A Project Report On

Feature Selection using Statistical hypothesis test Granger Causality and Analysis of Variance(ANOVA)

Submitted in partial fulfilment of the requirements for the award of the degree of **Bachelor of Engineering in Information Technology** of **The University of Burdwan**

Submitted by

DIPESH KUMAR (2016-3028, Regn. No. A4515 of 2016-17)
PRAKASH ANAND (2016-3016, Regn. No. A4516 of 2016-17)
ARJUN KUMAR (2016-3042, Regn. No. A5072 of 2016-17)
DEVENDRA KUMAR (2016-3064, Regn. No. A4533 of 2016-17)

Under the guidance of
MRS. KASTURI GHOSH
In- Charge
Dept. of Information Technology
University Institute of Technology, BU

Department of Information Technology

University Institute of Technology
The University of Burdwan
Golapbag (North), Burdwan-713104, West Bengal

Dept. of IT (8th Sem) July 2020 Project II [IT 891(OLD)

UNIVERSITY INSTITUTE OF TECHNOLOGY
THE UNIVERSITY OF BURDWAN

Certificate

This is to certify that **Mr. Prakash Anand (2016-3016),** have partially completed the project entitled "**Feature Selection using Statistical hypothesis test Granger Causality and Analysis of Variance (ANOVA)**" of the requirement for the degree of Bachelor of Engineering in Information Technology from University Institute of Technology, The University of Burdwan.

Mrs. Kasturi Ghosh
In - charge
Dept. of Information Technology
University Institute of Technology,
The University of Burdwan

ACKNOWLEDGEMENT

Many thanks to my project mentor **Mrs. Kasturi Ghosh** for helping us in completing this whole project. She regularly took an update of our work as well as gave us valuable suggestions about our project. From this project, we have learnt so much interesting facts about python and application of python in Data Mining project. With the recommendation of my project mentor, we have used GUI based Software like IBM SPSS to perform Analysis of Variance(ANOVA) F-test.

I am extremely thankful to all the professors of University Institute of Technology, and friend who gave me knowledge and courage to complete this project.

Mr. DIPESH KUMAR
Roll No.: 2016-3015

Mr. PRAKASH ANAND
Roll No.: 2016-3016

Mr. ARJUN KUMAR
Roll No.: 2016-3042

Mr. DEVENDRA KUMAR
Roll No.: 2016-3064

Table of Contents

List of Figures

ABSTRACT

The study of time series forecasting has progressed significantly in recent decades. The diabetes dataset is a binary classification problem where it needs to be analysed whether a patient is suffering from the disease or not on the basis of many available features in the dataset. Different methods and procedures of cleaning the data, feature extraction, feature engineering and algorithms to predict the onset of diabetes are used based for diagnostic measure on Pima Indians Diabetes Dataset.

we propose a feature selection algorithm specific to forecasting multivariate time series, based on (i) the notion of the Granger causality, and on (ii) Analysis of Variance(ANOVA). Lastly, we carry out experiments on several real data set(pima) and compare our proposed method to some of the most widely used dimension reduction and feature selection methods.

INTRODUCTION

Datasets is one of the crucial part of our project provided by our group mentor. Various dataset pre-processing techniques are used for data cleaning and data wrangling purposes. By the help of classification model, we have calculated the accuracy of our model on the original dataset. Feature selection techniques such as Granger Causality and ANOVA(F-test) are being used. Based on the results of feature selection, we have drop the trivial attributes present in the dataset. After that, we have calculated the accuracy on the reduced dataset.

Adequate selection of features may improve accuracy and efficiency of classifier methods. Feature ranking and selection for classification aims at reducing the dimensionality and noise in data sets. Feature selection improves algorithms performance and classification accuracy since the chance of overfitting increases with the number of features.

DATA MINING

1.1 How to do Data Mining

The accepted data mining process involves six steps:

1. **Business understanding**

   The first step is establishing the goals of the project are and how data mining can help you reach that goal. A plan should be developed at this stage to include timelines, actions, and role assignments.

2. **Data understanding**

   Data is collected from all applicable data sources in this step. Data visualization tools are often used in this stage to explore the properties of the data to ensure it will help achieve the business goals.

3. **Data preparation**

   Data is then cleansed, and missing data is included to ensure it is ready to be mined. Data processing can take enormous amounts of time depending on the amount of data analysed and the number of data sources. Therefore, distributed systems are used in modern database management systems (DBMS) to improve the speed of the data mining process rather than burden a single system. They're also more secure than having all an organization's data in a single data warehouse. It's important to include failsafe measures in the data manipulation stage so data is not permanently lost.

4. **Data Modelling**

   Mathematical models are then used to find patterns in the data using sophisticated data tools.

5. **Evaluation**

   The findings are evaluated and compared to business objectives to determine if they should be deployed across the organization.

6. **Deployment**

   In the final stage, the data mining findings are shared across everyday business operations. An enterprise business intelligence platform can be used to provide a single source of the truth for self-service data discovery.

1.2 Benefits of Data Mining

- **Automated Decision-Making**

  Data Mining allows organizations to continually analyse data and automate both routine and critical decisions without the delay of human judgment. Banks can instantly detect fraudulent transactions, request verification, and even secure personal information to protect customers against identity theft. Deployed within a firm's operational algorithms, these models can collect, analyse, and act on data independently to streamline decision making and enhance the daily processes of an organization.

- **Accurate Prediction and Forecasting**

  Planning is a critical process within every organization. Data mining facilitates planning and provides managers with reliable forecasts based on past trends and current conditions. Macy's implements demand forecasting models to predict the demand for each clothing category at each store and route the appropriate inventory to efficiently meet the market's needs.

- **Cost Reduction**

  Data mining allows for more efficient use and allocation of resources. Organizations can plan and make automated decisions with accurate forecasts that will result in maximum cost reduction. Delta imbedded RFID chips in passengers checked baggage and deployed data mining models to identify holes in their process and reduce the number of bags mishandled. This process improvement increases passenger satisfaction and decreases the cost of searching for and re-routing lost baggage.

