# THE ATTRITION AND THE BANK PERSONAL LOAN –

# LOGISTIC REGRESSION

The codes to execute the attrition logistic regression:

import pandas as pd

df1 = pd.read\_excel(r"C:\letsupgrage assignment\general data.xlsx", sheet\_name=0)

print(df1.head())

print(df1.describe())

df1.drop(["BusinessTravel", "Department", "EmployeeID", "MaritalStatus", "Over18", "EmployeeCount", "EducationField", "Gender", "JobRole", "StandardHours"], axis = 1, inplace = True)

y = df1["Attrition"]

x = df1[["Age", "DistanceFromHome", "Education", "JobLevel", "MonthlyIncome", "NumCompaniesWorked", "PercentSalaryHike", "StockOptionLevel", "TotalWorkingYears", "TrainingTimesLastYear", "YearsAtCompany", "YearsSinceLastPromotion", "YearsWithCurrMa0ger"]]

import statsmodels.api as sm

x1 = sm.add\_constant(x)

print(x1)

logistic = sm.Logit(y, x1)

result = logistic.fit()

result.summary()

The output of this code:

Age Attrition BusinessTravel Department \

0 51 0 Travel\_Rarely Sales

1 31 1 Travel\_Frequently Research & Development

2 32 0 Travel\_Frequently Research & Development

3 38 0 0n-Travel Research & Development

4 32 0 Travel\_Rarely Research & Development

DistanceFromHome Education EducationField EmployeeCount EmployeeID \

0 6 2 Life Sciences 1 1

1 10 1 Life Sciences 1 2

2 17 4 Other 1 3

3 2 5 Life Sciences 1 4

4 10 1 Medical 1 5

Gender ... NumCompaniesWorked Over18 PercentSalaryHike StandardHours \

0 Female ... 1 Y 11 8

1 Female ... 0 Y 23 8

2 Male ... 1 Y 15 8

3 Male ... 3 Y 11 8

4 Male ... 4 Y 12 8

StockOptionLevel TotalWorkingYears TrainingTimesLastYear YearsAtCompany \

0 0 1 6 1

1 1 6 3 5

2 3 5 2 5

3 3 13 5 8

4 2 9 2 6

YearsSinceLastPromotion YearsWithCurrMa0ger

0 0 0

1 1 4

2 0 3

3 7 5

4 0 4

[5 rows x 24 columns]

Age Attrition DistanceFromHome Education EmployeeCount \

count 4410.000000 4410.000000 4410.000000 4410.000000 4410.0

mean 36.923810 0.161224 9.192517 2.912925 1.0

std 9.133301 0.367780 8.105026 1.023933 0.0

min 18.000000 0.000000 1.000000 1.000000 1.0

25% 30.000000 0.000000 2.000000 2.000000 1.0

50% 36.000000 0.000000 7.000000 3.000000 1.0

75% 43.000000 0.000000 14.000000 4.000000 1.0

max 60.000000 1.000000 29.000000 5.000000 1.0

EmployeeID JobLevel MonthlyIncome NumCompaniesWorked \

count 4410.000000 4410.000000 4410.000000 4410.000000

mean 2205.500000 2.063946 65029.312925 2.683220

std 1273.201673 1.106689 47068.888559 2.499737

min 1.000000 1.000000 10090.000000 0.000000

25% 1103.250000 1.000000 29110.000000 1.000000

50% 2205.500000 2.000000 49190.000000 2.000000

75% 3307.750000 3.000000 83800.000000 4.000000

max 4410.000000 5.000000 199990.000000 9.000000

PercentSalaryHike StandardHours StockOptionLevel TotalWorkingYears \

count 4410.000000 4410.0 4410.000000 4410.000000

mean 15.209524 8.0 0.793878 11.256916

std 3.659108 0.0 0.851883 7.790928

min 11.000000 8.0 0.000000 0.000000

25% 12.000000 8.0 0.000000 6.000000

50% 14.000000 8.0 1.000000 10.000000

75% 18.000000 8.0 1.000000 15.000000

max 25.000000 8.0 3.000000 40.000000

TrainingTimesLastYear YearsAtCompany YearsSinceLastPromotion \

count 4410.000000 4410.000000 4410.000000

mean 2.799320 7.008163 2.187755

std 1.288978 6.125135 3.221699

min 0.000000 0.000000 0.000000

25% 2.000000 3.000000 0.000000

50% 3.000000 5.000000 1.000000

75% 3.000000 9.000000 3.000000

max 6.000000 40.000000 15.000000

YearsWithCurrMa0ger

count 4410.000000

mean 4.123129

std 3.567327

min 0.000000

25% 2.000000

50% 3.000000

75% 7.000000

max 17.000000

const Age DistanceFromHome Education JobLevel MonthlyIncome \

0 1.0 51 6 2 1 131160

1 1.0 31 10 1 1 41890

2 1.0 32 17 4 4 193280

3 1.0 38 2 5 3 83210

4 1.0 32 10 1 1 23420

... ... ... ... ... ... ...

