Saturdays.Ai - 4ta Edición Guadalajara

Infracciones GDL

Equipo Morado

Mentor: Alva, Abraham

Fecha: Noviembre de 2022

Autor(es): Flores, Oscar Alfonso
Buenrostro, Joel
Bello, Alfredo

Índice

1) Descripción del Problema (Comprensión del negocio)

1.1 Objetivos del Negocio: objetivos del negocio, beneficios esperados, metas y criterios de éxito.

El principal objetivo del negocio es promover más lugares para estacionarse en las calles del municipio de Guadalajara, así como evitar el levantamiento de infracciones, ya que en 2021, se registraron más de 156 mil infracciones de 11 tipos diferentes:

- **Tipo 1**: Omitir la tarifa del parquímetro.
- **Tipo 2**: Invadir dos cajones o más de estacionamiento.
- **Tipo 3**: Obstruir una cochera.
- **Tipo 4**: Estacionarse en una intersección o en línea amarilla.
- **Tipo 5**: Estacionarse en lugares exclusivos, bomberos, policía, servicios médicos o personas con discapacidad.
- **Tipo 6**: Cordón o batería.
- **Tipo 7**: Vehículo inmovilizado.
- **Tipo 8**: Por insultar o agredir verbalmente a un oficial vial.
- **Tipo 9**: Por abandono de vehículo.
- **Tipo 10**: Exceder tiempo en los espacios de carga y descarga.
- **Tipo 11**: Agredir físicamente a un oficial vial.

Donde el tipo 1, tipo 4 y tipo 5 son las más comunes, con un 41%, 31% y 25% respectivamente, sumando así el 97% del total de las infracciones. Identificando esta problemática, se buscará crear una mejor educación vial de los conductores y ayudar a los transeúntes para que puedan transitar por las vías designadas, generando una mejor conciencia y cultura vial en el país, además de evaluar las tendencias en el cumplimiento vial y su vinculación con las estrategias en políticas públicas.

1.2 Evaluación de situación actual.

El proceso de levantamiento de infracciones de movilidad del Municipio de Guadalajara se lleva a cabo mediante una aplicación móvil la cual está conectada vía internet con la base de datos del Ayuntamiento, donde dicha aplicación tiene como función levantar una infracción a los vehículos violando alguna de la normativa de movilidad del municipio. De igual forma, la aplicación genera un ticket o infracción con los detalles del delito vial y posterior a este proceso de levantamiento, las infracciones son enviadas al sistema web dónde solo tienen acceso el personal de la dependencia de Movilidad. Con lo anterior, se tuvo el consentimiento de parte de la misma dependencia para la obtención de la base de datos, donde se omitieron datos sensibles, que funcionó como entrenamiento para el modelo de Machine Learning.

1.3 Solución propuesta: Enfoque a utilizar para resolver el problema (aprendizaje automático).

El enfoque que se utilizó para este problema fue crear un modelo de predicción basado en la información histórica y geolocalizable de las infracciones registradas en el último año, donde utilizando un Dashboard, se pueda observar el histórico de estas infracciones a lo largo de la ciudad, para que los automovilistas puedan prevenir ser infraccionados.

1.4 Hipótesis.

La falta de espacios de estacionamiento y la falta de conocimiento de los parquímetros digitales causa el gran número de infracciones registradas año tras año en la Zona Metropolitana de Guadalajara.

1.5 Objetivo del proyecto: Propósito.

Se busca crear una mejor educación vial de los conductores y ayudar a los transeúntes para que puedan transitar por las vías designadas, generando una mejor conciencia y cultura vial en el país, además de evaluar las tendencias en el cumplimiento vial y su vinculación con las estrategias en políticas públicas, todo esto, enfocado a buscar reducir el número de infracciones que cada año aumenta más y más.

1.6 Justificación.

La ZMG recibe alrededor de 1,000 infracciones al día, de las cuales solo alrededor del 10% son pagadas cada año. Este modelo está enfocado en dos principales metas: Una para el ciudadano conductor, donde a partir del Estado de su placa, Modelo de su carro, mes y hora donde se estacionará, y latitud y longitud del lugar; a partir de esto, se podrá hacer una valoración si entra en perfil estadístico de infracciones en la ciudad y estereotipos de conductores que el modelo aprendió. Por otro lado, beneficiará a la Entidad Pública, ya que a partir de este modelo se puede tener una proyección más certera, basada en el histórico, sobre las infracciones a futuro. A partir de estas predicciones, se pueden crear o proponer políticas públicas o programas sociales que incentiven una mejor cultura vial en la ciudad. A partir del análisis de estos datos podemos determinar cuáles serían las zonas más eficientes dónde agregar cajones de estacionamiento parkimovil o estacionamientos públicos.

1.7 Mercado potencial e identificación de clientes/consumidores y usuarios.

Existen dos mercados potenciales muy fuertes dentro de este proyecto, el primero son los conductores en busca de un lugar para estacionarse, ya que suelen recurrir a no pagar una tarifa de parquímetro, o a utilizar espacios que no están designados para el estacionamiento de vehículos, como lo son líneas amarillas, rampas o lugares para discapacitados, espacios para bomberos, policía o servicios médicos, cayendo así en estas infracciones antes descritas, por lo que el poder obtener dichos lugares donde normalmente han sido infraccionados otros conductores anteriormente, ayudará a los futuros conductores a evitar estas infracciones. Por otro lado, el otro mercado potencial será la Secretaría de Movilidad, que podrá obtener un resultado de cuales son las zonas, fechas y horas con más infracciones y poder proponer soluciones a dicho problema vial.

1.8 Plan de actividades del proyecto.

El desarrollo del proyecto se llevó a cabo durante 7 semanas de trabajo, donde se comenzó con una lluvia de ideas de diferentes problemáticas que pueden ser solucionadas utilizando AI y ML, seguido por la recolección de datos, limpieza de los mismos, pruebas de diferentes modelos en este caso de predicción, para finalmente realizar el deployment y presentación de la solución encontrada al problema. Las semanas de trabajo fueron divididas de la siguiente manera:

Semana	Avances de proyecto
Semana 1 9 de octubre - 15 de octubre	● Lluvia de ideas sobre la problemática social a solucionar. ● Asignar roles de trabajo para el equipo. ● Creación del plan de actividades del proyecto.
Semana 2 16 de octubre - 22 de octubre	● Obtención de la base de datos. ● Presentación inicial de la solución para la problemática elegida.
Semana 3 23 de octubre - 29 de octubre	● Comprensión de la problemática. ● Redacción de la ficha de trabajo.
Semana 4 30 de octubre - 5 de noviembre	● Limpieza, comprensión y preparación de los datos obtenidos. ● Separación de la base de datos en entrenamiento y prueba.
Semana 5 6 de noviembre - 12 de noviembre	● Creación de los modelos de clasificación para la solución de la problemática.
Semana 6 13 de noviembre - 19 de noviembre	● Creación del documento rector. ● Despliegue del modelo. ● Creación de la presentación final. ● Pitch Day: 19 de noviembre. Ensayo interno del Demo Day.
Semana 7 20 de noviembre - 26 de noviembre	● Correcciones finales del trabajo ● Demo Day: 26 de noviembre. Presentación final del proyecto

Tabla 1. Plan de actividades semanal.

2) Comprensión de los datos

2.1 Descripción del set de datos (dimensiones) y sus variables, tipo de dato y origen (quitar columnas que no tienen datos).

Con la base de datos obtenida de las infracciones de la Zona Metropolitana de Guadalajara, se puede comenzar a realizar la exploración de la misma. Comenzando con las dimensiones de la misma, encontramos que la base de todas las infracciones de 2021 contiene 156,685 registros, mientras que la base con los registros de enero a agosto 2022 contiene 199,187 registros y ambas bases cuentan con 9 columnas.

	ESTADO	MARCA	CALLE	CRUCE	FECHA	HORA	INFRACCION	LATITUD	LONGITUD
17	JALISCO	MAZDA CX-3 5 PUERTAS (IMPORTADO)	duque de rivas	miguel blanco de Tejada	16/08/21	12:34:00	Clave 1 Omitir tarifa	20.67	-103.38
18	CIUDAD DE MEXICO	HONDA	Calle Miguel Lerdo de Tejada	duque de Rivas	16/08/21	12:38:00	Clave 1 Omitir tarifa	20.67	-103.38
19	JALISCO	MITSUBISHI OUTLANDER 4 PUERTAS IMP	Francisco de Quevedo	Miguel Lerdo de Tejada	16/08/21	12:42:00	Clave 1 Omitir tarifa	20.67	-103.38

Imagen 1. Muestra de la base de infracciones de 2021.

En ambas bases podemos encontrar las siguientes columnas:

1. **ESTADO**: Estado de la placa del automóvil infraccionado | Tipo String.
2. **MARCA**: Marca del automóvil infraccionado | Tipo String.
3. **CALLE**: Calle principal donde se realizó la infracción | Tipo String.
4. **CRUCE**: Cruce más cercano de la calle principal donde se realizó la infracción | Tipo String.
5. **FECHA**: Fecha de la infracción | Tipo Datetime.
6. **HORA**: Hola de la infracción | Tipo Datetime.
7. **LATITUD**: Latitud geográfica de la infracción | Tipo int.
8. **LONGITUD**: Longitud geográfica de la infracción | Tipo int.
9. **INFRACCIÓN**: Tipo de infracción cometida por el vehículo | Tipo String.

2.2 Exploración y calidad de datos.

La gran desventaja de los datos obtenidos, es que, al ser campos abiertos para el registro de las infracciones, existen demasiados valores únicos en las principales columnas como lo son **ESTADO**, **MARCA** o **INFRACCIÓN**. Esto puede observarse en la siguiente gráfica:

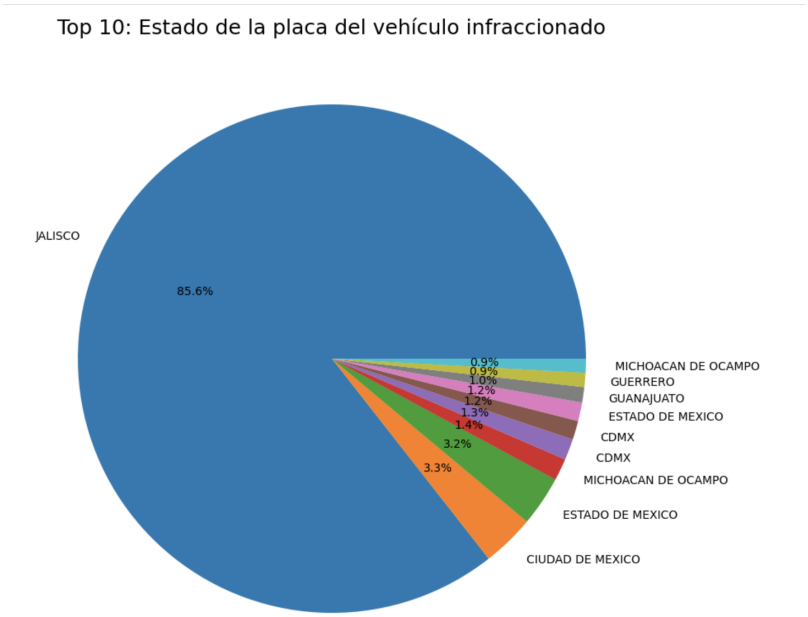


Gráfico 1. Gráfica de pastel representando los estados de las placas de los vehículos más infraccionados.

