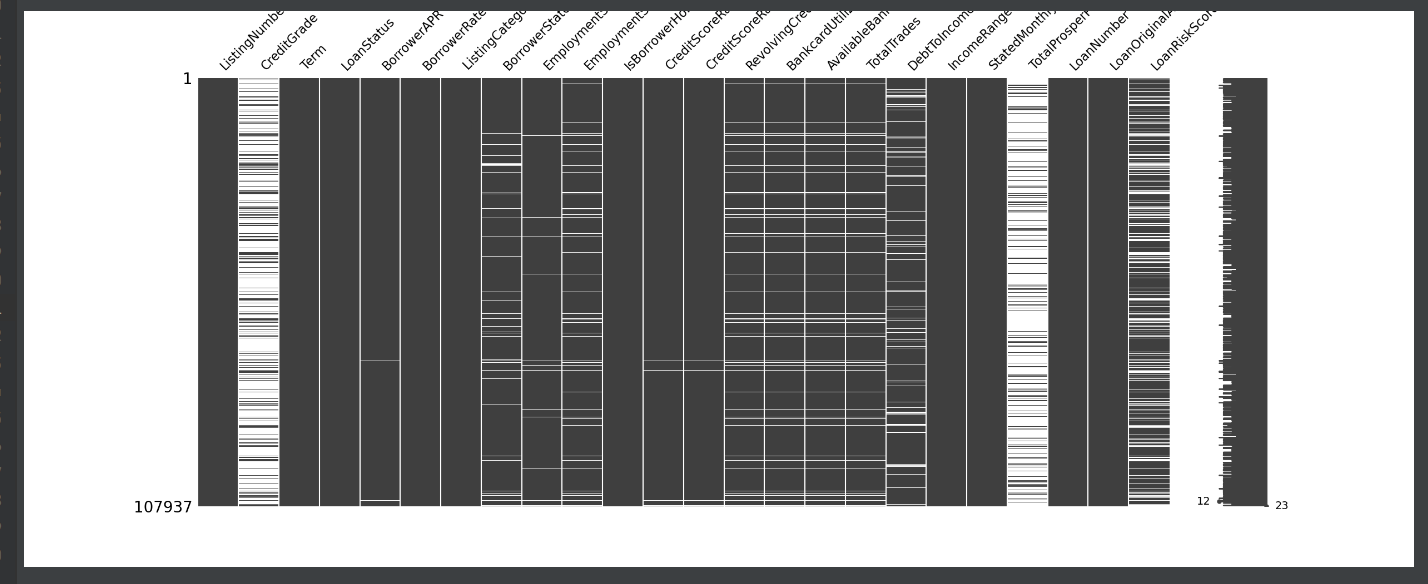
**Regression Models (Milestone 1)**

**Team ID: SC-5**

|  |  |
| --- | --- |
| Section | Name |
| 4 | نوران السيد عبد الحميد فرحات |
| 4 | ياسمين خالد محمد عبد الحميد |
| 4 | مريم هشام عيد حسن |
| 1 | إسراء عزت عبد المنعم علي |
| 2 | حسناء إيهاب عبد الشافي حسن |

**Preprocessing:**

1. Dealing with Null values:

* As feature “CreditGrade” has 80515 null values which represents 74% of the dataset , we have dropped the whole column as well as feature “TotalProsperPaymentsBilled” which has 87050 null values which represents 80% of the dataset, in “figure 1” it shows the features hat have null values and we observed that most values of “CreditGrade” & “TotalProsperPaymentsBilled” are null.
* For the rest of numerical features we first plot “boxplot” for each feature to see if there exist outlier values in each column.
* We observed from the “boxplot” (shown down) of each feature that all of them have outliers, so we replaced the null values of these features with median value of each feature as the mean will not a good choice as there are outliers.
* For the output column “LoanRiskScore” we applied ”boxplot” on it to see if it has outlier values but it doesn’t have outlier values, so we replaced null values in this column with the mean value of this column.
* For the “Categorical features” which are “BorrowerState” and “EmploymentStatus” we have taken the mode of each feature of them and replaced null values in there columns with it.

