## ****PROJECT PROPOSAL:** **CREDIT CARD FRAUD DETECTION****

**TABLE OF CONTENTS:**

* **INTRODUCTION**
* **PROBLEM DEFINITION**
* **DESIGN THINKING**
  + - * **DATA COLLECTION**
      * **DATA PREPROCESSING**
      * **FEATURE ENGINEERING**
      * **MODEL SELECTION**
      * **MODEL TRAINING**
      * **EVALUATION**

**INTRODUCTION:**

The problem is to develop a machine learning-based system for real-time credit card fraud detection. The goal is to create a solution that can accurately identify fraudulent transactions while minimizing false positives. This project involves data preprocessing, feature engineering, model selection, training, and evaluation to create a robust fraud detection system.

**PROBLEM DEFINITION:**

The mission of our project is to prevent real-time credit card fraud and take measures for it to stop.

**DATA COLLECTION:**

If you have access to a system that records transaction data, you can collect real transactions over a period of time. Ensure that the data is anonymized and follows privacy and legal guidelines.

Alternatively, you may find publicly available datasets on websites, government repositories, or data science platforms.

**Data Features:**

Ensure that your dataset includes relevant features such as transaction amount, timestamp, merchant information (name, location, category), card details (last digits, type), and any other useful attributes.

**Data Cleaning and Formatting:**

Process the raw data to remove any inconsistencies, errors, or irrelevant information.

Convert the data into a format suitable for analysis, such as a structured CSV or a database.

**Data Privacy and Security:**

Anonymize or encrypt sensitive information to comply with privacy and security regulations, particularly when dealing with card details and personal information.

**Dataset Documentation**:

Provide clear documentation explaining the dataset structure, features, and any preprocessing steps done.

**Data Collection:**

If you have access to a system that records transaction data, you can collect real transactions over a period of time. Ensure that the data is anonymized and follows privacy and legal guidelines. Alternatively, you may find publicly available datasets on websites, government repositories, or data science platforms.

**Data Features:**

Ensure that your dataset includes relevant features such as transaction amount, timestamp, merchant information (name, location, category), card details (last digits, type), and any other useful attributes.

**Data Cleaning and Formatting:**

Process the raw data to remove any inconsistencies, errors, or irrelevant information.

Convert the data into a format suitable for analysis, such as a structured CSV or a database.

**Data Privacy and Security:**

Anonymize or encrypt sensitive information to comply with privacy and security

regulations, particularly when dealing with card details and personal information.

**Dataset Documentation:**

Provide clear documentation explaining the dataset structure, features, and any preprocessing steps performed.

**DATA PREPROCESSING:**

**Data Cleaning**:

Data cleaning involves removing or correcting any inaccuracies or inconsistencies in the dataset.

**Remove Duplicates:**

Check for and remove duplicate rows, if any, to avoid redundancy.

**Handle Outliers:**

Identify and handle outliers in numerical features like transaction amount. You can remove outliers or transform them using statistical methods.

**Handle Incorrect Data:**

Check for and handle any data that doesn’t make sense or is clearly incorrect.

**2. Handling Missing Values:**

Missing values can adversely affect analysis and modelling. There are several strategies to handle them:

Imputation: Fill missing numerical values with mean, median, or mode of the respective feature. For

categorical data, you can use the most frequent value.

Deletion: Remove rows or columns with a significant number of missing values, if appropriate and feasible.

Advanced Imputation: Use more complex imputation techniques like K-nearest neighbours (KNN)

imputation for numerical data or imputing based on statistical models.

**3. Feature Normalization:**

Normalization is important to ensure that all features contribute equally to the learning process and to improve the convergence speed of the learning algorithm.

Min-Max Scaling: Scale numerical features to a specific range (e.g., [0, 1]) using the formula.

Standardization (Z-score normalization): Transform features to have a mean of 0 and a standard deviation of 1 using the formula.

Robust Scaling: Scale features using the interquartile range (IQR) to handle outliers.

