Optimización de la Seguridad Vial y Eficiencia de Conducción en Entornos Urbanos- Parte final

Un Análisis de Datos de Colisiones para la Innovación de Tecnologías de Vehículos Autónomos de Tesla

- NICOLÁS SAMUEL MARTIN VASQUEZ
- JUAN DAVID LOPEZ BECERRA
- JUAN DIEGO GONZÁLEZ JIMENEZ

Recordatorio

“En la era de la movilidad inteligente, Tesla se une a (Nuestra Consultora) para analizar los datos de colisiones de vehículos en Nueva York. El objetivo es optimizar las tecnologías de conducción autónoma y mejorar la seguridad y eficiencia en entornos urbanos, basándose en el análisis del conjunto de datos "Motor Vehicle Collisions - Vehicles" del NYPD.”

Entendimiento del Negocio

Desafíos de la Movilidad Urbana

- **Contexto:** Aumento de la congestión y los accidentes viales en ciudades densas.
- **Tesla:** Líder en la revolución de los vehículos eléctricos y autónomos.
- **Oportunidad:** Mejorar la seguridad vial y la eficiencia de la conducción en entornos urbanos

Colaboración Estratégica

- **Alianza:** Tesla y ACPR unen esfuerzos.
- **Objetivo:** Análisis de datos para optimizar tecnologías de conducción autónoma.
- **Enfoque:** Seguridad, eficiencia y experiencia del usuario.

Metas y Expectativas

- Análisis de patrones y factores de riesgo en accidentes viales.
- Desarrollo de algoritmos avanzados para la conducción autónoma.
- Implementación de rutas de prueba seguras y eficientes en Nueva York.

Selección de los Datos a Utilizar

Fuente de Datos Críticos

- Conjunto de datos: "Motor Vehicle Collisions - Vehicles" del NYPD.
- Cobertura: Detalles sobre cada vehículo involucrado en colisiones en la ciudad de NY.
- Relevancia: Información clave para adaptar tecnologías de Tesla.

Aplicaciones del Análisis de Datos

- Identificación de tendencias en accidentes viales.
- Afinamiento de sistemas de conducción autónoma y asistencia al conductor.
- Diseño de estrategias de prevención de accidentes urbanos.

Potencial de los Datos

- Información detallada para una planificación de rutas de prueba efectiva.
- Contribución a la mejora continua de las soluciones de movilidad de Tesla.
- Impacto en la reducción de accidentes y la promoción de una conducción segura.

Colección y Descripción de Datos

Procedimiento de Colección de Datos

- Ambiente de trabajo: Databricks para el manejo eficiente de grandes volúmenes de datos.
- Proceso: Carga, limpieza y preprocesamiento de los datos para análisis.

Características de los Datos

- Variedad: Datos numéricos, de texto, fecha y hora.
- Atributos clave: Tipo de vehículo, dirección de viaje, factores contribuyentes.
- Profundidad: Información precisa sobre el momento y circunstancias de cada colisión.

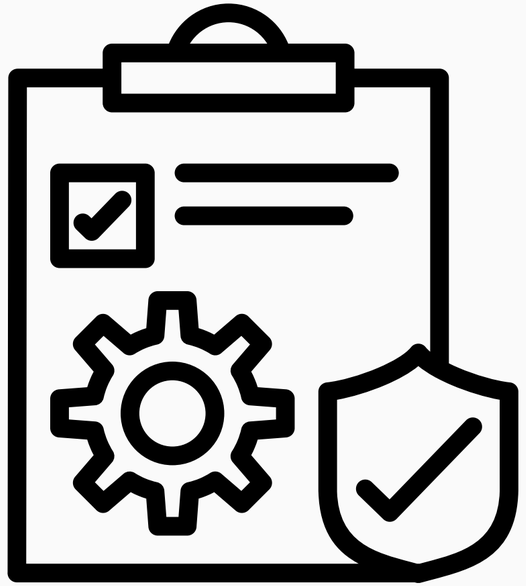
Análisis y Visualización de Datos

- Herramientas: Uso de técnicas estadísticas y de machine learning para extraer insights.
- Visualización: Representación gráfica de patrones y tendencias.

Reporte de Calidad de Datos

Técnicas propuestas para tratar valores faltantes:

- Eliminación de filas o columnas
- Imputación de valores
- Utilización de modelos de imputación
- Codificación de valores faltantes

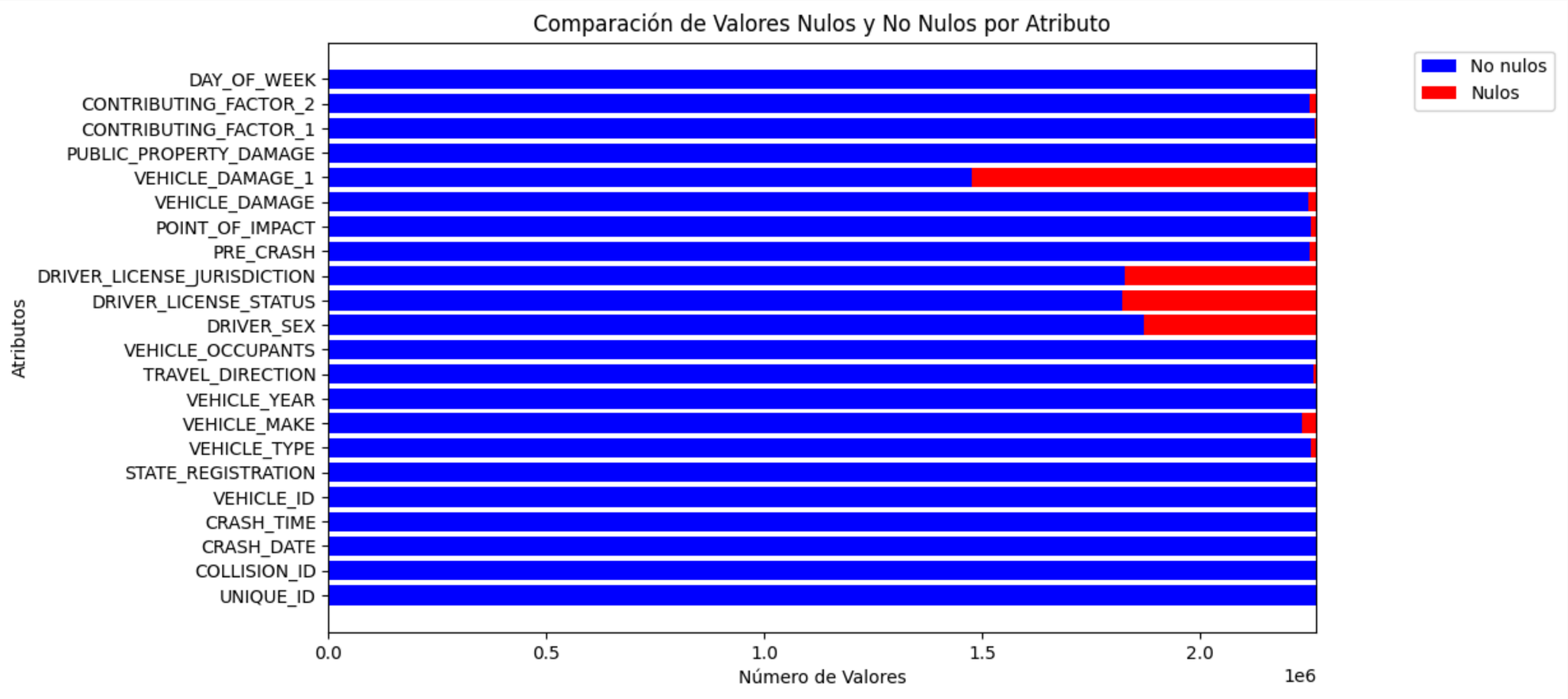


Filtros, Limpieza y Transformación Inicial

Por el momento tratamos de disminuir la dramática cantidad de valores nulos. Para esto tomamos algunas consideraciones:

- **Filtrado de valores atípicos:** Eliminación de años de vehículos anómalos > 2024.
- **Imputación de valores faltantes:** Uso de mediana y valor más frecuente.
- **Eliminación de columnas:** Remoción de columnas con >50% de datos faltantes.

