CREDIT-CARD FRAUD DETECTION

A COURSE PROJECT REPORT

By

UTKARSHA DIXIT (RA2111027010081)

BIKRAM ISHAAN (RA2111027010082)

TALAT BINTI FIRDOUS (RA2111027010115)

Under the guidance of

Mrs. D.Hemavathi

Assistant professor

Department of Data Science and Business Systems

In partial fulfilment for the Course

of

18CSE396T – DATA SCIENCE

in

Department of Data Science and Business Systems

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Kattankulathur, Chenpalpattu District

NOVEMBER 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this mini project report "**CREDIT-CARD FRAUD DETECTION**" is the bonafide work of Utkarsha Dixit (RA2111027010081) Bikram Ishaan (RA2111027010082) Talat Binti Firdous (RA2111027010115) who carried out the project work under my supervision.

Mrs. D.Hemavathi

Assistant professor
Department of Data Science and Business Systems
SRM institute of science and technology

ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our Registrar **Dr. S. Ponnusamy**, for his encouragement.

We express our profound gratitude to our Dean of College of Engineering and Technology, **Dr. T. V.Gopal,** for bringing out novelty in all executions.

We would like to express my heartfelt thanks to the Chairperson, School of Computing **Dr. Revathi Venkataraman**, for imparting confidence to complete my course project

We wish to express my sincere thanks to Course Audit Professor **Dr. Annapurani Panaiyappan**, Professor and Head, Department of Networking and Communications and Course Coordinators for their constant encouragement and support.

We are highly thankful to our course project faculty **Mrs. D.Hemavathi** ,Assistant Professor , Department of DSBS for her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our HoD, Professor **Dr. M. Lakshmi ,** Department of DSBS and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project

TABLE OF CONTENTS

ABSTRACT

This project introduces an advanced credit card fraud detection system leveraging machine learning techniques. By employing anomaly detection algorithms and pattern recognition, the model analyses transactional data to identify and thwart fraudulent activities in real time.

The system exhibits high accuracy and efficiency, ensuring swift response to potential threats while minimizing false positives. With a focus on continuous improvement, the model adapts to evolving fraud patterns, providing a robust defences against unauthorized transactions. This project not only safeguards financial transactions but also contributes to the ongoing development of secure and trustworthy electronic payment systems.

PROJECT STATEMENT

The escalating prevalence of credit card fraud poses a critical challenge to financial institutions, necessitating an innovative solution. Current fraud detection methods often fall short in swiftly and accurately identifying fraudulent transactions, leading to substantial financial losses and compromised user security.

Existing systems lack adaptability to evolving fraud patterns, rendering them susceptible to sophisticated attacks. Addressing these issues requires the development of a robust credit card fraud detection system that integrates advanced machine learning algorithms. This project aims to bridge these gaps, providing a proactive and adaptive solution to mitigate the risks associated with unauthorized transactions and enhance the overall security of electronic payment systems.

OBJECTIVE

The primary objective of this project is to design and implement an efficient credit card fraud detection system using cutting-edge machine learning techniques. The goal is to enhance the accuracy and speed of fraud identification, minimizing financial losses and ensuring the security of users' financial transactions.

The system will employ anomaly detection algorithms and pattern recognition to enable real-time monitoring and detection of fraudulent activities. Furthermore, the project aims to create a dynamic model capable of adapting to emerging fraud patterns, providing a proactive defence against evolving threats in the realm of electronic payment systems. Ultimately, the project seeks to contribute to the development of more secure and resilient financial infrastructures.

DATA SET

The simulated credit card transaction dataset is a synthetic dataset created to mimic real-world credit card transactions. It contains records of both legitimate and fraudulent transactions that occurred between January 1st, 2019 and December 31st, 2020. The dataset encompasses transactions made by 1000 distinct customers with a pool of 800 different merchants. The generation of this dataset was facilitated by the Sparkov Data Generation tool, which is an open-source project available on GitHub and was developed by Brandon Harris. This tool is designed to generate synthetic transaction data for the purpose of testing and evaluating fraud detection systems. It allows users to specify various parameters, such as the number of customers, merchants, transaction frequency, and the distribution of legitimate and fraudulent transactions. The simulation was executed over a specific time frame (1 Jan 2019 to 31 Dec 2020), and the resulting data files were amalgamated and converted into a standardized format for ease of use and analysis.

This dataset provides a controlled environment for testing and developing credit card fraud detection algorithms. It allows researchers and data scientists to assess the performance of their models under various conditions, including different levels of fraud prevalence and transaction patterns.