4405 1.0 42 5 4 1 60290

4406 1.0 29 2 4 1 26790

4407 1.0 25 25 2 2 37020

4408 1.0 42 18 2 1 23980

4409 1.0 40 28 3 2 54680

NumCompaniesWorked PercentSalaryHike StockOptionLevel \

0 1 11 0

1 0 23 1

2 1 15 3

3 3 11 3

4 4 12 2

... ... ... ...

4405 3 17 1

4406 2 15 0

4407 0 20 0

4408 0 14 1

4409 0 12 0

TotalWorkingYears TrainingTimesLastYear YearsAtCompany \

0 1 6 1

1 6 3 5

2 5 2 5

3 13 5 8

4 9 2 6

... ... ... ...

4405 10 5 3

4406 10 2 3

4407 5 4 4

4408 10 2 9

4409 0 6 21

YearsSinceLastPromotion YearsWithCurrMa0ger

0 0 0

1 1 4

2 0 3

3 7 5

4 0 4

... ... ...

4405 0 2

4406 0 2

4407 1 2

4408 7 8

4409 3 9

[4410 rows x 14 columns]

Optimization terminated successfully.

Current function value: 0.406908

Iterations 7

Logistic ourput:

|  |  |  |  |
| --- | --- | --- | --- |
| Dep. Variable:AttritionNo. | Personal Loan | **No. Observations:** | 5000 |
| **Model:** | Logit | **Df Residuals:** | 4991 |
| **Method:** | MLE | **Df Model:** | 8 |
| **Date:** | Thu, 06 Aug 2020 | **Pseudo R-squ.:** | 0.48 |
| **Time:** | 10:45:07 | **Log-Likelihood:** | -822.09 |
| **converged:** | TRUE | **LL-Null:** | -1581 |
| **Covariance Type:** | nonrobust | **LLR p-value:** | 0 |

Out[1]:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| coefstd errzP> | **coef** | **std err** | **z** | **P>|z|** | **[0.025** | **0.975]** |
| **const** | -14.1092 | 1.478 | -9.543 | 0 | -17.007 | -11.211 |
| **Age** | 0.1821 | 0.055 | 3.33 | 0.001 | 0.075 | 0.289 |
| **Experience** | -0.1749 | 0.055 | -3.202 | 0.001 | -0.282 | -0.068 |
| **Income** | 0.0434 | 0.002 | 21.527 | 0 | 0.039 | 0.047 |
| **Family** | 0.8425 | 0.069 | 12.133 | 0 | 0.706 | 0.979 |
| **CCAvg** | 0.0632 | 0.033 | 1.913 | 0.056 | -0.002 | 0.128 |
| **Mortgage** | 0.0001 | 0 | 0.223 | 0.823 | -0.001 | 0.001 |
| **Securities Account** | -0.5795 | 0.243 | -2.386 | 0.017 | -1.056 | -0.103 |
| **CD Account** | 2.5939 | 0.225 | 11.511 | 0 | 2.152 | 3.036 |

This code and this output show the classification of the attrition project. And it shows the logistic regression of the project. It shows the coefficient value, p, z, minimum and the maximum value of the dataset which is entered. And this model is only to find the logistic regression and not for the determination of the unseen data or to predict the future data.

BANK PERSONAL LOAN LOGISTIC REGRESSION:

Codes :

import pandas as pd

df1 = pd.read\_excel(r"C:\letsupgrage assignment\Bank\_Personal\_Loan\_Modelling.xlsx", sheet\_name=1)

print(df1.head())

print(df1.describe())

df1.drop(["ID", "ZIP Code", "Education", "Online", "CreditCard"], axis=1, inplace = True)

y = df1["Personal Loan"]

x = df1[["Age", "Experience", "Income", "Family", "CCAvg", "Mortgage", "Securities Account", "CD Account"]]

import statsmodels.api as sm

x1 = sm.add\_constant(x)

print(x1)

logistic = sm.Logit(y, x1)

result = logistic.fit()

result.summary()

The output:

