CENTER FOR DEVELOPMENT OF ADVANCED COMPUTING

REPORT ON BANK TRANSACTION ANALYSIS AND FRAUD DETECTION

PG-DBDA MARCH 2023

Submitted by:

Team 10 - The Augmenters

Sayali Rajesh Chaudari

Shamali Rajendra Shengokar

Shiva Kumar Sriram

Shivtej Anil Jadhav

Shrikant Eknathrao Narwade

Saurabh Nanasaheb Jadhav

1. INTRODUCTION:

Nowadays Bank transactions usage has been drastically increased across the world, now people believe in going cashless and are completely dependent on online transactions. The Bank transactions has made the digital transaction easier and more accessible. A huge number of dollars of loss are caused every year by the criminal Bank transactions transaction. The PwC global economic crime survey of 2017 suggests that 48% of organizations experienced economic crime.

In an era where digital transactions have become an integral part of our daily lives, ensuring the security of financial systems is of paramount importance. Financial institutions, such as banks, are increasingly relying on advanced technology and data analytics to analyze customer transactions and detect potential fraudulent activities. The "Bank Transaction Analysis and Fraud Detection Project" aims to enhance the security and integrity of financial operations by leveraging data-driven approaches to identify and prevent fraudulent transactions.

2. BACKGROUND AND MOTIVATION:

As the volume and complexity of financial transactions continue to grow, traditional manual methods of detecting fraud have proven insufficient. Modern financial systems generate vast amounts of data, including transaction history, account details, timestamps, location data, and more. Analyzing this data manually is not only time-consuming but also prone to errors, leading to delayed fraud detection and potential financial losses. Therefore, the need for automated and efficient fraud detection techniques has become crucial.

3. PROBLEM STATEMENT:

The core challenge of the "Bank Transaction Analysis and Fraud Detection" project is to develop a predictive model that can effectively distinguish between genuine and fraudulent transactions based on transactional attributes. The project aims to address the following key problem areas:

Dataset Preparation: Collect, preprocess, and clean transaction data to create a reliable dataset that represents the diversity of customer behavior and transaction patterns.

Feature Engineering: Select and engineer relevant features from the transaction data that provide valuable information for detecting fraudulent activities. The challenge is to identify features that capture the nuances of transactions while minimizing noise.

Imbalanced Data: Develop strategies to handle imbalanced datasets, where the number of genuine transactions significantly outweighs the fraudulent ones. Addressing this imbalance is crucial to prevent models from being biased towards the majority class.

Model Selection: Choose appropriate machine learning algorithms, such as classification techniques, that can effectively learn from the data and make accurate predictions about the legitimacy of transactions.

Model Evaluation: Develop a robust evaluation framework to assess the model's performance using metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve.

4. LITERATURE SURVEY

Financial fraud detection is a critical challenge faced by the banking industry in the era of digital transactions. As financial systems become increasingly sophisticated, so do the techniques employed by fraudsters. To address this issue, researchers and practitioners have turned to machine learning approaches to create effective fraud detection models. In this literature survey, we explore key studies, methodologies, and advancements in the domain of bank transaction analysis and fraud detection using machine learning.

1. Traditional Rule-based Methods:

Early efforts in fraud detection relied on rule-based systems that employed predefined heuristics and thresholds to identify suspicious transactions. These methods lacked adaptability and struggled to capture complex fraud patterns. While foundational, they were limited by their inability to handle the evolving nature of fraud tactics.

2. Machine Learning-Based Approaches:

Machine learning has emerged as a transformative tool for detecting fraud in real-time. Researchers have applied various supervised and unsupervised techniques to develop accurate and efficient fraud detection models. Some notable approaches include:

Supervised Learning: Researchers have employed classification algorithms like Decision Trees, Support Vector Machines, and Random Forests to classify transactions as either legitimate or fraudulent. These models learn from labeled data and use features such as transaction amount, location, and time to make predictions.

Anomaly Detection: Unsupervised learning techniques, such as clustering and neural networks, are used to identify anomalies in transaction data. These methods detect deviations from normal patterns and are especially useful for detecting previously unseen fraud techniques.

Ensemble Methods: Combining multiple models, such as ensemble classifiers or hybrid models, has proven effective in enhancing fraud detection accuracy. Ensembles mitigate the weaknesses of individual models and provide robust predictions.

