Project Proposal

Project Title: Home Credit Default Risk (HCDR)

Team Name: FP_Group_6

Abstract:

In this project, we aim to predict the probability of default for Home Credit clients based on various features from the data. Home Credit provides loans to clients but faces the challenge of evaluating the creditworthiness of clients with limited or no credit history. Our task is to use historical data from various sources to build a robust machine learning model that can accurately predict the risk of default. We will use different techniques to clean and prepare the data, create new features, and choose the best model to improve our predictions. The final model will help Home Credit make better lending decisions, reduce unpaid loans, and support financial services for people with limited access to banking.

Data Description:

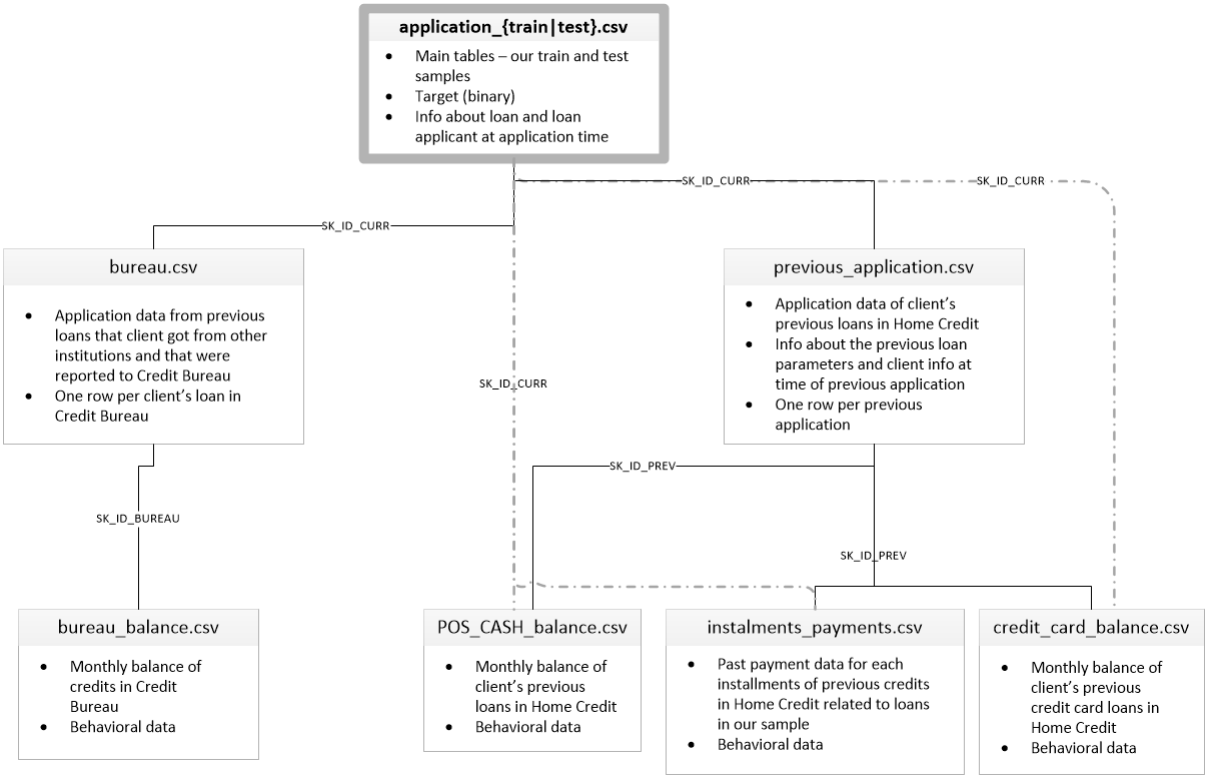
The Data is derived from a Kaggle initiative dubbed Home Credit Default Risk. Home Credit aspires to bolster financial inclusion for the unbanked populace by proffering a congenial and secure borrowing milieu. To guarantee that this marginalized community experiences a gratifying loan process, Home Credit harnesses a plethora of alternative data streams, encompassing telecommunication and transactional details, to divine their patrons' repayment proficiencies.

