ACT3

September 3, 2024

Realizar las transformaciones adecuadas a las variables predictoras.

Realizar el modelo de regresión con las variables significativas.

Probar si se deben agregar interacciones o términos polinomiales.

Interpretar la tabla ANOVA, R2, R2 ajustada, p-values y FIV.

Verificar el cumplimiento de los supuestos.

[ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn import linear_model
from sklearn.metrics import r2_score
import scipy.stats as stats
from sklearn.feature_selection import RFE
from statsmodels.stats.outliers_influence import variance_inflation_factor
import statsmodels.api as sm
```

1. Realizar las transformaciones adecuadas a las variables predictoras.

[ ]:
```
dataframe=pd.read_csv('dataset.csv')
dataframe
```

[ ]:
```
     Factor Coagulación  Índice pronóstico  Función de enzima  \
0                   6.7                 62                 81
1                   5.1                 59                 66
2                   7.4                 57                 83
3                   6.5                 73                 41
4                   7.8                 65                115
..                  …                  …                  …
103                 5.8                 70                 64
104                 5.4                 64                 81
105                 6.9                 90                 33
106                 7.9                 45                 55
107                 4.5                 68                 60

     Función de hígado  Edad  Género  Alcohol (moderado)  Alcohol (severo)  \
0                 2.59    50       0                   1                 0
```

```
1                   1.70    39       0                    0                  0
2                   2.16    55       0                    0                  0
3                   2.01    48       0                    0                  0
4                   4.30    45       0                    0                  1
..                   …    …      …                    …                …
103                 2.52    49       0                    1                  0
104                 1.36    62       0                    1                  0
105                 2.78    48       1                    0                  0
106                 2.46    43       0                    1                  0
107                 2.07    59       0                    0                  0

     Sobrevivencia (días)
0                     695
1                     403
2                     710
3                     349
4                    2343
..                    …
103                   589
104                   599
105                   655
106                   377
107                   642

[108 rows x 9 columns]
```

[ ]:
```
# Quito la columna 'Sobrevivencia(días)'
x_df = dataframe.drop('Sobrevivencia (días)', axis=1)
```

[ ]:
```
#x_df=dataframe.drop('Sobrevivencia(días)',axis=1)
y_df=dataframe['Sobrevivencia (días)']
```

[ ]:
```
x_df
```

[ ]:
```
     Factor Coagulación  Índice pronóstico  Función de enzima  \
0                   6.7                 62                 81
1                   5.1                 59                 66
2                   7.4                 57                 83
3                   6.5                 73                 41
4                   7.8                 65                115
..                  …                 …                 …
103                 5.8                 70                 64
104                 5.4                 64                 81
105                 6.9                 90                 33
106                 7.9                 45                 55
107                 4.5                 68                 60
```

```
     Función de hígado  Edad  Género  Alcohol (moderado)  Alcohol (severo)
0                 2.59    50       0                   1                 0
1                 1.70    39       0                   0                 0
2                 2.16    55       0                   0                 0
3                 2.01    48       0                   0                 0
4                 4.30    45       0                   0                 1
..                 …   …     …                   …                …
103               2.52    49       0                   1                 0
104               1.36    62       0                   1                 0
105               2.78    48       1                   0                 0
106               2.46    43       0                   1                 0
107               2.07    59       0                   0                 0

[108 rows x 8 columns]
```

[ ]:
```
y_df
```

[ ]:
```
0       695
1       403
2       710
3       349
4      2343
       …
103     589
104     599
105     655
106     377
107     642
Name: Sobrevivencia (días), Length: 108, dtype: int64
```

[ ]:
```
# Selecciono las tres últimas columnas (que no se deben escalar)
columns_to_exclude = x_df.iloc[:, -3:].to_numpy()

# Selecciono las columnas que se van a escalar (excluyendo las tres últimas)
columns_to_scale = x_df.iloc[:, :-3].to_numpy()
y=y_df.to_numpy()
```

[ ]:
```
y
```

[ ]:
```
array([ 695,  403,  710,  349, 2343,  348,  518,  749, 1056,  968,  745,
        257, 1573,  858,  702,  809,  682,  205,  550,  838,  359,  353,
        599,  562,  651,  751,  545, 1965,  477,  600,  443,  181,  411,
       1037,  482,  634,  678,  362,  637,  705,  536,  582, 1270,  538,
        482,  611,  960, 1300,  581, 1078,  405,  579,  550,  651,  302,
        767,  487,  242,  705,  716,  266,  361,  460, 1060,  502,  882,
        352,  307, 1227,  508,  419,  536,  902,  189, 1433,  815, 1144,
        571,  591,  533,  534,  374,  222,  881,  470,  913,  527,  676,
```

```
        850,   569,   182,   421,   245,   611,   338,   875,   750,   935,   583,
        319, 1158,   553, 1041,   589,   599,   655,   377,   642], dtype=int64)
```

[ ]:
```
#Scale data
scaler=StandardScaler()
# Escalo las columnas seleccionadas
scaled_columns = scaler.fit_transform(columns_to_scale)

# Combino las columnas escaladas con las columnas no escaladas
x = np.hstack((scaled_columns, columns_to_exclude))
#x=scaler.fit_transform(x)
```

[ ]:
```
#x
```

2. Realizar el modelo de regresión con las variables significativas.

[ ]:
```
model=linear_model.LinearRegression()
```

[ ]:
```
# Define model and RFE selector
selector = RFE(model, n_features_to_select=5)  # Adjust the number of features␣
 ↪to select

# Fit RFE
selector = selector.fit(x, y)

# Get selected features
x_selected = selector.transform(x)
x_selected
```

[ ]:
```
array([[ 0.57766012, -0.04760235,  0.43832891, -0.06817677,  0.        ],
       [-0.55024574, -0.22694146, -0.28184638, -0.95734248,  0.        ],
       [ 1.07111893, -0.34650086,  0.53435228, -0.49777368,  0.        ],
       [ 0.43667189,  0.60997435, -1.48213851, -0.64763307,  0.        ],
       [ 1.3530954 ,  0.13173675,  2.07072621,  1.64022026,  1.        ],
       [-0.05678693, -1.48231517,  0.00622374, -1.23708   ,  0.        ],
       [-0.12728104, -1.00407757, -0.42588143, -0.74753933,  1.        ],
       [-1.53716337,  0.31107585,  0.43832891, -0.08815802,  0.        ],
       [ 0.08420131,  0.25129615,  1.01446913, -0.1580924 ,  0.        ],
       [-1.53716337,  0.78931346,  1.06248082, -0.25799866,  0.        ],
       [ 0.29568365,  1.26755106,  0.53435228,  1.47037962,  0.        ],
       [ 0.57766012, -0.70517906, -1.38611514, -0.79749246,  0.        ],
       [-0.05678693,  1.98490747,  2.02271453,  1.29054835,  0.        ],
       [-0.05678693,  1.20777136,  0.7744107 ,  1.29054835,  0.        ],
       [ 1.28260128, -0.04760235, -0.23383469,  0.74106393,  1.        ],
       [ 1.07111893,  0.66975405, -0.185823  , -0.25799866,  0.        ],
       [ 0.08420131,  1.32733076, -2.10629042,  0.32145764,  0.        ],
       [-1.53716337, -0.70517906, -1.48213851, -1.10720186,  0.        ],
       [ 1.00062482,  0.31107585,  0.10224711,  0.90091394,  0.        ],
```

