**Home** **Credit Default Risk** (HCDR)

**Group 11**

Anuj Mahajan

Siddhant Patil

Shashwati Diware

Shubham Jambhale

**Team Members:**

Shubham Jambhale Siddhant Patil

[sjambhal@iu.edu](mailto:sjambhal@iu.edu) [sidpatil@iu.edu](mailto:sidpatil@iu.edu)

A person standing in front of a tree

Description automatically generated with medium confidence A person sitting on a red couch

Description automatically generated with low confidence

Anuj Mahajan Shashwati Diware

[anujmaha@iu.edu](mailto:anujmaha@iu.edu) [sdiware@iu.edu](mailto:sdiware@iu.edu)

A person wearing sunglasses and standing in front of a body of water

Description automatically generated with medium confidence A person standing in front of a building

Description automatically generated with medium confidence

**Phase Leader Plan**

|  |  |  |
| --- | --- | --- |
| Phase | Contributor | Contribution Description |
| Phase 1: Project Planning | Anuj Mahajan | Download Data, go through data, and load libraries. Create a pipeline diagram and describe the pipeline design. Describe Preprocessing, |
| Phase 1: Project Planning | Shashwati Diware | Project Abstract, ML Algorithm Names, and describe Metrics. |
| Phase 1: Project Planning | Shubham Jambhale(Phase Leader) | Understanding the problem statement, and writing table descriptions. Schedule meetings, coordinate tasks, plan phase |
| Phase 1: Project Planning | Siddhant Patil | Machine Learning Pipeline Steps and describes pipeline components. |
| Phase 2: Base Line Modelling and EDA | Anuj Mahajan (Phase Leader) | Creating Block Diagram EDA and one slide of the presentation. Schedule meetings, coordinate tasks, plan phase |
| Phase 2: Base Line Modelling and EDA | Shashwati Diware | Result Analysis EDA and one slide of the presentation. |
| Phase 2: Base Line Modelling and EDA | Shubham jambhale | Result Analysis and two slides of the presentation |
| Phase 2: Base Line Modelling and EDA | Siddhant  Patil | Result Analysis and two slides of the presentation |
| Phase 3: Hyperparameter Tuning | Shashwati  Diware (Phase Leader) | Testing Accuracy matrix and Schedule meetings, coordinating tasks, the planning phase |
| Phase 3: Hyperparameter Tuning | Siddhant Patil | Create and develop code for Hyperparameter tuning |
| Phase 3: Hyperparameter Tuning | Shubham Jambhale | Run and create analysis by testing the confusion / AUC matrix. Coordinate Tasks and one slide of the presentation |
| Phase 3: Hyperparameter Tuning | Anuj Mahajan | Run and analyze Lasso and ridge regression losses. Coordinate tasks and one slide of the presentation |
| Phase 4: Final Report Generation | Siddhant Patil (Phase Leader) | Plan Phase Schedule Meetings and Coordinate Tasks, analyze and go through the final results |
| Phase 4: Final Report Generation | Anuj Mahajan | Rearrange everything and go through the final documentation, list down the final recordings |
| Phase 4: Final Report Generation | Shashwati Diware | Prepare the final presentation |
| Phase 4: Final Report Generation | Shubham Jambhale | Check everything and submit the assignment before the deadline |

**Credit Assignment Plan**

**Phase 1:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task | Task Description | Hours spent | Assigned to | Start | End |
| Understanding problem statement | Go through the problem statement to understand the requirements | 6 | Shubham | 11/05/22 | 11/07/22 |
| Data Exploration | Explore and analyze the data for a better understanding | 6 | Anuj | 11/07/22 | 11/09/22 |
| Project Proposal | Creating the project proposal and preparing a basic report with Abstract, ML models, and Gantt diagram | 20 | Group | 11/09/22 | 11/14/22 |

**Phase 2:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task | Task Description | Hours Spent | Assigned to | Start | End |
| Creating Block Diagram | Creating the block diagram of the basic flow of execution. | 5 | Anuj | 11/13/22 | 11/15/22 |
| Creating Pipeline Diagram | Creating the pipeline diagram of the machine learning model from analyzing the data till the result analysis | 5 | Shashwati | 11/13/22 | 11/15/22 |
| Result Analysis | Analyzing the Result | 10 | Group | 11/26/22 | 11/29/22 |
| PowerPoint Presentation | Simultaneously prepare the PowerPoint presentation and add the analyzed data into it as per need | 10 | Group | 11/20/22 | 11/29/22 |

**Phase 3:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task | Task Description | Hours spent | Assigned to | Start | End |
| Create and develop code for hyperparameter tuning | Design and develop python helper function for hyperparameter tuning | 16 | Siddhant | 11/20/22 | 11/25/22 |
| Result Analysis | Analysis of Obtained Result | 2 | Group | 12/02/22 | 12/03/22 |
| Testing Accuracy matrix | Analyzing accuracy using accuracy matrix | 2 | Shashwati | 12/03/22 | 12/04/22 |
| Analyzing the Loss and AUC | Analyzing Loss and AUC matrix | 2 | Shubham | 12/03/22 | 12/04/22 |
| Creating Powerpoint presentation | Simultaneously prepare the PowerPoint presentation and add the analyzed data into it as per need | 2 | Anuj | 12/03/22 | 12/04/22 |

**Phase 4:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task | Task Description | Hours Spent | Assigned To | Start | End |
| Build MLP using Pytorch | Build a neural network model using Pytorch | 10 | Group | 12/03/22 | 12/08/22 |
| Final Documentation | Rearrange everything and go through the final documentation, list down the final recordings | 10 | Anuj | 12/03/22 | 12/08/22 |
| Final Results | Analyze final results obtained after the final testing | 6 | Siddhant | 12/05/22 | 12/08/22 |
| Final Presentation | Prepare the final presentation | 4 | Shashwati | 12/06/22 | 12/08/22 |
| Assignment Submission | Check everything and submit the assignment before the deadline | 1 | Shubham | 12/08/22 | 12/09/22 |

**Abstract**

Based on historical credit histories and repayment trends utilizing machine learning modeling, Home Credit offers unsecured lending. A user-generated credit score is calculated using criteria like the balance that the user has maintained. As part of this project, we are predicting the customer repayment status such as if the user is a defaulter or not using machine learning pipelines and models using the datasets provided by Kaggle. The data collection includes seven separate tables that aid in determining the user status, including bureau balance, credit card balance, home credit column detection, Installments payments, POS CASH balance, and previous applications. In phase 3, we provide feature engineering, hyperparameter tuning, and modeling pipelines. We experimented with selected features for Logistic regression, Decision Making Tree, Random Forest, Lasso, and Ridge Regressions. The Decision Tree has the highest test accuracy with 92.12, followed by Logistic regression and Random Forest with a test accuracy of 91.98. We received **0.5** ROC AUC from a Kaggle submission.

**Data and Task Description**

***Data source***

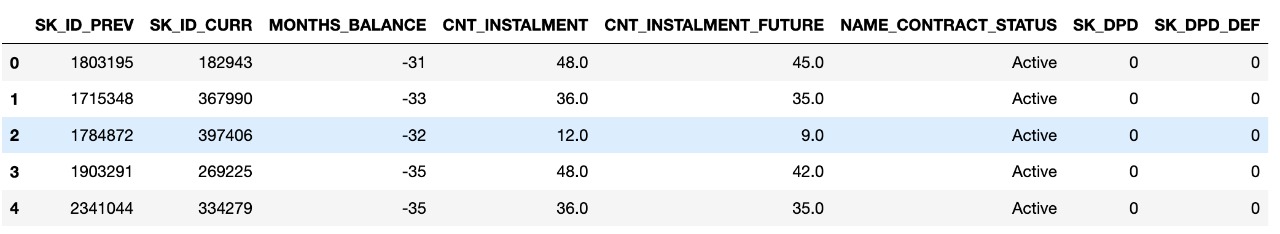
We are planning to use the existing datasets provided by Kaggle.

Source: <https://www.kaggle.com/c/home-credit-default-risk/data>

***POS\_CASH\_balance.csv***

This dataset gives information about previous credit information such as contract status, the number of installments left to pay, DPD(days past due), etc. of the current application.

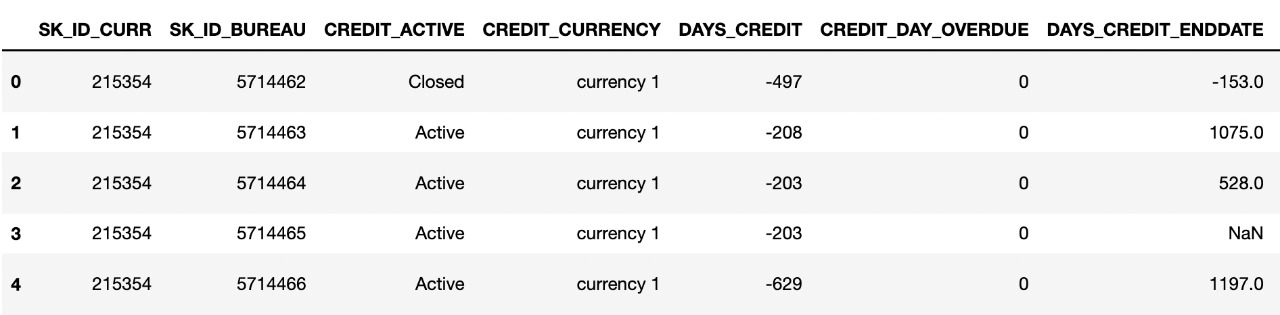
**Table 1. POS\_CASH\_balance.csv**



***bureau.csv***

This dataset gives information about the type of credit, debt, limit, overdue, maximum overdue, annuity, remaining days for previous credit, etc.

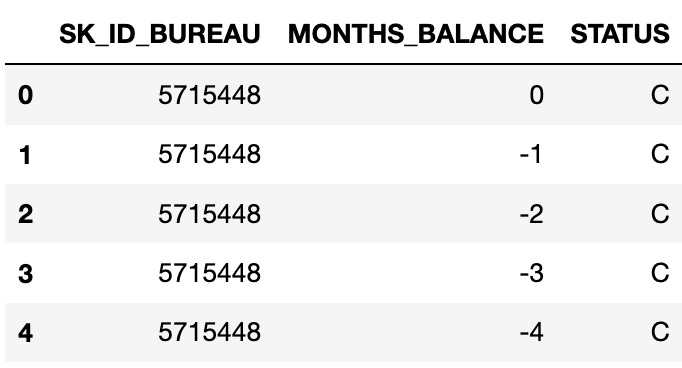
**Table 2. Bureau.csv**

****

***bureau\_balance.csv***

This dataset gives information about the Status of the Credit Bureau loan during the month, the Month of balance relative to the application date, Recoded ID of the Credit Bureau credit. 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.

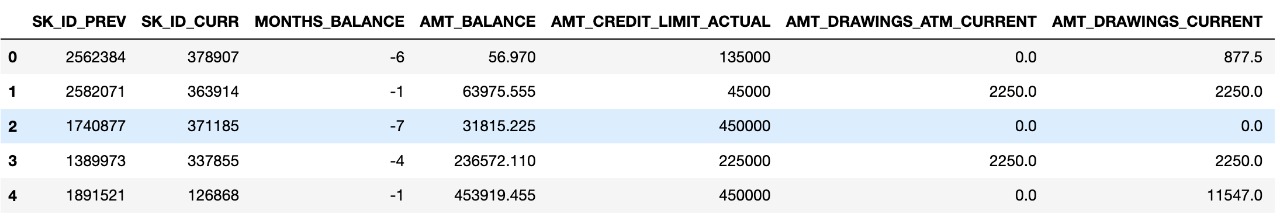
**Table 3. bureau\_balance.csv**

****

***credit\_card\_balance.csv***

This dataset gives information about financial transactions aggregated values such as amount received, drawings, number of transactions of previous credit, installments, etc.  Each row is one month of a credit card balance, and a single credit card can have many rows.

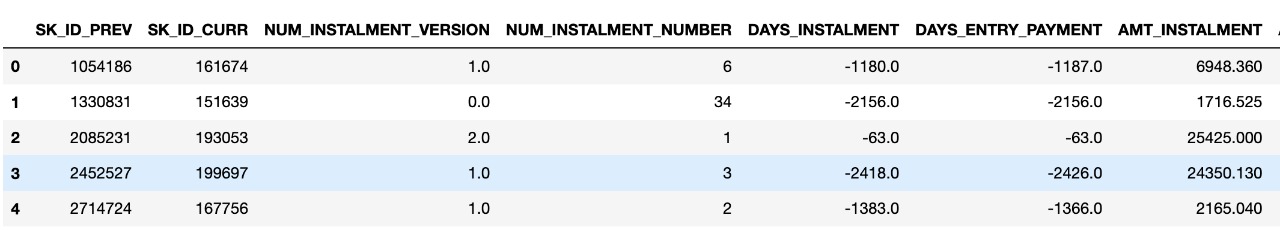
**Table 4. credit\_card\_balance.csv**

****

***installments\_payments.csv***

This dataset gives information about payments, installments supposed to be paid, and their details. There is one row for every made payment and one row for every missed payment.