Se puede observar que, claramente el estado de las placas de los vehículos más infraccionados es Jalisco con el 86%, pero podemos observar dos registros llamados CDMX, uno más llamado Ciudad de México y dos más de Estado de México. Esto es causado porque para los lenguajes de programación, no es lo mismo escribir 'CDMX' a 'CDMX ', por lo que un simple espacio puede ensuciar demasiado la base de datos. De igual modo, la columna **MARCA** causa el mismo problema:

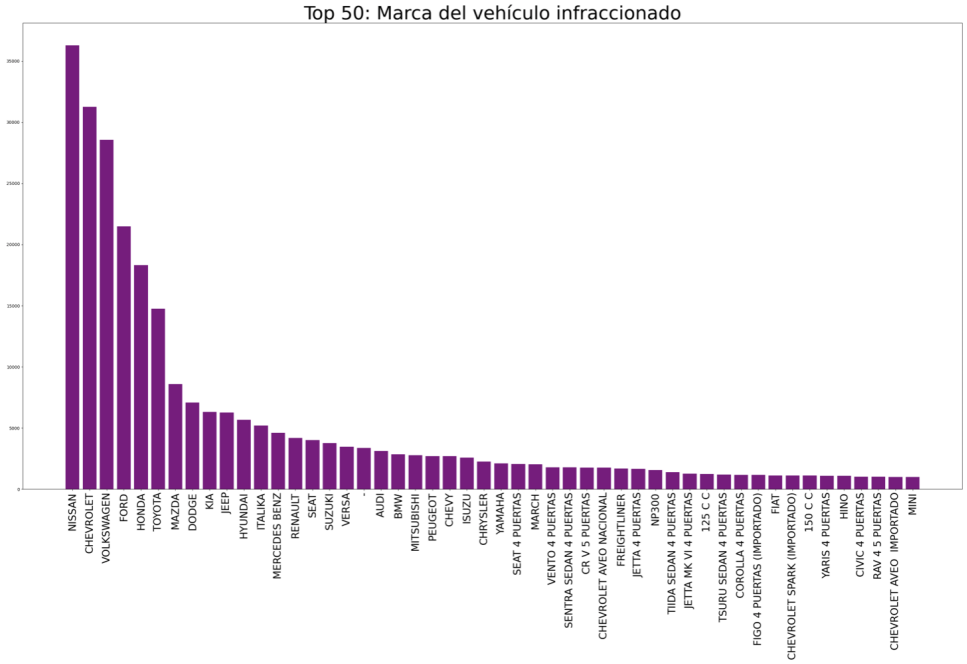


Gráfico 2. Gráfica de frecuencias de las 50 marcas más infraccionadas.

Podemos observar que Nissan es el auto más infraccionado con más de 35 mil registros, pero recorriendo la gráfica de frecuencias, encontramos que en vez de escribir la marca del vehículo, se registró la infracción con el modelo como por ejemplo Vento, Aveo, Jetta, Civic, entre otros, y/o la descripción del mismo, como lo es el número de puertas o si es importado o no, o hasta una combinación de marca y modelo como Chevrolet Aveo.

Las columnas de **CALLE** y **CRUCE** son el mismo caso, solo que estas columnas no se tienen planeado modificar, ya que la información geográfica la podemos crear utilizando la **LATITUD** y **LONGITUD** registrada en cada infracción. Las columnas **FECHA** y **HORA** son registradas automáticamente al momento de crear la infracción en la base de datos. Y finalmente, la clave en la columna **INFRACCIÓN** también es registrada de manera manual por el oficial vial:

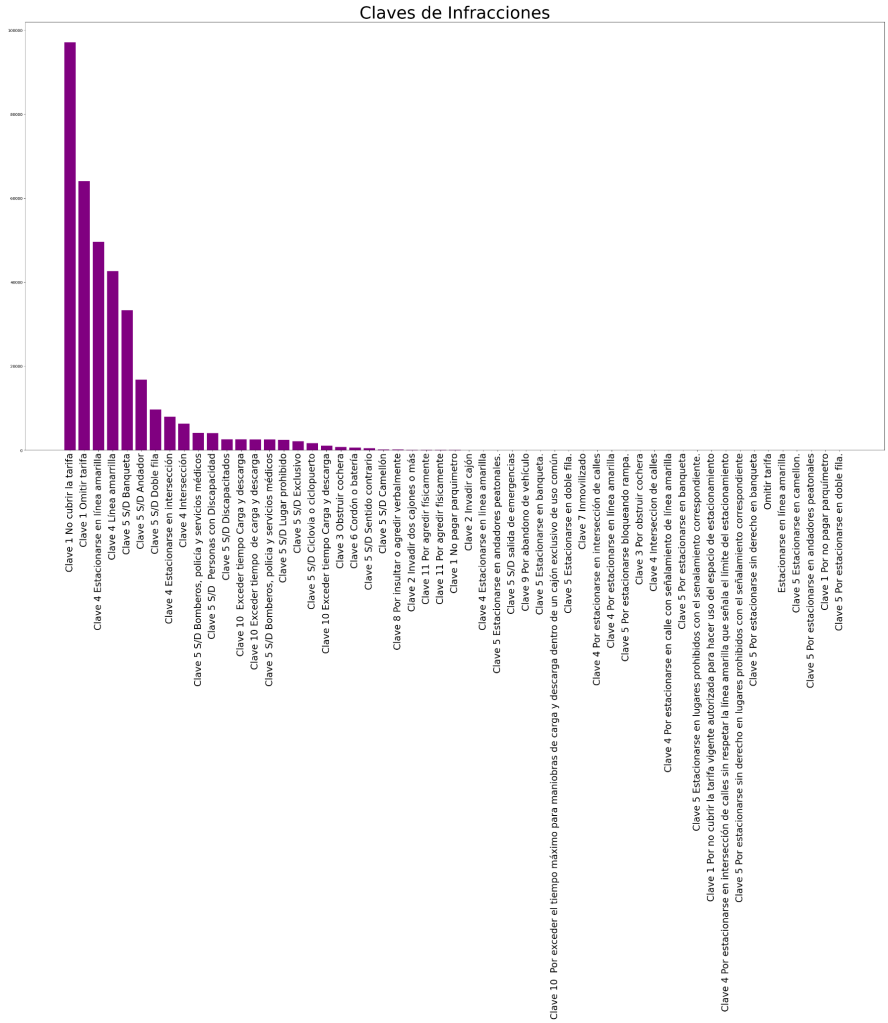


Gráfico 3. Gráfica de frecuencias de las claves de infracción.

El gráfico nos muestra que la clave 1 *no cubrir la tarifa* es la más registrada, pero inmediatamente se encuentra que la segunda infracción más registrada es la misma clave 1, pero bajo la descripción de *Omitir tarifa.* Los siguientes dos valores muestran el mismo caso, pero con la clave número 4, uno llamándose *Estacionarse en línea amarilla* y el segundo simplemente *línea amarilla.* Por lo que, las columnas **MARCA**, **ESTADO** e **INFRACCIÓN**, serán las que serán expuestas a un proceso de limpieza y corrección de datos.

3) Preparación de los datos

3.1 Selección de datos.

Para la creación de la solución al problema de las infracciones en la Zona Metropolitana de Guadalajara utilizando Machine Learning, se utilizarán los registros obtenidos durante todo el 2021, para así proponer un modelo de CLASIFICACIÓN que sea capaz de predecir el tipo de infracción basado en las diferentes características del vehículo o lugar geográfico donde se encontrará el mismo.

3.2 Limpieza de datos.

Como ya se mencionó antes, las columnas **MARCA**, **ESTADO** e **INFRACCIÓN**, serán las que se les realizará esta corrección de valores abiertos. Primeramente, los registros de texto de la base serán transformados a minúsculas para el mejor manejo de los datos. Comenzando por las marcas de los vehículos, se utilizarán técnicas de búsqueda y reemplazo en la columna; creando un arreglo con las principales marcas de los vehículos, donde se buscarán palabras clave y si se encuentra esta palabra, se reemplazará todo el registro por solamente dicha palabra clave, por ejemplo:

	ESTADO	MARCA	CALLE	CRUCE	FECHA	HORA	INFRACCION	LATITUD	LONGITUD
17	JALISCO	MAZDA CX-3 5 PUERTAS (IMPORTADO)	duque de rivas	miguel blanco de Tejada	16/08/21	12:34:00	Clave 1 Omitir tarifa	20.67	-103.38
18	CIUDAD DE MEXICO	HONDA	Calle Miguel Lerdo de Tejada	duque de Rivas	16/08/21	12:38:00	Clave 1 Omitir tarifa	20.67	-103.38
19	JALISCO	MITSUBISHI OUTLANDER 4 PUERTAS IMP	Francisco de Quevedo	Miguel Lerdo de Tejada	16/08/21	12:42:00	Clave 1 Omitir tarifa	20.67	-103.38

Imagen 2. Muestra dos de la base de datos de las infracciones en 2021.

En la Imagen 2, en la fila 17 y 19 de la columna **MARCA,** podemos observar los registros *MAZDA CX-3 5 PUERTAS (IMPORTADO)* y *MITSUBISHI OUTLANDER 4 PUERTAS IMP,* donde al aplicar el reemplazo, los registros quedarán como *MAZDA y MITSUBISHI* respectivamente.

```
for i in np.arange(len(marcas_list)):
    for j in np.arange(len(db)):
        if marcas_list[i] in str(db['MARCA'][j]):
            db['MARCA'][j] = db['MARCA'][j].replace(db['MARCA'][j],
                                                    marcas_list[i])
```

Imagen 3. Fragmento de código que realiza la búsqueda y reemplazo de palabras clave.

Para dicho proceso, se utilizó el ciclo anterior, donde *marcas_list* se compone de este arreglo de palabras claves a buscar en la columna respectiva, mientras que *db* es el DataFrame de las infracciones. Enseguida, los registros que simplemente cuentan con el nombre del modelo mas no la marca del vehículo, se aplicó un reemplazo con un diccionario de marcas de carros. Por ejemplo, si el registro se llama *Jetta 4 puertas* o *Vento 4 puertas*, fueron reemplazados por simplemente *Jetta* o *Vento.* Posteriormente, se aplicó un reemplazo con un diccionario de marcas y vehículos, donde si el registro es el modelo *Jetta* o *Vento* fue reemplazado por *Volkswagen*.

De la misma manera, la columna **ESTADO** se aplicaron dos reemplazos, el primero utilizando palabras clave, en este caso los 32 estados del país, y posteriormente el reemplazo específico de los registros, donde todos los estados extranjeros, la mayoría de Estados Unidos, fueron clasificados como *extranjeros*, y se corrigieron faltas

de ortografía o estados mal escritos, como por ejemplo *Chispas* se reemplazó con *Chiapas* y *Jalidco* se reemplazó con *Jalisco*.

La siguiente limpieza fue en la columna de **INFRACCIÓN**, donde simplemente nos quedamos con el número del registro en la infracción, utilizando la siguiente función:

```
def remove_special_characters(text):
    pattern = r'[^0-9\s]'
    text = re.sub(pattern, '', text)
    return text
```

Imagen 4. Fragmento de código que consta de una función para conservar únicamente los números de una cadena de texto.

Esta columna será después el target del modelo y la cual se transformó a tipo int., y se conservó solamente el número del tipo de infracción, por ejemplo *clave 1: no cubrir la tarifa* será simplemente un número 1, y *Clave 5: Estacionarse en línea amarilla* será un número 5. Se puede observar en la imagen # que el registro de la columna **MARCA,** ya cuenta con solamente la marca del vehículo sin el modelo, en minúsculas junto con la columna **ESTADO** y la columna **INFRACCIÓN** ya es de tipo numérico con el número del tipo de la infracción.

	ESTADO	MARCA	CALLE	CRUCE	FECHA	HORA	INFRACCION	LATITUD	LONGITUD
17	jalisco	mazda	duque de rivas	miguel blanco de tejada	16/08/21	12:34:00	1.0	20.67	-103.38
18	ciudad de mexico	honda	calle miguel lerdo de tejada	duque de rivas	16/08/21	12:38:00	1.0	20.67	-103.38
19	jalisco	mitsubishi	francisco de quevedo	miguel lerdo de tejada	16/08/21	12:42:00	1.0	20.67	-103.38

Imagen 5. Muestra de la base de datos después de la limpieza parcial.

Finalmente, se tomaron las siguientes consideraciones para obtener una base más limpia y ordenada: La primera consideración es quitar los nulos en la columna **ESTADO**. Después, se realizó un conteo de la columna **MARCAS** y tomando los que tengan más de 50 repeticiones, se conserva el 96% de los datos, por lo que se quitaron los que tuvieran menos de 50 repeticiones. Además, se quitaron los registros que no tuvieran **LATITUD** y **LONGITUD**, ya que serán unas variables vitales para probar en los modelos, y finalmente se quitaron los registros que se consideraron como *otro* en la columna **MARCA**. Recordando, la base inicial contaba con 156,000 registros aproximadamente, y con la limpieza de datos, la base queda con una longitud de 141,000 registros.

