

Modelo Predictivo: ACCIDENTES DE TRÁFICO EN NUEVA ZELANDA

David Víctor Gómez Ramírez

Técnico Superior en Desarrollo de
Aplicaciones Multiplataforma
especialidad en BIG DATA.

Amante de la estadística y un
enamorado de los datos, consciente
de la deriva tecnológica y potencial de
los mismos.

Github: <https://github.com/dvgr78/BOSQUES-ALEATORIOS-REGRESI-N>

email: dvgr78@hotmail.com

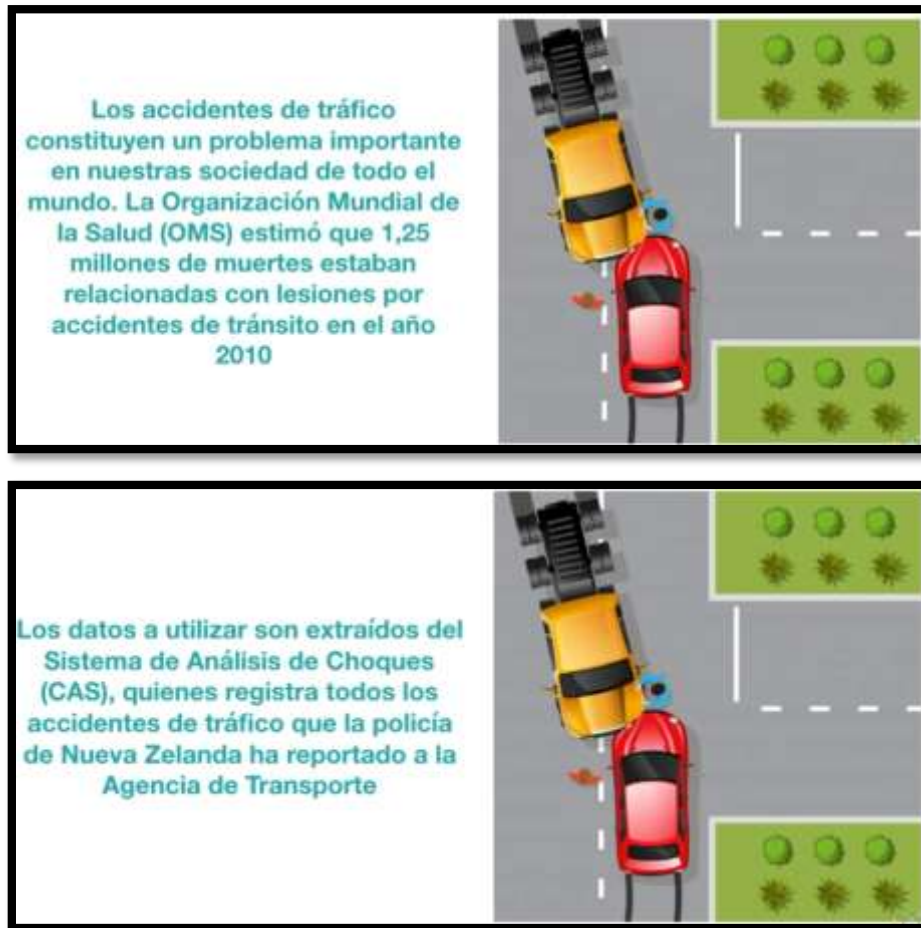
Contenido

BOSQUES ALEATORIOS REGRESIÓN	2
TÍTULO.....	2
Importamos librerías	2
Análisis de los datos.....	2
Procesamiento de los datos	4
Visualización de datos.....	5
Visualización gráfica.....	6
Pasar Objetos a datos	8
Entrenamiento del modelo	8
Código completo:	8
Precisión del Modelo	11
Referencias e itinerario de formación e investigación	11

BOSQUES ALEATORIOS REGRESIÓN

TÍTULO

Predecir el número de muertos en accidentes de tránsito en Nueva Zelanda



Importamos librerías

```
# Importaremos las librerías necesarias para el proyecto
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Análisis de los datos

```
# Primera vista general de los datos importados
print(data)
```

Modelo Predictivo: ACCIDENTES DE TRÁFICO EN NUEVA ZELANDA

