Day 21 Assignment – Linear and Logistic Regression

Logistic Regression:

Project 1: Bank Loan Modeling

```
import pandas as pd
import statsmodels.api as sm
dataset = pd.read_excel("Bank_Personal_Loan_Modelling.xlsx",sheet_name=1)
dataset.columns
Out[5]:
Index(['ID', 'Age', 'Experience', 'Income', 'ZIP Code', 'Family', 'CCAvg',
       'Education', 'Mortgage', 'Personal Loan', 'Securities Account',
       'CD Account', 'Online', 'CreditCard'],
      dtype='object')

dataset.dropna()
Out[6]:
      ID  Age  Experience  ...  CD Account  Online  CreditCard
0      1   25           1  ...           0       0           0
1      2   45          19  ...           0       0           0
2      3   39          15  ...           0       0           0
3      4   35           9  ...           0       0           0
4      5   35           8  ...           0       0           1
   ...  ...         ...  ...         ...     ...         ...
4995  4996   29           3  ...           0       1           0
4996  4997   30           4  ...           0       1           0
```

Day 21 Assignment – Linear and Logistic Regression

```
4997  4998   63        39  ...        0      0        0
4998  4999   65        40  ...        0      1        0
4999  5000   28         4  ...        0      1        1

[5000 rows x 14 columns]
```

```
dataset1=dataset.drop_duplicates()
dataset2=dataset1.drop(["ID","ZIP Code"],axis=1)
dataset2.columns
Out[10]:
Index(['Age', 'Experience', 'Income', 'Family', 'CCAvg', 'Education',
       'Mortgage', 'Personal Loan', 'Securities Account', 'CD Account',
       'Online', 'CreditCard'],
      dtype='object')

Y = dataset2["Personal Loan"]
X = dataset2[['Age', 'Experience', 'Income', 'Family', 'CCAvg', 'Education',
       'Mortgage', 'Securities Account', 'CD Account',
       'Online', 'CreditCard']]
X1 = sm.add_constant(X)
Logistic = sm.Logit(Y,X1)
result = Logistic.fit()
Optimization terminated successfully.
         Current function value: 0.128435
         Iterations 9
```

Day 21 Assignment – Linear and Logistic Regression

result.summary()

Out[19]:

<class 'statsmodels.iolib.summary.Summary'>

"""

Logit Regression Results

==========

Dep. Variable: Personal Loan No. Observations: 5000

Model: Logit Df Residuals: 4988

Method: MLE Df Model: 11

Date: Tue, 18 Aug 2020 Pseudo R-squ.: 0.5938

Time: 16:53:30 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

Day 21 Assignment – Linear and Logistic Regression

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

==========

"""

The Variables Income, Family , CCAvg ,Education , Securities Account ,CD Account , Online , CreditCard are significantly important for getting the Personal loan.

Project 2: Attrition Analysis

```
import pandas as pd

import statsmodels.api as sm

dataset=pd.read_csv("Attrition_Analysis.csv")

from sklearn import preprocessing

le=preprocessing.LabelEncoder()

dataset["Attrition"]=le.fit_transform(dataset["Attrition"])
dataset["BusinessTravel"]=le.fit_transform(dataset["BusinessTravel"])
dataset["Department"]=le.fit_transform(dataset["Department"])
dataset["EducationField"]=le.fit_transform(dataset["EducationField"])
dataset["Gender"]=le.fit_transform(dataset["Gender"])
dataset["MaritalStatus"]=le.fit_transform(dataset["MaritalStatus"])
dataset["JobRole"]=le.fit_transform(dataset["JobRole"])
dataset1=dataset.drop(['EmployeeCount', 'EmployeeID','Over18','StandardHours'],axis=1)
dataset1.columns
Out[53]:
Index(['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',
       'Education', 'EducationField', 'Gender', 'JobLevel', 'JobRole',
       'MaritalStatus', 'MonthlyIncome', 'NumCompaniesWorked',
       'PercentSalaryHike', 'StockOptionLevel', 'TotalWorkingYears',
       'TrainingTimesLastYear', 'YearsAtCompany', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')
```