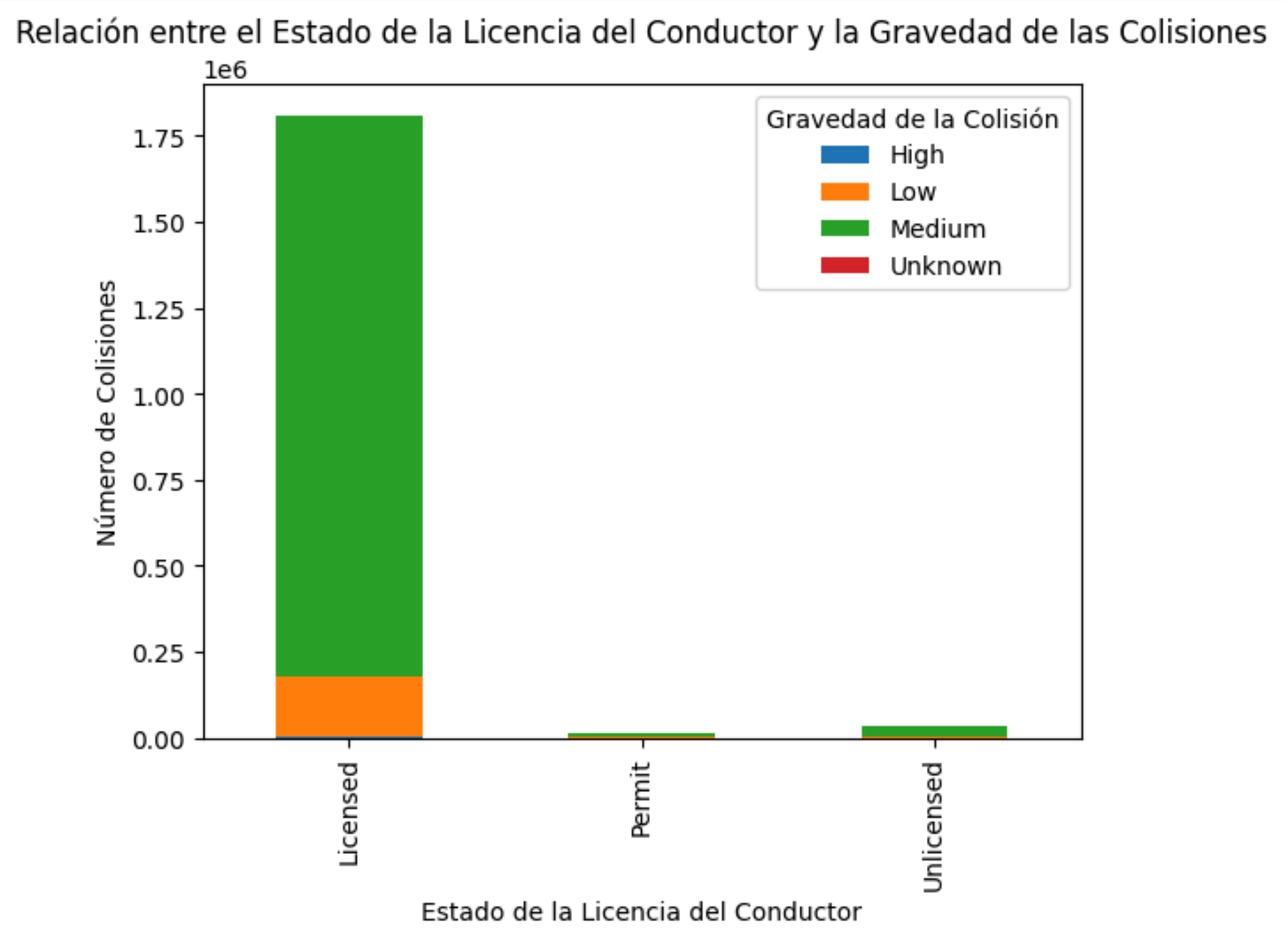
Cambio Dramático en la Cantidad de Valores Nulos.



Solución a preguntas problema

A continuación se presentaran las preguntas y sus soluciones

1.¿Hay alguna relación entre el estado de la licencia del conductor y la gravedad de las colisiones en las que están involucrados?

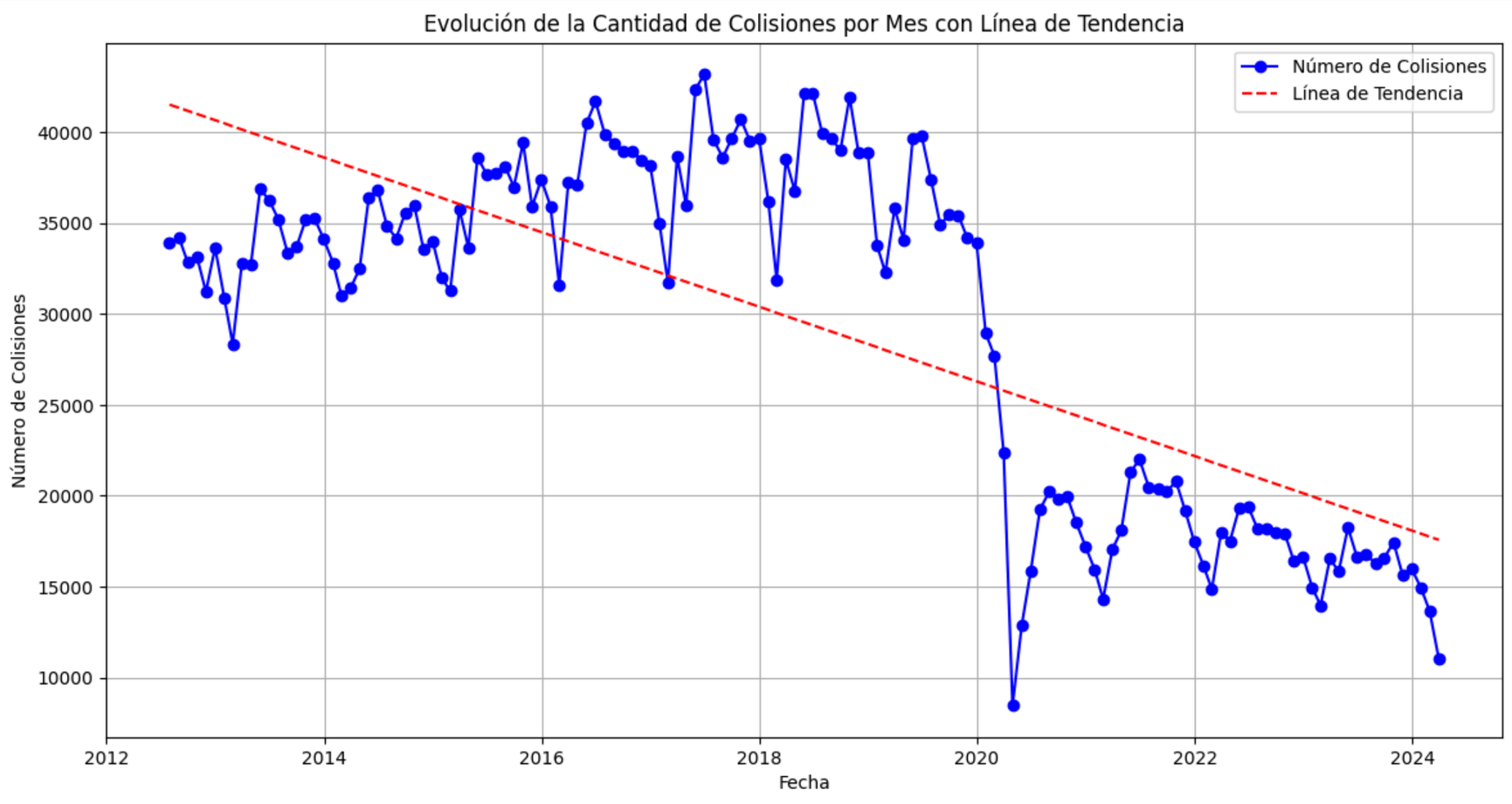


COLLISION_SEVERITY	
Medium	2158969
Unknown	1723937
Low	276126
High	6891

Se comparo el estado de la licencia con el numero de colisiones agrupado por gravedad de colisión

- La **gravedad media** son los mas **comunes.**
- Los que **tienen licencia** son los que **más colisiones** presentan.