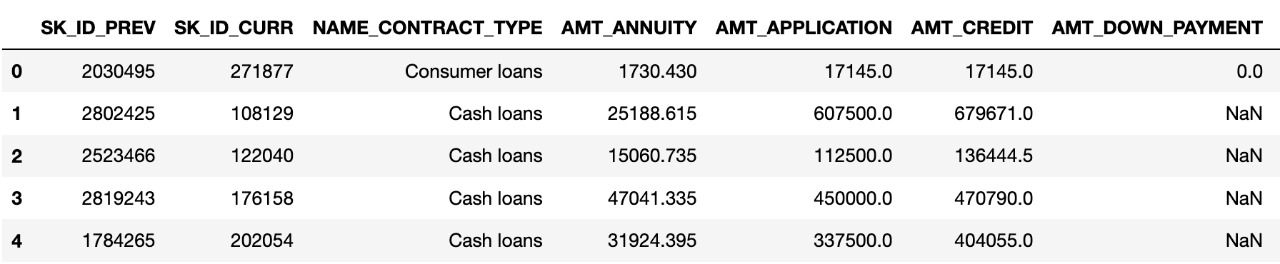
**Table 5. Installments\_payments.csv**



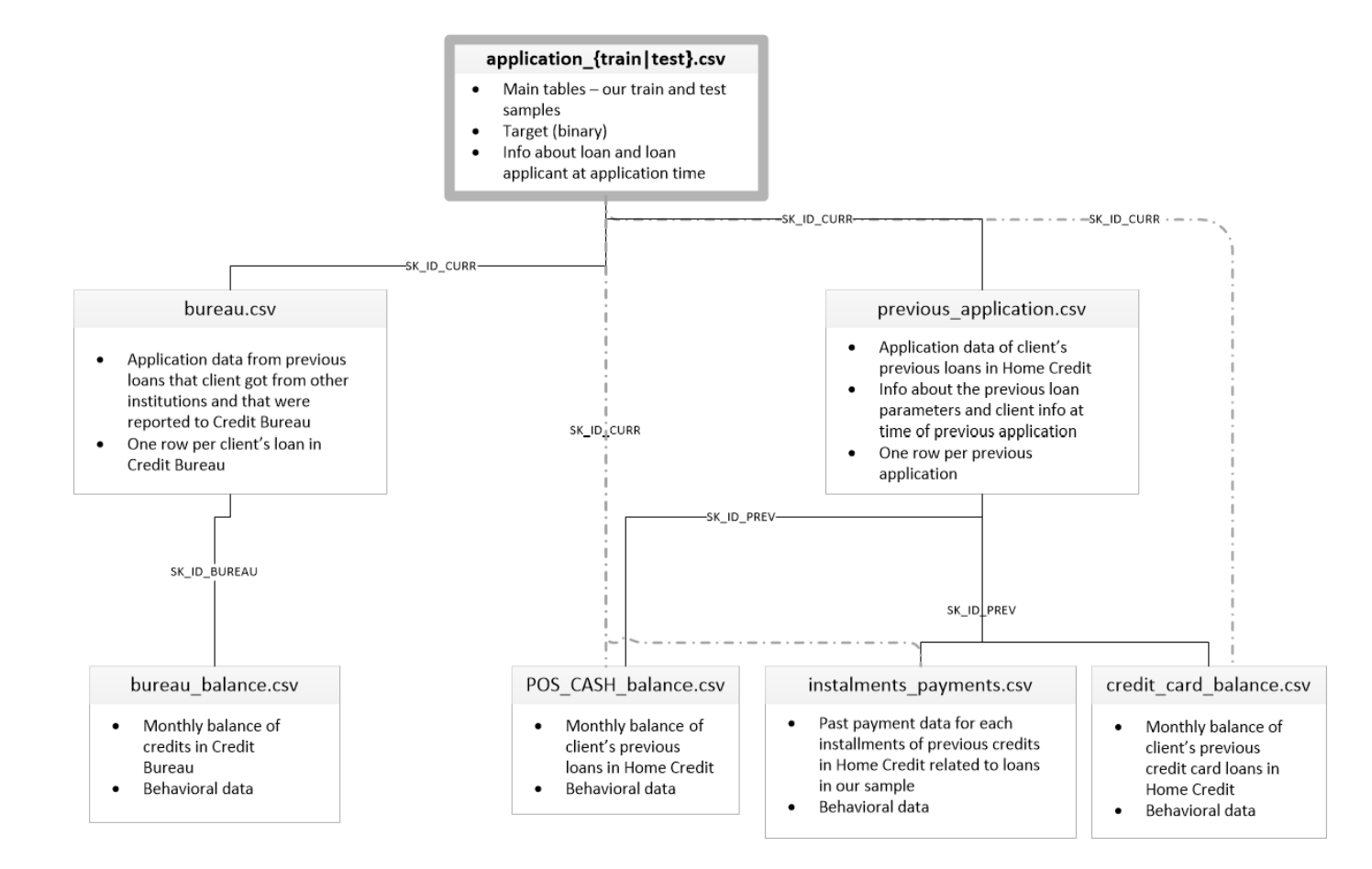
***previous\_application.csv***

This dataset contains information about previous application details of an application. 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.

**Table 6. previous\_application.csv**

****

**Figure 1: Data Description Diagram**

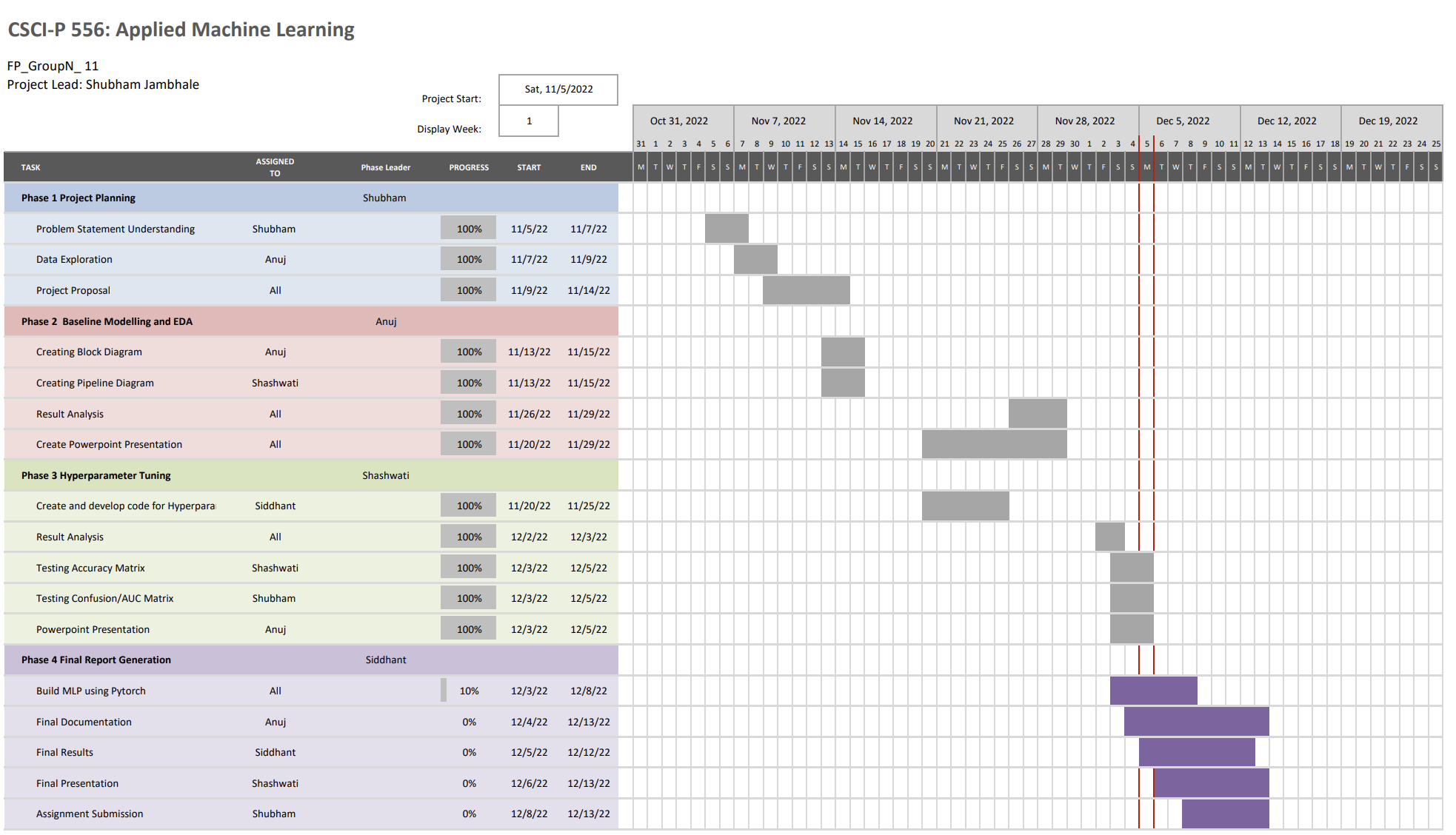
****

**Task To Be Tackled:**

* We need to predict the customer repayment status such as if the user is a defaulter or not by hyper tuning existing baseline machine learning pipelines and models using the datasets provided by Kaggle.

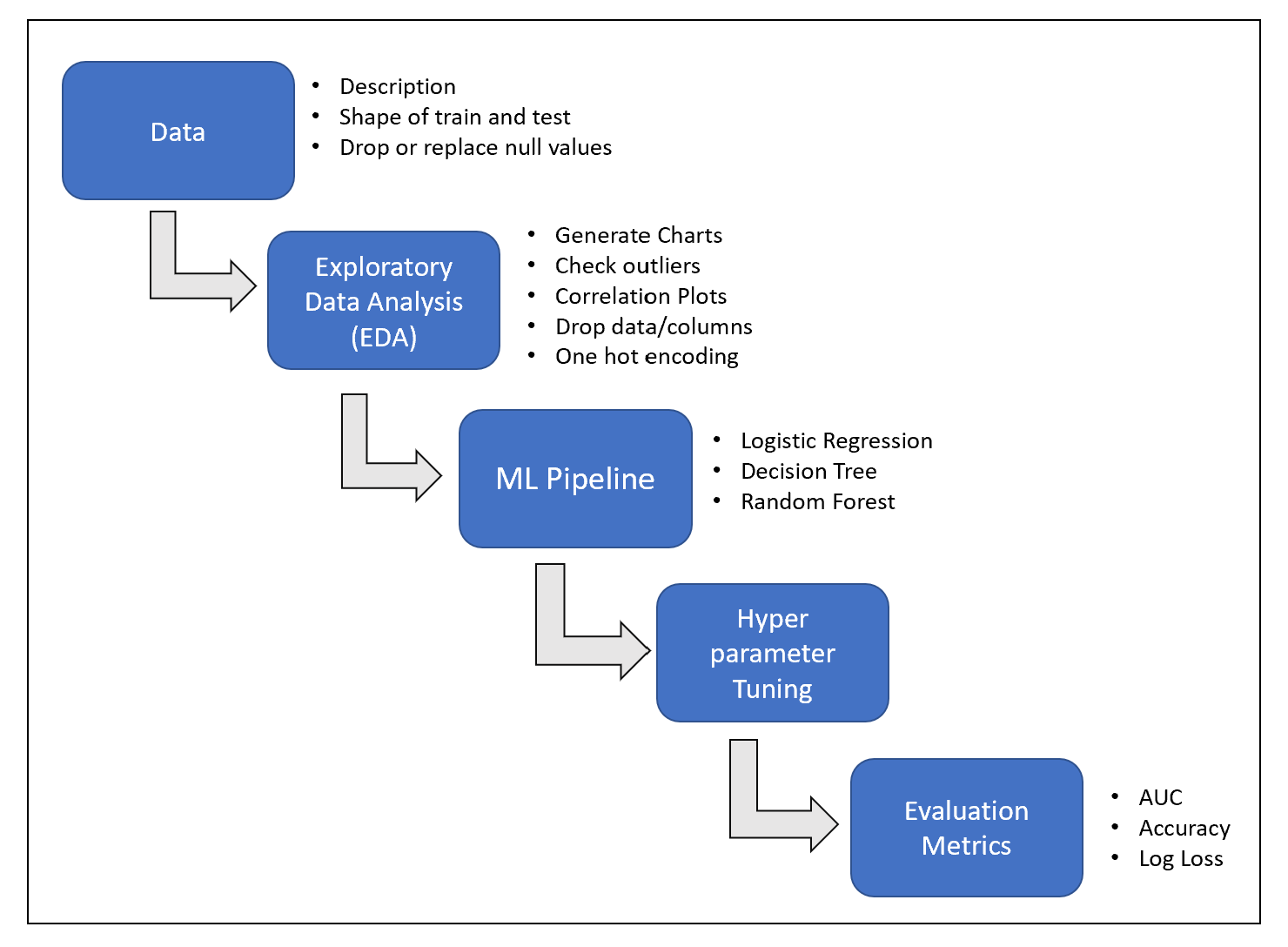
**Gantt Chart**

**Figure 2. Gantt Chart**



**Machine Learning Workflow:**

**Figure 3: Workflow**



**Machine Learning Algorithm and Metrics**

The outcome of this project is to predict, whether the customer will repay the loan or not. That’s why this is a classification task where the outcome is 0 or 1.  To classify this problem we will be building the following machine-learning models:

1. **Logistics Regression**:

* In our case, the number of features is relatively small i.e. <1000, and no. of examples is large. Hence logistic regression can be a good fit here for the classification.