The contest furnishes a dataset that amalgamates seven tables and one metadata dossier delineating the columns' explications of the tables. Among these seven, the "application_{train|test}" table is esteemed as the pivotal table, as it encompasses the applications we endeavour to prognosticate. Concomitant with this fundamental table, six auxiliary tables exhibit a hierarchical association with the primary table. The descriptions for all these tables (attributed to HCDR Kaggle Competition) are elucidated below.

```
Dataset:  application_train
Rows: 307511
Columns/Features: 122
Numerical features: 106
Categorical features: 16

Dataset:  application_test
Rows: 48744
Columns/Features: 121
Numerical features: 105
Categorical features: 16

Dataset:  bureau
Rows: 1716428
Columns/Features: 17
Numerical features: 14
Categorical features: 3

Dataset:  bureau_balance
Rows: 27299925
Columns/Features: 3
Numerical features: 2
Categorical features: 1
```

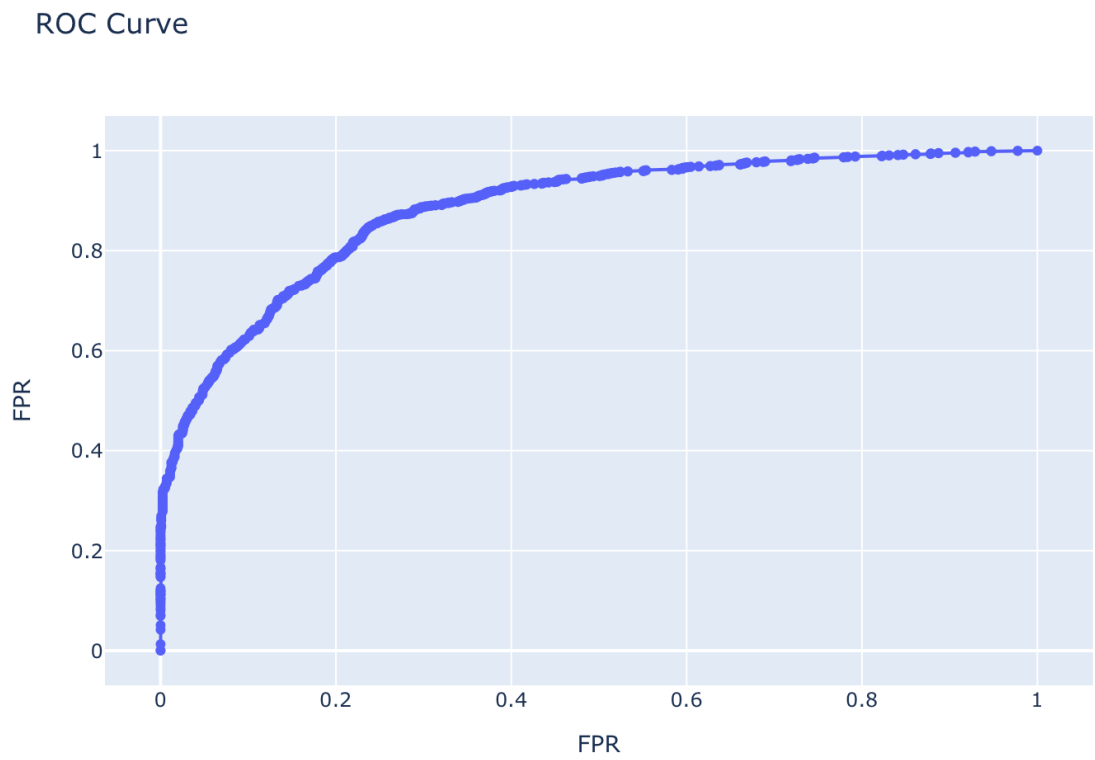
```
Dataset:  credit_card_balance
Rows: 3840312
Columns/Features: 23
Numerical features: 22
Categorical features: 1

Dataset:  previous_application
Rows: 1670214
Columns/Features: 37
Numerical features: 21
Categorical features: 16

Dataset:  installments_payments
Rows: 13605401
Columns/Features: 8
Numerical features: 8
Categorical features: 0

Dataset:  POS_CASH_balance
Rows: 10001358
Columns/Features: 8
Numerical features: 7
Categorical features: 1
```

