Supervised and Unsupervised Learning on Lending Club Data

Muhammad Waqar Ayub Khan, UEA ID. 100334069

May 17, 2021

Abstract

Classification and clustering are difficult tasks that can be used to predict results from various datasets. These are the tasks that make up the majority of data science work. On the lending club dataset, this study uses various KDD actions for classification and clustering, which enable the related institute to identifying whether a loan is potentially a bad loan or a good loan. To obtain reliable results, various machine learning algorithms for classification and clustering were applied to the dataset.

1 Introduction

This report covers two task, first task is to use lending club dataset and predict the loan status of the accounts, as we already know the loan status so this can be done by using different classification algorithms and then we can find the accuracy of our predictions by comparing the predicted results with the actual loan status. Second task is to use clustering algorithms to divide the data points into number of groups, each group represents the similar type of data points in our case loan status. Both tasks consists on Knowledge Discovery of Databases (KDD) steps, each step of KDD has a separate action on dataset which helps in getting more accurate results. In this study various techniques was used on dataset before making predictions such as removing abnormal data points known as Outliers, dealing with missing values, data balancing, dimensionality reduction which are the part of KDD

process. Classification and clustering both involves these steps for accurate predictions, these steps are known as Cleaning and Pre-processing of a data. After getting cleaned and pre-processed data, for classification various algorithms were applied to get predictions and similarly for clustering predictions different clustering algorithms were applied to get accurate clusters. Overall both classification and clustering involve cleaning, pre-processing, modeling, evaluation, each step for both clustering and classification will be discussed in this report.

2 Data / Feature Summary

Lending club data contains 108 different columns and have 77159 different loan account records. Out of 108, 17 columns have more than 60 percent null values, having that much null values might lead to bad prediction so these features were dropped from the dataset and new dataset was created which has 91 columns or features with same number of rows. Figure 1 shows the feature having missing values greater than 60 percent. After dropping these features we have a data with 91 features out of 91, 14 features are categorical, which are shown in table 1. Features named such as id, *emp_title* will be dropped because these two has too many unique values, *pymnt_plan* will also be dropped because it has only one unique value. *int_rate* and *revol_util* will be converted to numeric after removing % symbol from the last similarly *term* will be converted to numeric after removing *months* from last. There are some dates columns or features in the data we can use these dates and can convert those dates into some useful information like missing term or remaining term. Figure 2 shows the features which have been transformed from the dates columns. *loan_status* is a target feature which will be separated from the other columns or features. The remaining categorical features or columns will now be encoded into the numeric so that these features can be dealt by modeling algorithms. These are encoded before the train-test

	Missing_Values	Missing_Percentages
annual_inc_joint	67777	87.840693
verification_status_joint	67780	87.844581
hardship_reason	72668	94.179551
hardship_type	72668	94.179551
hardship_status	72668	94.179551
...	...	...
deferral_term	72668	94.179551
hardship_amount	71042	92.072215
hardship_payoff_balance_amount	71042	92.072215
hardship_last_payment_amount	71042	92.072215
orig_projected_additional_accrued_interest	71170	92.238106

17 rows × 2 columns

Figure 1: Features having more than 6 percent missing values

Categorical Features	
emp_title	28185
home_ownership	5
loan_status	7
int_rate	129
term	2
grade	7
earliest_cr_line	622
issue_d	16
last_pymnt_d	28
next_pymnt_d	9
verification_status	3
pymnt_plan	1
purpose	13
revol_util	1074

Table 1: List of Categorical Features with unique values

split because encoding before split reduce the computation and does not expose the test data to the training model. After feature engineering and encoding the new data has only

one categorical feature which is *loan_status* and it will be remove from data as it is target feature and new data has now 85 columns or features and 77159 records.

	issue_d	last_pymnt_d	next_pymnt_d	missing_term	remaining_term
43302	11/1/2017	5/1/2020	Jun-20	0.0	6.0
10245	12/1/2017	5/1/2020	Jun-20	0.0	31.0
48662	11/1/2017	9/1/2019	NaN	NaN	NaN