1. **Decision Tree**:

* Decision trees are better for categorical data and our target data is also categorical in nature that’s why decision trees are a good fit.

1. **Random Forest**:

* Random Forest works well with a mixture of numerical and categorical features.
* As we have a good amount of mixture of both types of features random forest can be a good fit.

1. **Lasso Regression:**

* The bias-variance trade-off is the basis for Lasso's superiority over least squares. The lasso solution can result in a decrease in variance at the cost of a slight increase in bias when the variance of the least squares estimates is very large. Consequently, this can produce predictions that are more accurate.

1. **Ridge Regression:**

* Any data that exhibits multicollinearity can be analyzed using the model-tuning technique known as ridge regression. This technique carries out L2 regularization. Predicted values differ much from real values when the problem of multicollinearity arises, least-squares are unbiased, and variances are significant.

**Loss Function:**

* **Log Loss:**
* How closely the forecast probability matches the associated real or true value is indicated by log-loss (0 or 1 in case of binary classification). The higher the log-loss number, the more the predicted probability deviates from the actual value.



**Metrics:**

1. **Confusion Metrics:**

* A confusion matrix, also called an error matrix, is used in the field of machine learning and more specifically in the challenge of classification. Confusion matrices show counts between expected and observed values. The result "TN" stands for True Negative and displays the number of negatively classed cases that were correctly identified. Similar to this, "TP" stands for True Positive and denotes the quantity of correctly identified positive cases. The term "FP" denotes the number of real negative cases that were mistakenly categorized as positive, while "FN" denotes the number of real positive examples that were mistakenly classed as negative. Accuracy is one of the most often used metrics in classification.

**Figure 4: Confusion Matrix**

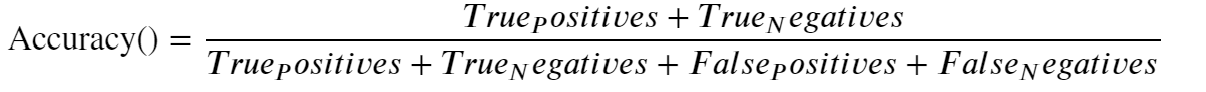


1. **AUC:**

* AUC stands for "Area under the ROC Curve." It measures the entire two-dimensional area underneath the entire ROC curve from (0,0) to (1,1). It is a widely used accuracy method for binary classification problems

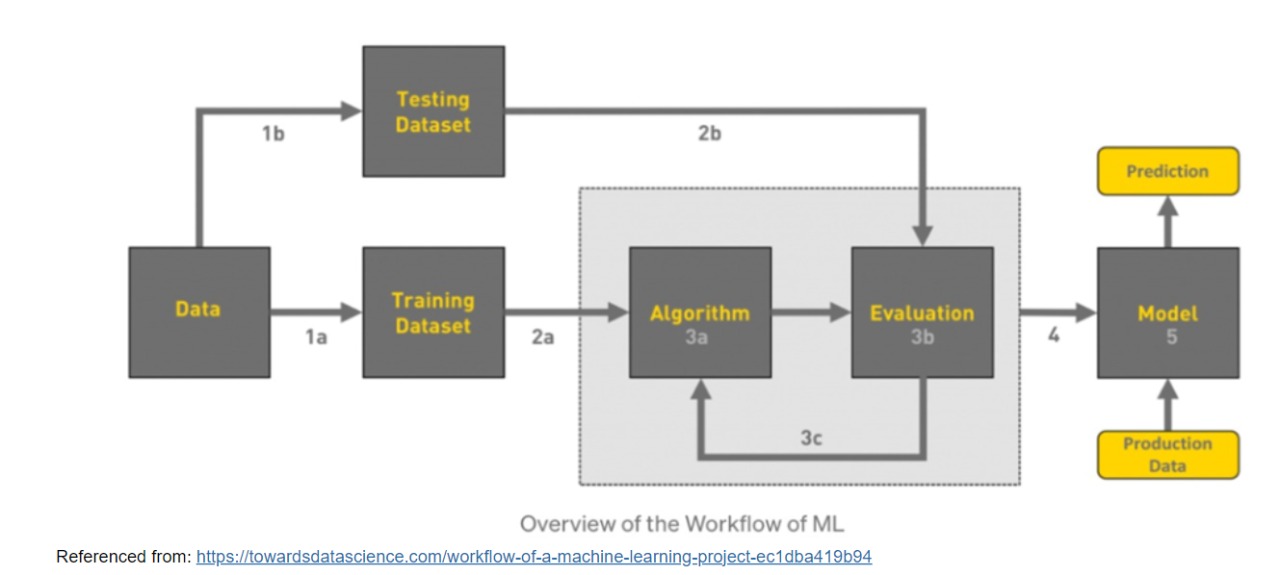
1. **Accuracy:**

* The accuracy score is used to gauge the model's effectiveness by calculating the ratio of total true positives to total true negatives across all made predictions. Accuracy is generally used to calculate binary classification models.



**Block Diagram:**

**Figure 5: Block Diagram of Project**

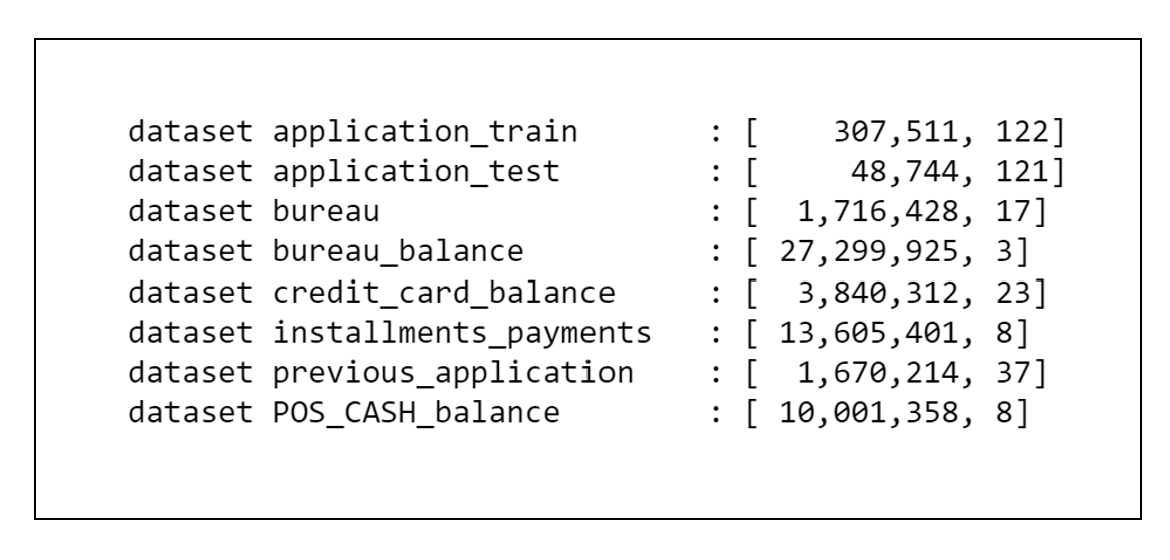


**Exploratory Data Analysis (EDA):**

1. **Data Set Dictionary and size:**

* The table below describes the dataset size like the number of rows and the number of columns.

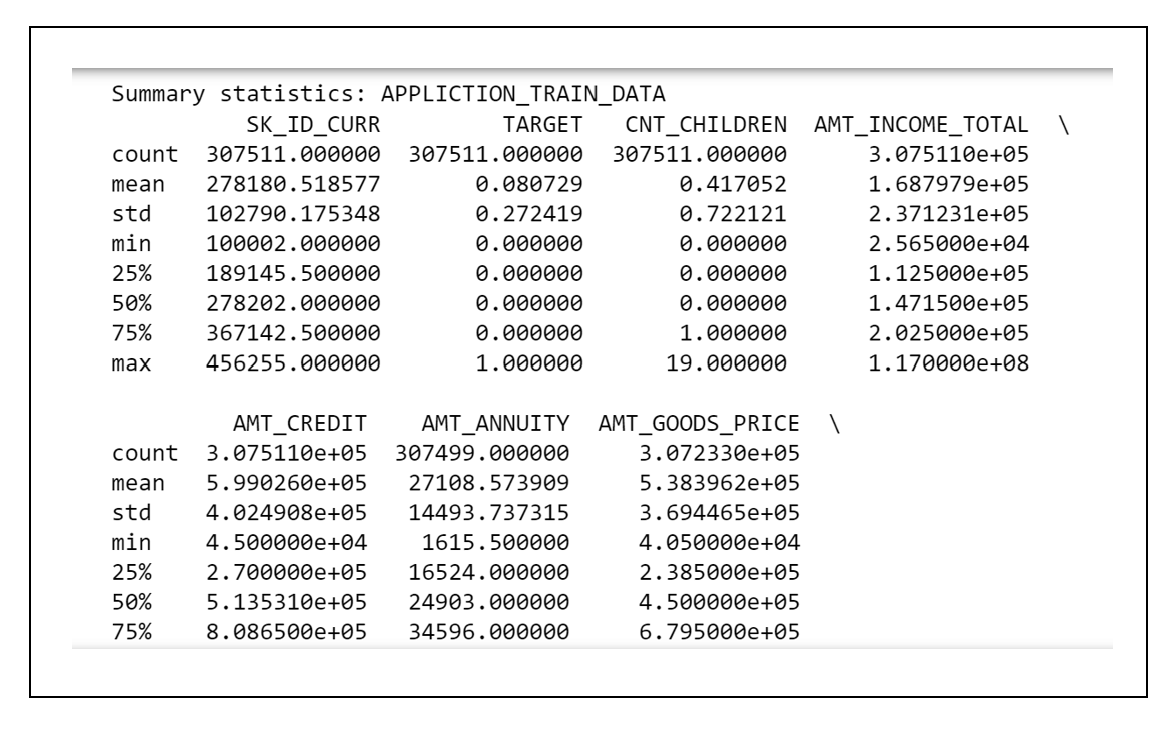
**Figure 6: Data Set Dictionary**



1. **Summary Statistic:**

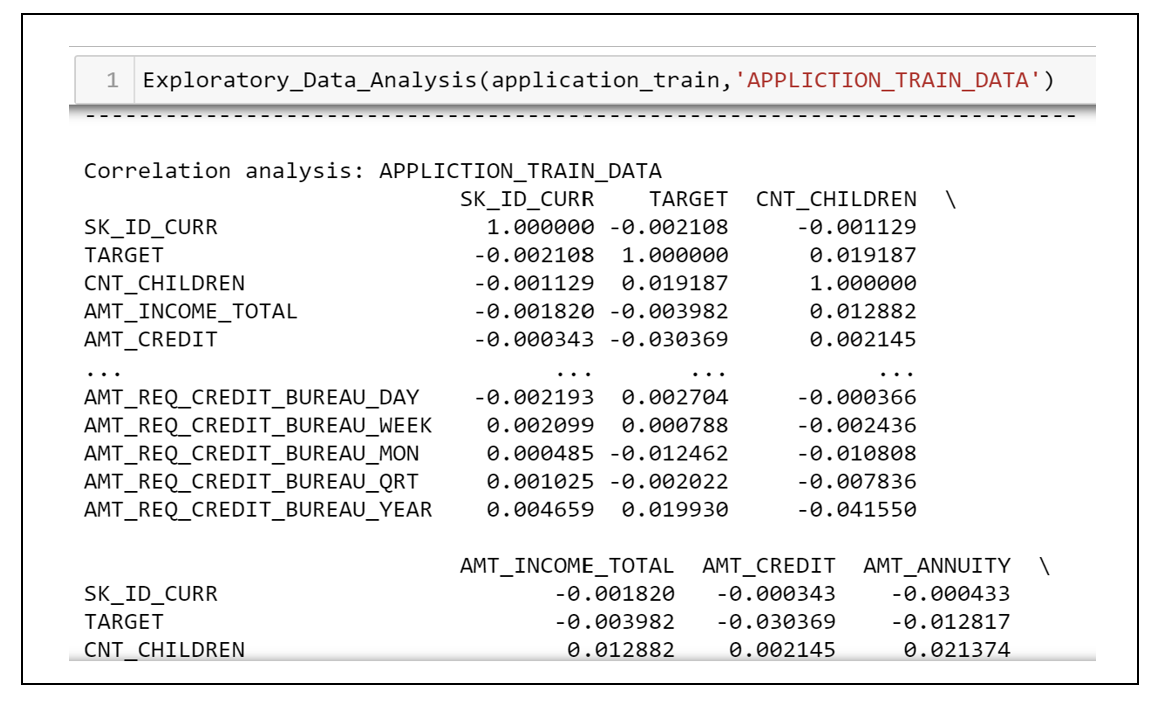
* The Table below describes the contents of the table.