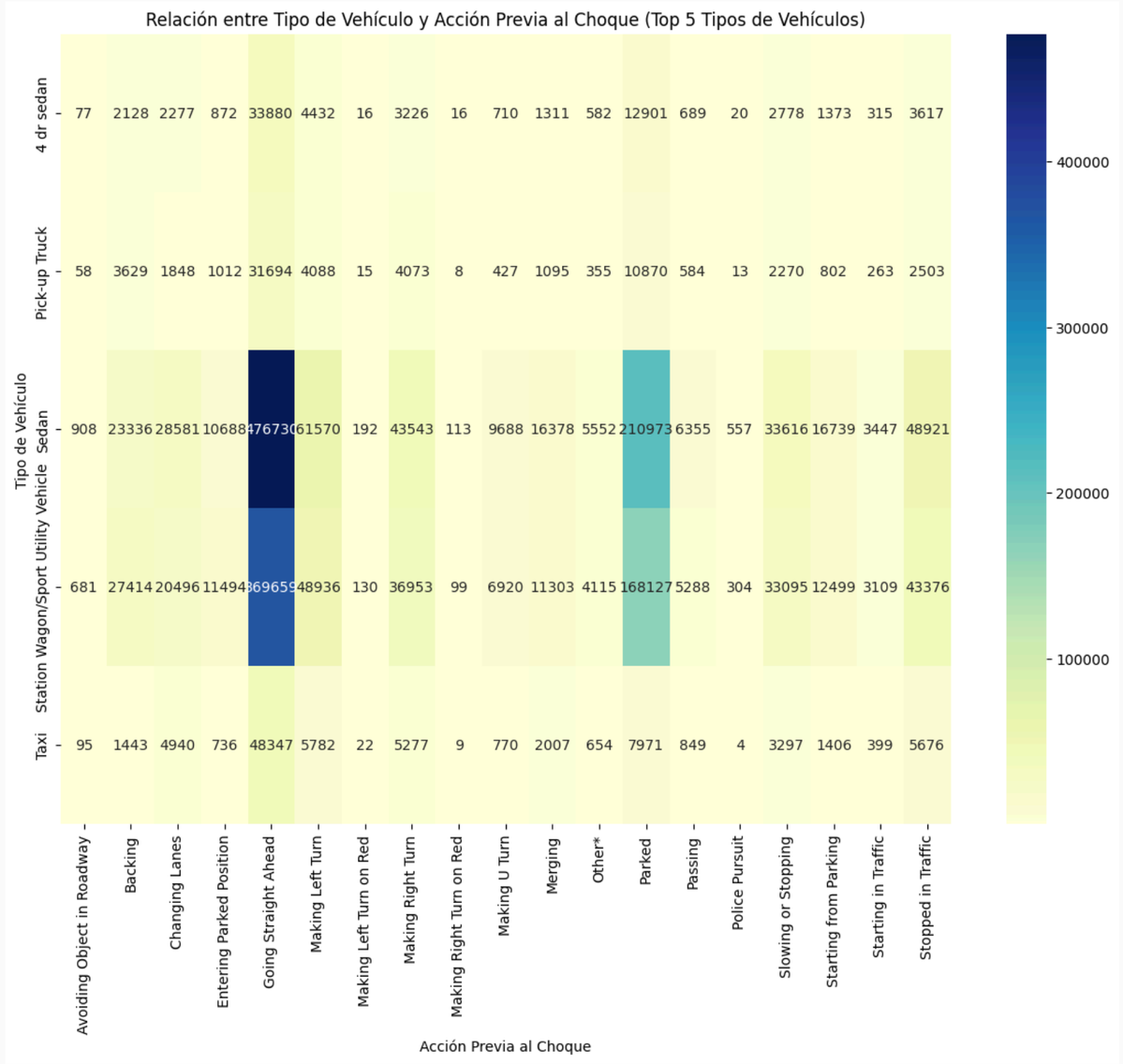
2. ¿Cómo ha evolucionado la cantidad de colisiones a lo largo del tiempo? ¿Hay alguna tendencia visible?



Se hizo un gráfico de líneas con tendencia comprando colisiones entre 2012 y 2024

- Se presenta una **tendencia negativa.**
- La **pandemia** (2020) presento **un pico negativo.**
- **Mayor numero** de colisiones a mediados de **2017**

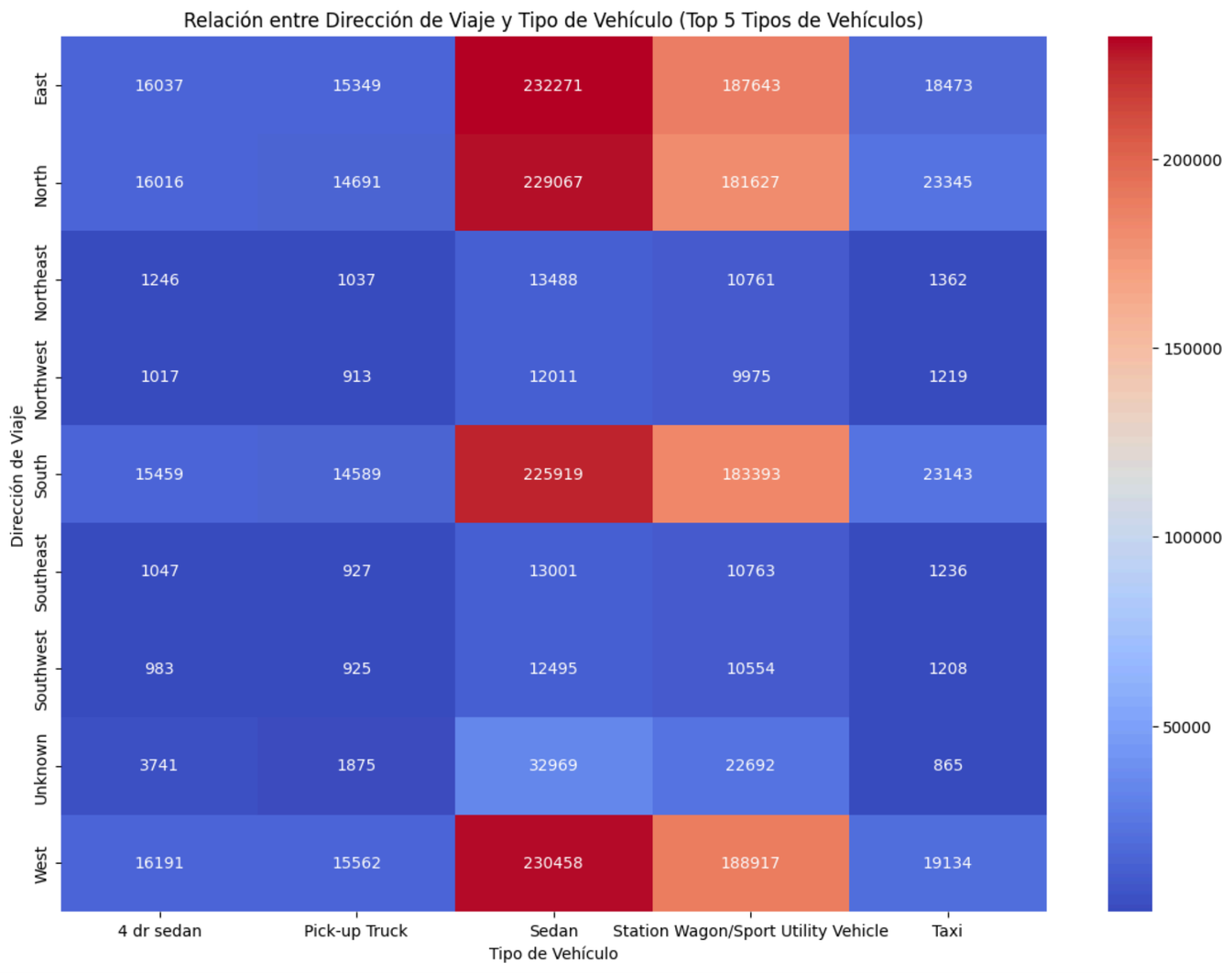
3. ¿Cómo podríamos utilizar los datos de colisiones para mejorar los algoritmos de conducción autónoma de Tesla?



