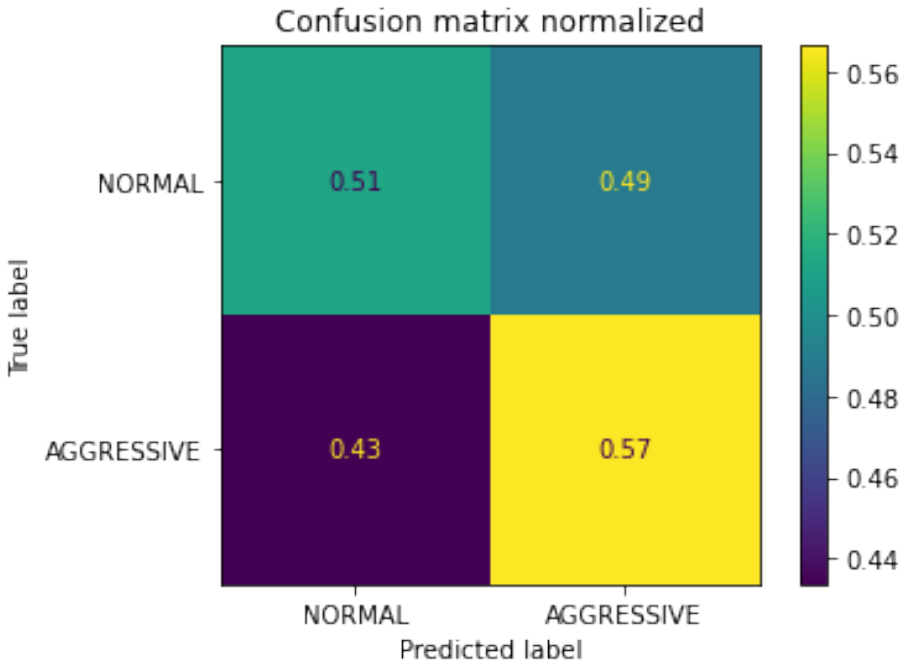


La calificación de este modelo dio 57.3163% de precisión para el conjunto de datos de prueba. Ocurre algo similar con este modelo de dos clases, sin embargo a diferencia del XG Boost binario, este modelo parece inclinarse más hacia la clase normal probablemente porque ya que no existen observaciones de la clase lenta en el conjunto de datos. Ahora los que abarcan la mayor concentración en el área central son los datos de clase normal.

3.9 Binary Random Forest

Resultados Calificación con conjunto de datos de prueba 51.6299%

Fig. 16. Matriz de confusión de Binary Random Forest



La calificación de este modelo dio 51.6299% de precisión para el conjunto de datos de prueba. De nuevo la tendencia de los modelos binarios continua en este caso favoreciendo la categoría de clase agresiva pero la precisión de pronóstico es similar a un volado.

4 Resultados

Posterior a hacer todos los modelos se vio que tan solo no se puede mejorar en si los resultados de predicción para cada clase, siempre la suma de los porcentajes de precisión de cada clase en total da alrededor de 120%, esto indica que si se balancean las clases para que tengan la misma precisión solo se tendría 60% en total de precisión, es decir, un poco mejor que un volado de precisión.

Después de ver estos resultados se llegó a la conclusión que o es muy difícil poder identificar un tipo de manejo o que tan solo el experimento estuvo mal diseñado, ya que revisando las gráficas y matrices de confusión en los análisis, se puede ver que los tres tipos de manejos no tienen muchas diferencias, los datos estas concentrados en los mismos rangos, entonces es muy difícil poder diferenciar de un tipo de manejo del otro. Y después viendo las velocidades calculadas se observó que la velocidad máxima era aproximadamente 14 km/h y eso es una muy baja velocidad para poder sacar datos extremos de cada estilo de manejo.

Entonces para poder tener un mejor dataset y sacar mejores conclusiones y talvez poder sacar un buen modelo se proponen las siguientes dos ideas.

4.1 Incrementar la cantidad de gente para el experimento

En este experimento solo hay un piloto y un copiloto que hacen todas las pruebas, se plantea la idea de que este experimento se haga de manera supervisada con al menos 20 pilotos para así obtener una mayor variación en los datos y poder encontrar más diferencias en los tipos de manejos, y que también que se haga mejor el experimento controlado, que se tengan mayores velocidad y más vueltas para poder encontrar más diferencias.

