**Logistic Regression**

**Bank\_Loan\_Modelling**

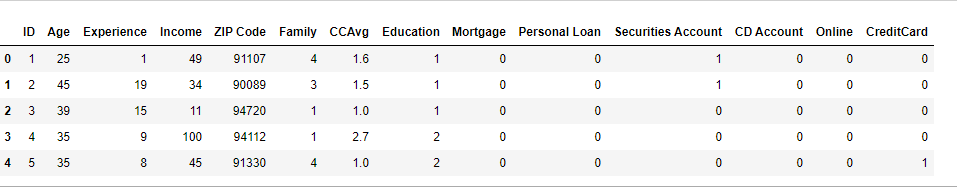
We are checking based on given data whether bank loan can be approved or not

import pandas as pd

dataset=pd.read\_excel("C:/Users/PC/Downloads/Bank\_Personal\_Loan\_Modelling.xlsx",sheet\_name=1)

dataset.head()

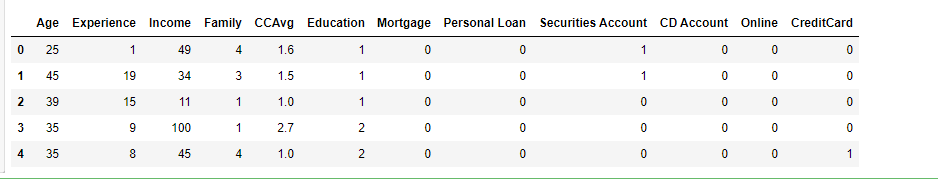
Output:



data1=dataset.drop(columns=['ID','ZIP Code']) // Removed unnecessary columns from dataset

data1.head()

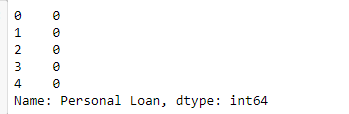
Output:



Y=data1['Personal Loan']

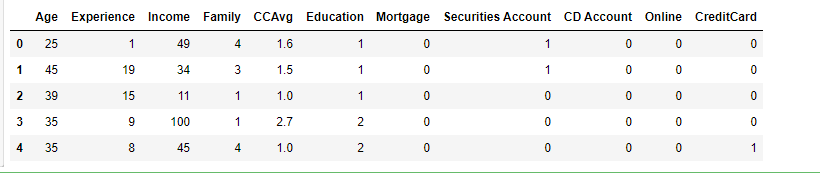
Y.head()

Output:



X=data1[['Age','Experience','Income','Family','CCAvg','Education','Mortgage','Securities Account','CD Account','Online','CreditCard']]

X.head()



import statsmodels.api as sm

X1=sm.add\_constant(X)

Logistic=sm.Logit(Y,X1)

result=Logistic.fit()

result.summary()

Output:

Optimization terminated successfully.

Current function value: 0.128435

Iterations 9

Out[18]:

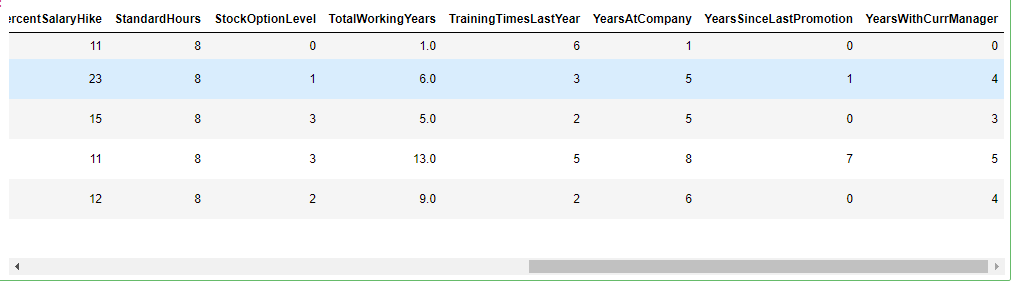
|  |  |  |  |
| --- | --- | --- | --- |
| Logit Regression Results | | | |
| **Dep. Variable:** | Personal Loan | **No. Observations:** | 5000 |
| **Model:** | Logit | **Df Residuals:** | 4988 |
| **Method:** | MLE | **Df Model:** | 11 |
| **Date:** | Wed, 25 Nov 2020 | **Pseudo R-squ.:** | 0.5938 |
| **Time:** | 17:05:10 | **Log-Likelihood:** | -642.18 |
| **converged:** | True | **LL-Null:** | -1581.0 |
| **Covariance Type:** | nonrobust | **LLR p-value:** | 0.000 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **coef** | **std err** | **z** | **P>|z|** | **[0.025** | **0.975]** |
| **const** | -12.1928 | 1.645 | -7.411 | 0.000 | -15.417 | -8.968 |
| **Age** | -0.0536 | 0.061 | -0.874 | 0.382 | -0.174 | 0.067 |
| **Experience** | 0.0638 | 0.061 | 1.046 | 0.295 | -0.056 | 0.183 |
| **Income** | 0.0546 | 0.003 | 20.831 | 0.000 | 0.049 | 0.060 |
| **Family** | 0.6958 | 0.074 | 9.364 | 0.000 | 0.550 | 0.841 |
| **CCAvg** | 0.1240 | 0.040 | 3.127 | 0.002 | 0.046 | 0.202 |
| **Education** | 1.7362 | 0.115 | 15.088 | 0.000 | 1.511 | 1.962 |
| **Mortgage** | 0.0005 | 0.001 | 0.856 | 0.392 | -0.001 | 0.002 |
| **Securities Account** | -0.9368 | 0.286 | -3.277 | 0.001 | -1.497 | -0.377 |
| **CD Account** | 3.8225 | 0.324 | 11.800 | 0.000 | 3.188 | 4.457 |
| **Online** | -0.6752 | 0.157 | -4.298 | 0.000 | -0.983 | -0.367 |
| **CreditCard** | -1.1197 | 0.205 | -5.462 | 0.000 | -1.522 | -0.718 |