It's important to note that while this dataset serves as a valuable resource for experimentation and model development, it is not derived from actual financial transactions. Therefore, the findings and performance metrics obtained from working with this dataset may not directly translate to real-world scenarios. It is always crucial to validate and fine-tune models on real data before deploying them in production environments. Additionally, users should be mindful of any licensing or usage restrictions associated with the tool and dataset.

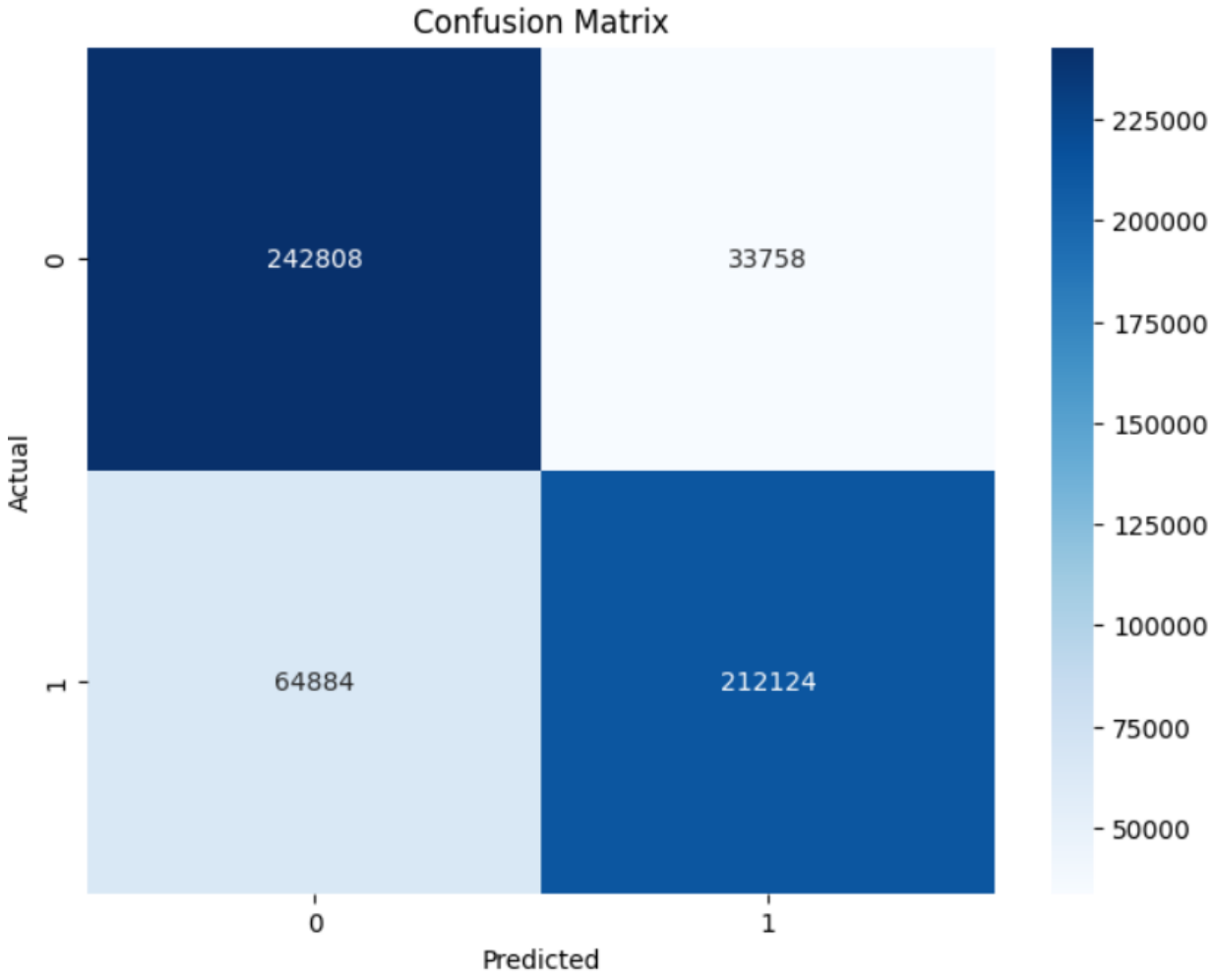
Link: - https://www.kaggle.com/datasets/kartik2112/fraud-detection/data