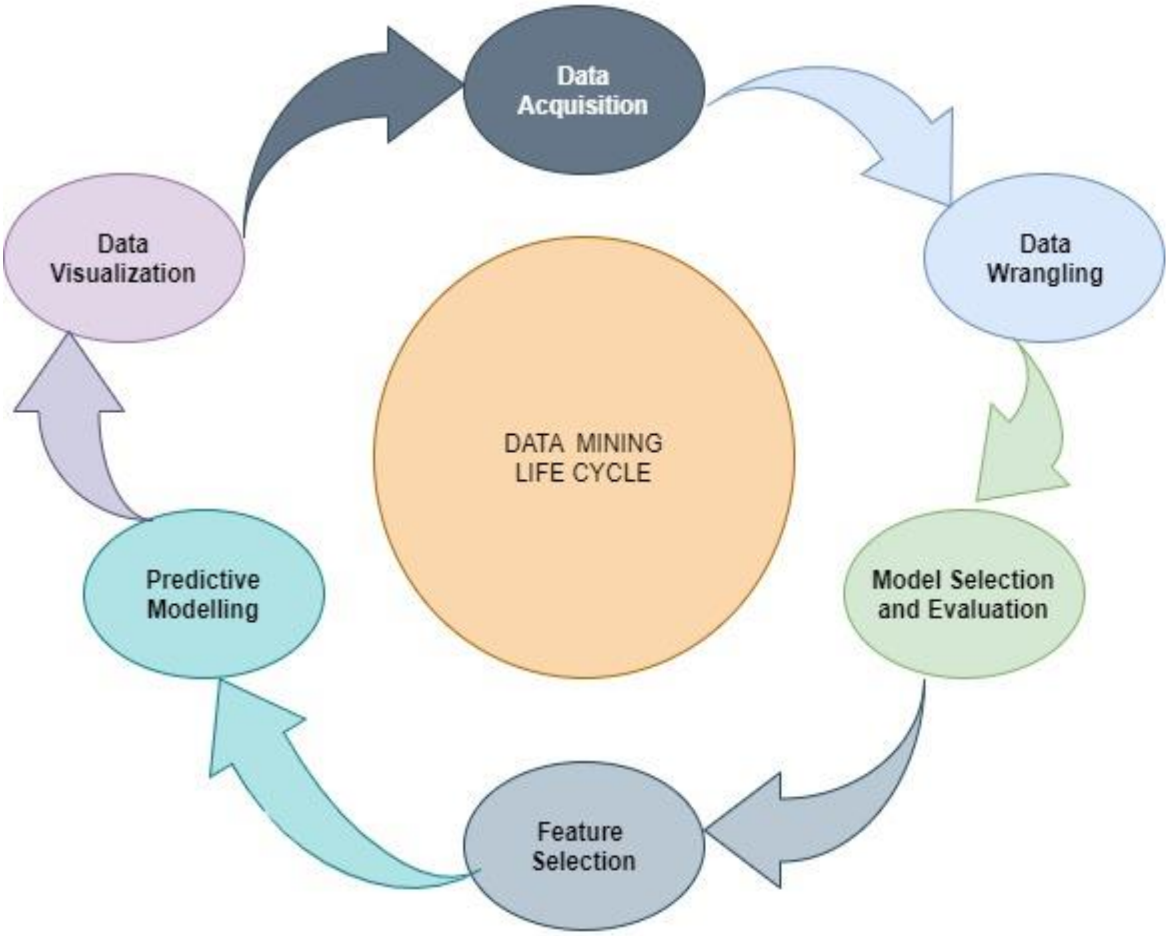
1.3 Challenges of Data Mining

While a powerful process, data mining is hindered by the increasing quantity and complexity of big data. Where exabytes of data are collected by firms every day, decision-makers need ways to extract, analyze, and gain insight from their abundant repository of data.

1. Over-Fitting Models

Over-fitting occurs when a model explains the natural errors within the sample instead of the underlying trends of the population. Over-fitted models are often overly complex and utilize an excess of independent variables to generate a prediction. Therefore, the risk of over-fitting is heighted by the increase in volume and variety of data. Too few variables make the model irrelevant, where as too many variables restrict the model to the known sample data. The challenge is to moderate the number of variables used in data mining models and balance its predictive power with accuracy.

Data Mining Life Cycle



Data mining Life Cycle

1.5 Stages of Data Mining Life Cycle

1. Data Acquisition:

The very first step of a data mining project is straightforward. We obtain the data that we need from available data sources. We are using Python, they have specific packages that can read data from these data sources directly into your data science programs.

2. Data Wrangling:

After obtaining data, the next immediate thing to do is scrubbing data. This process is for us to “clean” and to filter the data. In this process, we need to convert the data from one format to another and consolidate everything into one standardized format across all data. For example, if your data is stored in multiple CSV files, then you will consolidate these CSV data into a single repository, so that you can process and analyse it.

3. Model selection and Evaluation:

Once our data is ready to be used, and right before you jump into AI and Machine Learning, we will have to examine the data. After that, we will need to explore the data. For that, we will need to inspect the data and its properties. Different data types like numerical data, categorical data, ordinal and nominal data etc. require different treatments. Then, the next step is to compute descriptive statistics to extract features and test significant variables.

4. Feature Selection:

One of the first things we need to do in feature selection is to reduce the dimensionality of your data set. Not all our features or values are essential to predicting model. What we need to do is to select the relevant ones that contribute to the prediction of results.

5. Predictive Modelling:

We are at the final and most crucial step of a data science project, interpreting models and data. The predictive power of a model lies in its ability to generalise.

6. Data Visualization:

Data Visualization is the process of extracting and visualizing the data in a very clear and understandable way without any form of reading or writing by displaying the results in the form of pie charts, bar graphs, statistical representation and through graphical forms as well.

Data Acquisition

2.1 Introduction

In many applications, one must invest effort or money to acquire the data and other information required for machine learning and data mining. careful selection of the information to acquire can substantially improve generalization performance per unit cost. The costly information scenario that has received the most research attention (see Chapter X) has come to be called ”active learning,” and focuses on choosing the instances for which target values (labels) will be acquired for training. However, machine learning applications offer a variety of different sorts of information that may need to be acquired.

2.2 Overarching principles for selective data acquisition

In general selective data acquisition a learning algorithm can request the value of particular missing data, which is then provided by an oracle at some cost. There may be more than one oracle, and oracles are not assumed to be perfect. The goal of selective data acquisition is to choose to acquire data that is most likely to improve the system’s use-time performance on a specified modelling objective in a cost-effective manner. We will use to refer to the query for a selected piece of missing data. For instance, in traditional active learning this would correspond to querying for the missing label of a selected instance; while in the context of active feature-value acquisition, q is the request for a missing feature value. We will focus primarily (but not exclusively) on pool-based selective data acquisition, where we select a query from a pool of available candidate queries, e.g., the set of all missing feature-values that can be acquired on request.

2.3 Learning from feature labels

A simple way to utilize feature supervision is to use the labels on features to label examples, and then use an existing supervised learning algorithm to build a model. Consider the following straightforward approach. Given a representative set of words for each class, create a representative document for each class containing all the representative words. Then compute the cosine similarity between unlabelled documents and the representative documents. Assign each unlabelled document to the class with the highest similarity, and then train a classifier using these pseudo-labelled examples. This approach is very convenient as it does not require devising a new model, since it can effectively leverage existing supervised learning techniques such as Naive Bayes [46].

Data Wrangling

3.1 Introduction

Data wrangling is a broad term used, often informally, to describe the process of transforming raw data to a clean and organized format ready for use. For us, data wrangling is only one step in pre-processing our data, but it is an important step.

The most common data structure used to “wrangle” data is the data frame, which can be both intuitive and incredibly versatile. Data frames are tabular, meaning that they are based on rows and columns like you would see in a spreadsheet. Here is a data frame created from data about passengers on the *Titanic*:

```
# Load library
import pandas as pd

# Create URL
url =
'https://raw.githubusercontent.com/chrisalbon/sim_data/master/titanic.csv'

# Load data as a data frame
dataframe = pd.read_csv(url)

# Show first 5 rows
dataframe.head(5)
```

Name	PClass	Age	Sex	Survived	SexCode
0	Allen, Miss Elisabeth Walton	1st 29.00	female	1	1
1	Allison, Miss Helen Loraine	1st 2.00	female	0	1
2	Allison, Mr Hudson Joshua Creighton	1st 30.00	male	0	0
3	Allison, Mrs Hudson JC (Bessie Waldo Daniels)	1st 25.00	female	0	1
4	Allison, Master Hudson Trevor	1st 0.92	male	1	0

There are three important things to notice in this data frame.

First, in a data frame each row corresponds to one observation (e.g., a passenger) and each column corresponds to one feature (gender, age, etc.). For example, by looking at the first observation we can see that Miss Elisabeth Walton Allen stayed in first class, was 29 years old, ...

