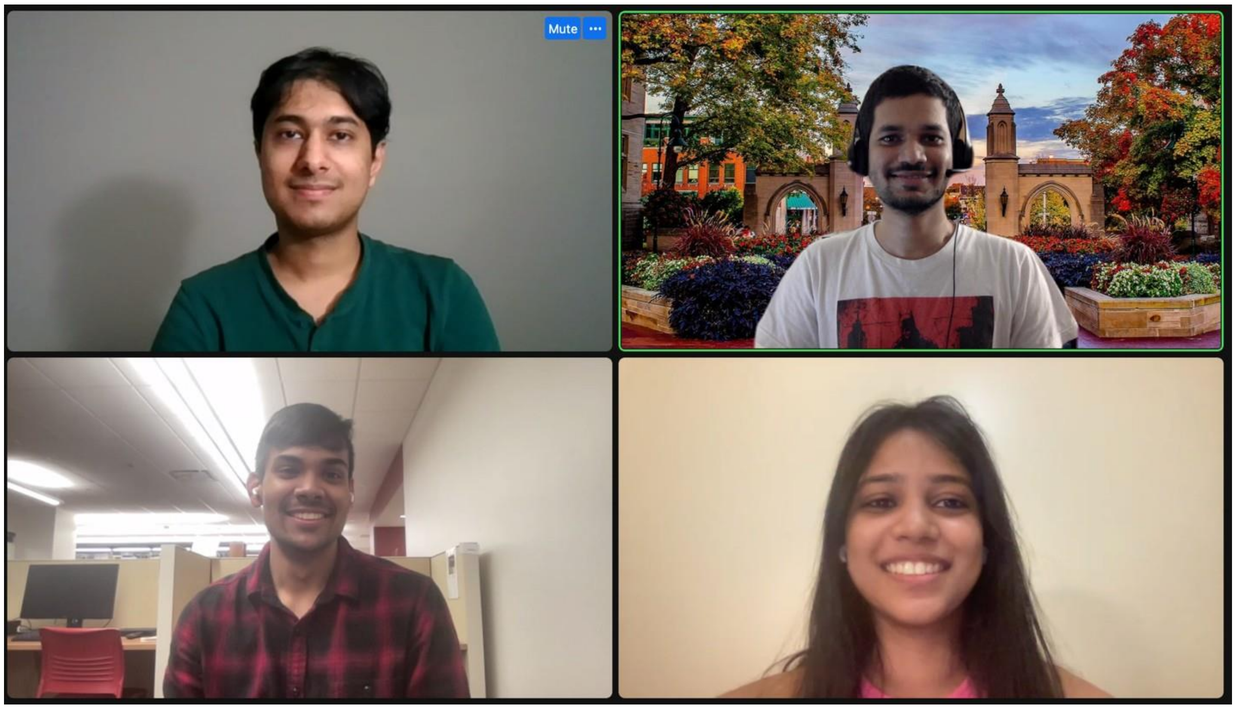
Phase 2:

Home Credit Default Risk (HCDR)- Group no. 15 Team: Data Mavericks

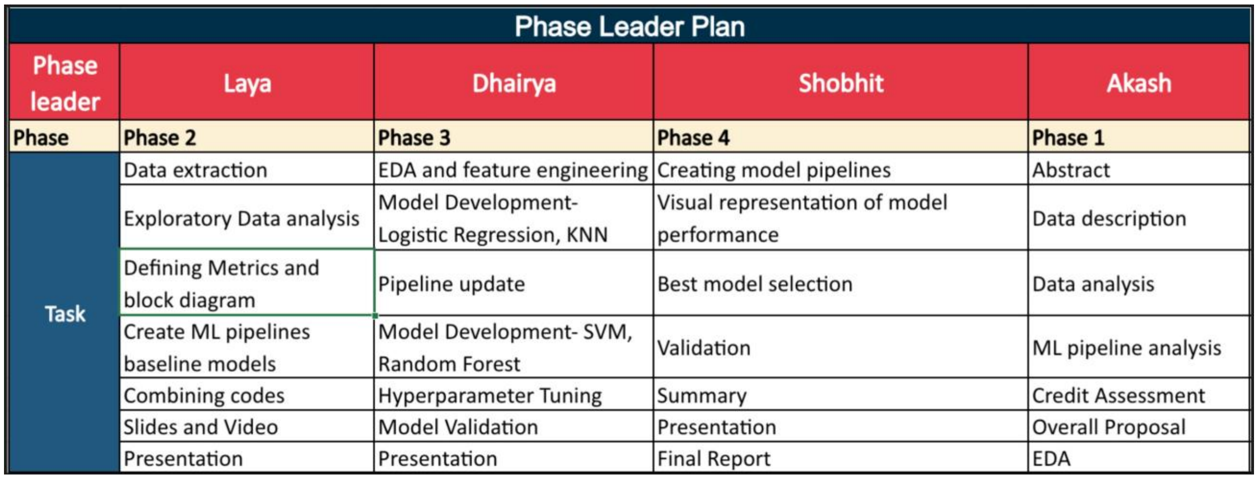
Team Members (Group Number 15)

Akash Gangadharan (akganga@iu.edu)
Laya Harwin (lharwin@iu.edu)
Dhairya Shah (shahds@iu.edu)
Shobhit Sinha(shosinha@iu.edu)

