ANALYSIS OF FEATURE SELECTION TECHNIQUES IN CREDIT RISK ASSESSMENT

Created by:

Deepak Kumar Swain

ABSTRACT

The project focuses on feature selection technique in credit risk dataset. Data Mining is an automated extraction of hidden knowledge from large amount of data. The computational complexity of the data mining algorithms increases rapidly as the number of features in the dataset increases. Real world credit datasets have accumulated large quantities of information about clients and their financial and payment history. Feature selection techniques are used on such high dimensional data to reduce the dimensionality by removing irrelevant and redundant features to improve the predictive accuracy of data mining algorithms. The objective of this work is studying the information gain, gain ratio and chi square correlation-based feature selection method to reduce the feature dimensionality Information gain measure identifies the entropy value of each specific feature. The amount of information gain or entropy is used to decide whether the feature is selected or deleted. Gain ratio applies normalization technique to information gain using spilt information value. The correlation-based feature selection uses heuristic search strategies to estimate how the features are correlated with the class attribute and how they are important of each other. Experiments were conducted on the German credit dataset available at UCI Machine Learning Repository to reduce the feature dimensionality using these feature selection methods.

TABLE OF CONTENTS

ABSTRACT 2
ACKNOWLEDGEMENT 3
TABLE OF FIGURES 5
1. INTRODUCTION 7
1.1 Basic of Feature Selection 7
1.2 Types of Feature Selection 8
1.2.1 Filter Method 9
1.2.2 Wrapper Method 11
1.2.3 Embeded Method 11
1.3 Dataset Description 12
2. Machine Learning Life Cycle 13
2.1 Gathering Data 13
2.2 Preprocess Data 14
2.3 Split and Feature Selection 14
2.4 Modeling 15
2.5 Evaluation 15
3.Flow Of Project 16
4. Data Preprocessing 17
4.1 Information about Dataframe 17
4.1.1 Missing and Unique values 18
4.2 Dealing with categorical data 19
4.3 Normalize numerical features 20
4.4 Outlier Detection 21
5 Feature Selection 22
5.1 Removing Constant & low Variance Features: 22
5.2 Information Gain 23
6. Model Building 24
6.1 Decision Tree 24
6.2 Model Evaluation 25

7. OUTPUT AND ANALYSIS 26
7. CONCLUSION 27
REFERENCES 28

TABLE OF FIGURES

Figure 1: Feature selection Techniques 9
Figure 2: Simple Machine Learning Life Cycle 13
Figure 3: Structure of Dataframe 18
Figure 4: Unique or missing values 19
Figure 5: Dealing with categorical data 19
Figure 6: Normalization 20
Figure 7:Description of numerical feature..... 21
Figure 8: Box plot for credit amount 22
Figure 9: Varience Threshold 23
Figure 10: mutual information score 24
Figure 11: decision Tree..... 25
Figure 12: comaprision with &without feature selection..... 26

1. INTRODUCTION

The tremendous growth in computing power and storage capacity has resulted in the growth of huge databases. Data Mining, also popularly referred to as Knowledge Discovery from Data (KDD), is the automated extraction of patterns from large amount of data. Data mining methods are used in different areas such as business, data management, scientific, engineering, banks, government administration and many other applications. The KDD process consists of an iterative sequence of the steps such as data cleaning, data integration, data selection, data transformation, data mining, pattern evaluation and knowledge representation. Data mining is an essential step in the process of knowledge discovery in data bases, in which intelligent techniques are applied in order to extract useful patterns. In general, data mining tasks can be grouped in two categories: descriptive and predictive. Descriptive mining tasks include characterized general properties of the data. Examples of descriptive mining tasks include association mining and cluster analysis. The inference from the current data is used by the predictive task in order to make prediction. Examples of predictive mining task include classification and prediction. In many practical situations there are far too many features for learning algorithms to handle, some of them are irrelevant or redundant for mining. Hence, it is necessary to select a subset of features which are relevant to use in learning. The performance, robustness and usefulness of data mining algorithms are improved when relatively few and relevant features are involved in the process. Literature shows that irrelevant or redundant features cause the performance of learning algorithms to deteriorate.

1.1 Basic of Feature Selection

There are two famous special methods of dimensionality reduction. The first one is feature extraction where the input data is transformed into a reduced representation set of features, so new attributes are generated from the initial ones. The second category is feature selection. In this category a subset of existing features is selected without a transformation. Generally, feature selection is preferred over feature extraction since it keeps all information about the importance of each single feature while in feature

extraction obtained variables are usually not interpretable. The idea behind feature selection is to reduce the effect of tricky features in the dataset, where tricky and unneeded features include.

1 Irrelevant features are those that can never contribute to improve the predictive accuracy of credit model, where the accuracy is how close a measured value is to the actual or true value. However, the algorithm may mistakenly include them in the model. Removing such features reduces the dimension of the search space and speeds up the learning algorithm

2 redundant features are those that can replace others in a feature subset. They basically bring similar information as other features. For example, a dataset may include two features that provide similar information as date of birth and age. Typically feature redundancy is defined in terms of feature correlation, where two features are redundant to each other if they are correlated.

