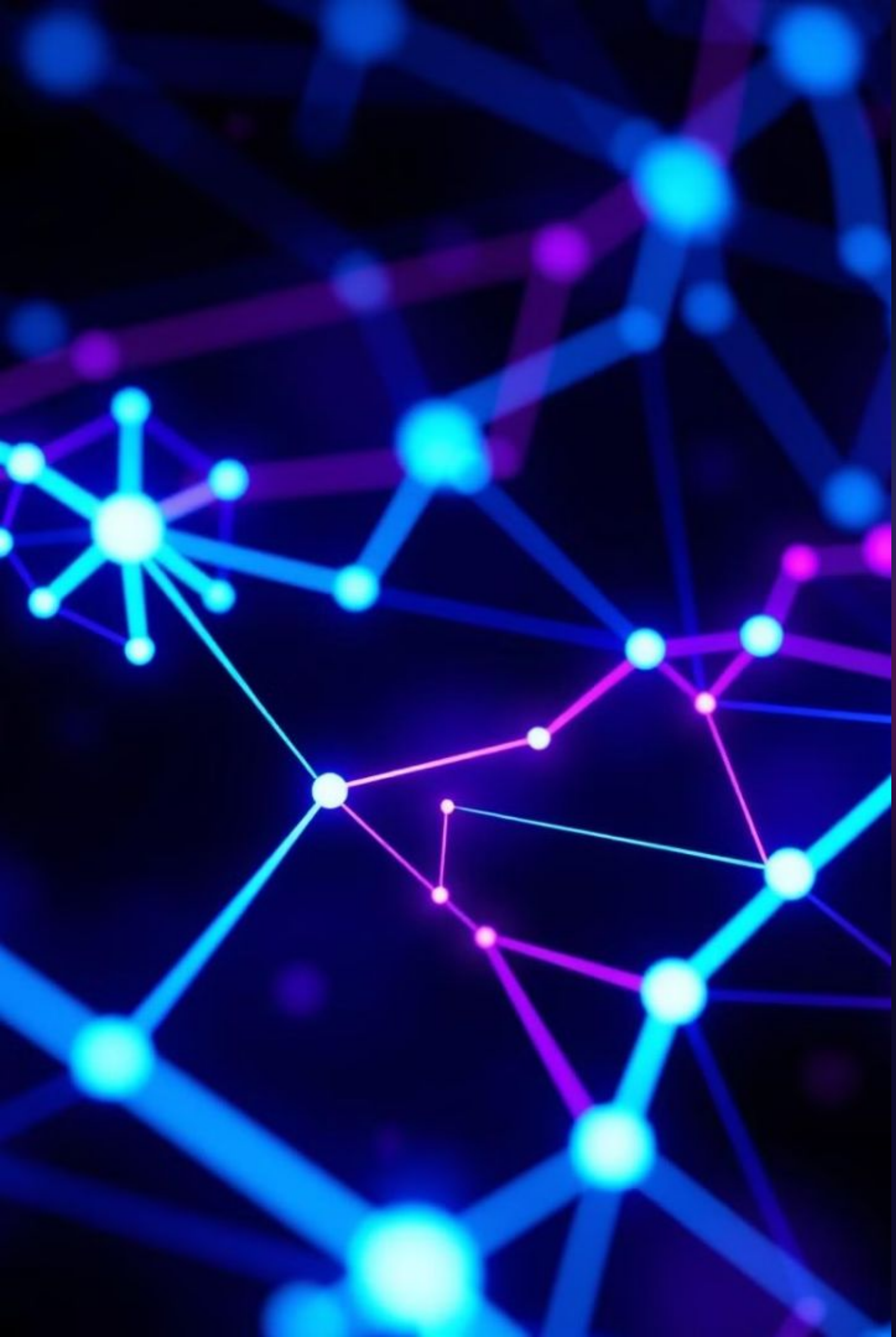


Fraud Detection Project Presentation

WELCOME

Welcome to our Fraud Detection Project. We'll implement algorithms, handle class imbalance, visualize performance, and calibrate model outputs to detect fraudulent activities effectively.