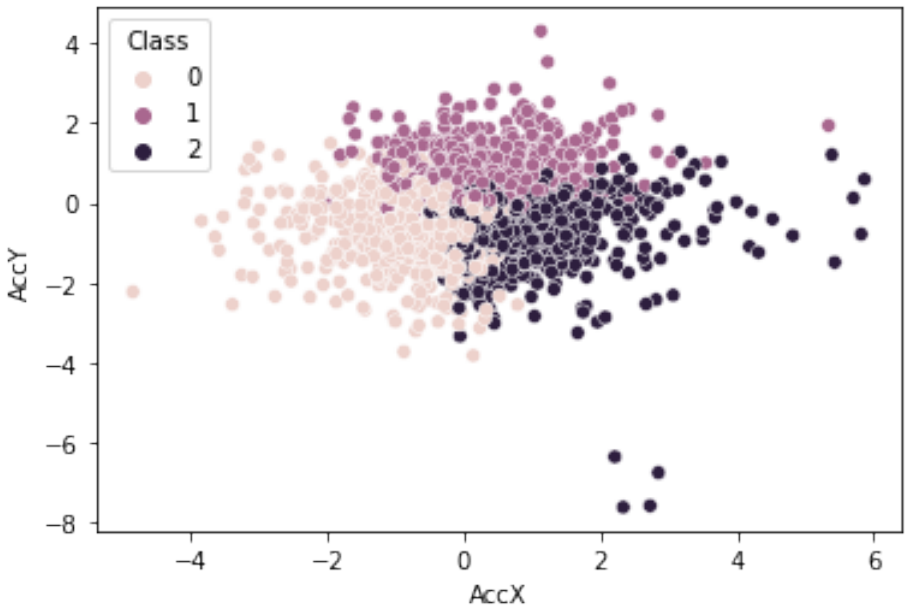
4.2 Entrenamiento no supervisado

La otra idea sería recolectar datos de muchos celulares, ya que tendrán datos más generales de cómo se comporta la gente en un día cotidiano.

K-means Para dar pruebas de como pudiera servir el entrenamiento no supervisado, se realizaron pruebas con el modelo de K-means. K-means es un algoritmo de clasificación no supervisado que agrupa conjuntos de datos basándose en sus características, estas agrupaciones se realizan minimizando la suma de distancias entre los diferentes grupos y sus centroides, por cada conjunto de datos, se selecciona un posible centroide, los objetos son asignados a un centroide dependiendo de la distancia en la que se encuentra, seleccionando el más cercano y se va actualizando la posición del centroide tomando el promedio de la posición de los objetos pertenecientes a ese grupo, estos últimos pasos se realizan hasta que el centroide deje de cambiar su posición con respecto al promedio o hasta que su cambio de posición esté debajo de un umbral.

El algoritmo de K-means resuelve problemas de optimización o minimizar, siendo la función a optimizar la suma de las distancias cuadráticas de cada objeto al centroide. Los objetos son vectores reales en los cuales el algoritmo construye grupos dependiendo de las características de los valores que tienen estos vectores.

Fig. 17. Gráfica de agrupamientos creada por el modelo



El resultado mostrado en la gráfica del modelo K means, muestra cómo sería una alternativa para el agrupamiento de los datos. Estos tres clusters son dados sin etiqueta y el algoritmo no genera ninguna interpretación y en este caso se decidió asignar el comportamiento del conductor basado en los límites en los que se encuentra cada conjunto de datos, poniendo en orden de izquierda a derecha lento, normal y agresivo.

Fig. 18. Gráfica de agrupamientos en base a clase para la variable de aceleración en X