Figure

Box Plot:

Chart, box and whisker chart

Description automatically generatedChart, box and whisker chart

Description automatically generated

Chart, box and whisker chart

Description automatically generatedChart

Description automatically generatedChart

Description automatically generatedChart, box and whisker chart

Description automatically generated

Chart

Description automatically generatedChart

Description automatically generatedChart

Description automatically generatedChart, box and whisker chart

Description automatically generated

|  |  |
| --- | --- |
| Column name | Why it was dropped |
| “ListingNumber” | Because it represent Load request indexing  (like ID). |
| “CreditGrade” | Because 74% of it contain null values. |
| “TotalProsperPaymentsBilled” | Because 80% of it contain null values. |

1. Dropped columns:
2. Dealing with Categorical data:

convert text or categorical data into numerical data which the model expects and perform better with it.

There are many ways to deal with String in dataset like Label encoding and one hot encoding.

The columns that are String in the dataset and we need to find a way to convert it into numerical data are LoanStatus, EmploymentStatus, BorrowerState, IsBorrowerHomeOwner and IncomeRange. Now we will list the techniques used to deal with the String

1. Label Encoding:

This approach is very simple and popular. It works in a way that is sorting the unique values based on the alphabetical order in the column and gives every unique value a number. Problem with Label encoding is the values are ranked based on the alphabet so that the model can capture that some unique values is more important than the other.(figure 2)

1. One hot encoding

It simply creates additional features based on the number of unique values in the categorical feature. Every unique value in the category will be added as a feature. In this technique you first use the Label encoding to get rid of the string value and then every unique value treated as a new column. This approach is more flexible but is adding too many dummy column which mean too many dummy features. Problems with the one hot encoding that Dummy Variable Trap as the outcome of one variable can easily be predicted with the help of the remaining variables. Which mean that there is a dependency between the independent features and this problem called multicollinearity. To git rid of this problem one of the dummy variables must be dropped.(figure 3)

The proposed way to deal with every categorical column in the dataset

|  |  |
| --- | --- |
| Column name | Proposed way to deal with it |
| LoanStatus | One hot encoder |
| EmploymentStatus | Label encoding |
| BorrowerState | Label encoding |
| IsBorrowerHomeOwner | Label encoding |
| IncomeRange | Label encoding (not based on the alphabetical order but based on the money range)  The higher the money range the higher the value of the number of that range |

Text

Description automatically generated

Figure

Text

Description automatically generated

Figure

1. Feature Selection:

* We used “Heatmap” to visualize the correlation between the features and the target column, so that we can decide which features will we select for our model. (figure 4)
* We used the output graph (figure 5) from “mutual information In Regression” ” to visualize the correlation between the features and the target column, so that we can decide which features will we select for our model.

Chart, histogram

Description automatically generatedA picture containing timeline

Description automatically generated

Figure 5

Figure 4

**Models:**

**Multi\_Linear\_Regression\_Model 1:**

1. We applied all Preprocessing which was mentioned upward but we have little bit changes:

* We have dropped all rows that have null values in “LoanRiskScore” column.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Feature Scaling | Feature selection | Train | Test | MSE | Accuracy | Training Time |
| Not applied | Not applied | 70% | 30% | 2.2754574070765967 | 0.6282723299948736% | 0.07280182838439941s |
| Applied with range (0-1) | Not applied | 70% | 30% | 2.275457407066473 | 0.6282723299965274% | 0.04980897903442383s |
| Applied with range (0-1) | Correlation Applied | 70% | 30% | 2.993740092501304 | 0.5109308459365154% | 0.02194046974182129s |

Calendar

Description automatically generatedFrom the “Heatmap” in “figure 6” we selected the top features that have a correlation value >0.2, so the selected features in order was “BorrowerAPR”, “BorrowerRate”, “CreditScoreRangeLower”, “CreditScoreRangeUpper”, “BankcardUtilization”, “AvailableBankcardCredit”, “LoanOriginalAmount”.

