1 Summary of Features

Lending club dataset is a large dataset with 77160 records data gathered by 108 columns(features). Data with different data types are separated as numerical and categorical features. Out of 108 feature records, 93 records as numerical features and 15 are categorical features (pandas, 2008). Here, date type features come under numerical features. Loan status is our target feature, which is a base feature for the implementation of the KDD process. Loan data is categorized into 7 loan status types. They are Fully Paid, Charged Off, Current, Late (16-30), Late(31-120), In Grace Period, and Default. We can observe the difference in the number of applicants falls in various types of loan status in figure 1. On seeing this we can notice, there is a class imbalance in a target column.

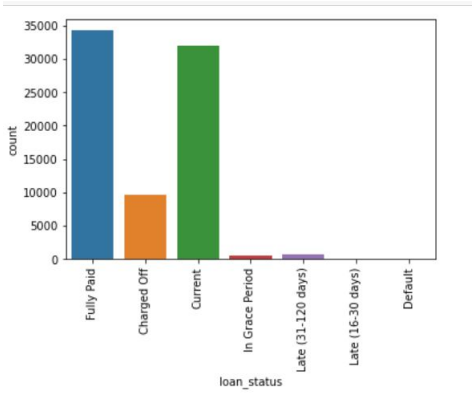

Figure 1: Visualisation of various loan status of total loan data

2 Data Preprocessing

Data Pre-processing involves the following tasks:

- Handling missing values
- Feature selection
- Encoding the categorical features
- Correlation of variables/features
- Imputation of missing values

- Train and Test Split
- Outliers detection
- Class Imbalance

2.1 Handling missing values

Handling missing values involves finding out the more percentage of missing data in features and removing those features from the dataframe. From the raw dataset, on applying the isnull().sum() (stack overflow, 2021) we have found 17 features that contain more than 60 percent of missing values. From figure 2, we notice that these 17 features are dropped from the dataframe using dropna(). Visually from the figure 3 and 4, we find that the missing values have been blunt. Thus, the dataframe has been reduced to 91 features.

```
Percentage of Missing data for the column:
 annual_inc_joint                             87.840693
verification_status_joint                     87.844581
hardship_reason                               94.179551
hardship_type                                 94.179551
hardship_status                               94.179551
                                                 ...
deferral_term                                 94.179551
hardship_amount                               92.072215
hardship_payoff_balance_amount                92.072215
hardship_last_payment_amount                  92.072215
orig_projected_additional_accrued_interest    92.238106
Length: 17, dtype: float64
```

Figure 2: percentage of missing data of features

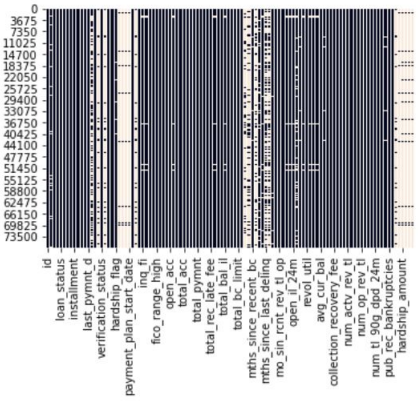

Figure 3: Before dropping features

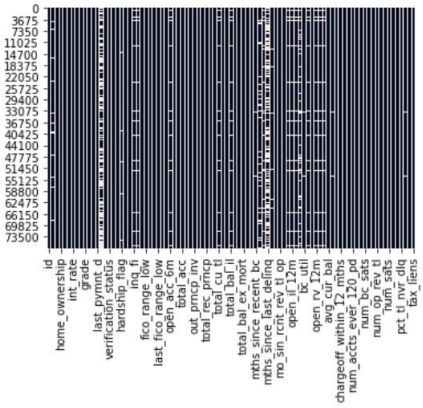

Figure 4: After dropping features

2.2 Feature selection

Feature selection has an intrinsic impact on the KDD process. The outcome from this process is completely based on features that are selected. From the 91 features, due to the high rate of unique values, the id, nextpymntd, and emptitle features can be dropped as it has no significance for the process. Before transforming all the categorical features to numerical features i.e. Encoding, all the selected features must be filled with dummy values. The flag that decides whether or not the borrower is on a hardship plan is termed as the hardshipflag feature. The hardshipflag feature contains NaN values that do not allow the transformation of values from string to numeric. So, before encoding NaN values of the hardshipflag feature they are filled with dummy values. Figure 5, shows the dummy values in the place of NaN values.

```
LendingClubLoan_data_new['hardship_flag'] = LendingClubLoan_data_new['hardship_flag'].fillna('1111').astype(str)
```

```
LendingClubLoan_data_new['hardship_flag'].value_counts()
```

```
N       71962
Y        3570
1111     1627
Name: hardship_flag, dtype: int64
```

Figure 5: : filling NaN values for hardship flag

2.3 Encoding the categorical features

After dropping off 60 percent content of missing value features and features selection for encoding, 12 features are undergoing the encoding process. Features like home ownership, grade, verification status, purpose, application

type, pymnt plan, term, initial list status, hardshipflag, issued, last pymntd undergo ordinal encoding to convert the categorical features into numerics. It assigns a unique value to each categorical value starting from 0. An ordinal encoding is preferred over label encoding as it targets converting all the features, unlike a label encoder for converting target variables alone.