Meet Our Awesome Team

DA8115 Asli

Team member

DA8118 Burcu

Team member

DA8127 Emre

Team member

DA8121 Hasan

Team member

DA8123 Nurdan

Team member

Introduction

1. Project Overview and Objectives
2. Problem Definition and Data Analysis
3. Model Selection and Implementation
4. Handling Class Imbalance
5. Model Performance Visualization
6. Model Calibration
7. Monitoring Model and Data Drift
8. Model Deployment
9. Results and Evaluation

1. Project Overview and Objectives

The Fraud Detection Project is designed to analyze transaction data and identify patterns indicative of fraudulent activity, utilizing machine learning models for detection and prevention.

The goal of this project is to predict whether a credit card transaction is fraudulent. The project involves working with an imbalanced dataset to detect fraud in transactions. The key objectives include analyzing the data, observing the class imbalance, exploring feature correlations, and evaluating model performance.

Logistic Regression, Random Forest, XGBoost, and Neural Network algorithms will be used to build the models. Unbalanced data techniques will be applied to improve performance. Additionally, tools like Deepchecks will be used to monitor data and model drift, and finally, the model will be prepared for deployment via an API.

Fraud Detection

Definition

Fraud detection identifies suspicious activities indicating fraudulent behavior. It monitors transactions and patterns to detect deviations from normal activity. This can involve monitoring transactions, behaviors, or patterns to detect deviations from normal activity that could suggest fraud.

Importance

It prevents financial losses, protects information, maintains trust, and ensures compliance with regulations.

Preventing Financial Losses: Avoiding monetary losses due to fraudulent transactions.

Protecting Personal and Financial Information: Safeguarding sensitive information from unauthorized access.

Maintaining Trust and Reputation: Ensuring trustworthiness and credibility of organizations.

Compliance: Meeting regulatory and legal requirements for fraud prevention.

Types

Common types include financial fraud, identity theft, insurance fraud, healthcare fraud, and e-commerce fraud.

Financial Fraud: Includes activities such as credit card fraud, investment fraud, and loan fraud.

Identity Theft: Illegally acquiring and using someone else's personal information.

Insurance Fraud: Submitting false claims to obtain financial benefits.

Healthcare Fraud: Misrepresenting medical services or billing to receive undue payments.

E-commerce Fraud: Includes fake purchases, account takeovers, and other fraudulent activities in online transactions.

2. Problem Definition and Data Analysis

Dataset

The data comes from Vesta's real-world e-commerce transactions and contains a wide range of features from device type to product features. The dataset is **unbalanced**, the positive class (frauds) account for small amount of all transactions.

Features

Includes transaction details, card info, email domains, and Vesta-engineered features. Identity table has network and digital signature info.

Challenges

Class imbalance in data. Lack of domain knowledge. Need for data preprocessing and visualization.

Content of Trained Data

The dataset used in this analysis comes from Vesta's real-world e-commerce transactions and includes a wide array of features that capture various aspects of transaction details, ranging from device type to product characteristics. Notably, the dataset is imbalanced, with fraudulent transactions representing a small proportion of the total transactions.

Feature Information

Transaction Table:

TransactionDT: Timedelta from a reference point (not an actual timestamp).

TransactionAMT: The transaction amount in USD.

ProductCD: The product code, representing the type of product involved in the transaction.

card1 - card6: Payment card details such as card type, issuing bank, and country.

addr: Address information.

dist: Distance between the transaction parties.

P_emaildomain & R_emaildomain: Purchaser and recipient email domains.

C1-C14: Count-related features (e.g., the number of addresses linked to a payment card). The exact meanings are masked for privacy.

D1-D15: Timedelta-related features (e.g., days since a previous transaction).

M1-M9: Match indicators (e.g., whether the cardholder’s name matches the address, etc.).

Vxxx: Engineered features from Vesta, including ranking, counting, and entity relationships.