Team Photo



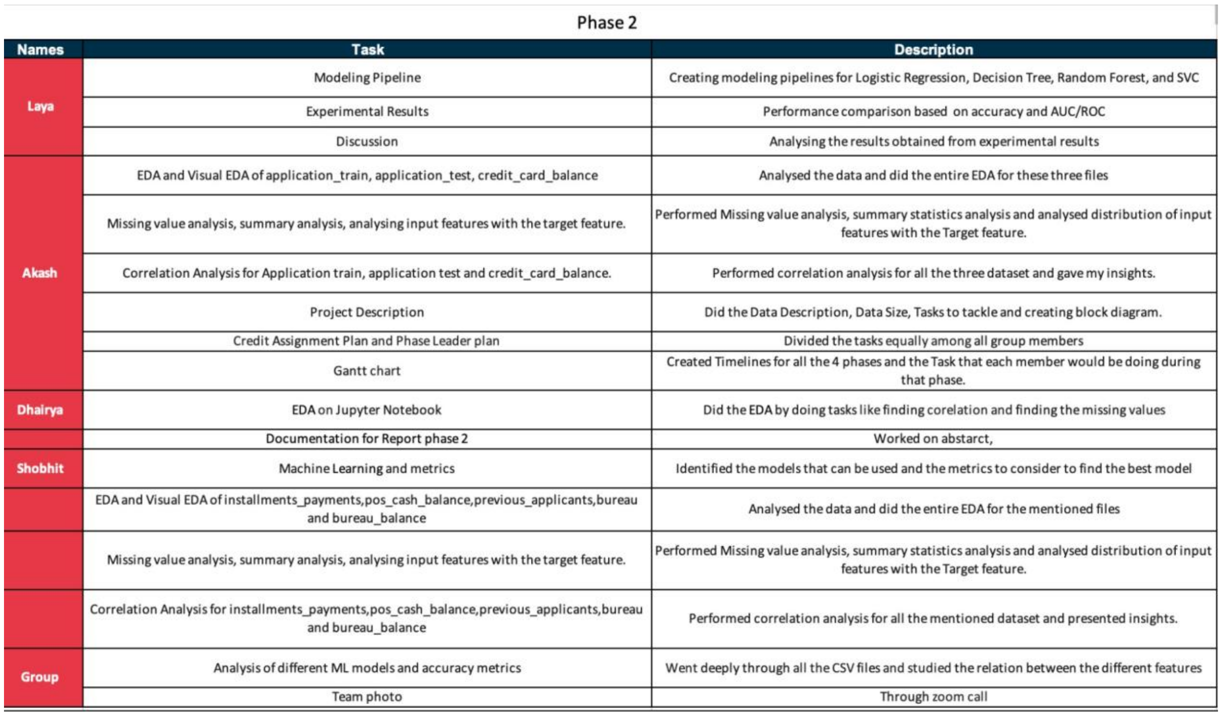
Phase Leader Plan



Phase Leader Plan

Phase leader	Laya	Dhairya	Shobhit	Akash
Phase	Phase 2	Phase 3	Phase 4	Phase 1
Task	Data extraction	EDA and feature engineering	Creating model pipelines	Abstract
	Exploratory Data analysis	Model Development- Logistic Regression, KNN	Visual representation of model performance	Data description
	Defining Metrics and block diagram	Pipeline update	Best model selection	Data analysis
	Create ML pipelines baseline models	Model Development- SVM, Random Forest	Validation	ML pipeline analysis
	Combining codes	Hyperparameter Tuning	Summary	Credit Assessment
	Slides and Video	Model Validation	Presentation	Overall Proposal
	Presentation	Presentation	Final Report	EDA

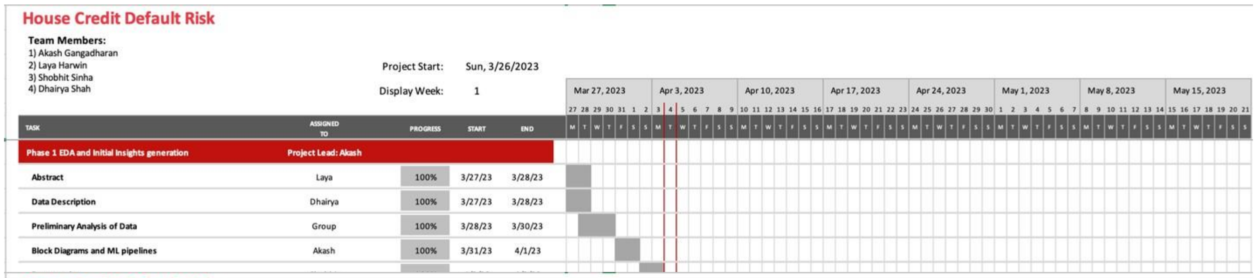
Credit Assignment Table



Phase 2

Names	Task	Description
Laya	Modeling Pipeline	Creating modeling pipelines for Logistic Regression, Decision Tree, Random Forest, and SVC
	Experimental Results	Performance comparison based on accuracy and AUC/ROC
	Discussion	Analysing the results obtained from experimental results
Akash	EDA and Visual EDA of application_train, application_test, credit_card_balance	Analysed the data and did the entire EDA for these three files
	Missing value analysis, summary analysis, analysing input features with the target feature.	Performed Missing value analysis, summary statistics analysis and analysed distribution of input features with the Target feature.
	Correlation Analysis for Application train, application test and credit_card_balance.	Performed correlation analysis for all the three dataset and gave my insights.
	Project Description	Did the Data Description, Data Size, Tasks to tackle and creating block diagram.
	Credit Assignment Plan and Phase Leader plan	Divided the tasks equally among all group members
	Gantt chart	Created Timelines for all the 4 phases and the Task that each member would be doing during that phase.
Dhairya	EDA on Jupyter Notebook	Did the EDA by doing tasks like finding corelation and finding the missing values
	Documentation for Report phase 2	Worked on abstarct,
Shobhit	Machine Learning and metrics	Identified the models that can be used and the metrics to consider to find the best model
	EDA and Visual EDA of installments_payments,pos_cash_balance,previous_applicants,bureau and bureau_balance	Analysed the data and did the entire EDA for the mentioned files
	Missing value analysis, summary analysis, analysing input features with the target feature.	Performed Missing value analysis, summary statistics analysis and analysed distribution of input features with the Target feature.
	Correlation Analysis for installments_payments,pos_cash_balance,previous_applicants,bureau and bureau_balance	Performed correlation analysis for all the mentioned dataset and presented insights.
Group	Analysis of different ML models and accuracy metrics	Went deeply through all the CSV files and studied the relation between the different features
	Team photo	Through zoom call

Gan5 chart for work distribution and project timeline



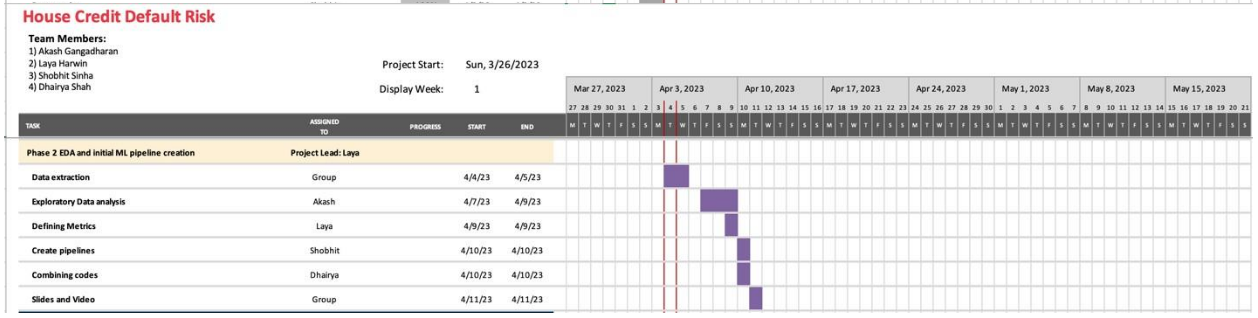
House Credit Default Risk

Team Members:
1) Akash Gangadharan
2) Laya Harwin
3) Shobhit Sinha
4) Dhairya Shah