**FEATURE ENGINEERING:**

Feature engineering is the process of selecting and transforming relevant features from the raw data to improve the performance of ML models. In demand prediction for drugs on pharmacies, some of the most important features are:

* **Time-based features**: These features capture trends and patterns over time. Examples include day of the week, month, year, and holidays.
* **Store-based features**: These features capture pharmacy-specific characteristics. Examples include the location of the pharmacy, the size of the pharmacy, and the customer demographics.

**Why Feature Engineering?**

Feature engineering is a critical step in building predictive models for demand forecasting. Here are some key reasons why it matters:

1. **Enhanced Model Performance:**  
   Well-engineered features can capture underlying patterns and relationships in the data, leading to more accurate predictions
2. **Improved Interpretability:**  
   Feature engineering can make your models more interpretable. By creating meaningful features, you can gain insights into which factors are driving demand and how they impact your predictions.
3. **Handling Non-linearity:**  
   Real-world demand data is often non-linear, and feature engineering allows you to transform variables to better fit the assumptions of your chosen machine learning algorithm.

Credit card fraud happens basically in two types: application fraud and transaction fraud. Application fraud is similar to identity fraud in that one person uses another person’s data to obtain a new card. Transaction fraud happens when a card is stolen or a lost card is obtained to conduct fraudulent transactions. Credit card fraud happens basically in two types: application fraud and transaction fraud. Application fraud is similar to identity fraud in that one person uses another person’s data to obtain a new card. Transaction fraud happens when a card is stolen or a lost card is obtained to conduct fraudulent transactions. Also, there has been a significant rise in counterfeit cards.

**Recent Transaction History:**

Analyze the transaction history of a card in the recent time period to identify sudden spikes or changes in transaction patterns.

**Time since Last Transaction:**

Calculate the time elapsed since the last transaction for a specific card. A sudden transaction after a long period of inactivity could be suspicious.

**Merchant Category Code (MCC) Frequency:**

Count the frequency of transactions with specific Merchant Category Codes (MCCs). Certain MCCs might be associated with a higher risk of fraud.

**Geographical Deviation:**

Calculate the geographical distance between the current transaction and the average location of the

Cardholder transactions. Unusually distant transactions might indicate fraud.

**Velocity of Transactions**:

Measure the speed at which transactions are occurring, especially if multiple transactions are happening within a short time frame. High velocity could be suspicious.

**Transaction Time Deviation:**

Calculate the deviation of the transaction time from the usual times when the cardholder makes transactions. Transactions at unusual times may raise suspicion.

**Account Age:**

Determine the age of the account based on the time since the first transaction. New accounts might be at higher risk.

**Flagging High-Risk Merchants:**

Identify merchants that have historically been associated with higher fraud rates and flag transactions with these merchants.

**Account Balance and Spending Behaviour:**

Analyze account balance and spending behaviour in relation to transaction amounts. Sudden high spending with low account balance can be suspicious.

**Card Usage Patterns:**

Analyze patterns of card usage such as predominantly online vs. in-person transactions. Sudden shifts in usage patterns can be red flags.

**Aggregation by time:**

* Average or maximum amounts spent per transaction in the past one week, two weeks, or XX weeks
* Average or maximum amounts spent per day in the past one week, two weeks, or XX weeks,
* Average or maximum amounts by merchant category in the past one week, two weeks, or XX weeks,

**Aggregation by merchant category code:**

* Average amount per day spent over 3 on all transactions up to this one on the same merchant type as this transaction
* Total number of transactions with the same merchant during the past 30 days
* The average amount spent over 1 week during the past 3 months on the same merchant type as this transaction

**Aggregation by merchant location and time:** The first two transactions in the above table happened in New York City (NYC) and Boston (BOS) within 8 minutes. It is likely the card has been compromised.

* Number of retail locations per day and the duration between the locations in the past one week, two weeks, or XX weeks,
* Minimum number of minutes between transactions of two retail locations in the past one week, two weeks, or XX weeks,

**Aggregation by transaction method:** Transactions by magnetic stripe are more prone to fraud than chip or PIN transactions. So we can create an aggregated amount by transaction type per card.