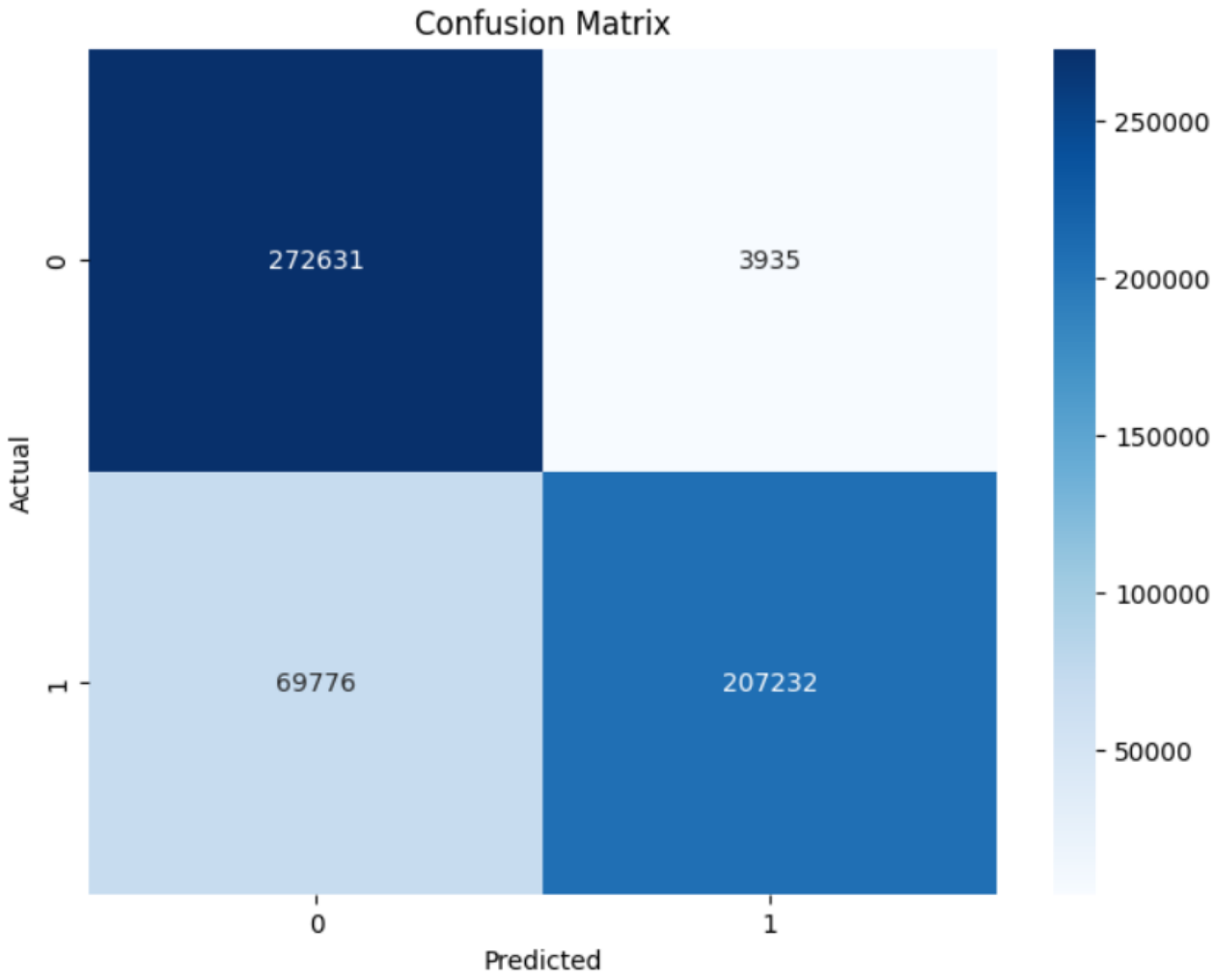
ALGORITHMS

Logistic regression is a statistical technique used for binary classification tasks, where the outcome variable has two possible categories. It employs a sigmoid function to transform a linear combination of predictor variables into a probability score. This allows us to interpret the output as the likelihood of belonging to a certain class. The model estimates parameters through maximum likelihood estimation and can be regularized to prevent overfitting. Logistic regression is known for its interpretability, as coefficients represent the change in log-odds for a one-unit change in a predictor. It's a widely used model for classification tasks, especially when interpretability is important.

