APPLIED DATA SCIENCE(LAB)

ASSIGNMENT NO : 1

TITLE : IMBALANCE HANDLING

NAME: Sandip Dattatray Jadhav

ROLL NO : 82

CLASS : ECE

GITHUB LINK :

https://github.com/sandipjadhav87/Applied_Data_Science/upload/main/Assignment_LAB

DATASET : Credit Card Fraud Dataset

CODE:

```
# ======================================
# CREDIT CARD FRAUD DETECTION - IMBALANCE HANDLING
# Dataset: credit_card_fraud_dataset.csv
# ======================================


# ======================================
# ADVANCED CREDIT CARD FRAUD ANALYSIS
# (Different Visualization Style)
# ======================================

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import (classification_report, confusion_matrix,
                             accuracy_score, roc_curve, roc_auc_score,
                             precision_recall_curve)
```

```
from sklearn.preprocessing import LabelEncoder
from imblearn.over_sampling import SMOTE

# Modern theme
sns.set_style("darkgrid")
plt.rcParams["figure.figsize"] = (8,5)

# ----------------------------
# 1. Load Dataset
# ----------------------------
data = pd.read_csv("credit_card_fraud_dataset.csv")

if "TransactionDate" in data.columns:
    data.drop("TransactionDate", axis=1, inplace=True)

le = LabelEncoder()

if "TransactionType" in data.columns:
    data["TransactionType"] = le.fit_transform(data["TransactionType"])

if "Location" in data.columns:
    data["Location"] = le.fit_transform(data["Location"])

# ----------------------------
# 2. Class Distribution (Horizontal Bar)
# ----------------------------
plt.figure()
data["IsFraud"].value_counts().plot(kind="barh", color=["teal","crimson"])
plt.title("Transaction Class Distribution")
plt.xlabel("Count")
plt.ylabel("Class (0 = Legit, 1 = Fraud)")
```

```
# ----------------------------
# 3. Split Data
# ----------------------------
X = data.drop("IsFraud", axis=1)
y = data["IsFraud"]

X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.3,
    random_state=42,
    stratify=y
)

# ----------------------------
# 4. Before SMOTE Model
# ----------------------------
model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
y_prob = model.predict_proba(X_test)[:,1]

print("BEFORE SMOTE")
print(classification_report(y_test, y_pred))

# Confusion Matrix (Styled Differently)
plt.figure()
sns.heatmap(confusion_matrix(y_test, y_pred),
            annot=True, cmap="magma", fmt="d")
plt.title("Confusion Matrix (Imbalanced Data)")

# ----------------------------
```

```
# 5. Apply SMOTE
# ----------------------------
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X_train, y_train)

# Balanced Distribution (Donut Chart)
plt.figure()
counts = pd.Series(y_resampled).value_counts()
plt.pie(counts,
        labels=["Legitimate","Fraud"],
        autopct="%1.1f%%",
        wedgeprops=dict(width=0.4))
plt.title("Balanced Dataset After SMOTE")

# ----------------------------
# 6. After SMOTE Model
# ----------------------------
model_smote = LogisticRegression(max_iter=1000)
model_smote.fit(X_resampled, y_resampled)
y_pred_smote = model_smote.predict(X_test)
y_prob_smote = model_smote.predict_proba(X_test)[:,1]

print("AFTER SMOTE")
print(classification_report(y_test, y_pred_smote))

# Confusion Matrix (Different Theme)
plt.figure()
sns.heatmap(confusion_matrix(y_test, y_pred_smote),
            annot=True, cmap="viridis", fmt="d")
plt.title("Confusion Matrix (Balanced Data)")

# ----------------------------
```