Se hizo un mapa de calor que representa la relación entre tipo de vehículo y acción previa al choque

- **Mayor** número de colisiones **476.730** en sedan Yendo recto
- **Menor** número **4** un taxi en una persecución policial
- Otro nivel alto se presenta parqueando

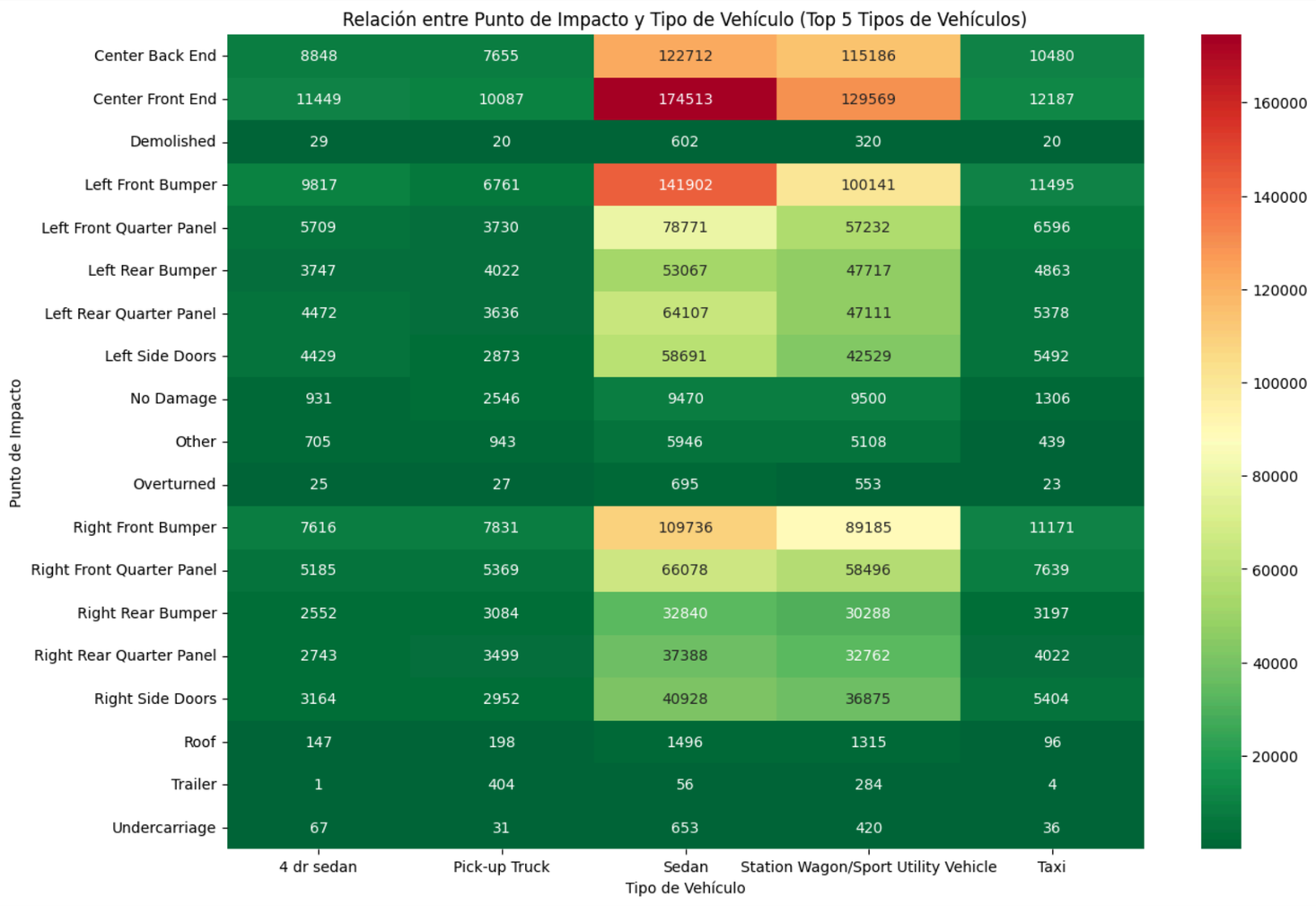
3. ¿Cómo podríamos utilizar los datos de colisiones para mejorar los algoritmos de conducción autónoma de Tesla?



Se hizo un mapa de calor que representa la relación entre tipo de vehículo y la dirección hacia donde se dirigían

- El **más alto** fue **232.271** colisiones de sedanes que iban hacia el **este.**
- El **más bajo** fue de **865** colisiones de un **taxi** que se dirigía hacia el **oeste.**
- El **Sedan** es el que presenta **MÁS** colisiones

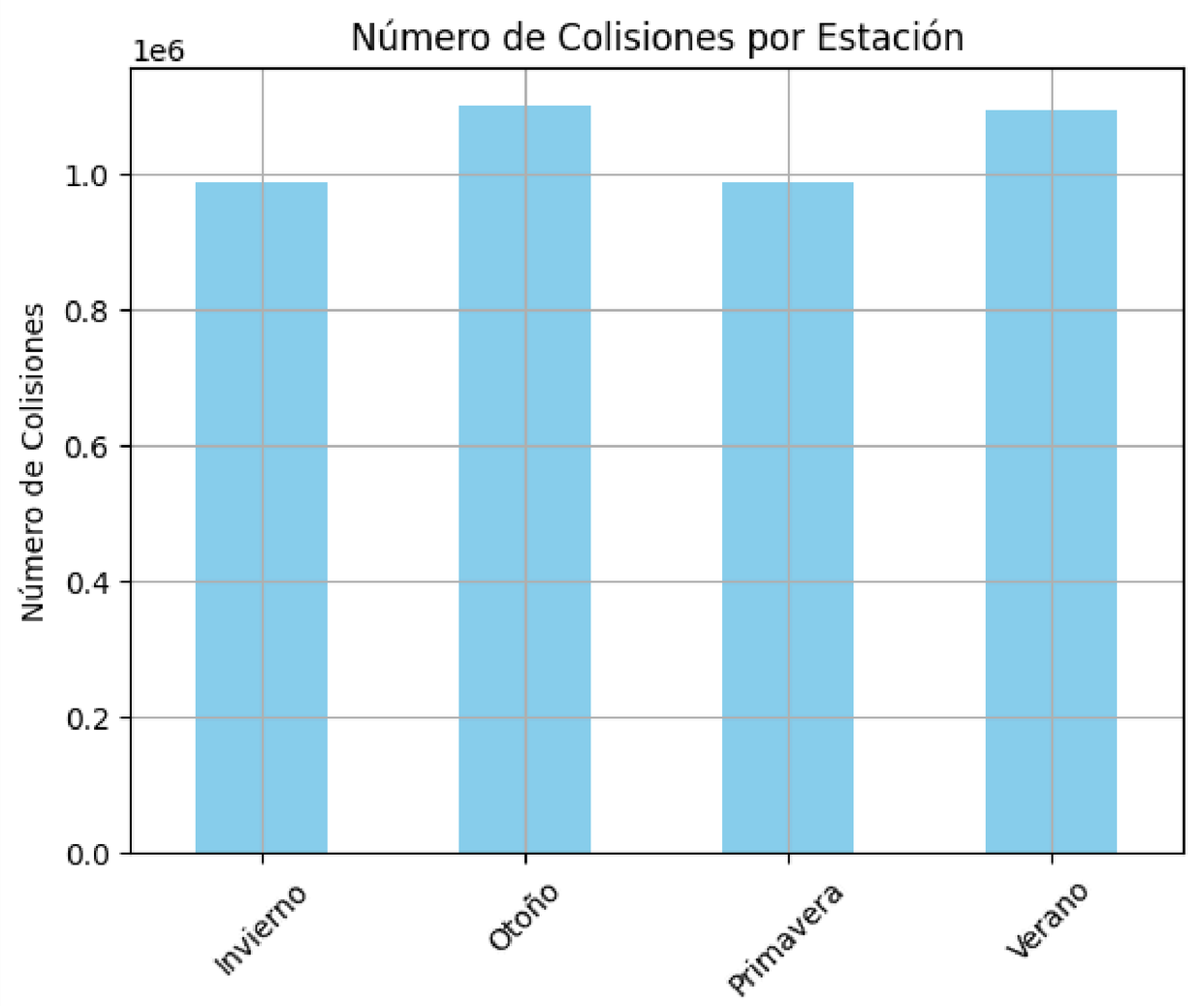
3. ¿Cómo podríamos utilizar los datos de colisiones para mejorar los algoritmos de conducción autónoma de Tesla?