1. **application_{train | test}.csv**: Primary dataset comprising 307,511 records and 122 attributes, detailing static information for all applicants. A binary target variable (0 for loan repayment, 1 for non-payment) is included.
2. **bureau.csv**: Dataset containing a client's previous credit records from various financial institutions, as reported to the credit bureau. Each row corresponds to a distinct credit account.
3. **bureau_balance.csv**: Data on the monthly balances of previous credits, with each row representing the balance for a specific credit account in a given month.
4. **POS_CASH_balance.csv**: Data on the monthly balances and statuses of clients' previous point-of-sale (POS) or cash loans with Home Credit. Each row corresponds to a specific loan's data for a particular month.
5. **credit_card_balance.csv**: Information on the monthly balances of clients' prior credit cards with Home Credit. Each row signifies the balance for a specific credit card during a given month.
6. **previous_application.csv**: Data containing the details of each client's past loan applications with Home Credit. Each row represents an individual application.
7. **installments_payments.csv**: Information on clients' previous loan payments with Home Credit, including records of both missed and completed payments. Each row corresponds to a specific payment.



Metrics:

1. **ROC AUC:** The primary evaluation metric used for this project is the Area Under the Receiver Operating Characteristic curve (ROC AUC). This metric captures the trade-off between true positive rate (sensitivity) and false positive rate (1-specificity), making it suitable for imbalanced classification problems like credit default prediction. While the F1 score can also be a useful metric, the ROC AUC provides a more comprehensive understanding of the classifier's performance across different decision thresholds, which is important in credit risk assessment.



This sample curve is a visual representation of the ROC curve.

2. **F1 Score:** The F1 score is another metric that can be used in this project, especially when dealing with imbalanced datasets. It is the harmonic mean of precision and recall, and ranges from 0 to 1, where 1 represents the best possible score. The F1 score balances the trade-off between precision (the proportion of true positive predictions among all positive predictions) and recall (the proportion of true positive predictions among all actual positives). This metric is useful when both false positives and false negatives are important to consider.

$$F1 = \frac{2 * true\ positives}{2 * true\ positives + false\ positives + false\ negatives}$$

3. **Balanced Accuracy:** Balanced accuracy is an alternative to the traditional accuracy metric, which may not be suitable for imbalanced datasets. Balanced accuracy is defined as the average of the true positive rate (sensitivity) and true negative rate (specificity). It is more robust to class imbalance, as it considers both the performance of the minority class (defaulters) and the majority class (non-defaulters).

Balanced Accuracy = (Sensitivity + Specificity) / 2

4. **Precision-Recall AUC (PR AUC):** The Precision-Recall Area Under the Curve (PR AUC) is another option for evaluating the performance of the model. Unlike the ROC AUC, PR AUC focuses on the positive class (defaulters) and is less sensitive to the presence of many true negatives. This metric can be more informative when dealing with imbalanced datasets, as it measures the relationship between precision and recall across different classification thresholds.
5. **Gini Coefficient (Domain-Specific Metric):** A financial industry metric measuring inequality and credit risk. Ranging from 0 (perfect equality) to 1 (complete inequality), it identifies the concentration of defaults across borrower segments. Including the Gini Coefficient in our project helps optimize the model by addressing credit risk distribution and catering to domain-specific needs.

Baseline Machine Learning Models:

The project will begin with the development of baseline models using standard machine learning algorithms such as:

1. **Logistic Regression(Lasso):** Lasso regression, also known as *Least Absolute Shrinkage and Selection Operator*, constitutes a linear regression methodology that employs regularization to contract specific coefficients to nullity, thereby diminishing the quantity of features utilized in the model. This technique proves particularly advantageous for feature extraction and mitigating the propensity for overfitting in machine learning endeavors, such as our housing default credit risk venture. As a straightforward linear model for binary classification problems, Lasso regression can be compared to other models, which serves as a good point of reference as we continue to work on developing increasingly potent models.