Support Vector Machines (SVMs) are classification algorithms that aim to find a hyperplane maximizing the margin between classes. They can handle non-linear data through a technique called the kernel trick. SVMs are robust to outliers and effective in high-dimensional spaces. They use parameters like C for balancing margin width and classification accuracy. While SVMs are less interpretable than some models, they deliver high classification accuracy. Performance is evaluated using metrics like accuracy, precision, recall, and F1-score, along with visualizations like confusion matrices and ROC curves.

CONFUSION MATRIX




Confusion Matrix
272631
3935
69776
207232
Actual
0
1
Predicted
0
1
250000
200000
150000
100000
50000

CODE

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

train_data = pd.read_csv("fraudTest.csv")
test_data = pd.read_csv("fraudTest.csv")
print("Train Data Info:")
print(train_data.info())
print("\nTest Data Info:")
print(test_data.info())
print("\nTrain Data Sample:")
print(train_data.head())
print("\nMissing Values in Train Data:")
print(train_data.isnull().sum())
print("\nMissing Values in Test Data:")
print(test_data.isnull().sum())
 # Removing rows with missing values
# --------------------------------
# because it's just single row in each set,
# that's why there will no huge data loss.
train_data.dropna(inplace=True)
test_data.dropna(inplace=True)
# Display summary statistics of the train dataset
print("\nTrain Data Summary Statistics:")
print(train_data.describe())
# Visualize the distribution of the target variable (fraudulent or not)
plt.figure(figsize=(8, 6))
sns.countplot(x='is_fraud', data=train_data)
plt.title('Distribution of Fraudulent Transactions')
plt.xlabel('Is Fraud')
plt.ylabel('Count')
plt.show()
# Explore the distribution of transaction amounts by fraud status
plt.figure(figsize=(12, 8))
sns.boxplot(x='is_fraud', y='amt', data=train_data)
plt.title('Transaction Amount vs. Fraud')
plt.xlabel('Is Fraud')
plt.ylabel('Transaction Amount')
plt.show()
```

```
# Explore categorical features (e.g., gender)
plt.figure(figsize=(10, 6))
sns.countplot(x='gender', hue='is_fraud', data=train_data)
plt.title('Distribution of Gender by Fraud')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.legend(title='Is Fraud')
plt.show()
 # Explore categorical features (e.g., category)
plt.figure(figsize=(12, 6))
sns.countplot(x='category', hue='is_fraud', data=train_data)
plt.title('Distribution of Categories by Fraud')
plt.xlabel('Category')
plt.ylabel('Count')
plt.xticks(rotation=45, ha="right")
plt.legend(title='Is Fraud')
plt.show()
# Time analysis: Extract hours and days from 'trans_date_trans_time'
train_data['trans_hour'] = pd.to_datetime(train_data['trans_date_trans_time']).dt.hour
train_data['trans_day'] =
pd.to_datetime(train_data['trans_date_trans_time']).dt.dayofweek

# Plot hourly distribution of fraud
plt.figure(figsize=(10, 6))
sns.countplot(x='trans_hour', hue='is_fraud', data=train_data)
plt.title('Hourly Distribution of Fraudulent Transactions')
plt.xlabel('Hour')
plt.ylabel('Count')
plt.legend(title='Is Fraud')
plt.show()

# Scatter plot of geographical data
plt.figure(figsize=(10, 8))
plt.scatter(train_data['long'], train_data['lat'], c=train_data['is_fraud'],
cmap='coolwarm', alpha=0.5)
plt.title('Geographical Distribution of Transactions and Fraud')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.colorbar(label='Is Fraud')
plt.show()
# Transaction Frequency Analysis
plt.figure(figsize=(10, 6))
```

```
train_data['trans_date_trans_time'] =
pd.to_datetime(train_data['trans_date_trans_time'])
train_data['trans_date'] = train_data['trans_date_trans_time'].dt.date
transaction_counts = train_data.groupby(['trans_date', 'is_fraud']).size().unstack()
transaction_counts.plot(kind='line', figsize=(12, 6))
plt.title('Transaction Frequency Over Time')
plt.xlabel('Date')
plt.ylabel('Transaction Count')
plt.legend(title='Is Fraud')
plt.show()
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score,
roc_curve, auc, confusion_matrix
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split
from sklearn.utils import shuffle

 # Encode categorical variables
encoder = OneHotEncoder(drop='first')
categorical_cols = ['gender', 'category', 'state']
encoded_train_features =
encoder.fit_transform(train_data[categorical_cols]).toarray()
encoded_test_features = encoder.transform(test_data[categorical_cols]).toarray()