Se hizo un mapa de calor que representa la relación entre tipo de vehículo y el puesto donde fue el impacto de la colisión

- El **más alto** fue de **174.513** colisiones de un sedan y **en el extremo delantero frontal**
- Se presentan **más colores verdes** por lo que **menos** cantidad de **colisiones**
- Se sigue tendencia del Sedan

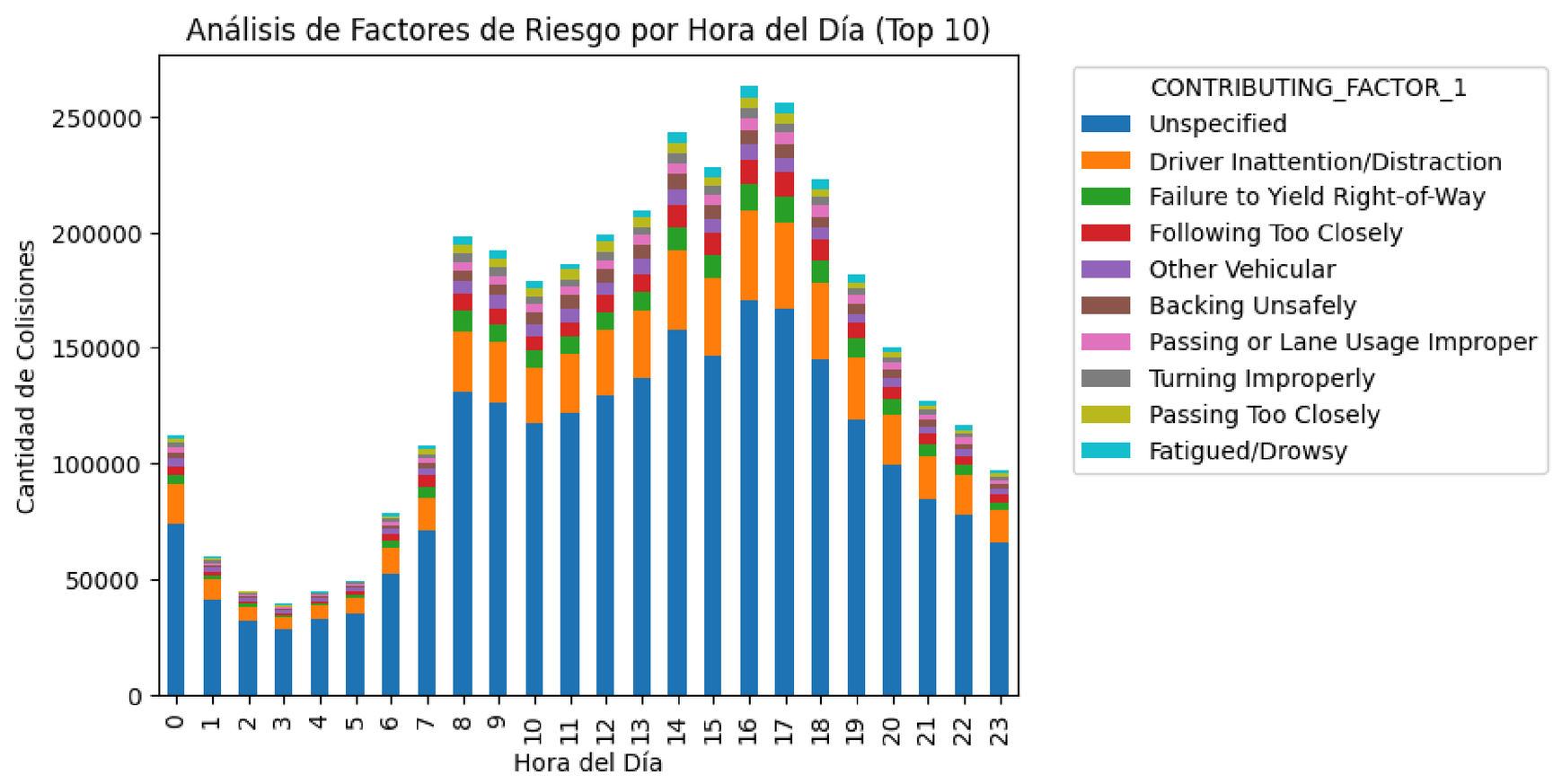
4. ¿Podríamos identificar patrones estacionales en las colisiones de vehículos? Por ejemplo, ¿hay más colisiones en invierno debido a las condiciones de la carretera?



Se hizo filtros para que se cogieran las estaciones del año y se comparo por número de colisiones

- El **invierno** es de los indicadores **más bajos** porque salen **menos cantidad de vehículos.**
- **El verano y el otoño son los más altos,** esto se puede interpretar por **aumento de viajes**
- No hay diferencia abismales.

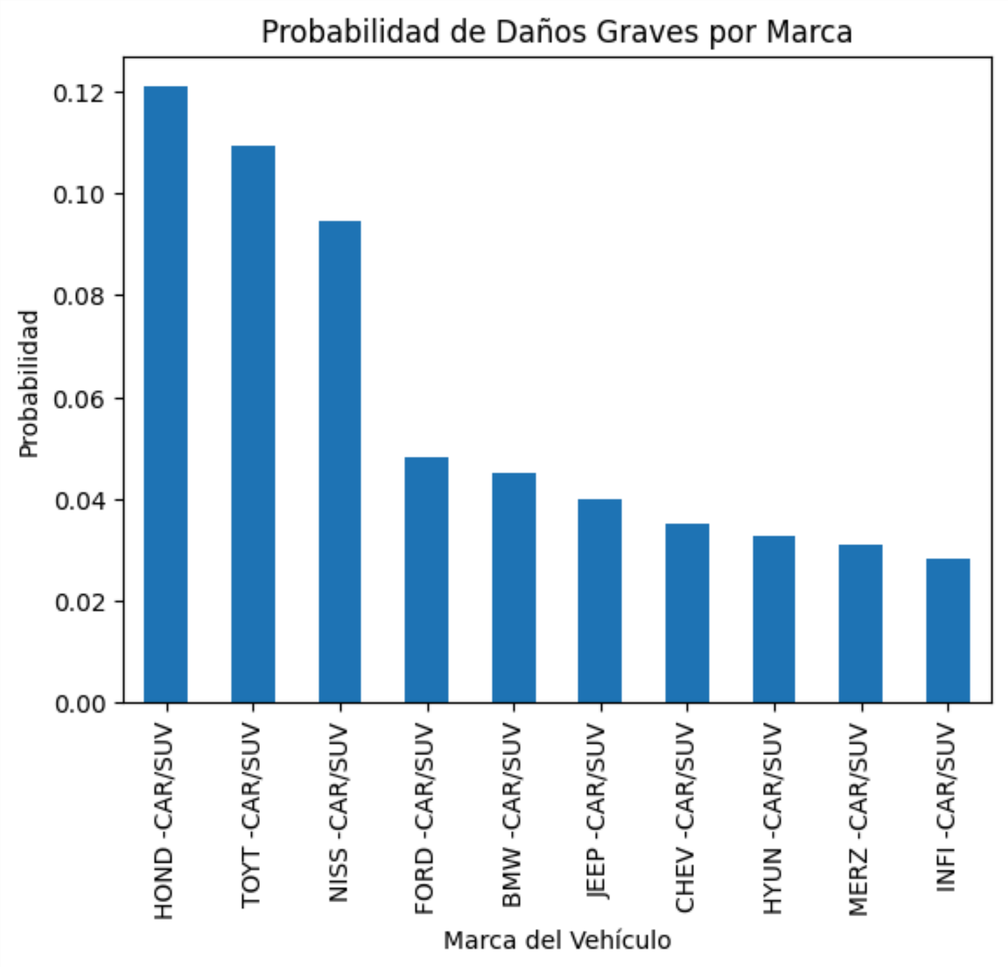
5. ¿Podríamos utilizar los datos de colisiones para informar a los conductores de Tesla sobre los factores de riesgo más comunes y en que horario pasan?