3.2 **Importance of Data Wrangling**

When large amounts of data are processed for interpretation chances are all of it is not relevant or outdated. Although data wrangling is a tedious process, conducting it will ensure that the data secured is not outdated or irrelevant. Therefore, data wrangling provides credibility to data analytics courses.

3.3 Data Wrangling Tools

Basic Data Munging Tools

Excel Power Query / Spreadsheets — the most basic structuring **tool** for manual **wrangling**.

Open Refine — more sophisticated solutions, requires programming skills. Google Data Prep - for exploration, cleaning, and preparation.

3.4 Conclusion

Data wrangling is extremely relevant today due to the large amounts of data that gets proceeded every day. We will not be able to do thorough analytics if we do not have a strong infrastructure of data storage and hence companies are investing heavily in data wrangling tools.

Model selection and Evaluation

4.1. Basic machine-learning modelling

The objective of machine learning is to discover patterns and relationships in data and to put those discoveries to use. This process of discovery is achieved through the use of modelling techniques that have been developed over the past 30 years in statistics, computer science, and applied mathematics. These various approaches can range from simple to tremendously complex, but all share a common goal: to estimate the functional relationship between the input features and the target variable.

These approaches also share a common workflow, as illustrated in figure 4.1: use of historical data to build and optimize a model that is, in turn, used to make predictions based on new data. This section prepares you for the practical sections later in the chapter. You'll look at the general goal of machine learning modelling in the next section, and move on to seeing how the end product can be used and a few important aspects for differentiating between ML algorithms.

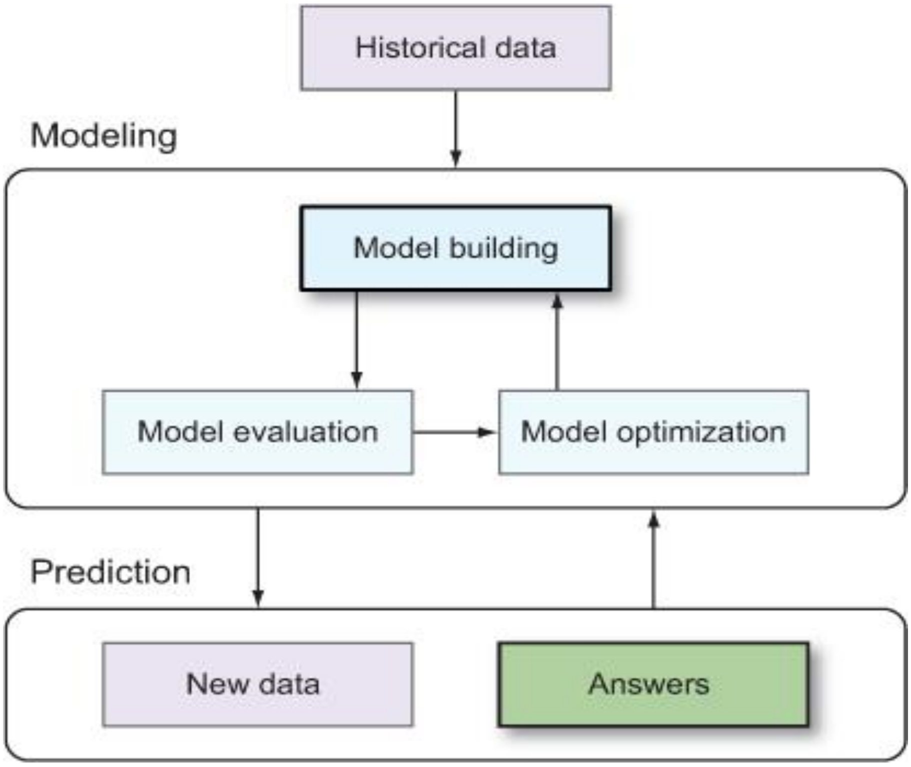


Figure 4.1. The basic ML workflow

4.1.1. Finding the relationship between input and target

Let’s frame the discussion of ML modelling around an example. The dataset contains metrics about automobiles, such as manufacturer region, model year, vehicle weight, horsepower, and number of cylinders. The purpose of the dataset is to understand the relationship between the input features and a vehicle’s miles per gallon (MPG) rating.

Input features are typically referred to using the symbol X, with subscripts differentiating inputs when multiple input features exist.

For instance, we'll say that X_1 refers to manufacturer region, X_2 to model year, X_3 to vehicle weight, and so forth. The collection of all the input features is referred to as the bold $\mathbf{X}$. Likewise, the target variable is typically referred to as Y.

4.2. Types of modelling methods

Now the time has come to dust off your statistics knowledge and dive into some of the mathematical details of ML modelling. Don't worry—we'll keep the discussion relatively broad and understandable for those without much of a statistics background!

4.2.1. Supervised versus unsupervised learning

Machine-learning problems fall into two camps: supervised and unsupervised. *Supervised problems* are ones in which you have access to the target variable for a set of training data, and *unsupervised problems* are ones in which there's no identified target variable.

There are many more like:

- parametric and non-parametric
- Reinforcement Learning models

4.3. Approaches for model selection

1. The 1st approach is based on the **'Structural Risk Minimization (SRM)':** it is useful when the learning algorithm depends on a parameter that controls the bias-complexity trade off (such as the degree of the fitted polynomial in the preceding example).
2. The 2nd approach relies on the concept of **'Validation':** the basic idea is to partition the training set into 2 sets. One used for training each of the candidate models, and the second is used for deciding which of them yields the best results.

4.4. What if model fails?

There are many elements that can be "fixed." The main approaches are listed in the following:

- Get a larger sample
- Change the hypothesis class by:

— Enlarging it

— Reducing it

— Completely changing it

— Changing the parameters you consider

- Change the feature representation of the data
- Change the optimization algorithm used to apply your learning rule

To understand the cause of the bad performance; we have to understand that the true error is decomposed of **approximation** and **estimation** error.

The approximation error of the class does not depend on the sample size or on the algorithm being used. It only depends on the distribution *D* and on the hypothesis class *H*. Therefore, if the approximation error is large, it will not help us to enlarge the training set size, and it also does not make sense to reduce the hypothesis class. What can be beneficial in this case is to enlarge the hypothesis class or completely change it (if we have some alternative prior knowledge in the form of a different hypothesis class). We can also consider applying the same hypothesis class but on a different feature representation of the data.

The estimation error of the class does depend on the sample size. Therefore, if we have a large estimation error we can make an effort to obtain more training examples. We can also consider reducing the hypothesis class. However, it doesn’t make sense to enlarge the hypothesis class in that case.

4.5 Conclusion

1. If learning involves parameter tuning, plot the model-selection curve to make sure that you tuned the parameters appropriately.

2. If the training error is excessively large consider enlarging the hypothesis class, completely change it, or change the feature representation of the data.

3. If the training error is small, plot learning curves and try to deduce from them whether the problem is estimation error or approximation error.

4. If the approximation error seems to be small enough, try to obtain more data. If this is not possible, consider reducing the complexity of the hypothesis class.

5. If the approximation error seems to be large as well, try to change the hypothesis class or the feature representation of the data completely.

Feature Selection

5.1 Introduction

A fundamental problem of machine learning is to approximate the functional relationship f () between an input X = {x_1, x_2, x_3, …., x_M} and an output Y, based on a memory of data points, {X_i, Y_i}, i = 1, 2, …, n, usually the X_i's are vectors of real Y_i's are real numbers. Some-times the output Yi's not determined by the complete set of the input features {x_1, x_2, x_3, …., x_M}, instead, it is decided only by a subset of them {$x_{(1)}$, $x_{(2)}$, $x_{(3)}$, …., $x_{(m)}$}, where m < M. With sufficient data and time, it is fine to use all the input features, including those irrelevant features, to approximate the underlying function between the input and the output. But in practice, there are two problems which may be evoked by the irrelevant features involved in the learning process.

