

Regresion Lineal Multiple

Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + e

Conceptos Basicos:

- Root Mean Squared Error (RMSE): la raiz cuadrada de la media del error al cuadrado, metrica para comprar modelos de Regresion
- Residuals Standard Error (RSE): lo mismo que RMSE solo que ajustado a los grados de libertad
- R-Squared (Coeficient of determination , R2): la proporcion de la varianza explicada por el modelo, rango entre 0 y 1
- t-Statistic : nos indica la importancia o el peso de la caracteristica que usa el modelo, El coeficiente para un Predictor dividido por el error estandar del coeficiente
- Weighted Regresion : la regresion con los registros teniendo diferentes pesos

Primer modelo de Regresion Lineal Multiple

```
In [1]: #importamos las librerias
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import linear_model
import numpy as np
```

```
In [2]: data=pd.read_csv('datasets\House_sales.csv', delimiter= "\t")
data
```

	DocumentDate	SalePrice	PropertyID	PropertyType	ym	zhvi_px	zhvi_idx	AdjSalePrice	NbrLivingUnits	SqFtLot	...	Bathrooms	Bedrooms	BldgGrade	YrBuilt	YrRenovated	TrafficNoise	LandVal	ImpsVal	...
1	2014-09-16	280000	1000102	Multiplex	2014-09-01	405100	0.930836	300805.0	2	9373	...	3.00	6	7	1991	0	0	70000	229000	...
2	2006-06-16	1000000	1200013	Single Family	2006-06-01	404400	0.929228	1076162.0	1	20156	...	3.75	4	10	2005	0	0	203000	590000	...
3	2007-01-29	745000	1200019	Single Family	2007-01-01	425600	0.977941	761805.0	1	26036	...	1.75	4	8	1947	0	0	183000	275000	...
4	2008-02-25	425000	2800016	Single Family	2008-02-01	418400	0.961397	442065.0	1	8618	...	3.75	5	7	1966	0	0	104000	229000	...
5	2013-03-29	240000	2800024	Single Family	2013-03-01	351600	0.807904	297065.0	1	8620	...	1.75	4	7	1948	0	0	104000	205000	...
...
27057	2011-04-08	325000	9842300710	Single Family	2011-04-01	318700	0.732307	443803.0	1	5468	...	1.75	3	7	1951	0	0	201000	172000	...
27058	2007-09-28	1580000	9845500010	Single Family	2007-09-01	433500	0.996094	1586196.0	1	23914	...	4.50	4	11	2000	0	1	703000	951000	...
27061	2012-07-09	165000	9899200010	Single Family	2012-07-01	325300	0.747472	220744.0	1	11170	...	1.00	4	6	1971	0	0	92000	130000	...
27062	2006-05-26	315000	9900000355	Single Family	2006-05-01	400600	0.920496	342207.0	1	6223	...	2.00	3	7	1939	0	0	103000	212000	...
27063	2007-01-09	465000	9906000035	Single Family	2007-01-01	425600	0.977941	475489.0	1	4400	...	1.50	3	7	1928	0	0	311000	268000	...

22687 rows x 22 columns

```
In [3]: #hacemos un subset con los valores que son de nuestro interes
subset =['AdjSalePrice','SqFtTotLiving','SqFtLot','Bathrooms','Bedrooms']
data=data[subset]
```

```
In [4]: #generamos 2 variables predictors , contendran nuestros X y outcome que contendra nuestra Y o target
predictors=['SqFtTotLiving','SqFtLot','Bathrooms','Bedrooms']
outcome= 'AdjSalePrice'

house_lm = linear_model.LinearRegression() #creamos el modelo
house_lm.fit(data[predictors],data[outcome]) #lo ajustamos con nuestro X e Y
```

```
Out[4]: LinearRegression()
```

```
In [5]: print('Intercept',house_lm.intercept_)

for name,coef in zip(predictors,house_lm.coef_): # la funcion zip toma iterables, los agrega y los imprime
    print(name,coef)

Intercept 96960.38147618511
SqFtTotLiving 327.82525951863505
SqFtLot -0.08468587956197739
Bathrooms 13256.96375140121
Bedrooms -71714.78530489167
```

```
In [6]: fitted= house_lm.predict(data[predictors])# pronosticamos
```

Atributos para medir el desempeño de nuestros modelos

RSE: Error Residual Estandar :

RSE = \sqrt{\frac{\sum_{i=1}^n (\bar{y} - y_i)^2}{(n - 1 - p)}}

siendo p= predictors

RMSE: Raiz Cuadrada de la media del error:

RSE = \sqrt{\frac{\sum_{i=1}^n (\bar{y} - y_i)^2}{n}}

Volviendo al codigo calcularemos estos valores :

```
In [8]: from sklearn.metrics import mean_squared_error

RMSE = np.sqrt(mean_squared_error(data[outcome],fitted))
RSE= np.sqrt(np.sum((data[outcome]-fitted)**2)/(data[outcome].size-1-len(predictors)))
print("RMSE: ",round(RMSE,3))
print("RSE: ",round(RSE,3))

RMSE: 272275.566
RSE: 272305.575
```

```
In [ ]:
```