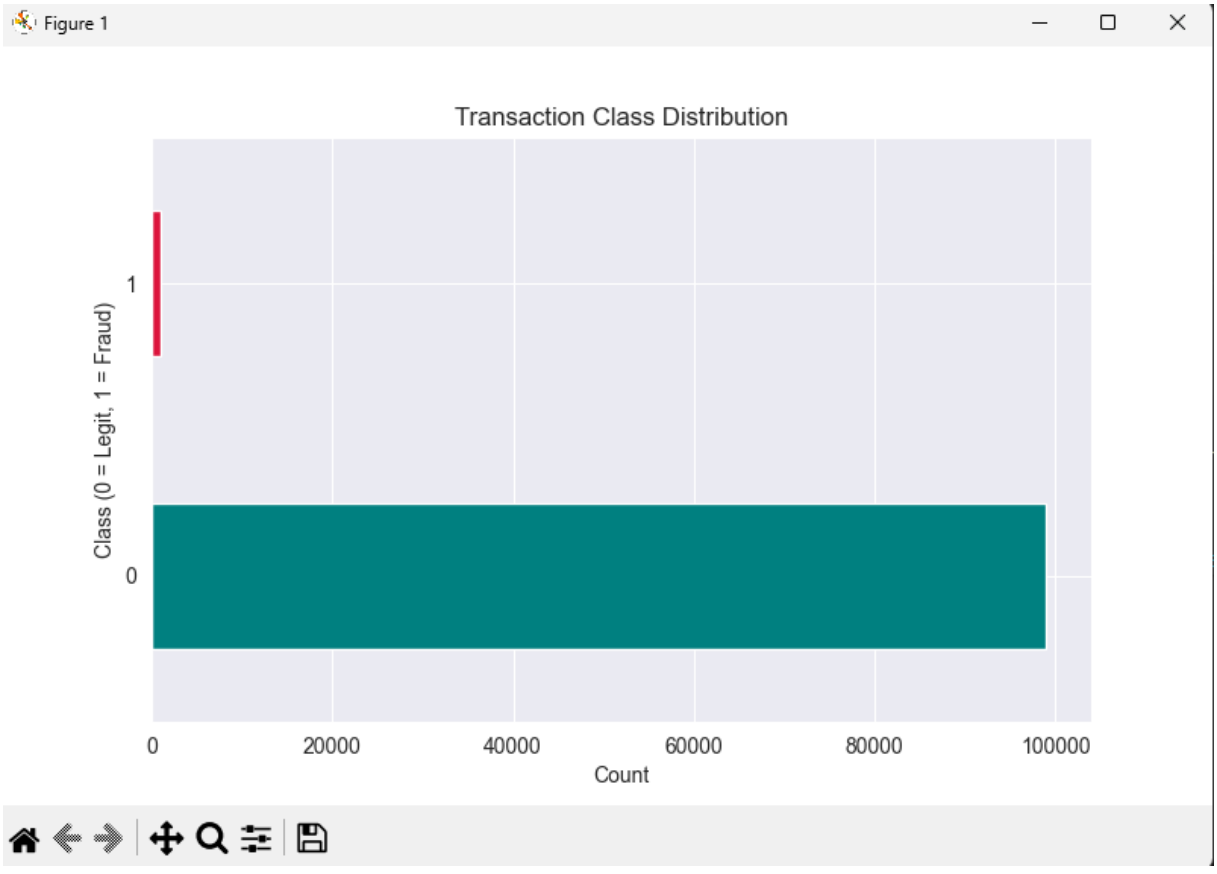
```
# 7. ROC Curve
# ----------------------------
fpr, tpr, _ = roc_curve(y_test, y_prob_smote)
roc_auc = roc_auc_score(y_test, y_prob_smote)

plt.figure()
plt.plot(fpr, tpr)
plt.plot([0,1],[0,1])
plt.title(f"ROC Curve (AUC = {roc_auc:.2f})")
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")

# ----------------------------
# 8. Precision-Recall Curve
# ----------------------------
precision, recall, _ = precision_recall_curve(y_test, y_prob_smote)

plt.figure()
plt.plot(recall, precision)
plt.title("Precision-Recall Curve")
plt.xlabel("Recall")
plt.ylabel("Precision")

# ----------------------------
# 9. Fraud Probability Distribution
# ----------------------------
plt.figure()
sns.histplot(y_prob_smote, bins=30, kde=True)
plt.title("Fraud Probability Distribution")

# ----------------------------
# 10. Feature Importance (Styled)
```

```
# ----------------------------
importance = model_smote.coef_[0]
feat_df = pd.DataFrame({
    "Feature": X.columns,
    "Importance": importance
}).sort_values(by="Importance", ascending=False)

plt.figure()
sns.barplot(x="Importance", y="Feature", data=feat_df)
plt.title("Feature Importance Ranking")

plt.show()
```