1.2 Types of Feature Selection

Feature selection is also related to dimensionally reduction techniques in that both methods seek fewer input variables to a predictive model. The difference is that feature selection select features to keep or remove from the dataset, whereas dimensionality reduction create a projection of the data resulting in entirely new input features. As such, dimensionality reduction is an alternate to feature selection rather than a type of feature selection.

We can summarize feature selection as follows.

Feature Selection: Select a subset of input features from the dataset.

Wrapper: Search for well-performing subsets of features.

Filter: Select subsets of features based on their relationship with the target.

Statistical Methods

Feature Importance Methods

Intrinsic: Algorithms that perform automatic feature selection during training.

Decision Trees

Dimensionality Reduction: Project input data into a lower-dimensional feature space.

The image below provides a summary of this hierarchy of feature selection techniques.

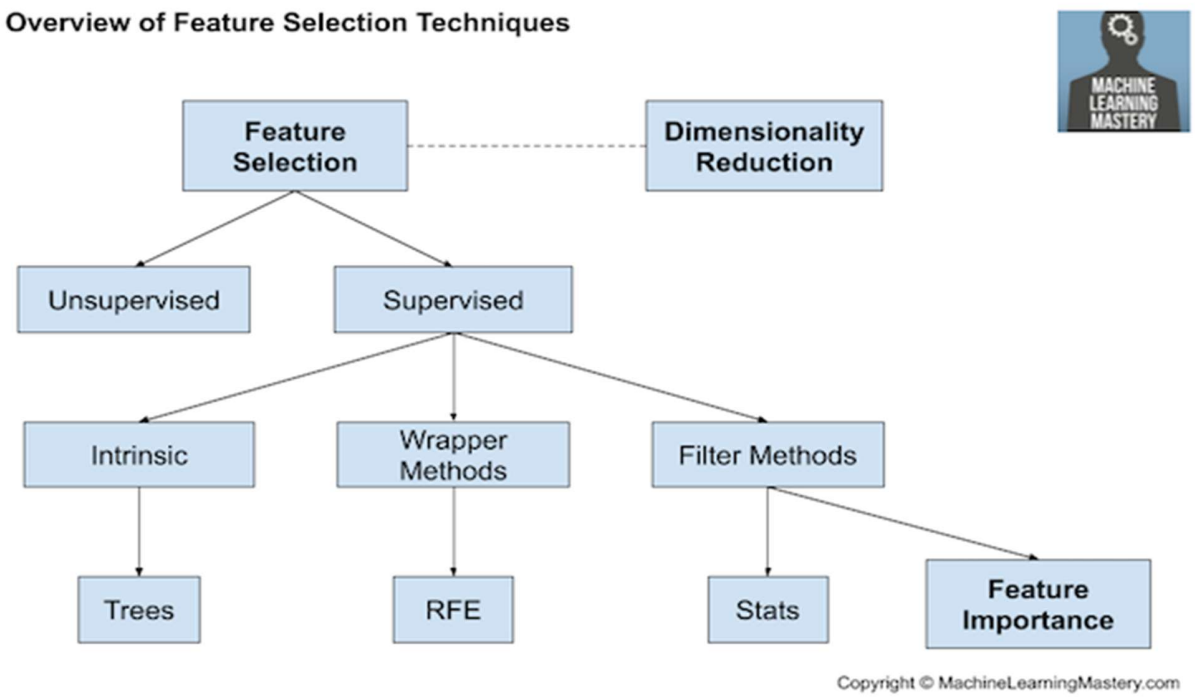


Figure 1: Feature selection Techniques

1.2.1 Filter Methods

Filter methods pick up the intrinsic properties of the features measured via univariate statistics instead of cross-validation performance. These methods are faster and less computationally expensive than wrapper methods. When dealing with high-dimensional data, it is computationally cheaper to use filter methods.

Information Gain

Information gain calculates the reduction in entropy from the transformation of a dataset. It can be used for feature selection by evaluating the Information gain of each variable in the context of the target variable.

Chi-square Test

The Chi-square test is used for categorical features in a dataset. We calculate Chi-square between each feature and the target and select the desired number of features with the best Chi-square scores. In order to correctly apply the chi-squared in order to test the relation between various features in the dataset and the target variable, the following conditions have to be met: the variables have to be categorical, sampled independently and values should have an expected frequency greater than 5.

Correlation Coefficient

Correlation is a measure of the linear relationship of 2 or more variables. Through correlation, we can predict one variable from the other. The logic behind using correlation for feature selection is that the good variables are highly correlated with the target. Furthermore, variables should be correlated with the target but should be uncorrelated among themselves.

If two variables are correlated, we can predict one from the other. Therefore, if two features are correlated, the model only really needs one of them, as the second one does not add additional information

Variance Threshold

The variance threshold is a simple baseline approach to feature selection. It removes all features which variance doesn't meet some threshold. By default, it removes all zero-variance features, i.e., features that have the same value in all samples. We assume that features with a higher variance may contain more useful information, but note that we are not taking the relationship between feature variables or feature and target variables into account, which is one of the drawbacks of filter methods.