Se hizo un gráfico de barras apilada entre la hora del día y el número de colisiones por factor que produjo la colisión

- Las horas donde **más** ocurren **colisiones** es entre **2pm y 6pm.**
- **La distracción** es el factor **más común.**
- **Pico a las 5pm,** la mayoría sale del trabajo

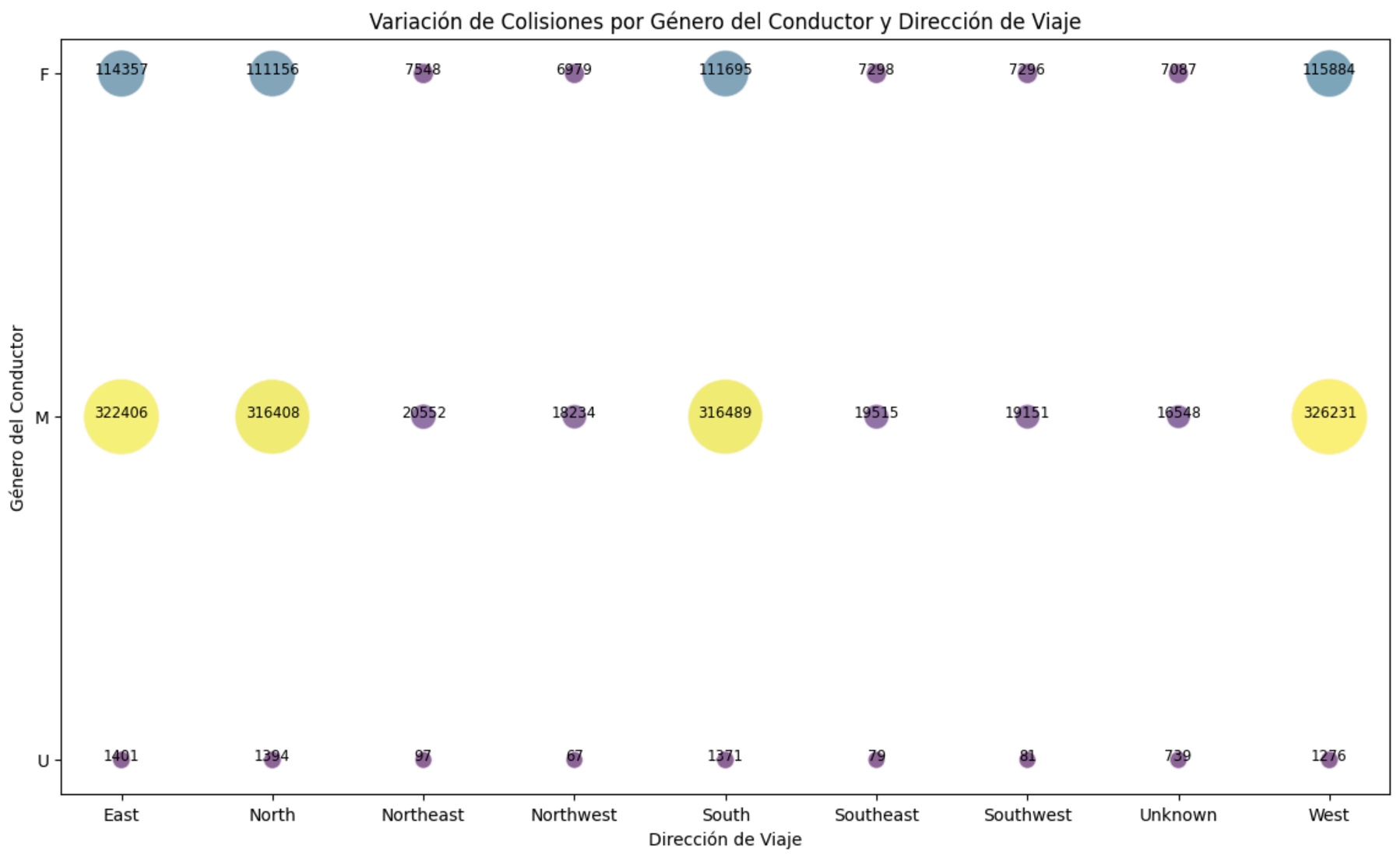
6. ¿Cuál es la probabilidad de que un vehículo con daños graves en una colisión sea de cierta marca o modelo?



Se calculo la probabilidad de que marcas y tipo de vehículo es mas probable de tener un daño grave

- Con la probabilidad de 0.12 la marca **Honda** es la más probable.
- Una marca como **infinity** aparece y es interesante

7. ¿Cómo varía la cantidad de colisiones vehiculares según el género del conductor y la dirección de viaje del vehículo?