OUTPUT:

```
C:\Users\thete\OneDrive\Desktop\ADSL(PE)\ADSL LAB ASSIGN\ASSIGN 1>python assign1.py
Dataset Shape: (100000, 7)
   TransactionID           TransactionDate   Amount  MerchantID TransactionType      Location  IsFraud
0              1  2024-04-03 14:15:35.462794  4189.27         688          refund   San Antonio        0
1              2  2024-03-19 13:20:35.462824  2659.71         109          refund        Dallas        0
2              3  2024-01-08 10:08:35.462834   784.00         394        purchase      New York        0
3              4  2024-04-13 23:50:35.462850  3514.40         944        purchase  Philadelphia        0
4              5  2024-07-12 18:51:35.462858   369.07         475        purchase       Phoenix        0

Fraud Class Distribution:
IsFraud
0    99000
1     1000
Name: count, dtype: int64

--- BEFORE SMOTE ---
[[29700     0]
 [  300     0]]
```

```
C:\Users\Dell\AppData\Roaming\Python\Python311\site-packages\sklearn\metrics\_classification.py:1833: UndefinedMetricW
arning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` paramet
er to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", result.shape[0])
              precision    recall  f1-score   support

           0       0.99      1.00      0.99     29700
           1       0.00      0.00      0.00       300

    accuracy                           0.99     30000
   macro avg       0.49      0.50      0.50     30000
weighted avg       0.98      0.99      0.99     30000

AFTER SMOTE
              precision    recall  f1-score   support

           0       0.99      0.50      0.67     29700
           1       0.01      0.55      0.02       300

    accuracy                           0.50     30000
   macro avg       0.50      0.53      0.35     30000
weighted avg       0.98      0.50      0.66     30000
```



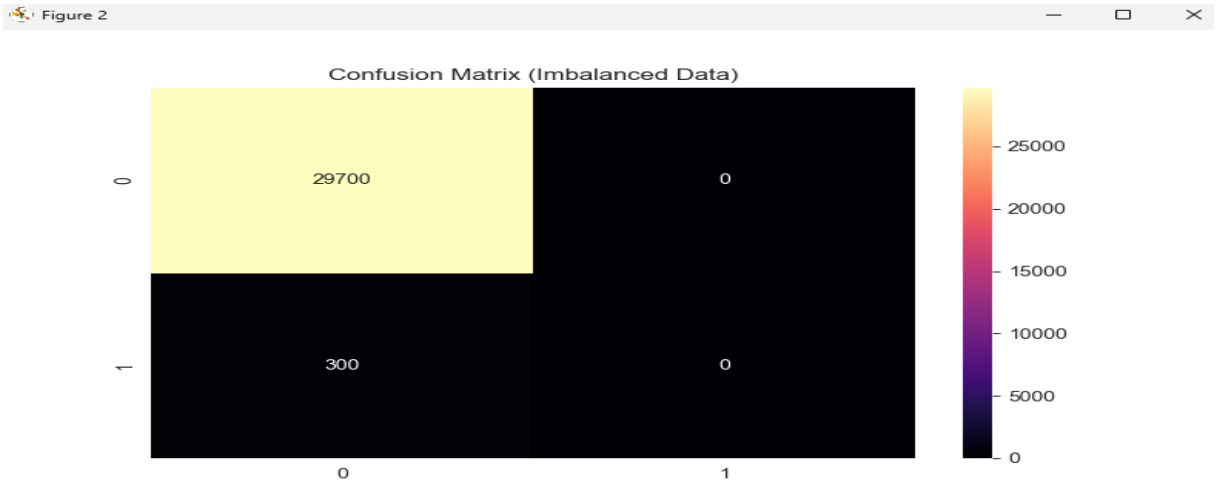


Figure 2
Confusion Matrix (Imbalanced Data)
29700
0
300
0
0
1
0
1
25000
20000
15000
10000
5000
0