3.3 Transformación de datos.

Posterior a la limpieza, se agregaron nuevas columnas a partir de los datos existentes: La columna **MES,** que es el mes del registro de la infracción (en inglés), la columna **HORA_NUM** que consta de solo el valor de la hora removiendo así los minutos de la columna **HORA**, **ESTADO_cat** y **MARCA_cat**, que son las columnas convertidas a un valor numérico correspondiente a cada uno de los registros únicos de dichas columnas, y no se tomaron en cuenta las columnas **CALLE** y **CRUCE,** quedando de la siguiente manera:

	ESTADO	MARCA	FECHA	ANO	MES	HORA	HORA_NUM	LATITUD	LONGITUD	ESTADO_cat	MARCA_cat	INFRACCION
2	jalisco	volkswagen	2021-08-16	2021	August	11:26:00	11	20.67	-103.38	15	65	1
3	jalisco	freightliner	2021-08-16	2021	August	11:27:00	11	20.67	-103.38	15	17	5
4	jalisco	volkswagen	2021-08-16	2021	August	11:38:00	11	20.67	-103.38	15	65	1

Imagen 6. Muestra de la base de datos con los nuevos datos transformados.

1. **ESTADO**: Estado de la placa del automóvil infraccionado | Tipo String.
2. **MARCA**: Marca del automóvil infraccionado | Tipo String.
3. **FECHA**: Fecha de la infracción | Tipo Datetime.
4. **AÑO**: Año de la infracción. | *Tipo int.*
5. **MES**: Longitud geográfica de la infracción | *Tipo String.*
6. **HORA**: Hola de la infracción | Tipo Datetime.
7. **HORA_NUM**: Hora de la infracción | *Tipo int.*
8. **LATITUD**: Latitud geográfica de la infracción | Tipo int.
9. **LONGITUD**: Longitud geográfica de la infracción | Tipo int.
10. **ESTADO_cat**: Valor numérico único por ESTADO. | *Tipo int.*
11. **MARCA_cat**: Valor numérico único por MARCA. | *Tipo int.*
12. **INFRACCIÓN**: Tipo de infracción cometida por el vehículo | *Tipo int: Valores 1, 4 o 5*

Para entender un poco mejor las columnas _cat, el siguiente diccionario contiene los valores que se utilizaron para la columna **ESTADO**:

```
{'aguascalientes': 0, 'baja california norte': 1, 'baja california sur': 2,
 'campeche': 3, 'chiapas': 4, 'chihuahua': 5, 'ciudad de mexico': 6,
 'coahuila': 7, 'colima': 8, 'durango': 9, 'estado de mexico': 10,
 'extranjero': 11, 'guanajuato': 12, 'guerrero': 13, 'hidalgo': 14,
 'jalisco': 15, 'michoacan': 16, 'morelos': 17, 'nayarit': 18,
 'nuevo leon': 19, 'oaxaca': 20, 'puebla': 21,
 'queretaro': 22, 'quintana roo': 23, 'san luis potosi': 24,
 'sinaloa': 25, 'sonora': 26, 'tabasco': 27, 'tamaulipas': 28,
 'tlaxcala': 29, 'veracruz': 30, 'yucatan': 31, 'zacatecas': 32}
```

Imagen 7. Diccionario de valores de la columna **ESTADO_cat**.

4) Aplicación de Técnicas de Modelación

4.1 Variables predictoras y variable target.

- Para crear el modelo de clasificación, se utilizarán las columnas **ESTADO_cat, MARCA_cat, MES, HORA_NUM, LATITUD, LONGITUD.**
- No se tomó en cuenta el año ya que es una variable constante, al tener solo datos de 2021.
- No se tomaron en cuenta tampoco las columnas **ESTADO** y **MARCA**, ya que son variables de tipo String y se utilizaron sus respectivos valores únicos en **ESTADO_cat** y **MARCA_cat.**
- Asimismo, la columna **HORA** contiene valores más específicos con la hora exacta de la infracción, por lo que se utilizó solo la hora, que se encuentra en la columna **HORA_NUM.**
- La columna **MES** se transformó a valores numéricos, donde January es el número 1, February el número 2 y así sucesivamente hasta Diciembre siendo el número 12.
- Finalmente, el target del modelo de clasificación será el tipo de infracción en la columna **INFRACCIÓN**, siendo 1, 4 o 5 los posibles valores de salida.

4.2 Explicación breve de los diferentes modelos de aprendizaje e hiperparámetros.

Existen diferentes modelos de Machine Learning para realizar una clasificación, entre los que podemos encontrar la regresión lineal, regresión logística, árboles de decisión, bosques de decisión, potenciación del gradiente, Naive Bayes, redes neuronales, entre muchos otros.

a) Regresión lineal

Un modelo de regresión es un modelo que permite describir cómo influye una variable dependiente X sobre otra variable independiente Y. El análisis de regresión lineal es uno de los modelos de predicción más simples y más utilizados para estudiar la relación entre variables, el cual consiste en un modelo matemático usado para aproximar la relación de dependencia entre una variable dependiente Y, las variables independientes Xi y un término independiente B. Este modelo puede ser expresado en forma de recta, quedando una fórmula tan simple como la siguiente:

$$Y = w \cdot x_i + B$$

El aprendizaje de dicho modelo se basa en encontrar cuáles son los mejores coeficientes para los datos existentes. Los mejores coeficientes serán los que minimicen alguna medida de error.

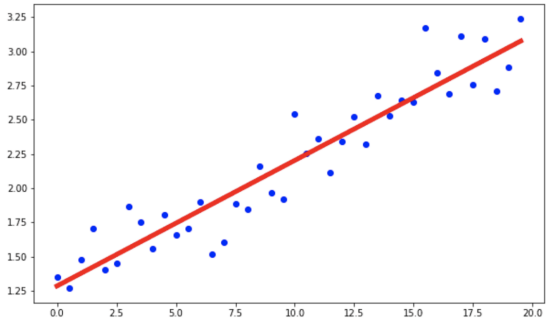


Imagen 8. Ejemplo gráfico de una regresión lineal ajustada a datos.

La manera más fácil de medir este modelo es con el coeficiente de determinación R^2, el cual determina la calidad del modelo para replicar los resultados, y la proporción de variación de los resultados que puede explicarse por el modelo. El rango de R^2 está entre 0 y 1, siendo 1 lo mejor.

b) Regresión logística

La regresión logística es un paso más adelante de la regresión lineal, la cual se utiliza para ayudar a crear predicciones un poco más precisas. Existen dos tipos de variables medibles, las variables o características explicativas y la variable de respuesta o variable binaria objetivo, que corresponde al resultado. La regresión logística se usa muy seguido porque es más eficiente y no necesita grandes cantidades de recursos computacionales.

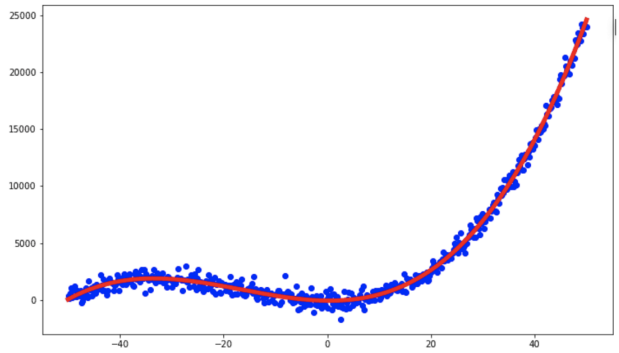


Imagen 9. Ejemplo gráfico de una regresión logística ajustada a datos.

Además, al igual que la regresión lineal, se puede interpretar fácilmente y no necesita escalar las características de entrada siendo fácil de regularizar. Se usa para describir datos y explicar la relación entre una variable dependiente y una o más variables independientes. En general, este modelo se puede utilizar para varios problemas de clasificación.

c) Árboles de decisión

Los árboles de decisión son otra técnica de aprendizaje automático, la cual está basada en tomar sus decisiones en forma de árbol. Los nodos intermedios, llamados ramas, representan soluciones, mientras que los nodos finales, llamados hojas, son la predicción del modelo.

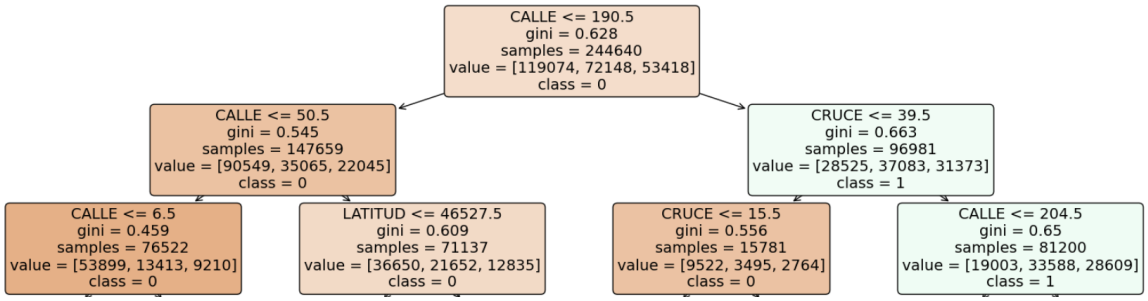


Imagen 10. Ejemplo gráfico de un árbol de decisión.

Los árboles de decisión se construyen usando un algoritmo voraz, el cual elige qué atributos y qué límites son los mejores para tomar las decisiones a medida que va aumentando el tamaño del árbol. Al usar un algoritmo voraz, no se tiene la garantía de que este sea el mejor árbol posible. Hoy en día, las librerías de modelos de aprendizaje tienen incluido este tipo de modelo, en los cuales se puede establecer parámetros como la profundidad máxima del árbol, el número mínimo de muestras necesarias antes de dividir este nodo, el número mínimo de muestras que debe haber en un nodo final (hoja), el número máximo de nodos finales, entre varios parámetros más.

d) Bosque Aleatorio

Un Random Forest es un conjunto, también conocido como ensemble, de árboles de decisión combinados con bagging, esto significa que distintos árboles ven distintas porciones de los datos y ningún árbol ve todos los datos de entrenamiento. Esto hace que cada árbol se entrene con distintas muestras de datos para un mismo problema, para que al combinar sus resultados, unos errores se compensan con otros y tenemos una predicción que generaliza mejor.

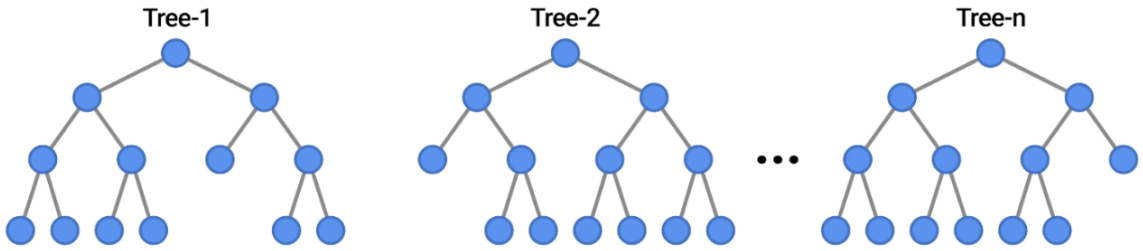


Imagen 11. Ejemplo gráfico de un bosque aleatorio.

La desventaja de los árboles de decisión es que tienen la tendencia de sobre-ajustar, tienden a aprender muy bien los datos de entrenamiento pero su generalización no es tan buena. Una forma de mejorar la generalización de los árboles de decisión es usar regularización. Para mejorar mucho más la capacidad de generalización de los árboles de decisión, deberemos combinar varios árboles. De igual forma, se puede manipular los parámetros de los árboles de decisión, pero de igual forma, se puede decidir el número de árboles que va a tener el bosque aleatorio, el número de cores computacionales que se pueden usar para entrenar los árbole, o el número máximo de features que utilizará cada árbol para que cada uno tenga su propio entrenamiento.

e) Gradient boosting

Un modelo Gradient Boosting está creado por un conjunto de árboles de decisión individuales, entrenados de forma secuencial, causando que cada nuevo árbol trate de mejorar los errores de los árboles anteriores. La predicción de una nueva observación se obtiene agregando las predicciones de todos los árboles individuales que forman el modelo.

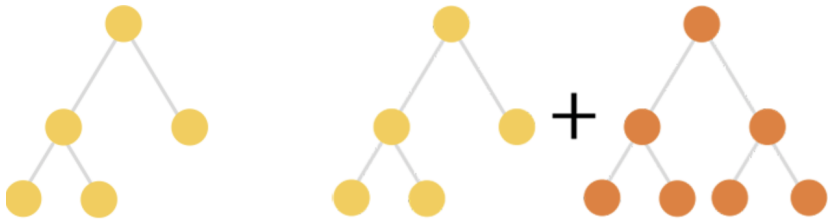


Imagen 12. Ejemplo gráfico de un Gradient Boosting.

