2.0.0.0. **Phase 2: EDA and Feature Extraction**

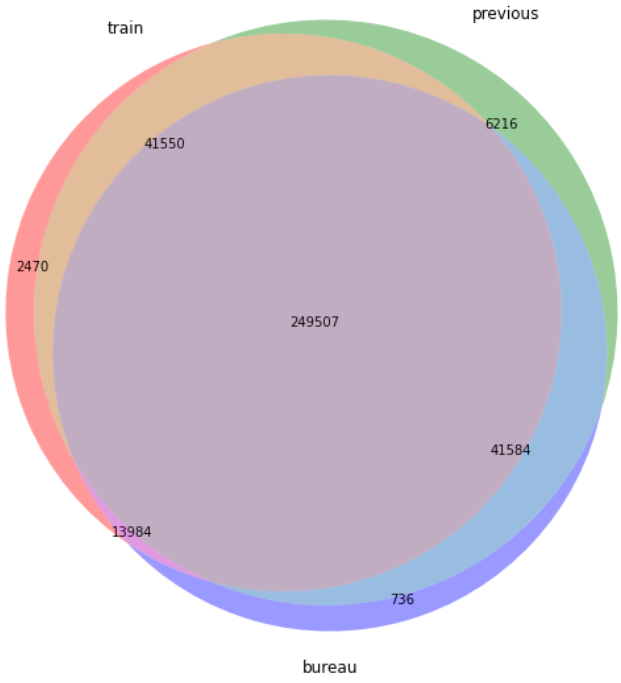
The ipynb file corresponding to this phase is divided into sections. This phase 2 documentation has to be read along with the ipynb file for correlation of different terminologies and outputs. Moreover, only a few visualisations are reproduced here just to give a glimpse. Detailed visualisations can be found in the ipynb file. The description given here follows the same section wise approach as the ipynb file.

2.1.0.0. Common commands: In this section google drive is mounted for accessing data files, important packages are installed and relevant libraries are imported. Three custom functions - one for dataframe optimisation, other for plotting group lot and last for plotting pie chart - are also defined.

2.2.0.0. Data set level analysis: This is high level analysis performed on the csv files provided as input. This analysis helps in getting a general overview of data and deciding the kind of operations to be performed during data preparation.

2.2.1.0. Number of data points are calculated for application_train, application_test and their overlap & exclusion with datpoints in bureau and previous_application.

2.2.2.0. It is observed that 14.31% of applications from application_train are not available in bureau. 5.35% of applications from application_train are not available in previous_application. Overall 81% of applications from application_train are available either in bureau or in previous_application. The overlaps are indicated through a venn diagram in the ipynb file corresponding to this stage of the project.An example of actual venn diagram is as under:



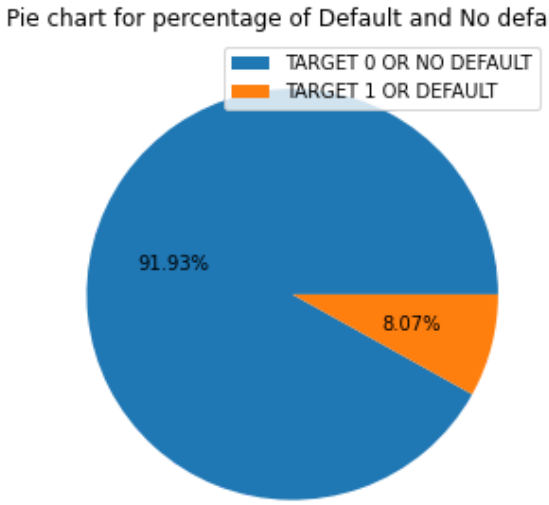
2.2.3.0. It is observed that 13.18% of applications from application_test are not available in bureau. 1.94% of applications from application_test are not available in previous_application. Overall 85.31% of applications from application_test are available either in bureau or in previous_application. The overlaps are indicated through a venn diagram in the ipynb file corresponding to this stage of the project. It is decided that features/columns from bureau and previous_application shall be added to application_train and application_test as a part of feature engineering to get more features.

2.2.4.0. It is observed that 67 columns of application_train and 64 columns of application_test have null values. In some of the columns more than 50% of the values are missing.

