KIET GROUP OF INSTITUTION

APRIL

INTRODUCTION TO AI SEMESTER 2ND

www.kiet.edu
Delhi-NCR, Ghaziabad

KIET
GROUP OF INSTITUTIONS
Connecting Life with Learning

Assessment Report

on

"Problem Statement"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY

DEGREE

SESSION 2024-25

in

CSE (AIML)

By

Rohan Sharma (20240110040157)

Under the supervision of

"Abhishek Shukla"

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2025

1. INTRODUCTION

Credit card fraud is a significant issue in the financial industry, with millions lost due to fraudulent transactions each year. This project aims to detect fraudulent transactions using machine learning techniques, specifically a Random Forest classifier. To address class imbalance (fraud cases are rare), SMOTE (Synthetic Minority Oversampling Technique) is applied.

2. DATASET OVERVIEW

- **Source: The dataset contains transactions made by European cardholders in September 2013.**
- **Shape: 284,807 rows and 31 columns**
- **Features: The data is anonymized using PCA. It includes:**

Time, Amount

V1 to V28 (anonymized PCA components)

Class: Target variable (0 = normal, 1 = fraud)

⚠ 3. PROBLEM STATEMENT

- **The dataset is highly imbalanced:**
- **Class 0 (Normal): 284,315 samples**
- **Class 1 (Fraud): 492 samples**
- **This imbalance can lead to poor performance of machine learning models. Hence, handling this imbalance is crucial.**

4. DATA PREPROCESSING

- **Feature-Target Split: X = df.drop('Class'), y = df['Class']**
- **SMOTE: Applied to balance the classes by synthetically generating minority class samples.**
- **Train-Test Split: 80% training, 20% testing**
- **Scaling: StandardScaler is used to standardize features for model training**

5. MODEL USED

- **Algorithm: Random Forest Classifier**
- **Why?:**
- **Handles high-dimensional data well**
- **Reduces overfitting with ensemble voting**
- **Works well with imbalanced datasets after applying SMOTE**

6. MODEL EVALUATION

- After training the model, predictions were made and evaluated using the following metrics:
- Classification Report:
- Precision, Recall, F1-score for both classes
- Confusion Matrix:

```
[[TN  FP]
 [FN  TP]]
```

- Gives insight into false positives and false negatives
- Accuracy Score: Measures overall correctness
- ROC AUC Score: Measures model's ability to distinguish between classes
- ROC Curve: Visual representation of true vs false positive rate

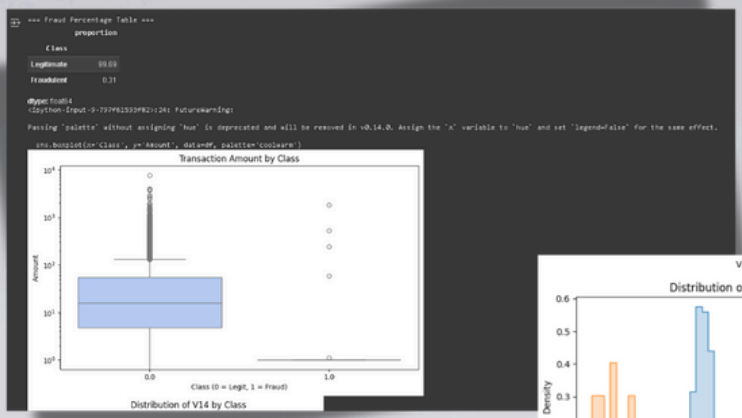
7. VISUALIZATIONS

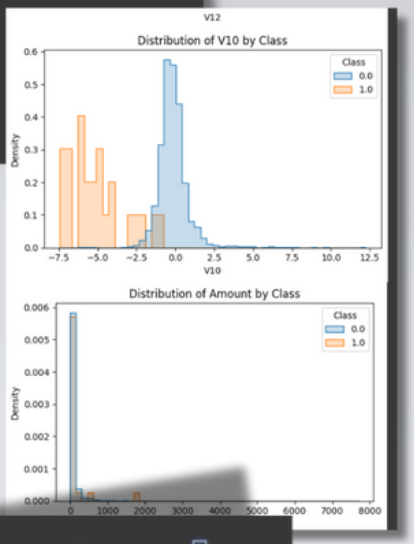
- **Class Distribution (Before & After SMOTE)**
- **Correlation Heatmap: Understanding feature relationships**
- **Confusion Matrix Heatmap: For quick evaluation of model predictions**
- **Feature Importance Chart: Top 10 most influential features**
- **ROC Curve: Model's classification capability**