**Figure 7: Summary Statistic**



1. **Correlation Analysis:**

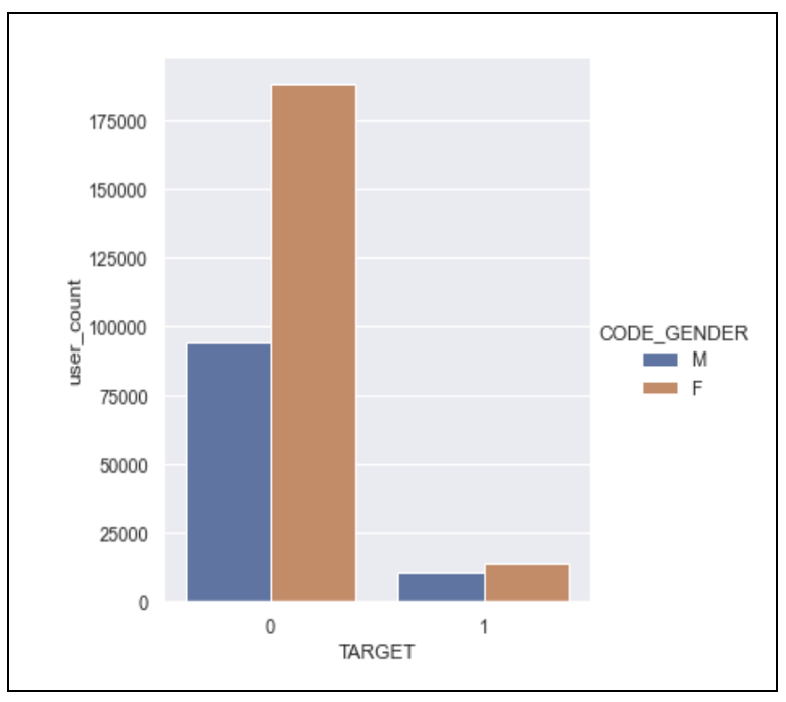
**Figure 8 : Correlation Analysis**



**Visual Exploratory Data Analysis (EDA)**

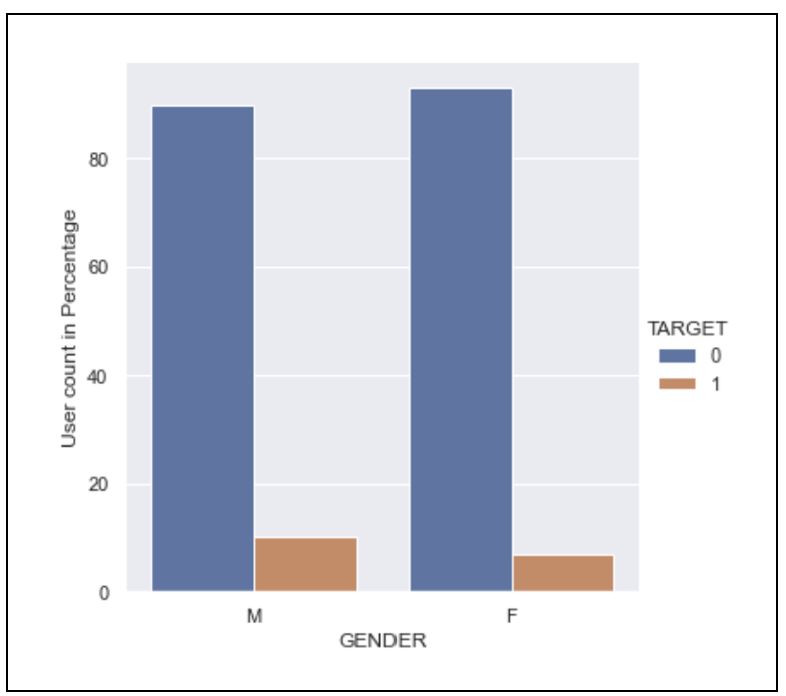
* **Descriptive Analysis**
* We performed descriptive analysis on the dataset, identifying the data type for each feature, its size (rows and columns = 307511, 122), and summary statistics for all features, including the number of observations, mean, standard deviation, maximum, minimum, and quartiles.
* We generated charts on descriptive statistics of the target dataset.
* **Summary Statistics**

**Figure 9: Target VS Borrowers based on Gender**

****

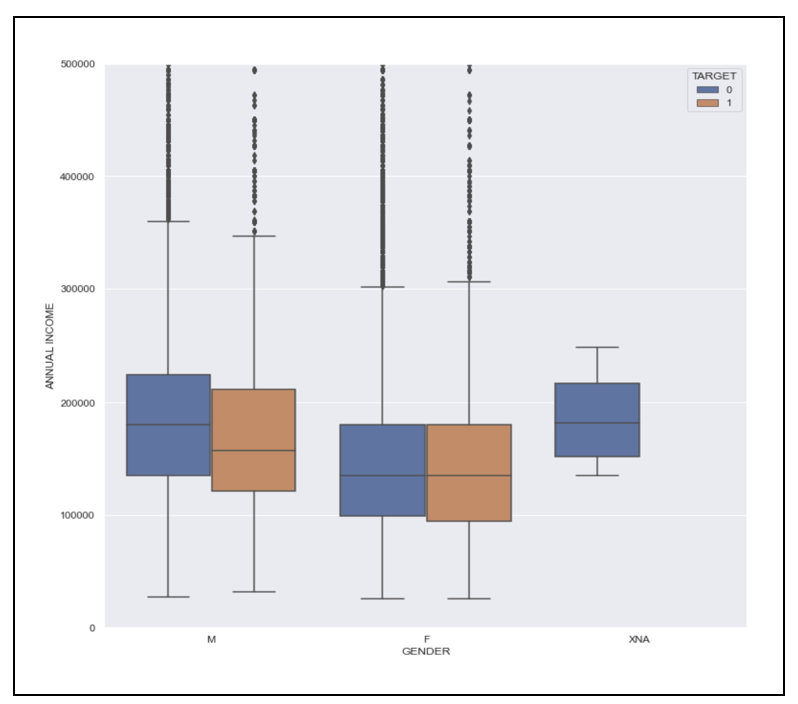
* Both borrowers and targets are more female than male.
* In the borrowing group, there was a significantly bigger disparity between men and women.

**Figure 10: Target vs Borrower based on gender**



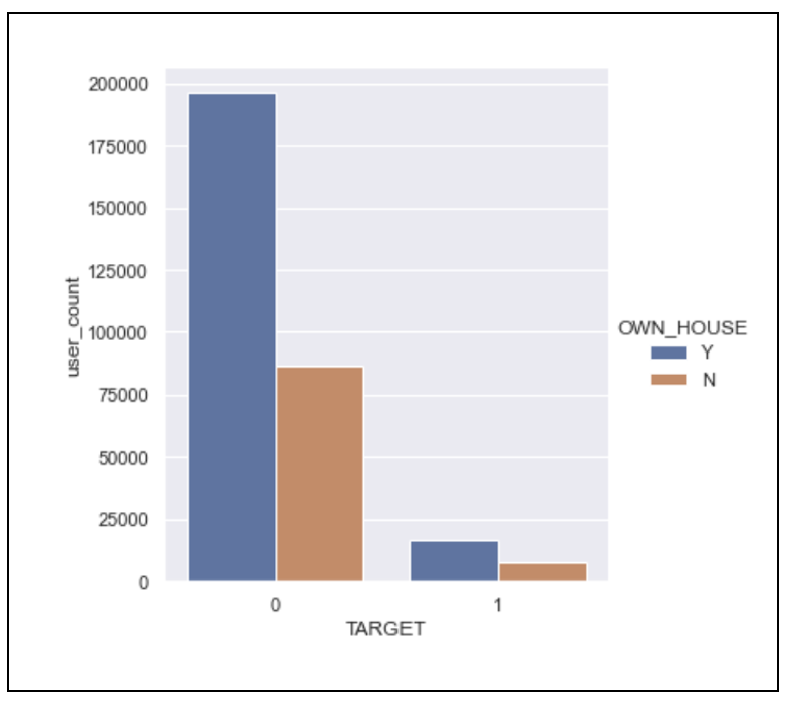
* Based on the proportion of defaulters count (Second Graph), men are more likely than women to default.

**Figure 11: Gender vs Income based on target**



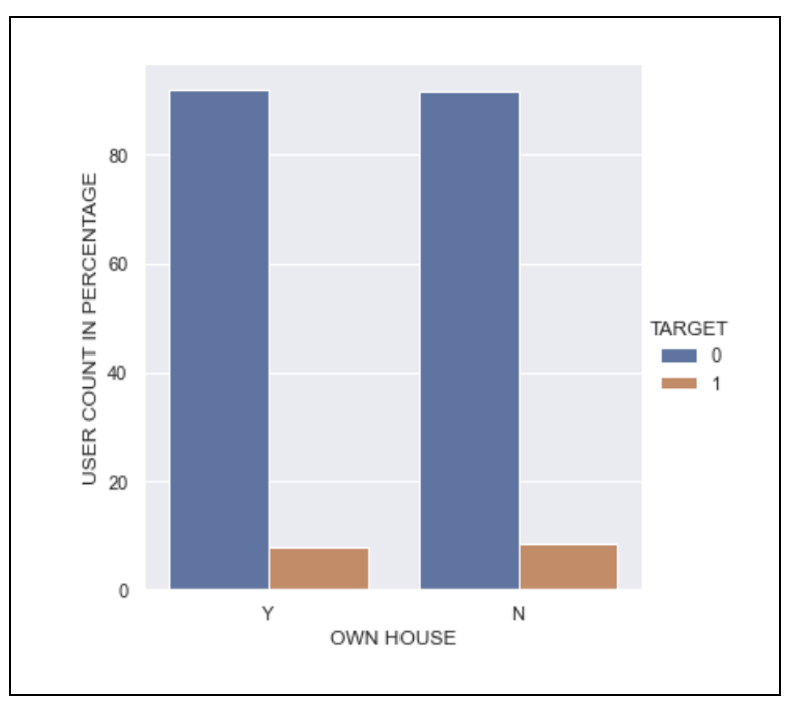
* In both target and non-target, men earned more than women. Men in non-target groups earn more money than men in target groups.

**Figure 12: Own House Count based on Target**



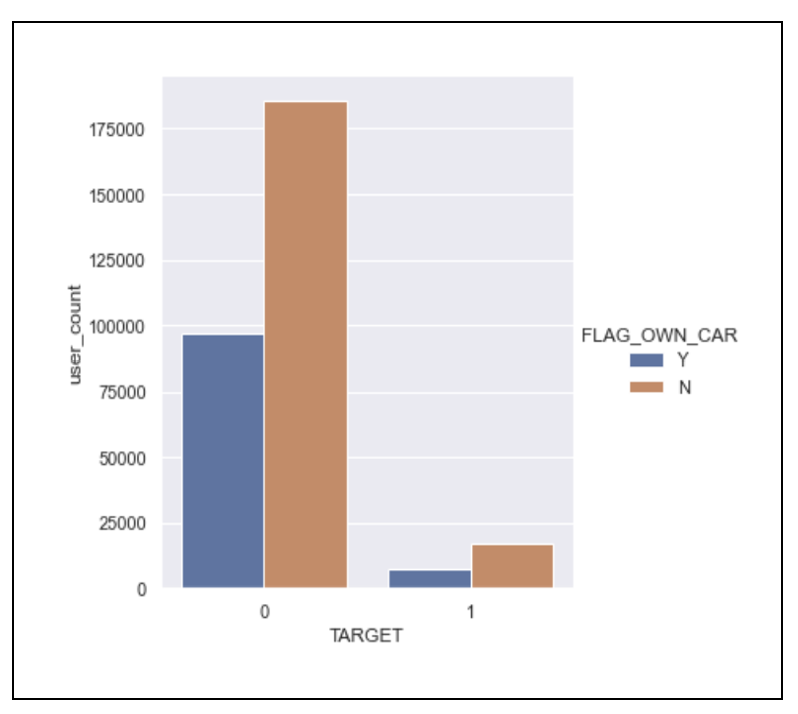
* The number of homeowners in non-target is higher than the number of renters.
* The target has a higher percentage of homeowners than renters.