```
[-0.19777516, -0.34650086,  0.72639902,  0.36142014,  1.        ],
[-0.47975162, -0.64539936,  0.19827048,  0.1915795 ,  0.        ],
[-1.74864572,  1.20777136, -0.90599829, -1.53679878,  0.        ],
[ 0.57766012, -2.19967158, -0.185823  , -0.55771744,  1.        ],
[-0.05678693,  0.25129615,  0.67838733,  0.74106393,  0.        ],
[ 0.29568365, -0.22694146,  1.35055093,  0.29148576,  0.        ],
[-0.05678693, -0.10738206,  0.05423542,  0.84097019,  0.        ],
[-0.47975162, -0.64539936,  0.67838733, -0.20804553,  0.        ],
[ 3.74989535,  0.78931346,  0.87043408,  2.929011  ,  1.        ],
[-0.47975162, -0.52583996, -0.76196323,  0.05171074,  0.        ],
[-0.05678693,  0.78931346, -0.61792817, -0.0781674 ,  0.        ],
[-1.88963395,  0.07195705, -0.32985806, -1.91644256,  0.        ],
[ 1.98754244, -1.06385727, -2.34634885, -0.13811115,  0.        ],
[-0.62073986, -0.22694146,  0.05423542,  0.84097019,  0.        ],
[-0.05678693,  0.55019465,  1.01446913,  0.64115767,  1.        ],
[-0.33876339, -0.28672116, -0.08979963, -0.01822364,  0.        ],
[-0.40925751, -0.70517906,  1.30253924, -0.05818615,  0.        ],
[-2.31259865,  0.66975405,  0.67838733, -0.60767057,  0.        ],
[-1.11419867, -3.27570619,  2.26277295,  0.1915795 ,  0.        ],
[-0.76172809, -0.10738206,  0.19827048, -0.20804553,  0.        ],
[-0.33876339, -0.64539936,  0.7744107 , -0.84744559,  0.        ],
[-0.47975162, -0.82473846,  0.00622374, -0.81747371,  0.        ],
[-1.60765748, -2.08011218,  1.30253924, -1.35696751,  1.        ],
[ 2.05803656,  1.38711046,  0.7744107 ,  3.73825169,  0.        ],
[ 0.43667189, -0.40628056,  0.24628216,  0.1915795 ,  0.        ],
[-1.74864572,  0.84909316,  1.01446913, -1.17713625,  0.        ],
[ 0.43667189, -1.36275577,  0.58236396,  0.34143889,  0.        ],
[-0.97321044,  0.60997435,  1.63862104,  0.39139202,  0.        ],
[-0.76172809,  1.38711046,  1.39856262,  1.44040774,  1.        ],
[-0.55024574,  0.25129615,  0.24628216,  0.20157013,  0.        ],
[-1.39617514,  1.14799166,  1.49458599,  1.8899859 ,  0.        ],
[ 0.507166  ,  0.84909316, -1.24208008, -0.70757683,  0.        ],
[ 0.36617777,  1.32733076, -1.5301502 , -1.44688314,  1.        ],
[ 0.36617777, -0.22694146,  0.63037565, -0.32793304,  0.        ],
[ 2.05803656,  0.90887286,  0.00622374,  0.54125141,  0.        ],
[ 0.85963658, -2.37901068,  0.29429385, -0.72755808,  0.        ],
[-0.69123397,  0.19151645,  0.91844576,  0.39139202,  0.        ],
[ 0.36617777,  1.62622927, -1.77020862, -1.59674253,  1.        ],
[-0.12728104, -1.66165427, -0.08979963, -0.52774556,  0.        ],
[ 0.15469542, -1.24319637, -0.13781132, -0.40785805,  1.        ],
[ 1.49408363, -2.13989188,  0.53435228, -0.62765182,  0.        ],
[ 0.64815424, -1.72143398, -1.00202166, -1.38693939,  0.        ],
[-0.83222221,  0.01217735, -1.72219694, -0.94735185,  1.        ],
[ 0.78914247, -0.94429787, -0.23383469, -1.05724873,  0.        ],
[ 0.57766012,  0.37085555, -0.32985806,  0.25152326,  1.        ],
[ 0.57766012, -1.00407757,  0.29429385,  0.60119516,  0.        ],
[-0.05678693, -0.16716176,  0.67838733,  0.45133578,  0.        ],
```