**Attrition Data set**

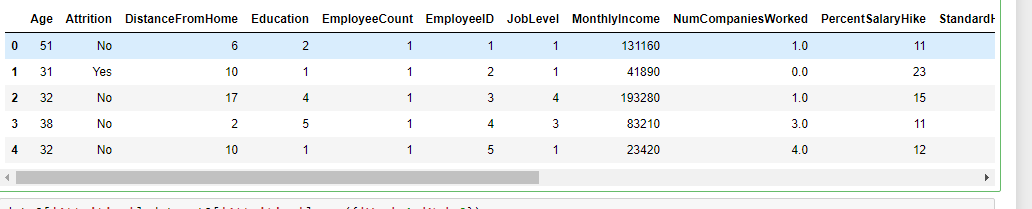
dataset2=pd.read\_csv("C:/Users/PC/Downloads/general\_data.csv")

dataset2.head()



data2=dataset2.drop(columns=['BusinessTravel','Department','EducationField','Gender','Over18','JobRole','MaritalStatus'])

data2.head()



Y1=data2['Attrition']

Y1.head()

0 0

1 1

2 0

3 0

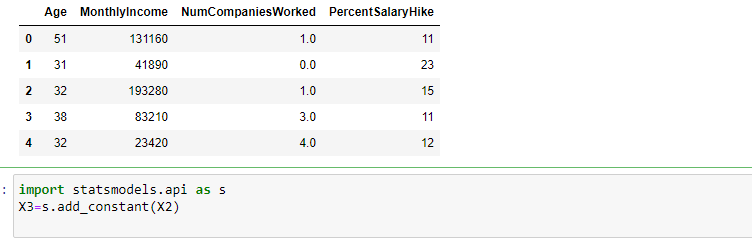
4 0

Name: Attrition, dtype: int64

X2=data2[['Age','MonthlyIncome','NumCompaniesWorked','PercentSalaryHike']]

X2.head()

Output:



**Linear Regression**

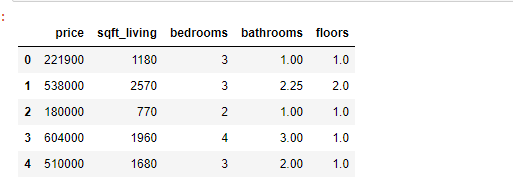
Applying linear regression b/w price and sqt\_living

import pandas as pd

import matplotlib.pyplot as plt

dataset=pd.read\_excel("C:/Users/PC/Downloads/Linear Regression.xlsx",sheet\_name=0)

dataset.head()



Y=dataset.iloc[:,:1] // Assigning sqfr\_living to Y

#X=dataset.sqft\_living

Y.head(1)

Y.shape

(21613, 1)

X=dataset.iloc[:,1:2] // Assigning price value to X

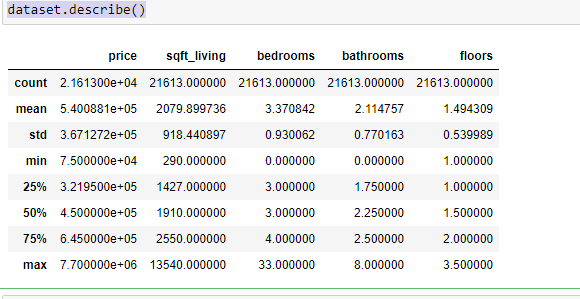
X.head(1)

