```
import pandas as pd
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.linear_model import LogisticRegression
```

```
dataset = pd.read_excel('/content/a_Dataset_CreditScoring.xlsx')
```

```
# shows count of rows and columns
dataset.shape
```

```
(3000, 30)
```

```
#shows first few rows of the code
dataset.head()
```

	TARGET	ID	DerogCnt	CollectCnt	BanruptcyInd	InqCnt06	InqTimeLast	InqFinanc
0	1	582	3	3	0	4	0.0	
1	1	662	15	9	0	3	1.0	
2	1	805	0	0	0	1	5.0	
3	1	1175	8	5	0	6	1.0	
4	1	1373	3	1	0	9	0.0	

5 rows × 30 columns

```
#dropping customer ID column from the dataset
dataset=dataset.drop('ID',axis=1)
dataset.shape
```

```
(3000, 29)
```

```
# explore missing values
dataset.isna().sum()
```

```
TARGET              0
DerogCnt            0
CollectCnt          0
BanruptcyInd        0
InqCnt06            0
InqTimeLast       188
InqFinanceCnt24     0
TLTimeFirst         0
TLTimeLast          0
TLCnt03             0
TLCnt12             0
```

```
TLCnt24            0
TLCnt              3
TLSum             40
TLMaxSum          40
TLSatCnt           4
TLDel60Cnt         0
TLBadCnt24         0
TL75UtilCnt       99
TL50UtilCnt       99
TLBalHCPct        41
TLSatPct           4
TLDel3060Cnt24     0
TLDel90Cnt24       0
TLDel60CntAll      0
TLOpenPct          3
TLBadDerogCnt      0
TLDel60Cnt24       0
TLOpen24Pct        3
dtype: int64
```

```
# filling missing values with mean
dataset=dataset.fillna(dataset.mean())
```

```
# explore missing values post missing value fix
dataset.isna().sum()
```

```
TARGET            0
DerogCnt          0
CollectCnt        0
BanruptcyInd      0
InqCnt06          0
InqTimeLast       0
InqFinanceCnt24   0
TLTimeFirst       0
TLTimeLast        0
TLCnt03           0
TLCnt12           0
TLCnt24           0
TLCnt             0
TLSum             0
TLMaxSum          0
TLSatCnt          0
TLDel60Cnt        0
TLBadCnt24        0
TL75UtilCnt       0
TL50UtilCnt       0
TLBalHCPct        0
TLSatPct          0
TLDel3060Cnt24    0
TLDel90Cnt24      0
TLDel60CntAll     0
TLOpenPct         0
TLBadDerogCnt     0
TLDel60Cnt24      0
TLOpen24Pct       0
dtype: int64
```

```
y = dataset.iloc[:, 0].values
X = dataset.iloc[:, 1:29].values


# splitting dataset into training and test (in ratio 80:20)

X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.2,
                                                    random_state=0,
                                                    stratify=y)


sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)


classifier = LogisticRegression()
classifier.fit(X_train,y_train)
y_pred = classifier.predict(X_test)


print(confusion_matrix(y_test,y_pred))
```

```
[[487  13]
 [ 87  13]]
```

```
print(accuracy_score(y_test,y_pred))
```

```
0.8333333333333334
```

```
prediction = classifier.predict_proba(X_test)
prediction
```

```
array([[0.61644691, 0.38355309],
       [0.9885656 , 0.0114344 ],
       [0.87069686, 0.12930314],
       ...,
       [0.94450568, 0.05549432],
       [0.46756903, 0.53243097],
       [0.94014209, 0.05985791]])
```

```
#writing model output file

df_prediction_prob = pd.DataFrame(prediction,columns=['prob_0','prob_1'])
df_prediction_target =pd.DataFrame(classifier.predict(X_test),columns=['prediction_TARGET
df_test_dataset = pd.DataFrame(y_test,columns= ['Actual outcume'])

dfx=pd.concat([df_test_dataset, df_prediction_prob, df_prediction_target], axis=1)

dfx.to_csv("/content/a_Dataset_CreditScoring.xlsx",sep =',',encoding='UTF-8')

dfx.head()
```

	Actual outcume	prob_0	prob_1	prediction_TARGET
0	1	0.616447	0.383553	0
1	0	0.988566	0.011434	0
2	1	0.870697	0.129303	0
3	0	0.953963	0.046037	0
4	1	0.726633	0.273367	0