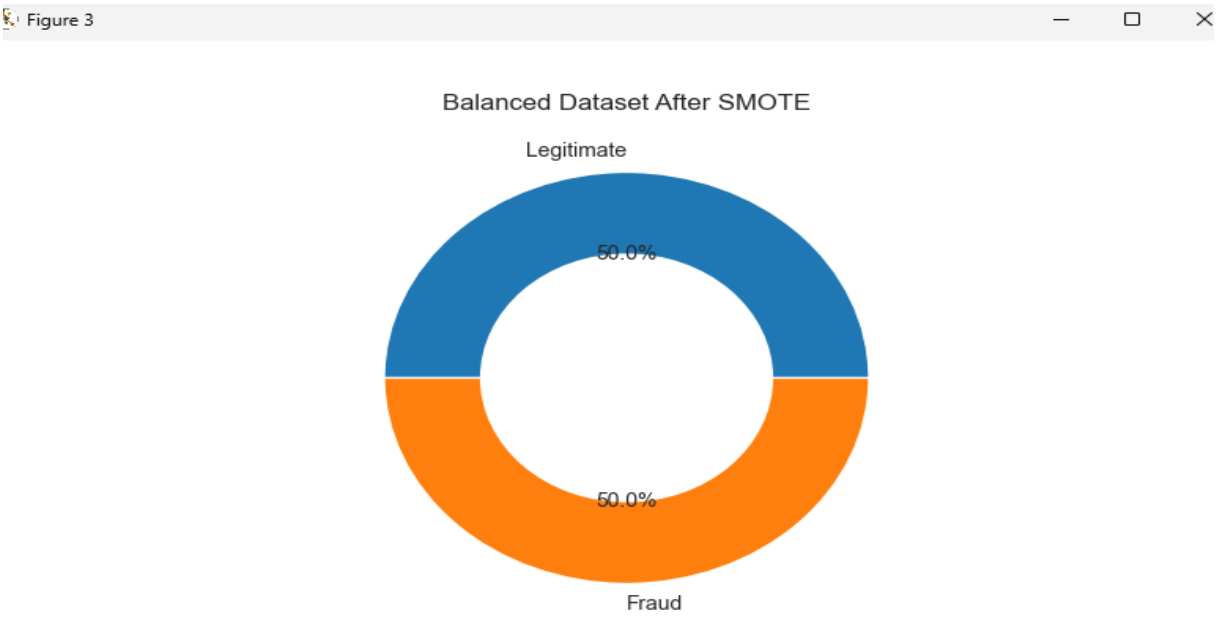


Figure 3
Balanced Dataset After SMOTE
Legitimate
50.0%
50.0%
Fraud

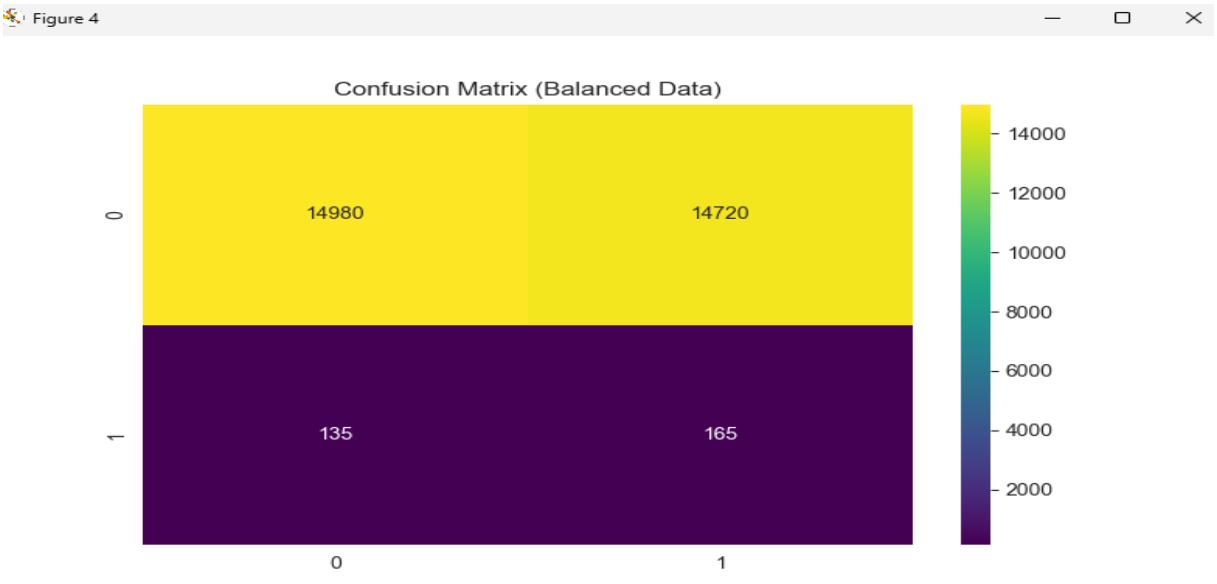


Figure 4
Confusion Matrix (Balanced Data)
14980
14720
135
165
0
1
0
1
14000
12000
10000
8000
6000
4000
2000

