FICO FORCE

Detect, Track, Secure

Project Report

FINNOVATION ‘25

FICO FORCE

Detect, Track, Secure

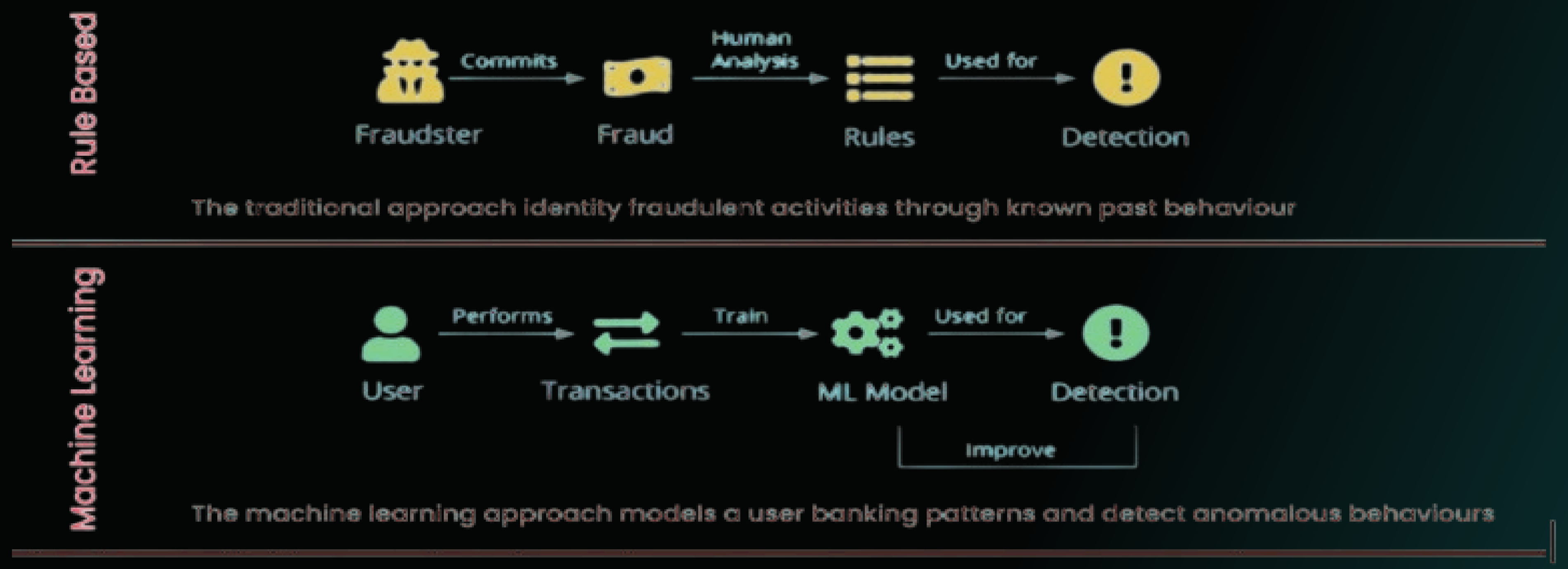
PROBLEM STATEMENT

Fraudulent Account/Transaction Detection

The goal is to identify fraudulent behavior at the transaction or account level using patterns in financial data. The challenge lies in the adversarial nature of fraud, where malicious actors often mimic normal behavior to evade detection. Additionally, the class imbalance—with few fraudulent cases—makes model training difficult.

Last Known Location Inference of Defaulters

The goal is to estimate the last known location of loan defaulters using anonymized, timestamped mobile tower connection logs. The challenge is to model users' movements over time while preserving temporal consistency and handling data noise and missing values, relying solely on the provided dataset for fraud detection but synthetic data for extracting location of the defaulter.



Precision Fraud Detection & Defaulter Tracking

FICO FORCE

Detect, Track, Secure

Dataset Overview

Customer Demographics & Identity

Captures basic customer information such as age, income band, and identity verification/compliance indicators (KYC, UID, eKYC).

Loan & Account Characteristics

Includes details about loan amounts, tenures, EMIs, and account-specific parameters like standing instructions and credit limits.

Behavioral Transaction Patterns

Provides 12-month time series data on credit, debit, average balances, and outstanding trends, reflecting the customer's financial behavior.