Cada nuevo árbol utiliza información del árbol anterior para aprender de sus errores, mejorando iteración a iteración. En cada árbol individual, las observaciones se van distribuyendo por nodos, generando la estructura del árbol hasta alcanzar el nodo terminal. La predicción sale de agregar las predicciones de todos los árboles individuales que forman el modelo. Este modelo está basado en dos principales características, el bagging y el boosting: en el bagging se ajustan múltiples modelos, cada uno con un conjunto diferente de los datos de entrenamiento. Para predecir, todos los modelos que forman el agregado participan aportando su predicción. Los modelos Random Forest están dentro de esta categoría como se describió anteriormente, mientras que en el boosting se ajustan secuencialmente múltiples modelos sencillos, de forma que cada modelo aprende de los errores del anterior. Tres de los algoritmos de boosting más empleados son AdaBoost, Gradient Boosting y Stochastic Gradient Boosting.

f) Naive Bayes

Los modelos de Naive Bayes son una clase especial de algoritmos de clasificación que están basados en una técnica de clasificación estadística llamada “teorema de Bayes”. Estos modelos son llamados algoritmos “Naive”, o “Inocentes” en español y asumen que las variables predictoras son independientes entre sí.

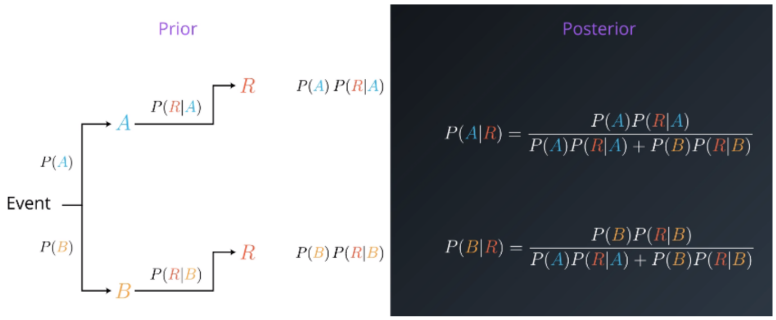


Imagen 13. Ejemplo gráfico de la teoría detrás de los modelos de Naive Bayes.

Proporcionan una manera fácil de construir modelos con un comportamiento muy bueno debido a su simplicidad. Lo consiguen proporcionando una forma de calcular la probabilidad posterior de que ocurra un cierto evento A, dadas algunas probabilidades de eventos anteriores.

g) Red Neuronal

Una red neuronal es otro método de inteligencia artificial, su característica principal es procesar datos de una manera que está inspirada en la forma en que lo hace el cerebro humano, el cual utiliza los nodos o las neuronas interconectados en una estructura de capas que se parece a un cerebro. Crea un sistema que el procesamiento computacional utiliza para aprender de sus errores y seguir mejorando. Las redes neuronales pueden ayudar a las computadoras a tomar decisiones inteligentes con asistencia humana limitada. Esto se debe a que pueden aprender y modelar las relaciones entre los datos de entrada y salida que no son lineales y que son complejos. Por ejemplo, pueden realizar las siguientes tareas.

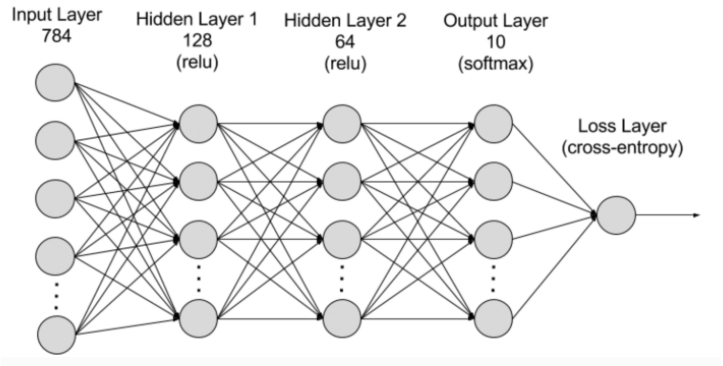


Imagen 14. Ejemplo gráfico de una red neuronal con dos capas ocultas.

El cerebro humano es lo que inspira a las redes neuronales. Las células del cerebro humano, llamadas neuronas, forman una red y con un alto nivel de interconexión y se envían señales eléctricas entre sí para ayudar a los humanos a procesar la información. De manera similar, una red neuronal artificial está formada por neuronas artificiales que trabajan juntas para resolver un problema. Las neuronas artificiales son módulos de software, llamados nodos. Una red neuronal básica tiene neuronas artificiales interconectadas en tres capas:

- **Capa de entrada:** Los datos entran en la red neuronal artificial desde la capa de entrada. Los nodos de entrada procesan los datos, los analizan o los clasifican y los pasan a la siguiente capa.
- **Capa(s) oculta(s):** Las redes neuronales artificiales pueden tener una gran cantidad de capas ocultas, las cuales toman su entrada de la capa de entrada o de otras capas ocultas y cada capa analiza la salida de la capa anterior, la procesa aún más y la pasa a la siguiente capa.
- **Capa de salida:** La capa de salida proporciona el resultado final de todo el procesamiento de datos que realiza la red neuronal artificial. Puede tener uno o varios nodos.

4.3 Explicación de la metodología para el entrenamiento y prueba.

Para crear el modelo de clasificación, se entrenó con el 75% de los datos y se realizaron las pruebas con el 25% restante, los cuales fueron mezclados y separados aleatoriamente, siempre usando una semilla para no obtener diferentes conjuntos de datos en los diferentes modelos que se realizaron.

4.4 Explicación breve de las métricas de evaluación.

Las métricas de evaluación realizadas ayudan a estimar la precisión de la generalización de un modelo sobre los datos futuros, las cuales se obtienen principalmente de la matriz de confusión que resulta de probar el modelo creado en el conjunto de test (el 25% de los datos no utilizados para el entrenamiento del modelo).

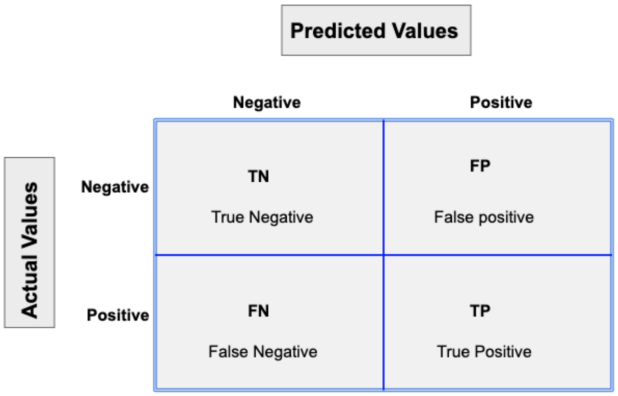


Imagen 15. Matriz de confusión de dos etiquetas.

donde:

- ➔ **Verdadero Positivo (TP):** Predicho Verdadero y Verdadero en realidad.
- ➔ **Verdadero Negativo (TN):** Predicho Falso y Falso en realidad.
- ➔ **Falso Positivo (FP):** Predicción de verdadero y falso en la realidad.
- ➔ **Falso Negativo (FN):** Predicción de falso y verdadero en la realidad.

Con estos valores, se puede llegar a diferentes métricas de evaluación, donde las principales son:

1) **Accuracy**: $\frac{TP + TN}{total}$

   Es el número de predicciones correctas hechas como una proporción de todas las predicciones hechas.

2) **Precision**: $\frac{TP}{TP + FP}$

   Es el número de predicciones identificadas correctamente como positivo de un total de elementos identificados como positivos.

3) **Recall**: $\frac{TP}{TP + FN}$

   Es la tasa positiva verdadera (TPR), que es la proporción de los verdaderos positivos a todo lo positivo.

4) **F1 Score**: $2 \cdot \frac{precision \cdot recall}{precision + recall}$

   El valor F1 se utiliza para combinar las medidas de *precision* y *recall* en un sólo valor. Esto es práctico porque hace más fácil el poder comparar el rendimiento combinado entre varias soluciones, donde el valor máximo es 1, siendo *precision* y *recall* perfectos.

4.5 Generación de los modelos y su evaluación a través de las diferentes métricas.

Para la creación del modelo de predicción del tipo de infracción que los automovilistas de la Zona Metropolitana pueden cometer, se decidió probar tres diferentes modelos: un Árbol de decisión, un Bosque aleatorio y un Gradient Boosting, con el 75% de los datos para entrenar y el 25% de los datos para probar:

a) Árboles de decisión

Para el árbol de decisión se utilizaron los siguientes features: ESTADO, MARCA, CALLE, CRUCE, FECHA, AÑO, MES, HORA, HORA_NUM, LATITUD, LONGITUD y el target INFRACCIÓN se reemplazaron los tipos de infracciones de la siguiente manera: Tipo 1 → 0, Tipo 4 → 1, Tipo 5 → 2, para mayor facilidad para el modelo. En seguida, se realizó un proceso de *OrdinalEncoder* a todos los features, para transformar los datos categóricos a numéricos, asignándoles un número único, quedando de la siguiente forma:

	ESTADO	MARCA	CALLE	CRUCE	FECHA	ANO	MES	HORA	HORA_NUM	LATITUD	LONGITUD
139665	ciudad de mexico	dodge	héroes de la independencia	ramón corona y degollado	2021-12-29	2021	12	16:49:00	16	20.672262	-103.346142
25114	jalisco	nissan	ocampo	lopez cotilla	2021-02-26	2021	2	00:16:32	0	20.674082	-103.349860
76666	jalisco	volkswagen	jose guadalupe montenegro	avenida chapultepec y progreso	2021-11-01	2021	11	00:18:44	0	20.669377	-103.367305

	ESTADO	MARCA	CALLE	CRUCE	FECHA	ANO	MES	HORA	HORA_NUM	LATITUD	LONGITUD
139665	1	1	1	1	1	1	1	1	1	1	1
25114	2	2	2	2	2	1	2	2	2	2	2
76666	2	3	3	3	3	1	3	3	2	3	3

Imagen 16. Muestra de la base de datos de entrenamiento antes y después de la transformación *OrdinalEncoder*

Con esta transformación, se comenzó la creación del modelo, utilizando como único hiperparámetro *max_depth = 6.* Primero, se evaluaron las importancias de cada una de las variables con las que este primer modelo entrenó y se puede visualizar en la siguiente gráfica:

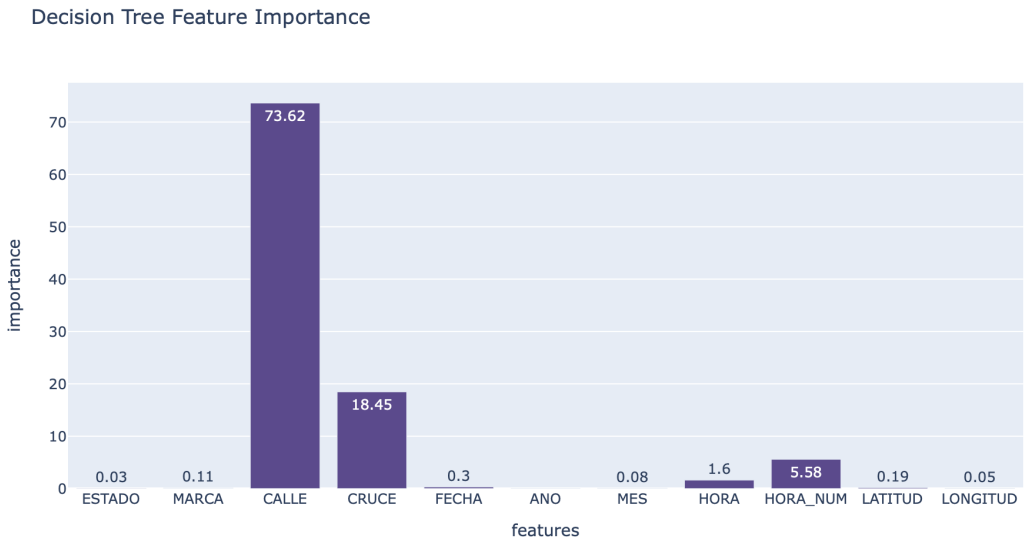


Gráfico 4. Importancia de las variables del modelo 1.