Figure 2: Transformation of Dates Columns

3 Classification

In classification the task is to identify the status of the loan whether a loan is a good loan or a bad loan. In the actual data we have seven different status of the loan e-g $fully\_paid$, $Current$, $charged\_off$, $late\_15\_30\_days$, $late\_30\_120\_days$, , $grace\_period$, , $default$. Status like $charged\_off$, $late\_15\_30\_days$, $late\_30\_120\_days$, , $grace\_period$, , $default$ represent the bad loan so these statuses converted into the 1 class which is named as bad_loan. Now our classification problem is converted into three class classification problem $fully\_paid$ which is a good loan, bad_loan which represents the loans which are bad investments by these banks and $Current$ loans which are still under observations. Figure 3 shows the statuses of the loans in the dataset. The working flow for the classification is well explained in the figure 4.

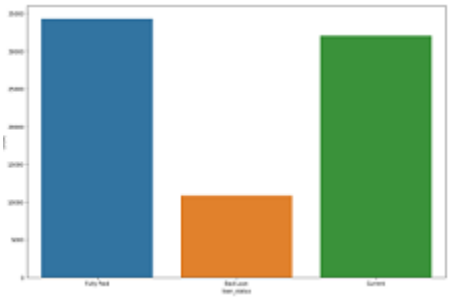

Figure 3: Loan Status Classes

In classification section, train-test split is done just after the feature engineering where we