Categorical Features in the Transaction Table:

- ProductCD
- card1 - card6
- addr1, addr2
- P_emaildomain
- R_emaildomain
- M1 - M9

Identity Table:

The identity table contains network and digital signature information related to each transaction. These variables capture information such as IP address, Internet Service Provider (ISP), proxy usage, and device/browser information. The exact meaning of many fields is masked for privacy reasons.

Categorical Features in the Identity Table:

- DeviceType
- DeviceInfo
- id_12 to id_38

Merging the Datasets

Both the **Transaction** and **Identity** tables must be merged based on common transaction IDs to form a comprehensive view of each transaction, combining both transactional and identity information.

Key Challenges

Class Imbalance: Fraudulent transactions make up only a small fraction of the overall dataset, which may bias the model if not properly handled.

Lack of Domain Knowledge: Since many features are masked or anonymized, domain knowledge is limited, making it difficult to interpret some features directly.

Exploratory Data Analysis (EDA)

General Characteristics: Explore the overall structure and characteristics of the dataset, including the number of transactions, number of fraud cases, and the distribution of various features.

Missing Values and Data Types: Identify and analyze missing data, and check for inconsistencies or anomalies in data types.

Basic Statistics: Compute basic statistics (e.g., mean, median, distribution) for continuous features like transaction amounts and distances.

Data Visualization:

- Visualize distributions of features to understand patterns.
- Explore correlations between features, focusing on potential relationships with the fraud label.

This analysis will guide feature engineering and model building, aiming to detect fraudulent transactions effectively despite the challenges.

3. Model Selection and Implementation

1 Logistic Regression

Basic model for binary classification. Performance metrics and results analyzed.

2 Random Forest

Ensemble learning method. Robust to overfitting. Results evaluated.

3 XGBoost

Gradient boosting algorithm. Known for high performance. Metrics assessed.

4 Neural Network

Deep learning approach. Capable of capturing complex patterns. Performance analyzed.

4. Handling Class Imbalance

SMOTE

Generates synthetic examples of minority class. Improves model generalization and reduces bias.

Undersampling

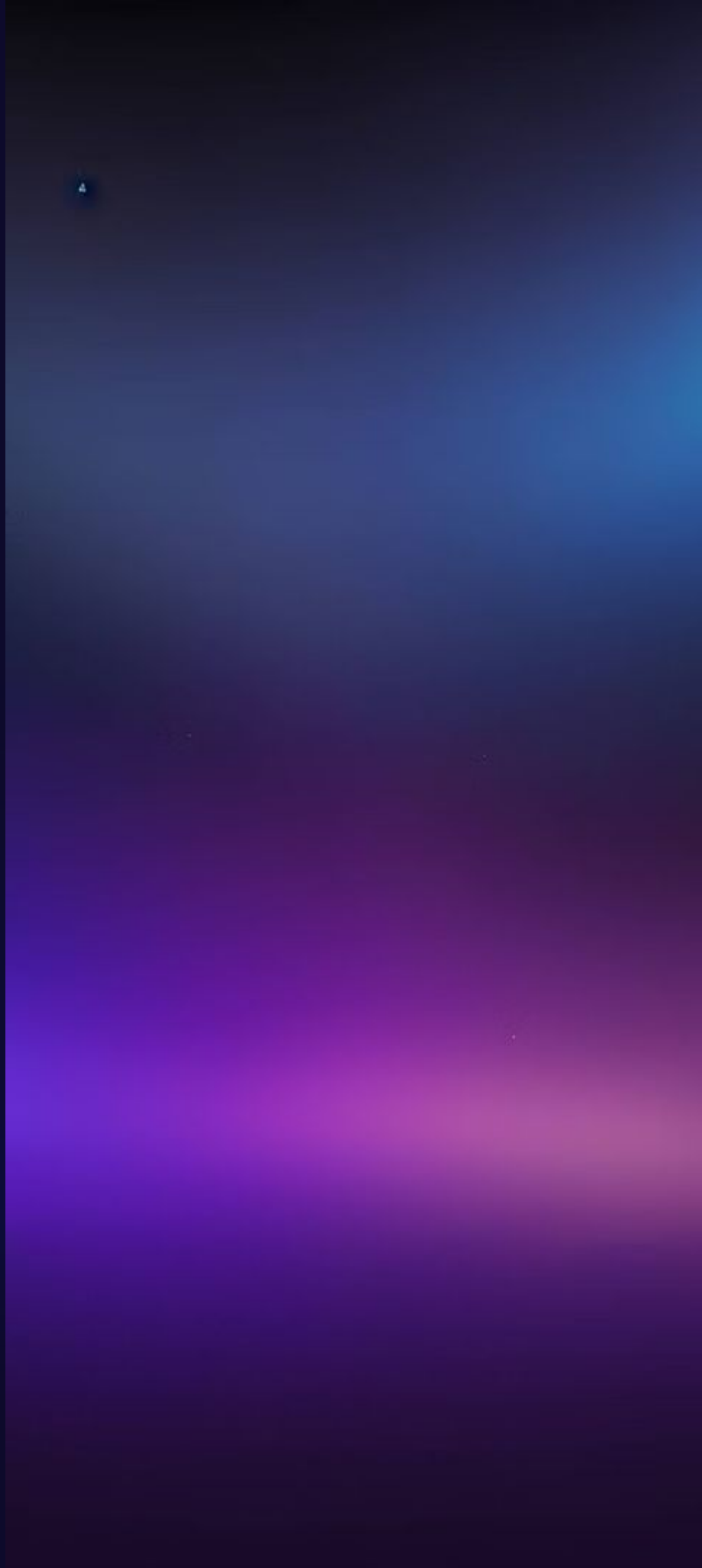
Reduces majority class samples. Balances dataset but may lose information.