Se puede resaltar que la variable más importante claramente es la **CALLE** donde se realizó la infracción, con más del 73% de importancia, seguido del **CRUCE** de la calle donde se realizó la infracción, y la

HORA_NUM numérica de la infracción. Posteriormente, se realizó la prueba del modelo entrenado con el dataset de prueba, quedando la matriz de confusión de la base de prueba de la siguiente manera:

		Predicted Values		
		Tipo 1	Tipo 4	Tipo 5
Real Values	Tipo 1	**11,264**	3,488	621
	Tipo 4	3,138	**6,106**	1,332
	Tipo 5	3,865	2,513	**1,980**

Tabla 2. Matriz de confusión del modelo 1.

El accuracy del modelo se puede obtener de la diagonal de Verdaderos Positivos entre la suma de todo el dataset de prueba:

$$Accuracy_{Model\ 1} = \frac{11{,}264 + 6{,}106 + 1{,}980}{34{,}307} = \frac{19{,}350}{34{,}307} = 56.4\%$$

Después, la precisión, que calcula el número de verdaderos positivos entre en total de los elementos identificados como positivos:

$$Precision_{Model\ 1:\ Tipo\ 1} = \frac{11{,}264}{11{,}264 + 3{,}138 + 3{,}865} = \frac{11{,}264}{18{,}267} = 61.7\%$$

$$Precision_{Model\ 1:\ Tipo\ 4} = \frac{6{,}106}{3{,}488 + 6{,}106 + 2{,}513} = \frac{6{,}106}{12{,}107} = 50.4\%$$

$$Precision_{Model\ 1:\ Tipo\ 5} = \frac{1{,}980}{621 + 1{,}332 + 1{,}980} = \frac{1{,}980}{3{,}933} = 50.3\%$$

$$Precision_{Model\ 1} = \frac{61.7\% + 50.4\% + 50.3\%}{3} = 54.1\%$$

Enseguida se calcula el recall, que es la proporción de los verdaderos positivos a todo lo positivo.

$$Recall_{Model\ 1:\ Tipo\ 1} = \frac{11{,}264}{11{,}264 + 3{,}488 + 621} = \frac{11{,}264}{15{,}373} = 73.3\%$$

$$Recall_{Model\ 1:\ Tipo\ 4} = \frac{6{,}106}{3{,}138 + 6{,}106 + 1{,}332} = \frac{6{,}106}{10{,}576} = 57.7\%$$

$$Recall_{Model\ 1:\ Tipo\ 5} = \frac{1{,}980}{3{,}865 + 2{,}513 + 1{,}980} = \frac{1{,}980}{8{,}358} = 23.7\%$$

$$Recall_{Model\ 1} = \frac{61.7\% + 50.4\% + 50.3\%}{3} = 51.6\%$$

Y finalmente se calculará el f1-score, para poder comparar el rendimiento combinado entre varias soluciones.

$$F1_{Model\ 1:\ Tipo\ 1} = 2 \cdot \frac{61.7\% \cdot 73.3\%}{61.7\% + 73.3\%} = 2 \cdot \frac{45\%}{135\%} = 67.0\%$$

$$F1_{Model\ 1:\ Tipo\ 4} = 2 \cdot \frac{50.4\% \cdot 57.7\%}{50.4\% + 57.7\%} = 2 \cdot \frac{29\%}{108\%} = 53.8\%$$

$$F1_{Model\ 1:\ Tipo\ 5} = 2 \cdot \frac{50.3\% \cdot 23.7\%}{50.3\% + 23.7\%} = 2 \cdot \frac{12\%}{74\%} = 32.2\%$$

$$F1_{Model\ 1} = \frac{67.0\% + 53.8\% + 32.2\%}{3} = 51.0\%$$

Además, un árbol de decisión es fácil de visualizar, por lo que se pueden ir observando las ramas y nodos que fue formando el algoritmo:

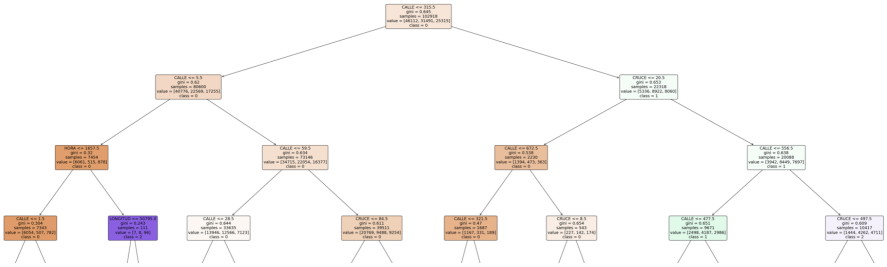


Imagen 17. Muestra de los primeros 3 niveles del árbol de decisión creado.

b) Bosque Aleatorio

Para el bosque de decisión se utilizaron los siguientes features: ESTADO, MARCA, CALLE, CRUCE, FECHA, AÑO, MES, HORA, HORA_NUM, LATITUD, LONGITUD y el target INFRACCIÓN se reemplazaron los tipos de infracciones de la siguiente manera: Tipo 1 → 0, Tipo 4 → 1, Tipo 5 → 2, para mayor facilidad para el modelo. En seguida, se realizó el mismo proceso de *OrdinalEncoder* a todos los features, igual que en el modelo anterior. De igual forma, se evaluaron las importancias de cada una de las variables con las que este primer modelo entrenó y se puede visualizar en la siguiente gráfica:

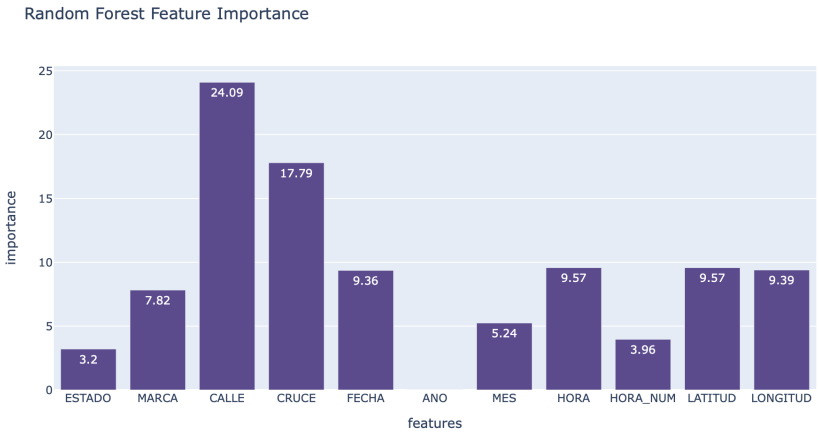


Gráfico 5. Importancia de las variables del modelo 2.

Se observa que nuevamente las dos variables más importantes son **CALLE** y **CRUCE**, pero ya existe después una mejor distribución de importancia entre todas las demás variables, quedando entre 7% y 9%. Con esta transformación, se comenzó la creación del modelo, utilizando como hiperparámetros

n_estimators = 200, min_samples_leaf = 2 y max_samples = 8, quedando la matriz de confusión de la base de prueba de la siguiente manera:

		Predicted Values		
		Tipo 1	Tipo 4	Tipo 5
Real Values	Tipo 1	**13,924**	841	608
	Tipo 4	2,200	**6,888**	1,488
	Tipo 5	2,258	1,752	**4,248**

Tabla 3. Matriz de confusión del modelo 2.

El accuracy del modelo se puede obtener de la diagonal de Verdaderos Positivos entre la suma de todo el dataset de prueba:

$$Accuracy_{Model\ 2} = \frac{13{,}924 + 6{,}888 + 4{,}248}{34{,}207} = \frac{25{,}060}{34{,}207} = 73.3\%$$

Después, la precisión, que calcula el número de verdaderos positivos entre en total de los elementos identificados como positivos:

$$Precision_{Model\ 2:\ Tipo\ 1} = \frac{13{,}924}{13{,}924 + 2{,}200 + 2{,}258} = \frac{13{,}924}{18{,}382} = 75.7\%$$

$$Precision_{Model\ 2:\ Tipo\ 4} = \frac{6{,}888}{841 + 6{,}888 + 1{,}752} = \frac{6{,}888}{9{,}481} = 72.7\%$$

$$Precision_{Model\ 2:\ Tipo\ 5} = \frac{4{,}248}{608 + 1{,}488 + 4{,}248} = \frac{4{,}248}{6{,}344} = 67.0\%$$

$$Precision_{Model\ 2} = \frac{75.7\% + 72.7\% + 67.0\%}{3} = 71.8\%$$

Enseguida se calcula el recall, que es la proporción de los verdaderos positivos a todo lo positivo.

$$Recall_{Model\ 2:\ Tipo\ 1} = \frac{13{,}924}{13{,}924 + 841 + 608} = \frac{13{,}924}{15{,}373} = 90.6\%$$

$$Recall_{Model\ 2:\ Tipo\ 4} = \frac{6{,}888}{2{,}200 + 6{,}888 + 1{,}488} = \frac{6{,}888}{0{,}576} = 65.1\%$$

$$Recall_{Model\ 2:\ Tipo\ 5} = \frac{4{,}248}{2{,}258 + 1{,}752 + 4{,}248} = \frac{4{,}248}{8{,}258} = 51.4\%$$

$$Recall_{Model\ 2} = \frac{90.6\% + 65.1\% + 51.4\%}{3} = 69.0\%$$

Y finalmente se calculará el f1-score, para poder comparar el rendimiento combinado entre varias soluciones.

$$F1_{Model\ 2:\ Tipo\ 1} = 2 \cdot \frac{75.7\% \cdot 90.6\%}{75.7\% + 90.6\%} = 2 \cdot \frac{69\%}{166\%} = 82.5\%$$

$$F1_{Model\ 2:\ Tipo\ 4} = 2 \cdot \frac{72.7\% \cdot 65.1\%}{72.7\% + 65.1\%} = 2 \cdot \frac{47\%}{138\%} = 68.7\%$$

$$F1_{Model\ 2:\ Tipo\ 5} = 2 \cdot \frac{67.0\% \cdot 51.4\ \%}{67.0\% + 51.4\%} = 2 \cdot \frac{34\%}{118\%} = 58.2\%$$

$$F1_{Model\ 2} = \frac{82.5\% + 68.7\% + 58.2\%}{3} = 69.8\%$$

c) Gradient boosting

Para el gradient boosting, se utilizaron los siguientes features: ESTADO_cat, MARCA_cat, MES, HORA_NUM, LATITUD, LONGITUD y el target INFRACCIÓN se reemplazaron los tipos de infracciones de la siguiente manera: Tipo 1 → 0, Tipo 4 → 1, Tipo 5 → 2, para mayor facilidad para el modelo.

	ESTADO	MARCA	CALLE	CRUCE	FECHA	MES	ANO	HORA	HORA_NUM	LATITUD	LONGITUD	ESTADO_cat	MARCA_cat
0	morelos	jeep	calle manuel lópez cotilla	francisco de quevedo	2021-08-16	8	2021	11:11:00	11	20.67	-103.38	17	30
1	jalisco	kicks	calle lope de vega	avenida de la paz	2021-08-16	8	2021	11:22:00	11	20.67	-103.38	15	33
2	jalisco	volkswagen	avenida de la paz	francisco de quevedo	2021-08-16	8	2021	11:26:00	11	20.67	-103.38	15	64

Imagen 18. Muestra del dataset de entrenamiento del modelo 3.

De igual forma, se evaluaron las importancias de cada una de las variables con las que este primer modelo entrenó y se puede visualizar en la siguiente gráfica:

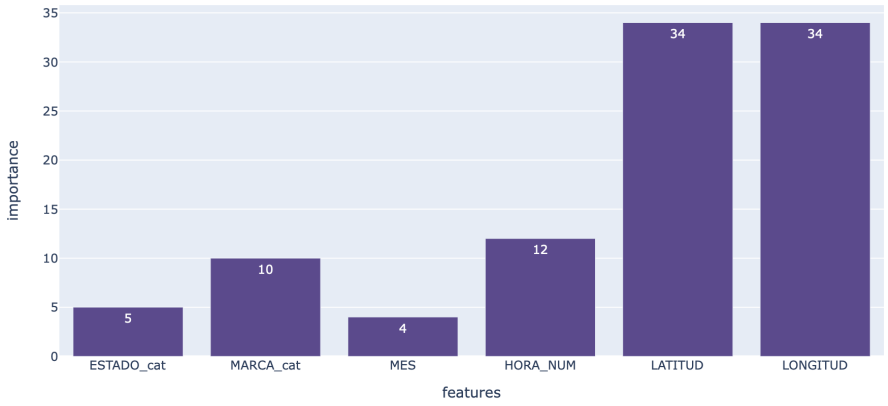


Gráfico 6. Importancia de las variables del modelo 3.