3. Feature Engineering and Selection:
The success of machine learning models heavily relies on the quality of features used. Researchers have explored feature engineering techniques to extract meaningful information from transaction data. Feature selection methods, such as Recursive Feature Elimination and LASSO, help improve model efficiency by focusing on the most informative attributes.

4. Imbalanced Data Handling:
Addressing the class imbalance between legitimate and fraudulent transactions is a key challenge. Techniques like oversampling, undersampling, and Synthetic Minority Over-sampling Technique (SMOTE) have been proposed to alleviate this issue and prevent models from being biased towards the majority class.

5. Deep Learning:
Recent advancements in deep learning have also influenced fraud detection. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks excel in sequential data analysis, making them suitable for capturing transaction sequences and identifying anomalies.

6. Real-time Processing and Big Data:
Fraud detection models need to operate in real-time to prevent fraudulent transactions from being processed. Distributed computing frameworks like Apache Spark and Hadoop enable efficient processing of large-scale transaction data, enabling timely predictions.

7. Evaluation Metrics:
Researchers evaluate the performance of fraud detection models using metrics like accuracy, precision, recall, F1-score, and Receiver Operating Characteristic (ROC) curve. These metrics provide insights into the model's ability to balance detection accuracy and false positives.

5.LIBRARIES USED

1. Pandas:

Pandas is an open-source Python library that provides fast, flexible, and efficient data structures and data analysis tools. It is designed to make data manipulation and analysis in Python easier and more intuitive. Pandas is widely used in data preprocessing, data cleaning, exploration, and transformation tasks, particularly in the field of data science.

Key Features:

- Data Import and Export
- Data Cleaning and Transformation
- Selection and Filtering
- Aggregation and Grouping
- Merging and Joining
- Time Series and Date Functionality

Pandas are also able to delete rows that are not relevant, or contains wrong values, like empty or NULL values. This is called cleaning the data.

2. Numpy:

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It is open-source software. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using

Numpy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

3. Matplotlib:

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

- Create publication quality plots.
- Make interactive figures that can zoom, pan, update.
- Customize visual style and layout.
- Export to many file formats.
- Embed in JupyterLab and Graphical User Interfaces.
- Use a rich array of third-party packages built on Matplotlib

4. Sklearn:

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

Important features of scikit-learn:

- Simple and efficient tools for data mining and data analysis. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means, etc.

- Accessible to everybody and reusable in various contexts.

- Built on the top of NumPy, SciPy, and matplotlib.

- Open source, commercially usable – BSD license

5. SciPy

• Python SciPy is basically a library which has Python NumPy andMathematical algorithms as its building blocks
• SciPy library is used at a great extent in the field of scientific computdtions ond processing.

Functionalities offered by Python SciPy:

- Linear Algebra
- Working with Polynomials
- Integration
- Fourier Transforms
- Interpolation Functions
- Special Functions

6. Imblearn:

Imbalanced-learn (imported as **imblearn**) is an open source, MIT-licensed library relying on scikit-learn and provides tools when dealing with classification with imbalanced classes.

Key Features:

- Resampling Techniques
- Combination of Sampling Techniques
- Ensemble Methods
- Cost Sensitive Learning
- Generating Synthetic samples
- Class Balancing Transformers

6. CLASSIFICATION MODELS USED:

1. Logistic Regression:

Logistic Regression is a statistical technique used for binary classification tasks, where the goal is to predict the probability that a given input belongs to a particular class. Despite its name, logistic regression is not a regression method but a classification method. It's called "logistic" because it uses the logistic function to model the relationship between input features and the probability of the binary outcome.

Key Features:

- Binary Classification
- Probabilistic Interpretation
- Linear Decision Boundary
- Log Odds Transformation
- Parameter Estimation
- Sigmoid Function
- Assumption of Linearity
- Interpretability

2. Decision Tree Classifier:

A Decision Tree Classifier is a machine learning algorithm used for classification tasks that involves partitioning the input space into segments to make predictions about the class labels of instances. It constructs a tree-like model where each internal node represents a decision based on a feature, each branch represents an outcome of the decision, and each leaf node represents a class label or a class distribution**.**

Key Features:

- Hierarchy of Decisions
- Binary or Multiclass Classification
- Recursive Partitioning
- Feature Selection