**Figure 13: Own house count based on Target (in percentage)**



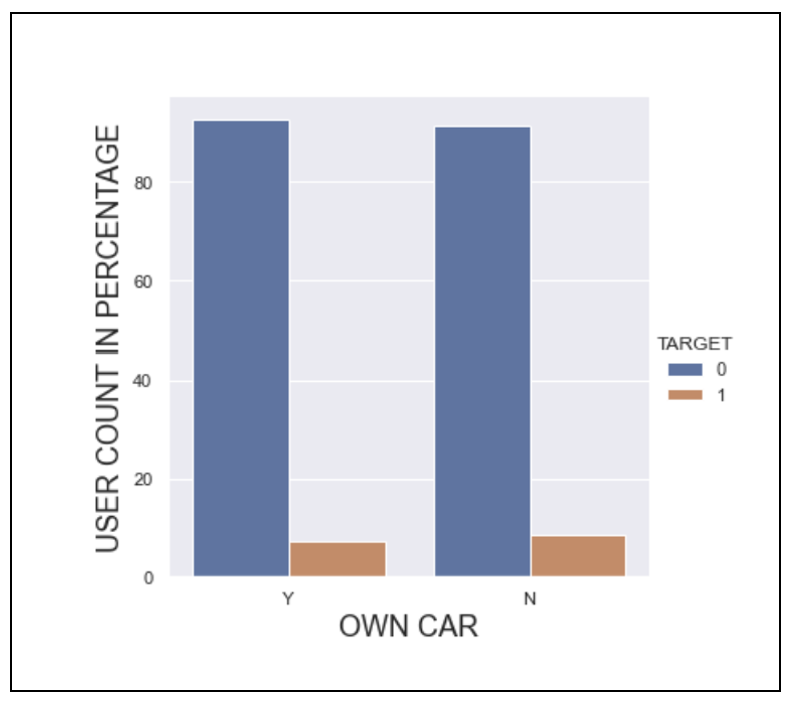
* Borrowers who own a home are more likely to pay back loans, while the difference is not great.

**Figure 14: Own Car count based on Target**



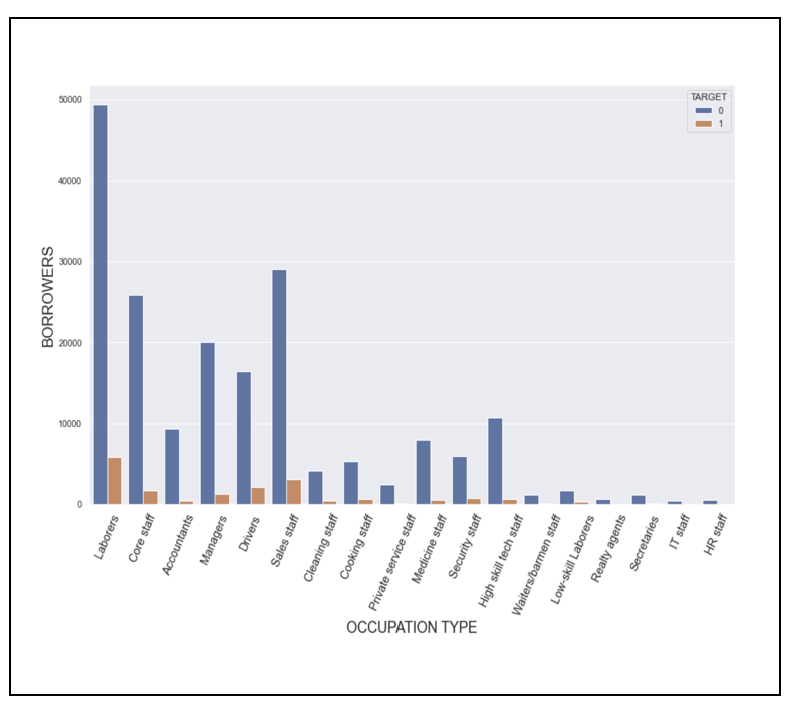
* Both in the target and non-target, there are more persons without automobiles than those who do.

**Figure 15: Own Car count based on target (in percentage)**

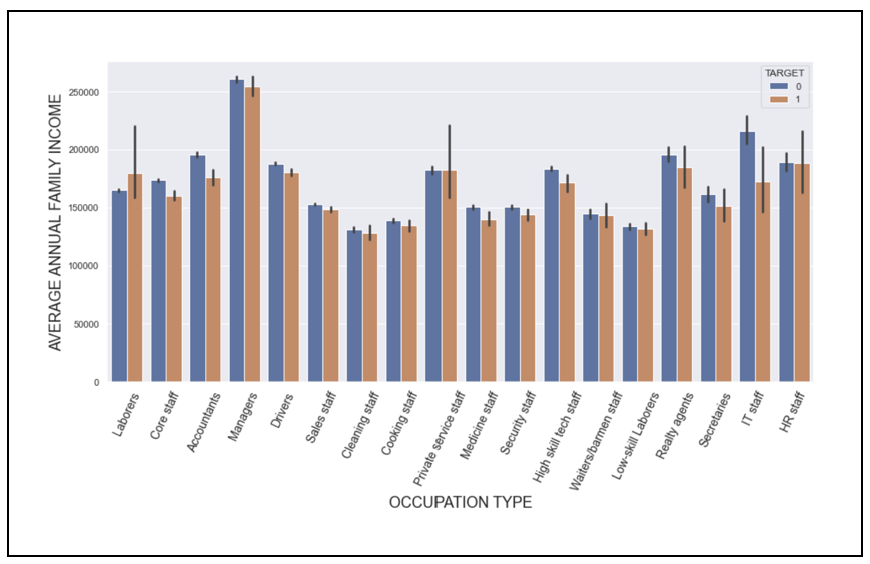


* Borrowers who own cars are more likely to make timely payments.

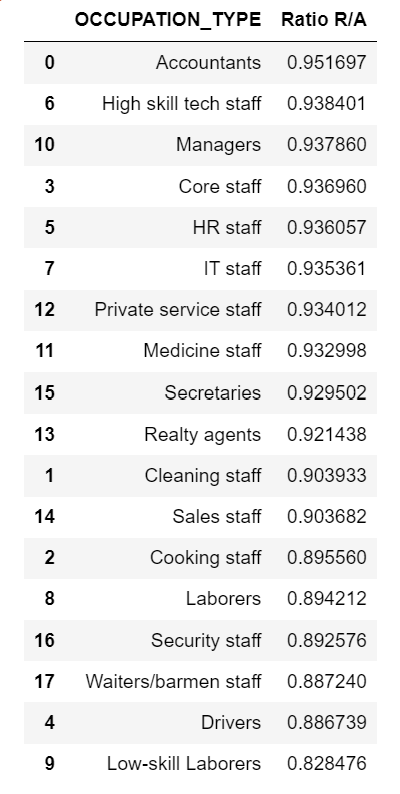
**Figure 16: Occupation Type Count based on Target**



**Figure 17: Occupation Type vs Income based on Target**

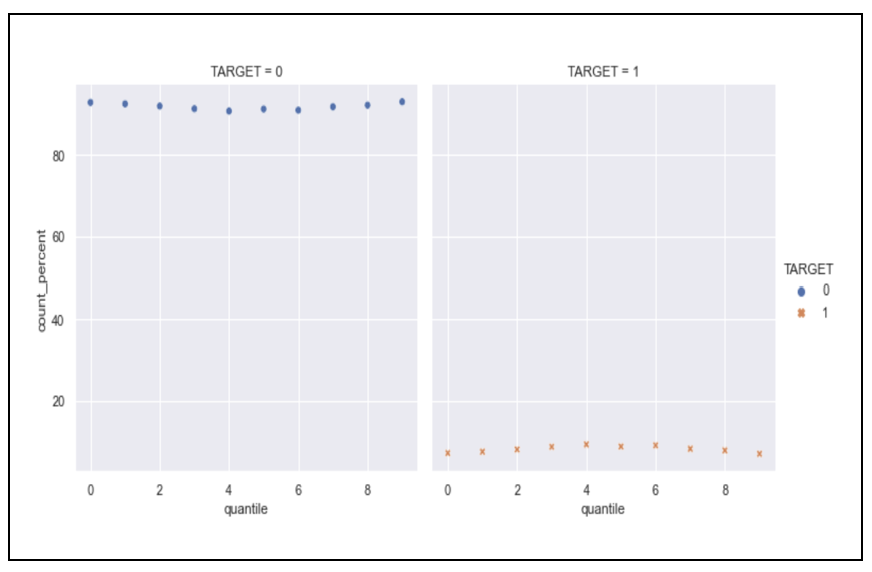


**Figure 18: Re Payers to Application Ratio**



* The above figure describes the ratio of repayment based on occupation type.

**Figure 19: Quantiles vs Income Credit Ratio**



* Defaulters percentage is less when IC\_ratio is either Low or High

**Feature Extraction**

* **Step 1:**
* Following columns give us the number of enquiries done.
* AMT\_REQ\_CREDIT\_BUREAU\_HOUR
* AMT\_REQ\_CREDIT\_BUREAU\_DAY
* AMT\_REQ\_CREDIT\_BUREAU\_WEEK
* AMT\_REQ\_CREDIT\_BUREAU\_MON
* AMT\_REQ\_CREDIT\_BUREAU\_QRT
* AMT\_REQ\_CREDIT\_BUREAU\_YEAR

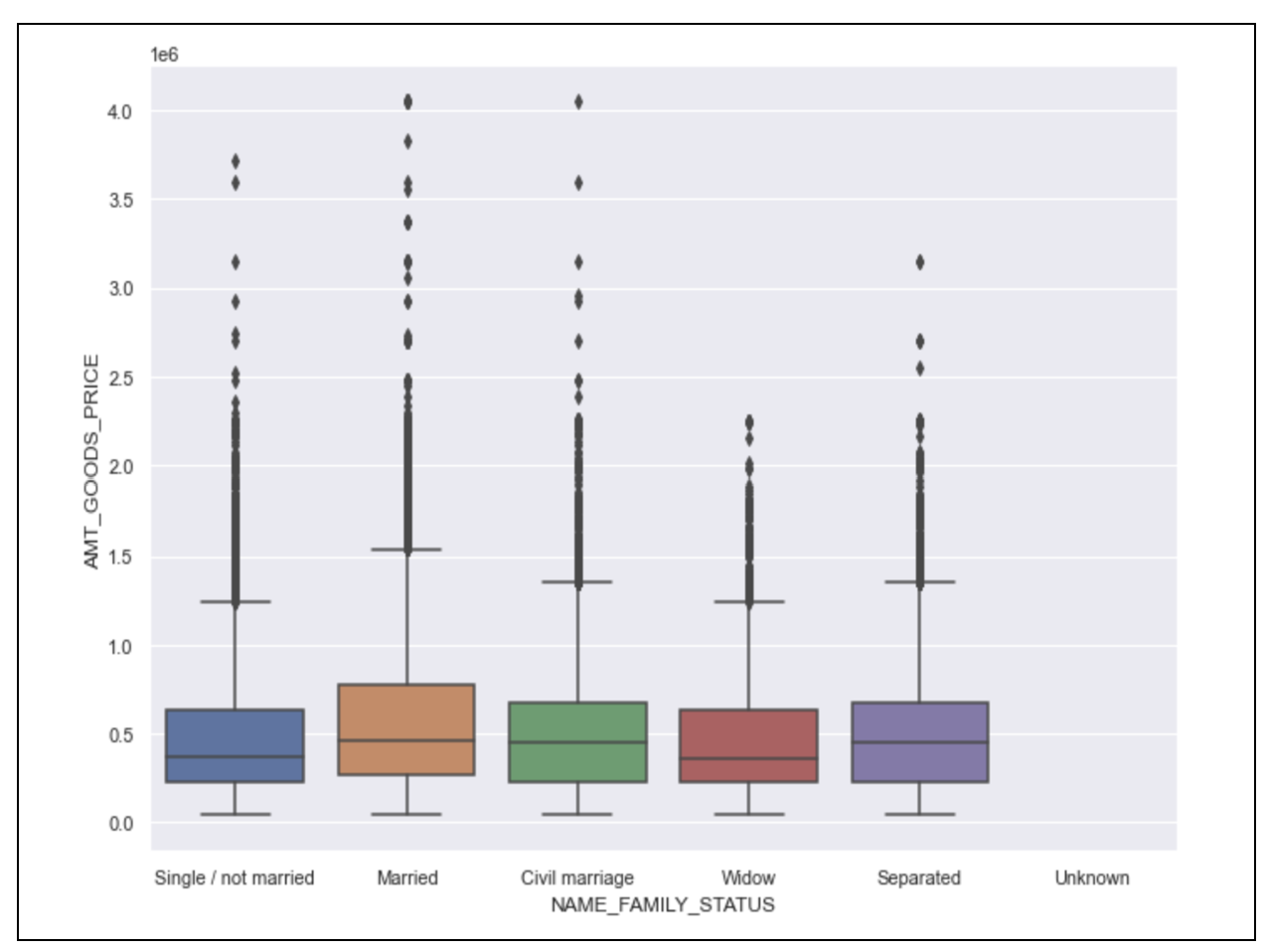
In all these columns for some entries values are not mentioned, so we can assume that there are no enquiries for those inputs. We can replace null values in such inputs by 0

* **Step 2:**
* Following columns give us the number of immediate connections who have a loan in Home Credit.
* OBS\_30\_CNT\_SOCIAL\_CIRCLE
* DEF\_30\_CNT\_SOCIAL\_CIRCLE
* OBS\_60\_CNT\_SOCIAL\_CIRCLE
* DEF\_60\_CNT\_SOCIAL\_CIRCLE

In all these columns for some entries, values are not mentioned, so we can assume that there are no immediate connections for those inputs. We can replace null values in such inputs with 0.