```
       [ 0.57766012, -0.40628056, -1.91424368, -1.12718312,  0.        ],
       [ 0.64815424, -0.70517906, -0.66593986, -0.47779243,  0.        ],
       [ 0.9301307 ,  1.92512777,  0.48634059,  2.01986404,  0.        ],
       [ 1.07111893, -0.64539936, -0.23383469,  0.62117642,  0.        ],
       [-0.40925751, -0.58561966, -0.47389312, -0.23801741,  0.        ],
       [-1.6781516 , -0.28672116,  0.58236396, -0.91737997,  0.        ],
       [ 0.64815424,  0.66975405,  0.34230554, -0.40785805,  1.        ],
       [-1.04370455, -0.94429787, -1.09804503, -0.23801741,  0.        ],
       [ 0.78914247,  0.19151645,  2.21476127,  2.02985467,  0.        ],
       [ 0.57766012, -0.10738206, -0.71395154,  1.21062334,  0.        ],
       [-0.19777516,  0.72953376,  1.49458599,  0.45133578,  0.        ],
       [ 0.71864835, -0.28672116,  0.7744107 ,  0.80100768,  0.        ],
       [ 0.22518954, -0.04760235, -0.71395154, -1.40692064,  1.        ],
       [-0.83222221,  2.04468717, -2.15430211, -0.88740809,  0.        ],
       [ 0.64815424,  0.37085555, -0.56991649,  0.24153263,  0.        ],
       [ 0.08420131,  0.60997435, -0.66593986, -1.43689252,  1.        ],
       [ 0.01370719, -0.76495876, -0.47389312,  0.53126078,  0.        ],
       [-0.26826928,  1.50666987,  0.10224711,  0.55124203,  0.        ],
       [-1.46666925, -0.46606026, -0.95400997, -1.24707063,  0.        ],
       [-1.11419867,  2.16424657,  0.53435228,  1.2705671 ,  0.        ],
       [ 0.507166  , -0.88451816, -0.8579866 ,  0.28149514,  1.        ],
       [ 0.22518954, -1.24319637, -0.42588143, -0.80748309,  1.        ],
       [-0.62073986, -0.16716176,  1.59060936,  0.51127953,  0.        ],
       [-0.05678693, -0.04760235,  0.48634059,  0.52127016,  0.        ],
       [-0.47975162, -0.40628056, -1.09804503, -1.17713625,  0.        ],
       [-0.12728104,  0.43063525, -0.61792817, -0.37788617,  1.        ],
       [-0.83222221,  0.07195705, -1.14605671, -1.35696751,  0.        ],
       [ 1.3530954 ,  0.66975405, -1.09804503, -0.0781674 ,  0.        ],
       [-0.76172809, -0.64539936, -1.29009177,  0.05171074,  0.        ],
       [-0.69123397,  0.55019465,  0.87043408,  0.85096081,  0.        ],
       [-0.90271632,  0.60997435, -0.71395154,  0.16160762,  0.        ],
       [ 0.01370719,  0.90887286, -0.08979963,  1.62023901,  0.        ],
       [-0.90271632,  0.37085555, -0.08979963,  0.51127953,  0.        ],
       [ 0.15469542, -0.58561966, -0.95400997, -0.81747371,  0.        ],
       [ 0.01370719,  1.50666987,  1.25452756,  0.67112955,  0.        ],
       [-0.83222221,  0.19151645, -0.185823  , -0.85743622,  1.        ],
       [ 3.18594242, -0.04760235,  0.63037565,  1.98989216,  0.        ],
       [-0.05678693,  0.43063525, -0.37786975, -0.13811115,  0.        ],
       [-0.33876339,  0.07195705,  0.43832891, -1.29702376,  0.        ],
       [ 0.71864835,  1.62622927, -1.866232  ,  0.12164512,  0.        ],
       [ 1.42358951, -1.06385727, -0.80997492, -0.19805491,  0.        ],
       [-0.97321044,  0.31107585, -0.56991649, -0.58768932,  0.        ]])
```

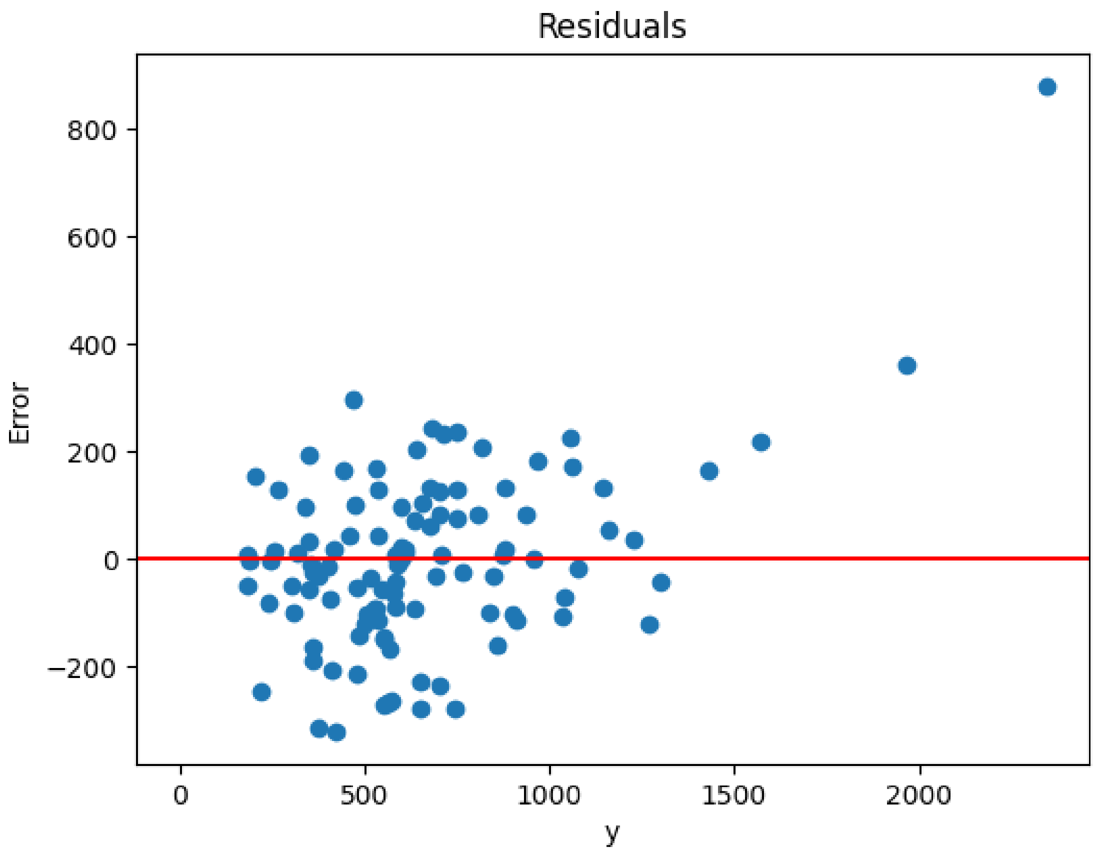
[ ]:

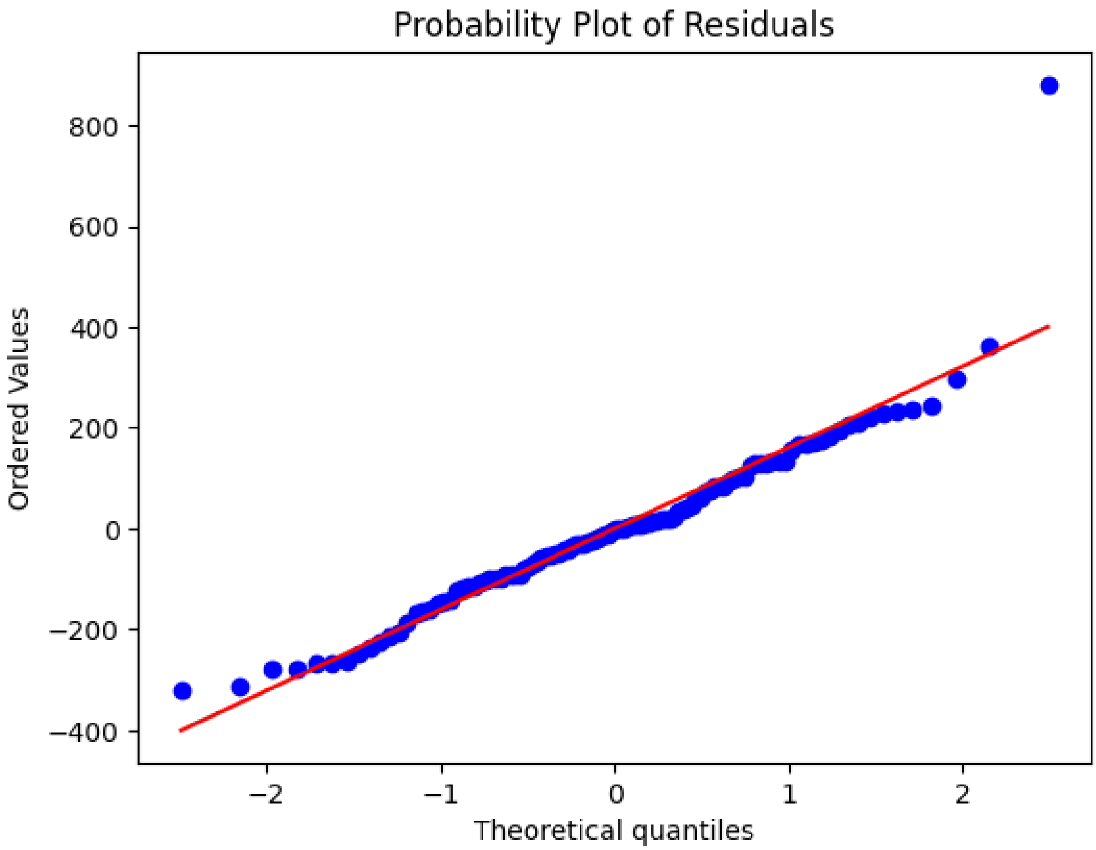
```
model.fit(x_selected,y)
y_pred = model.predict(x_selected)
```

[ ]:
```python
r = y - y_pred