Project Start: Sun, 3/26/2023

Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Phase 1 EDA and Initial Insights generation	**Project Lead: Akash**			
Abstract	Laya	100%	3/27/23	3/28/23
Data Description	Dhairya	100%	3/27/23	3/28/23
Preliminary Analysis of Data	Group	100%	3/28/23	3/30/23
Block Diagrams and ML pipelines	Akash	100%	3/31/23	4/1/23



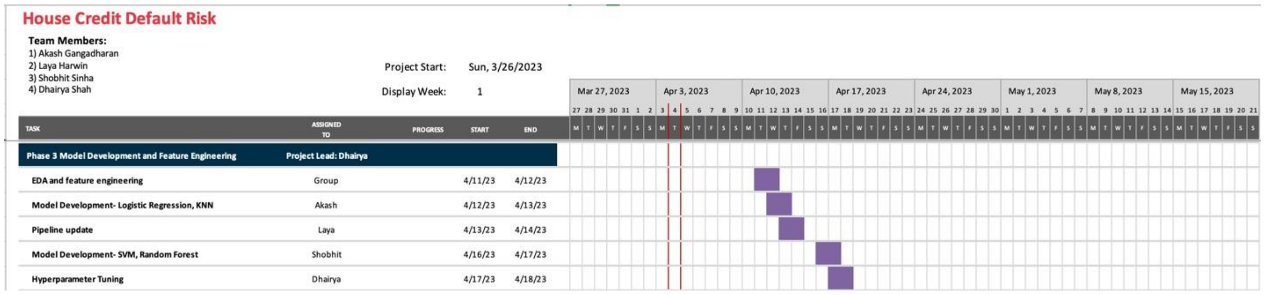
House Credit Default Risk

Team Members:
1) Akash Gangadharan
2) Laya Harwin
3) Shobhit Sinha
4) Dhairya Shah

Project Start: Sun, 3/26/2023

Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Phase 2 EDA and initial ML pipeline creation	**Project Lead: Laya**			
Data extraction	Group		4/4/23	4/5/23
Exploratory Data analysis	Akash		4/7/23	4/9/23
Defining Metrics	Laya		4/9/23	4/9/23
Create pipelines	Shobhit		4/10/23	4/10/23
Combining codes	Dhairya		4/10/23	4/10/23
Slides and Video	Group		4/11/23	4/11/23



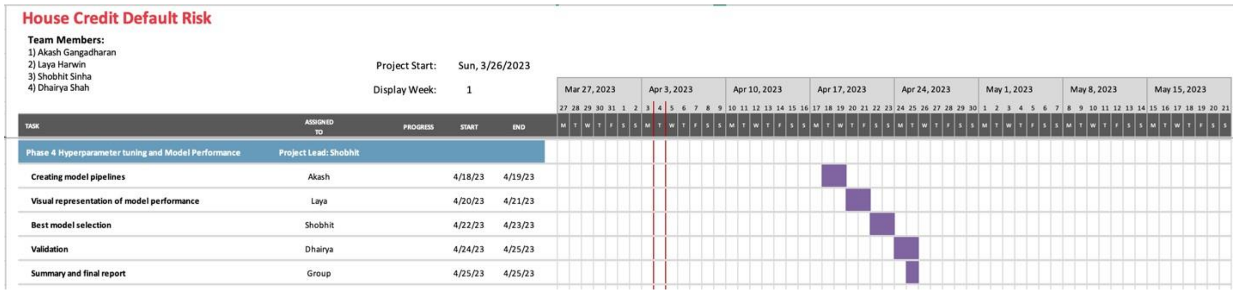
House Credit Default Risk

Team Members:
1) Akash Gangadharan
2) Laya Harwin
3) Shobhit Sinha
4) Dhairya Shah

Project Start: Sun, 3/26/2023

Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Phase 3 Model Development and Feature Engineering	**Project Lead: Dhairya**			
EDA and feature engineering	Group		4/11/23	4/12/23
Model Development- Logistic Regression, KNN	Akash		4/12/23	4/13/23
Pipeline update	Laya		4/13/23	4/14/23
Model Development- SVM, Random Forest	Shobhit		4/16/23	4/17/23
Hyperparameter Tuning	Dhairya		4/17/23	4/18/23



House Credit Default Risk

Team Members:
1) Akash Gangadharan
2) Laya Harwin
3) Shobhit Sinha
4) Dhairya Shah

Project Start: Sun, 3/26/2023

Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Phase 4 Hyperparameter tuning and Model Performance	**Project Lead: Shobhit**			
Creating model pipelines	Akash		4/18/23	4/19/23
Visual representation of model performance	Laya		4/20/23	4/21/23
Best model selection	Shobhit		4/22/23	4/23/23
Validation	Dhairya		4/24/23	4/25/23
Summary and final report	Group		4/25/23	4/25/23

Abstract:

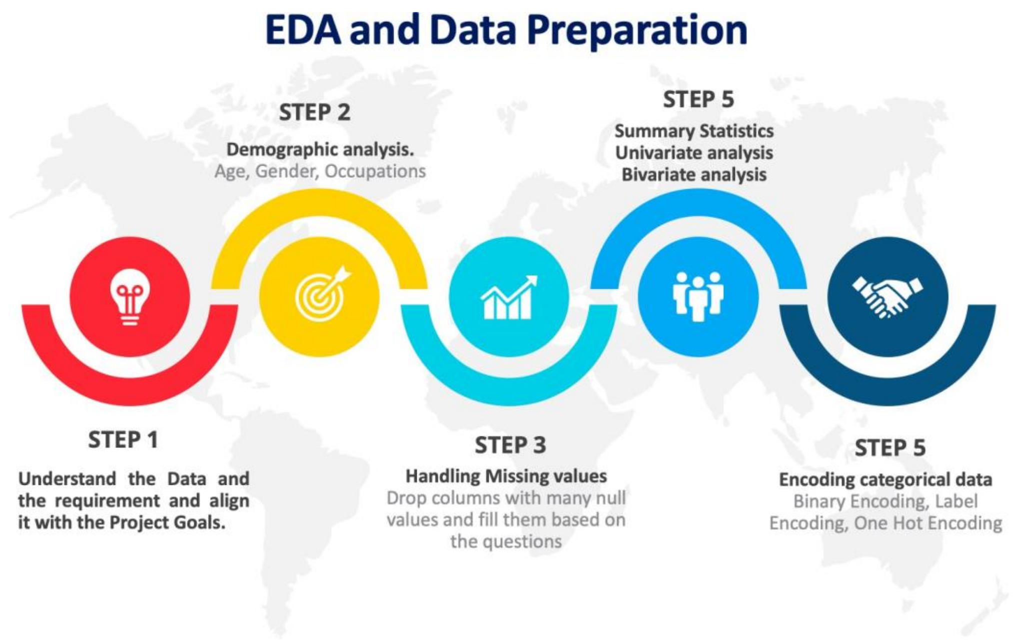
The goal of this project is to create a model to detect whether the customer can repay the loan based on the information of the background so that efficient decisions can be made. This project also aims to challenge of identifying creditworthy borrowers among those with little or no credit history. We initially plan to explore the data provided in the files and have some visual representation of it. Due to the immense number of missing values in the files we faced problems getting decent accuracy However, we plan to mitigate those problems by analyzing the missing values and then proceeding forward. We would also do a correlation analysis of the files to find whether the two files have some similarities or not. We would then get highly correlated features and apply our ML techniques only on those features. We are going to use several Machine learning algorithms like Logistic Regression, Random Forest Algorithm, Adaboost Classifier, Decision Trees etc. to assess the accuracy and finally determine the best model. From the experimental results we found out that Decision Tree has the best accuracy compared to other algorithms. Further, we plan to identify the most crucial elements of the model.

