# Machine Learning in Fraud Detection

**Project Team Members:**
Baoshu Feng (Team Leader): A20343813

## 1. Project Proposal

- **Introduction**

  Payment fraud has a long history and is the most common form of online fraud in the United States and around the world. However, in recent times, digital fraud has increased significantly, and many people are unable to distinguish well. The use of up-to-date systems with the ability to learn has become indispensable as companies strive to stay ahead of the game in the face of the relentless threat from criminals. This reflects how organized crime and state-sponsored fraudsters are stepping up their fraud efforts [1].

  The most common approaches to combating online fraud include rules and predictive models that are no longer adapted to the sophistication of today's increasingly sophisticated online threats. The vast majority of emerging attacks in the digital fraud space rely on machine learning and other automated techniques to perpetrate fraud. Integrating AI-based platforms into high-stakes games to detect online fraud is a key part of AI, and today, it enables us to extend online fraud prevention. Digital businesses with specific business models and their fraud analysts can derive results from fraud analysis based on surveillance and unattended machine learning to provide business models with the information they need to detect and stop threats at an early stage [2].

  The results of unsupervised and supervised machine learning are characterised by the detection of anomalies in emerging data, and integrating them into risk assessment is an important step towards artificial intelligence to detect online crime and enhance online prevention.

- **Research Goal**
  - In this case, fraud behavior generate huge influence to all customer experience since fraud behavior is hidden as a small partition in massive transaction. Our objective is to create a predictive model for expected fraud in the Vesta Corporation dataset, in order to identify the fraud before it happen. This fraud prevention system is actually saving consumers millions of dollars per year.

    We are not predicting fraudulent transactions. According to the competition host Lynn here. Once a client (credit card) has fraud, their entire account is converted to isFraud=1. Therefore we are predicting fraudulent clients (credit cards).

- **Specific Questions:**
  - Generate different statistical modesl for fraud detection (For each TransactionID in the test set, predict a probability for the isFraud variable).
  - Benchmark machine learning models on a challenging large-scale dataset.

- **A proposed methodology/approach to the analysis that will be performed.**
  - **Data Preparation and Data Discovery**
    - Import the kaggle dataset to dataframe
    - Analyse dataset structure and visulize different feature in the dataset
      - ➢ Feature Correlation Matrix
      - ➢ Charts for Data Distribution
  - **Data Wrangling and Modeling**
    - Our task is a binary classification task, so we use commonly used metrics for evaluating binary classification. I use Accuracy, Recall, Precision, F1-Score as the evaluation metrics.
    - **Candidate ML Model to Validate the Dataset**
      - ➢ Lightgbm
      - ➢ XGboost
      - ➢ RandomForest
      - ➢ Multilayer Perceptron

**2. Project Outline**

- **Data Source and References**
  - **Overview**
    - The data comes from Vesta's real-world e-commerce transactions and contains a wide range of features from device type to product features.
    - **Data Source:**
      - ➢ https://www.kaggle.com/competitions/ieee-fraud-detection/data [4]
    - **The Structure of the Data**
      - ➢ **Transaction Table**
        - **TransactionDT**: timedelta from a given reference datetime (not an actual timestamp)
        - **TransactionAMT**: transaction payment amount in USD
        - **ProductCD**: product code, the product for each transaction
        - **card1 - card6**: payment card information, such as card type, card category, issue bank, country, etc.
        - **addr**: address

- **dist**: distance
- **P_ and (R__) emaildomain**: purchaser and recipient email domain
- **C1-C14**: counting, such as how many addresses are found to be associated with the payment card, etc. The actual meaning is masked.
- **D1-D15**: timedelta, such as days between previous transaction, etc.
- **M1-M9**: match, such as names on card and address, etc.
- **Vxxx**: Vesta engineered rich features, including ranking, counting, and other entity relations.

➢ **Identity Table**

- Variables in this table are identity information – network connection information (IP, ISP, Proxy, etc) and digital signature (UA/browser/os/version, etc) associated with transactions. They're collected by Vesta's fraud protection system and digital security partners. (The field names are masked and pairwise dictionary will not be provided for privacy protection and contract agreement)