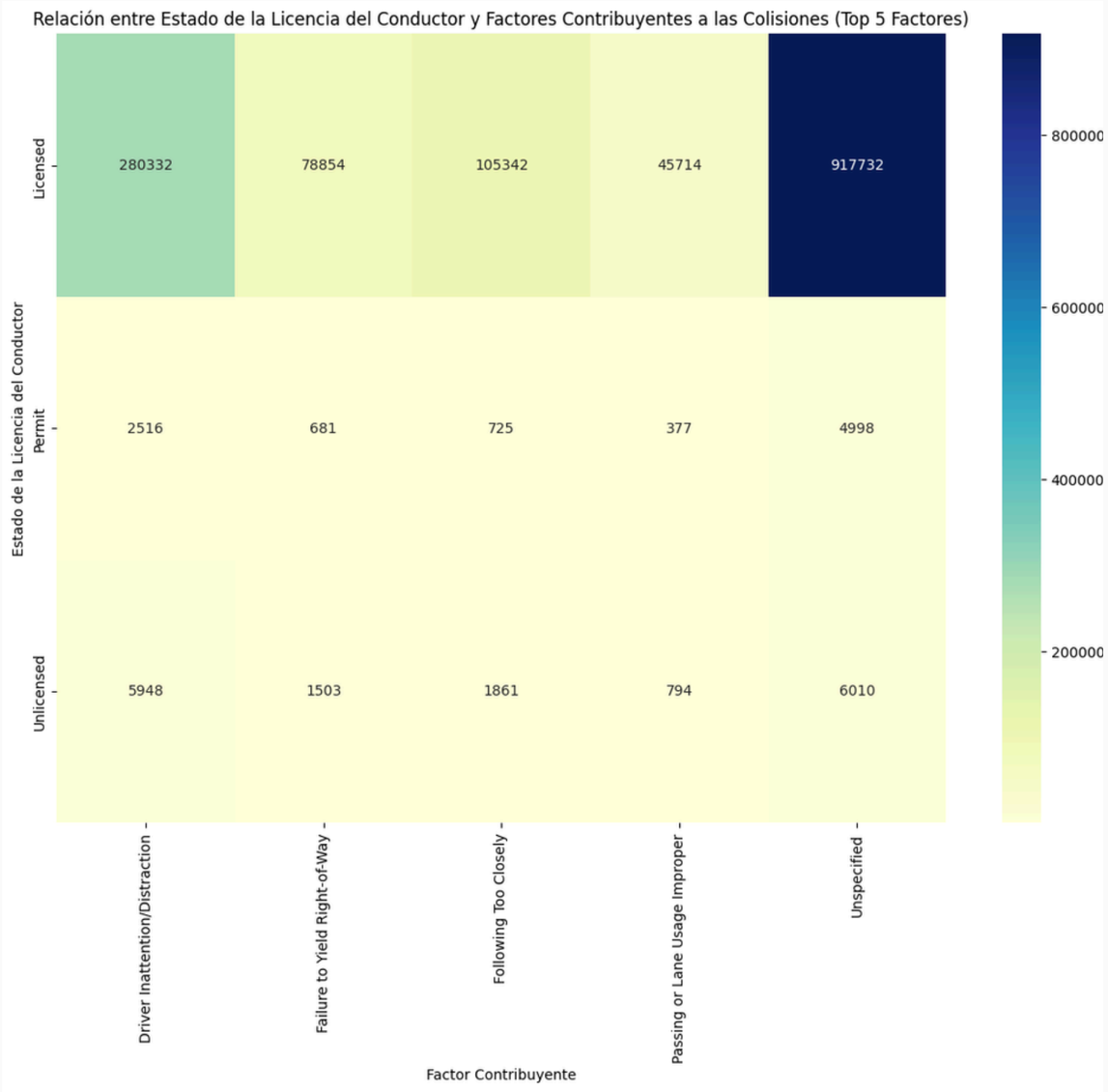
Un grafico de burbujas comprando el genero y hacia donde se dirigía el vehículo por número de colisiones

- El sexo masculino presenta mas colisiones.
- La cantidad de colisiones **mayor** es de **326.231** de un **conductor hombres** que se dirigan hacia el **Oeste**

8. ¿Cómo se distribuyen las colisiones vehiculares según el estado de la licencia del conductor y los factores que contribuyen a las colisiones?

Mapa de calor que representa la relación entre estado de licencia de conducción con el factor que provoco la colisión

- El número de colisiones **más alto 917.732** es de personas **con licencia .**
- Interesante que los que no tiene licencia no presentan tantas colisiones
- La **distracción** es el factor que **más colisiones tiene.**



Selección técnicas de Machine Learning

Aprendizaje No Supervisado: K-Means Clustering

- Técnica de **agrupamiento que segmenta los datos** en **grupos** basados en **similitudes.**
- El algoritmo K-Means **asigna cada dato a uno de los K clusters iniciales,** recalculando las posiciones de los centroides y repitiendo el proceso **hasta que los clusters se estabilicen.**
- **Ideal** para grandes **volúmenes de datos**

Aprendizaje Supervisado: Árboles de Decisión

- **Modelo** que utiliza muchos **arboles decisión**
- Agrega **secuencialmente árboles** de decisión a un modelo, cada uno **corrigiendo los errores de predicción del anterior.**
- Ofrece gran precisión.

Entrenamiento y Evaluacion de Modelos de Machine Learning

Aprendizaje No Supervisado: K Means

En este caso el parámetro que queríamos variar fue: **Numero de Clusters.**

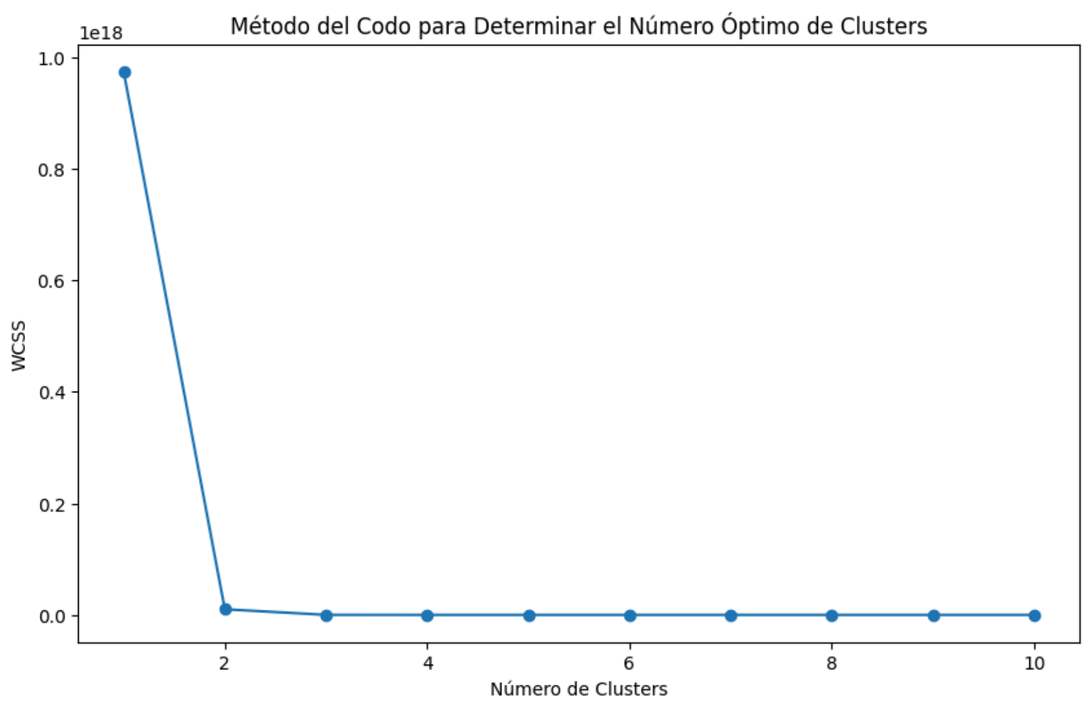
Para evaluar el rendimiento del modelo vamos a usar: **WCSS**

WCSS significa "Within-Cluster Sum of Squares" (Suma de Cuadrados Dentro del Cluster). La idea detrás de **WCSS es medir cuánto se desvían los puntos de datos dentro de un mismo cluster con respecto al centroide del cluster**, que es el punto promedio de todos los puntos en ese cluster.

Aprendizaje No Supervisado: K Means

- **Selección de Características:** Se eligieron 'VEHICLE_YEAR', 'VEHICLE_OCCUPANTS', y 'CONTRIBUTING_FACTOR_1' por su potencial para revelar patrones en la gravedad y las causas de las colisiones.
- **Preprocesamiento:** Las variables categóricas fueron transformadas mediante one-hot encoding para adecuarlas al modelo de K-Means, y los datos fueron estandarizados para normalizar la escala de las características numéricas, evitando así sesgos en la formación de clusters.
- **Optimización del Modelo:** Utilizando el método del codo, se determinó que dos clusters son óptimos, permitiendo una clara distinción entre tipos de accidentes sin sobreajustar el modelo a los datos.

Aprendizaje No Supervisado: K Means



1. **Punto de Codo Aparente**: El número óptimo de clusters es 2, donde el descenso en WCSS se estabiliza significativamente.
2. **Estabilización del WCSS**: Más allá de 2 clusters, el WCSS muestra poca variación, indicando mínima mejora en la separación de los datos.

Aprendizaje Supervisado: Árboles de Decision

- **Selección de Características:** Se incluyeron datos como el tipo de vehículo, dirección de viaje, y factores contribuyentes para capturar la complejidad y diversidad de las situaciones de conducción en Nueva York.

- **División de Datos:** El conjunto de datos fue dividido en un 70% para entrenamiento y un 30% para pruebas, asegurando tanto el entrenamiento robusto del modelo como una evaluación precisa de su rendimiento.