1.2.2 Wrapper Methods

Wrappers require some method to search the space of all possible subsets of features, assessing their quality by learning and evaluating a classifier with that feature subset. The

feature selection process is based on a specific machine learning algorithm that we are trying to fit on a given dataset. It follows a greedy search approach by evaluating all the possible combinations of features against the evaluation criterion. The wrapper methods usually result in better predictive accuracy than filter methods.

Forward Feature Selection

This is an iterative method wherein we start with the best performing variable against the target. Next, we select another variable that gives the best performance in combination with the first selected variable. This process continues until the preset criterion is achieved.

Backward Feature Elimination

This method works exactly opposite to the Forward Feature Selection method. Here, we start with all the features available and build a model. Next, we the variable from the model which gives the best evaluation measure value. This process is continued until the preset criterion is achieved.

Exhaustive Feature Selection

This is the most robust feature selection method covered so far. This is a brute-force evaluation of each feature subset. This means that it tries every possible combination of the variables and returns the best performing subset.

1.2.3 Embedded Methods:

These methods encompass the benefits of both the wrapper and filter methods, by including interactions of features but also maintaining reasonable computational cost. Embedded methods are iterative in the sense that takes care of each iteration of the model training process and carefully extracts those features which contribute the most to the training for a particular iteration

LASSO Regularization (L1)

Regularization consists of adding a penalty to the different parameters of the machine learning model to reduce the freedom of the model, i.e. to avoid over-fitting. In linear model regularization, the penalty is applied over the coefficients that multiply each of the predictors. From the different types of regularization, Lasso or L1 has the property that is able to shrink some of the coefficients to zero. Therefore, that feature can be removed from the model.

1.3 Data set Description

This dataset classifies people described by a set of attributes as good or bad credit risks. Comes in two formats (one all numeric). Also comes with a cost matrix.

Data Set Characteristics : Multivariate
Attribute Characteristics : Categorical, Integer
Associated Tasks : Classification
Number of Instances : 1000
Number of Attributes : 20
Missing Values : N/A

2 Machine Learning Life Cycle

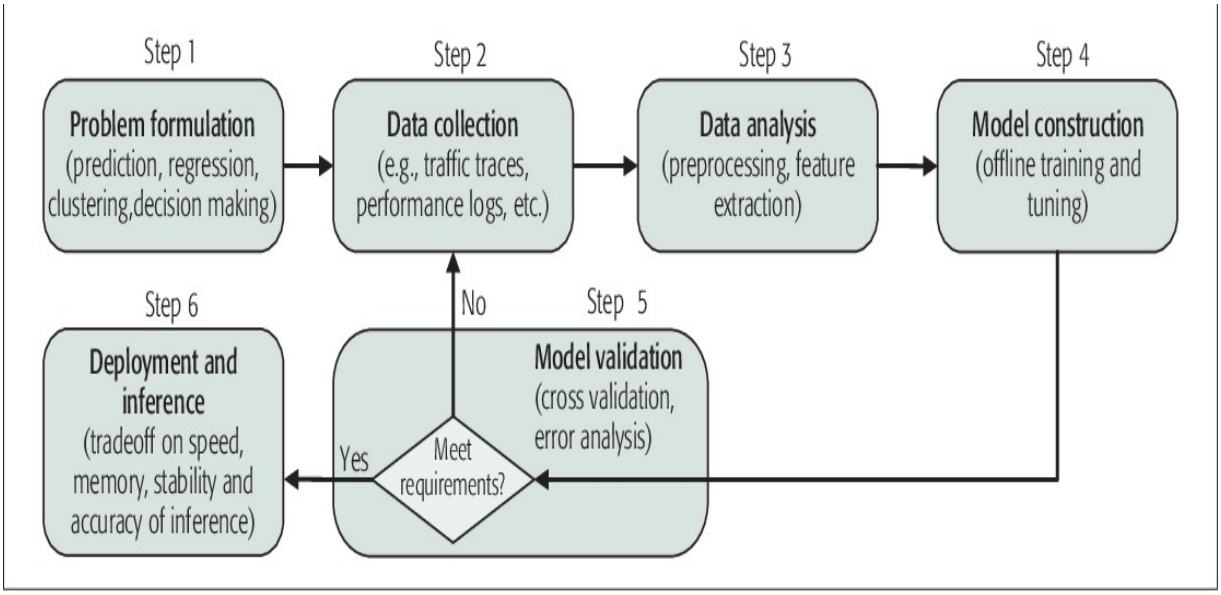


Figure 2: Simple Machine Learning Life Cycle

1. Gathering data
2. Data pre-processing
3. Data set splitting & Feature Selection
4. Modelling
5. Evaluation

2.1 Gathering Data

The process of gathering data depends on the type of project we desire to make, if we want to make an ML project that uses real-time data, then we can build an IoT system that using different sensors data. The data set can be collected from various sources such as a file, database, sensor and many other such sources but the collected data cannot be used directly for performing the analysis process as there might be a lot of missing

data, extremely large values, unorganized text data or noisy data. Therefore, to solve this problem Data Preparation is done.

2.2 Data Pre Processing

Data pre-processing is a process of cleaning the raw data i.e. the data is collected in the real world and is converted to a clean data set. In other words, whenever the data is gathered from different sources it is collected in a raw format and this data isn't feasible for the analysis.As we know that data pre-processing is a process of cleaning the raw data into clean data, so that can be used to train the model. So, we definitely need data pre-processing to achieve good results from the applied model in machine learning and deep learning projects.