2.2.5.0. All the columns shall be retained and missing values shall be tackled using imputation.

2.3.0.0. Univariate and multivariate analysis: With the domain specific knowledge collected during 1st phase of this project, univariate and multivariate analysis are performed on select features. During multivariate analysis the common feature is TARGET value as the ultimate business objective is to reduce default. These univariate and multivariate analysis will give us an insight in the percentage of default among various categories/groups. However, the decision cannot be based only on these analyses alone. Overall decision shall be taken based on final model output and its interpretation. Group plot and pie charts are plotted in the ipynb file. A few of them are reproduced here.

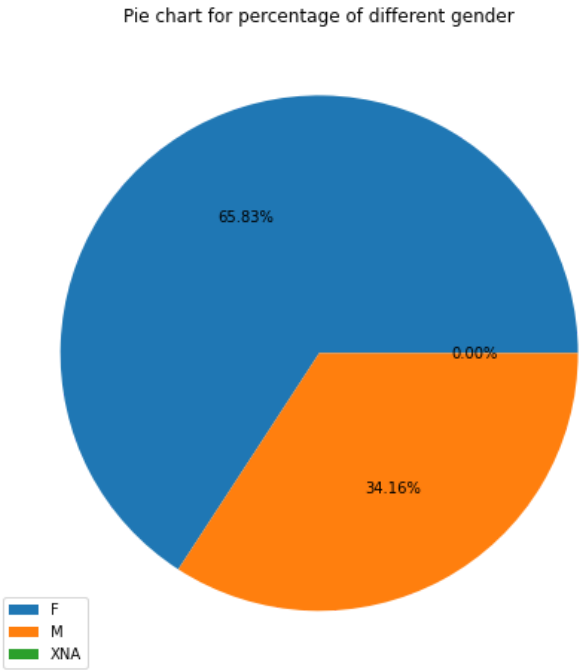
2.3.1.0. An analysis of default indicates that 8.07% of applicants in application_train are defaulters. From a business perspective, this percentage needs to be reduced. From a machine learning perspective, the dataset is imbalanced.



2.3.2.0. An analysis of family status of applicants and default indicates that maximum applicants are married and percentage of default is lower among married people. From a business point of view, married people should be further targeted for loans.
Default among widows is also very low however, the percentage of widow applicants is low. This group can be targeted and further analysed till there is a substantial population in this group.
Civil marriage and Single / not married categories are where focus should be to reduce the percentage of defaulters. The loan process may put an extra check for these groups.

2.3.3.0. An analysis of the type of loans and default indicates that the percentage of cash loans is much higher than revolving loans. The percentage of default is lower in case of revolving loans. Thrust should be laid on disbursing revolving loans, which generally have variable rates of interest.

2.3.4.0. An analysis based on gender and default indicates that the number of male applicants is double the number of female applicants. This result also has a social connotation apart from business impact. It can be concluded that more number of male are earning compared to females and a gender disparity exists. Policy makers will have the responsibility to remove this disparity or evaluate their steps taken earlier.



As the percentage of female defaulters is lesser than male defaulters, this calls for promotion of loan among females.
From this preliminary analysis, it seems that gender is an important parameter. However, the decision to retain or remove this parameter shall be taken after mathematical analysis of data in the feature selection section.

2.3.5.0. For conducting analysis based on income, the incomes data is divided among bins of 10 percentile each. Group plot for number and percentage of defaulters for each bin is plotted. There is not much significant disparity among different income groups except for 135000.0-147150.0 where loan applicants are very low.

Loan applicants are more in lower income groups compared to higher income groups. Loan defaulters are also leeser in higher income groups compared to lower income groups.