Day 21 Assignment – Linear and Logistic Regression

```
dataset2=dataset1.dropna()
dataset3=dataset2.drop_duplicates()
Y=dataset3.Attrition
dataset3.columns
Out[54]:
Index(['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',
       'Education', 'EducationField', 'Gender', 'JobLevel', 'JobRole',
       'MaritalStatus', 'MonthlyIncome', 'NumCompaniesWorked',
       'PercentSalaryHike', 'StockOptionLevel', 'TotalWorkingYears',
       'TrainingTimesLastYear', 'YearsAtCompany', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')

X=dataset3[['Age', 'BusinessTravel', 'Department', 'DistanceFromHome',
'Education', 'EducationField', 'Gender', 'JobLevel', 'JobRole',
'MaritalStatus', 'MonthlyIncome', 'NumCompaniesWorked',
'PercentSalaryHike', 'StockOptionLevel', 'TotalWorkingYears',
'TrainingTimesLastYear', 'YearsAtCompany', 'YearsSinceLastPromotion',
'YearsWithCurrManager']]
X1=sm.add_constant(X)
Logistic_Attrition=sm.Logit(Y,X1)
result=Logistic_Attrition.fit()
Optimization terminated successfully.
         Current function value: 0.392756
         Iterations 7

result.summary()
Out[56]:
<class 'statsmodels.iolib.summary.Summary'>
"""
                           Logit Regression Results
==============================================================================
Dep. Variable:              Attrition   No. Observations:                 1470
Model:                          Logit   Df Residuals:                     1450
Method:                           MLE   Df Model:                           19
Date:                Tue, 18 Aug 2020   Pseudo R-squ.:                  0.1108
Time:                        17:50:00   Log-Likelihood:                -577.35
converged:                       True   LL-Null:                       -649.29
Covariance Type:            nonrobust   LLR p-value:                 3.295e-21
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0650      0.717      0.091      0.928      -1.340       1.470
Age           -0.0306      0.012     -2.583      0.010      -0.054      -0.007
BusinessTravel    -0.0166      0.113     -0.146      0.884      -0.239       0.206
Department       -0.2421      0.141     -1.720      0.085      -0.518       0.034
DistanceFromHome    -0.0014      0.009     -0.145      0.884      -0.020       0.017
Education        -0.0625      0.074     -0.847      0.397      -0.207       0.082
```

Day 21 Assignment – Linear and Logistic Regression

EducationField -0.0965 0.058 -1.669 0.095 -0.210 0.017
Gender 0.0869 0.155 0.560 0.576 -0.217 0.391
JobLevel -0.0249 0.069 -0.363 0.717 -0.159 0.110
JobRole 0.0378 0.031 1.219 0.223 -0.023 0.099
MaritalStatus 0.5885 0.109 5.379 0.000 0.374 0.803
MonthlyIncome -1.868e-06 1.66e-06 -1.128 0.259 -5.11e-06 1.38e-06
NumCompaniesWorked 0.1184 0.032 3.729 0.000 0.056 0.181
PercentSalaryHike 0.0117 0.020 0.576 0.565 -0.028 0.052
StockOptionLevel -0.0645 0.089 -0.721 0.471 -0.240 0.111
TotalWorkingYears -0.0593 0.021 -2.856 0.004 -0.100 -0.019
TrainingTimesLastYear -0.1465 0.061 -2.406 0.016 -0.266 -0.027
YearsAtCompany 0.0136 0.032 0.428 0.669 -0.049 0.076
YearsSinceLastPromotion 0.1323 0.035 3.732 0.000 0.063 0.202
YearsWithCurrManager -0.1396 0.038 -3.642 0.000 -0.215 -0.064
==========
==========
"""

The Variables Age, Marital Status, NumcompaniesWorked, Totalworkingyears, TrainingTimesLastYear , YearsSinceLastPromotion , YearsWithCurrManager are significantly important for the Attrition rate in the company.

Linear Regression:

Project 3: Real Estate Analysis

Simple Linear Regression:

```
import pandas as pd

import numpy as np

dataset = pd.read_excel("Real_Estate_Analysis.xlsx",sheet_name=0)

dataset.head()

Out[60]:

   price  sqft_living  bedrooms  bathrooms  floors

0  221900        1180         3       1.00     1.0

1  538000        2570         3       2.25     2.0

2  180000         770         2       1.00     1.0
```

Day 21 Assignment – Linear and Logistic Regression

3 604000 1960 4 3.00 1.0

4 510000 1680 3 2.00 1.0

dataset.describe()