2.3 Dataset splitting & Feature Selection

A dataset used for machine learning should be partitioned into three subsets — training, test, and validation sets.

Training set. A data scientist uses a training set to train a model and define its optimal parameters — parameters it has to learn from data.

Test set. A test set is needed for an evaluation of the trained model and its capability for generalization. The latter means a model's ability to identify patterns in new unseen data after having been trained over a training data. It's crucial to use different subsets for training and testing to avoid model overfitting, which is the incapacity for generalization we mentioned above.

Feature selection is the process of reducing the number of input variables when developing a predictive model. As such, it can be challenging for a machine learning practitioner to select an appropriate statistical measure for a dataset when performing filter-based feature selection.

2.4 Modelling

During this stage, a data scientist trains numerous models to define which one of them provides the most accurate predictions.

Model training

After a data scientist has preprocessed the collected data and split it into three subsets, he or she can proceed with a model training. This process entails “feeding” the algorithm with training data. An algorithm will process data and output a model that is able to find a target value (attribute) in new data — an answer you want to get with predictive analysis. The purpose of model training is to develop a model.

Two model training styles are most common — supervised and unsupervised learning. The choice of each style depends on whether you must forecast specific attributes or group data objects by similarities.

2.5 Evaluation

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future.

To improve the model we might tune the hyper-parameters of the model and try to improve the accuracy and also looking at the confusion matrix to try to increase the number of true positives and true negatives.

3. Flow of Project

Step 1: Data Collection

•Data is collected from UCI Machine Learning Repository

Step 2: Data Visualization

•Data Visualization for Integer Variables is carried out through Boxplot and Histograms

Step 3: Data Pre-processing

•Data Formatting -> Dealing with categorical variables

•Data Cleaning -> Outlier Detection -> Missing Values Treatment

Step 4: Feature Selection

• Reducing the number of input variables

Step 5: Model Building

Decision Tree

Logistic Regression

Step 6: Evaluation

• Generalize The Accuracy

4. Data Pre-Processing

The German credit scoring data is a dataset provided by Prof. Hogmann in the file german data. The data set has information about 1000 individuals, on the basis of which they have been classified as risky or not. The variable response in the dataset

corresponds to the risk label, 1 has been classified as bad and 0 has been classified as good.

We get the data from the link. We need to provide names for the columns and change the response labels to 1 and 0:

4.1 Information about DataFrame

The dataframe contains 8 columns are type int64 which are of type numerical other 13 columns are type object which are of type categorical value

```
Data columns (total 21 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   default                    1000 non-null   int64
 1   account_check_status       1000 non-null   object
 2   duration_in_month          1000 non-null   int64
 3   credit_history             1000 non-null   object
 4   purpose                    1000 non-null   object
 5   credit_amount              1000 non-null   int64
 6   savings                    1000 non-null   object
 7   present_emp_since          1000 non-null   object
 8   installment_as_income_perc 1000 non-null   int64
 9   personal_status_sex        1000 non-null   object
 10  other_debtors              1000 non-null   object
 11  present_res_since          1000 non-null   int64
 12  property                   1000 non-null   object
 13  age                        1000 non-null   int64
 14  other_installment_plans    1000 non-null   object
 15  housing                    1000 non-null   object
 16  credits_this_bank          1000 non-null   int64
 17  job                        1000 non-null   object
 18  people_under_maintenance   1000 non-null   int64
 19  telephone                  1000 non-null   object
 20  foreign_worker             1000 non-null   object
dtypes: int64(8), object(13)
```

Figure 3: Structure of Dataframe

4.1.1 Count mising value or Unique value of Categorical Data

Missing values in data is a common phenomenon in real world problems. Knowing how to handle missing values effectively is a required step to reduce bias and to produce powerful models.

	column_name	unique_values	missing_value
0	account_check_status	4	0
1	credit_history	5	0
2	purpose	10	0
3	savings	5	0
4	present_emp_since	5	0
5	personal_status_sex	4	0
6	other_debtors	3	0
7	property	4	0
8	other_installment_plans	3	0
9	housing	3	0
10	job	4	0
11	telephone	2	0
12	foreign_worker	2	0

Figure 4: Unique or missing values

Well the first idea is to remove the lines in the observations where there is some missing data. But that can be quite dangerous because imagine this data set contains crucial information. It would be quite dangerous to remove an observation. So we need to figure out a better idea to handle this problem. And another idea that's actually the most common idea to handle missing data is to take the mean of the columns**.**

From above figure we can say that there is no missing value so there is no need of taking mean values.

4.2 Dealing with Categorical Data

In this dataset there are many categorical data which need to be removed.

Machine learning uses only numeric values (float or int data type). However, data sets often contain the object data type than needs to be transformed into numeric. In most cases, categorical values are discrete and can be encoded as dummy variables, assigning a number for each category.

```
def dummy_df(df,todummy_list):
    for x in todummy_list:
        dummies=pd.get_dummies(df[x],prefix=x,dummy_na=False)
        df=df.drop(x,1)
        df=pd.concat([df,dummies],axis=1)
    return df
todummy_list=['account_check_status','credit_history','purpose','savings','present_emp_since','personal_status','other_debtors',
expanded_data=dummy_df(data,todummy_list)
```

Figure 5: Dealing with categorical data

Dealing with numeric data is often easier than categorical data given that we do not have to deal with additional complexities of the semantics pertaining to each category value in any data attribute which is of a categorical type.