* **Step 3:**
* AMT\_GOODS\_PRICE values are depending on NAME\_FAMILY\_STATUS categories. So we replaced null/N/A values with medians with respect to NAME\_FAMILY\_STATUS. We can see in the below figure that AMT\_GOODS\_PRICE depends on NAME\_FAMILY\_STATUS.

**Figure 20: Name\_Family Status**



* **Step 4:**
* Used median to fill in the empty or missing values for CNT FAM MEMBERS
* **Step 5:**
* In order to replace the EXT SOURCE 2 null or N/A values, we discovered the some of the highly associated variables. We found out that there is high correlation between EXT SOURCE 2 and REGION RATING CLIENT. Due to the categorical nature of REGION RATING CLIENT, we fill null or NA values with the median depending on categories.
* **Step 6:**
* Same as step 8, we need to replace EXT SOURCE 3 null or N/A values. For this, we discovered some of the highly associated variables. We found out that there is a high correlation between EXT SOURCE 3 and DAYS BIRTH. Given that DAYS BIRTH is numerical, we used Linear Regression to fill in the null or NA values.

**Including new features in the training data.**

We tested and trained on a few chosen columns mentioned below,

* Salary-to-Credit Ratio
* Total External Source

**Additional Feature Engineering:**

* Along with the above-mentioned feature engineering we performed the following additional feature engineering
* **Step 1:**
* In the baseline pipeline, we eliminated features having 50 percent or more null values.
* We analyzed that we can drop features having 30 % or more null values as there is no significant impact on those features.
* **Step 2:**
* Dropped DAYS\_LAST\_PHONE\_CHANGE column because there is only 1 row.

**Including new features in the training data.**

We tested and trained on a few chosen columns mentioned below

* Salary-to-Credit Ratio
* Total External Source
* AMT Credit to Annuity Ratio.
* Annuity to Salary Ratio.

Besides the Salary-to-credit-Ratio and Total external Source added in the baseline, in additional feature engineering, we added 2 more columns, the Amt credit to annuity ratio and annuity-to-salary ratio.

We analyzed that ratio of AMT\_CREDIT to AMT\_ANNUITY and the ratio of AMT\_ANNUITY to AMT\_INCOME\_TOTAL can provide us the better result in the aspect of prediction.\

**Impact of newly added Feature:**

* We analyzed that newly added features impacted the decision tree model by a significant amount.
* Test accuracy of the decision tree model is increased by 6 % due to the newly added features as compared to its baseline model.
* There is no significant impact on logistic regression and the random forest model.

**Why we choose the technique and strategy:**

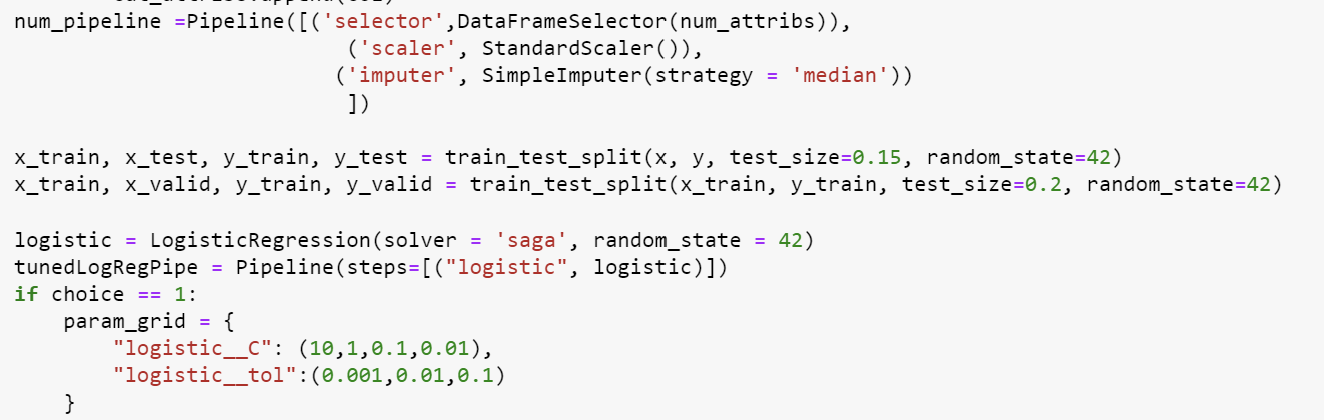
* We removed some of the features which have the most number of null values, as they will contribute very little to the predictions.
* For other features, It is best to handle filling in null values for categorical and continuous variables individually. It won't be possible to fill all category variables with the most or least frequent values and all continuous variables with the median value.
* A reliable metric to assess a person's reliability and repayment capacity would be ratios between income, credit requested, and credit to be paid per year. So, we thought about putting above.

**Hyper Parameter Tuning:**

* After additional feature engineering, we hyper-tuned our model to find the optimal parameters.
* We conducted the hyperparameter tuning by using a grid search on logistic regression, decision tree, random forest, lasso, and ridge regression.

1. Logistic Regression:

**Figure 21: Logistic Regression Hyperparameters**



In logistic regression, we used different C and tolerance values as hyper tuning parameters.

1. Decision Tree:

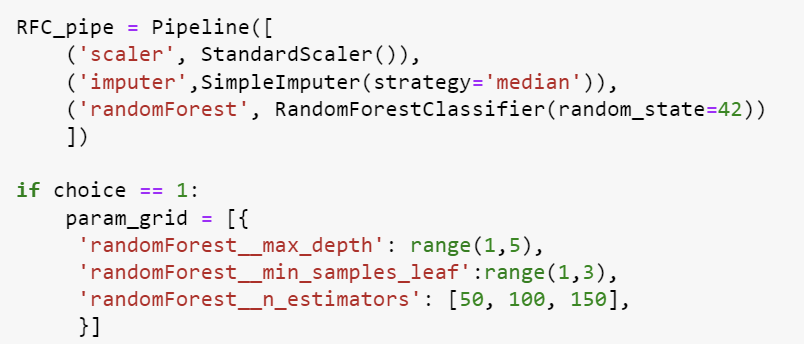
**Figure 22: Decision Tree Hyperparameters**



For a decision tree model, we used different hyper tuning parameters such as max depth, min samples leaf, and criterion.

1. Random Forest :

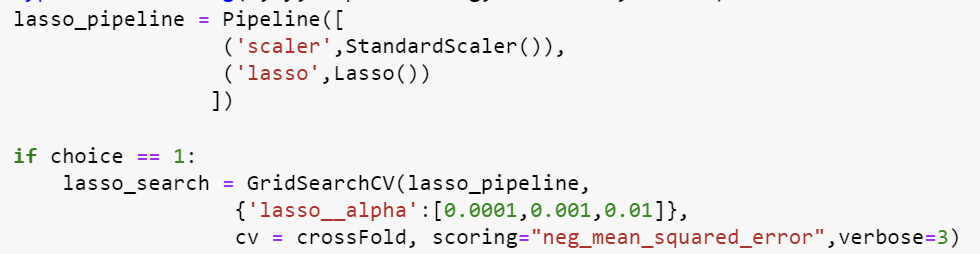
**Figure 23: Random Forest Hyperparameter.**



For a random forest model, we used different hyper-tuning parameters such as max depth, min samples leaf, and estimators.

1. Lasso Regression:

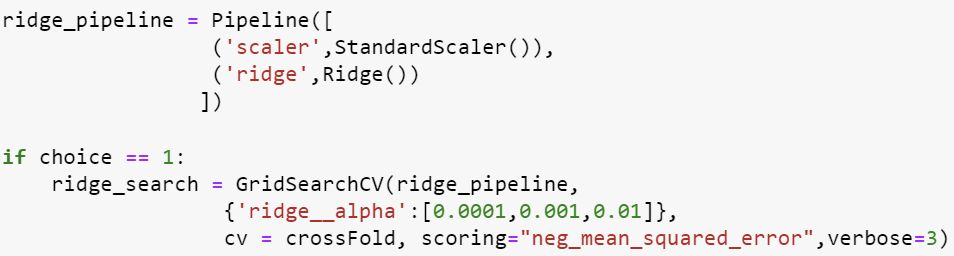
**Figure 24: Lasso Hyperparameter**



For Lasso regression we use different alpha parameters.

1. Ridge Regression:

**Figure 25: Ridge Hyperparameters**



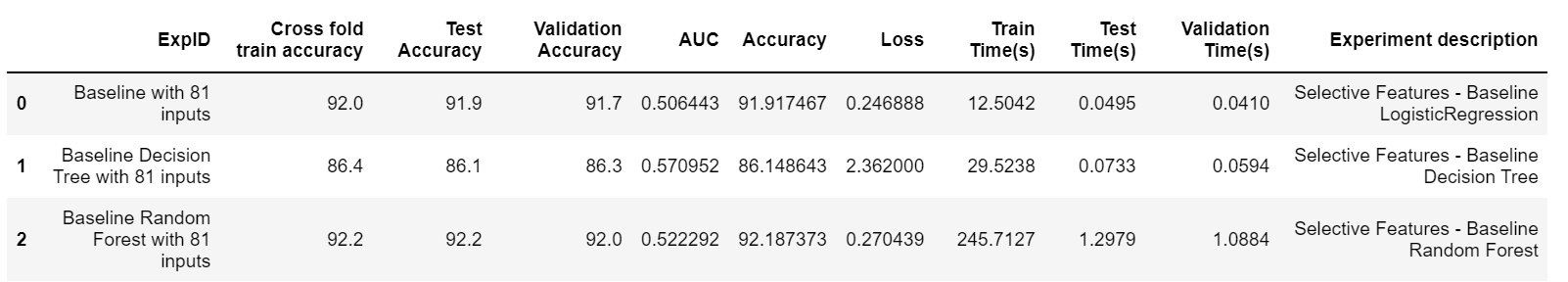
For ridge regression we use different alpha parameters.

**Experiments:**

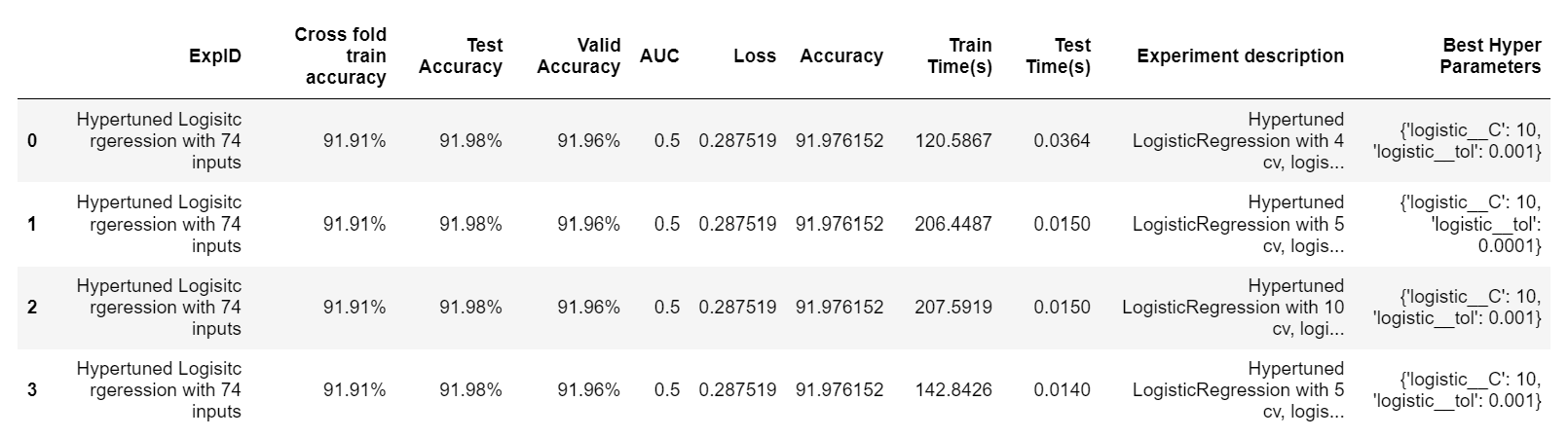
1. Logistic Regression:

* Train and test data were separated. With a random seed set to 42, we divided the 20% test data for accurate findings.
* Next, a logistic regression baseline process was constructed. Based on numerical properties and a common scaler, we construct a numerical pipeline. We use the median to impute the missing data. With this numerical pipeline, logistic regression is performed.
* Finally, using multiple splits and a test size of 0.3, we generate cross-validation splits. We use this cross-validation to compute test accuracy and AUC.
* In Logistic Regression we can see that the Testing Accuracy is quite high and a descent AUC, so logistic regression can be a good model for this dataset
* The Log Loss (0.28) for logistic regression is on the lower side which means our model is predicting accurate results.
* We conducted multiple experiments by using different values of cross folds, C and tolerance.
* In all the experiments we found that there is no significant improvement in AUC or accuracy.
* Since test accuracy for hyper tuned logistic regression is greater than the baseline logistic regression we should choose grid search logistic regression over baseline logistic regression.
* Following table shows all the experiments we have conducted on logistic regression.