Out[61]:

	price	sqft_living	bedrooms	bathrooms	floors
count	2.161300e+04	21613.000000	21613.000000	21613.000000	21613.000000
mean	5.400881e+05	2079.899736	3.370842	2.114757	1.494309
std	3.671272e+05	918.440897	0.930062	0.770163	0.539989
min	7.500000e+04	290.000000	0.000000	0.000000	1.000000
25%	3.219500e+05	1427.000000	3.000000	1.750000	1.000000
50%	4.500000e+05	1910.000000	3.000000	2.250000	1.500000
75%	6.450000e+05	2550.000000	4.000000	2.500000	2.000000
max	7.700000e+06	13540.000000	33.000000	8.000000	3.500000

x=dataset.iloc[:,:1]

x.head(2)

Out[62]:

	price
0	221900
1	538000

y=dataset.iloc[:,1:2]

y.head(2)

Out[63]:

	sqft_living
0	1180

Day 21 Assignment – Linear and Logistic Regression

1 2570

```
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,random_state=0)
from sklearn.linear_model import LinearRegression
lin_reg=LinearRegression()
lin_reg.fit(X_train,y_train)
Out[67]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)

lin_reg.coef_
Out[68]: array([[0.00174499]])

lin_reg.intercept_
Out[69]: array([1134.23538784])

ypred=lin_reg.predict(X_test)
ypred
Out[70]:
array([[1652.49798531],
       [3887.83262287],
       [2115.09534083],
       ...,
       [1779.79514519],
       [1657.73296104],
       [2139.26347879]])
```

Day 21 Assignment – Linear and Logistic Regression

```
from sklearn.metrics import mean_squared_error,r2_score

RMSE=np.sqrt(mean_squared_error(y_test,ypred))

r_square=r2_score(y_test,ypred)

print("The R square value is :-",r_square)
```

The R square value is :- 0.4813925927530154

```
print("The RMSE Value is:-",RMSE)
```

The RMSE Value is:- 647.3715598616461

48.13% the model this predicting good
647 .37 is the total mean error value

Multiple Linear Regression:

```
dataset.head(5)
```

Out[73]:

	price	sqft_living	bedrooms	bathrooms	floors
0	221900	1180	3	1.00	1.0
1	538000	2570	3	2.25	2.0
2	180000	770	2	1.00	1.0
3	604000	1960	4	3.00	1.0
4	510000	1680	3	2.00	1.0

```
dataset.describe()
```

Out[74]:

	price	sqft_living	bedrooms	bathrooms	floors
count	2.161300e+04	21613.000000	21613.000000	21613.000000	21613.000000
mean	5.400881e+05	2079.899736	3.370842	2.114757	1.494309
std	3.671272e+05	918.440897	0.930062	0.770163	0.539989
min	7.500000e+04	290.000000	0.000000	0.000000	1.000000
25%	3.219500e+05	1427.000000	3.000000	1.750000	1.000000
50%	4.500000e+05	1910.000000	3.000000	2.250000	1.500000
75%	6.450000e+05	2550.000000	4.000000	2.500000	2.000000
max	7.700000e+06	13540.000000	33.000000	8.000000	3.500000

Day 21 Assignment – Linear and Logistic Regression

```
X=dataset.iloc[:,1:]
X.head()
Out[75]:
   sqft_living  bedrooms  bathrooms  floors
0        1180         3       1.00     1.0
1        2570         3       2.25     2.0
2         770         2       1.00     1.0
3        1960         4       3.00     1.0
4        1680         3       2.00     1.0

y=dataset.iloc[:,0:1]
y.head()
Out[76]:
    price
0  221900
1  538000
2  180000
3  604000
4  510000

from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3,random_state=2)
dataset.shape
Out[77]: (21613, 5)

from sklearn.linear_model import LinearRegression
mul_reg=LinearRegression()
mul_reg.fit(X_train,y_train)
Out[78]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)

ypred=mul_reg.predict(X_test)
ypred
Out[79]:
array([[609328.77721079],
       [585038.87031192],
       [415562.45223517],
       ...,
       [599102.75434259],
       [339784.19873135],
       [516024.79183523]])
```

Day 21 Assignment – Linear and Logistic Regression

```
from sklearn.metrics import r2_score,mean_squared_error
print("The R-square...",r2_score(y_test,ypred))
The R-square... 0.5105722437453338

print("The RMSE value is...", np.sqrt(mean_squared_error(y_test,ypred)))
The RMSE value is... 261133.29646851748
```

51.05% the model this predicting good
261133.29 is the total mean error value