X.shape

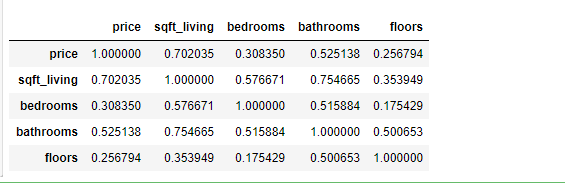
(21613, 1)

X.isna().sum()

dataset.describe()



dataset.corr()



dataset.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 21613 entries, 0 to 21612

Data columns (total 5 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 price 21613 non-null int64

1 sqft\_living 21613 non-null int64

2 bedrooms 21613 non-null int64

3 bathrooms 21613 non-null float64

4 floors 21613 non-null float64

dtypes: float64(2), int64(3)

memory usage: 844.4 KB

from sklearn.model\_selection import train\_test\_split

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,Y,test\_size=0.2)

from sklearn import datasets, linear\_model

lm = linear\_model.LinearRegression()

model = lm.fit(X\_train, y\_train)

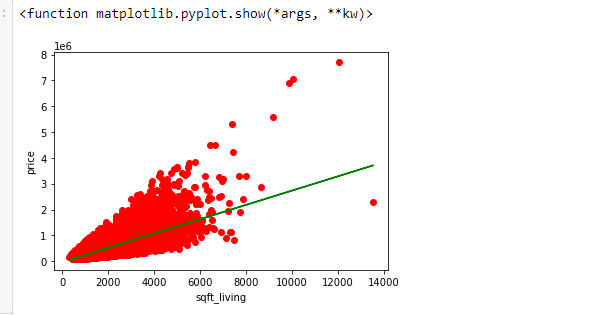
plt.scatter(X\_train,y\_train, color='red')

plt.plot(X\_train,lm.predict(X\_train),color='green')

plt.xlabel('sqft\_living')

plt.ylabel('price')

plt.show



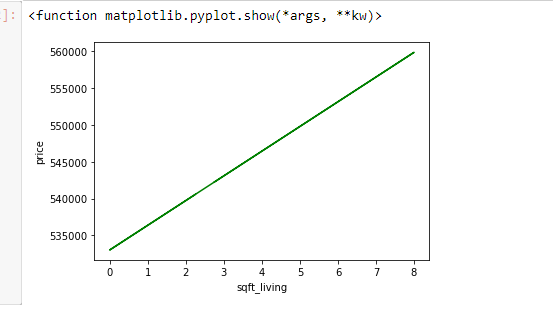
# plt.scatter(X\_test,y\_test, color='red')

plt.plot(X\_test,lm.predict(X\_test),color='green')

plt.xlabel('sqft\_living')

plt.ylabel('price')

plt.show



from sklearn.metrics import mean\_squared\_error,r2\_score

RMSE=np.sqrt(mean\_squared\_error(y\_test,predictions))

r\_square=r2\_score(y\_test,predictions)

print(" R square",r\_square)

print(RMSE)

R square 0.49970134684357326

272614.3763854597

Linear Regression between price and bedroom

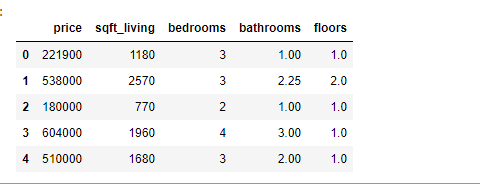
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

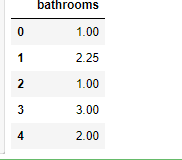
dataset2=pd.read\_excel("C:/Users/PC/Downloads/Linear Regression.xlsx",sheet\_name=0)

dataset2.head()



C=dataset2.iloc[:,3:4]

C.head()



D=dataset2.iloc[:,0:1]

D.head()



from sklearn.model\_selection import train\_test\_split

X\_train,X\_test,Y\_train,Y\_test=train\_test\_split(C,D,test\_size=0.2,random\_state=2)

from sklearn.linear\_model import LinearRegression

lm=LinearRegression()

lm1=lm.fit(X\_train,y\_train)

plt.scatter(X\_train,Y\_train,color='Blue')

plt.plot(X\_train,lm.predict(X\_train),color='red')

plt.xlabel('floor')

plt.ylabel('price')

plt.show



from sklearn.metrics import mean\_squared\_error,r2\_score

RMSE=np.sqrt(mean\_squared\_error(y\_test,lm.predict(X\_test)))

r\_square=r2\_score(y\_test,lm.predict(X\_test))

print(" R square",r\_square)

print(RMSE)

R square -3.723693406487705e-05

367602.8209407581