Data story telling

Inicialmente solo quería saber y entender por qué hay tantos accidentes en mi ciudad, Buenos Aires, en comparación con otras ciudades de Argentina.

Ya que tenemos un alto número de accidentes con 15 millones de habitantes en esta ciudad.
Para hacerlo utilicé el data set de la página oficial del gobierno de Buenos Aires llamado: "SeguridadVialAutopistasAUSA".
Empecé a analizar el data set haciendo gráficos en función de los accidentes ocurridos, con sus respectivos horarios y cantidad de muertes que tuvo cada accidente.

Índice

Importaciones

```
import statsmodels.regression.linear_model as sm
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import scipy
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.metrics import mean_absolute_error
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from statsmodels.api import add_constant
import statsmodels.api as sm
import os
#%matplotlib inline
#plt.style.use('ggplot')
from bokeh.resources import INLINE
import bokeh.io
from bokeh import *
import sklearn # Paquete base de ML
from scipy.stats import norm
from sklearn.cluster import KMeans
from sklearn.preprocessing import MinMaxScaler, MaxAbsScaler, RobustScaler,
StandardScaler


%matplotlib inline
```

Carga de data sets y pequeñas consultas

```
filename = 'intervenciones-de-seguridad-vial.csv'
filename2 = 'reclamos-ausa.xlsx'
data = pd.read_table(filename, header=0, sep=';')
data2=pd.read_excel(filename2, header=0)

print('''
            Intervenciones de seguridad vial
    ''')
print(data.shape)
print (data)
print('-'*80)
print('''
            Reclamos AUSA
    ''')
print(data2.shape)
print (data2)

print('-'*80)
totalFallecidos=data['fallecidos'].sum()
print(f'''
        La cantidad total de fallecidos debido a un accidente vial es de:
{totalFallecidos}
    ''')

print('-'*80)

maximaCantMuertos=data['fallecidos'].max()
print(f'''
        La cantidad maxima de fallecidos debido a un accidente vial es de:
{maximaCantMuertos}
    ''')
```

```
media=data['fallecidos'].mean()
mediaAcortada=round(media, 4)
print(f'''
        La media de fallecidos debido a un accidente vial es de:
{mediaAcortada}
    ''')
```

Acá cargue los conjuntos de datos que había descargado desde la página oficial del gobierno de la ciudad. Imprimí títulos y mostré los data sets por separado. También calcule la cantidad de fallecidos, la cantidad máxima de muertos que hubo por accidente y la media de fallecidos debido a un accidente vial.

Visualización prolija del Data set mediante DataFrame

```
df = pd.DataFrame(data, columns =
['fecha','hora','autopista','banda_y_o_ramal','pk','condiciones_meteorologic
as','superficie_de_la_via','lesionados','fallecidos','tipo_de_siniestro','mo
to','liviano','bus','camion'
])
df
```

Esto simplemente lo hice para tener una mejor visualización de mi conjunto de datos.

Resolviendo hipótesis

Me plantee una serie de hipótesis, de preguntas para responder, de sucesos que se podrían o no haber evitados. En muchos de ellos llegue a una determinada conclusión, en otros, quedo una respuesta abierta y/o difícil de responder, ya que no dependían de una variable en específico, sino que eran variables que yo no podía manejar.

Cantidad de fallecidos en determinados horarios

```
# Variable independiente = hora en la que ocurren los accidentes
# Variable dependiente = fallecidos

#Cual es la cantidad de fallecidos por hora
plt.bar(df.hora,df.fallecidos)
plt.xlabel('Hora')
plt.ylabel('Numero de muertes')
plt.show()
```

Lo que me plantee en esta ocasión fue: ¿Veamos, los accidentes ocurren siempre a la misma hora? ¿Hay un patrón en cuanto al horario de accidentes? ¿Se pueden evitar mas accidentes dependiendo en la hora en la que manejemos?

Mi conclusión a todas esas respuestas es que si, se puede evitar un accidente dependiendo el horario, aunque no lo crean, ya que la cantidad donde mas accidentes ocurren es a la mañana, en la hora pico de la ciudad, por lo tanto, si no quieres cruzarte con algún accidente o no quieres cometer un accidente, es recomendable que salgas antes o después de esos horarios claves.