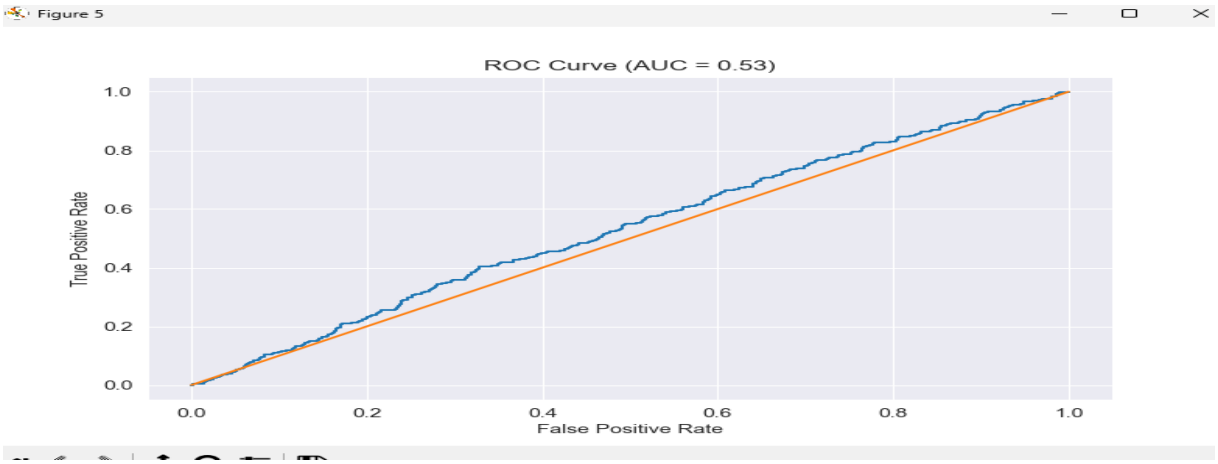
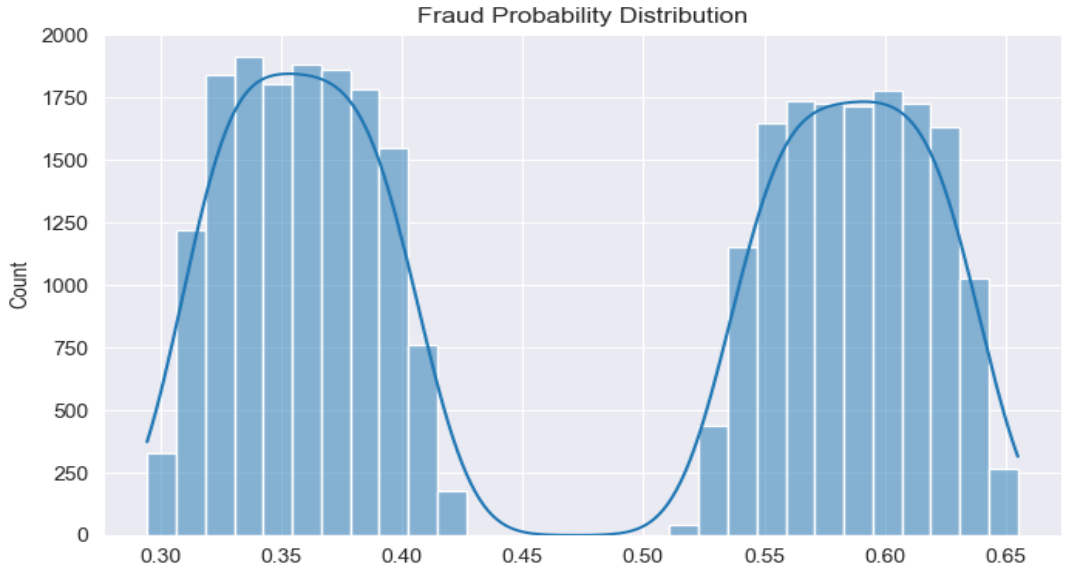