1.The irrelevant input features will induce greater computational cost. For example, using cached kd-trees as we discussed in last chapter, locally weighted linear regression's computational expense is O (m3+m2log N) for doing a single prediction, where Ni's the number of data points in memory and m is the number of features used. Apparently, with more features, the computational cost for predictions will increase poly-nomially; especially when there are a large number of such predictions, the computational cost will increase immensely.

2.The irrelevant input features may lead to overfitting. For example, in the domain of medical diagnosis, our purpose is to infer the relationship between the symptoms and their corresponding diagnosis. If by mistake we include the patient ID number as one input feature, an over-tuned machine learning process may come to the conclusion that the illness is determined by the ID number.

5.2 Motivation

Motivation for feature selection is that, since our goal is to approximate the underlying function between the input and the output, it is reasonable and important to ignore those input features with little effect on the output, so as to keep the size of the approximator model small.

5.3 Benefits of Feature Selection

Feature Selection is one of the core concepts in machine learning which hugely impacts the performance of your model. It is one of the proficient method used in data mining process to improve the accuracy of the given model. Irrelevant attributes present in the dataset leads to decrease the accuracy of the model.

Benefits of performing feature selection in our data

- Reduces Overfitting: Less redundant data means less opportunity to make decisions based on noise.

- Improves Accuracy: Less misleading data means modelling accuracy improves.
- Reduces Training Time: fewer data points reduce algorithm complexity and algorithms train faster.

5.4 Feature selection algorithms

In this section, we introduce the ANOVA (F- test) and granger causality feature selection algorithm:

we explore these algorithms in order to improve the computational efficiency without sacrificing too much accuracy.

5.4.1. ANOVA (F-test)

The biggest challenge in machine learning is selecting the best features to train the model. We need only the features which are highly dependent on the response variable.

ANOVA (**An**alysis **o**f **Va**riance) helps us to complete our job of selecting the best features.

We are discussing these feature of ANOVA

a. Impact of Variance

b. F-Distribution

c. One Way ANOVA

Impact of Variance

Variance is the measurement of the spread between numbers in a variable. It measures how far a number is from the mean and every number in a variable.

The variance of a feature determines how much it is impacting the response variable. If the variance is low, it implies there is no impact of this feature on response and vice-versa.

F-Distribution

A probability distribution generally used for the analysis of variance. It assumes Hypothesis as

H0: Two variances are equal

H1: Two variances are not equal.

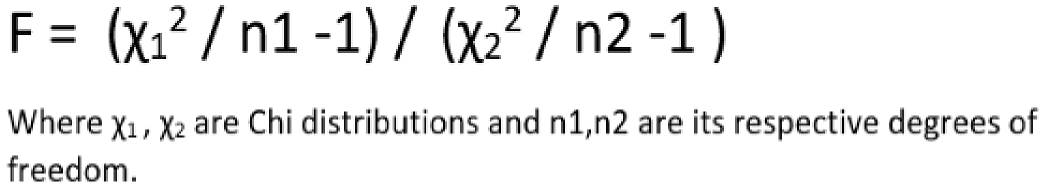
Degrees of Freedom

Degrees of freedom refers to the maximum number of logically independent values, which have the freedom to vary. In simple words, it can be defined as the total number of observations minus the number of independent constraints imposed on the observations.

Df = N -1 where N is the Sample Size

F- Value

It is the ratio of two Chi-distributions divided by its degrees of Freedom.



$$F = (\chi_1^2 / n1 -1) / (\chi_2^2 / n2 -1)$$

Where χ_1 , χ_2 are Chi distributions and n1,n2 are its respective degrees of freedom.

One Way ANOVA

One Way ANOVA tests the relationship between categorical predictor vs continuous response.

Steps to perform One Way ANOVA

1. Define Hypothesis
2. Calculate the Sum of Squares
3. Determine degrees of freedom
4. F-value
5. Accept or Reject the Null Hypothesis

5.4.2. Granger Causality

1. The Granger causality test is a statistical hypothesis test for determining whether one-time series is useful in forecasting another. As its name implies, Granger causality is not necessarily true causality. In fact, the Granger-causality tests fulfill only the Human definition of causality that identifies the cause-effect relations with constant conjunctions. The features selected by our

method also give potential causal relationships between the variables and the target time series.

2. Granger’s causality Tests the null hypothesis that the coefficients of past values in the regression equation is zero. In simpler terms, the past values of time series (x) do not cause the other series (y). So, if the p-value obtained from the test is lesser than the significance level of 0.05, then, you can safely reject the null hypothesis.

> A time series *X* is said to Granger-cause *Y* if it can be shown, usually through a series of t-tests and F-tests on lagged values of *X* (and with lagged values of *Y* also included), that those *X* values provide statistically significant information about future values of *Y*.

5.5 Conclusion

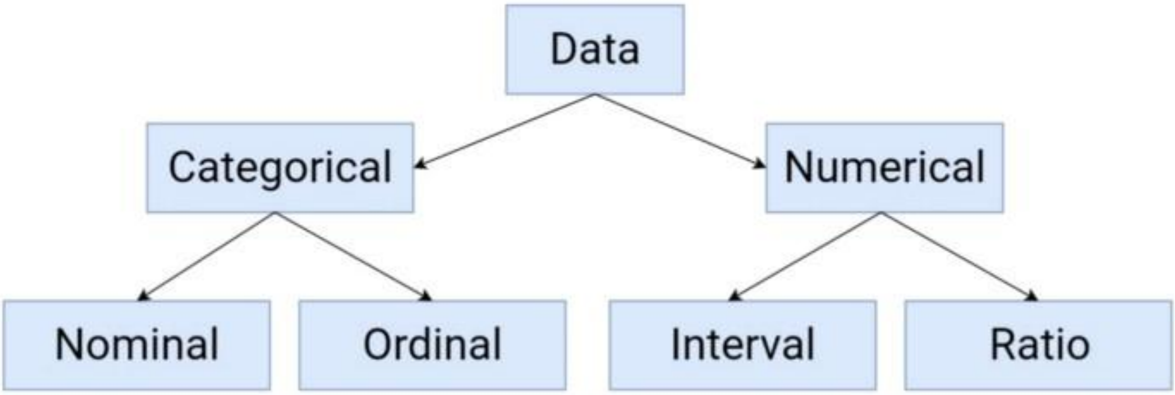
Feature Selection is one of the core concepts in machine learning which hugely impacts the performance of your model. It is one of the proficient method used in data mining process to improve the accuracy of the given model.

Data Pre-processing

6.1 Introduction

Data pre-processing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviours or trends, and is likely to contain many errors.

Data pre-processing is used database-driven applications such as customer relationship management and rule-based applications (like neural networks).



6.1 Statistical Data Types

6.2 Steps during Pre-processing:

Several steps of data Pre-processing involves:

1. Data Cleaning: Data is cleansed through processes such as filling in missing values, smoothing the noisy data, or resolving the inconsistencies in the data.

2. Data Integration: Data with different representations are put together and conflicts within the data are resolved.

3. Data Transformation: Data is normalized, aggregated and generalized.

4. Data Reduction: This step aims to present a reduced representation of the data in a data warehouse.

5. Data Standardization: This step helps to map the target column of a dataset into a standardized range, which makes computation easier.

Classification Model

7.1 Introduction

Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data.