Class Weights

Adjusts importance of classes. Penalizes misclassification of minority class more heavily.

Threshold Adjustment

Modifies decision threshold. Improves model's ability to detect minority class.



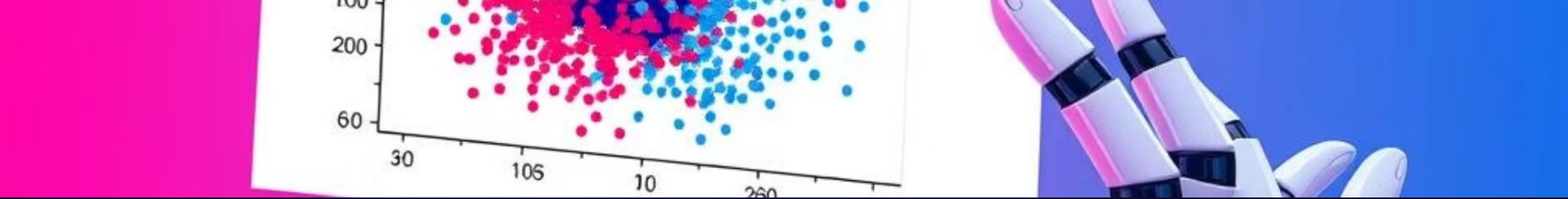
SMOTE (Synthetic Minority Over-sampling Technique)

What is SMOTE?

SMOTE is a powerful over-sampling technique designed to address class imbalance in datasets. In situations where one class (like fraud in fraud detection) is significantly under-represented compared to the majority class, SMOTE helps to generate synthetic examples of the minority class, thus balancing the dataset.

Purpose of SMOTE

The main goal of SMOTE is to improve the performance of machine learning models by providing a more balanced dataset, which can lead to better generalization and reduced bias.



SMOTE

How SMOTE Works

Select Minority Class Samples: SMOTE first identifies the under-represented minority class samples in the dataset.

Find Nearest Neighbors: For each minority class sample, SMOTE locates its k-nearest neighbors in the feature space.

Generate Synthetic Samples: New synthetic data points are created by interpolating between the selected sample and its neighbors. This is done by selecting a random point along the line segment connecting the original sample and one of its neighbors.

Why SMOTE is Important?

SMOTE is widely used to handle imbalanced datasets because it prevents overfitting, unlike simple over-sampling (which duplicates minority class samples), and ensures that the machine learning model learns more generalized patterns rather than memorizing specific examples.