While a lot of advancements have been made in various machine learning frameworks to accept complex categorical data types like text labels. Typically any standard workflow in feature engineering involves some form of transformation of these categorical values into numeric labels and then applying some encoding scheme on these values. After applying dummies method to 13 categorical value we have left with 54 encoded features.

4.3 Normalization/Scaling of Numerical variables:

Scaling is important if you need to specify that a change in one quantity is not equal to another change in another. With the help of scaling you ensure that just because some features are big they won't be used as the main predictor. For example, if you use the age and the salary of a person in prediction, some algorithms will pay attention to the salary more because it is bigger, which does not make any sense.

```
numvars = ['credit_amount', 'duration_in_month', 'installment_as_income_perc', 'present_res_since', 'age',
           'credits_this_bank', 'people_under_maintenance']

# Standardization
numdata_std = pd.DataFrame(StandardScaler().fit_transform(X[numvars]),columns=numvars)
```

Figure 6: Normalization

Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to use a common scale, without distorting differences in the ranges of values or losing information. Normalization is also required for some algorithms to model the data correctly.

- You can change all values to a 0-1 scale, or transform the values by representing them as percentile ranks rather than absolute values.
- You can apply normalization to a single column, or to multiple columns in the same dataset.
- If you need to repeat the experiment, or apply the same normalization steps to other data, you can save the steps as a normalization transform, and apply it to other datasets that have the same schema.

4.4 Outlier Detection and Outlier Removal:

Outliers are data points that are far from other data points. In other words, they're unusual values in a dataset. Outliers are problematic for many statistical analyses because they can cause tests to either miss significant findings or distort real results. There are a variety of ways to find outliers. All these methods employ different approaches for finding values that are unusual compared to the rest of the dataset.

	default	duration_in_month	credit_amount	installment_as_income_perc	present_res_since	age	credits_this_bank	people_under_maintenance
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	0.300000	20.903000	3271.258000	2.973000	2.845000	35.546000	1.407000	1.155000
std	0.458487	12.058814	2822.736876	1.118715	1.103718	11.375469	0.577654	0.362086
min	0.000000	4.000000	250.000000	1.000000	1.000000	19.000000	1.000000	1.000000
25%	0.000000	12.000000	1365.500000	2.000000	2.000000	27.000000	1.000000	1.000000
50%	0.000000	18.000000	2319.500000	3.000000	3.000000	33.000000	1.000000	1.000000
75%	1.000000	24.000000	3972.250000	4.000000	4.000000	42.000000	2.000000	1.000000
max	1.000000	72.000000	18424.000000	4.000000	4.000000	75.000000	4.000000	2.000000

Figure 7: Description of numerical features

- From The Table we may think that credit amount has high variance we can find if credit amount has outliers or not by box ploting the data.

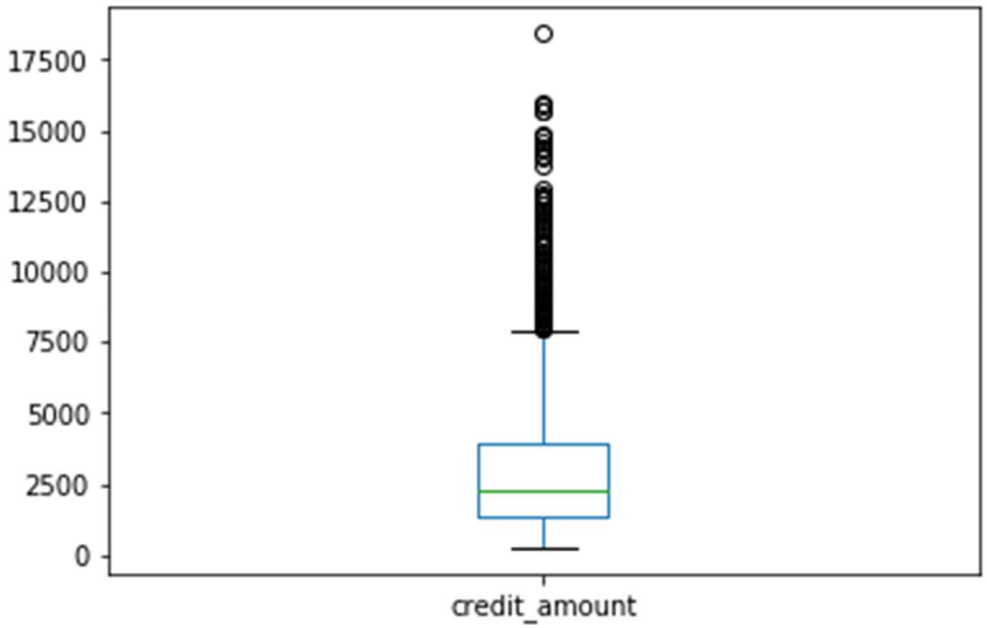


Figure 8: Box plot for credit amount

Above figure shows the box plot for credit amount and overlapping circles are nothing but comparison circles.