Loan Portfolio Overview

Summarizes the customer's total loans, sanctioned amounts, outstanding balances, and the age and status of their oldest and newest loans.

Delinquency & Risk History

Tracks customer defaults, risk grades (e.g., RG1–RG4), IRAC classification changes, and NPA-related behavior over time.

Credit Score & Inquiry Dynamics

Includes credit scores, credit score changes, and the frequency of credit report inquiries, indicating credit health and activity.

Credit Bureau Profile

Details the number, status, and performance of credit accounts reported to credit bureaus, including overdue and newly opened accounts.

Loan Default Indicator

The target variable flags whether a customer defaulted on their loan, forming the basis for supervised learning models.

FICO FORCE

Detect, Track, Secure

We face several challenges that require proactive solutions

Mixed Data Types & Non-Numeric Columns

- Inconsistent binary flags (eg. SI_FLG) contained ambiguous values like 'YN', 'NY', and 'YY' alongside standard 'Y'/'N', risking misclassification during numeric conversion.
- Unstructured duration strings (e.g., "2yrs 3mon") lacked standardized formatting, complicating automated parsing into numeric months without data loss or errors.

Missing Values

- The dataset has extensive missing values, particularly in credit history and behavioural flag features.
- This complicates analysis, reduces data quality, and may bias model results.

Categorical Columns

- The challenge lies in appropriately encoding these categories.
- For example, preserving order in INCOME_BAND1 while ensuring the nominal variables are numerically represented for modelling, despite their lack of inherent order.

High Dimensionality

- The dataset contained a total of 139 columns, spanning account-level, behavioral, transactional, and derived features.
- The large number of features presented risks such as redundancy, multicollinearity, and overfitting.

Handling Temporal and aggregated features

- Dataset includes monthly time-series features like credit (ONEMNTHCR–TWELVEMNTHCR), debit (ONEMNTHSDR–TWELVEMNTHSDR), and outstanding balances (ONEMNTHOUTSTANGBAL–TWELVEMNTHOUTSTANGBAL)
- Challenges: high dimensionality, multicollinearity, and missing values.

FICO FORCE

Detect, Track, Secure

CODE SNIPPETS

Binary Flags

- **Columns**: SI_FLG, LOCKER_HLDR_IND, UID_FLG, KYC_FLG, INB_FLG, EKYC_FLG
- **Code**:

```
df[flag_columns] = df[flag_columns].replace({'Y': 1, 'N': 0}).fillna(0).astype(int)
df[col] = df[col].map({'Y': 1, 'N': 0, 'YN': 1, 'NY': 1, 'NN': 0, 'YY': 1}).fillna(0)
```

Categorical Ranking

- **Columns**: INCOME_BAND1
- **AGREG_GROUP:** Unique values = ['#Total Auto Loan', '#Total Xpress Credit', '#Housing Loan', '#Education Loan Total']
- **PRODUCT_TYPE:** Unique values = ['AUTO LOAN', 'PERSONAL LOAN', 'HOME LOAN', 'EDUCATION LOAN']
- **TIME_PERIOD:** Unique values = ['DEC24', 'NOV24', 'JAN25']
- **Code**:

```
# Encoding INCOME_BAND1
income_band_mapping = {chr(65 + i): i + 1 for i in range(13)}
df['INCOME_BAND1'] = df['INCOME_BAND1'].map(income_band_mapping)
# Label encoding the remaining categorical columns
from sklearn.preprocessing import LabelEncoder
label_cols = ['AGREG_GROUP', 'PRODUCT_TYPE', 'TIME_PERIOD']
for col in label_cols:
    df[col] = LabelEncoder().fit_transform(df[col].astype(str))
```

Duration Strings

- **Columns**: AVERAGE_ACCT_AGE1, CREDIT_HISTORY_LENGTH1
- **Code**:

```
s = str(s).lower().strip()
pattern = r'(?:(\d+)\s*yrs?)?\s*(?:(\d+)\s*(?:months|mon))?'
match = re.match(pattern, s)
years, months = int(match.group(1) or 0), int(match.group(2) or 0)
return years * 12 + months
```

FICO FORCE

Detect, Track, Secure

Task 1: Detect Fraudulent Activity

Preprocessing Engine

- Imputed missing values using median (numeric) and mode (categorical) strategies
- Scaled numeric features via StandardScaler
- Feature selection employed Pearson correlation to identify top 20 linearly relevant predictors, while Kernel PCA (RBF kernel, 15 components) extracted non-linear patterns from 72 monthly time-series variables (credits, debits, balances).

