Solución de Problemas - Diego Jafet Garza Segovia

Primero se importara y guardara el archivo "Calificaciones.csv" en la variable AE. Para esto se debera importar la biblioteca de pandas (pd).

Se imprimira las dimensiones de la base de datos para saber su cantidad de filas y columnas usando el atributo shape, los tipo de datos que son sus columnas usando el atributo dtypes, y las primeras 5 filas de datos usando head.

In [115…

```
import pandas as pd

AE = pd.read_csv("Calificaciones.csv")
print("Dimensiones del data frame: ", AE.shape)
print("\nVariable  -  Tipo de dato\n", AE.dtypes)
print("\n",AE.head(5))
```

```
Dimensiones del data frame:  (395, 10)

Variable  -  Tipo de dato
 Escuela           object
Sexo              object
Edad               int64
HorasDeEstudio     int64
Reprobadas         int64
Internet          object
Faltas             int64
G1                 int64
G2                 int64
G3                 int64
dtype: object

   Escuela Sexo  Edad  HorasDeEstudio  Reprobadas Internet  Faltas  G1  G2  G3
0       GP    F    18               2           0       no       6   5   6   6
1       GP    F    17               2           0      yes       4   5   5   6
2       GP    F    15               2           3      yes      10   7   8  10
3       GP    F    15               3           0      yes       2  15  14  15
4       GP    F    16               2           0       no       4   6  10  10
```

Se puede ver que no todas las variables son cuantitativas, lo que representara un problema al querer modelar una funcion que describa el comportamiento de G3. Por lo tanto, se transformaran todas las variables categoricas de forma que sean utiles para generar un modelo de regresion lineal multiple.

Para todas las siguientes columnas nuevas, el tipo de variable sera de int64 y tendra como valor un 1 o 0 segun si es positivo o negativo.

La columna "Escuela" sera dividido en dos columnas, "Escuela_GP" y "Escuela_MS". La columna "Sexo" sera dividido en "Sexo_M" y "Sexo_F". La columna "HorasDeEstudio" sera

divido en "HorasDeEstudio_1", "HorasDeEstudio_2", "HorasDeEstudio_3", y "HorasDeEstudio_4".

De esta forma, se puede saber con el nombre a que variable y categoria se refiere

Adicionalmente, la columna "Internet" sera transformada a tipo int64.

Posteriormente, las columnas "Escuela", "Sexo", y "HorasDeEstudio" seran eliminadas para poder generar el modelo de la regresion lineal multiple, y se imprimira nuevamente las primeras 5 filas para ver los cambios realizados.

In [116…

```
# "Escuela"
AE["Escuela_GP"] = (AE["Escuela"] == "GP").astype("int64")
AE["Escuela_MS"] = (AE["Escuela"] == "MS").astype("int64")

# "Sexo"
AE["Sexo_M"] = (AE["Sexo"] == "M").astype("int64")
AE["Sexo_F"] = (AE["Sexo"] == "F").astype("int64")

# "HorasDeEstudio"
AE["HorasDeEstudio_1"]      = (AE["HorasDeEstudio"] == 1).astype("int64")
AE["HorasDeEstudio_2"]   = (AE["HorasDeEstudio"] == 2).astype("int64")
AE["HorasDeEstudio_3"] = (AE["HorasDeEstudio"] == 3).astype("int64")
AE["HorasDeEstudio_4"]  = (AE["HorasDeEstudio"] == 4).astype("int64")

# "Internet
AE["Internet"] = (AE["Internet"] == "yes").astype("int64")

AE = AE.drop(columns=["Escuela", "Sexo", "HorasDeEstudio"])
print(AE.head(5))
```

```
   Edad  Reprobadas  Internet  Faltas  G1  G2  G3  Escuela_GP  Escuela_MS  \
0    18           0         0       6   5   6   6           1           0
1    17           0         1       4   5   5   6           1           0
2    15           3         1      10   7   8  10           1           0
3    15           0         1       2  15  14  15           1           0
4    16           0         0       4   6  10  10           1           0

   Sexo_M  Sexo_F  HorasDeEstudio_1  HorasDeEstudio_2  HorasDeEstudio_3  \
0       0       1                 0                 1                 0
1       0       1                 0                 1                 0
2       0       1                 0                 1                 0
3       0       1                 0                 0                 1
4       0       1                 0                 1                 0

   HorasDeEstudio_4
0                 0
1                 0
2                 0
3                 0
4                 0
```

Ahora, se buscara identificar si hay valores atipicos para la variable "Faltas". Para esto, se debera obtener el valor del rango interquartil (Q1 y Q3), y se evaluara a k como 3.