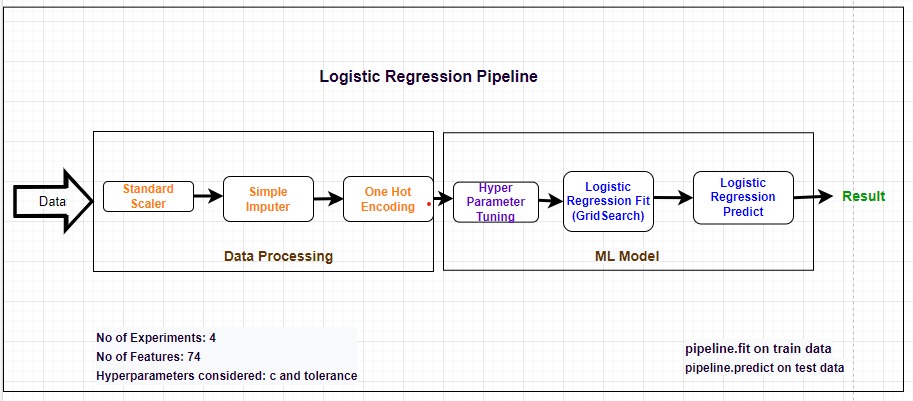
**Figure 26: Baseline Models**

****

**Figure 27: Hyper Tuned Logistic Regression Results.**



**Figure 28: Logistic Pipeline**



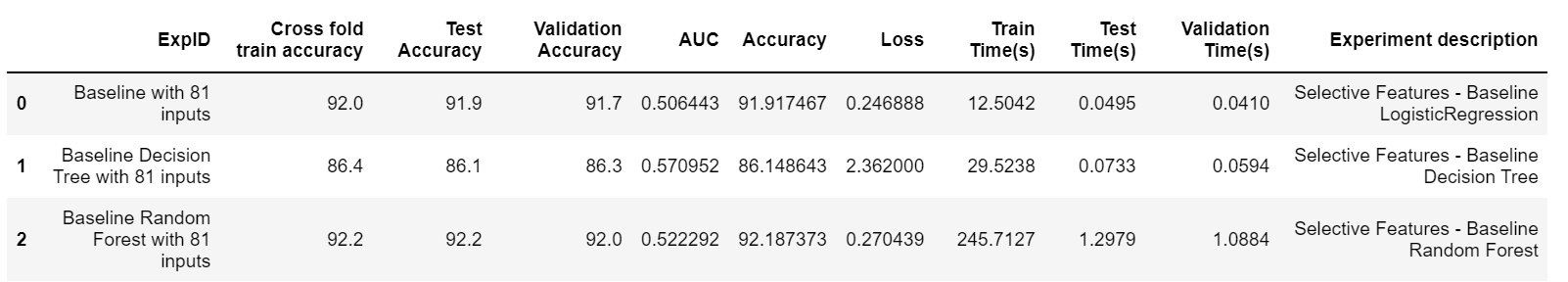
1. Decision Tree:

* Train and test data were separated. With a random seed set to 42, we divided the 20% test data for accurate findings.
* Next, a decision tree baseline process was constructed. We use the median to impute the missing data.
* Finally, using multiple splits and a test size of 0.3, we generate cross-validation splits. We use this cross-validation to compute test accuracy and AUC.
* For the decision tree, as compared to logistic regression the test accuracy is on the higher side.
* The Log Loss for the Decision Tree is on the higher side.
* As compared to baseline model of decision tree from phase 2, hypertuned decision tree model has significantly high test accuracy.
* We conducted multiple experiments by using different values of cross folds, min samples leaf , max depth and criterion.
* From all the experiments we found that there is slight increase in accuracy for 2 cross folds as compared to other values of CV.
* Since test accuracy for hyper tuned Decision tree is greater than the baseline decision tree we should choose grid search decision tree over baseline decision tree.
* Following table shows all the experiments we have conducted on logistic regression.

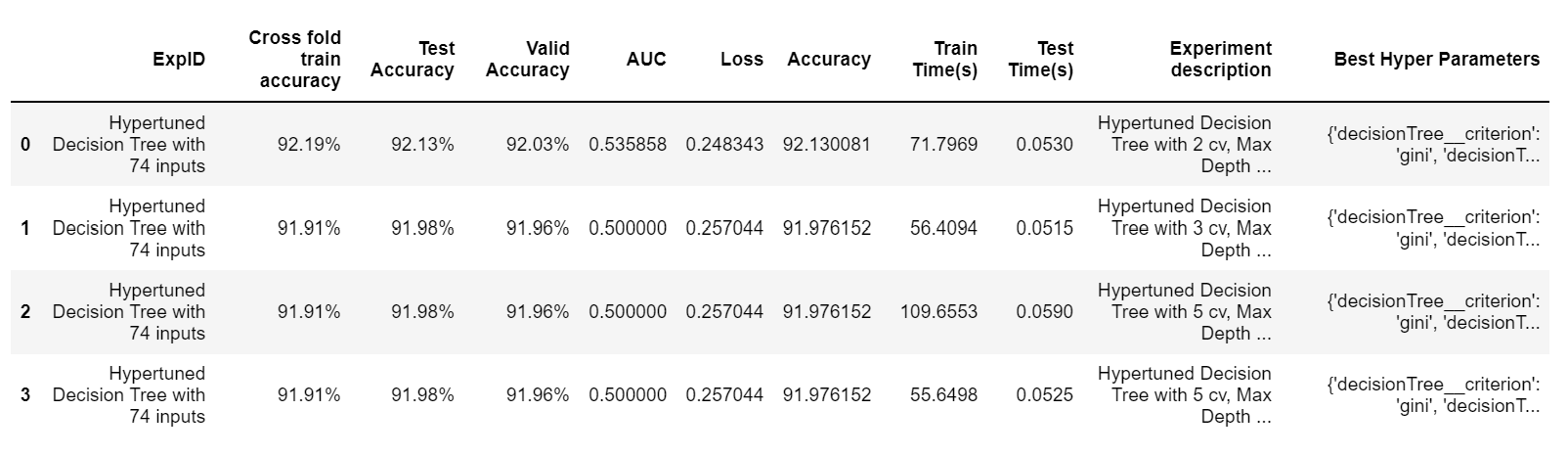
**The best Hyperparameters are:**

****

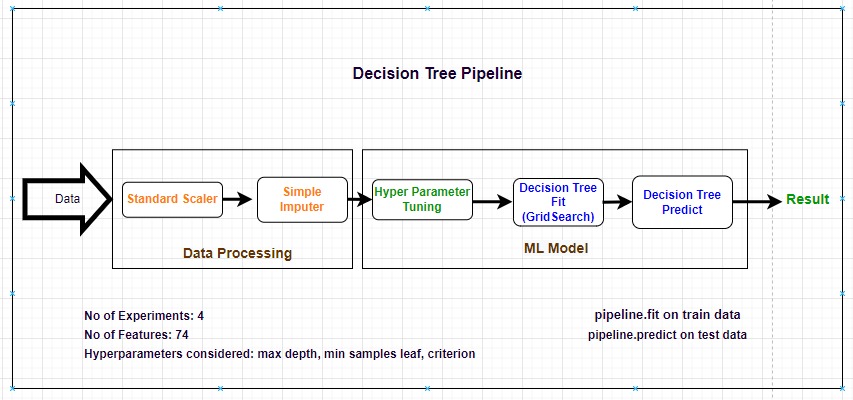
**Figure : Baseline Models**

****

**Figure : Hypertuned Decision Tree Results**



**Figure 22: Decision Tree Pipeline**



1. Random Forest:

* Train and test data were separated. With a random seed set to 42, we divided the 20% test data for accurate findings.
* Next, a random forest baseline process was constructed. We use the median to impute the missing data.
* Finally, using multiple splits and a test size of 0.3, we generate cross-validation splits. We use this cross-validation to compute test accuracy and AUC.
* After running the experiments on the model, the random forest provides us with the test accuracy of 91.98.
* Random forest gave us descent AUC and test accuracy so it can be a good fit model for the given data set.
* The Log Loss (0.27) for Random Forest is on the lower side which means our model is predicting accurate results.
* We conducted multiple experiments by using different values of cross folds, min samples leaf , max depth and estimators.
* From all the experiments we found that there is no significant improvement in either AUC or accuracy.
* As compared to baseline model of random forest from phase 2, hypertuned random forest model has comparatively low test accuracy.
* Since test accuracy for hyper tuned random forest is lower than the baseline random forest we should choose baseline random forest over grid search model.
* Following table shows all the experiments we have conducted on Random Forest.

Figure : Baseline Models

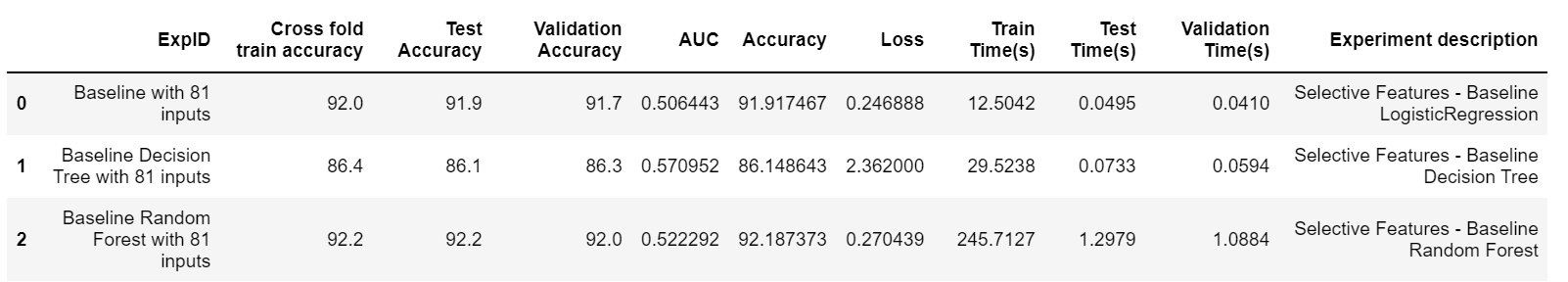
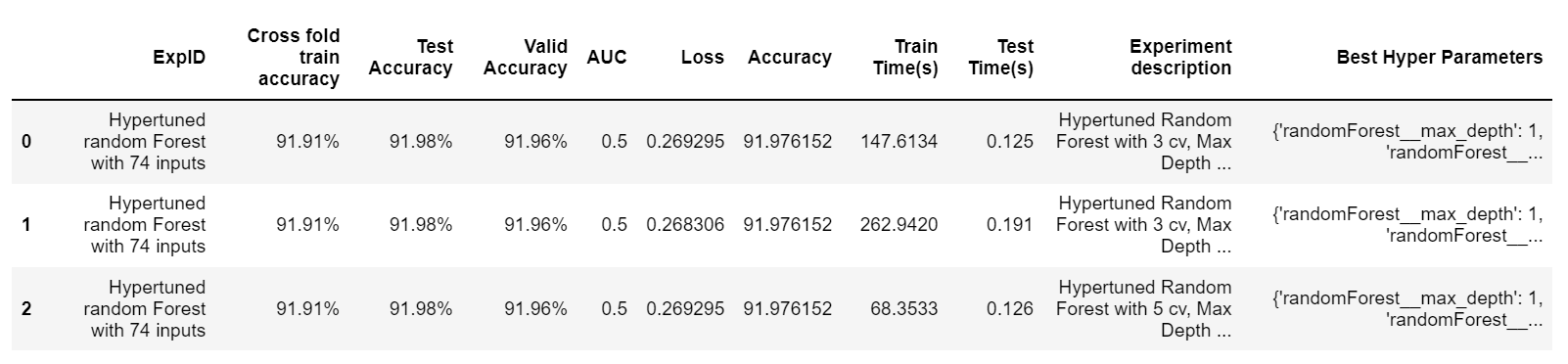
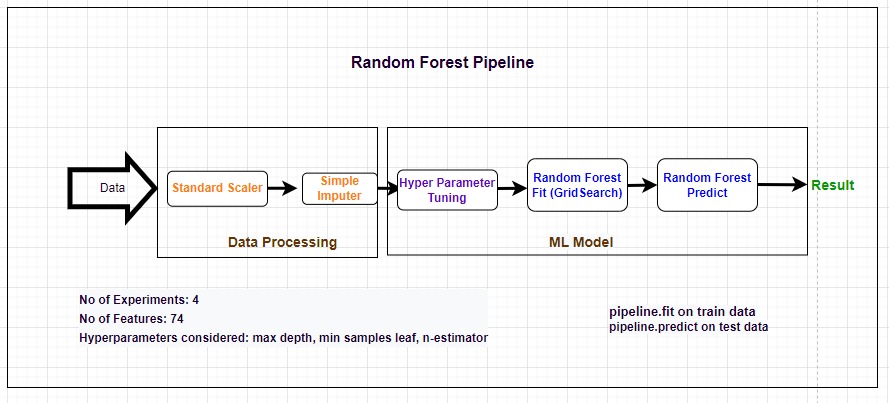
****

Figure : Random Forest Results.