- Splitting Criteria
- Overfitting prevention
- Ensemble Learning
- Categorical and Numerical Features
- Non linear relationships

3. Random Forest Classifier:

A Random Forest Classifier is an ensemble learning algorithm used for classification tasks that combines the predictions of multiple individual decision trees to make more accurate and robust predictions. It works by creating a "forest" of decision trees, each trained on different subsets of the data and using different subsets of features, and then aggregating their predictions to make the final classification**.**

Key Features:

- Ensemble of decision trees
- Bootstrap Aggregating(Bagging)
- Feature Randomness
- Voting or Averaging
- Reduction of Variance
- Robustness to overfitting
- Hyperparameter tuning
- Feature Importance

4. KNN Classifier:

The K-Nearest Neighbors (KNN) Classifier is a machine learning algorithm used for classification tasks that makes predictions based on the majority class of its k nearest neighbors in the feature space. It is a simple yet effective instance-based learning method that assigns a new instance to the class that is most common among its k nearest neighbors.

- Instance Based Learning
- Proximity Based Prediction
- Hyperparameter K

- Distance Metric
- Non Parametric Algorithm
- Local Decision Boundaries
- Lazy Learning
- Classification and Regression

5. XGBoost Classifier:

XGBoost (Extreme Gradient Boosting) Classifier is an ensemble learning algorithm that leverages the power of gradient boosting to create a strong predictive model for classification tasks. It is an optimized and efficient implementation of the gradient boosting framework that combines the strengths of multiple weak learners (decision trees) to make accurate predictions.

Key Features:

- Gradient Boosting Framework
- Optimized Implementation
- Handling missing values
- Hyperparameter tuning
- Parallel processing
- Early stopping
- Handling Imbalanced Classes
- Ensemble Learning
- Scalability

6. Naïve Bayes Classifier:

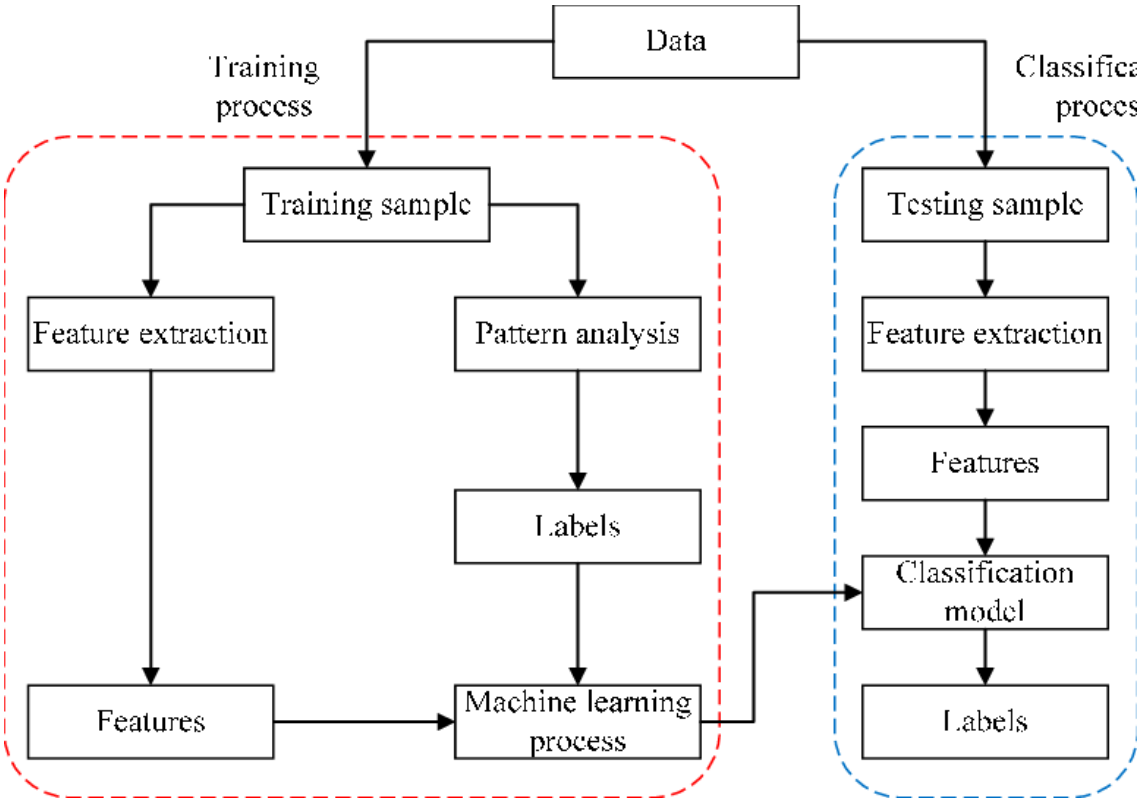
The Naive Bayes Classifier is a probabilistic machine learning algorithm used for classification tasks. It's based on the principles of Bayes' theorem and assumes that features are conditionally independent given the class label. Despite its "naive" assumption, Naive Bayes can be surprisingly effective in various text classification and simple categorization tasks.

