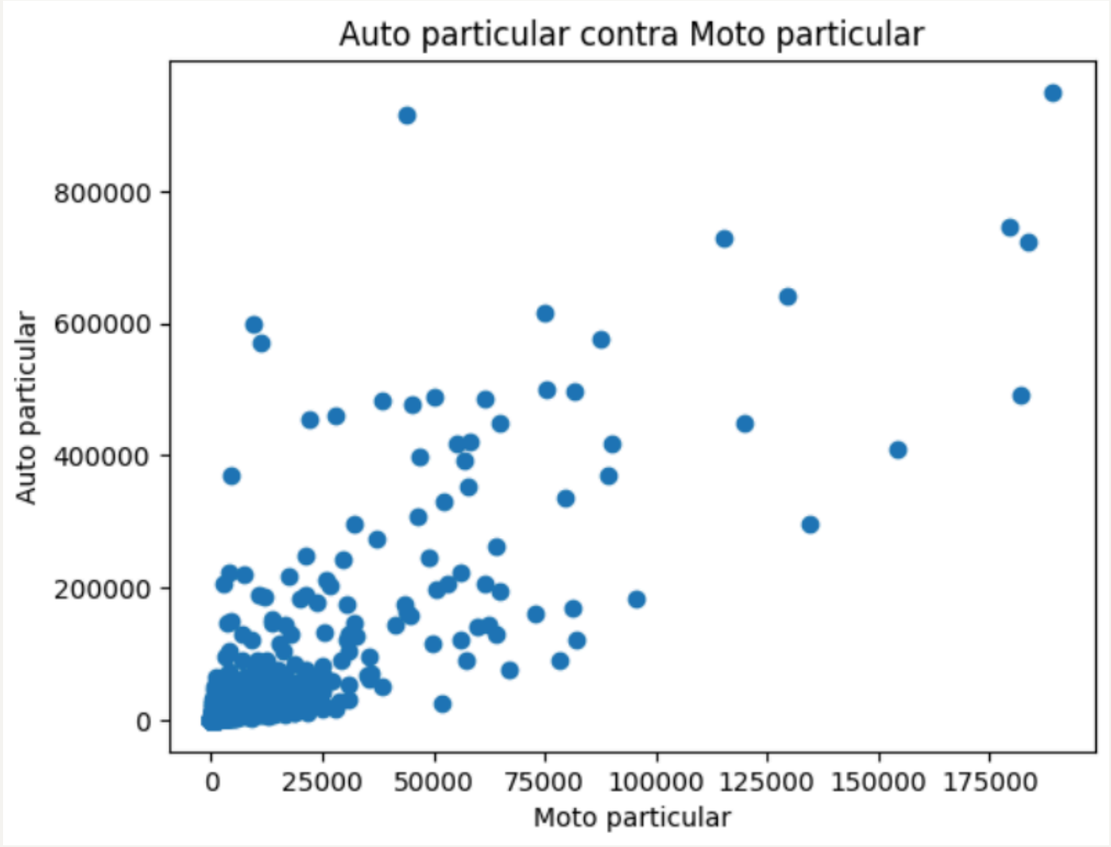
Analisis de Vehiculos en Circulación

Angel Gael Calleros Martínez #654375

Visualización de Hipotesis

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
datos = pd.read_csv('P1-Vehiculos-en-Circulacion.csv')
print("Nombres de variables y datos\n", datos.head())
print("\nTipo de datos\n", datos.dtypes)
print("\nDimensiones totales:", datos.shape, "\n")