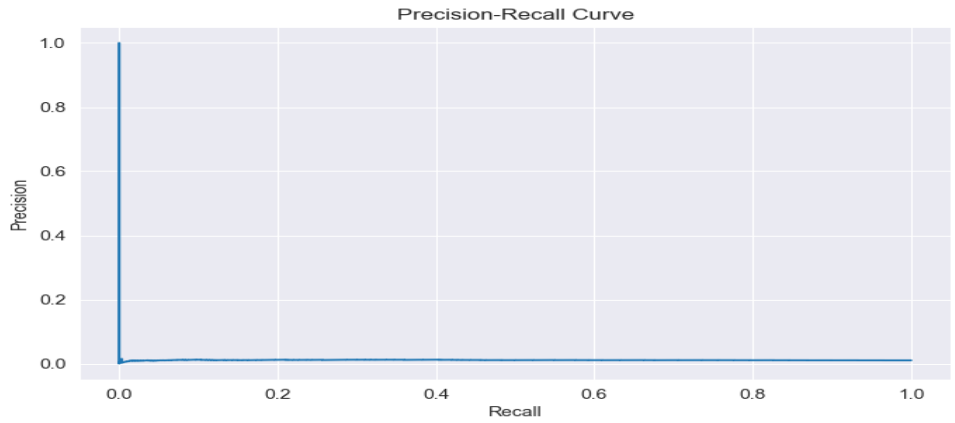
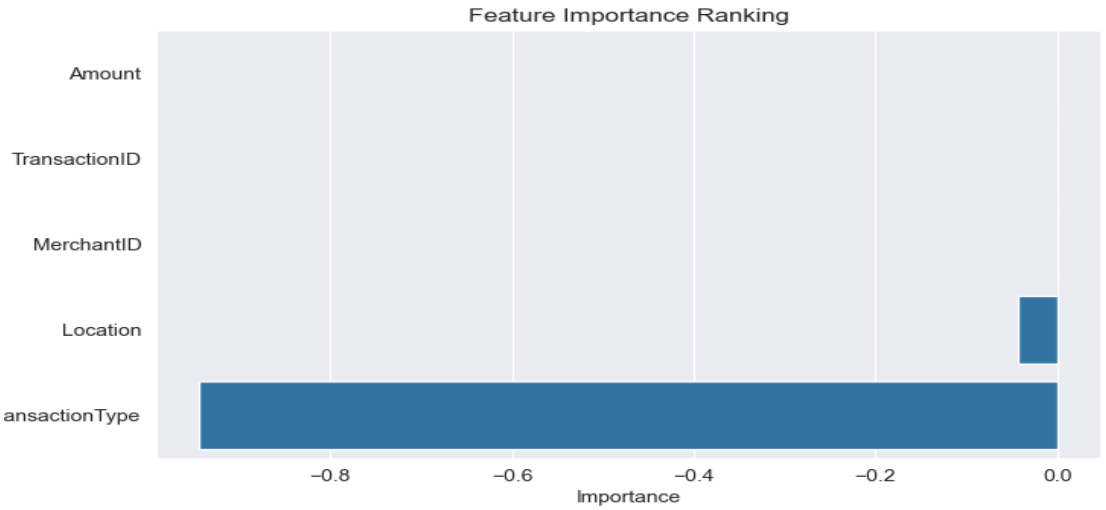