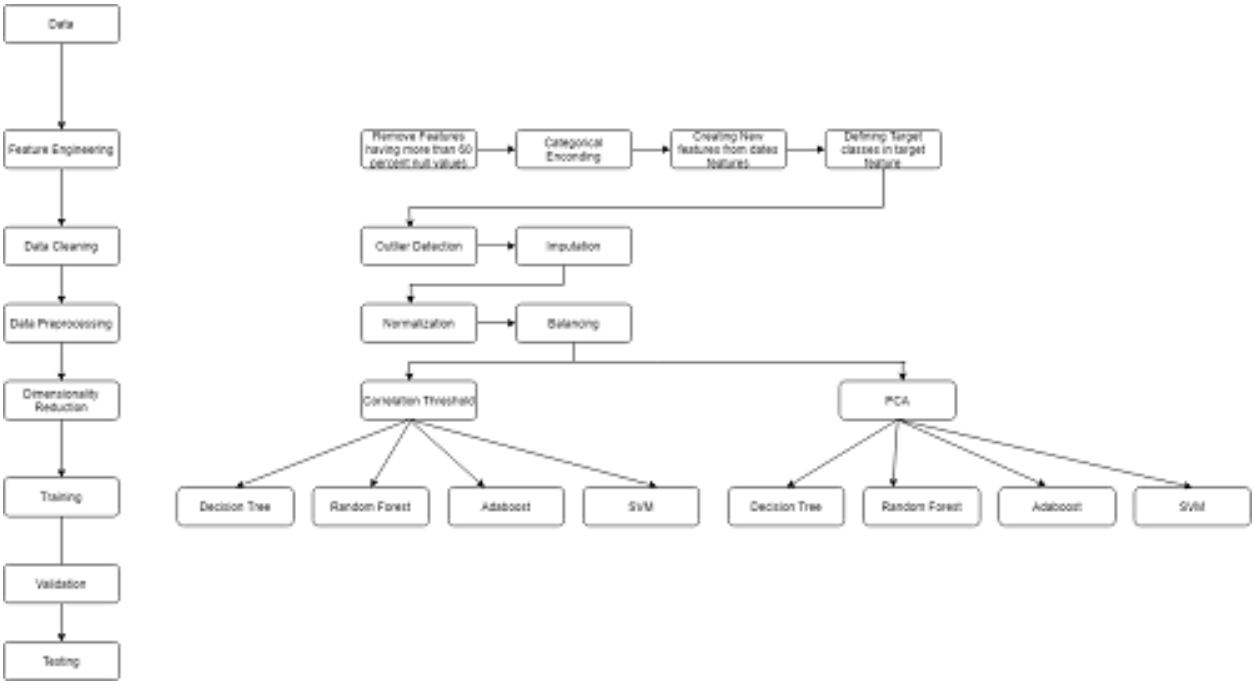

Figure 4: Classification Flow

create some new features like *missing_term*, *remaining_duration* by using dates features in the dataset. Removing test set before the cleaning and pre-processing prevents data leakage, test data remains unexposed to the training model while doing cleaning and pre processing of the training data, cleaning and pre processing on test set is done separately. The steps involve in each phase of the classification is describe below.

3.1 Data Cleaning

Data cleaning is used to eliminate noise, inconsistencies, and inaccuracies from training data. This should allow for the development of a viable classification model using a more comprehensive and representative data set. Unclean data can sometimes influence the classification accuracy of a model in most classification methods. Data cleaning involves outlier handling and imputation of the missing values. In this study outliers were handled first and than the

imputation. If we do imputation first that will create some sort of bias into the data which is not appropriate for model training. But if there are missing values in the dataset outlier detection algorithm won't work so in this study we first make the copy of data, impute the missing values with zero, detect outliers from that copied data find the indexes of the records that were detected as outliers and remove those indexes from the actual data. After removing outliers than some imputation was performed on actual data.

3.1.a Outlier Detection

In a dataset, an outlier is a value that is abnormally far off from other values. In some ways, this definition delegated the decision of what is abnormal to the analyst. Outliers can be detected or dealt with using a variety of methods. Outlier detection approaches must be chosen based on the dataset. If a dataset has too many outliers, we must place those observations under distributions to see what they represent. Another method is to use algorithms to detect outliers and delete them from the dataset. To make things easier in our case, the Isolation forest technique is utilised, which is an outlier identification technique that detects anomalous points and then removes those points from the dataset to clean it up. The result of the isolation forest is that there are more than 6000 outliers in the dataset as the number of outliers seems to be low in number so we can remove those data points.

3.1.b Imputation

Imputation is another important step in data cleaning because if there are missing values in the dataset than if we try to create a model with missing values, most machine learning algorithms gives an error. As a result, you'll need to pick one of the imputation tactics. There are many techniques for the imputation of missing values in the dataset such as dropping the rows, imputing with nearest values, imputing with up and down values or imputing with mean value. For simplicity in our case Mean-imputation is used, because imputing

with mean preserves the mean of that particular feature, the estimate of the mean remains unbiased. There are benefits and disadvantages of mean imputation that are not in the scope of this project.

3.2 Data Transformation

Data transformation is the process of taking raw data, normalising it, balancing it, and transforming it into data that can be put together for analysis. Although dimensionality reduction is a part of data transformation, it is treated as a separate phase in this report since a comparison research was undertaken to obtain more accurate findings with dimensionality reduction. Normalization and balancing of the data are the two steps which were performed in transformation phase which are describe below.

3.2.a Normalization

Normalization is a data transformation method that is regularly utilized in machine learning. Normalization is the way toward changing over the upsides of numeric columns in a dataset to a comparable scale without influencing the ranges of values. Neural Network also uses normalization to make data normalize which reduces computational time. Many machine learning algorithms, such as support vector machines and k means, are sensitive to normalization, although many others algorithm may function without it. Also because dimensionality reduction techniques like Principle Component Analysis (PCA) are sensitive to normalization, we must do normalization before dimensionality reduction.

3.2.b Normalization

Normalization is a data transformation method that is regularly utilized in machine learning. Normalization is the way toward changing over the upsides of numeric columns in a dataset to a comparable scale without influencing the ranges of values. Neural Network

also uses normalization to make data normalize which reduces computational time. Many machine learning algorithms, such as support vector machines and k means, are sensitive to normalization, although many others algorithm may function without it. Also because dimensionality reduction techniques like Principle Component Analysis (PCA) are sensitive to normalization, we must do normalization before dimensionality reduction.