Y_AP=datos.AUTO_PARTICULAR
X_MP=datos.MOTO_PARTICULAR
plt.scatter(X_MP, Y_AP)
plt.title("Auto particular contra Moto particular")
plt.xlabel("Moto particular")
plt.ylabel("Auto particular")
plt.show()

print(datos.PROD_EST.value_counts())
print(datos.COBERTURA.value_counts())
print(datos.ANIO.value_counts())
print(datos.ESTATUS.value_counts())

datos=datos.drop(['PROD_EST'], axis=1)
datos=datos.drop(['COBERTURA'], axis=1)
datos=datos.drop(['ANIO'], axis=1)
datos=datos.drop(['ESTATUS'], axis=1)

datos=datos.drop(['ID_MUNICIPIO'], axis=1)

print("\nNueva base de datos\n", datos.head(), "\n")
print(datos.dtypes)
```



Cualitativo a Cuantitativo

```
datos_encoded=datos.copy()

if datos_encoded["ID_ENTIDAD"].dtype == 'bool':
    datos_encoded["ID_ENTIDAD"] = datos_encoded["ID_ENTIDAD"].astype(int)
else:
    datos_encoded["ID_ENTIDAD"] = datos_encoded["ID_ENTIDAD"].astype("category")

datos_encoded = pd.get_dummies(datos_encoded, columns=["ID_ENTIDAD"], drop_first=True)

datos_encoded = datos_encoded.copy()
for col in datos_encoded.columns:
    if datos_encoded[col].dtype=='bool':
        datos_encoded[col]=datos_encoded[col].astype(int)
numeric_cols=datos_encoded.select_dtypes(include=np.number)

print("Shape final:", numeric_cols.shape)
print(numeric_cols.head())
```

Regresion Mixta

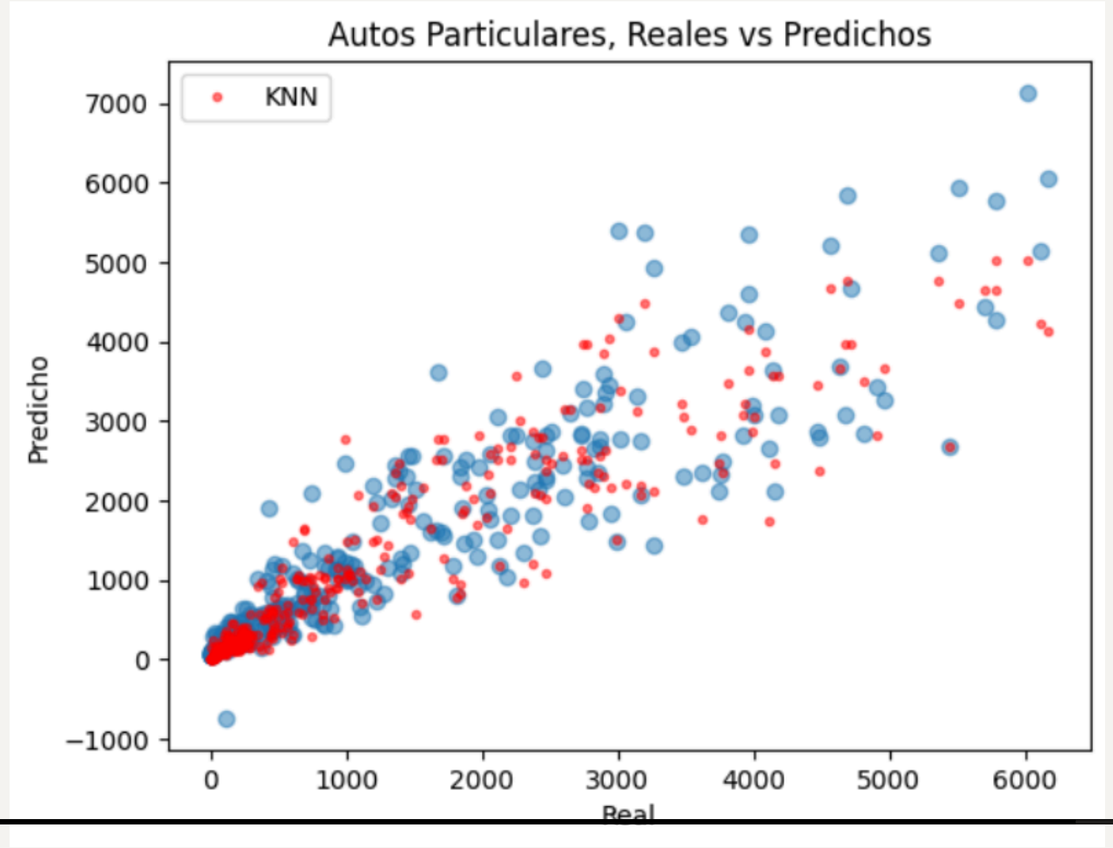
```
from mlxtend.feature_selection import SequentialFeatureSelector as SFS
from sklearn.linear_model import LinearRegression