   We aim to use **binary cross entropy loss** function for this lasso regularization .

$$\text{LASSO\_CXE} = -\frac{1}{m}\sum_{i=1}^{m}(y_i \cdot \log\ (p_i)\ +\ (1-y_i)\cdot\log(1-p_i)) + \lambda\sum_{j=1}^{n}|\theta_j|$$

2. **Random Forest:** Random Forest is an ensemble learning method that amalgamates multiple decision trees to construct a more robust and accurate model. By leveraging the power of numerous trees, it alleviates the susceptibility to overfitting and enhances the generalization capabilities of single decision trees, making it an efficacious

approach for addressing diverse machine learning tasks, including your housing default credit risk project.As a sophisticated, non-linear model for binary classification problems, Random Forest can be juxtaposed with logistic regression, which functions as a valuable benchmark as we persist in our pursuit of devising increasingly potent models. In contrast to logistic regression, which relies on linear relationships between features and the target variable, Random Forest can capture complex, non-linear interactions, thereby providing superior predictive performance in scenarios where linear models might falter.

3. **Gradient Boosting Machines (GBM):** XGBoost, or eXtreme Gradient Boosting, represents a formidable ensemble learning methodology that successively assembles decision trees while iteratively diminishing a loss function. By capitalizing on the merits of multiple weak learners, it establishes a more resilient and precise model, efficaciously combating overfitting and augmenting generalization proficiencies. Consequently, XGBoost emerges as an exceedingly efficacious approach for tackling various machine learning conundrums, encompassing our housing default credit risk endeavor.

   As a sophisticated non-linear paradigm for binary classification quandaries, XGBoost can be juxtaposed with logistic regression, which functions as a salient benchmark as we persist in the quest to cultivate progressively potent models. In contrast to logistic regression, which relies on linear associations between features and the target variable, XGBoost discerns intricate, non-linear interconnections, delivering exceptional predictive prowess in scenarios where linear models might flounder.

4. **KNN:** K-Nearest Neighbors (KNN) is a versatile and intuitive instance-based learning algorithm that identifies patterns by examining the proximity of data points in the feature space. By contemplating the characteristics of neighboring instances, it constructs a flexible and adaptive model, effectively addressing a wide array of machine learning tasks, including our housing default credit risk project.

   By integrating KNN into our housing default credit risk project, we stand to benefit from its adaptability and ability to discern complex relationships between features, thereby fostering a more accurate and flexible model.

5. **Support Vector Machines (SVM):** Support Vector Machines (SVM) is a robust and sophisticated supervised learning algorithm that excels at constructing optimal hyperplanes to segregate data points into distinct classes. By maximizing the margin between different classes, SVM constructs a highly accurate and generalizable model, effectively addressing a wide array of machine learning tasks, including our housing default credit risk project.

   As a powerful non-linear model for binary classification problems, SVM can be juxtaposed with baseline models , which functions as a germane benchmark as we persist in our pursuit of devising progressively potent models. SVM is capable of discerning intricate, non-linear relationships through the use of kernel functions, thereby offering superior predictive performance in situations where weaker models may be inadequate.

SVM hinge loss is a pivotal aspect of the algorithm, functioning as a cost function that quantifies the misclassification penalty while striving to maximize the margin between different classes, thereby enabling the construction of a robust and accurate model.

$$\max\left(0, 1 - y_i(\vec{w} \cdot \vec{x}_i - b)\right)^2 + \lambda \sum_{j=1}^{n} |\vec{w}|.$$

6. **Artificial Neural Networks (ANN):** Here we plan to use Multilayer Perceptrons (MLP) as it represents a class of artificial neural networks characterized by their feedforward architecture, consisting of multiple layers of interconnected neurons. By employing non-linear activation functions and learning through backpropagation, MLPs can approximate complex, non-linear functions, making them a potent approach for addressing a wide array of machine learning challenges, for out housing default credit risk project. We will use cross-entropy loss as a function to be minimized in our model .

$$\text{CXE} = -\frac{1}{m} \sum_{i=1}^{m} (y_i \cdot \log\ (p_i)\ +\ (1 - y_i) \cdot \log(1 - p_i))$$

Baseline Pipeline:

1. **Data Pre-processing:** Throughout this project, we shall harness an array of machine learning algorithms facilitated by the sklearn library. To conform to the library's exigencies, data ought to be structured as a two-dimensional tableau with dimensions delineating the number of observations and features.