Donde se puede observar que aquí no se incluyeron la **CALLE** y el **CRUCE**, resultando **LATITUD** y **LONGITUD,** las variables que más impacto tienen en el modelo con 34% ambas variables. Y posteriormente, se comenzó la creación del modelo, utilizando como único hiperparámetro *n_estimators=4000, learning_rate=0.01, max_depth=6,* quedando la matriz de confusión de la base de prueba de la siguiente manera:

	Predicted Values		
	Tipo 1	Tipo 4	Tipo 5

Real Values	Tipo 1	**14,962**	224	187
	Tipo 4	1,043	**8,234**	1,299
	Tipo 5	1,143	1,610	**5,605**

Tabla 4. Matriz de confusión del modelo 3.

El accuracy del modelo se puede obtener de la diagonal de Verdaderos Positivos entre la suma de todo el dataset de prueba:

$$Accuracy_{Model\ 3} = \frac{14{,}962 + 8{,}234 + 5{,}605}{34{,}307} = \frac{28{,}801}{34{,}307} = 84.0\%$$

Después, la precisión, que calcula el número de verdaderos positivos entre en total de los elementos identificados como positivos:

$$Precision_{Model\ 3:\ Tipo\ 1} = \frac{14{,}962}{14{,}962 + 1{,}043 + 1{,}143} = \frac{14{,}962}{17{,}148} = 87.3\%$$

$$Precision_{Model\ 3:\ Tipo\ 4} = \frac{8{,}234}{224 + 8{,}234 + 1{,}610} = \frac{8{,}234}{10{,}068} = 81.8\%$$

$$Precision_{Model\ 3:\ Tipo\ 5} = \frac{5{,}605}{187 + 1{,}299 + 5{,}605} = \frac{5{,}605}{7{,}091} = 79.0\%$$

$$Precision_{Model\ 3} = \frac{87.3\% + 81.8\% + 79.0\%}{3} = 82.7\%$$

Enseguida se calcula el recall, que es la proporción de los verdaderos positivos a todo lo positivo.

$$Recall_{Model\ 3:\ Tipo\ 1} = \frac{14{,}962}{14{,}+962 + 224 + 187} = \frac{14{,}962}{5{,}373} = 97.3\%$$

$$Recall_{Model\ 3:\ Tipo\ 4} = \frac{8{,}234}{1{,}043 + 8{,}234 + 1{,}299} = \frac{8{,}234}{10{,}068} = 82.7\%$$

$$Recall_{Model\ 3:\ Tipo\ 5} = \frac{5{,}605}{1{,}143 + 1{,}610 + 5{,}605} = \frac{5{,}605}{8{,}358} = 67.1\%$$

$$Recall_{Model\ 3} = \frac{97.3\% + 82.7\% + 67.1\%}{3} = 80.7\%$$

Y finalmente se calculará el f1-score, para poder comparar el rendimiento combinado entre varias soluciones.

$$F1_{Model\ 3:\ Tipo\ 1} = 2 \cdot \frac{87.3\% \cdot 97.3\%}{87.3\% + 97.3\%} = 2 \cdot \frac{85\%}{185\%} = 92.0\%$$

$$F1_{Model\ 3:\ Tipo\ 4} = 2 \cdot \frac{81.8\% \cdot 82.7\%}{81.8\% + 82.7\%} = 2 \cdot \frac{64\%}{160\%} = 79.8\%$$

$$F1_{Model\ 3:\ Tipo\ 5} = 2 \cdot \frac{79.0\% \cdot 67.1\%}{79.0\% + 67.1\%} = 2 \cdot \frac{53\%}{146\%} = 72.6\%$$

$$F1_{Model\ 3} = \frac{92.0\% + 79.8\% + 72.6\%}{3} = 81.4\%$$

4.6 Ajuste de modelos a través de hiperparámetros y Cross Validation.

Para todas las siguientes pruebas, se crearon diferentes hiperparámetros dependiendo el modelo, y se realizó un *KFold* de 5 cross validations distintos en cada modelo:

a) Árboles de decisión

Para el primer modelo, se implementaron los hiperparámetros en los parámetros de *max_depth* y *min_samples_leaf*, quedando de la siguiente forma:

- *max_depth* = [3, 4, 5, 6, 7, 8]
- *min_samples_leaf* = [0, 1, 2, 3]

Encontrando que los mejores hiperparámetros para este primer modelo son: *min_samples_leaf = 3, ymax_depth = 7.* Donde las nueva matriz de confusión quedó de la siguiente forma:

		Predicted Values		
		Tipo 1	Tipo 4	Tipo 5
Real Values	Tipo 1	**12,382**	2,453	538
	Tipo 4	2,817	**6,515**	1,244
	Tipo 5	3,174	2,690	**2,494**

Tabla 5. Matriz de confusión del modelo 1 con los mejores hiperparámetros.

El accuracy del modelo se puede obtener de la diagonal de Verdaderos Positivos entre la suma de todo el dataset de prueba:

$$Accuracy_{Model\ 1} = \frac{12{,}382 + 6{,}515 + 2{,}494}{34{,}307} = \frac{21{,}391}{34{,}307} = 62.4\%$$

Después, la precisión, que calcula el número de verdaderos positivos entre en total de los elementos identificados como positivos:

$$Precision_{Model\ 1:\ Tipo\ 1} = \frac{12{,}382}{12{,}382 + 2{,}817 + 3{,}174} = \frac{12{,}382}{18{,}373} = 67.4\%$$

$$Precision_{Model\ 1:\ Tipo\ 4} = \frac{6{,}515}{2{,}453 + 6{,}515 + 2{,}690} = \frac{6{,}515}{10{,}576} = 55.9\%$$

$$Precision_{Model\ 1:\ Tipo\ 5} = \frac{2{,}494}{538 + 1{,}244 + 2{,}494} = \frac{2{,}494}{4{,}276} = 58.3\%$$

$$Precision_{Model\ 1} = \frac{67.4\% + 61.6\% + 58.3\%}{3} = 60.5\%$$

Enseguida se calcula el recall, que es la proporción de los verdaderos positivos a todo lo positivo.

$$Recall_{Model\ 1:\ Tipo\ 1} = \frac{12{,}382}{12{,}382 + 2{,}453 + 538} = \frac{12{,}382}{15{,}373} = 80.5\%$$

$$Recall_{Model\ 1:\ Tipo\ 4} = \frac{6{,}515}{2{,}817 + 6{,}515 + 1{,}244} = \frac{6{,}515}{10{,}576} = 61.6\%$$

$$Recall_{Model\ 1:\ Tipo\ 5} = \frac{2{,}494}{3{,}174 + 2{,}690 + 2{,}494} = \frac{2{,}494}{} = 29.8\%$$

$$Recall_{Model\ 1} = \frac{80.5\% + 61.6\% + 29.8\%}{3} = 57.3\%$$

Y finalmente se calculará el f1-score, para poder comparar el rendimiento combinado entre varias soluciones.

$$F1_{Model\ 1:\ Tipo\ 1} = 2 \cdot \frac{67.4\% \cdot 80.5\%}{67.4\% + 80.5\%} = 2 \cdot \frac{54\%}{148\%} = 73.4\%$$

$$F1_{Model\ 1:\ Tipo\ 4} = 2 \cdot \frac{55.9\% \cdot 61.6\%}{55.9\% + 61.6\%} = 2 \cdot \frac{34\%}{117\%} = 58.6\%$$

$$F1_{Model\ 1:\ Tipo\ 5} = 2 \cdot \frac{60.5\% \cdot 29.8\%}{60.5\% + 29.8\%} = 2 \cdot \frac{\%}{\%} = 39.5\%$$

$$F1_{Model\ 1} = \frac{73.4\% + 58.6\% + 39.5\%}{3} = 57.2\%$$

b) Bosque Aleatorio

Para el segundo modelo, se implementaron los hiperparámetros en los parámetros de *n_estimators*, quedando de la siguiente forma:

- *n_estimators* = [10, 20, 30, 50, 100 ,200, 250, 300, 350]

Encontrando que los mejores hiperparámetros para este primer modelo son: *n_estimators = 350.* Donde las nueva matriz de confusión quedó de la siguiente forma:

		Predicted Values		
		Tipo 1	Tipo 4	Tipo 5
Real Values	Tipo 1	**14,083**	720	570
	Tipo 4	2,168	**6,917**	1,491
	Tipo 5	2,299	1,663	**4,396**

Tabla 6. Matriz de confusión del modelo 2 con los mejores hiperparámetros.

El accuracy del modelo se puede obtener de la diagonal de Verdaderos Positivos entre la suma de todo el dataset de prueba:

$$Accuracy_{Model\ 2} = \frac{14{,}083 + 6{,}917 + 4{,}396}{34{,}207} = \frac{25{,}396}{34{,}207} = 74.0\%$$

Después, la precisión, que calcula el número de verdaderos positivos entre en total de los elementos identificados como positivos:

$$Precision_{Model\ 2:\ Tipo\ 1} = \frac{14{,}083}{14{,}083 + 2{,}168 + 2{,}299} = \frac{14{,}083}{18{,}550} = 75.9\%$$

$$Precision_{Model\ 2:\ Tipo\ 4} = \frac{6{,}917}{720 + 6{,}917 + 1{,}663} = \frac{6{,}917}{9300,} = 74.4\%$$

$$Precision_{Model\ 2:\ Tipo\ 5} = \frac{4{,}396}{570 + 1{,}491 + 4{,}396} = \frac{4{,}396}{6{,}344} = 68.1\%$$

$$Precision_{Model\ 2} = \frac{75.9\% + 74.4\% + 68.1\%}{3} = 72.8\%$$

Enseguida se calcula el recall, que es la proporción de los verdaderos positivos a todo lo positivo.

$$Recall_{Model\ 2:\ Tipo\ 1} = \frac{14{,}083}{13{,}924 + 720 + 570} = \frac{14{,}083}{15{,}373} = 91.6\%$$

$$Recall_{Model\ 2:\ Tipo\ 4} = \frac{6{,}917}{2{,}168 + 6{,}917 + 1{,}491} = \frac{6{,}917}{10{,}576} = 65.4\%$$

$$Recall_{Model\ 2:\ Tipo\ 5} = \frac{4{,}396}{2{,}258 + 1{,}752 + 4{,}396} = \frac{4{,}396}{8{,}358} = 52.6\%$$

$$Recall_{Model\ 2} = \frac{91.6\% + 65.4\% + 52.6\%}{3} = 69.9\%$$

Y finalmente se calculará el f1-score del segundo modelo.

$$F1_{Model\ 2:\ Tipo\ 1} = 2 \cdot \frac{75.9\% \cdot 91.6\%}{75.9\% + 91.6\%} = 2 \cdot \frac{70\%}{168\%} = 83.0\%$$

$$F1_{Model\ 2:\ Tipo\ 4} = 2 \cdot \frac{74.4\% \cdot 65.4\%}{74.4\% + 65.4\%} = 2 \cdot \frac{45\%}{140\%} = 69.6\%$$

$$F1_{Model\ 2:\ Tipo\ 5} = 2 \cdot \frac{68.1\% \cdot 52.6\%}{68.1\% + 52.6\%} = 2 \cdot \frac{36\%}{121\%} = 59.3\%$$

$$F1_{Model\ 2} = \frac{82.5\% + 68.7\% + 58.2\%}{3} = 69.8\%$$

c) Gradient boosting

Para el tercer modelo, se implementaron los hiperparámetros en los parámetros de *n_estimators*,*max_depth* y *n_jobs* quedando de la siguiente forma:

- *n_estimators* = [1000, 1500, 2000, 2500, 3000, 4000]
- *max_depth* = [5, 6, 7, 8, 9]
- *n_jobs* = [0, 1, 2, 3]

Encontrando que los mejores hiperparámetros para este tercer modelo de Gradient Boost son: *n_estimators = 3,000, ymamax_depth = 8 y n_jobs = 2.* Donde las nueva matriz de confusión quedó de la siguiente forma:

	Predicted Values

		Tipo 1	Tipo 4	Tipo 5
Real Values	Tipo 1	**14,962**	220	184
	Tipo 4	1,067	**8,234**	1,305
	Tipo 5	1,193	1,580	**5,605**

Tabla 7. Matriz de confusión del modelo 3 con los mejores hiperparámetros.