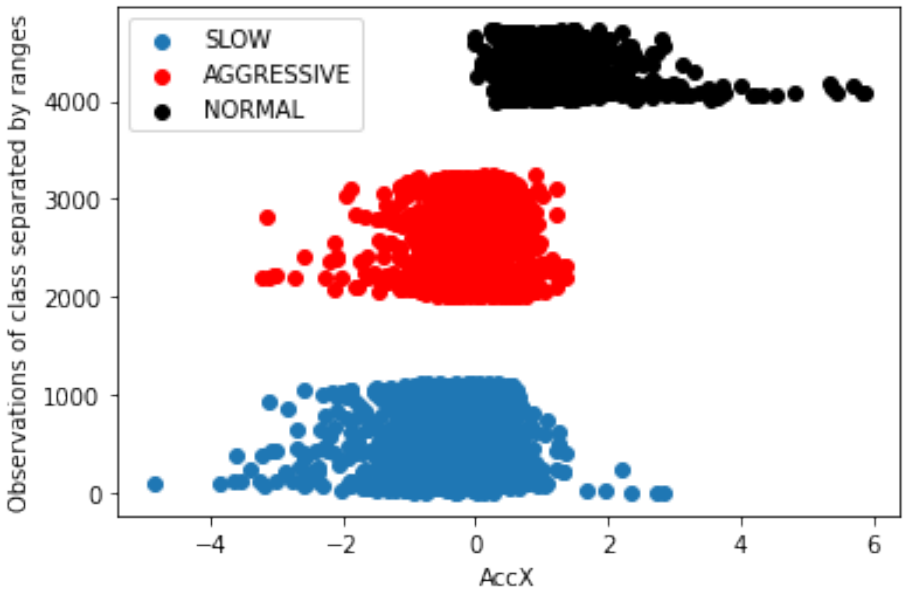
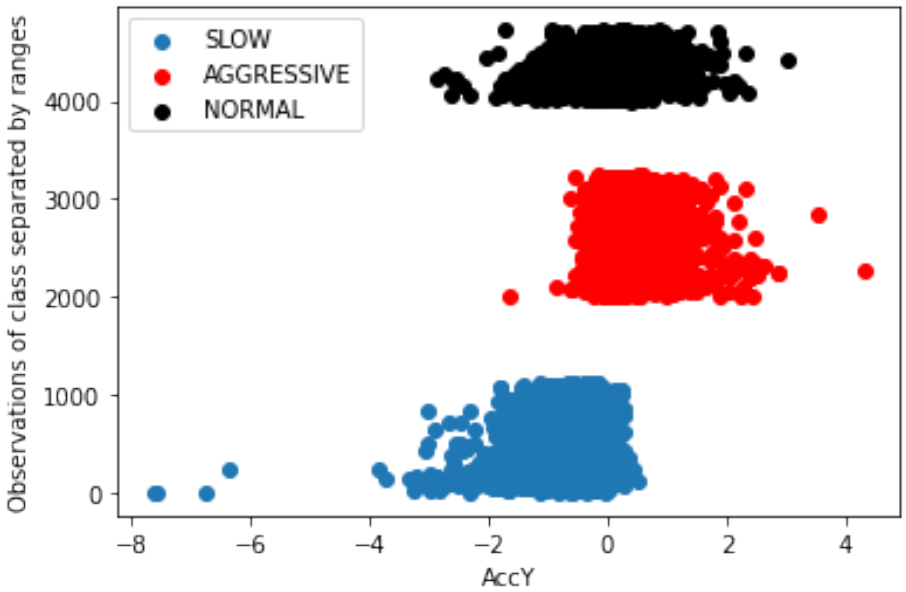


Fig. 19. Gráfica de agrupamientos en base a clase para la variable de aceleración en Y



No obstante el modelo no es perfecto ya que para las clases consideradas como lo dio el modelo se observa que en Aceleración en Y, designa que agresivo tiene

muy pocos negativos y pues un conductor agresivo va a frenar (Asumiendo que el celular estaba acostado viendo al frente Y seria hacia adelante y atrás) también al igual que cualquier otro tipo de conductor. Pero en las gráficas anteriores se puede ver que el modelo da mejores divisiones de los datos, así que existe la posibilidad de que un modelo no supervisado pudiera encontrar mejor las diferencias de un conductor agresivo a uno normal, pero esto es solo una hipótesis.

5 Conclusión

Después de haber hecho todos los modelos se verifico que hay discrepancias en los datos ya que es muy raro que todos los tipos de conductores solo tengan las aceleraciones en X y Y con diferentes promedios de la población, y que los datos están casi igual de concentrados en los mismo rangos de las variables. Así que de estos resultados se llegó a la conclusión que probablemente el experimento no fue bien implementado y se propusieron un par de ideas para expandir la cantidad de conductores que se usan para registrar los datos o usar entrenamiento no supervisado con muchos datos de múltiples conductores para talvez poder crear un mejor modelo.

References

1. Morde, V., 2022. XGBoost Algorithm: Long May She Reign!. [online] Medium. Available at: https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d [Accessed 15 September 2022].
2. Bosco Mendoza, J., 2022. Tutorial: XGBoost en Python. [online] Medium. Available at: https://medium.com/@jboscomendoza/tutorial-xgboost-en-python-53e48fc58f73 [Accessed 15 September 2022].
3. Unioviedo.es. 2022. El algoritmo k-means aplicado a clasificación y procesamiento de imágenes. [online] Available at: https://www.unioviedo.es/compnum/laboratorios_py/kmeans/kmeans.html [Accessed 15 September 2022].
4. Ibm.com. 2022. ¿Qué es el algoritmo de k vecinos más cercanos? | IBM. [online] Available at: https://www.ibm.com/mx-es/topics/knn [Accessed 15 September 2022].
5. Ibm.com. 2022. Regresión Logística | IBM. [online] Available at: https://www.ibm.com/docs/es/spss-statistics/saas?topic=regression-logistic [Accessed 15 September 2022].
6. Heras, J., 2022. Random Forest (Bosque Aleatorio): combinando árboles - IArtificial.net. [online] IArtificial.net. Available at: https://www.iartificial.net/random-forest-bosque-aleatorio/ [Accessed 15 September 2022].