Feature Engineering

- Generated monthly aggregates per user: mean, standard deviation, count of transaction amounts
- Computed linear trend slopes to capture spend acceleration or decline over time
- Created user-level behavioral profiles across multiple time windows

Model Training & Optimisation

- **XGBoost** hyperparameters tuned using **GridSearchCV**
- Deployed a **weighted hard voting ensemble:**
  - · Logistic Regression (weight = 1)
  - · XGBoost (weight = 2)
  - · Multi-layer Perceptron (weight = 1)
- Evaluation prioritized F1-score, ensuring a balance between precision and recall
- Resolved class imbalance using SMOTE to synthetically enhance minority (fraud) class representation

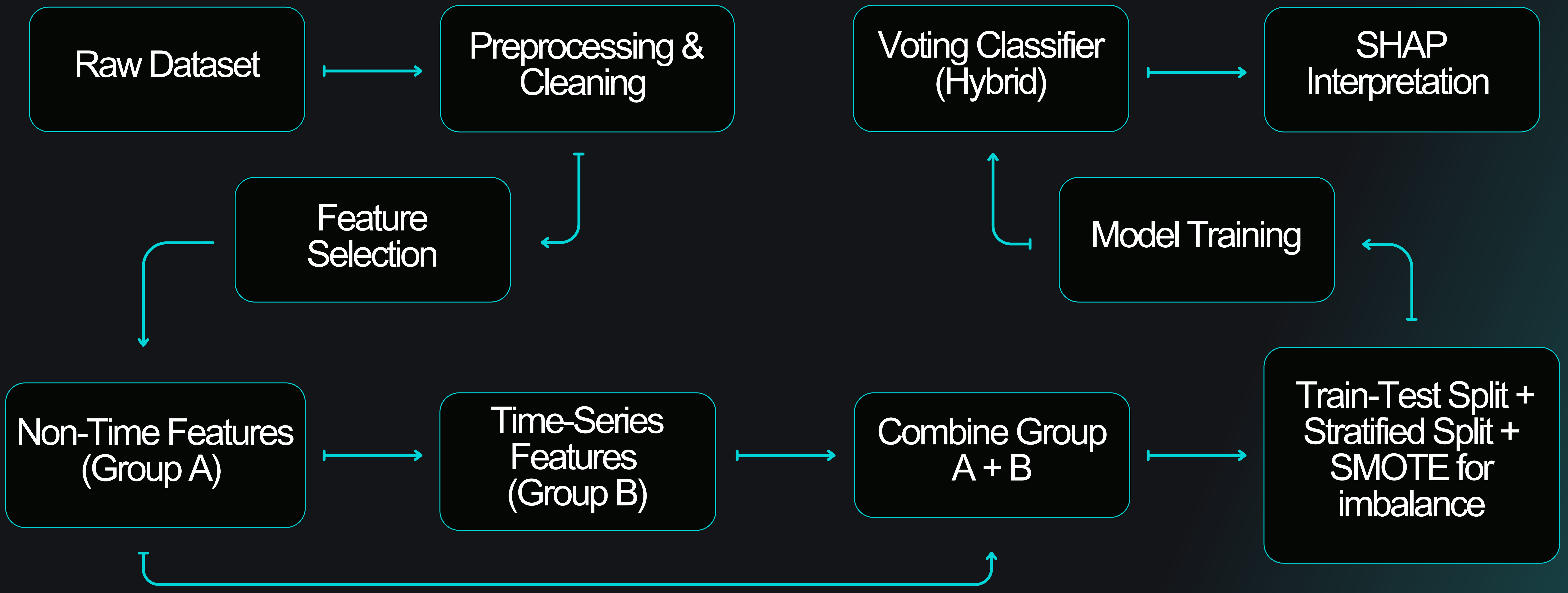
Post-Processing & Interpretability

- Integrated SHAP (SHapley Additive Explanations) for post-hoc interpretability
- Ranked features based on mean absolute SHAP values
- Top contributors to fraud prediction identified (e.g., credit_utilization, trend_slope, cancel_count)
- Supports explainable AI (XAI) requirements for risk and compliance

FICO FORCE

Detect, Track, Secure

Solution Architechture



FICO FORCE

Detect, Track, Secure

Feature Selection using Correlation

- Computed Pearson correlation to assess linear relationships between features and target.
- Shortlisted features with absolute correlation > 0.1; top value observed was ~0.2, indicating weak dependencies.
- Selected top 20 features with highest absolute correlation for modeling.