* The average amount by transaction method per day in the past one week, two weeks, or XX weeks,
* Number of transactions by transaction method per day in the past one week, two weeks, or XX weeks,

**MODEL SELECTION:**

**LOGISTIC REGRESSION:**

Credit card fraud is a critical issue for financial institutions and their customers. In modern times, credit card theft has developed a major concern for banks, as identifying fraud in the credit card system has become increasingly difficult. Machine learning plays a key role in detecting credit card fraud in transactions in order to address this challenge. Banks utilize a variety of machine learning techniques to forecast these transactions, collecting historical data and adding new variables to improve prediction capabilities. The suggested method builds the classifier using logistic regression to avoid credit card fraud. A pre-processing phase is employed to handle dirty data and ensure high detection accuracy. To clean the data, the preprocessing step employs two innovative essential strategies: the mean-based technique as well as the clustering based technique. They are frequently confused with valid approaches that compare both fraud and normal data, but this is never enough to detect fraud adequately.

In recent times, with the increasing number of credit card users the common way of transferring money i.e., credit cards are being used in wide range. These credit card services are much more efficient and economical to use. But with rapid increasing of the credit cards, it has become easy for the attackers to prey the people by sending fraud messages and asking for the OTP etc. In order to help the people to stay out of such frauds we have created a classifier which will help the people know whether the transaction is legit or fraud. The classification algorithm used here is the Logistic Regression classifier. Although there are many such algorithms available but LR always provides us with higher accuracy. It's a Machine Learning algorithm-based classification method. It's a technique for forecasting a categorized outcome variable from a set of individual factors. The logistic function's curve reflects the probability of items such as how the cells are carcinogenic or not, whether a rodent is obese or not based on its bodyweight, and so on. In logistic regression, the threshold value is used to quantify the probability of 0 or 1. More often than not, values greater than or equal to the threshold value are 1, while data rounded down equivalent to the threshold value are typically 0.

**RANDOM FOREST:**

After the model has been trained, it has to be evaluated on the test set.

Classic evaluation metrics can be used, such as sensitivity and specificity, or Cohen’s Kappa. All of these measures rely on the predictions provided by the model. In most data analytics tools, model predictions are produced based on the class with the highest probability, which in a binary classification problem is equivalent to using a default 0.5 threshold on one of the class probabilities.

However, in the case of fraud detection, we will be more conservative regarding fraudulent transactions. This means we would rather double-check a legitimate transaction and risk bothering the customer with a potentially useless call rather than miss out on a fraudulent transaction.

**MODEL TRAINING:**

**a) Validation:**

-Model validation is a phase of machine learning that quantifies the ability of an ML or statistical model to produce predictions or outputs with enough fidelity to be used reliably to achieve business objectives.

-By using a cross-validation tuning method where the training dataset is split into several equal parts, training the forecasting models with different sets of hyper-parameters. The goal of this step is to figure out which model's parameters have the most accurate forecast

-This step requires the optimization of the forecasting model parameters to achieve high performance.

-By using a cross-validation tuning method where the training dataset is split into several equal parts, data scientists train forecasting models with different sets of hyper-parameters.

-The goal of this step is to figure out which model’s parameters have the most accurate forecast.

**B) Improvement:**

-When researching the best business solutions, data scientists usually develop several machine learning models and then choose the ones that cover the project’s requirements the best. -The improvement step involves the optimization of analytic results.

-For example, using model ensemble techniques, it’s possible to reach a more accurate forecast. In that case, the accuracy is calculated by combining the results of multiple forecasting models.

**EVALUATION:**

**FI SCORE:**

F1-score is another one of the good performance metrics which leverages both precision and recall metrics. F1-score can be obtained by simply taking ‘Harmonic Mean’ of precision and recall. Unlike precision which mostly focuses on false-positive and recall which mostly focuses on false-negative, F1-score focuses on both false positive and false negative.

To explain F1-score and its use case, we shall consider “rose and jasmine flower” example i.e we have to predict if a flower is Rose or Jasmine.