7.2 Types of classification algorithms we used in Machine Learning:

1. K- Nearest Neighbour: The k-nearest-neighbours algorithm is a classification algorithm, and it is supervised: it takes a bunch of labelled points and uses them to learn how to label other points. To label a new point, it looks at the labelled points closest to that new point (those are its nearest neighbours), and has those neighbours vote, so whichever label the most of the neighbours have is the label for the new point (the “k” is the number of neighbours it checks).

2. Logistic Regression: It is a statistical method for analysing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent

(predictor or explanatory) variables. This is better than other binary classification like nearest neighbour since it also explains quantitatively the factors that lead to classification.

3. Support Vector Machines(SVM): The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N - the number of features) that distinctly classifies the data points. Support vector machine is highly preferred by many as it produces significant accuracy with less computation power. Support Vector Machine, abbreviated as SVM can be used for both regression and classification tasks.

4. Decision Trees: Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches and a leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

7.3 K-Nearest Neighbour(KNN)

It is supervised lazy learning algorithm where all computation is deferred until classification. It is a non-parametric method used for classification. Also, It is an instance based learning algorithm where the function is approximated locally. There is no explicit training phase and the algorithm does not perform any generalization of the training data. This algorithm is mainly used when non-linear decision boundaries between class.

Input Features

It can be both quantitative and qualitative.

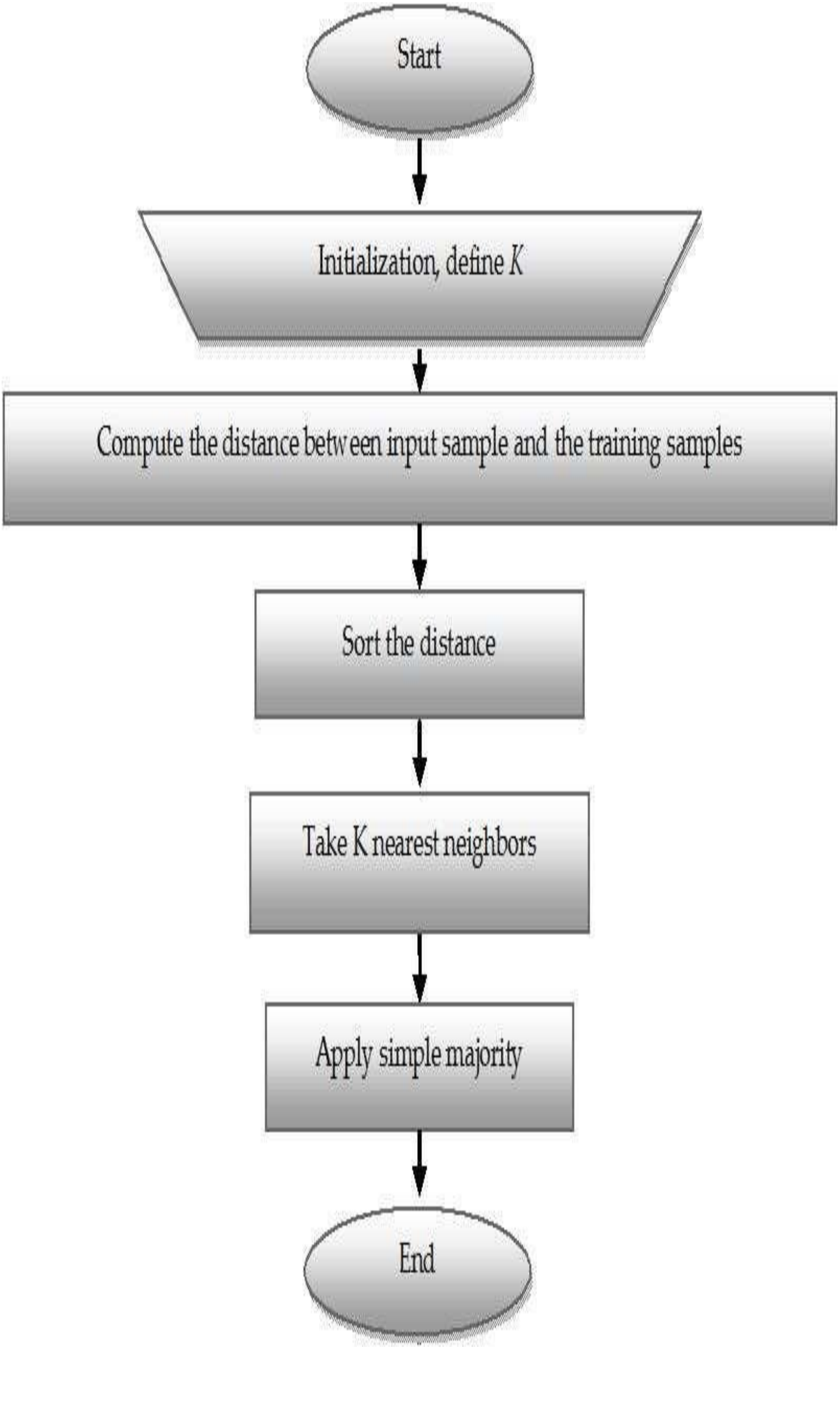
Output:

Outputs are categorical values, which typically are the classes of the data.

Assumptions taken in this algorithm

- Being non-parametric, the algorithm does not make any assumptions about the underlying data distribution.
- Select the parameter ‘k’ based on the data.
- Requires a decision metric to define proximity between any two data points. For Ex: Euclidian Distance, Manhattan Distance, Hamming Distance.

KNN Classifier Algorithm


Start
Initialization, define *K*
Compute the distance between input sample and the training samples
Sort the distance
Take K nearest neighbors
Apply simple majority
End

The KNN Classification is performed using the following four steps:

1. Compute the distance metric between the test data point and all the labelled data point.
2. Order the labelled data points in the increasing order of their distance metric.
3. Select the top ‘k’ labelled data points and look at the class labels.
4. Find the class label that the majority of these ‘k’ labelled data points have and assign it to the test data points.

Factor affecting the KNN Algorithm

- Parameter Selection.
- Presence of noise.
- Feature selection and scaling.
- Curse of dimensionality

Implementation of k-nearest neighbour algorithm

k-NN (train, test, cl, k=1)

Arguments:

- train: Matrix or data frame of training set cases.
- test: Matrix or data frame of test set cases.
- cl: factor of true classification of training set.
- k: Numbers of neighbours considered.

Complexity

- Basic k-NN algorithm stores all examples
- Suppose we have n examples each of dimension d
- **O(d)** to compute distance to one examples
- **O (n* d)** to computed distances to all examples
- More **O (n * k)** time to find k closest examples
- Total time: **O (n * k + n * d)**
- Very expensive for a large number of samples
- But we need a large number of samples for k-NN to work well.

Advantages of KNN classifier:

- Can be applied to the data from any distribution for example, data does not have to be separable with a linear boundary
- Very simple and intuitive
- Good classification if the number of samples is large enough

Applications of KNN Classifier

- Used in classification
- Used to get missing values
- Used in pattern recognition
- Used in gene expression
- Used in protein- protein prediction
- Used to get 3D structure of protein
- Used to measure document similarity

7.4 Logistic Regression

It is primary used as a classification algorithm. It belongs to supervised learning algorithm where data is labelled. Decision boundary used to classify the datasets can be linear or non-linear. The main goal of this algorithm is to predict the class from which the data point is likely to have originated.

Input Features

- It can be qualitative or quantitative.
- It is generally used for binary classification problems.