   We will pre-process the aforementioned septet of tables to synthesize a single consolidated table. The ensuing sub-tables will coalesce into the primary training table, application_train.

   During pre-processing, we shall transmute textual data into numeric data employing OneHotEncoder and OrdinalEncoder, tackle missing values, and concatenate multiple rows into a singular row concomitant with each parent table row. Moreover, we will compute the mean for one-hot encoded columns, epitomizing the proportion of the original row embodying each category, and segregate numeric and ordinal encoded values into a triad of columns to ascertain the central tendency and range of the initial rows.

   In order to develop the baseline pipeline, it is essential to integrate 3 distinct pre-processing transformers, as indicated by the datasets. The required transformers are:

    1. **Imputers** : Scikit-learn estimators suggest the imputation of absent values with numerical counterparts to facilitate seamless functioning and optimization through their objective functions. A rudimentary tactic for managing incomplete datasets entails discarding rows and/or columns comprising missing values; however, this jeopardizes the retention of invaluable data. A more sophisticated approach involves deducing absent values from extant ones, utilizing a univariate technique such as sklearn's SimpleImputer or a multivariate estimator like IterativeImputer. The quintessential strategy may oscillate contingent upon the dataset in question.
    2. **Encoders :**Feature encoding encompasses the metamorphosis of categorical data into numerical data, given that the preponderance of machine learning models can solely decipher numerical data as opposed to string data. To address nominal categorical features, One-hot encoders shall be deployed, engendering boolean variables corresponding to each categorical echelon. Conversely, ordinal encoding is harnessed for ordinal variables that adhere to a cogent sequence.
    3. **Scalers :**Normalization (or standardization) is often necessary for datasets, as many machine learning algorithms demonstrate improved performance or faster convergence when features are on a relatively similar scale and/or close to normally distributed. Unscaled data can lead to features with larger values having a more significant impact on model predictions than features with smaller values, simply due to differences in measurement units. To circumvent this arbitrary overweighting issue, we will scale our data .

2. **Feature Engineering:** Create new features based on domain knowledge and by combining existing features to improve model performance. one way to achieve this is by using transformer feature from sklearn.
    1. **Pipeline Composers:** In pursuit of implementing disparate transformations for numerical and categorical columns, we shall capitalize on sklearn.compose. ColumnTransformer. This empowers us to encode textual features antecedent to scaling, whilst concurrently adopting diverse imputation stratagems tailored to each data type.
3. **Feature Selection:** Identify the most relevant features using techniques such as Recursive Feature Elimination (RFE) or LASSO regularization.
4. **Model Training:** Train the baseline models using cross-validation and tune hyperparameters using grid search or random search.
5. **Model Evaluation:** Assess the models' performance using the ROC AUC metric and choose the best performing model.

Pipeline Description:

The pipeline begins with the initial phase of downloading the dataset from Kaggle. Once the dataset is successfully imported, the pipeline moves on to the subsequent phases of the data science workflow.

The Exploratory Data Analysis phase involves a comprehensive assessment of the dataset, intending to reveal underlying patterns, relationships, and trends. In this phase, visualizing and identifying patterns can help select the appropriate model and determine the necessary data pre-processing techniques. EDA entails performing tasks such as detecting normal distribution patterns and identifying outliers to clean data to enhance data training efficiency. Additionally, exploring the correlations between the target and input variables

facilitates the model's ability to understand the relationships among them. This phase also includes analysing data for imbalances, missing values, and other inconsistencies that may affect the model's accuracy.preprocessing are conducted to prepare the data for modelling as pre-processing involves preparing the data for analysis by performing various transformations such as scaling, encoding, and imputation. By scaling and encoding the data, the features are transformed to better fit the model, while imputing missing values ensures that the model can analyse all available data.