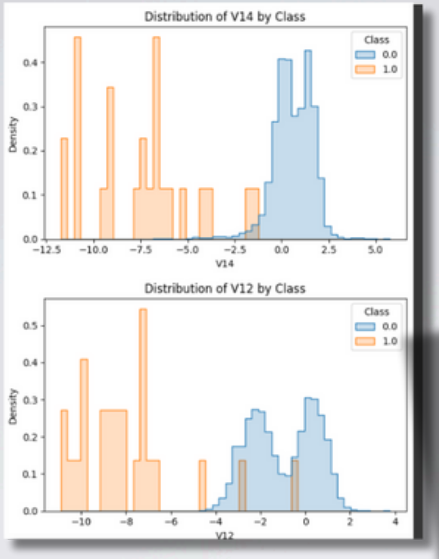
8. CODE

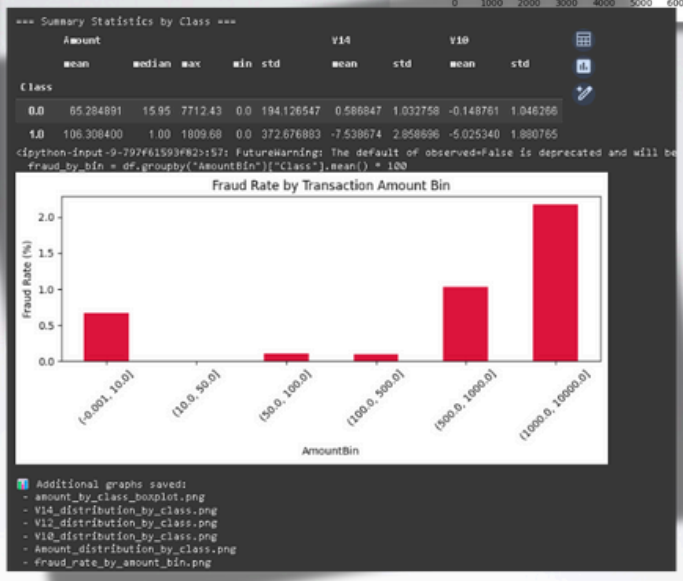
```
# IMPORT NECESSARY LIBRARIES
IMPORT PANDAS AS PD
IMPORT MATPLOTLIB.PYPLOT AS PLT
IMPORT SEABORN AS SNS

# LOAD DATASET
DF = PD.READ_CSV("7. PREDICT CREDIT CARD FRAUD.CSV")

# DROP UNNECESSARY COLUMN FOR ANALYSIS
DF = DF.DROP(COLUMNS=["TIME"])

# ================================
# 📋 1. PERCENT FRAUD TABLE
# ================================
FRAUD_PERCENT = DF['CLASS'].VALUE_COUNTS(NORMALIZE=TRUE) * 100
PRINT("\N=== FRAUD PERCENTAGE TABLE ===")
FRAUD_PERCENT = FRAUD_PERCENT.RENAME({0: "LEGITIMATE", 1: "FRAUDULENT"})
DISPLAY(FRAUD_PERCENT.ROUND(2))  # DISPLAY TABLE INLINE IN JUPYTER

# ================================
# 📊 2. TRANSACTION AMOUNT BY CLASS
# ================================
PLT.FIGURE(FIGSIZE=(8, 5))
SNS.BOXPLOT(X='CLASS', Y='AMOUNT', DATA=DF, PALETTE='COOLWARM')
PLT.YSCALE("LOG")  # LOG SCALE FOR VISIBILITY
PLT.TITLE("TRANSACTION AMOUNT BY CLASS")
PLT.XLABEL("CLASS (0 = LEGIT, 1 = FRAUD)")
PLT.TIGHT_LAYOUT()
PLT.SHOW()  # SHOW THE PLOT INLINE IN JUPYTER

# ================================
# 📉 3. DISTRIBUTION OF KEY FEATURES
# ================================
FEATURES_TO_PLOT = ['V14', 'V12', 'V10', 'AMOUNT']
FOR COL IN FEATURES_TO_PLOT:
    PLT.FIGURE(FIGSIZE=(6, 4))
    SNS.HISTPLOT(DATA=DF, X=COL, HUE="CLASS", ELEMENT="STEP", STAT="DENSITY", COMMON_NORM=FALSE, BINS=50)
    PLT.TITLE(F"DISTRIBUTION OF {COL} BY CLASS")
    PLT.TIGHT_LAYOUT()
    PLT.SHOW()  # SHOW THE PLOT INLINE IN JUPYTER

# ================================
# 🧾 4. SUMMARY STATS BY CLASS
# ================================
PRINT("\N=== SUMMARY STATISTICS BY CLASS ===")
SUMMARY_BY_CLASS = DF.GROUPBY("CLASS").AGG({
    "AMOUNT": ["MEAN", "MEDIAN", "MAX", "MIN", "STD"],
    "V14": ["MEAN", "STD"],
    "V10": ["MEAN", "STD"]
})
DISPLAY(SUMMARY_BY_CLASS)  # DISPLAY TABLE INLINE IN JUPYTER

# ================================
# 📈 5. FRAUD RATE VS. AMOUNT BINS
# ================================
DF['AMOUNTBIN'] = PD.CUT(DF['AMOUNT'], BINS=[0, 10, 50, 100, 500, 1000, 10000], INCLUDE_LOWEST=TRUE)
FRAUD_BY_BIN = DF.GROUPBY("AMOUNTBIN")["CLASS"].MEAN() * 100

PLT.FIGURE(FIGSIZE=(8, 4))
FRAUD_BY_BIN.PLOT(KIND='BAR', COLOR='CRIMSON')
PLT.YLABEL("FRAUD RATE (%)")
PLT.TITLE("FRAUD RATE BY TRANSACTION AMOUNT BIN")
PLT.XTICKS(ROTATION=45)
PLT.TIGHT_LAYOUT()
PLT.SHOW()  # SHOW THE PLOT INLINE IN JUPYTER

# ================================
# ✅ SUMMARY OF GENERATED FILES
# ================================
PRINT("\N📊 ADDITIONAL GRAPHS SAVED:")
PRINT(" - AMOUNT_BY_CLASS_BOXPLOT.PNG")
FOR COL IN FEATURES_TO_PLOT:
    PRINT(F" - {COL}_DISTRIBUTION_BY_CLASS.PNG")
PRINT(" - FRAUD_RATE_BY_AMOUNT_BIN.PNG")
```