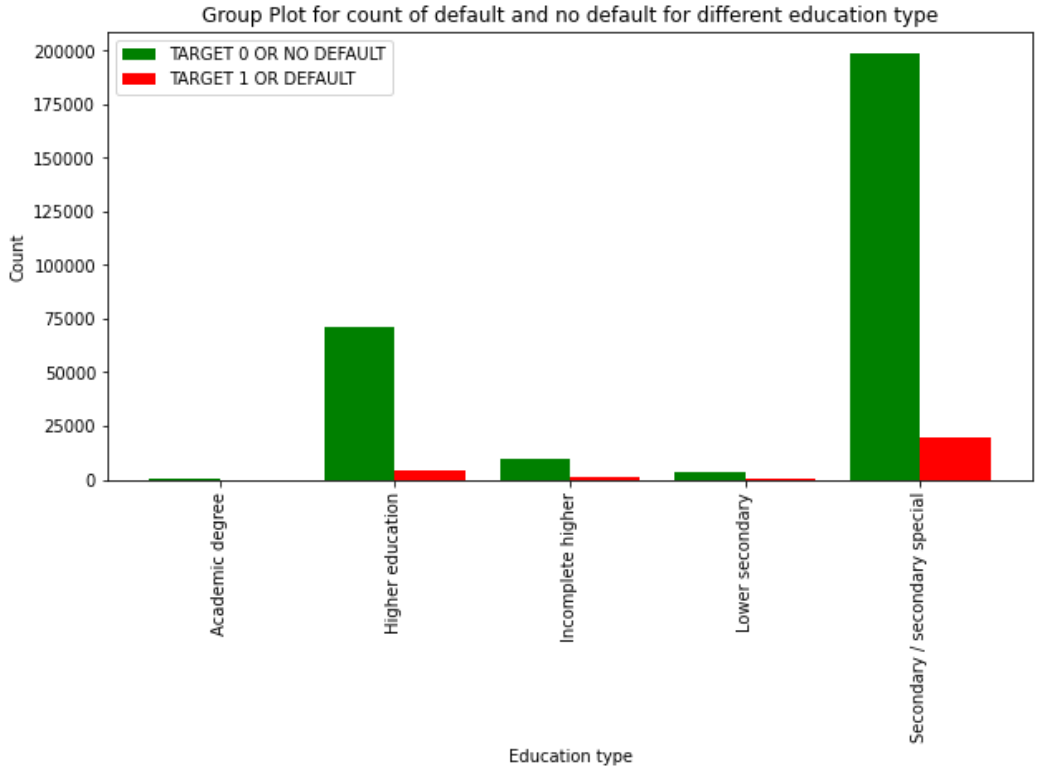
Two particular income groups with higher percentage of applicants are 112500.0-135000.0 and 180000.0-225000.0. This data also indicates a few inordinately high income applicants. These data points may be outliers which will be dealt separately in the outlier detection and removal section.

2.3.6.0. An analysis based on income type indicates huge disparity among the groups with different income types.

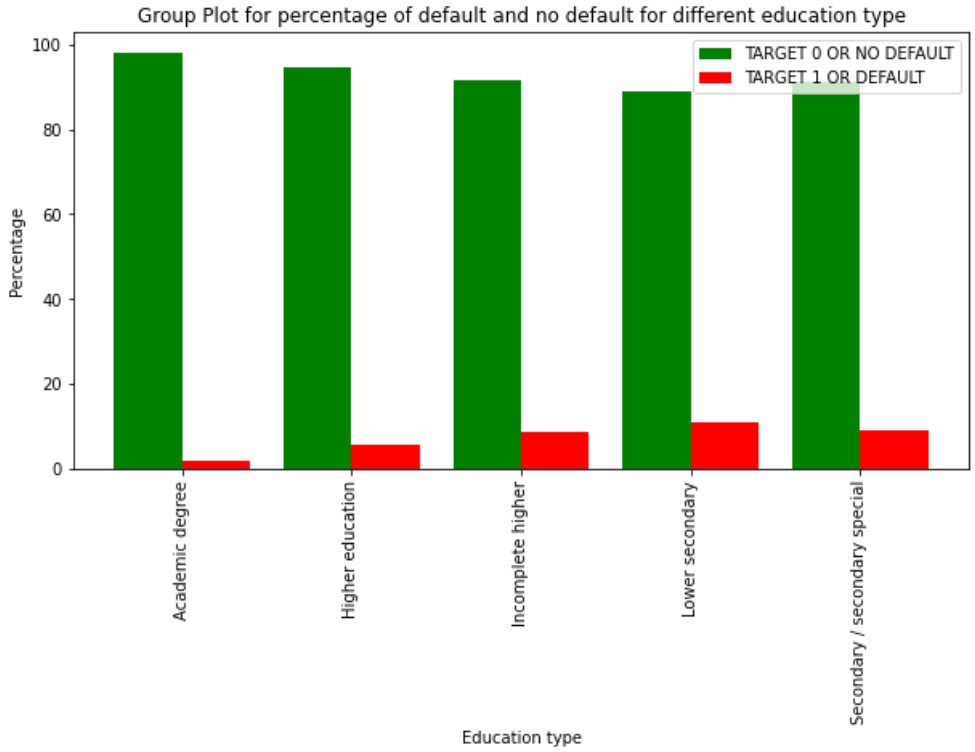
The default rate is very low among pensioners and government servants. They become a target group for promotion of loans and further evaluation of results. Groups with lower numbers of applicants can be targeted for loans and results shall be evaluated after some time.