Other Techniques for Handling Class Imbalance

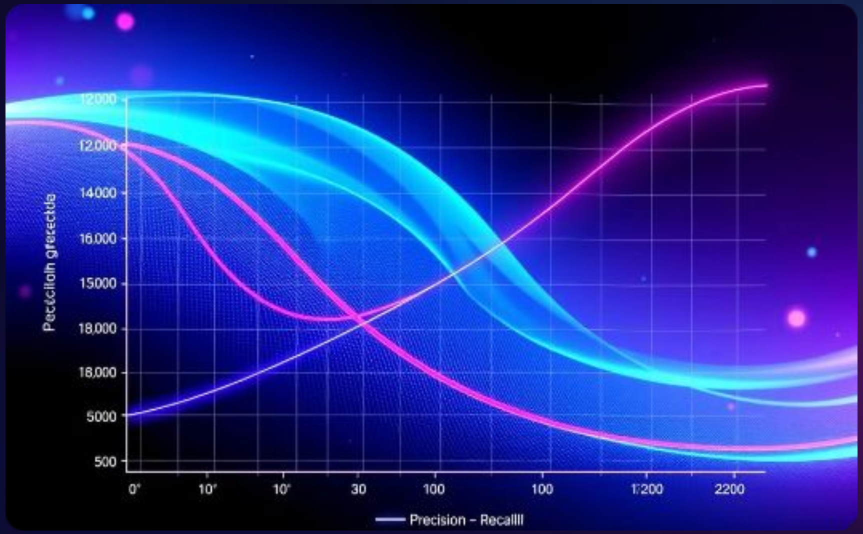
To handle class imbalance, several techniques can be applied alongside SMOTE.

Undersampling involves reducing the number of majority class samples to balance the dataset, though it risks losing valuable information.

Class weight adjustments assign higher weights to the minority class during model training, making the model more sensitive to minority class misclassifications without altering the dataset, though it can lead to overfitting.

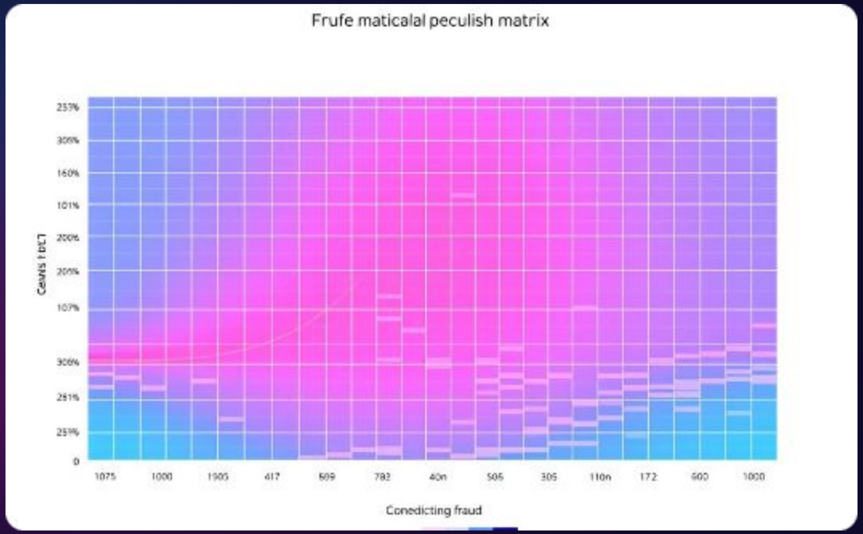
Threshold adjustments modify the decision boundary of the classifier by lowering the probability threshold for predicting the minority class, improving recall but potentially increasing false positives. These methods help ensure better performance in imbalanced datasets, such as in fraud detection.

5. Model Performance Visualization



PRC Curves

Visualizes trade-off between precision and recall. Useful for imbalanced datasets.



Confusion Matrix

Shows true positives, false positives, true negatives, and false negatives.

F1 Scores

Harmonic mean of precision and recall. Balances both metrics.