- **Entrenamiento y Limitaciones del Modelo:** El modelo se entrenó con una profundidad máxima limitada para evitar el sobreajuste y garantizar que las decisiones del modelo se basen en patrones genuinos y no en anomalías de los datos.

Aprendizaje Supervisado: Árboles de Decisión

Predecir: DAÑO A PROPIEDAD PUBLICA

Rendimiento:

Categoría	Precisión	Recall	F1-Score	Soporte
No Daño (N)	0.94	1.00	0.97	192,066
Indefinido	0.82	0.00	0.00	10,636
Daño (Y)	1.00	0.00	0.00	1,284

- **Alta Precisión en 'No Daño (N)':** 94% de precisión y 100% de recall, indicando una excelente identificación de incidentes sin daño a la propiedad.
- **Desafíos en Categorías 'Daño (Y)' y 'Indefinido':** A pesar de la alta precisión, el recall de 0% muestra que el modelo no identificó correctamente casos reales de daño.
- **Necesidad de Mejoras:** El bajo recall en daños a la propiedad sugiere la necesidad urgente de mejorar la sensibilidad del modelo para capturar todos los casos relevantes.

Bono

Entrenamiento Red Neuronal

En este caso utilizamos un MLP (Multilayer Perceptron) simple para clasificar el **point of impact.**

```
# Modelo de red neuronal
model = Sequential()
model.add(Dense(256, activation='relu', input_shape=(X_train_transformed.shape[1],)))
model.add(Dense(128, activation='relu'))
model.add(Dense(y_train.shape[1], activation='softmax'))  # Capa de salida con activación softmax para clasificación multiclase

# Compilar el modelo
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Entrenar el modelo
history = model.fit(X_train_transformed, y_train, epochs=10, batch_size=32, validation_split=0.2, verbose=1)

# Evaluar el modelo en el conjunto de prueba
loss, accuracy = model.evaluate(X_test_transformed, y_test, verbose=0)
print(f'Test Accuracy: {accuracy:.2f}')
```

Capas: Una de entrada, dos ocultas y la de salida.
Train/Test Split: 80% Train, 20% Split.
Optimizer: Adam
Loss: Categorical Crossentropy

Rendimiento:

```
Epoch 1/10
112/112 [==============================] - 7s 18ms/step - loss: 2.1063 - accuracy: 0.3980 - val_loss: 1.4471 - val_accuracy: 0.6689
Epoch 2/10
112/112 [==============================] - 1s 11ms/step - loss: 1.0621 - accuracy: 0.7352 - val_loss: 1.0060 - val_accuracy: 0.7248
Epoch 3/10
112/112 [==============================] - 2s 14ms/step - loss: 0.8014 - accuracy: 0.7702 - val_loss: 0.9260 - val_accuracy: 0.7360
Epoch 4/10
112/112 [==============================] - 1s 11ms/step - loss: 0.7058 - accuracy: 0.7881 - val_loss: 0.9331 - val_accuracy: 0.7338
Epoch 5/10
112/112 [==============================] - 2s 17ms/step - loss: 0.6463 - accuracy: 0.7974 - val_loss: 0.9472 - val_accuracy: 0.7204
Epoch 6/10
112/112 [==============================] - 2s 15ms/step - loss: 0.5949 - accuracy: 0.8102 - val_loss: 0.9658 - val_accuracy: 0.7204
Epoch 7/10
112/112 [==============================] - 2s 17ms/step - loss: 0.5500 - accuracy: 0.8273 - val_loss: 1.0151 - val_accuracy: 0.7103
Epoch 8/10
112/112 [==============================] - 1s 12ms/step - loss: 0.5183 - accuracy: 0.8391 - val_loss: 1.0035 - val_accuracy: 0.6991
Epoch 9/10
112/112 [==============================] - 1s 7ms/step - loss: 0.4823 - accuracy: 0.8483 - val_loss: 1.0127 - val_accuracy: 0.6935
Epoch 10/10
112/112 [==============================] - 1s 8ms/step - loss: 0.4532 - accuracy: 0.8564 - val_loss: 1.0373 - val_accuracy: 0.7013
35/35 [==============================] - 0s 2ms/step - loss: 0.9653 - accuracy: 0.7350

Test accuracy: 0.74
```

Después de 10 Epochs llegamos a obtener un Accuracy del **74%** que es bastante bueno para una exploración inicial.

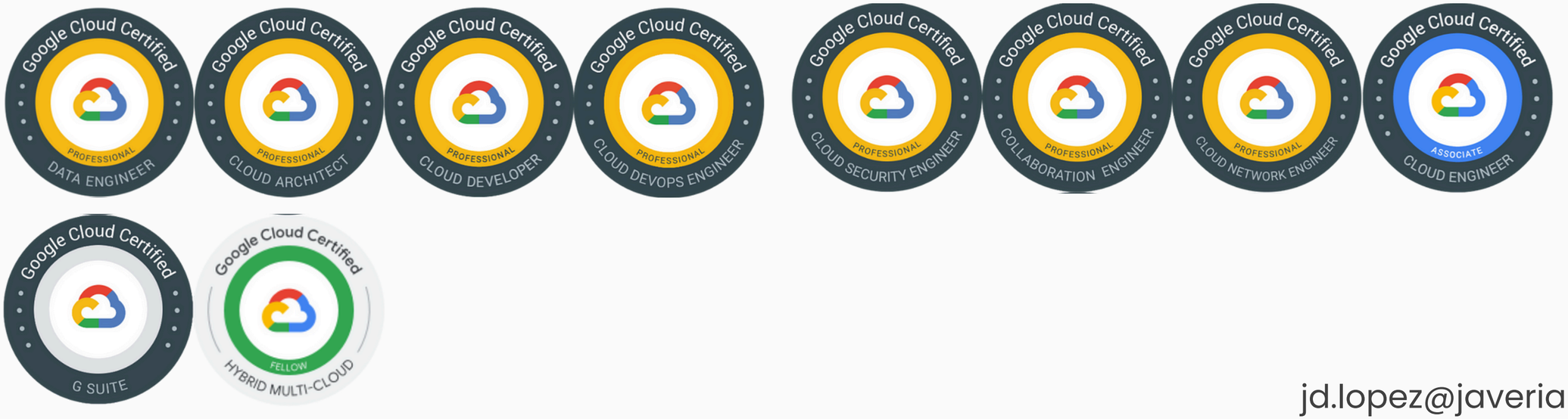
Conclusiones:

- **Los datos muestran que los impactos frontales directos son la principal causa de daños en accidentes vehiculares.** Esto resalta la importancia de reforzar las medidas de seguridad en los sistemas de frenado automático de emergencia y de alerta de colisión frontal en los vehículos autónomos de Tesla.
- **Los SUV y sedanes, dos de los modelos más vendidos por Tesla, aparecen desproporcionadamente involucrados en accidentes.** Tesla podría aprovechar estos insights para optimizar los sistemas avanzados de asistencia al conductor específicamente para estos modelos populares.
- La identificación de las horas pico de mayor riesgo de accidentes podría permitir que **Tesla ajuste dinámicamente los algoritmos de conducción autónoma para adoptar un enfoque más cauteloso y defensivo durante esos períodos críticos**, minimizando así el riesgo de colisiones.

Invitación a Certificarse con GCP.

Los exámenes de Google Cloud tienen un costo de alrededor de **200 Dólares**.

Si desean obtener el entrenamiento necesario para certificarse con Google Cloud y hacer el examen de forma **100% gratuita**. Me pueden contactar directamente, hay cupos limitados.



jd.lopez@javeriana.edu.co

Gracias

Referencias

- Data.gov Home - Data.gov. (2022). Data.gov. https://catalog.data.gov/dataset/?q=Precinct&tags=__&_tags_limit=0&_organization_limit=0&organization_type=City+Government
- Average Annual Population of NYC Neighborhoods, 2016-2020. (2016). Ny.gov. https://www.health.ny.gov/statistics/cancer/registry/appendix/neighborhoodpop.htm