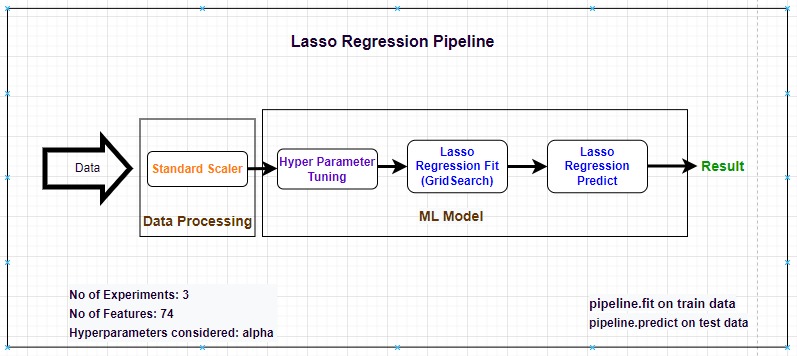
**Figure 23: Random Forest Pipeline**



1. Lasso Regression:

* AUC for lasso regression has increase to 0.755 which is best among all the models we have tested.
* On the other hand, test accuracy has decreased to -6.87 %
* We believe that because of the score function used in Lasso regression we are getting these negative results.
* Although AUC has increased, since test accuracy is not up to the mark, Lasso Regression is not the model to look up for.
* We conducted multiple experiments by using different values of cross folds and alpha.

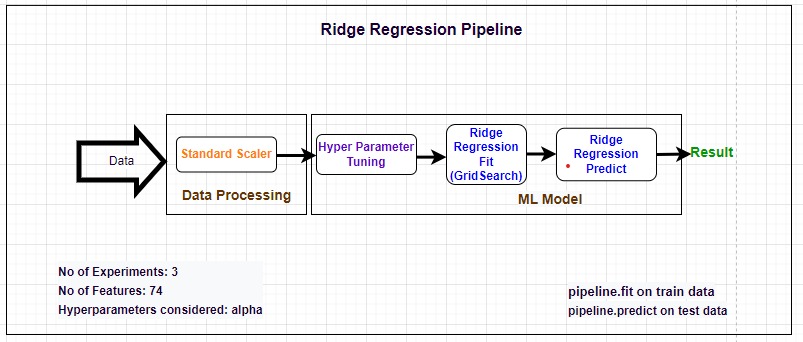
**Figure : Lasso Regression Pipeline**



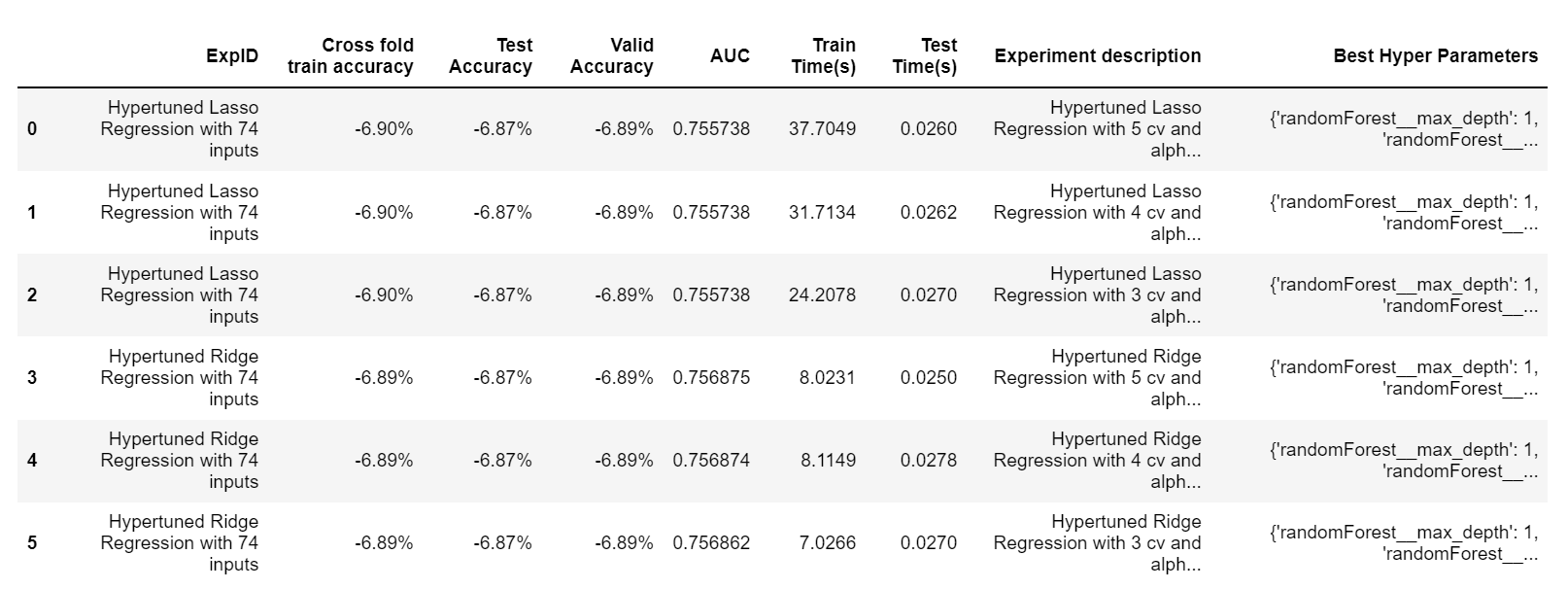
1. Ridge Regression:

* AUC for ridge regression has increase to 0.755 which is best among all the models we have tested.
* On the other hand, test accuracy has decreased to -6.87 %.
* We believe that because of the score function used in Ridge regression we are getting these negative results.
* Although AUC has increased, since test accuracy is not up to the mark, Ridge Regression is not the model to look up for.
* We conducted multiple experiments by using different values of cross folds and alpha

**Figure : Ridge Regression Pipeline**



**Figure: Test Results of Lasso And Ridge Regression**



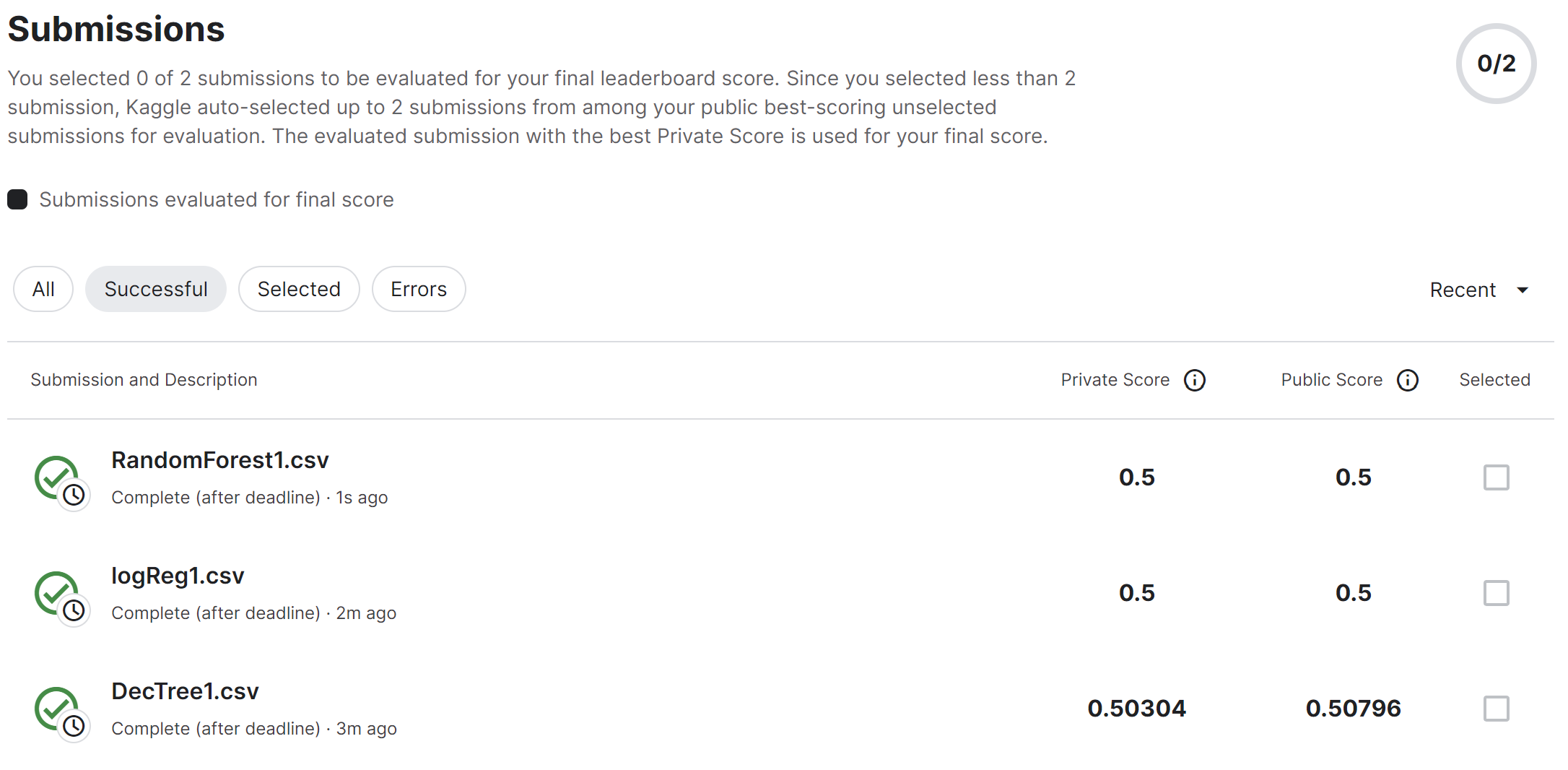
**Result and Discussion:**

From the experiment log table above, describes the accuracy, AUC, and loss of hyper tuned machine learning model logistic regression, Decision Tree, random forest, lasso regression and ridge regression. For the hyper tuned model decision tree model, we can see that the train (92.19) and test (92.13) accuracy has increased significantly as compared to its baseline model, which means it is performing well on the provided dataset. The log loss for decision tree is on the lower side which is 0.24 and has significantly dropped as compared to its baseline model as well as its AUC is also 0.53. So, the algorithm is performing well for given set of input features. The overall accuracy of decision tree has increased by comparatively large margin and went upto 92 %.

Both Random Forest and logistic regression have approximately the same train and test accuracy and log loss as compared to baseline. There is no significant improvement on their hyper tuned parameter model. But hyper tuned Decision Tree remains the best-fit algorithm as it beats others by a very small margin in all the criteria. We observed an increase of 6 percent in test accuracy, and 6 percent in overall accuracy. The log loss for decision tree (0.24) has significantly decreased as compared to its baseline model and is on the lower side and hence it beats the other models.

For Lasso and Ridge Regression we observed that AUC has increased to .75. So, both the models seem to predict the target quite correctly as compared to all other models, but, on the other hand accuracy has decreased dramatically. So, even if the models have high AUC, Lasso and Ridge are not the models to look for. They fail to perform appropriately on HCDR dataset.

**Figure 24: Kaggle Submission**

****

**Conclusion**

The HCDR project's goal is to forecast the population's capacity for payback among those who are underserved financially. Because both the lender and the borrower want reliable estimates, this project is crucial. Real-time Home credit's ML pipelines, which acquire data from the data sources via APIs, run EDA, and fit it to the model to generate scores, which allows them to present loan offers to their consumers with the greatest amount and APR.

Hence if NPA expected to be less than 5% in order to maintain a profitable firm, risk analysis becomes extremely important. Credit history is an indicator of a user's trustworthiness that is created using parameters such as the average, minimum, and maximum balances that the user maintains, Bureau scores that are reported, salary, etc. Repayment patterns can be analysed using the timely defaults and repayments that the user has made in the past. Other criteria such as location information, social media data, calling/SMS data, etc. are included in alternative data. As part of this project, we would create machine learning pipelines, do exploratory data analysis on the datasets provided by Kaggle, and evaluate the models using a variety of evaluation measures before deploying one.

Phase 3 involved the estimation of several models. Data imputation and feature selection were done. We started by selecting features and imputed values. The values of certain features that were missing were filled in. Then, based on our past understanding, we chose to include pertinent features. We trained and assessed several models, including Random Forest, Decision Tree Model, Logistic Regression, Lasso Regression, and Ridge Regression to discover the best one. We hyper tuned them on the best parameters using GridSearch.

We have concluded from phase 3 that the Lasso, Ridge and Logistic Regression models is unable to defeat the other hyper parameter tuned models. The decision tree model performs the best out of all the models. In phase 4 we plan to implement MLP using PyTorch

**Bibliography:**

**Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition**

* by Aurélien Géron

**Lab-End\_to\_end\_Machine\_Learning\_Project**

* by James Shahnan