```
      X      Y  OBJECTID_1  ...  unknownVeh  vanOrUtili  Pedestrian
0    175.226860 -37.820620    667848  ...        0          0          0
1    175.289010 -37.998924    667849  ...        0          0          0
2    176.193733 -37.665581    667850  ...        0          0          0
3    176.160955 -37.695747    667851  ...        0          0          0
4    176.981378 -37.957300    667852  ...        0          0          0
...      ...      ...      ...  ...      ...      ...      ...
674316  174.686684 -36.889784   1338843  ...        0          0          0
674317  174.874939 -36.996796   1338844  ...        0          0          0
674318  174.863792 -37.018396   1338845  ...        0          0          0
674319  174.807860 -36.965696   1338846  ...        0          0          0
674320  174.844429 -36.992993   1338847  ...        0          0          0

[674321 rows x 90 columns]
```

```
# para conocer la estructura de los datos
print(data.shape)
```

```
(674321, 90)
```

```
# para conocer el formato de los datos
print(data.dtypes)
```

```
X      float64
Y      float64
OBJECTID_1  int64
OBJECTID    int64
crashYear    int64
...
taxi      int64
truck     int64
unknownVeh  int64
vanOrUtili  int64
Pedestrian  int64
Length: 90, dtype: object
```

```
# Para conocer los datos nulos
print(data.isnull().sum())
```

Modelo Predictivo: ACCIDENTES DE TRÁFICO EN NUEVA ZELANDA

```
X          0
Y          0
OBJECTID_1 0
OBJECTID    0
crashYear   0
..
taxi        0
truck       0
unknownVeh  0
vanOrUtili  0
Pedestrian  0
Length: 90, dtype: int64
```

Procesamiento de los datos

```
# Eliminación de columnas innecesarias
data = data.drop(['X', 'Y', 'OBJECTID_1', 'OBJECTID'], axis = 1)

# Eliminación de columnas a las que le faltan datos
data =
data.drop(['crashDirec', 'crashRPDir', 'crashRPSH', 'crashRPNew', 'trafficCon', 'roadLane', 'corner
Road'], axis = 1)

# Reducimos el conjunto de datos y trabajaremos con los informados a partir del año 2010
data = data.loc[(data['crashYear']>2010), :]
```

```
crashYear crashFinan crashSever ... unknownVeh vanOrUtili Pedestrian
404513      2011  2010/2011      F ...          0          0          0
404514      2011  2010/2011      F ...          0          0          0
404515      2011  2010/2011      F ...          0          0          0
404516      2011  2010/2011      F ...          0          0          0
404517      2011  2010/2011      F ...          0          0          0
...          ...          ...    ... ...          ...          ...          ...
674316      2018  2018/2019      N ...          0          0          0
674317      2018  2018/2019      N ...          0          0          0
674318      2018  2018/2019      N ...          0          0          0
674319      2018  2018/2019      N ...          0          0          0
674320      2018  2018/2019      N ...          0          0          0

[258808 rows x 79 columns]
```

```
crashYear      int64
crashFinan     object
crashSever     object
fatalCount     int64
seriousInj     int64
...
taxi           int64
truck          int64
unknownVeh     int64
vanOrUtili     int64
Pedestrian     int64
Length: 79, dtype: object
```

```
crashYear      0
crashFinan     0
crashSever     0
fatalCount     0
seriousInj     0
..
taxi           0
truck          0
unknownVeh     0
vanOrUtili     0
Pedestrian     0
Length: 79, dtype: int64
```

Visualización de datos

```
# Obtenemos el valor del año más bajo
inicio = data.crashYear.min()
# Obtenemos el valor del año más alto
final = data.crashYear.max()
# Sumamos la cantidad de "muertes"
muertes = data['fatalCount'].sum()
# Sumamos la cantidad de "heridos graves"
graves = data['seriousInj'].sum()
# Sumamos la cantidad de "heridos leves"
leves = data['minorInjur'].sum()