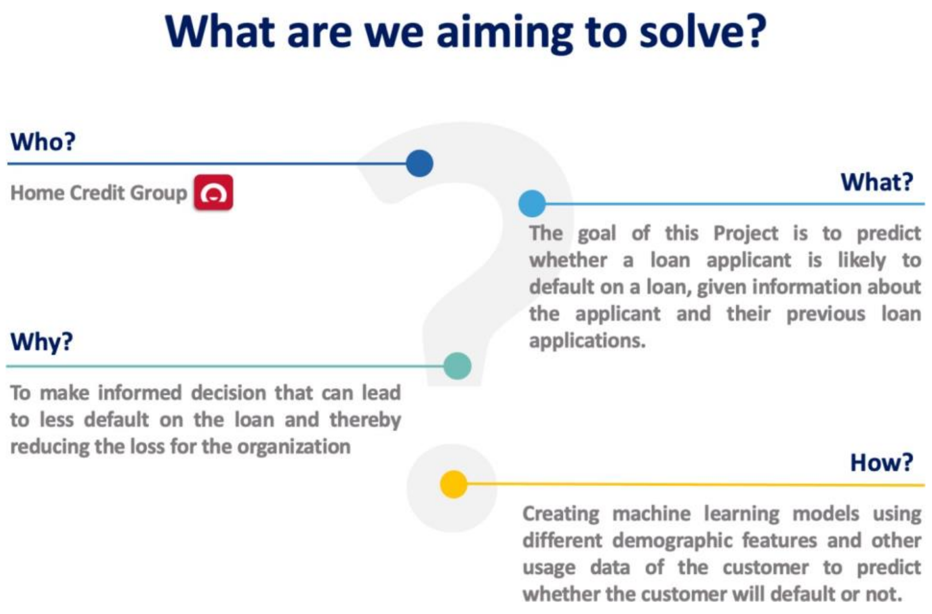
Introduction:

The Home Credit Default Risk (HCDR) dataset, which contains data pertaining to individuals' loan applications and repayment histories, is a widely used dataset in the field of credit risk analysis. The purpose of this research is to use this dataset to create a prediction model that is capable of accurately predicting whether a loan application would default on their loan. This is a critical responsibility for banks and lending organizations because loan defaults can result in large financial losses. Exploratory data analysis, data preprocessing, feature engineering, and machine learning model creation are all aspects of the project.

The ultimate goal is to create a model that can reliably anticipate loan default risk and help banks make educated lending decisions. The project also includes difficulties like as dealing with missing data, imbalanced classes, and choosing appropriate evaluation criteria. The project's findings have the potential to give lending institutions with significant insights into their risk assessment processes, allowing them to reduce the likelihood of financial losses due to loan defaults.

Project Description:


EDA and Data Preparation
STEP 2
Demographic analysis.
Age, Gender, Occupations
STEP 5
Summary Statistics
Univariate analysis
Bivariate analysis
STEP 1
Understand the Data and the requirement and align it with the Project Goals.
STEP 3
Handling Missing values
Drop columns with many null values and fill them based on the questions
STEP 5
Encoding categorical data
Binary Encoding, Label Encoding, One Hot Encoding

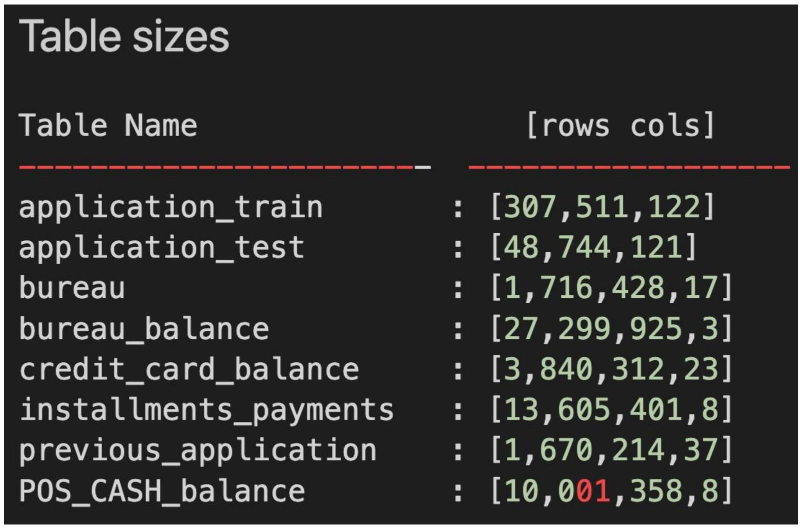

What are we aiming to solve?
Who?
Home Credit Group
What?
The goal of this Project is to predict whether a loan applicant is likely to default on a loan, given information about the applicant and their previous loan applications.
Why?
To make informed decision that can lead to less default on the loan and thereby reducing the loss for the organization
How?
Creating machine learning models using different demographic features and other usage data of the customer to predict whether the customer will default or not.