From this preliminary analysis, it seems that income type is an important parameter. However, the decision to retain or remove this parameter shall be taken after mathematical analysis of data in the feature selection section.

2.3.7.0. An analysis based on education type indicates huge disparity among the groups with different education types.



It is observed that as the level of education increases, the percentage of default decreases. Education is in general correlated to income. Hence, this observation suggests that groups with higher levels of education should be targeted for loans.
From this preliminary analysis, it seems that education is an important parameter. However, the decision to retain or remove this parameter shall be taken after mathematical analysis of data in the feature selection section.



2.3.8.0. 31.35% data is missing in case of occupation type. Among the available data, labourers constitute the highest percentage of defaulters.
Groups with low percentage of default e.g., Accountants, Core staff, HR staff, High skill tech staff, IT staff etc., should be targeted for disbursing loans.
Accountants have the lowest percentage of default and also low percentage of application. They form an important target group.
From this preliminary analysis, it seems that occupation type is an important parameter. However, the decision to retain or remove this parameter shall be taken after mathematical analysis of data in the feature selection section.

2.3.9.0. An analysis based on day of the week indicates, most loan applications are done on weekdays. Number of loan applications decreases on Saturday and becomes very less on Sunday.
This data is very useful for staff management as more staff is required on weekdays compared to weekends.

2.4.0.0. Feature Engineering: Two types of feature engineering are performed. They are performed on both application_train and application_test.

2.4.1.0. Three new ratios are added. These ratios are added based on the domain knowledge acquired during the 1st phase of this project. These ratios are considered by financial institutions for taking decisions pertaining to loans. These ratios help financial institutions to decide whether to give loans or not and how much loan to sanction. These ratios measure the loan repayment capability of applicants. They are:

2.4.1.1. Debt-to-Income Ratio - This is the ratio of loan annuity (AMT_ANNUITY) and income (AMT_INCOME_TOTAL) of the applicants.

2.4.1.2. Loan-to-Value Ratio - This is the ratio of loan amount (AMT_CREDIT) and price of the goods for which loan is given (AMT_GOODS_PRICE) to the applicants.

2.4.1.3. Loan-to-Income Ratio - This is the ratio of loan amount (AMT_CREDIT) and income (AMT_INCOME_TOTAL) of the applicants.

2.4.2.0. Features are merged from bureau and previous_application: Based on the dataset level analysis, it is decided to add features from bureau and previous_application to application_train and application_test. The columns of bureau which are common in application_train/application_test are renamed by appending _BUREAU to common column names of bureau. The columns of previous_application which are common in application_train/application_test are renamed by appending _PREVIOUS_APPLICATION to common column names of previous_application.

Left merge is performed for merging bureau and previos_application data with application_train and application_test. Merge is performed on column named SK_ID_CURR. SK_ID_BUREAU and SK_ID_PREV columns are dropped.

While merging numerical data, the mean of all the data corresponding to each unique SK_ID_CURR in each column is calculated and merged with application_train and application_test against corresponding SK_ID_CURR.

While merging categorical data, the categorical data is 1st one hot encoded and then the median of all the data corresponding to each unique SK_ID_CURR in each column is calculated and merged with application_train and application_test against corresponding SK_ID_CURR.

A track of columns originally having numerical values and categorical values is kept for further use. A target data is maintained which consists of values from the TARGET column of application_train.

At the end of this stage we find that there are 322 columns in application_train_final (prepared by merging bureau and previous_application data with application_train) and 321 columns in application_test_final (prepared by merging bureau and previous_application data with application_test). Application_train_final has an additional column because of the presence of the TARGET column.

Note: Only bureau and previous_application are merged with application_train and application_test mainly because of limited resources (RAM limitations encountered in Google Colaboratory). This also reduces complexity. However, with no dearth of resources, it is preferred to merge other data tables also. Finally, the most important features can be selected during feature selection.

2.5.0.0. Data Preparation: One hot encoding, imputation and standard scaling: In this section, data is prepared for mathematical operations.

2.5.1.0. One hot encoding: Before performing one hot encoding on remaining categorical columns, application_train_final and application_test_final are vertically concatenated. After one hot encoding is performed, slicing of data is done to obtain application_train_final_ohe and application_train_final_ohe. application_train_final_ohe is one hot encoded data of application_train merged with bureau and previous application. application_test_final_ohe is one hot encoded data of application_test merged with bureau and previous application.