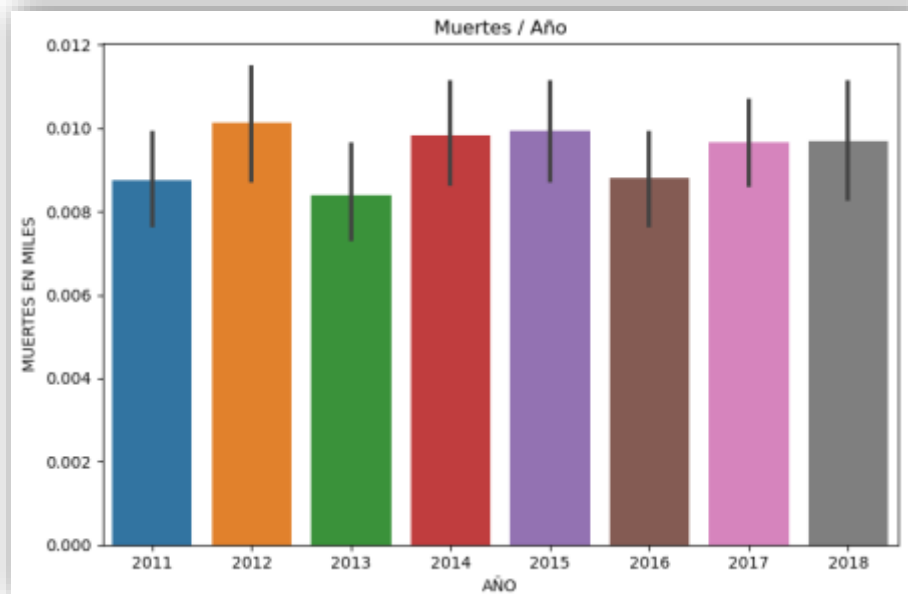
print("*****")
print("*****          DATOS RELEVANTES          *****")
print("*****")
print("**** Resultados desde el año:      {}".format(inicio)," hasta el año:  {}".format(final))
print("**** Muertes hasta el año {}:      {}".format(final,muertes))
print("**** Heridos graves:                {}".format(graves))
print("**** Herdidos leves:                {}".format(leves))
print("*****")
```

```
*****
*****          DATOS RELEVANTES          *****
*****
**** Resultados desde el año:      2011 hasta el año:  2018
**** Muertes hasta el año 2018:    2430
**** Heridos graves:                17693
**** Herdidos leves:                79943
*****
```

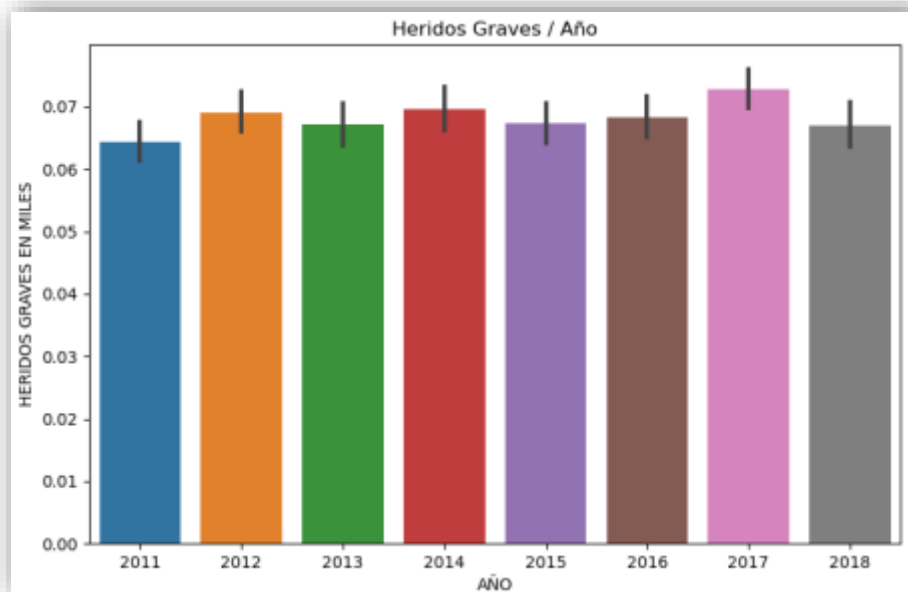
Modelo Predictivo: ACCIDENTES DE TRÁFICO EN NUEVA ZELANDA

Visualización gráfica

