Actividad de Regresion lineal

Ernesto Reynoso Lizárraga - A01639915

```
import pandas as pd
import numpy as np
import statsmodels.api as sm
import statsmodels.formula.api as smf
from scipy import stats
from sklearn.model_selection import train_test_split
```

```
df = pd.read_csv("/content/drive/MyDrive/Inteligencia Artificial/ds_salaries.csv")
```

```
df.head()
```

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_
0	0	2020	MI	FT	Data Scientist	70000	
1	1	2020	SE	FT	Machine Learning Scientist	260000	
2	2	2020	SE	FT	Big Data Engineer	85000	
3	3	2020	MI	FT	Product Data Analyst	20000	

No se encuentran datos faltantes

```
df.isnull().sum()
```

```
Unnamed: 0            0
work_year             0
experience_level      0
employment_type       0
job_title             0
salary                0
salary_currency       0
salary_in_usd         0
employee_residence    0
remote_ratio          0
company_location      0
company_size          0
dtype: int64
```

Se eliminan los datos que no se van a utilizar

```
df.drop('Unnamed: 0',axis=1,inplace=True)
df.drop('work_year',axis=1,inplace=True)
df.drop('job_title',axis=1,inplace=True)
df.drop('salary_currency',axis=1,inplace=True)
df.drop('employee_residence',axis=1,inplace=True)
df.drop('company_location',axis=1,inplace=True)
df.drop('company_size',axis=1,inplace=True)
```

```
df.head()
```

	experience_level	employment_type	salary	salary_in_usd	remote_ratio
0	MI	FT	70000	79833	0
1	SE	FT	260000	260000	0
2	SE	FT	85000	109024	50
3	MI	FT	20000	20000	0
4	SE	FT	150000	150000	50

```
df['experience_level'].unique()
```

```
array(['MI', 'SE', 'EN', 'EX'], dtype=object)
```

```
df['employment_type'].unique()
```

```
array(['FT', 'CT', 'PT', 'FL'], dtype=object)
```

```
dummies_exp_level=pd.get_dummies(df['experience_level'], prefix='experience_level')
dummies_exp_level
```

	experience_level_EN	experience_level_EX	experience_level_MI	experience_level_
0	0	0	1	
1	0	0	0	
2	0	0	0	
3	0	0	1	
4	0	0	0	
...	...	...	...	
602	0	0	0	
603	0	0	0	
604	0	0	0	
605	0	0	0	
606	0	0	1	

607 rows × 4 columns

```
dummies_emp_type = pd.get_dummies(df['employment_type'],prefix='employment_type')
dummies_emp_type
```

	employment_type_CT	employment_type_FL	employment_type_FT	employment_type_PT
0	0	0	1	0
1	0	0	1	0
2	0	0	1	0
3	0	0	1	0
4	0	0	1	0
...	...	...	...	...
602	0	0	1	0
603	0	0	1	0
604	0	0	1	0
605	0	0	1	0
606	0	0	1	0

607 rows × 4 columns

```
df = pd.concat([df,dummies_exp_level,dummies_emp_type],axis=1)
df.drop('experience_level',axis=1,inplace=True)
df.drop('employment_type',axis=1,inplace=True)
df.head()
```

	salary	salary_in_usd	remote_ratio	experience_level_EN	experience_level_EX	experience_level_MI
0	70000	79833	0	0	0	1
1	260000	260000	0	0	0	0
2	85000	109024	50	0	0	0
3	20000	20000	0	0	0	1
4	150000	150000	50	0	0	0

Correlacion de los datos

```
correlacion = df.corr()
```

```
alta_corr=np.where((correlacion > 0.95) & (correlacion < 1))
alta_corr
```

```
(array([], dtype=int64), array([], dtype=int64))
```

```
baja_corr=np.where((correlacion < -0.95) & (correlacion > -1))
baja_corr
```

```
(array([], dtype=int64), array([], dtype=int64))
```

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
df_estandarizada = scaler.fit_transform(df)
df_estandarizada
```

```
array([[-0.16460538, -0.45790445, -1.74361532, ..., -0.0814463 ,
         0.17975796, -0.12942341],
       [-0.0414754 ,  2.08328151, -1.74361532, ..., -0.0814463 ,
         0.17975796, -0.12942341],
       [-0.15488459, -0.04617667, -0.51437665, ..., -0.0814463 ,
         0.17975796, -0.12942341],
       ...,
       [-0.12637028,  0.2355771 , -1.74361532, ..., -0.0814463 ,
         0.17975796, -0.12942341],
       [-0.11276118,  0.53177399,  0.71486203, ..., -0.0814463 ,
         0.17975796, -0.12942341],
       [-0.08035855,  1.23700468,  0.71486203, ..., -0.0814463 ,
         0.17975796, -0.12942341]])
```

```
df_estandarizada = pd.DataFrame(df_estandarizada, columns=df.columns)
```

```
entrenamiento, prueba=train_test_split(df_estandarizada,test_size=0.20, random_state=42)
```