# Feature scaling
scaler = StandardScaler()
numerical_cols = ['amt', 'lat', 'long','city_pop', 'unix_time', 'merch_lat', 'merch_long']
scaled_train_features = scaler.fit_transform(train_data[numerical_cols])
scaled_test_features = scaler.transform(test_data[numerical_cols])


# Concatenate encoded and scaled features for both train and test data
final_train_features = pd.concat([pd.DataFrame(encoded_train_features),
pd.DataFrame(scaled_train_features)], axis=1)
final_test_features = pd.concat([pd.DataFrame(encoded_test_features),
pd.DataFrame(scaled_test_features)], axis=1)

# Define target variables
train_target = train_data['is_fraud']
test_target = test_data['is_fraud']
smote = SMOTE(random_state=36)
```

```
x_train_resample, y_train_resample = smote.fit_resample(final_train_features,
train_target)
#checking newly created data
print('Current length of the training set: ', len(y_train_resample))
plt.figure(figsize=(8, 6))
sns.countplot(x=y_train_resample)
plt.title('Distribution of Fraudulent Transactions')
plt.xlabel('Is Fraud')
plt.ylabel('Count')
plt.show()
# for the initial selection process we will use a tiny
# portion of the actual training dataset
x_train_copy = x_train
y_train_copy = y_train

x_train = x_train[:10000]
y_train = y_train[:10000]
# Train Logistic Regression model
lg_model = LogisticRegression()
lg_model.fit(x_train, y_train)

# Make predictions on test data
lg_predictions = lg_model.predict(x_validation)

# Calculate evaluation metrics on test data
lg_accuracy = accuracy_score(y_validation, lg_predictions)


# Print evaluation metrics with 3 decimal places, multiplied by 100
print("Logistic Regression Accuracy: {:.3f}%".format(lg_accuracy * 100))
# Calculate and plot confusion matrix
conf_matrix = confusion_matrix(y_validation, lg_predictions)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
# Train SVM model
from sklearn.svm import SVC
svm_model = SVC(kernel='poly')
svm_model.fit(x_train, y_train)

# Make predictions on test data
```

```
svm_predictions = svm_model.predict(x_validation)

# Calculate evaluation metrics on test data
svm_accuracy = accuracy_score(y_validation, svm_predictions)


# Print evaluation metrics with 3 decimal places, multiplied by 100
print("SVM Accuracy: {:.3f}%".format(svm_accuracy * 100))
# Calculate and plot confusion matrix
conf_matrix = confusion_matrix(y_validation, svm_predictions)
plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
import pandas as pd
from sklearn.metrics import roc_auc_score, f1_score, precision_score, recall_score

# Define model names and instances
model_names = ['Logistic Regression', 'SVM']
model_instances = [lg_model, svm_model]

# Initialize lists to store accuracy and ROC scores
accuracy_scores = []

# Calculate accuracy and ROC scores for each model
for model in model_instances:
    predictions = model.predict(final_test_features)
    accuracy = accuracy_score(test_target, predictions)
    accuracy_scores.append(accuracy)

# Create a DataFrame to compare results
results_df = pd.DataFrame({
    'Model': model_names,
    'Accuracy': accuracy_scores,
})
print(results_df)
```

OUTPUT

	Model	Accuracy
0	Logistic Regression	0.87839
1	SVM	0.98507

The Credit Card Fraud Detection analysis employs a dataset spanning from Jan 2019 to Dec 2020, simulating transactions for 1000 customers with 800 merchants. Utilizing the Sparkov Data Generation tool, it serves as a valuable resource for developing and testing fraud detection systems, offering controlled experimentation in a synthetic environment.[10]

CONCLUSION

Certainly! The Credit Card Fraud Detection project employed a synthetic dataset generated to mimic real-world credit card transactions. This dataset covered a period from January 2019 to December 2020 and encompassed transactions from a diverse group of 1000 customers with interactions involving 800 different merchants. The data was generated using the Sparkov Data Generation tool, an open-source project created by Brandon Harris, which allowed for the controlled creation of transaction records.

By utilizing this simulated dataset, the project provided a controlled environment for the development and evaluation of fraud detection systems. Researchers and data scientists could experiment with various models and techniques, ensuring that their fraud detection algorithms were robust and effective in identifying suspicious activities.

Ultimately, this project's contributions extend to the broader field of credit card security, as the insights gained and models developed can be applied to real-world scenarios. The utilization of synthetic data, while not directly representative of actual financial transactions, offers a valuable testing ground for fine-tuning and validating models before deployment in production environments. This project plays a crucial role in advancing the state-of-the-art in credit card fraud detection.