3.2.c Feature Selection/Dimensionality Reduction

Imbalanced data occurs when the amount of observations for all of the classes in a classification dataset is not equal. Many machine learning classifiers struggle with unbalanced training datasets because they are sensitive to the ratios of various classes. As a result, these algorithms prefer the class with the highest set of observations, which might lead to inaccurate results. This can be especially problematic when we are looking for the rare class identification since many algorithms are unable to find sufficient data for learning. We can delete entries from the majority class to balance the data, but this may result in the loss of some crucial information; another option is to add duplicate values, which is also inefficient. Our data is also highly imbalance imputation is necessary otherwise algorithm won't have enough data for minority class for learning, so for this SMOTE technique is used, SMOTE creates new records rather than duplicating the records from the dataset which is very good for highly imbalanced data.

3.3 Dimensionality Reduction

Dimensionality reduction is a critical step in a machine learning project since characteristics that are connected to each other produce data redundancy, which can lead to overfitting during model training. It's critical to eliminate elements that aren't necessary for model training. Both the test set and the train set should have their dimensions reduced. For feature selection, a variety of methods and approaches can be applied, however correlation threshold

and principle component analysis were applied in this study. We can see which features for the model training are essential via correlation, but we only get reduced dimensioned vectors via PCA. Although the outputs from both ways are different, they are both quite impressive. Comparison of the results will be discussed in the report later.

3.4 Training

The term "model training" refers to feeding data into a machine learning system so that it can generate predictions. To obtain predictions, Decision Tree, Random Forest, Adaboost, and SVM are used in this study. These techniques were used twice, the first time with correlation threshold features and the second time using principle component analysis (PCA). GridSearch is used while training these machine learning algorithm so that model gets trained with good parameters this is also known as Parameter Hyper-tuning. The results from both trials are pretty impressive, with over 80% accuracy on both the test and the train set. Table 2 shows the accuracies of the models which are applied on the training set.

Models Result		
	Correlation Threshold	PCA
0 Decision Trees	0.9397	0.8532
Random Forest	0.9309	0.8731
Adaboost	0.9224	0.8649
SVM		

Table 2: Training set accuracies with respect to feature reduction technique

3.5 Validation

Validation on the dataset is done to test if the model will operate with newly discovered data. To do this, we must either split our validation and training sets or do cross-validation on the training set. In this study, cross validation is used on each model to ensure that the

trained model performs as expected. Cross-validation produces above 80 percent outcomes on each iteration of the each model, demonstrating that the trained model will perform well in the test set as well.

3.6 Testing

In testing, we must examine our test data accuracies on trained models; in our study, we feed test data to our trained models to obtain test data predictions. Models trained using PCA provide more than 80 percent accuracy on the test set, whereas models trained with correlation threshold provide more than 90 percent accuracy on the test set. Table 2 shows the accuracy of the test sets on the trained models.

Models Test Set Results		
	Correlation Threshold	PCA
Decision Trees	0.9406	0.8101
Random Forest	0.9428	0.8396
Adaboost	0.9375	0.8545
SVM		

Table 3: Test set accuracies with respect to feature reduction technique

4 Clustering

5 Conclusion

References

Breiman, L. (1996), 'Bagging predictors', *Machine learning* **24**(2), 123–140.

Freund, Y., Schapire, R. E. et al. (1996), Experiments with a new boosting algorithm, *in* 'icml', Vol. 96, Citeseer, pp. 148–156.

Huang, F., Xie, G. & Xiao, R. (2009), Research on ensemble learning, *in* '2009 International Conference on Artificial Intelligence and Computational Intelligence', Vol. 3, IEEE, pp. 249–252.

Wang, W. (2008), Some fundamental issues in ensemble methods, *in* '2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)', IEEE, pp. 2243–2250.

Zhou, Z.-H. (2012), *Ensemble methods: foundations and algorithms*, CRC press.