Box plot is a visual representation of displaying a distribution of data, usually across groups, based on a five number summary: the minimum, first quartile, the median (second quartile), third quartile, and the maximum**.**

The simplest of box plots display the full range of variation from minimum to maximum, the likely range of variation, and a typical value. A box plot will also show the outliers

5. Feature Selection

Feature selection also known as feature reduction is a technique of selecting a subset of relevant features from the original dataset. Feature selection process removes the irrelevant and redundant features and increases the predictive accuracy of the data mining algorithm. To evaluate the quality of a subset of features, the feature selection methods mainly adopt the following approaches: the filter and the wrapper. In filter method, the feature set is filtered to select the most informative subset before learning commences based on general characteristics of the data itself and independent of the mining algorithm. In wrapper method, the learning algorithm is wrapped into the selection procedure it uses the performance mining algorithm to evaluate and determine the quality of features. Filter methods evaluate feature individually based its correlation with the target attribute and then select features with highest value. They are generally faster and more practical to use on high dimensional datasets. In this study we use information gain, gain ratio and correlation based feature selection measures of filter methods.

5.1 Removing Constant or low Variance Columns

```
constant_filter=VarianceThreshold(threshold=0.01)
constant_filter.fit(x_train)
x_train_filter=constant_filter.transform(x_train)
x_test_filter=constant_filter.transform(x_test)
```

Figure 9:Varience Threshold

The idea is when a feature doesn't vary much within itself, it generally has very little predictive power. Variance Threshold doesn't consider the relationship of features with the target variable

5.2 Information Gain or mutual information

Information gain calculates the reduction in entropy from the transformation of a dataset. It can be used for feature selection by evaluating the Information gain of each variable in the context of the target variable.

We calculated mutual information of all the features the features have higher mi score they have higher influence to the information content and lower mi score have low influence to the information content.

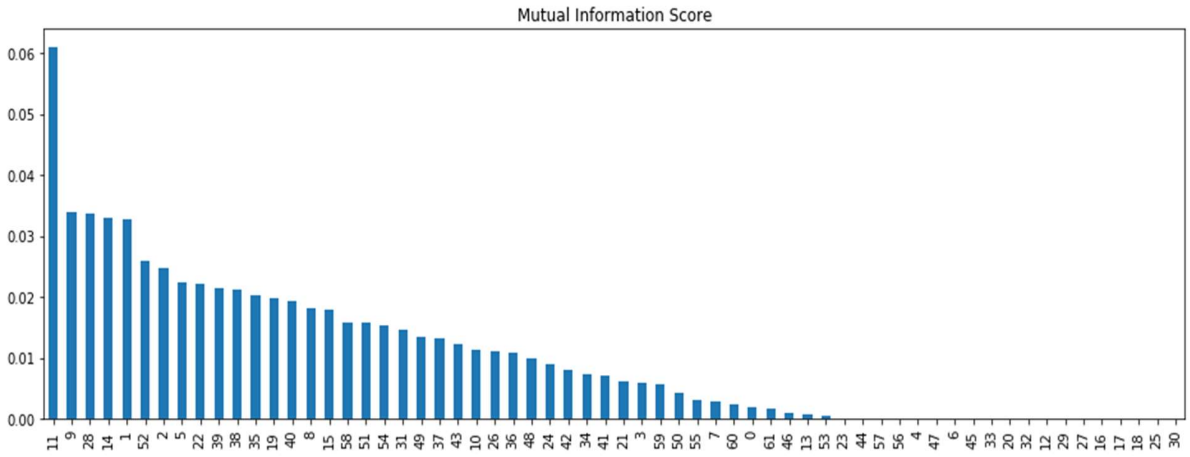


Figure 10:mutual information score

From the above Figure we can find that around 50-60 percent feature has not put that much information individually so we discard them From the all 60 features we have left with 19 features. All the selected features will trained with machine leaning model and tested with out feature selection model.

6. Model Building

6.1 Decision Tree Classifier

The classification technique is a systematic approach to build classification models from an input data set. For example, decision tree classifiers, rule-based classifiers, neural networks, support vector machines, and naive Bayes classifiers are different technique to solve a classification problem. Each technique adopts a learning algorithm to identify a model that best fits the relationship between the attribute set and class label of the input

data. Therefore, a key objective of the learning algorithm is to build predictive model that accurately predict the class labels of previously unknown records.

Decision Tree Classifier is a simple and widely used classification technique. It applies a straightforward idea to solve the classification problem. Decision Tree Classifier poses a series of carefully crafted questions about the attributes of the test record. Each time it receive an answer, a follow-up question is asked until a conclusion about the class label of the record is reached.