```
entrenamiento
```

	salary	salary_in_usd	remote_ratio	experience_level_EN	experience_level_EX	experience_level_MI	experience_level_SE	emp
9	-0.128962	0.179159	-0.514377	-0.411773	-0.211543	-0.735261	1.080674	
227	-0.161365	-0.333488	-0.514377	-0.411773	-0.211543	1.360061	-0.925348	
591	-0.116096	0.459192	0.714862	-0.411773	-0.211543	-0.735261	1.080674	
516	-0.111141	0.567036	0.714862	-0.411773	-0.211543	-0.735261	1.080674	
132	-0.185084	-1.042301	0.714862	-0.411773	-0.211543	1.360061	-0.925348	
...	...	...	...	...	...	...	...	
71	-0.185991	-0.988746	-0.514377	-0.411773	-0.211543	1.360061	-0.925348	
106	-0.057677	1.059879	0.714862	-0.411773	-0.211543	1.360061	-0.925348	
270	-0.162985	-0.561334	0.714862	2.428524	-0.211543	-0.735261	-0.925348	
435	-0.164605	-0.291738	0.714862	-0.411773	-0.211543	1.360061	-0.925348	
102	6.918609	-1.072499	-0.514377	-0.411773	-0.211543	1.360061	-0.925348	

485 rows × 11 columns

Generación del modelo

Primer modelo

```
modelo = smf.ols(formula='salary_in_usd~salary+remote_ratio+experience_level_EN+experience_level_EX+experience_level_MI+employment_type_C
modelo = modelo.fit()
print(modelo.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:          salary_in_usd   R-squared:                       0.264
Model:                            OLS   Adj. R-squared:                  0.252
Method:                 Least Squares   F-statistic:                     21.37
```

```
Date:                Thu, 17 Aug 2023   Prob (F-statistic):           8.41e-28
Time:                        00:04:03   Log-Likelihood:                -627.06
No. Observations:                 485   AIC:                             1272.
Df Residuals:                     476   BIC:                             1310.
Df Model:                           8
Covariance Type:            nonrobust
=========================================================================================
                          coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
Intercept               0.0167      0.040      0.413      0.680      -0.063       0.096
salary                 -0.1455      0.065     -2.251      0.025      -0.272      -0.018
remote_ratio            0.0592      0.040      1.463      0.144      -0.020       0.139
experience_level_EN    -0.3717      0.044     -8.423      0.000      -0.458      -0.285
experience_level_EX     0.1902      0.040      4.698      0.000       0.111       0.270
experience_level_MI    -0.3225      0.044     -7.387      0.000      -0.408      -0.237
employment_type_CT      0.0980      0.050      1.974      0.049       0.000       0.196
employment_type_FL     -0.0042      0.058     -0.073      0.941      -0.117       0.109
employment_type_FT      0.0879      0.057      1.531      0.126      -0.025       0.201
==============================================================================
Omnibus:                      242.000   Durbin-Watson:                   1.979
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             2005.484
Skew:                           2.000   Prob(JB):                         0.00
Kurtosis:                      12.123   Cond. No.                         2.46
==============================================================================

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

Segundo modelo eliminando las variables con un p-valor > 0.05

```
modelo2 = smf.ols(formula='salary_in_usd~salary+experience_level_EN+experience_level_EX+experience_level_MI+employment_type_CT',data=entr
modelo2 = modelo2.fit()
print(modelo2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:          salary_in_usd   R-squared:                       0.256
Model:                            OLS   Adj. R-squared:                  0.249
Method:                 Least Squares   F-statistic:                     33.04
Date:                Thu, 17 Aug 2023   Prob (F-statistic):           5.65e-29
Time:                        00:19:25   Log-Likelihood:                -629.63
No. Observations:                 485   AIC:                             1271.
Df Residuals:                     479   BIC:                             1296.
Df Model:                           5
Covariance Type:            nonrobust
=========================================================================================
                          coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
Intercept               0.0179      0.041      0.443      0.658      -0.062       0.098
salary                 -0.1474      0.065     -2.282      0.023      -0.274      -0.020
experience_level_EN    -0.3904      0.043     -9.147      0.000      -0.474      -0.307
experience_level_EX     0.1920      0.041      4.734      0.000       0.112       0.272
experience_level_MI    -0.3292      0.044     -7.549      0.000      -0.415      -0.244
employment_type_CT      0.0583      0.041      1.428      0.154      -0.022       0.139
==============================================================================
Omnibus:                      239.133   Durbin-Watson:                   1.991
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1932.279
Skew:                           1.980   Prob(JB):                         0.00
Kurtosis:                      11.941   Cond. No.                         1.87
==============================================================================

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

Debido a que la R^2 es menor en el segundo modelo nos quedaremos con el primero donde R^2 = 0.264

Ecuación matematica que describe el modelo

Ecuacion:

y = -0.1455 * x1 - 0.3717 * x2 + 0.1902 * x3 - 0.3225 * x4 + 0.0980 * x5

Variable de respuesta:

- salary_in_usd

Variables regresoras:

- x1 = salary
- x2 = experience_level_EN
- x3 = experience_level_EX
- x4 = experience_level_MI

- x5 = employment_type_CT