# Plot residuals
plt.scatter(y, r)
plt.axline((0, 0), slope = 0, color = 'red')
plt.xlabel('y')
plt.ylabel('Error')
plt.title('Residuals')
plt.show()

# Q-Q plot for residuals
stats.probplot(r, dist="norm", plot=plt)
plt.title('Probability Plot of Residuals')
plt.show()
```





4.1 Interpretar la tabla ANOVA, R2, R2 ajustada, p-values y FIV.

```
[ ]: # Ajustar el modelo
X_with_const = sm.add_constant(x_selected)
modelANOVA = sm.OLS(y, X_with_const)
results = modelANOVA.fit()

# Imprimir el resumen del modelo
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.772
Model:                            OLS   Adj. R-squared:                  0.761
Method:                 Least Squares   F-statistic:                     69.01
Date:                Tue, 03 Sep 2024   Prob (F-statistic):           3.63e-31
Time:                        19:23:20   Log-Likelihood:                -704.94
No. Observations:                 108   AIC:                             1422.
Df Residuals:                     102   BIC:                             1438.
Df Model:                           5
Covariance Type:            nonrobust
```

```
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        615.1143     18.341     33.537      0.000     578.734     651.495
x1            71.3024     20.276      3.517      0.001      31.085     111.520
x2           134.3976     18.658      7.203      0.000      97.389     171.406
x3           183.9036     20.848      8.821      0.000     142.551     225.256
x4            79.8884     24.987      3.197      0.002      30.326     129.451
x5           221.2692     42.488      5.208      0.000     136.994     305.545
==============================================================================
Omnibus:                       44.132   Durbin-Watson:                   1.744
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              187.195
Skew:                           1.285   Prob(JB):                     2.24e-41
Kurtosis:                       8.916   Cond. No.                         3.52
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[ ]:
```python
# Add a constant to the model (intercept term)
x_const = sm.add_constant(x)
```

[ ]:
```python
# Fit the model using statsmodels
model2 = sm.OLS(y, x_const).fit()
```

[ ]:
```python
# Calculate R^2 with the training set
print("R²: ", r2_score(y, y_pred),'\n')
```

```
R²:  0.7718284123679723
```

[ ]:
```python
# Adjusted R²
adjusted_r2 = model2.rsquared_adj
print(f'Adjusted R²: {adjusted_r2}','\n')
```

```
Adjusted R²: 0.7571187501040445
```

[ ]:
```python
# P-values for the coefficients
p_values = model2.pvalues
print('P-values:')
print(p_values,'\n')
```

```
P-values:
[6.97458724e-33 4.85628994e-04 1.50571066e-10 4.47571564e-14
 3.72991129e-03 6.58677691e-01 7.01993996e-01 2.86739421e-01
 1.71168229e-04]
```

[ ]:
```
# Compute VIF for each feature
vif_data = pd.DataFrame()
vif_data['Feature'] = pd.DataFrame(x_const).columns
vif_data['VIF'] = [variance_inflation_factor(x_const, i) for i in range(x_const.
 ↪shape[1])]

print(vif_data)
```

```
   Feature       VIF
0        0  4.599619
1        1  1.577947
2        2  1.323686
3        3  1.629249
4        4  2.401849
5        5  1.020800
6        6  1.068036
7        7  1.363438
8        8  1.444596
```

$R^2 = 0{,}7718$ significa que el modelo explica aproximadamente el 77,18% de la varianza en la variable objetivo. Esto sugiere un ajuste del modelo relativamente fuerte, ya que el modelo captura una alta proporción de la varianza de los datos.

R^2 ajustado $= 0{,}7571$ es ligeramente inferior a R^2, lo que indica que al tener en cuenta el número de predictores, el modelo todavía explica alrededor del 75,71% de la varianza.

Los primeros cuatro predictores tienen valores p extremadamente bajos ($< 0{,}05$), lo que indica que son estadísticamente significativos y tienen una fuerte relación con la variable objetivo. Los siguientes cuatro predictores (6.587e-01, 7.020e-01, 2.867e-01) tienen valores de p superiores a 0,05, lo que significa que estos predictores no son estadísticamente significativos y es posible que no contribuyan mucho al modelo.

Todos los valores de VIF están por debajo de 5, lo que indica una baja multicolinealidad entre los predictores. Esto significa que los predictores no están altamente correlacionados entre sí y los coeficientes del modelo son confiables.

3. Probar si se deben agregar interacciones o términos polinomiales.

[ ]:
```
# Loop through each column and create squared columns
for col in x_df.columns:
    squared_col_name = f'{col}_squared'  # New column name for squared values
    x_df[squared_col_name] = x_df[col] ** 2

# Display the updated DataFrame with squared columns
x_df
```

[ ]:
```
     Factor Coagulación  Índice pronóstico  Función de enzima  \
0                   6.7                 62                 81
1                   5.1                 59                 66
2                   7.4                 57                 83
```

```
3                    6.5                  73                 41
4                    7.8                  65                115
..                   …                  …                 …
103                  5.8                  70                 64
104                  5.4                  64                 81
105                  6.9                  90                 33
106                  7.9                  45                 55
107                  4.5                  68                 60

     Función de hígado  Edad  Género  Alcohol (moderado)  Alcohol (severo)  \
0                 2.59    50       0                   1                 0
1                 1.70    39       0                   0                 0
2                 2.16    55       0                   0                 0
3                 2.01    48       0                   0                 0
4                 4.30    45       0                   0                 1
..                 …   …     …                 …               …
103               2.52    49       0                   1                 0
104               1.36    62       0                   1                 0
105               2.78    48       1                   0                 0
106               2.46    43       0                   1                 0
107               2.07    59       0                   0                 0

     Factor Coagulación_squared  Índice pronóstico_squared  \
0                         44.89                       3844
1                         26.01                       3481
2                         54.76                       3249
3                         42.25                       5329
4                         60.84                       4225
..                          …                        …
103                       33.64                       4900
104                       29.16                       4096
105                       47.61                       8100
106                       62.41                       2025
107                       20.25                       4624

     Función de enzima_squared  Función de hígado_squared  Edad_squared  \
0                         6561                     6.7081          2500
1                         4356                     2.8900          1521
2                         6889                     4.6656          3025
3                         1681                     4.0401          2304
4                        13225                    18.4900          2025
..                         …                       …           …
103                       4096                     6.3504          2401
104                       6561                     1.8496          3844
105                       1089                     7.7284          2304
106                       3025                     6.0516          1849
107                       3600                     4.2849          3481
```

```
     Género_squared  Alcohol (moderado)_squared  Alcohol (severo)_squared
0                 0                           1                         0
1                 0                           0                         0
2                 0                           0                         0
3                 0                           0                         0
4                 0                           0                         1
..              ...                         ...                       ...
103               0                           1                         0
104               0                           1                         0
105               1                           0                         0
106               0                           1                         0
107               0                           0                         0

[108 rows x 16 columns]
```