El accuracy del modelo se puede obtener de la diagonal de Verdaderos Positivos entre la suma de todo el dataset de prueba:

$$Accuracy_{Model\ 3} = \frac{14{,}962 + 8{,}234 + 5{,}605}{34{,}307} = \frac{28{,}801}{34{,}307} = 83.8\%$$

Después, la precisión, que calcula el número de verdaderos positivos entre en total de los elementos identificados como positivos:

$$Precision_{Model\ 3:\ Tipo\ 1} = \frac{14{,}962}{14{,}962 + 1{,}067 + 1{,}193} = \frac{14{,}962}{17{,}222} = 86.9\%$$

$$Precision_{Model\ 3:\ Tipo\ 4} = \frac{8{,}234}{220 + 8{,}234 + 1{,}580} = \frac{8{,}234}{10{,}034} = 82.1\%$$

$$Precision_{Model\ 3:\ Tipo\ 5} = \frac{5{,}605}{184 + 1{,}305 + 5{,}605} = \frac{5{,}605}{7{,}904} = 79.0\%$$

$$Precision_{Model\ 3} = \frac{86.9\% + 82.1\% + 79.0\%}{3} = 82.6\%$$

Enseguida se calcula el recall, que es la proporción de los verdaderos positivos a todo lo positivo.

$$Recall_{Model\ 3:\ Tipo\ 1} = \frac{14{,}962}{14{,}962 + 220 + 184} = \frac{14{,}962}{15{,}366} = 97.4\%$$

$$Recall_{Model\ 3:\ Tipo\ 4} = \frac{8{,}234}{1{,}067 + 8{,}234 + 1{,}305} = \frac{8{,}234}{10{,}606} = 77.6\%$$

$$Recall_{Model\ 3:\ Tipo\ 5} = \frac{5{,}605}{1{,}193 + 1{,}580 + 5{,}605} = \frac{5{,}605}{8{,}378} = 66.9\%$$

$$Recall_{Model\ 3} = \frac{97.4\% + 77.6\% + 66.9\%}{3} = 80.6\%$$

Y finalmente se calculará el f1-score del tercer modelo:

$$F1_{Model\ 3:\ Tipo\ 1} = 2 \cdot \frac{86.9\% \cdot 97.4\%}{86.9\% + 97.4\%} = 2 \cdot \frac{85\%}{184\%} = 91.8\%$$

$$F1_{Model\ 3:\ Tipo\ 4} = 2 \cdot \frac{82.1\% \cdot 77.6\%}{82.1\% + 77.6\%} = 2 \cdot \frac{64\%}{160\%} = 79.8\%$$

$$F1_{Model\ 3:\ Tipo\ 5} = 2 \cdot \frac{79.0\% \cdot 66.9\%}{79.0\% + 66.9\%} = 2 \cdot \frac{53\%}{146\%} = 72.5\%$$

$$F1_{Model\ 3} = \frac{91.8\% + 79.8\% + 72.5\%}{3} = 81.4\%$$

4.7 Evaluación y selección de modelo(s) de acuerdo a las métricas. Hiperparámetros utilizados en los modelos.

Recordemos que los modelos fueron entrenados solamente con información de 2021, y las infracciones de 2022 fueron separadas para realizar una prueba de tipo backtesting, prediciendo la información que ya se tiene y que no sea solamente lo que se separó para la prueba del modelo.

El primer modelo, el árbol de decisión, como se vio anteriormente, tienden a aprender muy bien los datos de entrenamiento pero su generalización no es tan buena:

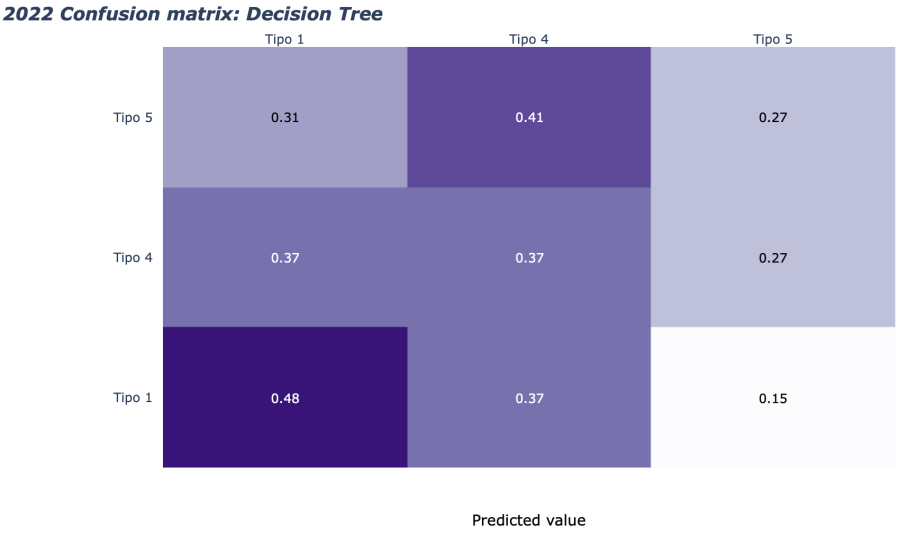


Gráfico 7. Matriz de confusión del backtesting del modelo 1.

El segundo modelo, el bosque aleatorio, mejora un poco más la predicción de tipo 1, pero sigue teniendo un poco de problemas para los demás tipos de infracciones:

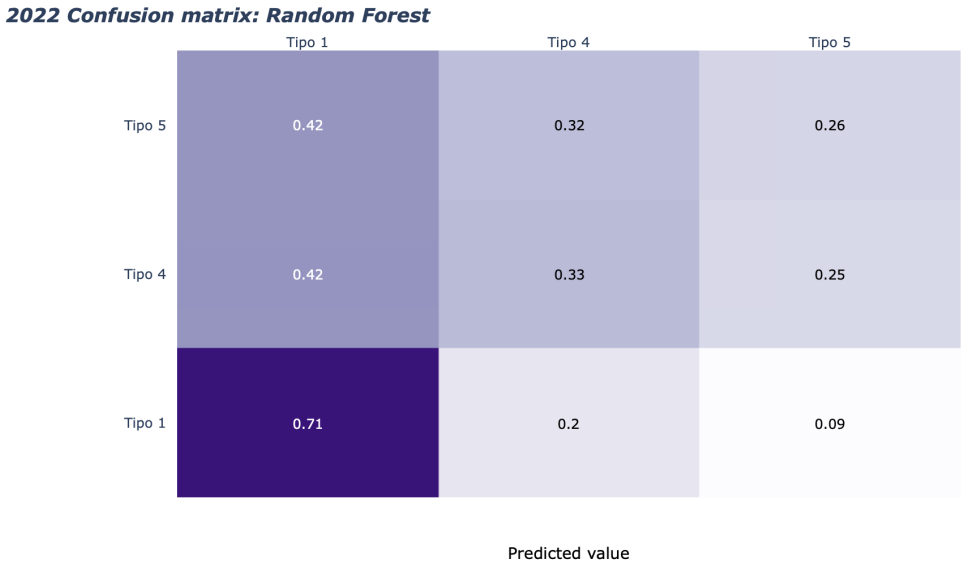


Gráfico 8. Matriz de confusión del backtesting del modelo 2.

Y finalmente, el tercer modelo, el gradient boosting, es el modelo que más equilibrado sale en esta prueba de predecir 2022:

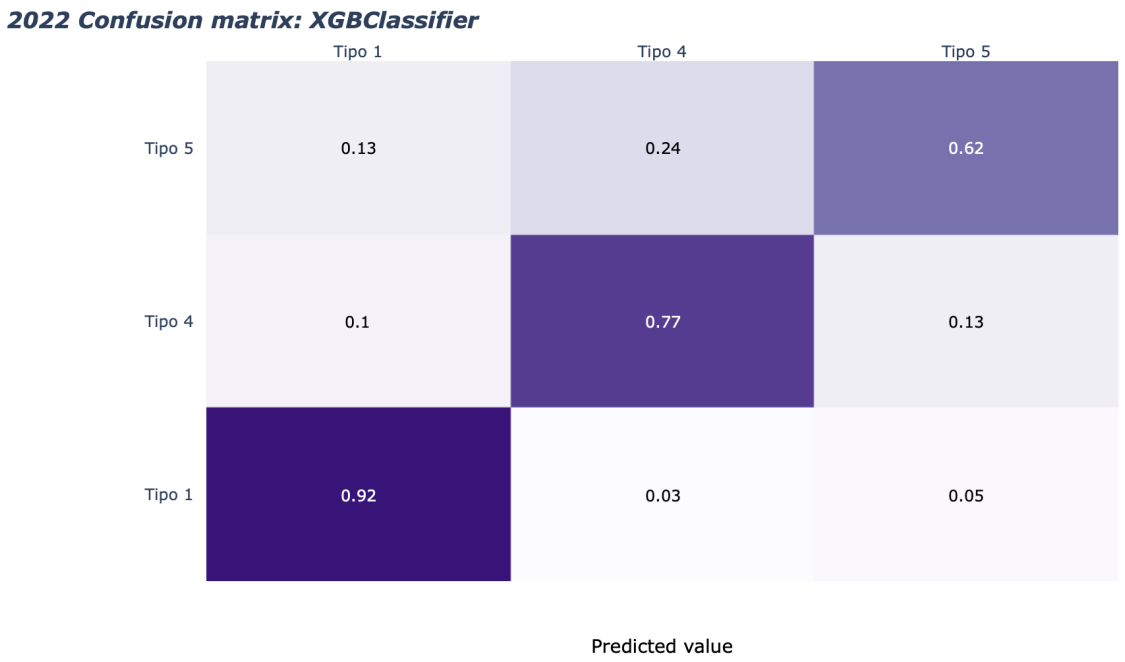


Gráfico 9. Matriz de confusión del backtesting del modelo 3.

Se puede observar que la matriz se encuentra mejor equilibrada y el modelo logra captar los diferentes tipos de infracciones, siendo el tipo 5 el peor desempeño, pero contando con un 62% de *accuracy*. Después de la evaluación de los tres modelos de clasificación, las métricas de evaluación y los mejores hiperparámetros, se eligió el modelo de **GRADIENT BOOSTING,** el cual tiene el mejor accuracy en el dataset de prueba, el mejor backtesting para 2022 y los hiperparámetros más adecuados para el problema que se presentó.

5) Evaluación

5.1 Evaluación de resultados: Entender e interpretar los resultados obtenidos, su impacto y utilidad, considerando los criterios de éxito del negocio.

Aún con las pocas variables que se obtuvieron permiso para utilizar en el modelo, ya que existía mucha información privada o sensible, se logró crear un modelo que logra identificar bastante bien el tipo de infracción, y se pudo observar con el backtesting de 2022. El impacto viene desde el número que existe de infracciones, donde se investigó el número de infracciones pagadas, y resultó ser menos del 10% de ambos años. Una de las principales utilidades, se puede ver en el dashboard, es identificar las calles y cruces con más infracciones en este último año y medio, y poder así, empezar a crear conciencia en los automovilistas, logrando reducir el número de infracciones en los siguientes meses.

5.2 Revisión del proceso: Sumarizar todo el proceso, principales problemas, posibles mejoras.

El proceso desarrollado en estas 7 semanas de trabajo, comenzó con una lluvia de ideas de diferentes problemáticas sociales en las cuales se pudiera implementar Inteligencia Artificial para poder lograr una posible solución. Se obtuvieron varias ideas, pero el siguiente problema fue la obtención de los datos para intentar resolver la problemática. Una vez obtenida una problemática y sus datos, se realizó una limpieza profunda de la base de datos obtenida, ya que, al ser la mayoría texto sin validación, podía tener valores duplicados, que la máquina detectaba como diferentes por el simple hecho de tener un espacio extra, o estar mal escrito. Posteriormente, se realizaron diferentes modelos para encontrar la solución con mejores métricas de evaluación y además, se probaron dichos modelos con diferentes hiperparámetros, incluyendo también el proceso de Cross-Validation. Finalmente, se realizó el deployment del modelo a un dashboard web, donde se incluyen diferentes análisis gráficos, que pueden ser interpretados y usados por quien los posee, ya que son datos abiertos. Con el uso del dashboard, los ciudadanos podrán identificar que días se infracciona más, a que hora y que zonas. Y, obviamente, se le busca dar un buen uso a estos datos, creando conciencia y se buscará reducir el número de infracciones con una buena cultura vial, no simplemente evitando los días con más infracciones o las zonas donde se suele infraccionar más, ya que eso no es una buena cultura vial.