Also, we haven't chosen One-Hot Encoding in our process, since it creates an additional feature that increases the data frame size.

	application_type	home_ownership	loan_amnt	int_rate	term	installment	grade	annual_inc	verification_status	pymnt_plar
count	77159.000000	77159.000000	77159.000000	77159.000000	77159.000000	77159.000000	77159.000000	7.715900e+04	77159.000000	77159.(
mean	0.121593	2.430747	15156.519978	0.131401	0.315699	443.024291	1.603585	7.836316e+04	0.887155	0.(
std	0.326817	1.417764	9749.195616	0.051024	0.464797	279.917146	1.188673	8.492549e+04	0.782882	0.(
min	0.000000	0.000000	1000.000000	0.053200	0.000000	7.610000	0.000000	0.000000e+00	0.000000	0.(
25%	0.000000	1.000000	7800.000000	0.099300	0.000000	239.520000	1.000000	4.500000e+04	0.000000	0.(
50%	0.000000	3.000000	12175.000000	0.126200	0.000000	368.900000	2.000000	6.500000e+04	1.000000	0.(
75%	0.000000	4.000000	20000.000000	0.160200	1.000000	591.285000	2.000000	9.400000e+04	2.000000	0.(
max	1.000000	4.000000	40000.000000	0.309900	1.000000	1717.630000	6.000000	1.099920e+07	2.000000	0.(

8 rows × 83 columns

Figure 6: : conversion of features

Figure 6, shows the statistical outcome of categorical features that are completely converted to numerical features. After converting categorical values of hardshipflag features, all "1111" are again converted back to np.nan. earliestcrline is changed over into Datetime information type by passing year as a numerical value.

2.4 Correlation of variables/features

After encoding and handling missing values, we are leftover with 88 features for KDD processing. With reference from (ALDON, 2017), 22 features are highly correlated with more than 80 percent of correlation out of 88 features. Extract those features from the dataframe by using corr(). Those features are dropped from the dataframe for better performance. After removing highly correlated features now the total number of features is filtered to 62 features.

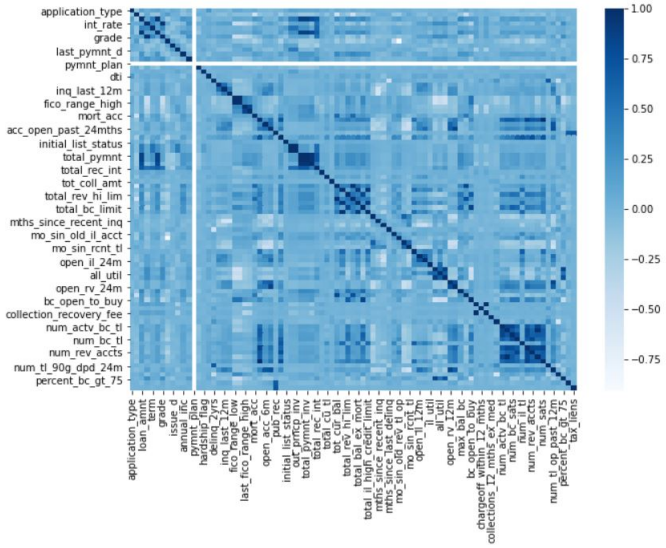

Figure 7: : correlation of variance

2.5 Imputation of missing values

The final step to handle missing values is filling those cells by using the imputation process. KNNImpute is used to impute missing values. The KNN imputer applies K-Nearest Neighbors to swap the missing values in the dataframe with the average value of the nearest neighbors in the set. In this, the parameter is set to 2 nearest neighbors.

```
In [39]: missing_val_count_by_column = (LendingClubLoan_data_new.isnull().sum())
         print('Missing columns for the data:\n',missing_val_count_by_column[missing_val_count_by_column>0])

         Missing columns for the data:
          Series([], dtype: int64)
```

Figure 8: : results after KNNImpute

2.6 Train and Test Split

Data must be divided into train and test sets. Outlier detection is applied after splitting the data to ensure that the missing values are rightly predicted. Here the data is divided into 80 percent of train data and 20 percent of test. figure 9, shows the train data with respected to loan status feature and figure 10, shows the test data with respect to loan status feature.