[ ]:
```
x=x_df.to_numpy()
#Scale data
scaler=StandardScaler()
x=scaler.fit_transform(x)
x
```

[ ]:
```
array([[ 0.57766012, -0.04760235,  0.43832891, ..., -0.96362411,
         0.98164982, -0.49130368],
       [-0.55024574, -0.22694146, -0.28184638, ..., -0.96362411,
        -1.01869321, -0.49130368],
       [ 1.07111893, -0.34650086,  0.53435228, ..., -0.96362411,
        -1.01869321, -0.49130368],
       ...,
       [ 0.71864835,  1.62622927, -1.866232  , ...,  1.03774904,
        -1.01869321, -0.49130368],
       [ 1.42358951, -1.06385727, -0.80997492, ..., -0.96362411,
         0.98164982, -0.49130368],
       [-0.97321044,  0.31107585, -0.56991649, ..., -0.96362411,
        -1.01869321, -0.49130368]])
```

[ ]:
```
# Define model and RFE selector
selector = RFE(model, n_features_to_select=5)  # Adjust the number of features
 ↪to select

# Fit RFE
selector = selector.fit(x, y)

# Get selected features
x_selected = selector.transform(x)
x_selected
```

```
[ ]: array([[-4.76023543e-02,  4.38328906e-01, -1.84603492e-01,
          3.19401572e-01, -2.15863741e-01],
        [-2.26941457e-01, -2.81846376e-01, -3.61318717e-01,
         -4.12825563e-01, -8.21875833e-01],
        [-3.46500858e-01,  5.34352277e-01, -4.74260679e-01,
          4.28322433e-01, -5.40051085e-01],
        [ 6.09974354e-01, -1.48213851e+00,  5.38322427e-01,
         -1.30112832e+00, -6.39330979e-01],
        [ 1.31736748e-01,  2.07072621e+00,  8.74470809e-04,
          2.53235469e+00,  1.65416949e+00],
        [-1.48231517e+00,  6.22373700e-03, -1.35296861e+00,
         -1.37866802e-01, -9.60534371e-01],
        [-1.00407757e+00, -4.25881432e-01, -1.02582638e+00,
         -5.41338896e-01, -7.01549554e-01],
        [ 3.11075850e-01,  4.38328906e-01,  1.95115172e-01,
          3.19401572e-01, -2.32243733e-01],
        [ 2.51296149e-01,  1.01446913e+00,  1.29394634e-01,
          1.01277584e+00, -2.88573765e-01],
        [ 7.89313456e-01,  1.06248082e+00,  7.55930431e-01,
          1.07487401e+00, -3.66346983e-01],
        [ 1.26755106e+00,  5.34352277e-01,  1.37905850e+00,
          4.28322433e-01,  1.42670663e+00],
        [-7.05179062e-01, -1.38611514e+00, -7.89719262e-01,
         -1.24533959e+00, -7.31468435e-01],
        [ 1.98490747e+00,  2.02271453e+00,  2.43058711e+00,
          2.45630933e+00,  1.19586302e+00],
        [ 1.20777136e+00,  7.74410704e-01,  1.29775976e+00,
          7.12247240e-01,  1.19586302e+00],
        [-4.76023543e-02, -2.33834690e-01, -1.84603492e-01,
         -3.68659482e-01,  5.54233969e-01],
        [ 6.69754055e-01, -1.85823005e-01,  6.09884791e-01,
         -3.23829249e-01, -3.66346983e-01],
        [ 1.32733076e+00, -2.10629042e+00,  1.46133087e+00,
         -1.59900031e+00,  1.28925569e-01],
        [-7.05179062e-01, -1.48213851e+00, -7.89719262e-01,
         -1.30112832e+00, -8.99252250e-01],
        [ 3.11075850e-01,  1.02247108e-01,  1.95115172e-01,
         -4.09006691e-02,  7.30985512e-01],
        [-3.46500858e-01,  7.26399019e-01, -4.74260679e-01,
          6.54133976e-01,  1.67018574e-01],
        [-6.45399362e-01,  1.98270479e-01, -7.39576925e-01,
          5.87220702e-02,  8.63103428e-03],
        [ 1.20777136e+00, -9.05998287e-01,  1.29775976e+00,
         -9.26546822e-01, -1.08147966e+00],
        [-2.19967158e+00, -1.85823005e-01, -1.72684545e+00,
         -3.23829249e-01, -5.80620136e-01],
        [ 2.51296149e-01,  6.78387333e-01,  1.29394634e-01,
```

```
 5.96684863e-01,  5.54233969e-01],
[-2.26941457e-01,  1.35055093e+00, -3.61318717e-01,
  1.46141024e+00,  1.00689130e-01],
[-1.07382055e-01,  5.42354225e-02, -2.44482205e-01,
 -8.97158114e-02,  6.63751358e-01],
[-6.45399362e-01,  6.78387333e-01, -7.39576925e-01,
  5.96684863e-01, -3.27857176e-01],
[ 7.89313456e-01,  8.70434075e-01,  7.55930431e-01,
  8.30466224e-01,  3.67914601e+00],
[-5.25839960e-01, -7.61963230e-01, -6.36371339e-01,
 -8.17958036e-01, -1.14917278e-01],
[ 7.89313456e-01, -6.17928174e-01,  7.55930431e-01,
 -7.03391886e-01, -2.24069610e-01],
[ 7.19570472e-02, -3.29858061e-01, -6.19251545e-02,
 -4.56327492e-01, -1.19366356e+00],
[-1.06385727e+00, -2.34634885e+00, -1.07012689e+00,
 -1.68367964e+00, -2.72638191e-01],
[-2.26941457e-01,  5.42354225e-02, -3.61318717e-01,
 -8.97158114e-02,  6.63751358e-01],
[ 5.50194653e-01,  1.01446913e+00,  4.67733701e-01,
  1.01277584e+00,  4.47890997e-01],
[-2.86721157e-01, -8.97996339e-02, -4.18276517e-01,
 -2.32176329e-01, -1.74358238e-01],
[-7.05179062e-01,  1.30253924e+00, -7.89719262e-01,
  1.39532716e+00, -2.07626129e-01],
[ 6.69754055e-01,  6.78387333e-01,  6.09884791e-01,
  5.96684863e-01, -6.13554713e-01],
[-3.27570619e+00,  2.26277295e+00, -2.02477856e+00,
  2.84317763e+00,  8.63103428e-03],
[-1.07382055e-01,  1.98270479e-01, -2.44482205e-01,
  5.87220702e-02, -3.27857176e-01],
[-6.45399362e-01,  7.74410704e-01, -7.39576925e-01,