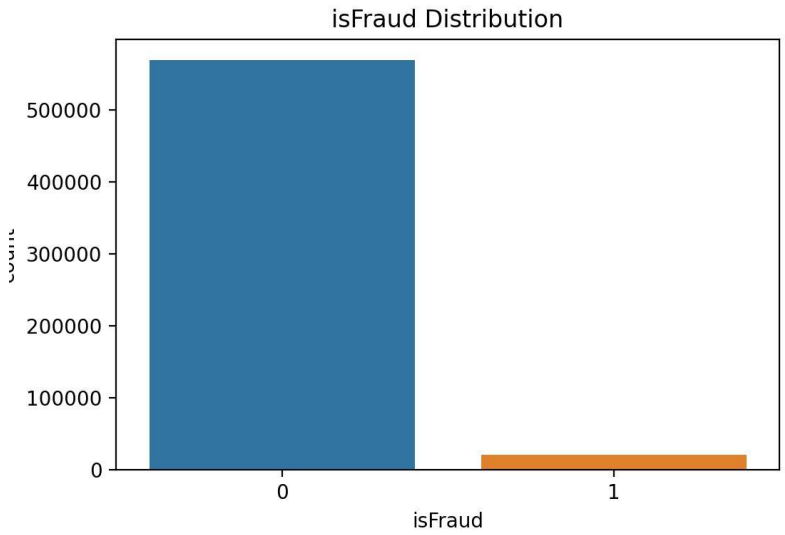
➢ Except **TransactionDT** and **TransactionAMT**, the rest of the variables are categorical features in transaction table. In identity table, **DeviceType**, **DeviceInfo** and **id_12 - id_38** are categorical features.

- **Data processing**
  - **Preventing Overfitting**
    - To prevent overfitting the train and overfitting public test dataset, we must not use client UID directly nor the dozens of columns in the dataset that help identify client like certain D, V, and ID columns.

      We cannot add UID as a new column because 68.2% of clients in private test dataset are not in the training dataset. Instead we must create aggregated group features. For example we can take all the C, M columns and do this

      new_features = df.groupby('uid')[CM_columns].agg(['mean']).

      Then we delete the column uid. Now our model has the ability to classify clients that it has never seen before.
  - **Fraudulent Clients**
    - Below is one example among 2134 clients data contain isFraud=1. For client = 2988694, the key to identifying clients are the three columns card1, addr1, and D1. The column D1 is "days since client (credit card) began". Therefore if we create D1n = TransactionDay minus D1, we get the day the card began where TransactionDay = TransactionDT / (24*60*60).




- **Model selection**
  - Lightgbm
    - Detail of the Model
      - ➢ The full name of LightGBM is a lightweight gradient boosting machine, which is a top-level Boosting algorithm framework. Like XGBoost, LightGBM is also an engineering implementation of the GBDT algorithm framework, but it is faster and more efficient. Detail of the implementation of LightGBM include four directions, histogram algorithm, unilateral gradient sampling, exclusive feature bundling algorithm and leaf-wise growth strategy [6].
    - Optimization
      - ➢ Directly supporting category features (no need to perform one-hot processing on category features), efficient parallelization and Cache (cache) hit rate optimization, etc.
  - XGboost
    - Detail of the Model
      - ➢ XGBoost is still a GBDT algorithm in essence, but it is superior to the traditional GBDT algorithm in algorithm accuracy, speed and generalization ability. From the perspective of algorithm accuracy, XGBoost expands the loss function to the second derivative to make it more close to the real loss; from the perspective of algorithm speed, XGBoost uses two techniques: weighted quantile sketch and sparse perception algorithm. Cache optimization and model parallelism are used to improve the speed of the algorithm; from the point of view of the generalization ability of the algorithm, methods such as adding a

regularization term to the loss function, setting the reduction rate and column sampling in the additive model, etc., can prevent the model from overrunning fit.

- RandomForest
  - Detail of the Model
    - ➢ Random forest (RF) is an ensemble learning algorithm designed based on the Bagging framework. Random forest is integrated with decision tree as the base classifier, and further introduces the method of randomly selecting data features in the decision tree training process. The random forest is named because of this randomness in the process of building the model [5] [7] [8].
- Multilayer Perceptron
  - Detail of the Model
    - ➢ MLP is also called artificial neural network[9]. A standard MLP is composed of input layer, hidden layer and output layer, of which there can be more than one hidden layer. The structure of an MLP model with two hidden layers [9].

**Reference:**

[1] Richard J Sullivan. The changing nature of us card payment fraud: Issues for industry and public policy. In *WEIS. Citeseer*, 2010.
[2] Gary W Adams, David R Campbell, Mary Campbell, and Michael P Rose. Fraud prevention. *The CPA Journal*, 76(1):56, 2006.
[3] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015.
[4] https://www.kaggle.com/competitions/ieee-fraud-detection/overview/description.
[5] Sotiris B Kotsiantis. Decision trees: a recent overview. *Artificial Intelligence Review*, 39 (4):261–283, 2013.
[6] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.
[7] Steven J Rigatti. Random forest. Journal of Insurance Medicine, 47(1):31–39, 2017.
[8] Mohammed Khalilia, Sounak Chakraborty, and Mihail Popescu. Predicting disease risks from highly imbalanced data using random forest. *BMC medical informatics and decision making*, 11(1):1–13, 2011.
[9] Sun-Chong Wang. Artificial neural network. *In Interdisciplinary computing in java programming*, pages 81–100. Springer, 2003.