Dentro de las posibles mejoras, estaría en la recolección de datos, o en el registro de las mismas, ya que hay cierta cantidad de registros que contenían nulos, o valores que no se podían reemplazar porque no se sabía a qué correspondía, y esto ayudaría a tener aún más datos y poder entrenar con ellos.

5.3 Impacto social principal.

Recordemos que existen dos mercados potenciales muy fuertes dentro de este proyecto, el primero son los conductores en busca de un lugar para estacionarse y el mercado es la Secretaría de Movilidad, que podrá obtener un resultado de cuales son las zonas, fechas y horas con más infracciones y poder proponer soluciones a dicho problema vial. Se cree que este primer desarrollo de la problemática puede ayudar a pensar en soluciones para el problema vial, principalmente de estacionamientos, donde participan los dos mercados principales identificados en el proyecto.

5.4 Impacto hacia los Objetivos de Desarrollo Sostenible.

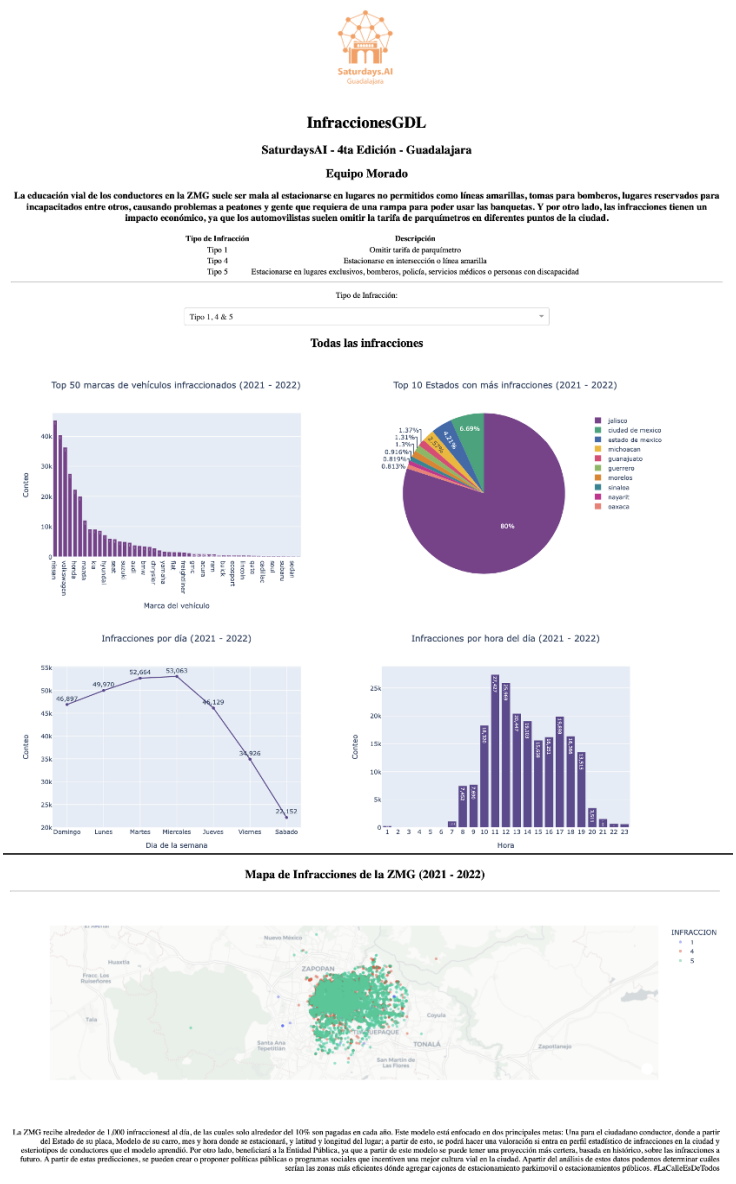
El desarrollo de este modelo de clasificación está alineado con los Objetivos de Desarrollo Sostenible de las Naciones Unidas, específicamente con el **objetivo número 4, una educación de calidad** y el **objetivo número 11: Ciudades y comunidades sostenibles.**

Imagen #. Los 18 Objetivos de Desarrollo Sostenible de la ONU.

Ya que el desconocimiento o negligencia de los lugares correctos para estacionarse tiene un efecto negativo en los peatones y automovilistas, las leyes no tienen un efecto deseado y los automovilistas reinciden cometiendo infracciones degradando en ciertas zonas la movilidad peatonal.

6) Despliegue

6.1 Descripción del prototipo funcional.



Como despliegue de prototipo funcional se decidió realizar un Dashboard para que los dos mercados principales, automovilistas y agentes viales, puedan consultar la información histórica desde 2021 hasta agosto de 2022 de las infracciones. En este dashboard se puede encontrar primeramente información general del equipo que lo desarrolló, donde se observa también el logotipo de Saturdays AI como encabezado, seguido de una pequeña descripción de la problemática que se buscó solucionar en este proyecto. Seguido, se ven los tres tipos de infracciones que se utilizaron como target para este modelo, y la explicación de cada una de las infracciones que se aplican en la ZMG, la cuales se pueden filtrar en las siguientes 4 principales gráficas del Dashboard: Top 50 marcas de vehículos infraccionados, Top 10 de Estado de la placa del vehículo infraccionado, Número de infracciones por día de la semana y Número de infracciones por hora del día. Y finalmente, se encuentra un mapa interactivo donde está registrado geográficamente cada una de las más de 305,000 infracciones que se han registrado a través del sistema que se tiene en la ciudad. Se puede seleccionar los tres tipos de infracciones, o bien seleccionar una sola y ver las mismas gráficas filtradas para el tipo de infracción seleccionado. Como se mencionó antes, este dashboard puede ser de gran ayuda para los automovilistas de ubicar las zonas donde se aplican regularmente las infracciones de omitir tarifas, buscando así mejorar la educación vial e implementar las buenas prácticas de evitar no pagar un parquímetro, o asimismo ubicar las zonas donde normalmente se están infraccionado por estacionarse en lugares donde no está permitido y así evitar hacerlo. Por otro lado, los agentes viales y la secretaría correspondiente pueden ubicar estas zonas donde hacen falta estacionamientos para los automovilistas y poder llegar así a soluciones en conjunto y lograr una mejor cultura vial. Este dashboard se encuentra disponible en el siguiente link: https://infraccionesgdl.herokuapp.com/

7) Recomendaciones

7.1 Recomendaciones al negocio.

Como se mencionó anteriormente, se recomienda al negocio, en este caso a los oficiales viales, o encargados del registro de las infracciones, tener un mejor registro de las mismas ya que se facilita demasiado cualquier trabajo relacionado con sus datos. De no ser posible, se pudiera implementar un tipo drop-down en cada una de las categorías, donde se pueda elegir el modelo de carro, el estado de la placa del vehículo por ejemplo. Además, si la aplicación utiliza datos geolocalizables, podría de manera automática detectar la latitud y longitud de la infracción. Todo esto, facilitará en un futuro el manejo de la base de datos y futuras incorporaciones de datos al modelo ya creado.

7.2 Recomendaciones técnicas.

Ser muy cautelosos y cuidadosos con los datos, ya que la limpieza fue uno de los mayores problemas y retos enfrentados. Como se mencionó antes, se decidió recortar los datos a los que tuvieran más de 50 repeticiones en marca, ya que existen datos que no fueron limpiados, simplemente fueron excluidos por la poca repetición (no fue una muestra significativa, fue menos de un 5%, pero no significa que puedan estos datos no ayudar al modelo para su mejoramiento). Además, se recomienda ampliar la base de datos con más variables transformadas, puede ser por hora laborable no laborable, buscar incluir flags de días festivos o días de asueto, y explorar si estos días atípicos tienen algún impacto en las multas.

8) Siguientes pasos.

- Como todo proyecto, el siguiente paso es seguir iterando, mejorar un poco los accuracies, en especial del tipo 5 de infracción.
- Ir al campo de recolección de datos y realizar investigaciones de los dos mercados principales que se encontraron en el proyecto:
    - Evaluar el conocimiento y cultura vial de los automovilistas, ya que muchos son infraccionados por la poca cultura y conciencia vial.
    - Evaluar el desempeño de los agentes viales, y comenzar a buscar, en conjunto, soluciones para reducir el número de infracciones.
- Dar un seguimiento profundo al número de infracciones y número de infracciones pagadas, ya que no necesariamente por mayor número de infracciones, mayor recolección de impuestos a través de infracciones pagadas.
- Lograr implementar el deployment del modelo en el dashboard. Implementar un predictor, donde se puedan insertar variables, como la latitud, longitud, marca y modelo del vehículo propio, y obtener qué tipo de infracción es más probable obtener.
- Agrandar las variables dependientes. Como se menciona anteriormente, se puede buscar crear un calendario de días festivos y cruzar este flag a la base de datos, para posteriormente analizar su impacto.
- Crear un modelo de probabilidades, donde se obtenga cual es la probabilidad de cada una de las infracciones dependiendo las variables dependientes que se le den al modelo, ya que únicamente predice cuál será la infracción, pero no cuáles serán las probabilidades de cada tipo de infracción.

9) Fuentes bibliográficas en formato APA.

Amat Rodrigo, J. (2020) *Gradient Boosting con Python.* Recuperado de: https://www.cienciadedatos.net/documentos/py09_gradient_boosting_python.html

Amazon Web Services (2022) *¿Qué es una red neuronal?.* Recuperado de: https://aws.amazon.com/es/what-is/neural-network/

Charming Data (2022) *Easiest Way to Deploy a Dash App to the Web.* Recuperado de: https://www.youtube.com/watch?v=Gv910_b5ID0

Chauman, N. (2020) *Métricas De Evaluación De Modelos En El Aprendizaje Automático*. Recuperado de: https://www.datasource.ai/es/data-science-articles/metricas-de-evaluacion-de-modelos-en-el-aprendizaje-automatico

Gonzalez, L. (2019) *Regresión Logística – Teoría.* Recuperado de: https://aprendeia.com/algoritmo-regresion-logistica-machine-learning-teoria/

Hossack, A. (2022) *DashTools - Plotly Dash Command Line Tools - Create, Run and Deploy Templated Python Apps from Terminal.* Recuperado de: https://github.com/andrew-hossack/dash-tools

Hossack, A. (2022). *Interactive data analytics.* Recuperado de: https://github.com/Coding-with-Adam/Dash-by-Plotly/

Martinez Heras, J. (2020) *Árboles de Decisión con ejemplos en Python.* Recuperado de: https://www.iartificial.net/arboles-de-decision-con-ejemplos-en-python/

Martinez Heras, J. (2020) *Precision, Recall, F1, Accuracy en clasificación.* Recuperado de: https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/

Martinez Heras, J. (2020) *Random Forest (Bosque Aleatorio): combinando árboles.* Recuperado de: https://www.iartificial.net/random-forest-bosque-aleatorio/

Martinez Heras, J. (2020) *Regresión Lineal: teoría y ejemplos en Python.* Recuperado de: https://www.iartificial.net/regresion-lineal-con-ejemplos-en-python/

Martinez Heras, J. (2020) *Regresión Polinómica en Python con scikit-learn.* Recuperado de: https://www.iartificial.net/regresion-polinomica-en-python-con-scikit-learn/

Plotly (2022). *Plotly: Low-Code Data App Development.* Recuperado de: https://plotly.com/

Roman, V. (2019) *Algoritmos Naive Bayes: Fundamentos e Implementación.* Recuperado de: https://medium.com/datos-y-ciencia/algoritmos-naive-bayes-fudamentos-e-implementaci%C3%B3n-4bcb24b307f

UNDP (2015). *Objetivos de Desarrollo Sostenible | Programa de las Naciones Unidas para el Desarrollo.* Recuperado de: https://www.undp.org/es/sustainable-development-goals