  7.12247240e-01, -7.60593712e-01],
[-8.24738464e-01,  6.22373700e-03, -8.87083022e-01,
 -1.37866802e-01, -7.43213778e-01],
[-2.08011218e+00,  1.30253924e+00, -1.67426902e+00,
  1.39532716e+00, -1.01234086e+00],
[ 1.38711046e+00,  7.74410704e-01,  1.54457689e+00,
  7.12247240e-01,  5.22062707e+00],
[-4.06280559e-01,  2.46282164e-01, -5.29271203e-01,
  1.09529667e-01,  8.63103428e-03],
[ 8.49093157e-01,  1.01446913e+00,  8.30413707e-01,
  1.01277584e+00, -9.32916943e-01],
[-1.36275577e+00,  5.82363962e-01, -1.27702488e+00,
  4.83779091e-01,  1.47908584e-01],
[ 6.09974354e-01,  1.63862104e+00,  5.38322427e-01,
  1.87185593e+00,  1.95921642e-01],
```

```
[ 1.38711046e+00,  1.39856262e+00,  1.54457689e+00,
  1.52815748e+00,  1.38751845e+00],
[ 2.51296149e-01,  2.46282164e-01,  1.29394634e-01,
  1.09529667e-01,  1.76939950e-02],
[ 1.14799166e+00,  1.49458599e+00,  1.21743465e+00,
  1.66364440e+00,  2.00533938e+00],
[ 8.49093157e-01, -1.24208008e+00,  8.30413707e-01,
 -1.15667535e+00, -6.77043054e-01],
[ 1.32733076e+00, -1.53015020e+00,  1.46133087e+00,
 -1.32802646e+00, -1.04819590e+00],
[-2.26941457e-01,  6.30375648e-01, -3.61318717e-01,
  5.39899901e-01, -4.18899458e-01],
[ 9.08872858e-01,  6.22373700e-03,  9.05870622e-01,
 -1.37866802e-01,  3.44722442e-01],
[-2.37901068e+00,  2.94293850e-01, -1.79840782e+00,
  1.61001416e-01, -6.89359793e-01],
[ 1.91516449e-01,  9.18445761e-01,  6.46477337e-02,
  8.90571944e-01,  1.95921642e-01],
[ 1.62622927e+00, -1.77020862e+00,  1.88729732e+00,
 -1.45255489e+00, -1.10224035e+00],
[-1.66165427e+00, -8.97996339e-02, -1.45958193e+00,
 -2.32176329e-01, -5.60478459e-01],
[-1.24319637e+00, -1.37811319e-01, -1.19718660e+00,
 -2.78334865e-01, -4.77054778e-01],
[-2.13989188e+00,  5.34352277e-01, -1.70104406e+00,
  4.28322433e-01, -6.26506334e-01],
[-1.72143398e+00, -1.00202166e+00, -1.49317243e+00,
 -9.95618588e-01, -1.02457824e+00],
[ 1.21773464e-02, -1.72219694e+00, -1.23751142e-01,
 -1.42897750e+00, -8.16463452e-01],
[-9.44297865e-01, -2.33834690e-01, -9.80552232e-01,
 -3.68659482e-01, -8.74253715e-01],
[ 3.70855551e-01, -3.29858061e-01,  2.61809348e-01,
 -4.56327492e-01,  6.34849614e-02],
[-1.00407757e+00,  2.94293850e-01, -1.02582638e+00,
  1.61001416e-01,  4.06242645e-01],
[-1.67161756e-01,  6.78387333e-01, -3.03387280e-01,
  5.96684863e-01,  2.54584869e-01],
[-4.06280559e-01, -1.91424368e+00, -5.29271203e-01,
 -1.51930212e+00, -9.09029454e-01],
[-7.05179062e-01, -6.65939859e-01, -7.89719262e-01,
 -7.42244754e-01, -5.26274115e-01],
[ 1.92512777e+00,  4.86340592e-01,  2.33760471e+00,
  3.73529927e-01,  2.19578853e+00],
[-6.45399362e-01, -2.33834690e-01, -7.39576925e-01,
 -3.68659482e-01,  4.27003333e-01],
[-5.85619661e-01, -4.73893118e-01, -6.88460951e-01,
```

```
 -5.82848371e-01, -3.51046293e-01],
[-2.86721157e-01,  5.82363962e-01, -4.18276517e-01,
  4.83779091e-01, -8.00035844e-01],
[ 6.69754055e-01,  3.42305535e-01,  6.09884791e-01,
  2.13137316e-01, -4.77054778e-01],
[-9.44297865e-01, -1.09804503e+00, -9.80552232e-01,
 -1.06203375e+00, -3.51046293e-01],
[ 1.91516449e-01,  2.21476127e+00,  6.46477337e-02,
  2.76447567e+00,  2.21066067e+00],
[-1.07382055e-01, -7.13951545e-01, -2.44482205e-01,
 -7.80433471e-01,  1.09656725e+00],
[ 7.29533755e-01,  1.49458599e+00,  6.82420792e-01,
  1.66364440e+00,  2.54584869e-01],
[-2.86721157e-01,  7.74410704e-01, -4.18276517e-01,
  7.12247240e-01,  6.19563473e-01],
[-4.76023543e-02, -7.13951545e-01, -1.84603492e-01,
 -7.80433471e-01, -1.03257777e+00],
[ 2.04468717e+00, -2.15430211e+00,  2.52454313e+00,
 -1.61726448e+00, -7.83322538e-01],
[ 3.70855551e-01, -5.69916488e-01,  2.61809348e-01,
 -6.63874866e-01,  5.42632798e-02],
[ 6.09974354e-01, -6.65939859e-01,  5.38322427e-01,
 -7.42244754e-01, -1.04433898e+00],
[-7.64958763e-01, -4.73893118e-01, -8.38887961e-01,
 -5.82848371e-01,  3.34580180e-01],
[ 1.50666987e+00,  1.02247108e-01,  1.71398983e+00,
 -4.09006691e-02,  3.54896449e-01],
[-4.66060259e-01, -9.54009972e-01, -5.83308090e-01,
 -9.61414781e-01, -9.65026171e-01],
[ 2.16424657e+00,  5.34352277e-01,  2.71537610e+00,
  4.28322433e-01,  1.17084861e+00],
[-8.84518165e-01, -8.57986601e-01, -9.34304446e-01,
 -8.91014712e-01,  9.13404712e-02],
[-1.24319637e+00, -4.25881432e-01, -1.19718660e+00,
 -5.41338896e-01, -7.37356979e-01],
[-1.67161756e-01,  1.59060936e+00, -3.03387280e-01,
  1.80178793e+00,  3.14390887e-01],
[-4.76023543e-02,  4.86340592e-01, -1.84603492e-01,
  3.73529927e-01,  3.24469661e-01],
[-4.06280559e-01, -1.09804503e+00, -5.29271203e-01,
 -1.06203375e+00, -9.32916943e-01],
[ 4.30635252e-01, -6.17928174e-01,  3.29477161e-01,
 -7.03391886e-01, -4.55484614e-01],
[ 7.19570472e-02, -1.14605671e+00, -6.19251545e-02,
 -1.09424510e+00, -1.01234086e+00],
[ 6.69754055e-01, -1.09804503e+00,  6.09884791e-01,
 -1.06203375e+00, -2.24069610e-01],
```