Output

- The probability of a “Yes” or “No” gives a better understanding of the sample’s membership to a particular category.
- Estimating the binary outputs from the probabilities is straight through simple thresholding.

Implementation of Logistic Regression

- Import the Logistic Regression model & other packages required.
- Split data into Training and Test data.
- Instantiated the logistic regression function.

- Calculate the modal score or accuracy using model score function.

7.4.1 Advantages of Logistic Regression

1. Logistic Regression performs well when the **dataset is linearly separable**.

2. Logistic regression is less prone to over-fitting but it can overfit in high dimensional datasets. You should consider Regularization (L1 and L2) techniques to avoid over-fitting in these scenarios.

3. Logistic Regression not only gives a measure of how relevant a predictor (coefficient size) is, but also its direction of association.

4. Logistic regression is easier to implement, interpret and very efficient to train.

7.4.2 Disadvantages of Logistic Regression

1. Main limitation of Logistic Regression is the **assumption of linearity** between the dependent variable and the independent variables. In the real world, the data is rarely linearly separable. Most of the time data would be a jumbled mess.

2. If the number of observations are lesser than the number of features, Logistic Regression should not be used, otherwise it may lead to overfit.

3. Logistic Regression can only be **used to predict discrete functions**. Therefore, the dependent variable of Logistic Regression is restricted to the discrete number set. This restriction itself is problematic, as it is prohibitive to the prediction of continuous data.

7.4.3 Applications

Logistic regression is used in various fields, including machine learning, most medical fields, and social sciences. For example, the Trauma and Injury Severity Score (TRISS), which is widely used to predict mortality in injured patients, was originally developed by Boyd *et al.* using logistic regression. Many other medical scales used to assess severity of a patient have been developed using logistic regression. Logistic regression may be used to predict the risk of developing a given disease (e.g. diabetes; coronary heart disease), based on observed characteristics of the patient (age, sex, body mass index, results of various blood tests, etc.)

7.5 Support Vector Machine

In machine learning, **support-vector machines** (**SVMs**, also **support-vector networks**) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. The Support Vector Machine (SVM) algorithm is a popular machine learning tool that offers solutions for both classification and regression problems. Developed at AT&T Bell Laboratories by Vapnik with colleagues (Boser et al., 1992, Guyon et al., 1993, Vapnik et al., 1997), it presents one of the most robust prediction methods, based on the statistical learning framework or VC theory proposed by Vapnik and Chervonekis (1974) and Vapnik (1982, 1995).

7.5.1 Motivation

Classifying data is a common task in machine learning. Suppose some given data points each belong to one of two classes, and the goal is to decide which class a *new* data point will be in. In the case of support-vector machines, a data point is viewed as a p-dimensional vector (a list of p numbers), and we want to know whether we can separate such points with a (p-1)-dimensional hyperplane. This is called a linear classifier. There are many hyperplanes that might classify the data. One reasonable choice as the best hyperplane is the one that represents the largest separation, or margin, between the two classes. So, we choose the hyperplane so that the distance from it to the nearest data point on each side is maximized. If such a hyperplane exists, it is known as the *maximum-margin hyperplane* and the linear classifier it defines is known as a *maximum-margin classifier*; or equivalently, the *perceptron of optimal stability.*

7.5.2 Advantages

1. Regularization capabilities:

SVM has L2 Regularization feature. So, it has good generalization capabilities which prevent it from over-fitting.

2. Handles non-linear data efficiently:

SVM can efficiently handle non-linear data using Kernel trick.

3. Solves both Classification and Regression problems:

SVM can be used to solve both classification and regression problems. SVM is used for classification problems while **SVR (Support Vector Regression)** is used for regression problems.

4. Stability: A small change to the data does not greatly affect the hyperplane and hence the SVM. So, the SVM model is stable.

7.5.3 Disadvantages of Support Vector Machine (SVM)

1. Choosing an appropriate Kernel function is difficult:

Choosing an appropriate Kernel function (to handle the non-linear data) is not an easy task. It could be tricky and complex. In case of using a high dimension Kernel, you might generate too many support vectors which reduce the training speed drastically.

2. Extensive memory requirement:

Algorithmic complexity and memory requirements of SVM are very high. You need a lot of memory since you have to store all the support vectors in the memory and this number grows abruptly with the training dataset size.

3. Requires Feature Scaling:

One must do feature scaling of variables before applying SVM.

4. Long training time:

SVM takes a long training time on large datasets.

5. Difficult to interpret:

SVM model is difficult to understand and interpret by human beings unlike Decision Trees.

7.6 Decision Tree

7.6.1 Introduction

A decision tree is a flowchart-like structure in which each internal node represents a test on a feature (e.g. whether a coin flip comes up heads or tails) , each leaf node represents a class label (decision taken after computing all features) and branches represent conjunctions of features that lead to those class labels. The paths from root to leaf represent classification rules. Below diagram illustrate the basic flow of decision tree for decision making with labels (Rain(Yes), No Rain(No)).

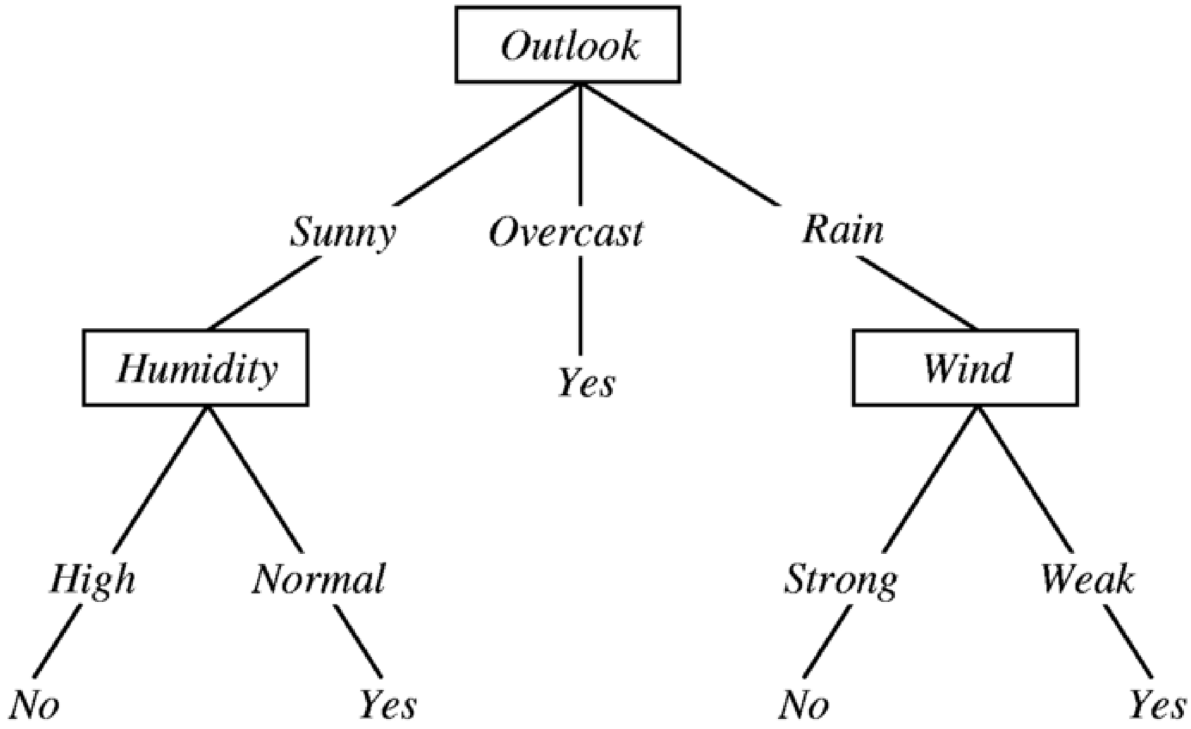


Figure 7.2 Decision Tree for Rain Forecasting

Decision tree is one of the predictive modelling approaches used in statistics, data mining and machine learning.