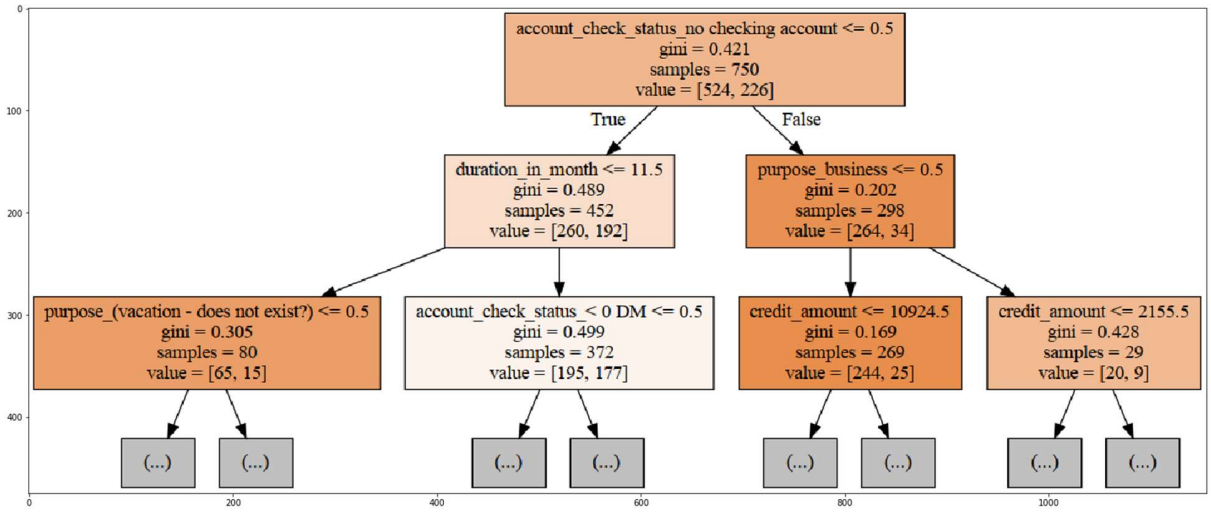


Figure 11: decision Tree

The above figure shows the decision Tree build on German credit dataset. A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

6.1.1 Evaluation of model with & without feature extraction

Below is the comparison of different depth decision tree model is tested with out feature selection technique and with feature selection technique.

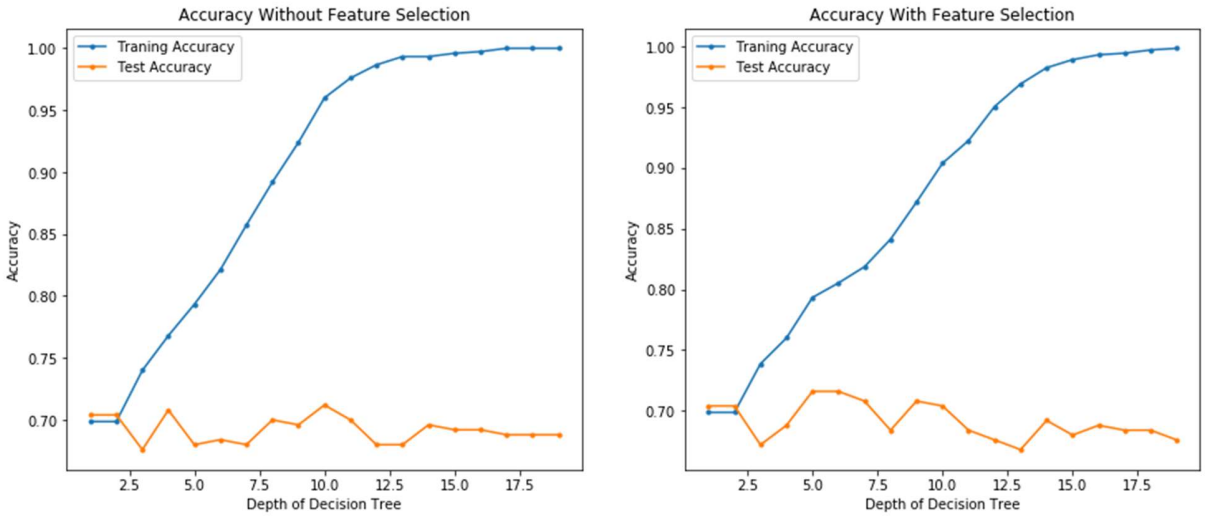


Figure 12: comparison without & with feature selection technique(information_gain)

From the above two graph we can find that removing features which are not contributing much information dosen't affect the model performance. The accuracy of both the model is almost same.

6.2 Logistic Regression

Logistic regression is a statistical method widely used in the field of machine learning and statistics for binary classification tasks, where the goal is to predict one of two possible outcomes based on input features. It's a fundamental algorithm that plays a pivotal role in various fields, including healthcare, finance, marketing, and social sciences. This overview delves into the essence of logistic regression, its principles, applications, and its significance in predictive modeling.

At its core, logistic regression is an extension of linear regression, adapted to handle classification problems. While linear regression predicts continuous numeric values, logistic regression models the probability that an input belongs to one of two classes (commonly referred to as "0" and "1" or "negative" and "positive"). It achieves this by applying a logistic function (sigmoid) to the linear combination of input features. This logistic function transforms the output into a range between 0 and 1, representing the

probability of the positive class. This probability can then be compared to a threshold (usually 0.5) to make binary predictions.