```
# Graficamos el número de muertes por año
g_muertes = sns.barplot(x= "crashYear", y ="fatalCount",data=data)
g_muertes.set(xlabel="AÑO", ylabel = "MUERTES EN MILES")
g_muertes.set_title('Muertes / Año')
plt.show()
```

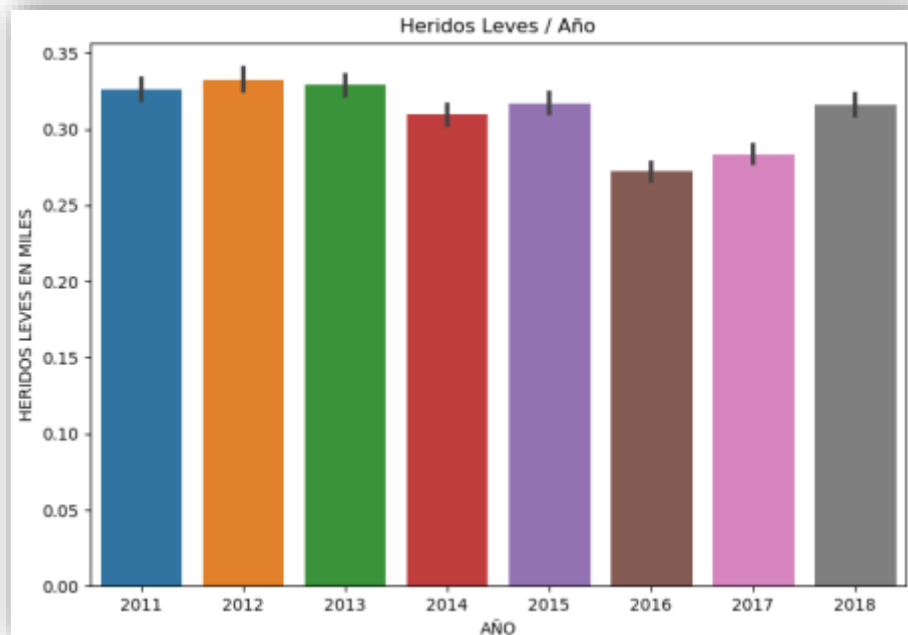


```
# Graficamos el número de heridos graves por año
g_graves = sns.barplot(x= "crashYear", y ="seriousInj", data=data)
g_graves.set(xlabel="AÑO", ylabel = "HERIDOS GRAVES EN MILES")
g_graves.set_title('Heridos Graves / Año')
plt.show()
```

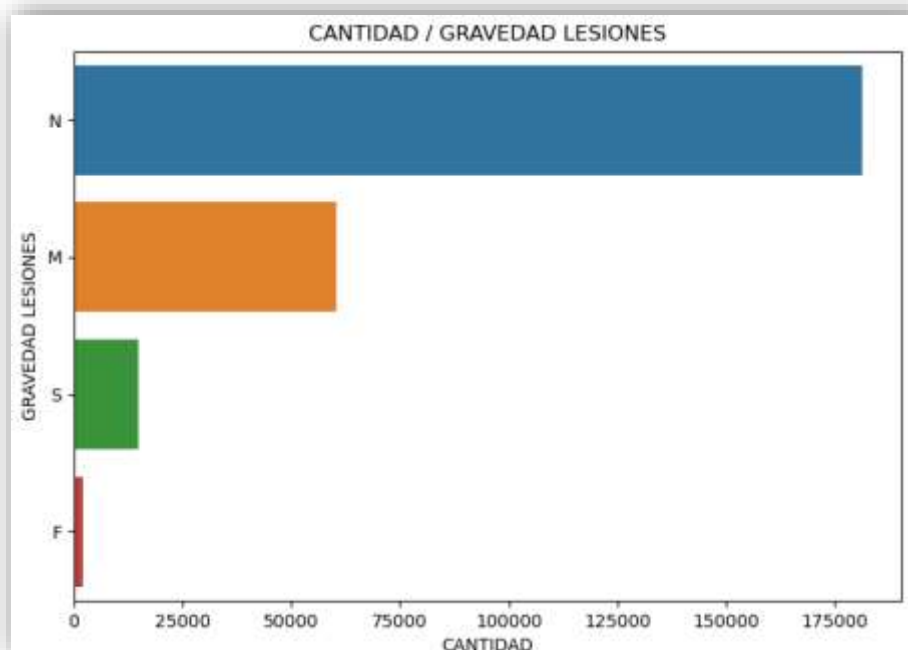


Modelo Predictivo: ACCIDENTES DE TRÁFICO EN NUEVA ZELANDA