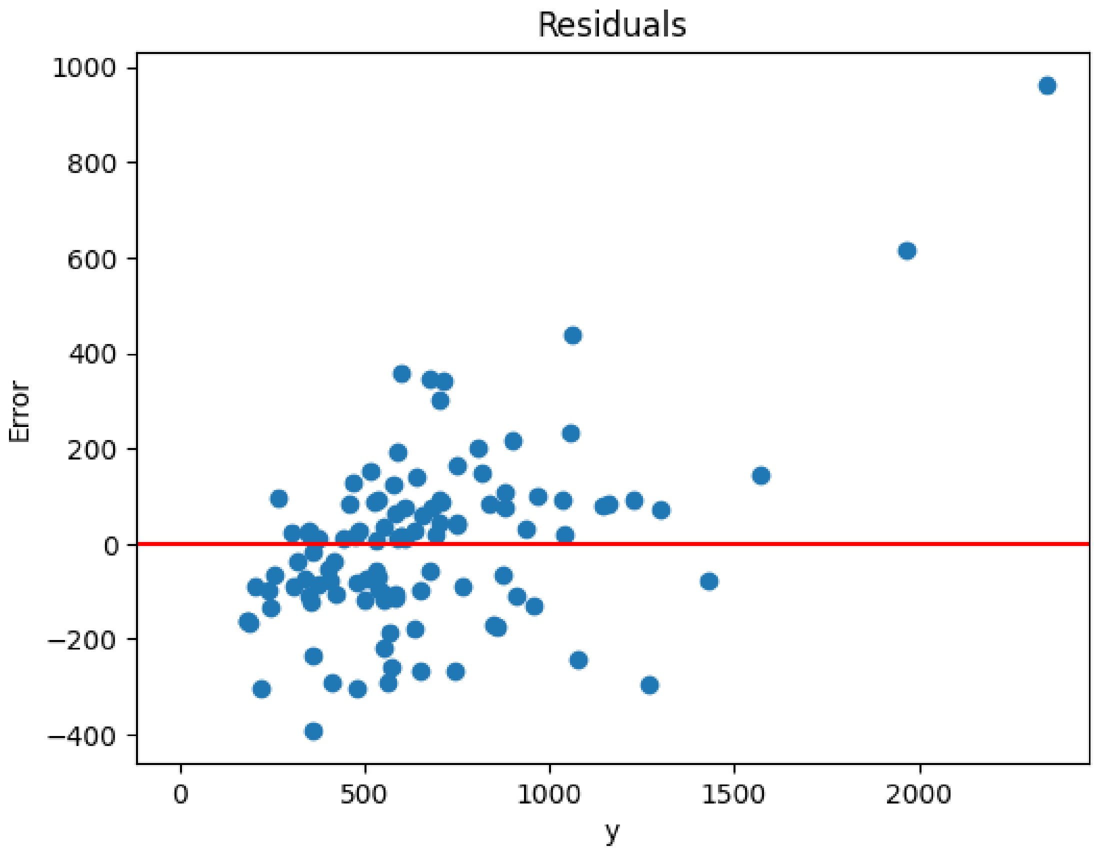
```
[-6.45399362e-01, -1.29009177e+00, -7.39576925e-01,
 -1.18689425e+00, -1.14917278e-01],
[ 5.50194653e-01,  8.70434075e-01,  4.67733701e-01,
  8.30466224e-01,  6.74877690e-01],
[ 6.09974354e-01, -7.13951545e-01,  5.38322427e-01,
 -7.80433471e-01, -1.83673830e-02],
[ 9.08872858e-01, -8.97996339e-02,  9.05870622e-01,
 -2.32176329e-01,  1.62693299e+00],
[ 3.70855551e-01, -8.97996339e-02,  2.61809348e-01,
 -2.32176329e-01,  3.14390887e-01],
[-5.85619661e-01, -9.54009972e-01, -6.88460951e-01,
 -9.61414781e-01, -7.43213778e-01],
[ 1.50666987e+00,  1.25452756e+00,  1.71398983e+00,
  1.32990822e+00,  4.79460575e-01],
[ 1.91516449e-01, -1.85823005e-01,  6.46477337e-02,
 -3.23829249e-01, -7.66323535e-01],
[-4.76023543e-02,  6.30375648e-01, -1.84603492e-01,
  5.39899901e-01,  2.15136256e+00],
[ 4.30635252e-01, -3.77869747e-01,  3.29477161e-01,
 -4.99165270e-01, -2.72638191e-01],
[ 7.19570472e-02,  4.38328906e-01, -6.19251545e-02,
  3.19401572e-01, -9.87009010e-01],
[ 1.62622927e+00, -1.86623200e+00,  1.88729732e+00,
 -1.49771719e+00, -5.39208542e-02],
[-1.06385727e+00, -8.09974916e-01, -1.07012689e+00,
 -8.54818450e-01, -3.20063982e-01],
[ 3.11075850e-01, -5.69916488e-01,  1.95115172e-01,
 -6.63874866e-01, -6.00476114e-01]])
```

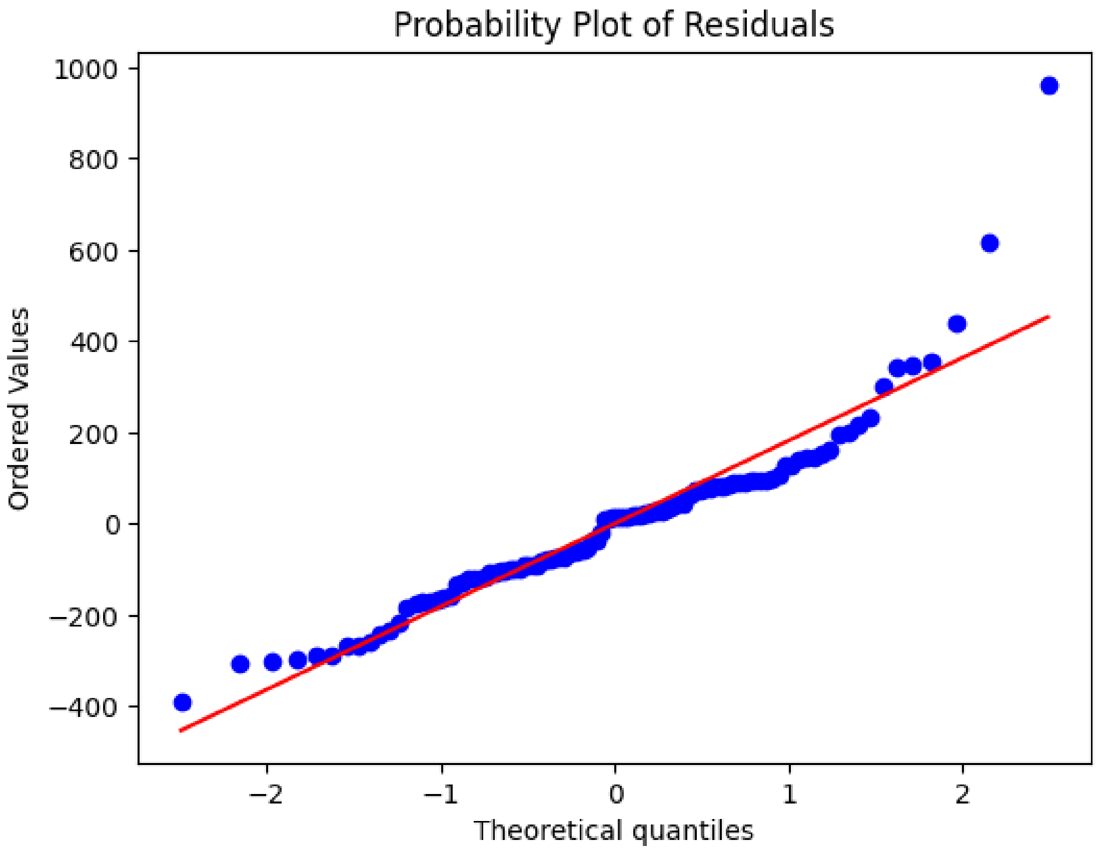
[ ]:
```python
model.fit(x_selected,y)
y_pred = model.predict(x_selected)
```