Key Features:

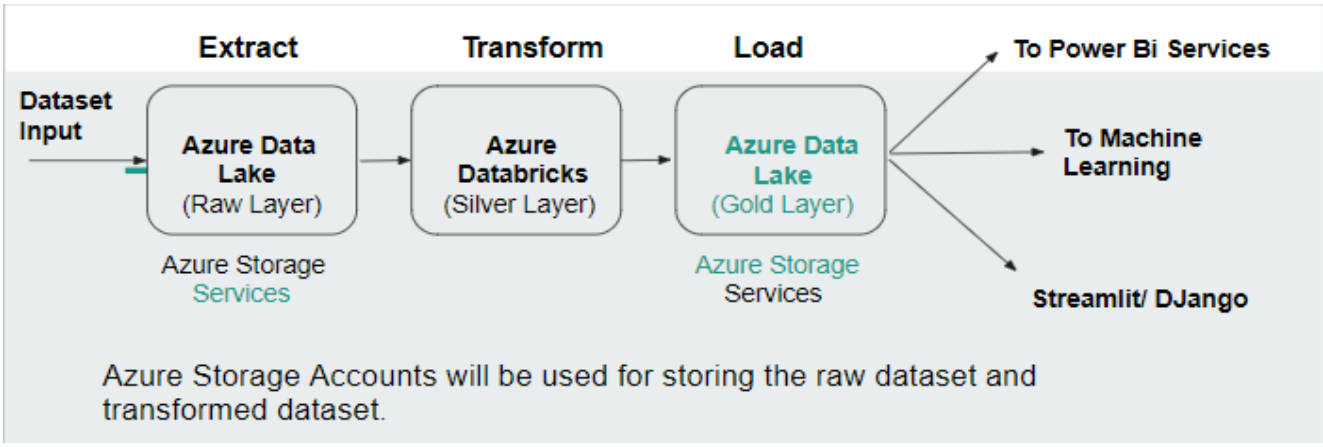
- Probabilistic Approach

- Based on Bayes Theorem
- Conditional Independence
- Multinominal Naïve Bayes
- Gaussian Naïve Bayes
- Sparse Data Handling
- Efficient training
- Limited Complex Relationships