Luego se imprimiran todas las filas con una variable atipica en "Faltas".

In [117…
```
Q1 = AE["Faltas"].quantile(0.25)
Q3 = AE["Faltas"].quantile(0.75)
k = 3

VarAtipicas = AE[(AE["Faltas"] < Q1 - k*(Q3 - Q1)) | (AE["Faltas"] > Q3 + k*(Q3-Q1)
print(VarAtipicas)
```

```
     Edad  Reprobadas  Internet  Faltas  G1  G2  G3  Escuela_GP  Escuela_MS  \
74     16           0         1      54  11  12  11           1           0
183    17           0         1      56   9   9   8           1           0
276    18           0         1      75  10   9   9           1           0
307    19           1         1      38   8   9   8           1           0
315    19           1         1      40  13  11  11           1           0

     Sexo_M  Sexo_F  HorasDeEstudio_1  HorasDeEstudio_2  HorasDeEstudio_3  \
74        0       1                 0                 1                 0
183       0       1                 0                 1                 0
276       0       1                 0                 1                 0
307       1       0                 1                 0                 0
315       0       1                 0                 0                 1

     HorasDeEstudio_4
74                  0
183                 0
276                 0
307                 0
315                 0
```

Debido a que de 395 datos unicamente son 5 datos atipicos, se optara por eliminar los dichos datos atipicos de la base de datos. Adicionalmente, debido a que todos los datos atipicos generados son por haber superado el limite superior, se considera que el "hueco" creado no sera de mayor impacto.

Por otro lado, aunque se podria buscar reducir estos numeros de Faltas aproximandolo a los vecinos mas cercanos, debido a que no hay ningun dato atipico que es menor al limite inferior, unicamente se reduciria el numero de faltas total, posiblemente modificando de manera artificial del resultado final.

In [118…
```
AE = AE.drop(VarAtipicas.index)
```

Ahora, se generara una matriz de correlaciones para encontrar potenciales problemas de colinealidad. Consecuentemente se hara un heatmap para poder visualizar de forma mas sencilla los resultados.

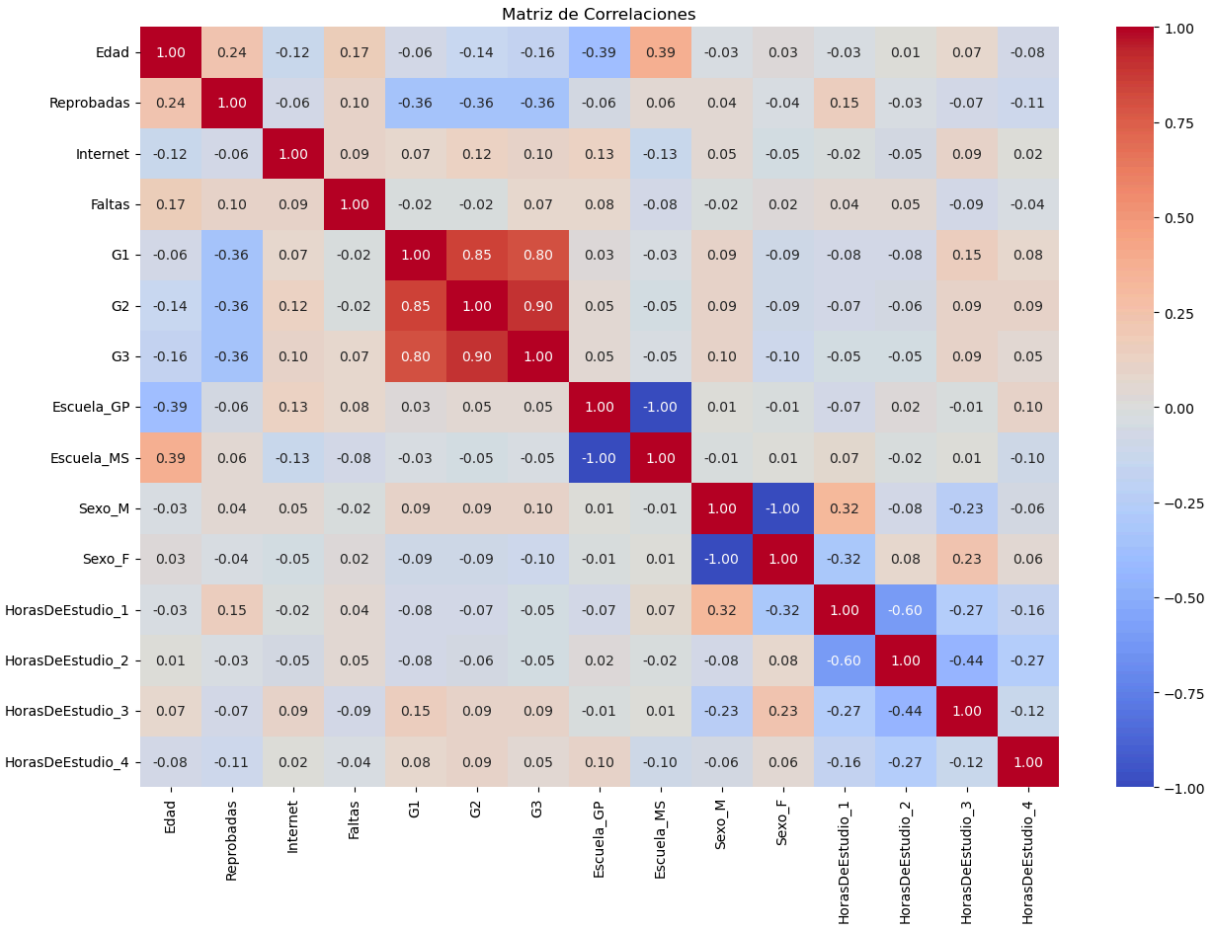
Para esto se deberan importar las bibliotecas matplot.lib (plt) y seaborn (sns).