```
# Graficamos el número de heridos leves por año
g_leves = sns.barplot(x= "crashYear", y ="minorInjur", data=data)
g_leves.set(xlabel="AÑO", ylabel = "HERIDOS LEVES EN MILES")
g_leves.set_title('Heridos Leves / Año')
plt.show()
```



```
# Graficamos cantidad según gravedad de heridas de forma horizontal
# N: sin daños
# M: leves
# S: graves
# F: muertes
g_gravedad = sns.countplot(y ="crashSever", data=data, order="NMSF")
g_gravedad.set(xlabel="CANTIDAD", ylabel = "GRAVEDAD LESIONES")
g_gravedad.set_title('CANTIDAD / GRAVEDAD LESIONES')
plt.show()
```



Pasar Objetos a datos

```
# importamos nueva librería
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder() # convierte datos repetitivos en
números
# Evaluamos cada una de las columnas para encodar tipos objetos
for i in data:
    if data[i].dtype == 'object':
        encoder.fit(data[i].astype(str))
        data[i]= encoder.transform(data[i])
data = pd.get_dummies(data)
```

Entrenamiento del modelo

```
# importamos nuevas librerías para entrenar nuestro modelo
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score

# Definimos los datos dependientes e independientes
X = data.drop('fatalCount', axis=1)
y = data['fatalCount']

# Seperamos entre datos de entrenamiento y datos de prueba
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.20,random_state=1)
# Definimos el algoritmo a utilizar "Bosques Aleatorios Regresión"
algoritmo = RandomForestRegressor(n_estimators=60)
# Entrenamos el algortimo
algoritmo.fit(X_train, y_train)
# Realizamos una predicción
y_test_pred = algoritmo.predict(X_test)
# Calculamos la precisión del modelo con R2
print("*****")
print("**** PRECISIÓN DEL MODELO R2: ",r2_score(y_test,y_test_pred), "
****")
print("*****")
```

Código completo:

```
# Importaremos las librerías necesarias para el proyecto

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# importaremos nuestros datos a un dataframe de pandas
data = pd.read_csv('Crash_Analysis_System_CAS_data.csv')

# *****
# ***** ANALIZAR LOS DATOS *****
# *****

# Primera vista general de los datos importados
print(data)

# para conocer la estructura de los datos
print(data.shape)
```

Modelo Predictivo: ACCIDENTES DE TRÁFICO EN NUEVA ZELANDA

```
# para conocer el formato de los datos
print(data.dtypes)
# Para conocer los datos nulos
print(data.isnull().sum())

# *****
# ***** PROCESAMIENTO DE LOS DATOS *****
# *****

# Eliminación de columnas innecesarias
data = data.drop(['X','Y','OBJECTID_1','OBJECTID'], axis = 1)

# Eliminación de columnas a las que le faltan datos
data =
data.drop(['crashDirec','crashRPDir','crashRPSH','crashRPNew','trafficCon','roadLane','co
rnerRoad'], axis = 1)

# Reducimos el conjunto de datos y trabajaremos con los informados a partir del año 2010
data = data.loc[(data['crashYear']>2010), :]

print(data)
print(data.shape)
print(data.dtypes)
print(data.isnull().sum())