Data:

There are 7 different sources of data:
* **__application_train/application_test (307k rows, and 48k rows):__** the main training and testing data with information about each loan application at Home Credit. Every loan has its own row and is identified by the feature SK_ID_CURR. The training application data comes with the TARGET indicating **0: the loan was repaid** or **1: the loan was not repaid** . The

target variable defines if the client had payment difficulties meaning he/she had late payment more than X days on at least one of the first Y installments of the loan. Such case is marked as 1 while other all other cases as 0.
* **__bureau (1.7 Million rows):** data concerning client's previous credits from other financial institutions. Each previous credit has its own row in bureau, but one loan in the application data can have multiple previous credits.
* **__bureau_balance (27 Million rows):** monthly data about the previous credits in bureau. Each row is one month of a previous credit, and a single previous credit can have multiple rows, one for each month of the credit length.
* **__previous_application (1.6 Million rows):_** previous applications for loans at Home Credit of clients who have loans in the application data. Each current loan in the application data can have multiple previous loans. Each previous application has one row and is identified by the feature SK_ID_PREV.
* **__POS_CASH_BALANCE (10 Million rows):** monthly data about previous point of sale or cash loans clients have had with Home Credit. Each row is one month of a previous point of sale or cash loan, and a single previous loan can have many rows.
* credit_card_balance: monthly data about previous credit cards clients have had with Home Credit. Each row is one month of a credit card balance, and a single credit card can have many rows.
* **__installments_payment (13.6 Million rows):** payment history for previous loans at Home Credit. There is one row for every made payment and one row for every missed payment.

Table sizes:



```
Table sizes

Table Name                 [rows cols]
------------------------  ------------------
application_train        : [307,511,122]
application_test         : [48,744,121]
bureau                   : [1,716,428,17]
bureau_balance           : [27,299,925,3]
credit_card_balance      : [3,840,312,23]
installments_payments    : [13,605,401,8]
previous_application     : [1,670,214,37]
POS_CASH_balance         : [10,001,358,8]
```

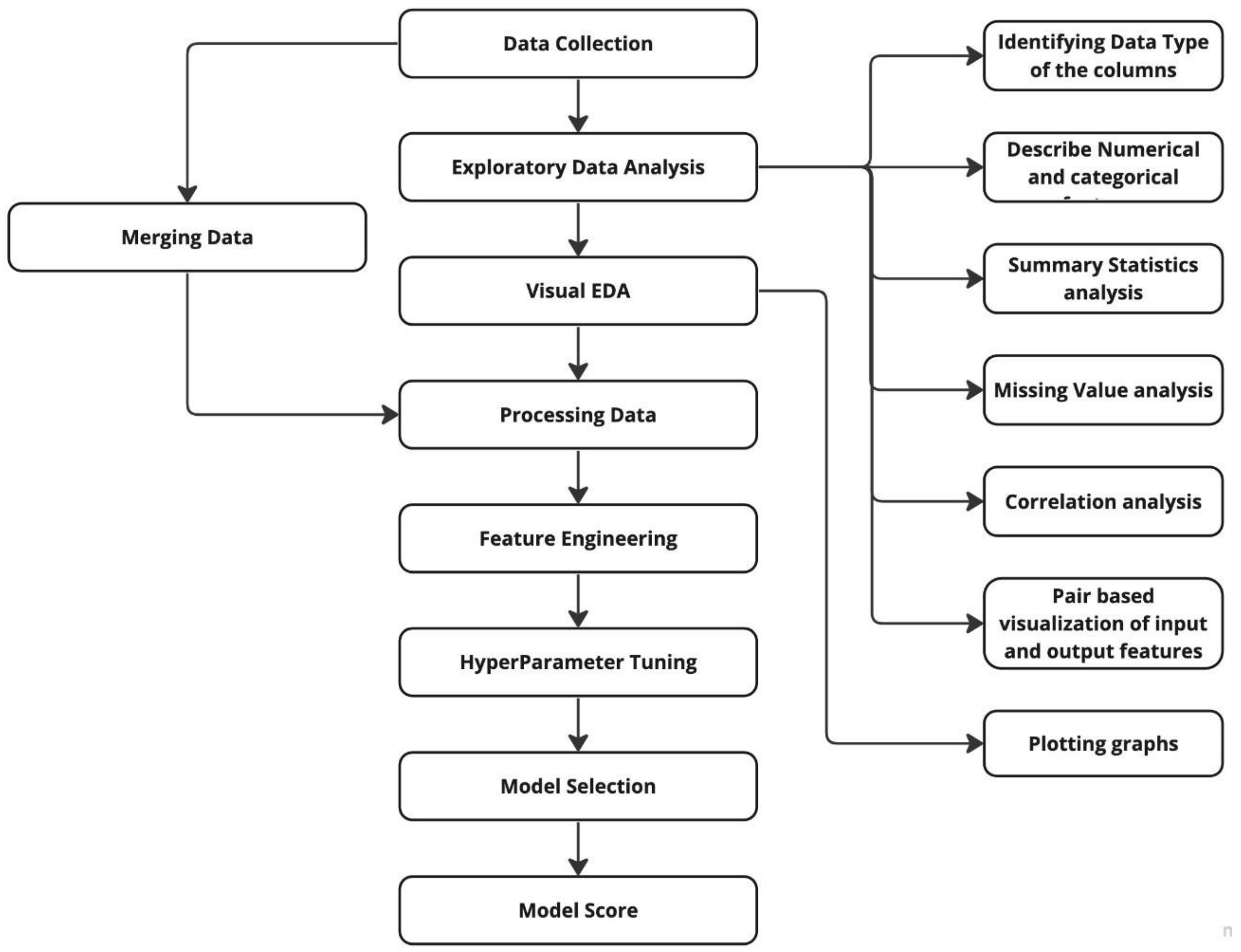
Goal:

In this phase, our goal was to EDA and Visual EDA to understand the data better and experiment with different Baseline Machine learning models and find which algorithms perform best in predicting the output.

Task to betackled:

Understand different columns and divide them into numerical and categorical features and how they correlate with thetarget.

ML Pipeline:


Data Collection
Exploratory Data Analysis
Merging Data
Visual EDA
Processing Data
Feature Engineering
HyperParameter Tuning
Model Selection
Model Score
Identifying Data Type of the columns
Describe Numerical and categorical features
Summary Statistics analysis
Missing Value analysis
Correlation analysis
Pair based visualization of input and output features
Plotting graphs

Data Collection: This step involves collecting the data.
Exploratory Data Analysis: This step involves Identifying data types of the columns, discerning the numerical and categorical features, summary statistics analysis, missing value analysis, and correlation analysis.
Visual EDA: It contains all the visual part of the EDA we just did above and generate insights.
Processing Data: Here we are merging the data from all the files.

Feature Engineering: AXer we have merged the data, we will try to find the correlation and join two highly correlated features to create one feature using the dimensionality reduction technique.
Hyper Parameter Tuning: We will try to tune the parameters of the selected ML model to get the best parameter value that predicts the output well.
Then we will select the best model and get the scores.