Para obtener la correlacion se usara la funcion corr(), y para hacer el heatmap se usara sns.heatmap(). Se mantendra "annot = True" para desplegar los valores numericos de estos y

"fmt = '.2f'" para desplegar hasta dos decimales. Se usara el color de paleta azul-rojo, coolwarm, para su visualizacion.

In [119…

```
import matplotlib.pyplot as plt
import seaborn as sns

matriz = AE.corr()
plt.figure(figsize = (15, 10))
sns.heatmap(matriz, annot = True, fmt = ".2f", cmap = "coolwarm")
plt.title ("Matriz de Correlaciones")
plt.show()
```



Se puede ver que "G1", "G2" y "G3" tienen una alta asociacion entre si. Debido a que G1 y G2 sirven para saber como iba el estudiante antes de su calificacion final, se puede sacar el promedio de estos y guardar como "Pre_G3", asi reduciendo la cantidad de columnas con colinealidad.

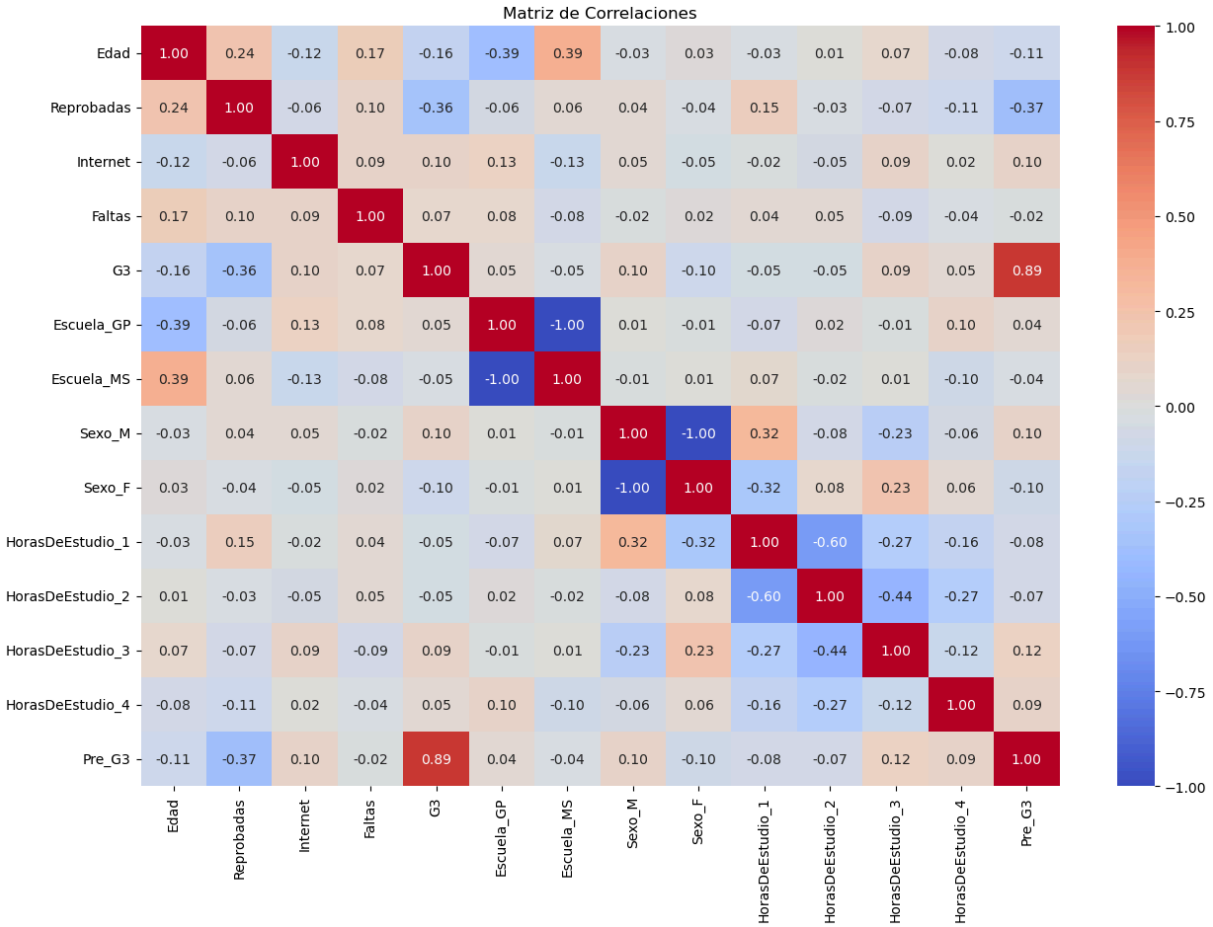
Al hacer esto, todavia se deberia de poder saber como iba el estudiante antes de G3, pero no influira de manera tan pesada en el modelo a generar.

Posteriormente, se mostraran los primeros 5 datos con las columnas cambiadas, y la matriz de correlaciones actualizada.

In [120…

```
AE["Pre_G3"] = ((AE["G1"] + AE["G2"]) / 2)
AE = AE.drop(columns=["G1", "G2"])

print(AE.head(5), "\n")

matriz = AE.corr()
plt.figure(figsize = (15, 10))
sns.heatmap(matriz, annot = True, fmt = ".2f", cmap = "coolwarm")
plt.title ("Matriz de Correlaciones")
plt.show()
```

```
   Edad  Reprobadas  Internet  Faltas  G3  Escuela_GP  Escuela_MS  Sexo_M  \
0    18           0         0       6   6           1           0       0
1    17           0         1       4   6           1           0       0
2    15           3         1      10  10           1           0       0
3    15           0         1       2  15           1           0       0
4    16           0         0       4  10           1           0       0

   Sexo_F  HorasDeEstudio_1  HorasDeEstudio_2  HorasDeEstudio_3  \
0       1                 0                 1                 0
1       1                 0                 1                 0
2       1                 0                 1                 0
3       1                 0                 0                 1
4       1                 0                 1                 0

   HorasDeEstudio_4  Pre_G3
0                 0     5.5
1                 0     5.0
2                 0     7.5
3                 0    14.5
4                 0     8.0
```



Ahora, se buscaran generar dos interacciones entre las variables.

Se buscara ver si hay relacion con la calificion antes de G3 por las faltas, y con la cantidad de materias reprobadas por la edad.

Se imprimiran las primeras 5 filas con las nuevas columnas de interaccion.