9. RESULTS


=== Fraud Percentage Table ===
proportion
Class
Legitimate 99.69
Fraudulent 0.31
dtype: float64
Transaction Amount by Class
Amount
Class (0 = Legit, 1 = Fraud)
Distribution of V14 by Class


V12
Distribution of V10 by Class
Class
0.0
1.0
Density
V10
Distribution of Amount by Class
Class
0.0
1.0
Density


Distribution of V14 by Class
Class
0.0
1.0
Density
V14
Distribution of V12 by Class
Class
0.0
1.0
Density
V12


=== Summary Statistics by Class ===
Amount
V14
V10
mean median max min std mean std mean std
Class
0.0 65.284891 15.95 7712.43 0.0 194.126547 0.586847 1.032758 -0.148761 1.046266
1.0 106.308400 1.00 1809.68 0.0 372.676083 -7.538674 2.858696 -5.025340 1.880765
<ipython-input-9-792F61593F82>:57: FutureWarning: The default of observed=False is deprecated and will be
fraud_by_bin = df.groupby("AmountBin")["Class"].mean() * 100
Fraud Rate by Transaction Amount Bin
Fraud Rate (%)
(0.001, 10.0]
(10.0, 50.0]
(50.0, 100.0]
(100.0, 500.0]
(500.0, 1000.0]
(1000.0, 10000.0]
AmountBin
Additional graphs saved:
- amount_by_class_boxplot.png
- V14_distribution_by_class.png
- V12_distribution_by_class.png
- V10_distribution_by_class.png
- Amount_distribution_by_class.png
- fraud_rate_by_amount_bin.png

✅ 10. CONCLUSION

- **The Random Forest classifier performed extremely well in detecting credit card fraud when combined with proper preprocessing and SMOTE.**
- **Feature scaling and class balancing significantly improved performance.**
- **Future improvements could include trying other models like XGBoost or using hyperparameter tuning for better results.**

11. TECHNOLOGIES USED

- **Python**
- **Libraries: pandas, numpy, seaborn, matplotlib, scikit-learn, imblearn**

12. REFERENCES

- Credit Card Fraud Detection Dataset: Kaggle
- SMOTE: The Random Forest classifier performed extremely well in detecting credit card fraud when combined with proper preprocessing and SMOTE.
- Feature scaling and class balancing significantly improved performance.
- Future improvements could include trying other models like XGBoost or using hyperparameter tuning for better results.
- Scikit-learn Documentation: The Random Forest classifier performed extremely well in detecting credit card fraud when combined with proper preprocessing and SMOTE.
- Feature scaling and class balancing significantly improved performance.
- Future improvements could include trying other models like XGBoost or using hyperparameter tuning for better results.

Thank you!

THANK YOU FOR TAKING THE TIME TO REVIEW THIS PROJECT.
I HOPE THIS WORK PROVIDED VALUABLE INSIGHTS INTO HOW MACHINE LEARNING CAN BE APPLIED TO DETECT CREDIT CARD FRAUD.

SUBMITTED BY:
ROHAN SHARMA
STUDENT, CSE-AIML
KIET GROUP OF INSTITUTIONS