Cantidad de fallecidos que fallecieron en un accidente con un camión involucrado

```
# Elijo los datos a analizar

# Variable independiente = camion
# Variable dependiente = fallecidos

#Cual es la cantidad de muertes que manjean camiones
plt.bar(df.camion,df.fallecidos)
plt.xlabel('Numero de camiones')
plt.ylabel('Numero de muertes')
plt.show()
```

Aquí analice lo siguiente: ¿Hay un patrón en cuanto al tipo de vehículo con los que se producen los accidentes de tránsito?
La respuesta que encontré en base a los datos analizados fue que no, los accidentes no depende n del tipo de vehículo, todos tenían una cantidad de accidentes bastante similar. (Puede ser que sea un poco más alto la cantidad de accidentes en moto, pero no reflejo una comparación abismal como para determinarlo como patrón)

Cantidad de fallecidos en dependiendo del clima

```
#Cual es la cantidad de fallecidos dependiendo del cli ma
plt.bar(df['fallecidos'], df['condiciones_meteorologicas'])
plt.xlabel('Numero de fallecidos')
plt.ylabel('Clima')
plt.show()
```

¿Hay algún patrón que me indiqué que el clima puede ser un factor fundamental para determinar si pueden ocurrir más o menos accidentes?
Esa fue la pregunta que me hice, y los datos me llevaron a esta respuesta:

Si, el patrón a analizar es la niebla, cuando hay niebla, todo empeora, los accidentes crecen exponencialmente, si esta lluvioso disminuye, pero aun, así como la calzada (autopista) esta resbalosa los accidentes siguen ocurriendo, pero hay una notable disminución de accidentes cuando el clima esta despejado y soleado.

Análisis para ver como se comportan las distribuciones de los fallecidos

```
sns.distplot(data['fallecidos'])
```

Esta distribución la saque y analice mas que nada para ver como se comportan la cantidad de fallecidos en un accidente de transito por día, pude ver que la densidad es alta cuando la cantidad de fallecidos decrece, y la densidad es baja cuando la cantidad de fallecidos aumenta.

Análisis para ver cómo se comportan las distribuciones de las horas en las que ocurren los accidentes

```
sns.distplot(data['hora'])
```

Esto lo hice con el mismo objetivo que el análisis visto anteriormente, quería ver como se manejaba la densidad en cuanto a las horas, y si, claramente aumentaba en las horas pico.

Análisis para ver las horas donde el clima está en buenas condiciones para manejar

```
df_muertosClima = df[df['condiciones_meteorologicas'] == 'BUENO']
columnaX =df_muertosClima['hora']

plt.figure(figsize=(8,6))
df_muertosClima =
df_muertosClima[df_muertosClima['condiciones_meteorologicas'] == 'BUENO']
columnaX = df_muertosClima['hora']
# Crear la densidad
sns.distplot(columnaX,  bins=10,  kde=False)
plt.title('Horas donde el clima esta "Bueno"', fontsize=16)
plt.xlabel('Horas', fontsize=14)
plt.xticks(fontsize=12)
plt.ylabel('Density', fontsize=14)
plt.yticks(fontsize=12)
plt.show()
```

Con la hipótesis que había analizado anteriormente, viendo los horarios y climas críticos donde más accidentes ocurrían, plantee una solución, pensando en lo opuesto analizado principalmente. Me pregunte ¿Cuáles son las horas donde el clima mayormente estáen buenas condiciones para conducir?
Y esto fue un desafío y un paradigma muy difícil de resolver, ya que en Buenos Aires, casi siempre, el clima donde esta "bueno" para conducir es justamente en los horarios picos, pero, analizando la totalidad de datos y la cantidad de accidentes que se produjeron en horarios picos y con buen clima, llegue a la conclusión de que hay clima bueno en horarios "casi picos" 1 hora o 2 horas de diferencia ya sean horas anteriores o horas posteriores para poder conducir con tranquilidad