In [121…

```
AE["Pre_G3 : Faltas"] = AE["Pre_G3"] * AE["Faltas"]
AE["Reprobadas : Edad"] = AE["Reprobadas"] * AE["Edad"]

print(AE.head(5))
```

```
   Edad  Reprobadas  Internet  Faltas  G3  Escuela_GP  Escuela_MS  Sexo_M  \
0    18           0         0       6   6           1           0       0
1    17           0         1       4   6           1           0       0
2    15           3         1      10  10           1           0       0
3    15           0         1       2  15           1           0       0
4    16           0         0       4  10           1           0       0

   Sexo_F  HorasDeEstudio_1  HorasDeEstudio_2  HorasDeEstudio_3  \
0       1                 0                 1                 0
1       1                 0                 1                 0
2       1                 0                 1                 0
3       1                 0                 0                 1
4       1                 0                 1                 0

   HorasDeEstudio_4  Pre_G3  Pre_G3 : Faltas  Reprobadas : Edad
0                 0     5.5             33.0                  0
1                 0     5.0             20.0                  0
2                 0     7.5             75.0                 45
3                 0    14.5             29.0                  0
4                 0     8.0             32.0                  0
```

Ahora, se realizara el modelo de regresion lineal multiple. Se dividiran los datos aleatoriamente (80% para entrenamiento, y 20% para pruebas).

Para esto se deberan importar las clases LinearRegression de sklearn.linear_model, train_test_split de sklearn.model_selection, y la libreria statsmodels.api (sm)

Primero se separaran las columnas en las variables x e y (G3), y se dividiran entre xEntre, yEntre, xPrueba y yPrueba usando train_test_split.

Se creara el modelo usando la funcion OLS con los parametros yEntre y xEntre. Se calculara el valor de R2 para poder analizar mejor la grafica mas adelante. Se hara una grafica comparando las y predichas con las y verdaderas.

El modelo, el valor de R^2, y la grafica se imprimiran.

In [122…

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import statsmodels.api as sm

x = AE.drop(columns = ["G3"])
y = AE["G3"]

xEntre, xPrueba, yEntre, yPrueba = train_test_split(x, y, test_size=0.2)

xEntre = sm.add_constant(xEntre)
modelo = sm.OLS(yEntre, xEntre).fit()
print(modelo.summary())

xPrueba = sm.add_constant(xPrueba)
yGorrito = modelo.predict(xPrueba)

RSS = np.sum((yPrueba - yGorrito)**2)
TSS = np.sum((yPrueba - np.mean(yPrueba))**2)
```

```
R2 = 1 - (RSS/TSS)
print("\nR^2 = ", R2)