Figure 6

**Multi\_Linear\_Regression\_Model 2:**

1. We applied all Preprocessing which was mentioned upward.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Feature Scaling | Feature selection | Train | Test | MSE | Accuracy | Training Time |
| Not applied | Not applied | 70% | 30% | 2.6567581420801885 | 0.4155242994663443% | 0.8057270050048828s |
| Applied with range (0-1) | Not applied | 70% | 30% | 2.6567581420808835 | 0.4155242994661914 % | 0.13563776016235352s |
| Applied with range (0-1) | Correlation Applied  D | 70% | 30% | 2.831968305241117 | 0.37697879499151166% | 0.5816071033477783s |
| Not applied | Correlation Applied  D | 70% | 30% | 2.831968305240221 | 0.37697879499170883% | 0.23137497901916504s |

From the “Heatmap” in “figure 7” we selected the top features that have a correlation value >0.2, so the selected features in order was “BorrowerAPR”, “BorrowerRate”, “CreditScoreRangeLower”, “CreditScoreRangeUpper”, “BankcardUtilization”, “AvailableBankcardCredit”, “LoanOriginalAmount”. Calendar

Description automatically generated

Figure 7

**Multi\_Linear\_Regression\_Model 3:**

1. We applied all Preprocessing which was mentioned upward except:

* We have Encoded non-categorical values of 'BorrowerState' & 'EmploymentStatus' with normal “LabelEncoder”.
* We have Encoded non-categorical values of 'LoanStatus', 'IsBorrowerHomeowner','IncomeRange' with “one\_hot\_encoder”.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Feature Scaling | Feature selection | Train | Test | MSE | Accuracy | Training Time |
| Not applied | Not applied | 70% | 30% | 2.8073626162316064e-24 | 100 % | 0.1795191764831543s |

**Random Forest Regressor model:**

A Random Forest is an ensemble technique capable of performing both regression and classification tasks

1. .We applied all Preprocessing which was mentioned upward.
2. We also applied feature encoding on columns :'LoanStatus', 'BorrowerState', 'EmploymentStatus', 'IsBorrowerHomeowner', 'IncomeRange'.
3. We have applied mutual information technique which Estimate mutual information for a continuous target variable which measures the dependency between the variables. It is equal to zero if and only if two random variables are independent, and higher values mean higher dependency.
4. We have used Random forest regressor with parameters : n\_estimators=60, max\_depth=16.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Feature Scaling | Feature selection | Train | Test | MSE | Accuracy | Training Time |
| Not applied | Not applied | 75% | 25% | 0.7492372949007975 | 0.834591478123209% | 32.34708547592163s |
| Applied with range (0-1) | Not applied | 75% | 25% | 0.7494977766319867 | 0.8345339717785234% | 31.152504682540894s |
| Not applied | Mutation\_info Applied | 75% | 25% | 0.8259762534485952 | 0.8176498792597459% | 10.991551876068115s |
| Applied with range (0-1) | Mutation\_info Applied | 75% | 25% | 0.8262248362977701 | 0.8175949998217817 % | 10.79414987564087s |

From “figure 4” we found that Coulmns “BorrowerState”, “EmploymentStatusDuration” ,“ IsBorrowerHomeowner” ,“ RevolvingCreditBalance”,” BankcardUtilization”,” AvailableBankcardCredit”,” TotalTrades”,” DebtToIncomeRatio”,” IncomeRange”,” StatedMonthlyIncome”,” Term” have less correlation with “LoanRiskScore”, so we dropped them.

Note: Feature selection is applied in “Model 3.py” and not applied in “Model 1.py”

**Conclusion:**

Applying different ways in pre-processing affects the accuracy of the model even if we used the same model.

The problem was how to handle different type of non-categorical data in Encoding and how to handle null values in the best way to achieve least MSE.

We handled null values problem of each column based on its nature for example, we dropped the columns that has large number of null values and replaced null values in other columns with its “mean” or “median” or “mode” according to the conditions that we have mentioned earlier.

We handled the problem of non-categorical data by using two techniques which are “Label Encoding” & “One hot encoding”, using each of theses techniques depended on the nature of the column.