Model Performance Visualization

In fraud detection and other machine learning tasks, visualizing data and model performance is crucial for understanding outcomes. Two popular tools for visualization are **Seaborn** and **Matplotlib**. Seaborn is ideal for generating informative and aesthetically pleasing statistical plots, while Matplotlib offers flexibility for creating custom plots and fine-tuning visual elements.

Performance Graphs

PRC Curves (Precision-Recall Curves): These curves are useful in imbalanced datasets, like fraud detection, as they show the trade-off between precision and recall across different thresholds.

Confusion Matrix: A confusion matrix visually displays the counts of true positives, false positives, true negatives, and false negatives, giving insight into how well a model distinguishes between classes.

F1 Scores and Other Metrics: The F1 score, which balances precision and recall, along with accuracy, precision, and recall, can be plotted to evaluate model performance across various thresholds or models.

These tools and graphs help in interpreting model effectiveness and making data-driven improvements.

6. Model Calibration

Model Calibration Techniques for Probability Adjustment

In machine learning, particularly in fraud detection, accurate probability estimates are crucial for effective decision-making. Calibration techniques adjust predicted probabilities to improve their alignment with observed outcomes, enhancing model performance. Two common methods are Isotonic Regression and Platt Scaling.

Platt Scaling is a parametric calibration method that fits a logistic regression model to the base model's predicted probabilities. This transformation aims to create more accurate probability estimates.

Isotonic Regression

Training: This technique fits a monotonically increasing function to the predicted probabilities while minimizing error. The function is piecewise constant, ensuring a smooth adjustment.

Application: The calibrated function is then applied to the predicted probabilities, leading to more reliable estimates.

Advantages of Isotonic Regression

Isotonic Regression is flexible and can capture complex, non-linear relationships without requiring specific functional forms. It is especially well-suited for situations where the relationship between predictions and actual outcomes is non-linear.

Disadvantages of Isotonic Regression

Isotonic Regression requires more data than Platt Scaling and can be prone to overfitting with small datasets. It can also be computationally intensive, especially when dealing with large datasets.

Model Calibration

Platt Scaling

Training: A logistic regression model is trained on the predicted probabilities, using the true binary outcomes as targets. This model aims to capture the relationship between the predicted probabilities and the actual outcomes.

Application: The fitted logistic model is then used to convert base model predictions into calibrated probabilities, resulting in improved estimates.

Advantages of Platt Scaling

Platt Scaling is a simple and fast method, computationally efficient, and easy to implement. It works particularly well when the base model's predicted probabilities are not well-calibrated, especially in binary classification tasks.

Disadvantages of Platt Scaling

Platt Scaling assumes a logistic relationship between predicted probabilities and true outcomes. It is less effective with complex, non-linear relationships than Isotonic Regression.

Model Calibration

Calibrated Model Performance

Calibration techniques provide several key benefits:

Better Probability Estimates: Calibrated probabilities are more accurate, crucial for risk assessment and decision-making.

Enhanced Decision-Making: Well-calibrated models lead to more informed decisions, especially in applications where probability estimates are vital.

Improved Metrics: Calibration can improve metrics sensitive to probability estimates, such as the Brier score and log-loss.

Evaluation of Calibration

Calibration Plots: Use calibration curves or reliability diagrams to visually assess how well the predicted probabilities align with actual outcomes.

Metrics: Evaluate calibration performance with metrics like Brier score and log-loss to quantify how well the probabilities are calibrated.

These techniques ensure that the model's probability outputs are reliable and suitable for critical applications like fraud detection.

7. Monitoring Model and Data Drift

1

Data Drift

Changes in data distribution over time. Can affect model performance.

2

Model Drift

Decline in model performance. Impacts decision-making and business outcomes.

3

Deepchecks

Open-source library for evaluating and validating ML models and data pipelines.

Monitoring Model and Data Drift

What Are Data Drift and Model Drift?