[ ]:
```python
r = y - y_pred

# Plot residuals
plt.scatter(y, r)
plt.axline((0, 0), slope = 0, color = 'red')
plt.xlabel('y')
plt.ylabel('Error')
plt.title('Residuals')
plt.show()

# Q-Q plot for residuals
stats.probplot(r, dist="norm", plot=plt)
plt.title('Probability Plot of Residuals')
plt.show()
```


Residuals
Error
y
1000
800
600
400
200
0
−200
−400
0
500
1000
1500
2000



4.1 Interpretar la tabla ANOVA, R2, R2 ajustada, p-values y FIV.

[ ]:
```
# Ajustar el modelo
X_with_const = sm.add_constant(x_selected)
modelANOVA = sm.OLS(y, X_with_const)
results = modelANOVA.fit()

# Imprimir el resumen del modelo
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.702
Model:                            OLS   Adj. R-squared:                  0.687
Method:                 Least Squares   F-statistic:                     48.04
Date:                Tue, 03 Sep 2024   Prob (F-statistic):           2.62e-25
Time:                        19:23:21   Log-Likelihood:                -719.37
No. Observations:                 108   AIC:                             1451.
Df Residuals:                     102   BIC:                             1467.
Df Model:                           5
Covariance Type:            nonrobust
```

```
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        658.1389     18.716     35.164      0.000     621.015     695.263
x1           203.1107     95.940      2.117      0.037      12.814     393.407
x2          -148.1411    108.453     -1.366      0.175    -363.257      66.974
x3          -104.2790     97.227     -1.073      0.286    -297.128      88.570
x4           310.6310    109.208      2.844      0.005      94.018     527.244
x5           130.4800     23.038      5.664      0.000      84.785     176.175
==============================================================================
Omnibus:                       48.672   Durbin-Watson:                   2.055
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              186.411
Skew:                           1.498   Prob(JB):                     3.32e-41
Kurtosis:                       8.696   Cond. No.                         14.2
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[ ]:
```
# Add a constant to the model (intercept term)
x_const = sm.add_constant(x)
```

[ ]:
```
# Fit the model using statsmodels
model2 = sm.OLS(y, x_const).fit()
```

[ ]:
```
# Calculate R^2 with the training set
print("R²: ", r2_score(y, y_pred),'\n')
```

```
R²:  0.701928271875816
```

[ ]:
```
# Adjusted R²
adjusted_r2 = model2.rsquared_adj
print(f'Adjusted R²: {adjusted_r2}','\n')
```

```
Adjusted R²: 0.7827365118697556
```

[ ]:
```
# P-values for the coefficients
p_values = model2.pvalues
print('P-values:')
print(p_values,'\n')
```

```
P-values:
[7.03066635e-63 4.79089944e-01 7.28939517e-03 1.81519318e-01
 7.45104552e-01 6.59404034e-01 3.71732085e-01 3.28359912e-01
 1.08536499e-04 1.29442922e-01 2.00073544e-01 9.17204760e-04
 4.97879317e-01 6.66387043e-01 3.71732085e-01 3.28359912e-01
```

```
1.08536499e-04]
```

[ ]:
```python
# Compute VIF for each feature
vif_data = pd.DataFrame()
vif_data['Feature'] = pd.DataFrame(x_const).columns
vif_data['VIF'] = [variance_inflation_factor(x_const, i) for i in range(x_const.
 ↪shape[1])]

print(vif_data)
```

```
    Feature         VIF
0         0    1.000000
1         1   33.051033
2         2   28.519650
3         3   34.244153
4         4   21.466364
5         5  108.670150
6         6         inf
7         7         inf
8         8         inf
9         9   34.672740
10       10   29.302243
11       11   34.461753
12       12   22.598180
13       13  108.203302
14       14         inf
15       15         inf
16       16         inf
```

```
c:\Users\Aviance\AppData\Local\Programs\Python\Python311\Lib\site-
packages\statsmodels\stats\outliers_influence.py:197: RuntimeWarning: divide by
zero encountered in scalar divide
  vif = 1. / (1. - r_squared_i)
```

Conclusión sobre el Modelo de Regresión para Cirugía de Hígado y Sobrevivencia

El Modelo Original presenta un R^2 de 0.7718, lo que indica que el modelo explica el 77.18% de la variabilidad en la variable de interés, en este caso, la "Sobrevivencia" tras la "Cirugía de Hígado". Este valor es superior al R^2 del Modelo Actualizado con términos al cuadrado, que es 0.7019. Esta disminución en el R^2 sugiere que el modelo actualizado no mejora la capacidad predictiva en comparación con el modelo original y podría haber introducido ruido en lugar de mejorar el ajuste.

En el Modelo Original, la mayoría de los valores p estaban por debajo de 0,05, indicando que los predictores eran estadísticamente significativos. Sin embargo, en el Modelo Actualizado, la inclusión de términos al cuadrado resultó en más predictores con valores p superiores a 0,05, lo que sugiere que muchos de estos nuevos términos no aportan significativamente al modelo.

Interpretación: La adición de términos al cuadrado en el Modelo Actualizado probablemente introdujo predictores irrelevantes que no explican significativamente la varianza en la "Sobrevivencia".

Esto ha reducido la efectividad general del modelo.

En términos de Multicolinealidad, el Modelo Original mostró valores de VIF muy por debajo de 5, indicando una baja multicolinealidad. Sin embargo, en el Modelo Actualizado, varios valores de VIF son extremadamente altos (por encima de 10) e incluso alcanzan el infinito (inf), sugiriendo graves problemas de multicolinealidad.

Interpretación: La introducción de términos al cuadrado ha provocado una severa multicolinealidad, indicando que estos términos están altamente correlacionados con los términos originales. Esto hace que los coeficientes de regresión sean inestables y poco confiables, lo que sugiere que la adición de términos al cuadrado ha empeorado significativamente el modelo.