In this phase, feature selection aims to identify the most influential features to the dependent feature, enabling the model to leverage the most relevant inputs. The next phase in the pipeline is model selection, where we decide which models are best suited for the dataset. After training the models, hyperparameter tuning is performed to optimize the parameters, producing the best accuracy. In this phase, we use the Grid Search method, which involves systematically searching for the optimal values of hyperparameters for a given model. The models we plan on using are logistic regression, decision trees, random forest, KNN,SVM, neural networks, and XGBoost. It is essential to ensure that the model is neither overfitting or underfitting the data, as this can lead to inaccurate results.

After hyperparameter tuning, model ensembles are experimented with, combining multiple models to produce better performance. The goal is to produce a more accurate model than any of the individual models, by leveraging the strengths of each model to produce better results.

The final phase of the pipeline is to verify the performance of the model, calculating accuracy, recall, and F1 scores. These evaluating factors can help determine which model best fits the dataset and the problem at hand. The final model should be selected based on its ability to provide the best performance as defined in the above section of this document .

Other Planned Pipelines:

1. **Advanced Feature Engineering:** Advanced feature engineering techniques such as aggregation, interaction, and polynomial features go beyond basic feature encoding and scaling approaches. Aggregation involves grouping observations based on shared categorical features to calculate summary statistics and generate additional numerical features. Interaction features combine existing features in a linear or non-linear manner, while polynomial features create higher-order terms to capture non-linear relationships. These techniques enable the generation of new features that capture complex data patterns, facilitating more accurate and robust predictive models.
2. **Dimensionality Reduction:** Employing techniques such as PCA (Principal Component Analysis) or t-SNE (t-Distributed Stochastic Neighbour Embedding) is instrumental in reducing the number of features, while preserving the pertinent information. PCA involves transforming the initial features into a new set of uncorrelated components, ranked in descending order of the variance they explain. Subsequently, a subset of these components with the most substantial variances can be retained, resulting in a reduced number of features that account for the maximum variance. In comparison, t-SNE facilitates data visualization and dimensionality

reduction by transforming high-dimensional data points into low-dimensional counterparts, while retaining essential data relationships.

3. **Hyperparameter Optimization:** Employing more sophisticated methods such as Bayesian optimization or genetic algorithms can facilitate tuning model hyperparameters. Bayesian optimization utilizes a probabilistic model to identify the most promising hyperparameters through an iterative process of selection and refinement. By sequentially refining the hyperparameters and using the results to update the model, Bayesian optimization can achieve optimal hyperparameters with minimal experimentation. Alternatively, genetic algorithms model the search for optimal hyperparameters as an evolutionary process, mimicking the mechanisms of natural selection and reproduction. Through iterative selection and reproduction of high-performing hyperparameters, genetic algorithms can converge towards optimal hyperparameters.
4. **Ensemble Methods:** Combine different models, such as stacking or bagging, to improve overall performance.
5. **Deep Learning:** Venturing into deep learning models such as feed-forward neural networks or Long Short-Term Memory (LSTM) networks can capture intricate patterns in the data. Feed-forward neural networks involve multiple layers of interconnected nodes that transmit information in a unidirectional flow, enabling complex non-linear relationships to be modelled. In contrast, LSTM networks incorporate recurrent connections that permit information to be stored over extended periods, facilitating the modelling of temporal dependencies in time-series data. These deep learning models can detect patterns and relationships that may be elusive to traditional machine learning models, making them well-suited to handling complex data.

Gantt Chart :

Team Member Details:

1. Viswa Suhaas Penugonda- vpenugon@iu.edu
2. Sai Sumanth Muvva- saimuvva@iu.edu
3. Teja Chintha- tnchinth@iu.edu
4. Deepak Kasi Nathan- dekasi@iu.edu

Team Member Pictures:

Viswa Suhaas Penugonda

Sai Sumanth Muvva

Teja Chintha

Deepak Kasi Nathan