2.5.2.0. Imputation: Median imputation is performed on numerical columns. These numerical columns are those which were numerical in the original datasets (application_train, application_test, bureau and previous).

2.5.3.0. Standard scaling: Standard scaling is performed on numerical columns. These numerical columns are those which were numerical in the original datasets (application_train, application_test, bureau and previous). Standard scaling is not performed on the columns resulting from one hot encoding.

2.5.4.0. X_train_final, X_validate_final and X_test_final are obtained as prepared data after performing one hot encoding, imputation and standard scaling. These are the data split from application_train. application_test_final_ohe_combined is obtained as prepared data corresponding to application_test. y_train, y_validate and y_test are target values corresponding to X_train_final, X_validate_final and X_test_final respectively. application_test_final_ohe_combined is also obtained which is prepared data corresponding to appliucation_test merged with bureau and previous_application.

2.5.5.0. All these data are saved as csv files with the same name for further use. These data are saved to create a restore point. Any further action can be performed by reading data from these csv files. This is also important as RAM issues are encountered in Google Colaboratory.

2.6.0.0. Outlier detection and removal

2.6.1.0. When a box plot for data under AMT_INCOME_TOTAL of X_train_final is plotted, it is observed that the box is not even visible and some data points are extending well beyond the whisker limits of the box plot. This indicates the presence of outliers and calls for outlier detection and subsequent removal.

2.6.2.0. Outlier detection is performed using Local Outlier Factor (LOF) based outlier detection module of pyod library. Pyod needs to be installed and updated in Google Colaboratory. Contamination is set at 0.05 which indicates that 5% of the total data points shall be detected as outliers. Total count of outliers and inliers is stored in variables named outliers and inliers respectively. A new dataframe named X_train_final_outlier is created and data from X_train_final is copied to it. A new column named outlier is added to X_train_final_outlier which indicates whether a data point is outlier or not. Target values are also horizontally concatenated with X_train_final_outlier and outlier removal is done based on the values in the column named outlier. X_train_final_outlier_removed and y_train_outlier_removed are the datasets obtained after removing outlier points.

2.6.3.0. Box plot is again plotted for data under AMT_INCOME_TOTAL of X_train_final_outlier_removed. The plot has improved significantly.

2.6.4.0. It is observed that the percentage of defaulters has not changed significantly after outlier removal. Percentage does not look significant but the actual number considering the volume of applicants may be significant. It can even become more significant if the loan amount involved is very high. Hence, it is always suggested to take the opinion of domain experts.

2.7.0.0. Feature selection

2.7.1.0. After the complete preparation of data, 444 columns/features are observed. This is a lot of features to deal with. It is quite possible that many of these features will not contribute towards model performance but increase the complexity. Hence, feature selection is performed.

2.7.2.0. XGBoost based and Gradient Boosting based feature selection are performed. Top 20 features with their feature importance are plotted in bar graph for both selection criteria.

2.7.3.0. Common features from top 225 features based on both selection methodology are selected. Thus 176 features are selected. It is observed that the ratios created during feature engineering figure among selected features.

2.7.4.0. Heat map is generated for 20 selected numerical features. It is observed that only a few features are strongly correlated to other/others. From this perspective also feature selection is correctly done.