Figure 5
ROC Curve (AUC = 0.53)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0


Fraud Probability Distribution
Count
0
250
500
750
1000
1250
1500
1750
2000
0.30
0.35
0.40
0.45
0.50
0.55
0.60
0.65


Precision-Recall Curve
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0



INTERPRETATION:

Interpretation of Credit Card Fraud Detection with SMOTE and Visualizations Data Preprocessing and Encoding

The dataset was first loaded and examined for structure and data types.
The `TransactionDate` column was removed because it is non-numeric and does not directly contribute to prediction.

Categorical features such as `TransactionType` and `Location` were converted into numerical values using Label Encoding.

Interpretation:
This step prepares the dataset for machine learning since algorithms like Logistic Regression require numerical input. Removing unnecessary columns improves model efficiency and reduces noise in the data.

Class Distribution Analysis

The class distribution graph shows that legitimate transactions (Class 0) are significantly higher than fraudulent transactions (Class 1).

This confirms that the dataset is highly imbalanced.

Interpretation:

• The majority class dominates the dataset.
• The model may become biased toward predicting legitimate transactions.
• High accuracy alone is not a reliable metric in imbalanced datasets.

Model Performance Before Applying SMOTE

A Logistic Regression model was trained on the original imbalanced dataset.
The confusion matrix reveals the number of correctly and incorrectly classified transactions.

Interpretation:

• High True Negatives indicate good prediction of legitimate transactions.
• Low True Positives show poor detection of fraud cases.
• The model achieves good overall accuracy but performs poorly in identifying fraudulent transactions.

This demonstrates the impact of class imbalance on model performance.

Applying SMOTE (Synthetic Minority Oversampling Technique)

SMOTE was applied to balance the training dataset.
It generates synthetic samples of the minority class (fraud) based on nearest neighbors.

After applying SMOTE, the dataset shows nearly equal numbers of legitimate and fraudulent transactions.

Interpretation:

• The imbalance problem is reduced.
• The model gets sufficient fraud samples to learn patterns effectively.
• Bias toward the majority class decreases.

Model Performance After SMOTE

The Logistic Regression model was retrained using the balanced dataset.
The updated confusion matrix shows improved fraud detection.

Interpretation:

• Increase in correctly predicted fraud transactions (higher recall).
• The model becomes more sensitive to fraudulent behavior.
• Slight increase in false positives may occur.

In fraud detection systems, false positives are acceptable because preventing fraud is more important than mistakenly flagging a few legitimate transactions.

ROC Curve Analysis

The ROC curve measures the trade-off between True Positive Rate and False Positive Rate.
The Area Under Curve (AUC) represents the model’s ability to distinguish between classes.

Interpretation:

• A higher AUC value indicates better classification performance.
• After applying SMOTE, the ROC curve shows improved discrimination between fraud and legitimate transactions.

Precision-Recall Analysis

The Precision-Recall curve focuses on performance for the minority class.

Interpretation:

• Improved precision means fewer false fraud alerts.
• Improved recall means more fraud cases are detected.
• The balance between precision and recall determines overall fraud detection efficiency.

Feature Importance Analysis

The feature importance graph displays the influence of each feature on fraud prediction. Positive coefficients indicate a higher probability of fraud, while negative values indicate legitimate transaction patterns.

Interpretation:

• Features with higher absolute coefficient values have stronger impact on prediction.
• Understanding important features helps in improving fraud detection strategies.