Data Preprocessing – purpose of each function blocks

Standardizing levels that convey similar information to common attributes in order to reduce the number of similar categories in 'purpose'

The column 'earliest\_cr\_line' describes the date when the first credit line was established. Usually, longer the credit line held by one, the more desirable he/she will be as a borrower. This feature will therefore be more useful if converted to a measure of how long one has held a credit line. To do this, convert the column from object to datetime and calculate the time duration from the establishment of the first credit line to the current date

Drop columns missing more than 30% of the data

Due to high correlation between the fico range variables, we extract the average. This avoids information loss while taking care of multicollinearity

Missing value imputation - fill with '0' and mode imputation of 'emp\_length'

Remove the 'months' in column 'term' to convert to numerical by splitting the number from the string

Clubbing 'any' and 'none' with 'other' and leaving the others as is in 'home\_ownership'

Feature scaling of numerical columns that have a large range

Group address states into Regions

Impute the remaining missing values - Median imputation for numerical features and Mode imputation for categorical features

Encoding the dependent/target variable - this column contains text values that need to be converted to numerical values for the model to be trained

Removing features which aren't loan amount, irrelevant features, post-loan application features and the target

Create dummy Variables of all the categorical features. This function removes one of the levels to avoid linear dependency using parameter 'drop\_first'

Grouping all the above functions into a main function to create efficiency in the flow of code structure

Enter the path to the raw dataset, call the main function that performs all the necessary prepping and save the final preprocessed data into a CSV file

Sampling Data

A fraction of data is extracted due to dataset being large and for saving computation time

Class Imbalance Check

Analysing target variable to check for imbalance among classes

Oversampling Train Set

Post splitting, train sets are oversampled using SMOTE to mitigate class imbalance issues

Default Models

Fit and predict using Logistic Regression, Decision Tree Classifier, Random Forest Classifier, Adaboost Classifier and Gradient Boosting Classifier

Generating evaluation metrics to enable comparison of models’ performance and to infer the best of all

Hyperparameter tuning of the models to discover the best set of parameters for modeling

Saving the best performing model as a pickle file in the CWD