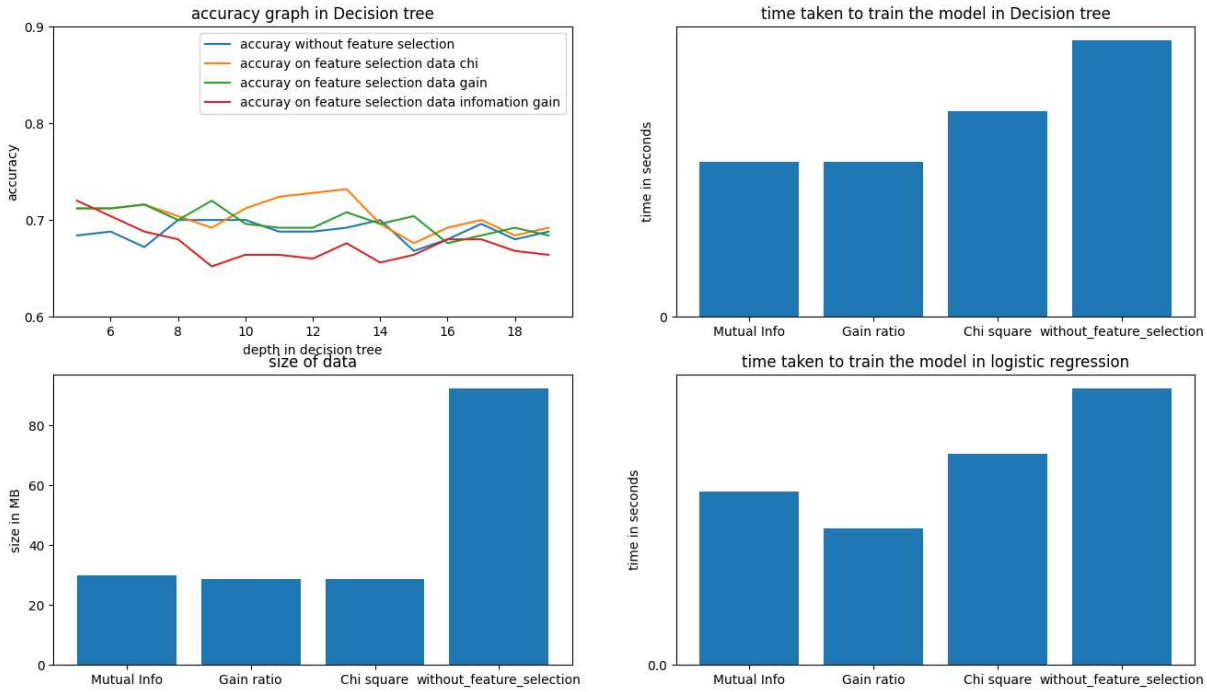
The primary strength of logistic regression lies in its simplicity, interpretability, and efficiency. It's relatively easy to implement and understand, making it an excellent choice for tasks where the decision-making process needs to be transparent. Furthermore, logistic regression handles both numerical and categorical features, making it versatile in various applications.

```
from sklearn.linear_model import LogisticRegression
from sklearn import metricsX_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.3, random_state=0)
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
```

this method LogisticRegression takes the data and fit it in the model, we will use the logistic regression to find the defaulter in credit dataset.

7. OUTPUT ANALYSIS

We have used 3 feature selection technique based on it we have reduced the input data and run both decision tree and logistic model on these dataset. Based on the output below is the graph.



1. From the above figure 1 represents the accuracy of model on different data, accuracy test data while model trained on whole data and datas after selecting subset of features with different feature selection technique.

2. we can see even after reduce the dataset with different feature selection technique the accuracy on test data not verying much

3. figure 2,4 shows time taken by the model to train on different set of data. we see the after applying feature selection technique we can reduce the data size which are taking less time to traing the model.

4. figure 3 shows the size of dataset with and without applying feature selection technique, we see by applying feature selection technique we nearly reduced the size of data by half.

8. CONCLUSION

In this Project we have applied different Feature Selection Technique in german credit dataset and compare with the without feature selection algorithm with performance and time. We have currenty applied mutual information or information gain technique to choose selective features from german credit dataset.

We concluded that some of feature does not contribute much information to the feature so we can remove them with out losing information and make the model faster.

REFERENCES

[1] Stjepan Oreski, Goran Oreski “Genetic algorithm-based heuristic for feature selection in credit risk assessment”, Expert Systems with Application, vol:41,(pp 2052–2064),Elsevier,2014.

[2] Stjepan Oreski, Goran Oreski “Hybrid system with genetic algorithm and arificial neural networks and its application to retail credit risk assessment”, Expert Systems with Application, vol:39,(pp 12605– 12617),Elsevier,2012.

[3] K.C Tan, E.J. Teoh, Q.Yu, K.C.Goh “A hybrid evolutionary algorithm for attribute selection in data mining” Expert Systems with Applications, 36, 8616-8630, 2009.

[4] Ekrem Duman, M.Hamdi Ozecelik “Detecting credit card fraud by genetic algorithm and scatter search” Expert Systems with Applications,38, 13057-13063,2011.

[5] Pappa.G.L, Freitas A.A, Kaestner C.A.A “A multi objective genetic algorithm for attribute selection” In proceedings of the fourth International conference on recent advances in soft computing (RASC-2002)(pp.116-121) Berlin: Springer,2002