EDA: (yet to be completed)

AXer performing the EDA, we have got some important columns that has the highest affect on the Target variable. It means that these columns could predict the outcome instead of using all the columns

EXT_SOURCE_1, EXT_SOURCE_2, and EXT_SOURCE_3: These columns are moderately correlated with the target variable and also moderately correlated with each other, so they may be good features to include in amodel.

DAYS_BIRTH: This column is moderately negatively correlated with the target variable, indicating that younger applicants are more likely to default on their loans.

DAYS_EMPLOYED: This column has a weak negative correlation with the target variable, indicating that longer-term employment may be associated with a lower risk of default.

AMT_CREDIT: This column has a moderate positive correlation with the target variable, indicating that higher loan amounts may be associated with a higher risk of default.

AMT_GOODS_PRICE: This column has a moderate positive correlation with the target variable, similar to AMT_CREDIT.

Loss Function:

Logistic Regression: Logarithmic Loss Function

$$-\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

Binary Cross-Entropy / Log Loss

Random Forest Classifier: Gini Impurity
Decision Tree: Splitting Criterion used in Gini Impunity

$$Gini = 1 - \sum_{i=1}^{n} (p_i)^2$$

AdaBoost Classifier: Exponential Loss Function

$$L(y, f(x)) = \exp(-yf(x))$$

Baseline Models:

- We established a preparation pipeline for the HCDR dataset using scikit-learn's ColumnTransformer. Both category and numerical features are likely included in the dataset, thus the preparation pipeline must treat them separately.
- To change categorical and numerical information, the algorithm first generates two distinct pipelines. The pipeline applies one-hot encoding to categorical features before using SimpleImputer to fill in missing values with the value that occurs the most frequently. The pipeline uses StandardScaler to scale the data for numerical characteristics and SimpleImputer to replace missing values with the median value.
- The appropriate pipeline is then applied to each type of feature using the ColumnTransformer that was established before. Categorical_trans pipeline is applied to categorical characteristics by the "cat" transformer, whereas the numerical_trans pipeline is applied to numerical features by the "num" transformer.
- The ColumnTransformer object is added to the parameter of the transformers to establish the preprocessing pipeline. The data can then be preprocessed using the resulting pp_pipe object before being fitted to a machine-learning model.
- Overall, by applying the proper transformations to each type of feature, this code is building a preprocessing pipeline that can handle both categorical and numerical characteristics in the HCDR dataset. The pipeline can assist in making sure that the data is appropriately prepared for modeling and can increase the precision of machine learning models developed using the dataset.

Models:

We implemented the following models to find which one gives the best performance:

1. Logistic Regression: A common model for situations involving binary classification is logistic regression. It operates by utilizing a logistic function to estimate the likelihood of the target variable (default) with respect to the input parameters (such as age, income, and loan amount). Based on the applicant's demographic and financial data, logistic regression can be applied in the HCDR dataset to forecast the likelihood of default.

2. Random Forest Classifier: Several Decision Trees are used in an ensemble model called Random Forest to lessen overfitting and boost accuracy. It operates by randomly choosing a subset of features at each split while training a series of Decision Trees on bootstrapped subsets of the data.

3. Decision Tree Classifier: A decision tree model divides the data recursively into subsets depending on the most significant attributes until a stopping requirement is satisfied. As a result, a tree-like model is produced, with each internal node standing in for a judgment call based on a feature and each leaf node for a prediction.

4. Adaboost Classifier: A popular machine learning approach for categorization issues is called Adaboost (Adaptive Boosting). It functions by fusing weak classifiers to produce a powerful classifier.

Logistic_Regression:

- For Logistic Regression we set up a hyperparameter grid to be searched through during model selection and created a scikit-learn pipeline for a Logistic Regression model.
- The LogisticRegression() object wrapped in an estimator with the identifier "lr" is the only step used to generate the Pipeline object. The LogisticRegression() object's hyperparameters are specified in the hyperparameter grid using the term 'lr'.
- The Logistic_Regressionparams dictionary is used to specify the hyperparameter grid. The Logistic Regression model's "C" hyperparameter, which determines the regularization strength, is set to the value 0.01. The "penalty" hyperparameter, which has the values "l1" and "l2" in a list, regulates the kind of regularization the model employs. These variables indicate whether L1 regularization or L2 regularization should be used to fit the model.
- GridSearchCV or another hyperparameter tuning method can use the hyperparameter grid to find the ideal set of hyperparameters for the Logistic Regression model. The L1 and L2 regularization may be compared during the search thanks to this hyperparameter grid configuration, and the value of the 'C' hyperparameter regulates the regularization's strength.
- In general, we created a pipeline and hyperparameter grid that can be used for model selection, hyperparameter tuning, and a Logistic Regression model on the HCDR dataset.

Decision Tree:

- In addition to putting up a hyperparameter grid to be searched through during model selection, we created a scikit-learn Pipeline for a Decision Tree model.
- The DecisionTreeClassifier() object wrapped in an estimator with the name "dt" is the only step required to generate the Pipeline object. The DecisionTreeClassifier() object's hyperparameters are specified in the hyperparameter grid using the term 'dt'.
- The hyperparameter grid is specified as a dictionary named Decision_Treeparams. The maximum depth of the Decision Tree is controlled by the'max_depth' hyperparameter, which has the values 5 and 10 in its list of possible values. These numbers indicate that a maximum depth of 5 or 10 should be used to fit the model.

- The maximum depth hyperparameter for the Decision Tree model can be tuned using GridSearchCV or any hyperparameter tuning method with the help of the hyperparameter grid. The code enables for the performance of the Decision Tree model to be assessed at various maximum depths and regulates the model's complexity through the maximum depth by adjusting the hyperparameter grid in this manner.
- For a Decision Tree model on the HCDR dataset, we created a Pipeline and hyperparameter grid that can be used for hyperparameter tuning and model selection.

AdaBoost classifier:

- The Pipeline function of Scikit-Learn is used to build an AdaBoost classifier. An ensemble classifier known as the AdaBoostClassifier combines a number of "weak" classifiers to produce a single, stronger classifier. In this instance, a decision tree classifier is the basic estimator.
- Hyperparameters that are provided to the AdaBoostClassifier during training are contained in the AdaBoost_Classifierparams dictionary. Scikit-learn will try all possible combinations of the hyperparameters to identify the combination that yields the best performance on the dataset. These hyperparameters are supplied in a grid-search format.
- The hyperparameters in this instance are configured as follows:

ada_n_estimators: The number of weak classifiers to be included in the ensemble is specified by this hyperparameter. The value is set to 20 in this instance.
ada_learning_rate: This parameter regulates how much each weak classifier contributes to the final ensemble. Each weak classifier has a bigger effect on the result when the learning rate is increased. In this instance, 0.01 and 0.1 learning rates are attempted.
ada_base_estimator_max_depth: The base estimator's decision tree's maximum depth is controlled by this hyperparameter. A decision tree that is less complicated and not as prone to overfit the training set has less depth. In this instance, the maximum depths of 1 and 5 are tested.