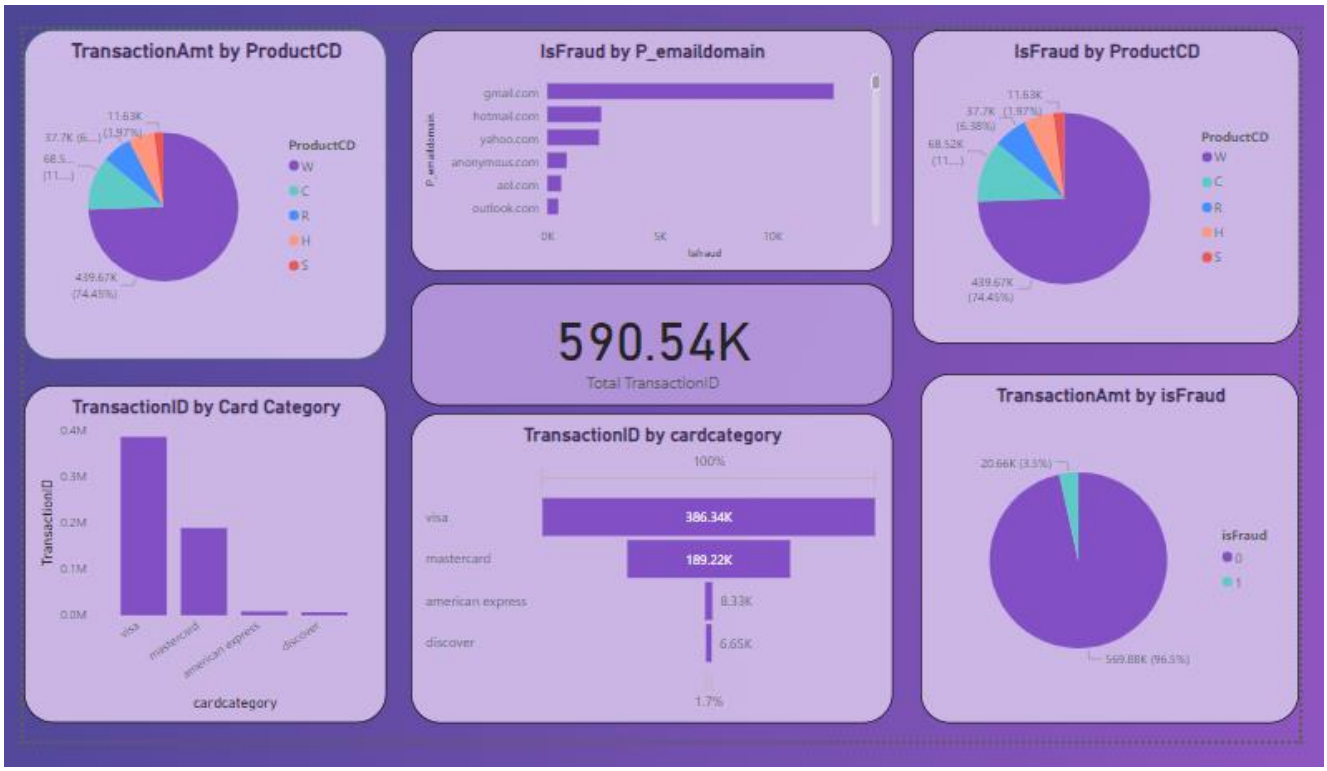
7. FLOWCHART:


Data
Training process
Classifica proces
Training sample
Testing sample
Feature extraction
Pattern analysis
Feature extraction
Features
Labels
Classification model
Features
Machine learning process
Labels

8. PIPELINE FLOW:



9. DATASET VISUALIZATION USING POWERBI:



10. EVALUATION METRICS

1. Logistic Regression:

```
Accuracy: 0.8944932073180576
Confusion Matrix:
 [[135995   6878]
 [ 23185 118881]]
Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.95      0.90    142873
           1       0.95      0.84      0.89    142066

    accuracy                           0.89    284939
   macro avg       0.90      0.89      0.89    284939
weighted avg       0.90      0.89      0.89    284939
```

2. Decision Tree:

```
Accuracy: 0.9799536041047382
Confusion Matrix:
 [[139780   3093]
 [  2619 139447]]
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.98      0.98    142873
           1       0.98      0.98      0.98    142066

    accuracy                           0.98    284939
   macro avg       0.98      0.98      0.98    284939
weighted avg       0.98      0.98      0.98    284939
```

3.Random Forest:

```
confusion_matrix
[[142542   2833]
 [   331 139233]]

accuracy_score
0.9888958689403697

classification_report
              precision    recall  f1-score   support

           0       1.00      0.98      0.99    145375
           1       0.98      1.00      0.99    139564

    accuracy                           0.99    284939
   macro avg       0.99      0.99      0.99    284939
weighted avg       0.99      0.99      0.99    284939
```

4.XGBoost Classifier:

```
Accuracy: 0.9673298495467451
Confusion Matrix:
 [[140036   2837]
 [  6472 135594]]
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.98      0.97    142873
           1       0.98      0.95      0.97    142066

    accuracy                           0.97    284939
   macro avg       0.97      0.97      0.97    284939
weighted avg       0.97      0.97      0.97    284939
```