Top 20 features (grp A)

['CRIFF_11', 'TIMES_IRAC_SLIP', 'TIMES_IRAC_UPR', 'CRIFF_22', 'LOAN_TENURE', 'NO_YRS_RG3', 'FIRST_NPA_TENURE', 'LAST_1_YR_RG2', 'LATEST_NPA_TENURE', 'LATEST_CR_DAYS', 'ACCT_AGE', 'CREDIT_HISTORY_LENGTH1', 'LAST_1_YR_RG3', 'TOT_IRAC_CHNG', 'OLDEST_RESIDUAL_TENURE', 'INCOME_BAND1', 'CRIFF_33', 'DEC_CRIFFCHNG1', 'ACCT_RESIDUAL_TENURE', 'LATEST_LON_TAKEN']

Formula used to find Pearson correlation

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}}$$

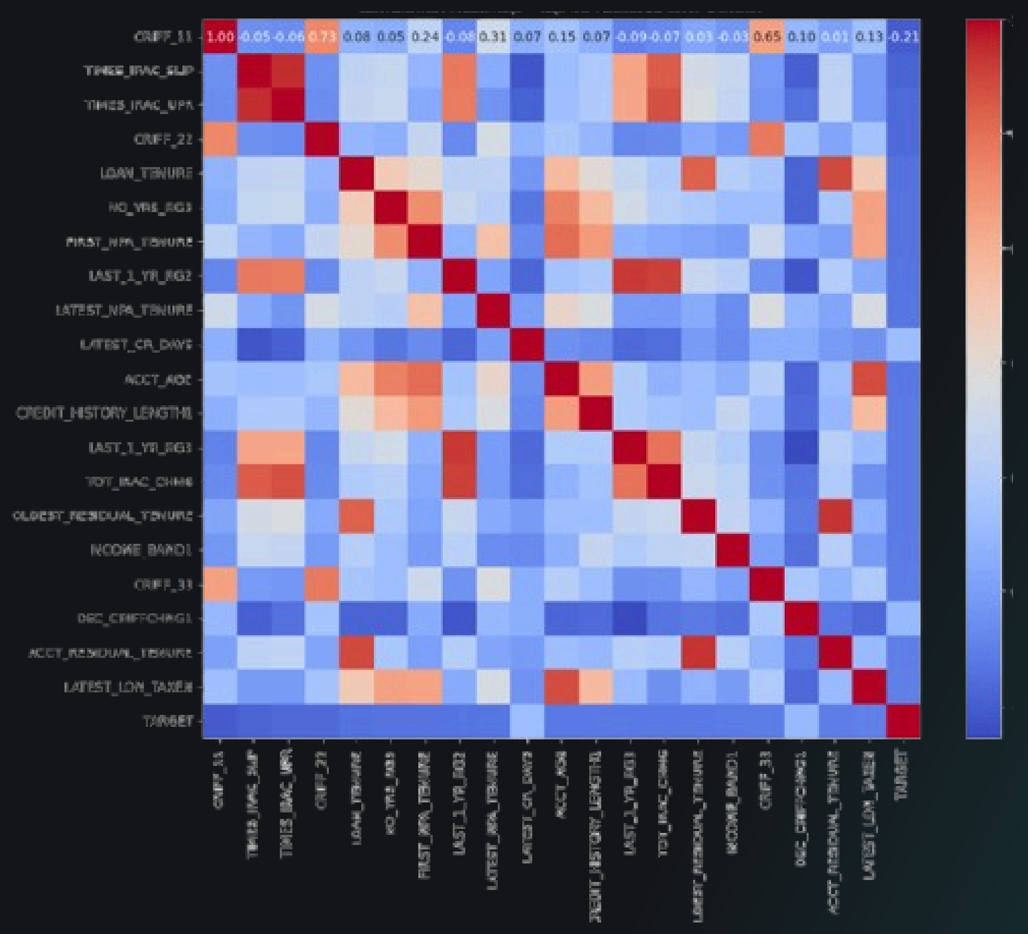
Significance

It measures the linear relationship between two variables, ranging from -1 (strong negative) to +1 (strong positive).