ID Age Experience Income ZIP Code Family CCAvg Education Mortgage \

0 1 25 1 49 91107 4 1.6 1 0

1 2 45 19 34 90089 3 1.5 1 0

2 3 39 15 11 94720 1 1.0 1 0

3 4 35 9 100 94112 1 2.7 2 0

4 5 35 8 45 91330 4 1.0 2 0

Personal Loan Securities Account CD Account Online CreditCard

0 0 1 0 0 0

1 0 1 0 0 0

2 0 0 0 0 0

3 0 0 0 0 0

4 0 0 0 0 1

ID Age Experience Income ZIP Code \

count 5000.000000 5000.000000 5000.000000 5000.000000 5000.000000

mean 2500.500000 45.338400 20.104600 73.774200 93152.503000

std 1443.520003 11.463166 11.467954 46.033729 2121.852197

min 1.000000 23.000000 -3.000000 8.000000 9307.000000

25% 1250.750000 35.000000 10.000000 39.000000 91911.000000

50% 2500.500000 45.000000 20.000000 64.000000 93437.000000

75% 3750.250000 55.000000 30.000000 98.000000 94608.000000

max 5000.000000 67.000000 43.000000 224.000000 96651.000000

Family CCAvg Education Mortgage Personal Loan \

count 5000.000000 5000.000000 5000.000000 5000.000000 5000.000000

mean 2.396400 1.937913 1.881000 56.498800 0.096000

std 1.147663 1.747666 0.839869 101.713802 0.294621

min 1.000000 0.000000 1.000000 0.000000 0.000000

25% 1.000000 0.700000 1.000000 0.000000 0.000000

50% 2.000000 1.500000 2.000000 0.000000 0.000000

75% 3.000000 2.500000 3.000000 101.000000 0.000000

max 4.000000 10.000000 3.000000 635.000000 1.000000

Securities Account CD Account Online CreditCard

count 5000.000000 5000.00000 5000.000000 5000.000000

mean 0.104400 0.06040 0.596800 0.294000

std 0.305809 0.23825 0.490589 0.455637

min 0.000000 0.00000 0.000000 0.000000

25% 0.000000 0.00000 0.000000 0.000000

50% 0.000000 0.00000 1.000000 0.000000

75% 0.000000 0.00000 1.000000 1.000000

max 1.000000 1.00000 1.000000 1.000000

const Age Experience Income Family CCAvg Mortgage \

0 1.0 25 1 49 4 1.6 0

1 1.0 45 19 34 3 1.5 0

2 1.0 39 15 11 1 1.0 0

3 1.0 35 9 100 1 2.7 0

4 1.0 35 8 45 4 1.0 0

... ... ... ... ... ... ... ...

4995 1.0 29 3 40 1 1.9 0

4996 1.0 30 4 15 4 0.4 85

4997 1.0 63 39 24 2 0.3 0

4998 1.0 65 40 49 3 0.5 0

4999 1.0 28 4 83 3 0.8 0

Securities Account CD Account

0 1 0

1 1 0

2 0 0

3 0 0

4 0 0

... ... ...

4995 0 0

4996 0 0

4997 0 0

4998 0 0

4999 0 0

[5000 rows x 9 columns]

Optimization terminated successfully.

Current function value: 0.164417

Iterations 9

|  |  |  |  |
| --- | --- | --- | --- |
| **Dep. Variable:** | Personal Loan | **No. Observations:** | 5000 |
| **Model:** | Logit | **Df Residuals:** | 4991 |
| **Method:** | MLE | **Df Model:** | 8 |
| **Date:** | Thu, 06 Aug 2020 | **Pseudo R-squ.:** | 0.48 |
| **Time:** | 10:45:07 | **Log-Likelihood:** | -822.09 |
| **converged:** | TRUE | **LL-Null:** | -1581 |
| **Covariance Type:** | nonrobust | **LLR p-value:** | 0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **coef** | **std err** | **z** | **P>|z|** | **[0.025** | **0.975]** |
| **const** | -14.1092 | 1.478 | -9.543 | 0 | -17.007 | -11.211 |
| **Age** | 0.1821 | 0.055 | 3.33 | 0.001 | 0.075 | 0.289 |
| **Experience** | -0.1749 | 0.055 | -3.202 | 0.001 | -0.282 | -0.068 |
| **Income** | 0.0434 | 0.002 | 21.527 | 0 | 0.039 | 0.047 |
| **Family** | 0.8425 | 0.069 | 12.133 | 0 | 0.706 | 0.979 |
| **CCAvg** | 0.0632 | 0.033 | 1.913 | 0.056 | -0.002 | 0.128 |
| **Mortgage** | 0.0001 | 0 | 0.223 | 0.823 | -0.001 | 0.001 |
| **Securities Account** | -0.5795 | 0.243 | -2.386 | 0.017 | -1.056 | -0.103 |
| **CD Account** | 2.5939 | 0.225 | 11.511 | 0 | 2.152 | 3.036 |