5. KNN Classifier:

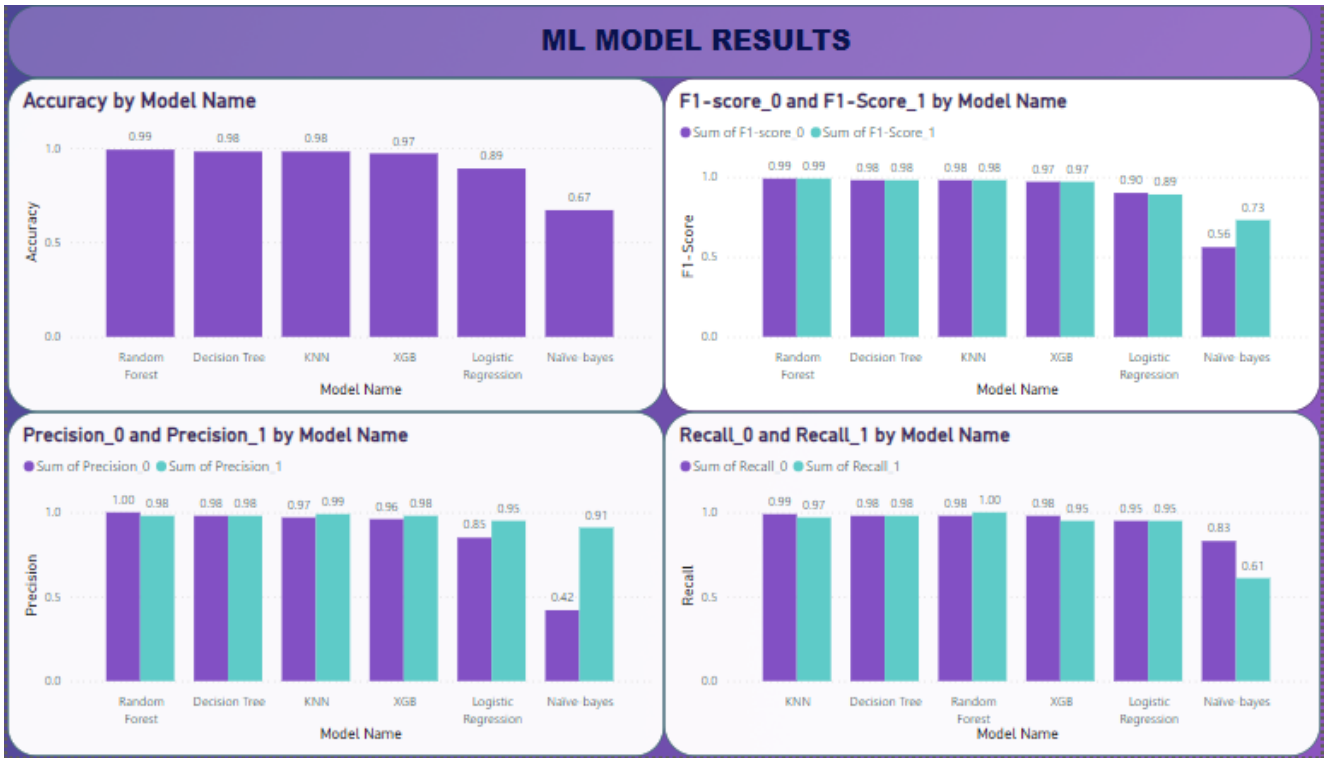
```
confusion_matrix
[[138462   1812]
 [  4411 140254]]

accuracy_score
0.9781602378052846

classification_report
              precision    recall  f1-score   support

           0       0.97      0.99      0.98    140274
           1       0.99      0.97      0.98    144665

    accuracy                           0.98    284939
   macro avg       0.98      0.98      0.98    284939
weighted avg       0.98      0.98      0.98    284939
```

6. Naive Bayes:

```
confusion_matrix
[[ 60518  12538]
 [ 82355 129528]]

accuracy_score
0.6669708253345453

classification_report
              precision    recall  f1-score   support

           0       0.42      0.83      0.56     73056
           1       0.91      0.61      0.73    211883

    accuracy                           0.67    284939
   macro avg       0.67      0.72      0.65    284939
weighted avg       0.79      0.67      0.69    284939
```

11. CLASSIFICATION MODELS RESULTS:

The results are visualized using PowerBI



12. CONCLUSION AND FUTURE SCOPE

CONCLUSION:

- The features those are significant in predicting whether the transaction is fraudulent are not are the Account number,which type of card is used whether it is Visa,Rupay,AmericanExp etc. , whether card is Debit/ credit card , what kind of transaction are we performing for eg - buying a product , subscribing to some paid platform , refund of product , the pincode and email associated with transaction , how many times a card is being used , how many times money is credited / debited among other features.