Decision trees are constructed via an algorithmic approach that identifies ways to split a data set based on different conditions. It is one of the most widely used and practical methods for supervised learning. Decision Trees are a non-parametric **supervised learning** method used for both **classification** and **regression** tasks.

7.6.2 Steps for Making decision tree

- Get list of rows (dataset) which are taken into consideration for making decision tree (recursively at each nodes).
- Calculate uncertanity of our dataset or Gini impurity or how much our data is mixed up etc.
- Generate list of all question which needs to be asked at that node.
- Partition rows into True rows and False rows based on each question asked.
- Calculate information gain based on gini impurity and partition of data from previous step.
- Update highest information gain based on each question asked.

- Update best question based on information gain (higher information gain).
- Divide the node on best question. Repeat again from step 1 again until we get pure node (leaf nodes).

7.6.3 **Advantage of Decision Tree**

- Easy to use and understand.
- Can handle both categorical and numerical data.
- Resistant to outliers, hence require little data pre-processing.

7.6.4 Disadvantage of Decision Tree

- Prone to overfitting.
- Require some kind of measurement as to how well they are doing.
- Need to be careful with parameter tuning.
- Can create biased learned trees if some classes dominate.

7.6.7 How to avoid overfitting the Decision tree model

Overfitting is one of the major problem for every model in machine learning. If model is overfitted it will poorly generalized to new samples. To avoid decision tree from overfitting **we remove the branches that make use of features having low importance.** This method is called as **Pruning or post-pruning.** This way we will reduce the complexity of tree, and hence improves predictive accuracy by the reduction of overfitting.

Pruning should reduce the size of a learning tree without reducing predictive accuracy as measured by a cross-validation set. There are 2 major Pruning techniques.

- *Minimum Error:* The tree is pruned back to the point where the cross-validated error is a minimum.
- *Smallest Tree:* The tree is pruned back slightly further than the minimum error. Technically the pruning creates a decision tree with cross-validation error within 1 standard error of the minimum error.

Implementation and Results

8.1 Overview

This chapter describes the implementation of the different algorithms and improving the accuracy result of different dataset. We are going to implement Granger causality and ANOVA (F- test) to and going to see how this reduce the attribute and improve the result.

I am going to show you only one of the dataset (Abalone), I will show you the result for the rest of the dataset.

8.2 Tools and Technology Required

- **Computer/ Laptop**
    - Processor – minimum i3
    - RAM – 4GB
    - System type – x86 bit or x64 bit

- **Jupyter Notebook**
- **Python**

8.3 Collect all required packages and file

8.3.1 To import required packages

```
 pip install statsmodels

import numpy as np

import pandas as pd
```

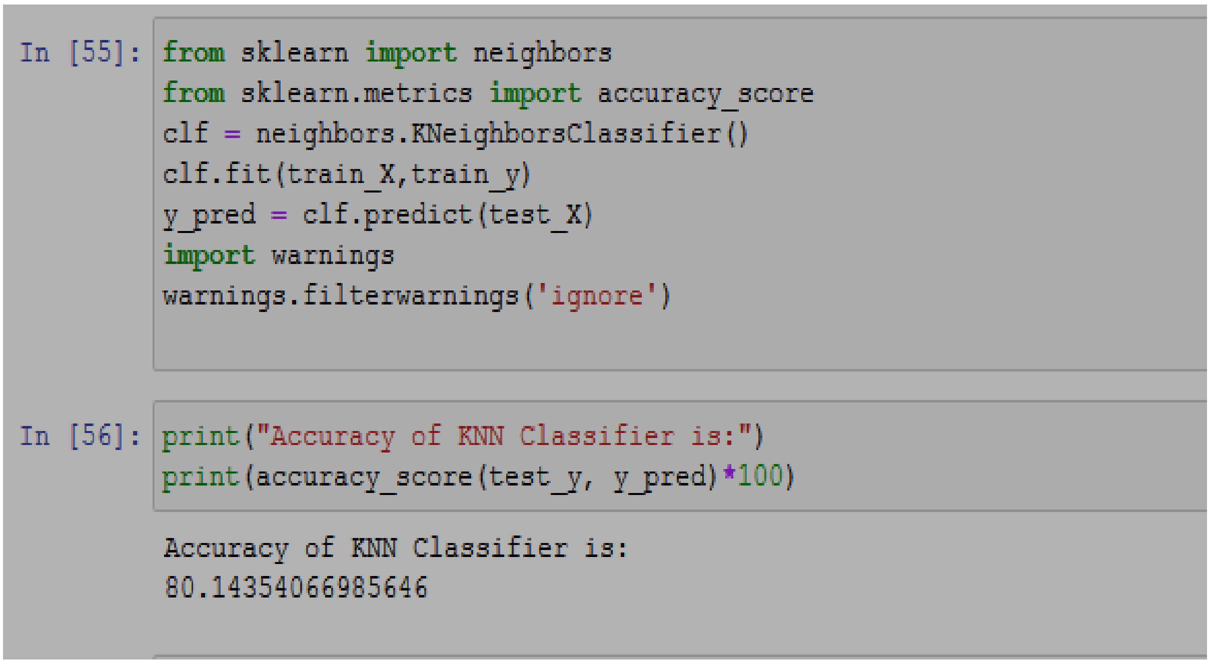
8.3.2 Read the file

```
abalone=pd.read_csv('abalone .csv')
```

8.3.3 How to train and test the data?

```
nobs=1

x_train,x_test=abalone[:-nobs],abalone[-nobs:]

abalone.head()
```

8.4 Accuracy of the Original Dataset



```
In [55]: from sklearn import neighbors
         from sklearn.metrics import accuracy_score
         clf = neighbors.KNeighborsClassifier()
         clf.fit(train_X,train_y)
         y_pred = clf.predict(test_X)
         import warnings
         warnings.filterwarnings('ignore')

In [56]: print("Accuracy of KNN Classifier is:")
         print(accuracy_score(test_y, y_pred)*100)

         Accuracy of KNN Classifier is:
         80.14354066985646
```

Fig 8.1 Accuracy of the original dataset

Accuracy of the original Dataset is 80.14**.**

8.5 Implementation of Granger Causality Hypothesis Test

8.5.1 Import Granger Causality Test methods

Granger causality test is pre-defined function in python which can be used by importing “statsmodels.tsa.stattools” package.

```
from statsmodels.tsa.stattools import grangercausalitytests
```

```
grangercausalitytests(data[r ,c], maxlag = maxlag , verbose=False)
```

Arguments:

- Data [r, c] is a two-dimensional matrix,
- max-lag is maximum number of iteration up to which granger causality test will run, we decide max-lag value by choosing the minimum from many corresponding Akaike Information Criteria(AIC) value.
- verbose represent current state of function.