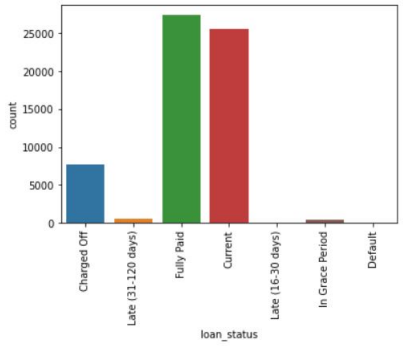

Figure 9: : Visualisation of train data with Loan status

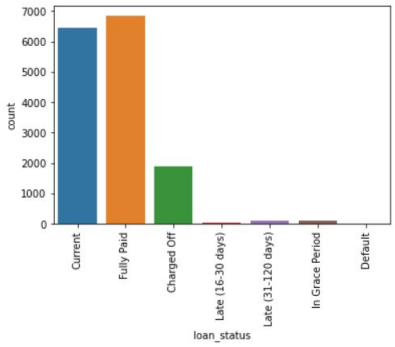

Figure 10: : Visualisation of test data with Loan status

2.7 Outliers detection

Isolation forest algorithm is used for outlier handling which is a part of the unsupervised machine learning algorithm. the performance of outlier detection is faster as it works on less memory and just follows the principle of the decision tree Algorithm. The total number of outliers detected for the train data set is 618 and for the test data set it is 155. The train is shaped to (61109, 61). figure 11, shows the outcomes and data after the removing outliers.

```
Total number of outliers identified is:  618
```

```
mask = preds != -1

Loan_Status_column= train_y[mask]
print (Loan_Status_column.shape)

LendingClubCompleteCleaned_data=train_X[mask]
print(LendingClubCompleteCleaned_data.shape)
```

```
(61109,)
(61109, 61)
```

Figure 11: : Total number outlier detection

2.8 Class Imbalance

Before class imbalance, loan status variables are divided into two categories Good and Bad loans.(stack overflow, 2020) Fully paid and current status are defined as Good loans, where as remaining all status are defined as Bad loans. Before class imbalance, good loans and bad loans undergo ordinal encoding and are converted to numeric values like 1 and 0.

The resample() function will cluster all the observations based on the frequency. Further it would be divided into two sample classes termed as majority and minority classes. Good loans fall under the majority class, where as Bad loans fall under minority class. In this we follow upsample approach, which means that sampling of two classes can be done based on the values of majority classes. The frequency of the class is used to equalise the gap between majority and minority classes. Figure 12, shows the outcome of Good and Bad Loans after the upsampling of variables.

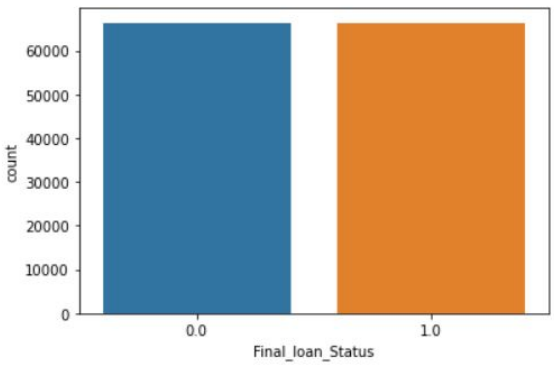

Figure 12: : upsampling of classes

3 Supervised Model Training and Evaluation

The point of Supervised Learning is to think or deduce a mapping function that can be applied to at least one or more input variables, and produce an output variable or result. Classification problems like Decision tree, Support Vector Machines (SVM), K Neighbors classifiers are used for Machine Learning Algorithms.

3.1 Decision Tree Classifier

Decision trees consider multiple algorithms to resolve a node into two or more sub-nodes that raises the similarities of result set. The decision tree takes all the variables to split and later identifies the one that results in most similar sub-nodes.

```
best classifier is: DecisionTreeClassifier(criterion='entropy')  with an accuracy of:  0.9587869362363919
```

```
array([0.70470451, 0.86391913, 0.95878694])
```

Figure 13: : Accuracy comparision of classifiers

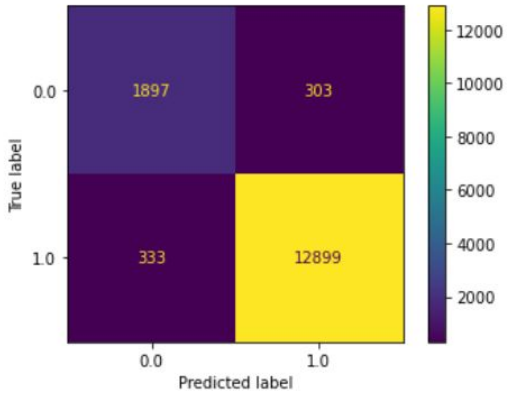

Figure 14: : confusion matrix

we observed from the visualisation of the confusion matrix in the figure 14, 303 false positive values out of 1897 were true positive values and there is 12899 correctly predicted no event values. on the otherside we have noticed 333 are false negatives.