Data Drift refers to changes in the statistical properties of the input data over time, while **Model Drift** is the decline in model performance as the data changes. Both are crucial to monitor as they can reduce the accuracy and reliability of machine learning models, negatively impacting decision-making and business operations. Regular monitoring, detection, and adaptation strategies like retraining and updating models are essential to ensure long-term model effectiveness and performance.

Why is Model Drift Important?

Performance Degradation: As data evolves, model predictions may become less accurate, leading to poor decision-making and business outcomes.

User Trust: Consistent model performance builds trust; model drift can erode confidence in the model’s outputs.

Operational Efficiency: Keeping models up-to-date and effective is vital for maintaining smooth, efficient business operations.

Monitoring Model and Data Drift

Deepchecks for Model Monitoring

What is Deepchecks?

Deepchecks is an open-source library designed to evaluate and validate machine learning models and data pipelines. It offers automated checks and tools that help ensure models continue to perform well, remain robust, and maintain their value over time. Deepchecks focuses on three key areas:

Data Integrity: Ensures that the data used for both training and inference is accurate, consistent, and of high quality.

Train-Test Validation: Validates that the model generalizes well from the training data to unseen test data, preventing overfitting.

Model Evaluation: Assesses model performance, including aspects like calibration and robustness, helping detect issues like drift.

Examples of Using Deepchecks

Deepchecks can be used to:

Monitor Data Drift: Automatically detect shifts in data distribution that may impact model performance.

Validate Model Performance: Continuously assess the model’s accuracy and calibration over time.

These tools help in proactively identifying issues and taking corrective actions, ensuring machine learning models remain effective and reliable.

8. Model Deployment

1

Deployment Process

Streamlit API used for model deployment. Ensures accessibility and ease of use.

2

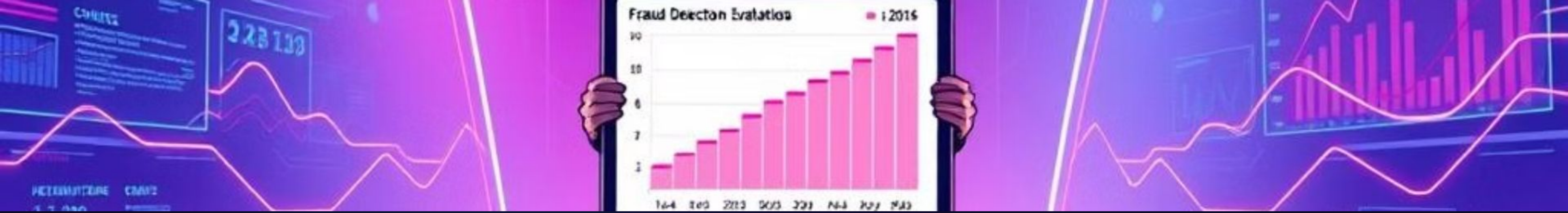
User Interface

Interactive interface designed for inputting transaction data and receiving fraud predictions.

3

Demonstration

Simple demo showcasing the deployed model's functionality and real-time predictions.



9. Results and Evaluation

Model Performance Summary

Best Performing Models: Highlight the top-performing models based on accuracy, F1 score, precision-recall metrics, and overall efficiency in detecting fraud. These models demonstrated strong generalization and reliability in identifying fraudulent transactions.

Lessons Learned

Data Preprocessing: Effective handling of missing values, feature engineering, and balancing the class distribution significantly impacted model performance.

Model Selection: Different algorithms offer varying strengths; selecting models that best handle imbalanced data, such as tree-based methods, resulted in superior outcomes.

Evaluation: Consistent monitoring and evaluation of key metrics like precision, recall, and F1 score helped fine-tune models for real-world scenarios, balancing false positives and false negatives.

Future Work

Potential Improvements: Enhancing feature selection and exploring advanced techniques like deep learning models could further improve detection accuracy.

Additional Features: Incorporating more granular transaction details, external data sources, or behavioral analytics could enrich model predictions.

Impact on Global Industries: The future of fraud detection lies in developing more adaptable, fair, and ethical systems. By advancing technology and integrating continuous monitoring and feedback loops, we can create robust fraud detection solutions that benefit both organizations and users on a global scale.