Code

```
correlations = df_numeric.corr()['TARGET']
selected_features =
correlations[abs(correlations) >
0.1].drop('TARGET').index.tolist(
```

Heat Map for the top 20 features



NOTE: Correlation only captures linear relationships and may miss important nonlinear dependencies—hence, this was used as just one of several selection methods.

FICO FORCE

Detect, Track, Secure

Feature Selection using PCA

We applied Kernel Principal Component Analysis (Kernel PCA) to capture complex, non-linear patterns in monthly financial time-series data.

Formula of Kernel PCA

$$K_{ij} = k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$$

The input included 72 features across 12 months, such as:

- Monthly averages (ONEMNTHAVGMTD, AVGQTD, AVGYTD)
- Credits and debits (ONEMNTHCR, TWOMNTHCR, etc.)
- Outstanding balances for each month

Kernel PCA extends traditional PCA by projecting data into a higher-dimensional space using a non-linear kernel function, enabling the detection of relationships that linear PCA might overlook.

We used the Radial Basis Function (RBF) kernel, which is effective in revealing curved or clustered structures in financial behavior across time.

This transformation helped reduce dimensionality while preserving non-linear trends in customer activity, supporting better downstream modeling (e.g., fraud detection or risk profiling).

Process Involved

- Filling missing values with column means
- Standardizing the features to zero mean and unit variance
- Applying Kernel PCA with 15 components to reduce dimensionality while preserving important information

FICO FORCE

Detect, Track, Secure

MODEL ARCHITECTURE & ENSEMBLE STRATEGY

XGBoost

- Tree-based ensemble.
- Captures non-linear feature interactions.
- Native feature importance.
- SMOTE support for class balancing.
- GridSearchCV tuned (max_depth, learning_rate, etc.)

Logistic Regression

- High interpretability (baseline model)
- L1 regularization
- Fast inference
- Useful for transparent decision boundaries