Análisis para calcular la cantidad de accidentes y compararlo con la cantidad de fallecidos y lesionados dependiendo el vehículo

```
#Eleccion de metodo de feature selection


totalDeIncidentes=sum(data.lesionados)+sum(data.fallecidos)+sum(data.moto)+s
um(data.liviano)+sum(data.bus)+sum(data.camion)
print(f'Hubo una totalidad de: {totalDeIncidentes} incidentes')


totalDeIncidentes2=data[['fallecidos','lesionados','moto','liviano','bus','c
amion']]
#totalDeIncidentes2=data.lesionados+data.fallecidos+data.moto+data.liviano+d
ata.bus+data.camion
totalDeIncidentesPD=pd.DataFrame(totalDeIncidentes2)
totalDeIncidentesPD['totalDeIncidentes']=totalDeIncidentes
X=totalDeIncidentes2.drop("totalDeIncidentes", 1)    # feature matrix
y=totalDeIncidentes2['totalDeIncidentes']            # target feature
totalDeIncidentesPD.head()
```

Acá utilice el método “feature selection” para analizar todas las variables, pero centrado en un objetivo en concreto, la cantidad total de accidentes, para reducir el peso en los modelos. Entonces, dependiendo del tipo de automóvil con el que se haya producido el accidente quería analizar cuantos accidentes se concretaron a lo largo de 2020 y 2021.

Análisis de modelo de regresión – Volstat

```
model1 = 'VolStat~fallecidos+ lesionados+ moto+ bus + liviano +camion'
'''
y=model1
x=data[['fallecidos','lesionados','moto','bus','liviano','camion']]
print(np.asarray(x))
print(np.asarray(y))
'''
#lm1 = sm.OLS(X,y).fit()
xdat=data['fallecidos']
ydat=data['lesionados']
model=sm.OLS(ydat,xdat).fit()
print(model.summary())
```

Acá me indica la variable que estamos analizando, en este caso la variable es "lesionados" y "fallecidos" realicé una comparación obteniendo datos como R2, coeficientes y percentiles. Viendo el R2, me doy cuenta que el modelo no contiene un término de intersección en sí, también me di cuenta que el coeficiente de variación del modelo es bajo, esta por debajo del 30%, esto quiere decir que mi modelo es **no homogéneo**, y por último, tengo los percentiles que me van a indicar como se fueron moviendo los números dependiendo de los promedios mas altos y mas bajos

Análisis de métricas

```
#Calculo de metricas

fallecidos_x=data[['fallecidos','lesionados']]
fallecidos_y=data[['fallecidos','moto']]
arr=fallecidos_y.to_numpy()
arr2=fallecidos_x.to_numpy()


X_train,X_test,y_train,y_test = train_test_split(arr,arr2)


#Calculo de MAE
print('MAE', mean_absolute_error(X_test,y_test))
```

El MAE me dio 0.18 (18%), este es el promedio de la diferencia absoluta entre el valor que quiero observar (el total de los fallecidos que se lesionaron) y los valores predichos (los que fallecieron andando en moto)

Conectar una API de incidentes de trafico

```
#Conectandose a una API de inicidentes de trafico

import requests

url='https://api.open511.gov.bc.ca/events?limit=100'
data = requests.get(url)
if data.status_code==200:
    data = data.json()
    print(data)
```

Acá conecte una API para obtener mucha más información acerca de los accidentes de trafico en el mundo, como, por ejemplo, cuantos accidentes hubo en USA, los modelos de auto que tienen accidentes, la severidad del accidente, el punto geográfico donde ocurrió ese accidente, el día, la velocidad y una descripción

Data Wrangling

```
#Data Wrangling
#Reemplazo de valores nulos en la tabla

# replace "?" to NaN
df.replace("?", np.nan, inplace = True)
df.head(5)
```

Acá lo que hice fue reemplazar todos los valores NaN, para dejar el conjunto de datos limpio.

Control de datos perdidos

```
#Control de datos perdidos

missing_data = df.isnull()
missing_data.head(5)
for column in missing_data.columns.values.tolist():
    print(column)
    print (missing_data[column].value_counts())
    print("")
```

Lo que hice acá es ver cuántos datos perdí y cuantos logre recuperar haciendo el Data Wrangling