- Scikit-learn will fit the AdaBoost_Classifier pipeline using the chosen parameters on the training data. The final model is going to be used to forecast the probability of loan default on a holdout set, and the algorithm's performance may be assessed using measures.

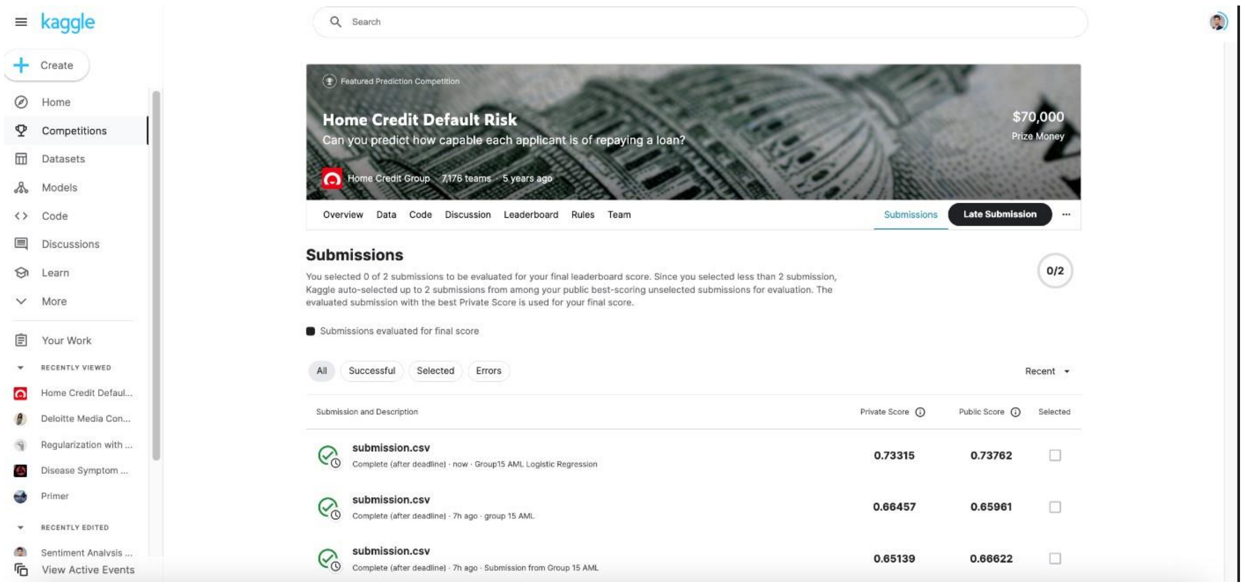
Random Forest

- We set up a hyperparameter grid to be searched through during model selection and created a scikit-learn Pipeline for a Random Forest model.
- The RandomForestClassifier() object is wrapped in an estimator with the name "rf" to generate the Pipeline object in a single step. The RandomForestClassifier() object's hyperparameters are specified in the hyperparameter grid using the name 'rf'.
- The Random_Forestparam dictionary is the one used to specify the hyperparameter grid. The 'n_estimators' hyperparameter, which is specified as 20, regulates the number of decision trees in the Random Forest model. These numbers indicate that 20 decision trees should be used to fit the model. Each decision tree's "criterion" hyperparameter, which is set to the values "gini" and "log_loss," regulates the quality of the split. These numbers indicate that the decision trees' node splitting criteria should be either Gini impurity or cross-entropy in order to fit the model.

- The hyperparameter grid is made to make it possible for GridSearchCV or any hyperparameter tuning technique to look for the ideal set of hyperparameters for the Random Forest model. The code allows for the comparison of the number of decision trees and the quality of the splits in the decision trees throughout the search and manages the complexity and performance of the model using these hyperparameters by configuring the hyperparameter grid in this manner.
- Overall, we created a Pipeline and hyperparameter grid for a Random Forest model on the HCDR dataset that can be used for hyperparameter tuning and model selection.

Experimental results:

We got the following scores aXer Kaggle Submission:



Home Credit Default Risk

Can you predict how capable each applicant is of repaying a loan?

Home Credit Group · 7,176 teams · 5 years ago

$70,000
Prize Money

Submissions

You selected 0 of 2 submissions to be evaluated for your final leaderboard score. Since you selected less than 2 submission, Kaggle auto-selected up to 2 submissions from among your public best-scoring unselected submissions for evaluation. The evaluated submission with the best Private Score is used for your final score.