estimator=LinearRegression()

sfsForward=SFS(estimator,
                  k_features=(3,10),
                  forward=True,
                  scoring='r2',
                  cv=10)
modeloF=sfsForward.fit(X_train, Y_train)
selected_variablesF=list(modeloF.k_feature_names_)


from sklearn.metrics import r2_score
x_train_selectedF=X_train[selected_variablesF]
x_test_selectedF=X_test[selected_variablesF]


sfsFBward =SFS(estimator,
                  k_features=(3,10),
                  forward=False,
                  scoring='r2',
                  cv=10)
ModeloFB=sfsFBward.fit(x_train_selectedF,Y_train)

print("Variables seleccionadas:\n", ModeloFB.k_feature_names_)

X_train_selectedFB=X_train[list(ModeloFB.k_feature_names_)]
X_test_selectedFB=X_test[list(ModeloFB.k_feature_names_)]
estimator.fit(X_train_selectedFB,Y_train)
```

Metricas

```
                            OLS Regression Results
==============================================================================
Dep. Variable:        AUTO_PARTICULAR   R-squared:                       0.844
Model:                            OLS   Adj. R-squared:                  0.843
Method:                 Least Squares   F-statistic:                     815.9
Date:                Mon, 08 Sep 2025   Prob (F-statistic):               0.00
Time:                        08:55:02   Log-Likelihood:                -11951.
No. Observations:                1524   AIC:                         2.392e+04
Df Residuals:                    1513   BIC:                         2.398e+04
Df Model:                          10
Covariance Type:            nonrobust
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const             51.1246     21.477      2.380      0.017       8.997      93.252
AUTO_PUBLICO       2.1109      0.229      9.234      0.000       1.663       2.559
CAM_PAS_PUBLICO    6.7747      1.225      5.529      0.000       4.371       9.178
CYC_CARGA_PARTICULAR  0.8040      0.015     54.249      0.000       0.775       0.833
MOTO_PARTICULAR    0.3483      0.021     16.605      0.000       0.307       0.389
ID_ENTIDAD_13    669.5005     84.681      7.906      0.000     503.395     835.606
ID_ENTIDAD_15   1492.0281    102.952     14.492      0.000    1290.084    1693.973
ID_ENTIDAD_17    775.1106    131.032      5.915      0.000     518.087    1032.134
ID_ENTIDAD_18  -1051.1788    189.910     -5.535      0.000   -1423.693    -678.664
ID_ENTIDAD_24   -781.9315    103.845     -7.530      0.000    -985.627    -578.236
ID_ENTIDAD_29   1706.2173    117.400     14.533      0.000    1475.934    1936.501
==============================================================================
Omnibus:                      462.251   Durbin-Watson:                   1.941
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             3349.605
Skew:                           1.217   Prob(JB):                         0.00
Kurtosis:                       9.843   Cond. No.                     2.31e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.31e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

```
Comparacion del RSS
 RSS lineal:  126493531.37699278
 RSS KNN:  115370014.52000001

Comparacion del RSE
 RSE lineal:  576.1978994774155
 RSE KNN:  550.280325888876

Comparacion del R2
 R2 lineal:  0.8402404143300133
 R2 KNN:  0.8542892627171292
```

Grafico Final



¡Gracias por su atención!