plt.scatter(yPrueba, yGorrito, alpha = 0.5)
plt.plot([yGorrito.min(), yGorrito.max()], [yGorrito.min(), yGorrito.max()], color
plt.xlabel("G3 Real")
plt.ylabel("G3 Estimada")
plt.title("G3 Real vs G3 Estimada")
plt.show()
```

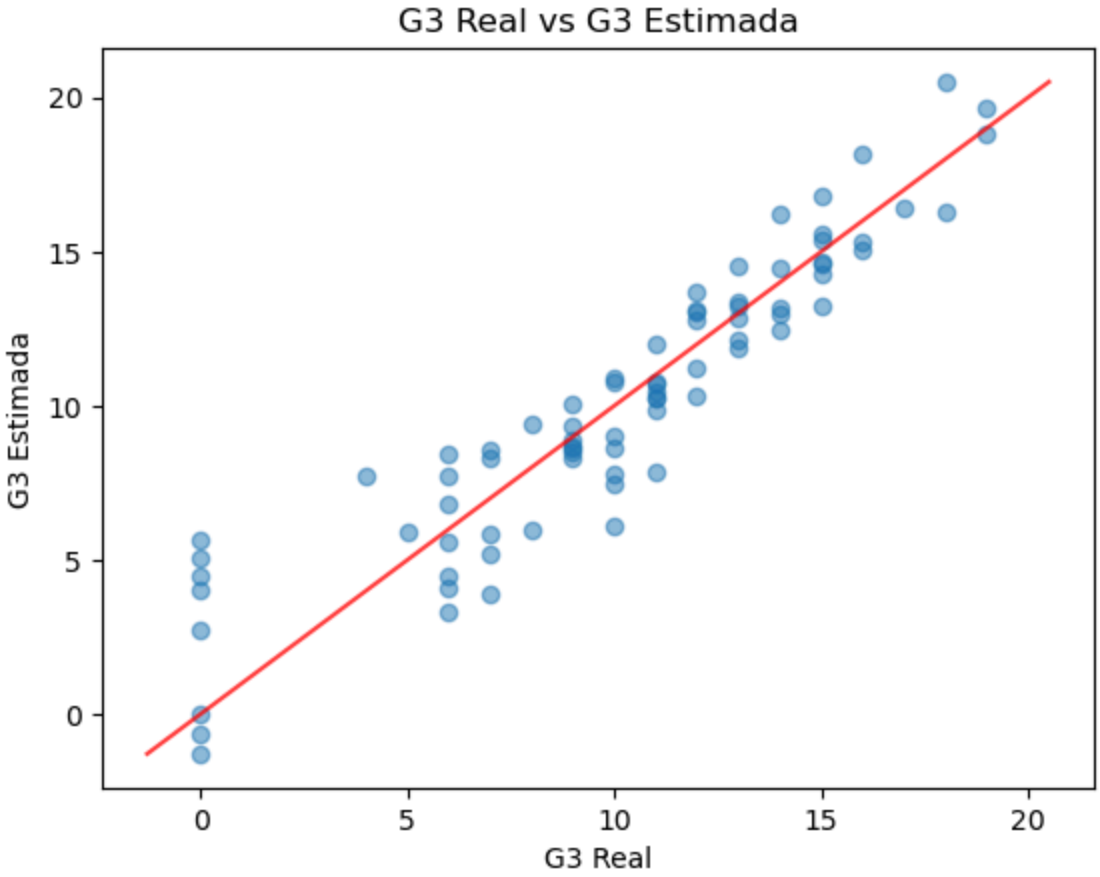
OLS Regression Results

Dep. Variable:	G3	R-squared:	0.807
Model:	OLS	Adj. R-squared:	0.799
Method:	Least Squares	F-statistic:	104.1
Date:	Thu, 28 Aug 2025	Prob (F-statistic):	3.89e-99
Time:	07:44:24	Log-Likelihood:	-658.00
No. Observations:	312	AIC:	1342.
Df Residuals:	299	BIC:	1391.
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.8292	0.980	1.867	0.063	-0.099	3.757
Edad	-0.4763	0.122	-3.905	0.000	-0.716	-0.236
Reprobadas	-3.4428	1.611	-2.137	0.033	-6.614	-0.272
Internet	-0.0417	0.324	-0.129	0.898	-0.679	0.595
Faltas	0.3449	0.070	4.959	0.000	0.208	0.482
Escuela_GP	0.7280	0.444	1.639	0.102	-0.146	1.602
Escuela_MS	1.1013	0.602	1.831	0.068	-0.083	2.285
Sexo_M	0.9024	0.516	1.750	0.081	-0.112	1.917
Sexo_F	0.9269	0.496	1.867	0.063	-0.050	1.904
HorasDeEstudio_1	0.6633	0.307	2.164	0.031	0.060	1.267
HorasDeEstudio_2	0.6269	0.307	2.040	0.042	0.022	1.232
HorasDeEstudio_3	0.6269	0.384	1.633	0.103	-0.128	1.382
HorasDeEstudio_4	-0.0879	0.413	-0.213	0.831	-0.900	0.724
Pre_G3	1.2878	0.048	26.917	0.000	1.194	1.382
Pre_G3 : Faltas	-0.0245	0.006	-3.812	0.000	-0.037	-0.012
Reprobadas : Edad	0.1930	0.093	2.067	0.040	0.009	0.377

Omnibus:	94.219	Durbin-Watson:	2.097
Prob(Omnibus):	0.000	Jarque-Bera (JB):	250.412
Skew:	-1.408	Prob(JB):	4.20e-55
Kurtosis:	6.366	Cond. No.	8.73e+17

```
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
[2] The smallest eigenvalue is 2.92e-30. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

R^2 =  0.8580177466068493
```



Se puede ver que el valor de R^2 es 0.85, mostrando que si hay una relacion establecida entre la G3 Estimada y el G3 Real, significando que el modelo creado es adecuado. Esto es apoyado visualmente al ver que la mayoria de los datos se aproximan a la linea roja.

Sin embargo, en el extremo inferior es donde mas se puede ver dispercion, posiblemente señalando una falta de datos con valores G3 bajos.

Las variables con P < 0.05 son "Edad", "Faltas", "Pre_G3" y "Pre_G3 : Faltas", significando que estas son las variables con las que se puede decir con mas de 95% que tienen un impacto significativo sobre G3.