

Importing Required Libraries

```
In [125... import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Importing Dataset

```
In [126... data1 = pd.read_csv('customer_data.csv')
```

```
In [127... data1.shape
```

```
Out[127]: (1125, 13)
```

```
In [128... data1.head()
```

```
Out[128]:
```

	label	id	fea_1	fea_2	fea_3	fea_4	fea_5	fea_6	fea_7	fea_8	fea_9	fea_10	
0	1	54982665	5	1245.5	3	77000.0	2	15	5	109	5	151300	244
1	0	59004779	4	1277.0	1	113000.0	2	8	-1	100	3	341759	207
2	0	58990862	7	1298.0	1	110000.0	2	11	-1	101	5	72001	1
3	1	58995168	7	1335.5	1	151000.0	2	11	5	110	3	60084	1
4	0	54987320	7	NaN	2	59000.0	2	11	5	108	4	450081	197

```
In [129... data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1125 entries, 0 to 1124
Data columns (total 13 columns):
#   Column      Non-Null Count  Dtype
---  -
0   label      1125 non-null   int64
1   id          1125 non-null   int64
2   fea_1       1125 non-null   int64
3   fea_2       976 non-null    float64
4   fea_3       1125 non-null   int64
5   fea_4       1125 non-null   float64
6   fea_5       1125 non-null   int64
7   fea_6       1125 non-null   int64
8   fea_7       1125 non-null   int64
9   fea_8       1125 non-null   int64
10  fea_9       1125 non-null   int64
11  fea_10      1125 non-null   int64
12  fea_11      1125 non-null   float64
dtypes: float64(3), int64(10)
memory usage: 114.4 KB
```

```
In [130... data1.isnull().sum()
```

```
Out[130]: label      0
          id        0
          fea_1     0
          fea_2    149
          fea_3     0
          fea_4     0
          fea_5     0
          fea_6     0
          fea_7     0
          fea_8     0
          fea_9     0
          fea_10    0
          fea_11    0
          dtype: int64
```

Filling the missing values with mean

```
In [131... data1.fillna(data1.fea_2.mean(), inplace=True)
```

```
In [132... data1.isnull().sum()
```

```
Out[132]: label      0
          id        0
          fea_1     0
          fea_2     0
          fea_3     0
          fea_4     0
          fea_5     0
          fea_6     0
          fea_7     0
          fea_8     0
          fea_9     0
          fea_10    0
          fea_11    0
          dtype: int64
```

```
In [133... data1.isin(['?']).sum()
```

```
Out[133]: label      0
          id        0
          fea_1     0
          fea_2     0
          fea_3     0
          fea_4     0
          fea_5     0
          fea_6     0
          fea_7     0
          fea_8     0
          fea_9     0
          fea_10    0
          fea_11    0
          dtype: int64
```

```
In [134... data1.corr()
fig,ax=plt.subplots(figsize=(10,8))
sns.heatmap(data1.corr(),annot=True)
```

```
Out[134]: <Axes: >
```



Splitting

```
In [135... y = data1['label'].values
x = data1.drop('label', axis=1).values
```

```
In [136... 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_s
```

Logistic Regressions

```
In [137... x_train.shape, y_test.shape
```

```
Out[137]: ((900, 12), (225,))
```

```
In [138... from sklearn.linear_model import LogisticRegression
```

```
logreg = LogisticRegression()
logreg.fit(x_train,y_train)
```

```
Out[138]: ▼ LogisticRegression
LogisticRegression()
```

```
In [139... pred_test = logreg.predict(x_test)
```

```
In [140... pred_train = logreg.predict(x_train)
```

```
In [141... from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
print("Train:", accuracy_score(y_train, pred_train))
print("Test:", accuracy_score(y_test, pred_test))
```

Train: 0.8088888888888889
Test: 0.7644444444444445

```
In [142... print(classification_report(y_test, pred_test))
```

	precision	recall	f1-score	support
0	0.76	1.00	0.87	172
1	0.00	0.00	0.00	53
accuracy			0.76	225
macro avg	0.38	0.50	0.43	225
weighted avg	0.58	0.76	0.66	225

/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

_warn_prf(average, modifier, msg_start, len(result))

/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

_warn_prf(average, modifier, msg_start, len(result))

/usr/local/lib/python3.9/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.

_warn_prf(average, modifier, msg_start, len(result))

Checking Imbalanced Problem

```
In [143... data1.label.value_counts()
```

```
Out[143]: 0    900
          1    225
          Name: label, dtype: int64
```

There is imbalanced problem.

```
In [144... from imblearn.over_sampling import SMOTE
```

```
In [145... oversample = SMOTE()
```

```
In [146... X, Y = oversample.fit_resample(x_train, y_train)
```

```
In [147... counter = Counter(Y)
print(counter)
```

Counter({0: 728, 1: 728})

```
In [148... X.shape
```

```
Out[148]: (1456, 12)
```

```
In [149... Y.shape
```

Out[149]: (1456,)

```
In [150... logreg2 = LogisticRegression()
logreg2.fit(X,Y)
```

Out[150]: **LogisticRegression**
LogisticRegression()

```
In [151... pred_test2 = logreg2.predict(x_test)
```

```
In [152... pred_test2.shape
```

Out[152]: (225,)

```
In [153... y_test.shape
```

Out[153]: (225,)

```
In [154... pred_train2 = logreg2.predict(x_train)
```

```
In [155... pred_train2.shape
```

Out[155]: (900,)

```
In [158... from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
print("Test:", accuracy_score(y_test, pred_test2))
print("Train:", accuracy_score(y_train, pred_train2))
```

Test: 0.4888888888888889
Train: 0.5233333333333333

```
In [159... print(classification_report(y_test, pred_test2))
```

	precision	recall	f1-score	support
0	0.82	0.42	0.56	172
1	0.27	0.70	0.39	53
accuracy			0.49	225
macro avg	0.55	0.56	0.48	225
weighted avg	0.69	0.49	0.52	225