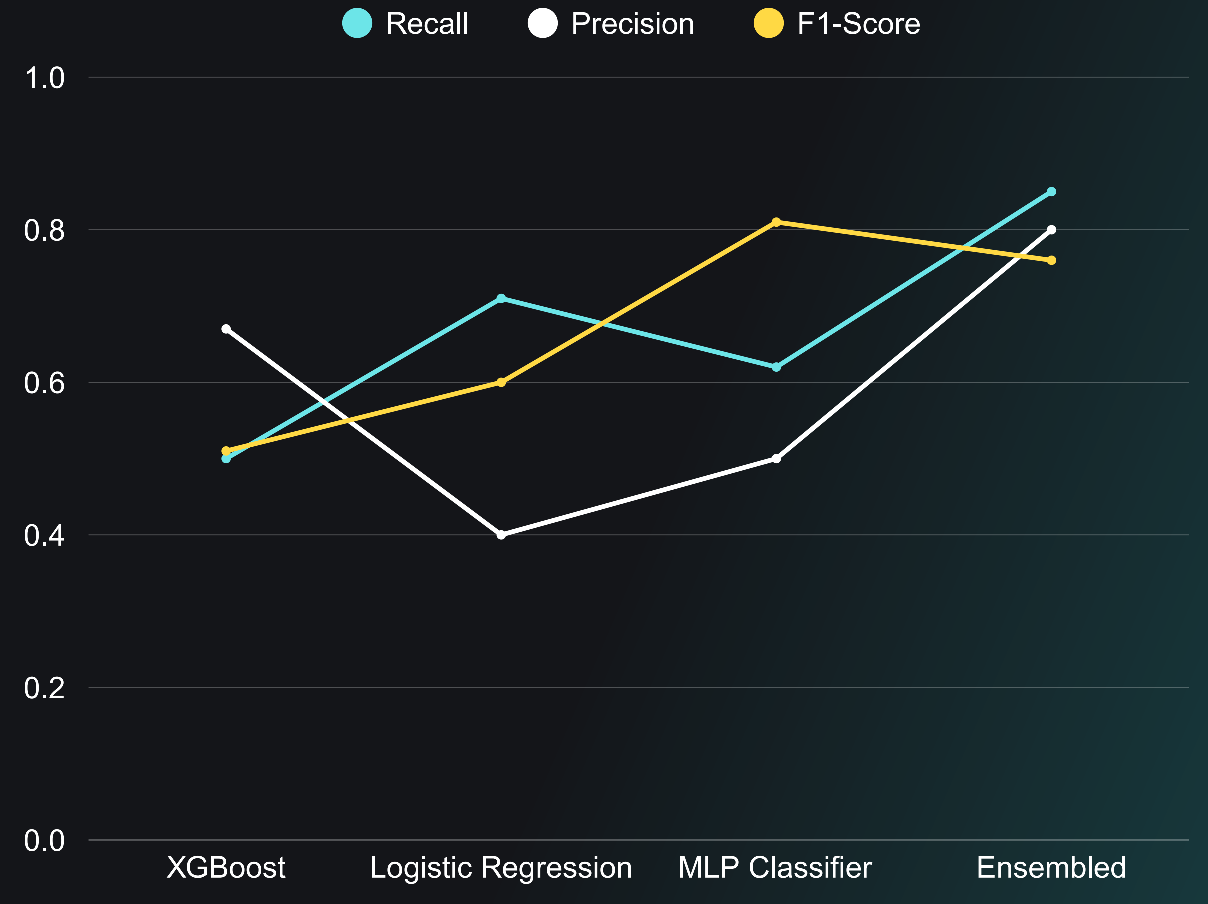
MLP Classifier

- Neural network for complex patterns
- [64, 32, 16] dense layers
- ReLU activations
- Dropout regularization to prevent overfitting

XGBoost Weight: 2 + **Logistic Weight: 1** + **MLP Weight: 1** = **Ensemble Weighted Hard Voting**

Final Achievement: 91% Accuracy, 85% Recall, 80% Precision with Full Interpretability.

Model performance comparison



Interpretability

- SHAP Summary Plot ranks top drivers:
  - Credit_utilization
  - trend_slope
  - cancel_count
- Global + Local interpretability
- Stakeholder and compliance ready
- Aligns with Explainable AI (XAI) norms

FICO FORCE

Detect, Track, Secure

Location Inference of Defaulters

We aim to find the last known location of defaulters by analyzing the sequence of their mobile tower connections over time.

1 Clean the Data and Sort by Time

We organize each person's tower connection history in ascending order of time to understand their movement.

2 Check Signal Strength and Duration

We check how strong and long each tower connection was.

3 Combine All Tower Interactions

We analyze all tower connections together to spot patterns and frequent locations.

4 Identify the Most Likely Location

We identify the tower where the person was likely last seen.

FICO FORCE

Detect, Track, Secure

Last Location Inference

user_id	tower_id	timestamp	signal_strength	duration
U001	T102	2025-05-01 08:02:10	-75	45
U001	T110	2025-05-03 15:12:22	-70	120
U001	T110	2025-05-05 09:47:01	-68	80
U001	T115	2025-05-08 17:20:45	-85	30

Assumption:
The last meaningful tower connection is the closest proxy to the person's last known location.

Steps to be followed after identifying defaulter

Group by user_id

Sort each user's records by timestamp (ascending)

Select the last tower entry as the predicted location

Example Output:

user_id: U001
last_tower_id: T115
last_seen_time:
2025-05-08 17:20:45

Handling Edge Cases

Old Last Ping:
If the last tower connection is over 30 days old, mark the location as outdated and unreliable.

Frequent Location Changes:
For rapid tower switches, prioritize stable and strong signals over recency to avoid false predictions.

Sparse Connection History:
If very few logs exist, flag the case as "insufficient data" and suggest manual review or revalidation.

Simultaneous Tower Connections:
When multiple towers appear at the same time, identify the most frequent or central one as the likely region.

Conclusion

In this project, we designed and implemented a robust, interpretable, and modular pipeline for fraud detection and defaulter localization. Starting from a noisy and heterogeneous dataset with 139 features, we built a data-centric workflow combining statistical rigor, dimensionality reduction, and ensemble modeling to achieve high predictive performance.

Our hybrid model—an ensemble of XGBoost, Logistic Regression, and MLP using weighted soft voting—achieved:

Accuracy: 91%
Recall: 85%
Precision: 80%

Defaulter Localisation

- Inferred location using latest tower connection
- Timestamp-based, simple, and scalable method
- Interpretable and real-time ready
- No external data or mapping used

FICO FORCE

Detect, Track, Secure

Thank you

Team Name: FICO FORCE

Team Members

Vansh	Suramya
Avani	Mansi
Shraddhan	Disha