```
y_pre = -0.1455 * prueba['salary'] - 0.3717 * prueba['experience_level_EN'] + 0.1902 * prueba['experience_level_EX'] - 0.3225 * prueba['e
```

```
y_pre
```

```
563    0.358337
289    0.358832
76    -0.313609
78     0.754617
182   -0.306254
         ...
249    0.355532
365    0.358493
453   -0.315495
548    0.362222
235   -0.314552
Length: 122, dtype: float64
```

Interpretación de los datos

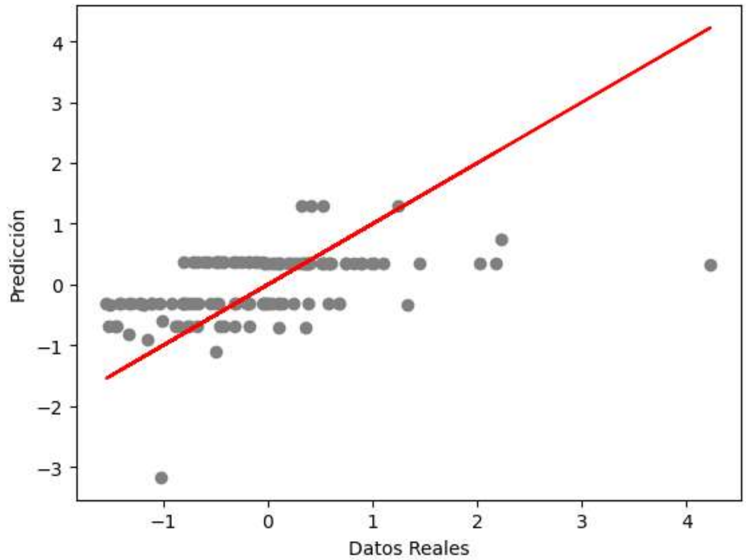
```
tabla = pd.DataFrame({'Real':prueba['salary_in_usd'], 'Prediccion': y_pre, 'Errores':prueba['salary_in_usd']-y_pre})
tabla
```

	Real	Prediccion	Errores
563	0.394254	0.358337	0.035917
289	0.320205	0.358832	-0.038627
76	-0.173457	-0.313609	0.140152
78	2.224328	0.754617	1.469711
182	-1.217128	-0.306254	-0.910873
...	...	...	...
249	0.813866	0.355532	0.458334
365	0.370981	0.358493	0.012489
453	0.108636	-0.315495	0.424130
548	-0.186856	0.362222	-0.549078
235	-0.032411	-0.314552	0.282141

122 rows × 3 columns

```
import matplotlib.pyplot as plt
plt.scatter(prueba['salary_in_usd'],y_pre, color='gray')
plt.plot(prueba['salary_in_usd'], prueba['salary_in_usd'], color='red')
plt.xlabel("Datos Reales")
plt.ylabel("Predicción")
```

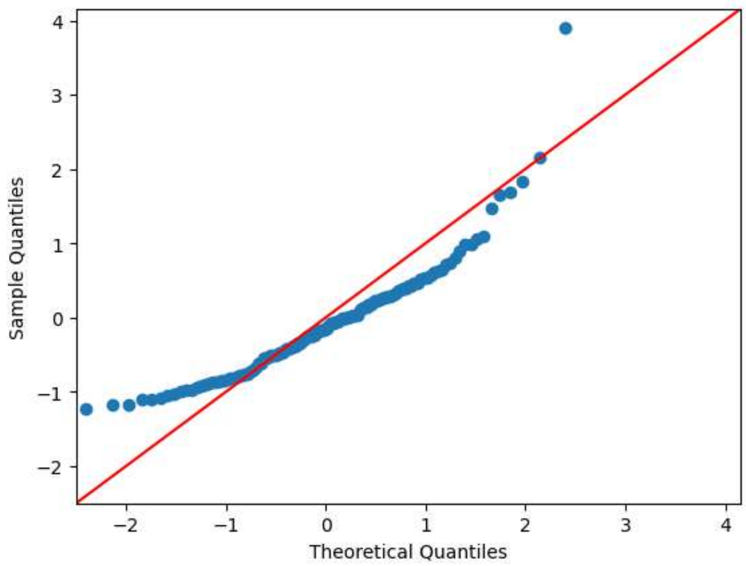
```
Text(0, 0.5, 'Predicción')
```



Como se puede apreciar, el modelo no se apega a los datos reales del dataset, esto se puede deber a que R^2 del modelo es muy bajo como para ser un modelo confiable

QQplot de los errores

```
import statsmodels.api as sm
from scipy import stats
QQ=sm.qqplot(tabla['Errores'],stats.norm, line='45')
```



✓ 0 s completado a las 18:19