3.2 Explaination

In Support Vector Machines (SVM), the hyperplane that best separates the tags is generated by a support vector machine using these data points. This line serves as a decision boundary. Anything falling on one side will be classified as blue, and anything falling on the other will be classified as red, which will be based on usage of two-group classification problem.

On the other hand a distance based calculation between query in the dataset, selects more frequent labels by considered the specific number of examples (k) closest to the query in the KNeighbor Classifier.

Comparing the three classifiers viz SVM, KNN, and decision tree, Decision tree outputs highest percentage of accuracy that accounts to 95 percent. Also it lends about 90 percent precision for good loans.

4 Un-Supervised learning Algorithms

Cluster analysis, also known as clustering, is a task that require unsupervised ML. It corresponds to finding of natural grouping in data automatically .Unlike supervised learning, clustering analysis only evaluates the input data and look for natural clusters in feature space.

After applying the elbow method on the entire standardised data set, we have observed that there are two clusters. Using PCA with n-components = 2 so that data is decomposed and loan status omitted

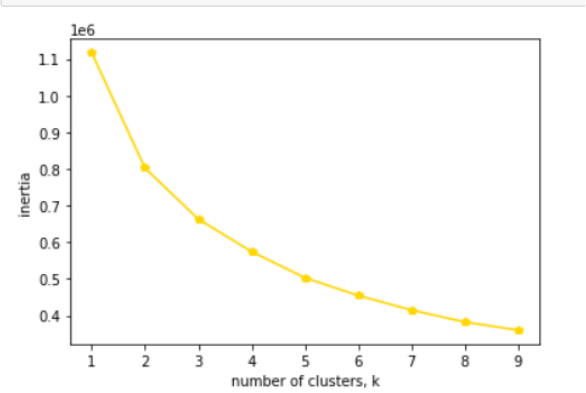

Figure 15: : Visualisation of elbow method

5 K-means Clustering Algorithm

The paper Mourcos (2019) explains the usage of PCA and K-means for Clustering. To find groups that haven't been specifically labelled in the data, the K-means clustering algorithm is used. The business makes assumptions about the types of groups that exist, as well as to classify unknown groups in large data sets.

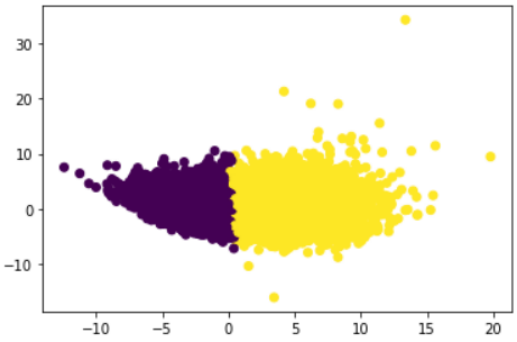

Figure 16: : Visualisation of K means clustering with PCA

6 Conclusion

The Overall View of the complex process of KDD process undergoes Reading data and performing data cleaning and pre-processing actions like handling missing values feature selection, encoding categorical features, detecting and removing corrleated variables, impute the missing values using KNNimpute, spliting the data into train and test data sets and performing class imbalance using resampling. After the cleaning and spliting data, the cleaned data is used to perform Supervised Model Training and Evaluation like decision tree classifier which are then visualized for clear understanding. Finally unsupervised clustering algorithms are applied by removing target column and verifying wheather the application is probably a Good Loan or a Bad Loan, for better understanding we have visualised the difference.

References

ALDON, C. (2017). Drop highly correlated features
https://chrisalbon.com/machine_learning/feature_selection/drop_highly_correlated_features/.

Mourcos, A. (2019). Using pca and k-means for clustering
https://andrewmourcos.github.io/blog/2019/06/06/pca.html.

pandas (2008). pandas.dataframe.select dtypes
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.dataframe.select_dtypes.

stack overflow (2020). select two pandas column with np.select
https://stackoverflow.com/questions/63731684/select-two-pandas-column-with-np-select.

stack overflow (2021). python isnull().sum() handle headers
https://stackoverflow.com/questions/48666721/python-isnull-sum-handle-headers.