**Positive —**Flower classified as a rose.

**Negative —**Flower classified as jasmine.

**False Positive (FP):** Predicted flower as a rose but in actual flower is jasmine.

**False Negative (FN):** Predicted flower as jasmine but in actual flower is rose.

## When to use F1-score:

As mentioned above, F1-score focuses on both false positive and false negative, at any cost we do not want the rose to be classified as jasmine and jasmine to be classified as a rose. In this case, we focus on both false-positive and false-negative, and try to decrease both false positive and false negative thereby increase F1-score.

**Conclusion:** Keeping the above reason in mind, we can say that our focus should be on both false positive and false negative, and try to decrease both false positive and false negative thereby increase F1-score.

**PRECISION:**

**General Definition:**Precision measures what proportion of predicted positive label is actually positive.

To explain precision and its use case, we shall consider “course recommendation” example i.e we have to recommend students to opt for xyz machine learning course.

**Positive —**Course recommended to students

**Negative**— Course not recommended to students

**True Positive (TP):** Predicted as course recommended to students and in actual also course recommended to students.

**False Positive (FP):** Predicted as course recommended to students but in actual course not recommended to students.

**False Negative (FN):** Predicted as course not recommended to students but in actual course recommended to students.

## When to use Precision?

Precision is used when we want to mostly focus on false-positive i.e to decrease false-positive value thereby increase precision value. A question might arise why we want to mostly focus on false-positive and not false-negative. To answer this question, let us consider “course recommendation” example.

**False Positive (FP):** It represents our predicted label is positive but the actual label is negative — wrongly predicted. Applying false positive on our example- it means we have predicted that course has been recommended to students but in the actual course was not recommended to students. If our false positive value is high, it clearly means we ought to miss a few or most students to recommend the course. This is going to be a loss to the institute(s) which is missing few or most students to recommend the course. So, we mostly focus on false-positive value and we try to decrease it to the least possible value.

**False Negative (FN):** It represents our predicted label is negative but the actual label is positive — wrongly predicted. Applying false negative on our example- it means we have predicted that course has not been recommended to students but in the actual course was recommended to students. If our false negative value is high, it clearly means we have to recommend a course to students who were already been recommended. It is not at all an issue if we recommend the course to students who were already been recommended. So, we don’t much focus on false negative value.

**RECALL:**

**General Definition:**Recall measures what proportion of actual positive label is correctly predicted as positive.

To explain recall and its use case, we shall consider ***‘Cancer Diagnosis’***example i.e we have to predict if a patient is diagnosed with cancer or not.

**Positive**— Patient diagnosed with cancer.

**Negative**— Patient not diagnosed with cancer.

**True Positive (TP):** Predicted as a patient diagnosed with cancer and in actual also patient diagnosed with cancer.

**False Positive (FP):** Predicted as a patient diagnosed with cancer but actual patient not diagnosed with cancer.

**False Negative (FN):** Predicted as a patient not diagnosed with cancer but actual patient diagnosed with cancer.

**When to use Recall?**

Recall is used when we want to mostly focus on false-negative i.e to decrease false negative value thereby increase recall value. A question might arise why we want to mostly focus on a false-negative and not false positive. To answer this question, let us consider “cancer diagnosis”*example.*

**False Negative (FN):** It represents our predicted label is negative but the actual label is positive — wrongly predicted. Applying false negative on our example- it means we have predicted that the patient is not diagnosed with cancer but the actual patient is diagnosed with cancer. If this is the case, patient, as per prediction might not get treatment to cure cancer. But the truth is patient is diagnosed with cancer. Our wrong negative prediction will lead to death of a patient. So, we mostly focus on false-negative value and try to decrease it to the least possible value.

**False Positive (FP):** It represents our predicted label is positive but the actual label is negative — wrongly predicted. Applying false positive on our example- it means we have predicted that the patient is diagnosed with cancer but the actual patient is not diagnosed with cancer. If this is the case, patient, as per prediction will get check-up for cancer diagnosis. To his happiness, he will come to know that he is not diagnosed with cancer. Hurrah! He is free from cancer now. So, we don’t much focus on false-positive value.