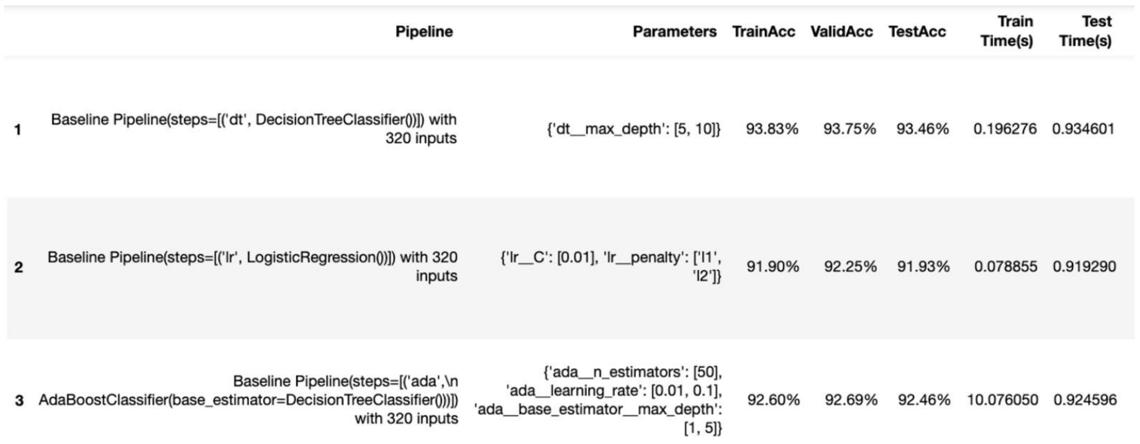
0/2

■ Submissions evaluated for final score

All · Successful · Selected · Errors · Recent

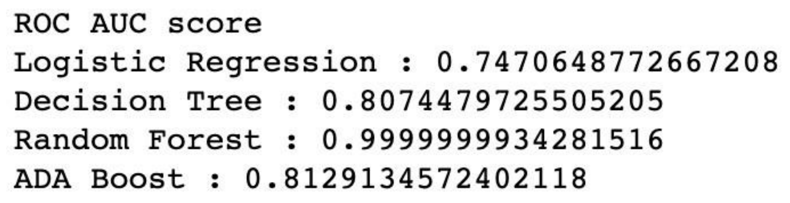
Submission and Description	Private Score	Public Score	Selected
submission.csv Complete (after deadline) · now · Group15 AML Logistic Regression	0.73315	0.73762	☐
submission.csv Complete (after deadline) · 7h ago · group 15 AML	0.66457	0.65961	☐
submission.csv Complete (after deadline) · 7h ago · Submission from Group 15 AML	0.65139	0.66622	☐

Here, are the performance details for all the models we implemented:



	Pipeline	Parameters	TrainAcc	ValidAcc	TestAcc	Train Time(s)	Test Time(s)
1	Baseline Pipeline(steps=[('dt', DecisionTreeClassifier())]) with 320 inputs	{'dt__max_depth': [5, 10]}	93.83%	93.75%	93.46%	0.196276	0.934601
2	Baseline Pipeline(steps=[('lr', LogisticRegression())]) with 320 inputs	{'lr__C': [0.01], 'lr__penalty': ['l1', 'l2']}	91.90%	92.25%	91.93%	0.078855	0.919290
3	Baseline Pipeline(steps=[('ada',\n AdaBoostClassifier(base_estimator=DecisionTreeClassifier()))]) with 320 inputs	{'ada__n_estimators': [50], 'ada__learning_rate': [0.01, 0.1], 'ada__base_estimator__max_depth': [1, 5]}	92.60%	92.69%	92.46%	10.076050	0.924596

Following are the results we obtained for the ROC AUC score:



```
ROC AUC score
Logistic Regression : 0.7470648772667208
Decision Tree : 0.8074479725505205
Random Forest : 0.9999999934281516
ADA Boost : 0.8129134572402118
```

Discussion:

In Phase 1, we implemented 4 algorithms and out of the four algorithms, we got the maximum test accuracy for the Decision Tree of 93.46%. Training accuracy is also highest for the Decision Tree at 93.83%. Similarly, validation accuracy is also highest at 93.75%. The next best algorithm is the AdaBoost algorithm with a test accuracy of 92.46%, validation accuracy of 92.69%, and train accuracy of 92.60%. Furthermore, in the next phases, we will do feature engineering and hyperparameter tuning to improve the performance of models and we will also be experimenting with more models.
With regard to the ROC AUC score, we found that the Random Forest algorithm has the highest score which means that the model translates to a higher ability of the model to distinguish between the positive and negative classifications. The next best score is for ADA Boost followed by Random Forest and finally logistic regression. In the next phase, we will be adding feature engineering and hyperparameter tuning to improve the performance of the model.

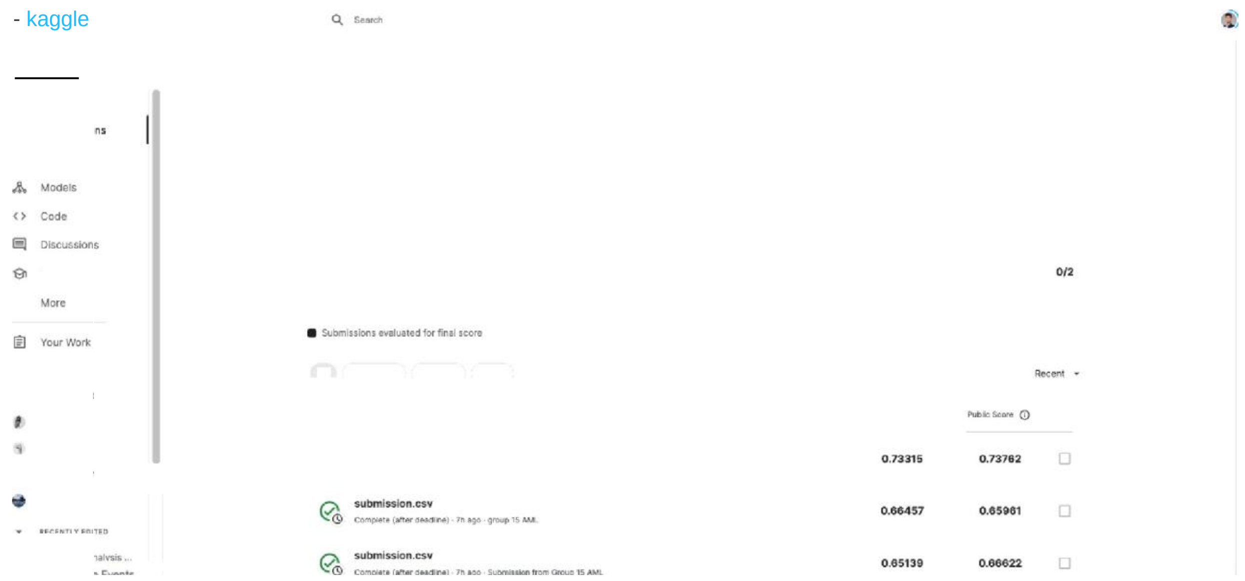
Conclusion:

Nowadays it is very important to identify whether a person can repay a loan or not. To accomplish this task, we designed a Machine Learning model to determine whether the person can pay back the loan or not. The crucial part of we worked on this phase is Exploratory Data Analysis and visual representation of the data in the files. We did the analysis on the target which includes implementing new features or revamping the existing features, understanding the missing data, and visualizing outliers. We experimented with different models to simplify the data. The analysis of our model includes the new features and identifying which model performs better on the given data. We found out that the Decision Tree model has the highest accuracy compared to the rest of the models we implemented. In the future we plan to refine the existing models by using feature engineering and hyperparameter tuning and test with various other models to determine which model works best so we can provide better accuracy. So, we plan to create a process that is feasible for both Borrowers and lenders.

Kaggle Submission

The screenshot should show the different details of the submission and not just the score.

- kaggle



0/2

■ Submissions evaluated for final score

		Public Score	
	0.73315	0.73762	☐
submission.csv Complete (after deadline) · 7h ago · group 15 AML	0.66457	0.65961	☐
submission.csv Complete (after deadline) · 7h ago · Submission from Group 15 AML	0.65139	0.66622	☐