- Among the six Classification models, Random Forest is performing the best.

- 75% of fradulent cases are done with discover type card from card6 category

- From card4 category, the maximum number of fradulent cases are done with creditcard with 66%

- In ProductCD category, the maximum number of fradulent cases are done with cashwithdrawls with 12%.

- Protonmail.com has highest fradulent transactions with 5%

- most of fraudlent cases are done from mobile devices as compare to desktop devices with 10% and 6% respectively.

FUTURE SCOPE:

The future scope of a bank transaction analysis and fraud detection project is extensive, driven by advancements in technology, data availability, and the evolving strategies of fraudsters. Here are some potential areas of future development and enhancement for such projects:

Advanced Machine Learning Models: As machine learning techniques continue to evolve, there is a potential for the development of more advanced models that can detect sophisticated and evolving fraud patterns. This includes exploring deep learning models, ensemble methods, and hybrid models that combine multiple algorithms for improved accuracy.

Anomaly Detection: Improving anomaly detection techniques to identify unusual patterns that might not fit into known fraud profiles. This could involve exploring unsupervised learning techniques and clustering algorithms to discover novel fraud patterns.

Real-time Detection: Enhancing the system to perform real-time fraud detection during transactions. This requires high-speed processing and instantaneous decision-making to prevent fraudulent transactions from being completed.

Feature Engineering: Continuously refining feature engineering techniques to extract more relevant information from transaction data, including both structured and unstructured data sources.

Big Data and Streaming Analytics: Incorporating big data technologies and stream processing to handle the vast amounts of transaction data generated by modern financial systems. This can improve the accuracy and efficiency of fraud detection.

Behavioural Analysis: Developing models that learn and adapt to customer behaviours over time, distinguishing between legitimate changes in behavior and suspicious activities.

Network Analysis: Exploring the connections between different entities involved in transactions, such as customers, merchants, and accounts, to detect complex fraud networks.

Multi-channel Detection: Expanding fraud detection beyond individual transactions to multiple channels, including online banking, mobile apps, and ATM transactions.

Enhanced User Authentication: Incorporating biometric and behavioral authentication methods to ensure the identity of customers and reduce the risk of account takeover fraud.

Explainable AI: Developing models that not only provide predictions but also explain the reasoning behind them, improving transparency and allowing analysts to understand why a particular transaction was flagged as fraudulent.

Cross-Industry Collaboration: Collaborating with other financial institutions and organizations to share insights and data about emerging fraud trends and attack vectors..

Adapting to New Fraud Techniques: As fraudsters develop new tactics, the system needs to adapt quickly. This might involve incorporating external threat intelligence feeds and machine learning models that can learn from new patterns.

AI-Powered Decision Support: Using artificial intelligence to provide analysts with actionable insights and recommendations for investigating flagged transactions.

Cloud-Based Solutions: Leveraging cloud computing for scalable infrastructure, enabling faster data processing and real-time fraud detection.

Hybrid Models: Combining rule-based systems with machine learning models to harness the strengths of both approaches**.**

In essence, the future of bank transaction analysis and fraud detection lies in the continuous adaptation and improvement of technologies to stay ahead of ever-evolving fraud tactics while providing efficient and seamless experiences for legitimate customers.

13. REFERENCES:

1. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16).

2. Dal Pozzolo, A., Caelen, O., Le Borgne, Y. A., Waterschoot, S., Bontempi, G. (2015). Learned lessons in credit card fraud detection from a practitioner perspective. Expert Systems with Applications, 41(10), 4915-4928.

3. Wu, Y., Du, Y., & Zhang, S. (2020). Credit card fraud detection using a multilayer autoencoder neural network. IEEE Access, 8, 115912-115920.

4. Bhattacharya, S., Srinivasan, D., Das, R., Sivakumar, S., & Srikumar, A. (2020). Detecting Fraud in Transactional Data Using Machine Learning: A Case Study. IEEE Transactions on Industrial Informatics.

5. Zhang, L., Cao, J., & Zhang, G. (2021). Fraud Detection in Mobile Payment with Deep Learning. In International Conference on Information and Communication Technology Convergence (ICTC) (pp. 438-443).