**CLOUD APPLICATION DEVELOPMENT**

**GROUP 3**

**PHASE 3 SUBMISSION: ML MODELS WITH IBM WATSON**

**CODE IMPLEMENTATION:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

#LOADING DATASET

dataset = pd.read\_csv('/content/Churn\_Modelling.csv')

print(dataset.head())

dataset.info()

dataset.describe()

dataset= dataset.drop(columns = ['RowNumber','CustomerId','Surname'])

dataset.info()

dataset['Gender'].unique()

dataset= pd.get\_dummies(data=dataset,drop\_first=True)

dataset

#VISUALIZATION

dataset.Exited.plot.hist()

(dataset.Exited==1).sum()

dataset\_2=dataset.drop(columns='Exited')

dataset\_2.corrwith(dataset['Exited']).plot.bar(figsize=(16,9), title='Correlated with Exited Column', rot = 45,grid = True)

corr=dataset.corr()

#VISUALIZATION

plt.figure(figsize=(16,9))

sns.heatmap(corr,annot=True)

X= dataset.drop(columns='Exited')

y= dataset['Exited']

#MODEL TRAINING

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)

X\_test.shape

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X\_train= scaler.fit\_transform(X\_train)

X\_test= scaler.transform(X\_test)

X\_train

from sklearn.linear\_model import LogisticRegression

clf = LogisticRegression(random\_state=0).fit(X\_train, y\_train)

y\_pred= clf.predict(X\_test)

from sklearn.metrics import accuracy\_score, confusion\_matrix, f1\_score, precision\_score, recall\_score

acc=accuracy\_score(y\_test,y\_pred)

f1=f1\_score(y\_test,y\_pred)

prec=precision\_score(y\_test,y\_pred)

rec=recall\_score(y\_test,y\_pred)

results=pd.DataFrame([['Logistic regression',acc,f1,prec,rec]],columns=['Model','Accuracy','F1','Precision','Recall'])

results

print(confusion\_matrix(y\_test,y\_pred))

from sklearn.ensemble import RandomForestClassifier

clf = RandomForestClassifier(random\_state=0).fit(X\_train, y\_train)

y\_pred= clf.predict(X\_test)

acc=accuracy\_score(y\_test,y\_pred)

f1=f1\_score(y\_test,y\_pred)

prec=precision\_score(y\_test,y\_pred)

rec=recall\_score(y\_test,y\_pred)

RF\_results=pd.DataFrame([['Random Forest Classifier',acc,f1,prec,rec]],columns=['Model','Accuracy','F1','Precision','Recall'])

results.append(RF\_results,ignore\_index=True)

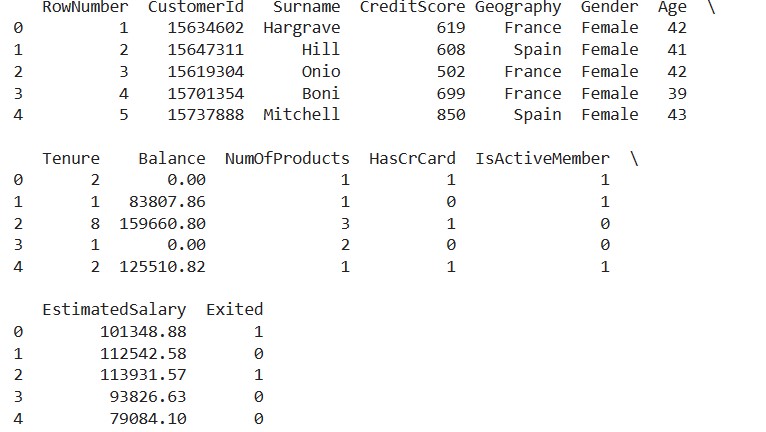
print(confusion\_matrix(y\_test,y\_pred))

dataset.head()

single\_obs=[[647,40,3,85000.45,2,0,0,92012.45,0,1,1]]

clf.predict(scaler.fit\_transform(single\_obs))

**OUTPUT:**

****

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A graph with a number of different sizes

Description automatically generated with medium confidence

A colorful chart with black and white squares

Description automatically generated with medium confidence