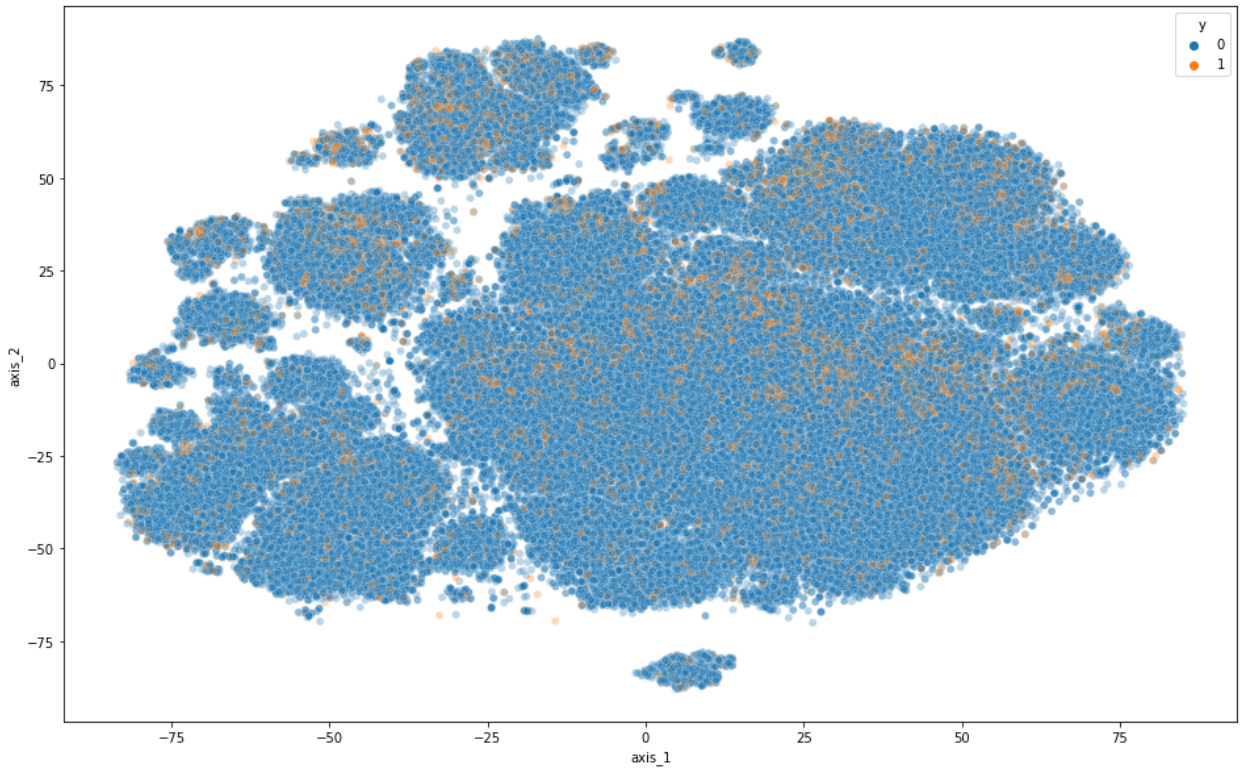
2.7.5.0. Based on these features selected, X_train_final_feature_selected, X_validate_final_feature_selected, X_test_final_feature_selected are defined each with 176 selected features. y_train_final_feature_selected, y_validate_final_feature_selected and y_test_final_feature_selected are corresponding datasets with target values. application_test_final_feature_selected is also defined with selected features.

2.7.6.0. All these datasets are saved as csv files with the same name. This is done to create a restore point so that further actions can be done by importing data from these csv files.

2.8.0.0. Perform TSNE

2.8.1.0. TSNE is performed with some values of perplexity and iterations. A typical render with 30 as perplexity and 1000 as iteration value takes 1 hour and 30 minutes on Google Colaboratory. This gives an indication of the scale of data we are dealing with.

2.8.2.0. Two plots - 1 with perplexity as 30 & 1000 iterations and other with perplexity as 50 and 1000 iterations - are retained in ipynb file. TSNE scatter plot with perplexity as 50 and 1000 iterations is reproduced here. The two axes are 2 dimensions representing all the features.



Some clusters are obtained but the 2 target values are mixed. Separability shall be further evaluated after model training. Different models shall be tried.

2.9.0.0. Conclusion

2.9.1.0. All the relevant files can be accessed through the following link: https://drive.google.com/drive/folders/1evFZRwFWh4zkR9CiT46llB9PlaXFLfLA?usp=sharing

2.9.2.0. This is a very important and lengthy phase as a lot of hits and trials need to be done. The datasets required to be fed to the machine learning model get prepared at this stage.

2.9.3.0. Useful mathematical and business insights are obtained in this phase. Visualisations obtained at this stage may be directly useful to managers.

2.9.4.0. Limitation of resources is a constraint at this stage as is clearly evinced by RAM limitation of Google Colaboratory. At this stage the importance of computing power is also understood.

2.9.5.0. It is important that while dealing with any machine learning project, presence or guidance of domain experts is highly valuable. Domain knowledge is especially required for feature engineering. It is also helpful in deciding whether to retain a column/parameter/feature with a lot of missing values. Domain knowledge is useful in deciding whether to do mean imputation or mean imputation.

2.9.6.0. In the section named ‘Common commands’ in the ipynb file, a function named df_size_optimizer() is used. This is very effective in optimising the size of dataframe obtained by importing data from csv files. The due credits for this function is given in the comment before the start of the function.