# *****
# ***** VISUALIZACIÓN DE LOS DATOS *****
# *****

# Obtenemos el valor del año más bajo
inicio = data.crashYear.min()

# Obtenemos el valor del año más alto
final = data.crashYear.max()

# Sumamos la cantidad de "muertes"
muertes = data['fatalCount'].sum()

# Sumamos la cantidad de "heridos graves"
graves = data['seriousInj'].sum()

# Sumamos la cantidad de "heridos leves"
leves = data['minorInjur'].sum()

print("*****")
print("***** DATOS RELEVANTES *****")
print("*****")
print("**** Resultados desde el año: {}\".format(inicio),\" hasta el año:
{}\".format(final))
print("**** Muertes hasta el año {}: {}\".format(final,muertes))
print("**** Heridos graves: {}\".format(graves))
print("**** Heridos leves: {}\".format(leves))
print("*****")

# *****
# ***** VISUALIZACIÓN GRÁFICA *****
# *****

# Graficamos el número de muertes por año
g_muertes = sns.barplot(x= "crashYear", y ="fatalCount",data=data)
g_muertes.set(xlabel="AÑO", ylabel = "MUERTES EN MILES")
```

Modelo Predictivo: ACCIDENTES DE TRÁFICO EN NUEVA ZELANDA

```
g_muertes.set_title('Muertes / Año')
plt.show()

# Graficamos el número de heridos graves por año
g_graves = sns.barplot(x= "crashYear", y ="seriousInj", data=data)
g_graves.set(xlabel="AÑO", ylabel = "HERIDOS GRAVES EN MILES")
g_graves.set_title('Heridos Graves / Año')
plt.show()

# Graficamos el número de heridos leves por año
g_leves = sns.barplot(x= "crashYear", y ="minorInjur", data=data)
g_leves.set(xlabel="AÑO", ylabel = "HERIDOS LEVES EN MILES")
g_leves.set_title('Heridos Leves / Año')
plt.show()

# Graficamos cantidad según gravedad de heridas de forma horizontal
# N: sin daños
# M: leves
# S: graves
# F: muertes
g_gravedad = sns.countplot(y ="crashSever", data=data, order="NMSF")
g_gravedad.set(xlabel="CANTIDAD", ylabel = "GRAVEDAD LESIONES")
g_gravedad.set_title('CANTIDAD / GRAVEDAD LESIONES')
plt.show()

# PARA PASAR LOS OBJETOS A DATOS
# importamos nueva librería
from sklearn.preprocessing import LabelEncoder
encoder = LabelEncoder() # convierte datos repetitivos en números

# Evaluamos cada una de las columnas para encodar tipos objetos
for i in data:
    if data[i].dtype == 'object':
        encoder.fit(data[i].astype(str))
        data[i]= encoder.transform(data[i])
data = pd.get_dummies(data)

# *****
# *****  ENTRENAMIENTO DEL MODELO  *****
# *****

# importamos nuevas librerías para entrenar nuestro modelo
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score

# Definimos los datos dependientes e independientes
X = data.drop('fatalCount', axis=1)
y = data['fatalCount']

# Seperamos entre datos de entrenamiento y datos de prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=1)

# Definimos el algoritmo a utilizar "Bosques Aleatorios Regresión"
algoritmo = RandomForestRegressor(n_estimators=60)

# Entrenamos el algortimo
algoritmo.fit(X_train, y_train)

# Realizamos una predicción
y_test_pred = algoritmo.predict(X_test)

# Calculamos la precisión del modelo con R2
```

Modelo Predictivo: ACCIDENTES DE TRÁFICO EN NUEVA ZELANDA

```
print("*****")
print("**** PRECISIÓN DEL MODELO R2: ", r2_score(y_test, y_test_pred), " ****")
print("*****")
```

Precisión del Modelo

```
*****
**** PRECISIÓN DEL MODELO R2:  0.8929510439705043  ****
*****
```

Referencias e itinerario de formación e investigación

Lidgi González, AprendeIA: <https://escuela.aprendeia.com/>

Aprende Inteligencia Artificial: https://www.youtube.com/playlist?list=PLJjOveEiVE4AdmqW_wmIKmlyx06LhMfJO

Aprende Machine Learning con Python: <https://www.youtube.com/playlist?list=PLJjOveEiVE4Dk48EI7l-67PEleEC5nxc3>