Code snippet:

```
def grangers_causation_matrix(data, variables, test='ssr_chi2test', verbose=False):

    X_train = pd.DataFrame(np.zeros((len(variables), len(variables))), columns=variables, index=variables)

    for c in X_train.columns:

        for r in X_train.index:

            test_result = grangercausalitytests(data[[r, c]], maxlag=maxlag, verbose=False)

            p_values = [round(test_result[i+1][0][test][1],4) for i in range(maxlag)]

            if verbose: print(f'Y={r}, X={c}, P Values = {p_values}')

            min_p_value=np.min(p_values)

            X_train.loc[r, c] = min_p_value

    X_train.columns =[var + '_x' for var in variables]

    X_train.index =  [var + '_y' for var in variables]

    return X_train
```

How to call this function?

```
grangers_causation_matrix(x_train,variables=x_train.columns)
```

Output of Granger Causality test:

m*n dimension matrix which consist of p-values of each attributes corresponding to the whole dataset.

	Sex_x	**Length_x**	**Diameter_x**	**Height_x**	**W Weight_x**	**S Weight_x**	**V Weight_x**	**Shell weight_x**	**Rings_x**
Sex_y	1.0000	0.0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Length_y	0.0000	1.0	0.5603	0.0158	0.8123	0.0007	0.3785	0.0051	0.0000
Diameter_y	0.0000	0.0	1.0000	0.0510	0.1925	0.0000	0.0149	0.0630	0.0000
Height_y	0.0000	0.0	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000
W Weight_y	0.0000	0.0	0.0000	0.7873	1.0000	0.0000	0.0001	0.0133	0.0000
S Weight_y	0.0004	0.0	0.0000	0.1625	0.0015	1.0000	0.0000	0.0063	0.0000
V Weight_y	0.0000	0.0	0.0000	0.0035	0.0000	0.0000	1.0000	0.0000	0.0043
Shell weight_y	0.0000	0.0	0.0000	0.0003	0.0000	0.0000	0.0000	1.0000	0.0000
Rings_y	0.0000	0.0	0.0000	0.0000	0.0000	0.0025	0.0000	0.0000	1.0000

Table 8.1 Result of Granger Causality test

By the help of p-values, we conclude that how many attributes can be affected by one and from how many attributes one would affect. By simplifying this, we observe which attributes are significant to dataset.

We decide the result on the basis of P-Values. If the P-value is less than significance value (0.05), then null hypothesis (H_0) will be rejected and alternate hypothesis(H_1) will be selected which represent the given variable cause the required variable.

After analysing the result, we get that the attribute ‘Age’ and ‘Rings’ is of no-use and we are going to drop these attributes, and check the accuracy of this dataset.

8.6 Check the accuracy

8.6.1 Drop the attributes which was reduced our accuracy

```
X=abalone.drop(['Age','Rings'], axis=1)
```

8.6.2 Import the required classifier

```
from sklearn import neighbors

from sklearn.metrics import accuracy_score

import warnings
```

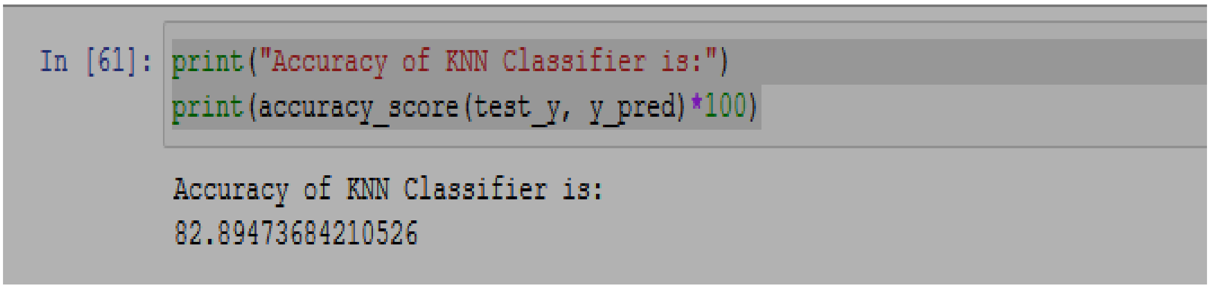
8.6.3 How to get accuracy by knn method

```
clf = neighbors.KNeighborsClassifier()

clf.fit(train_X,train_y)

y_pred = clf.predict(test_X)

warnings.filterwarnings('ignore')
```

Print the accuracy of Knn by calling accuracy_score method by passing the tes and prediction argument.

```
Print ("Accuracy of KNN Classifier is:")

print (accuracy_score (test_y, y_pred) * 100)
```

8.6.4 Output:



```
In [61]: print("Accuracy of KNN Classifier is:")
         print(accuracy_score(test_y, y_pred)*100)

         Accuracy of KNN Classifier is:
         82.89473684210526
```

Fig 8.2 Accuracy after implementing Granger Causality test

The accuracy of the data increased from 80.14 to 82.89.

After applying different classifier like

- **SVM (Support Vector Machine):** Accuracy increased from ***51.59 to 68.77*** and
- **Decision Tree**: Accuracy increased from ***83.82 to 84.08***.

8.7 Implementation of ANOVA (F-test)

An ANOVA test is a way to find out if experiments results are significant or not. ANOVA help us to figure out if there is need to **reject** the null hypothesis or **accept** the alternate hypothesis.

8.7.1 Steps in ANOVA

1. Define Hypothesis
2. Calculate the Sum of Squares
3. Determine degrees of freedom
4. F-value
5. Accept or Reject the Null Hypothesis

8.7.2 Impact of Variance

Variance is the measurement of the spread between numbers in a variable. It measures how far a number is from the mean and every number in a variable.

The variance of a feature determines how much it is impacting the response variable. If the variance is low, it implies there is no impact of this feature on response and vice-versa.

8.7.3 F-Distribution

A probability distribution generally used for the analysis of variance. It assumes Hypothesis as

H0: Two variances are equal

H1: Two variances are not equal.

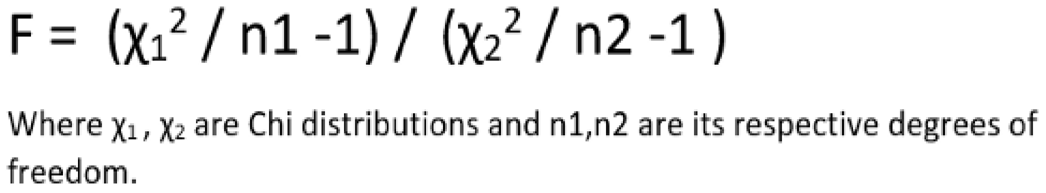
8.7.4 Degrees of Freedom

Degrees of freedom refers to the maximum number of logically independent values, which have the freedom to vary. In simple words, it can be defined as the total number of observations minus the number of independent constraints imposed on the observations.

Df = N -1 where N is the Sample Size

8.7.5 Finding F- Value

It is the ratio of two Chi-distributions divided by its degrees of Freedom.



$$F = (\chi_1^2 / n1 -1) / (\chi_2^2 / n2 -1)$$

Where χ_1 , χ_2 are Chi distributions and n1,n2 are its respective degrees of freedom.

After finding the value of F we decide which attribute we are going to take and which are not.

8.7.6 Output

In this dataset, we did not find any attribute that fades away by the ANOVA. So, the accuracy remains the same as the original.

But we find increment in accuracy in others dataset. In the next section We will show you the output of the accuracy for every dataset.

8.8 Accuracy of all dataset in different classifier

In this we will show present the accuracy of different dataset in different classifier.

- **Different Dataset:** Pima, Heart-Disease, Hepatitis B, Lung-Cancer, Dermatology, Lymphography, Breast- cancer, Arrhythmia, Parkinsons Disease, and Abalone (Already shown above).
- **Different Classifier:** k-NN, Support Vector Machine, Decision tree.

Dataset	Classifier	Original Accuracy (%)	After ANOVA	After Granger
Pima	k-NN	66	79.22	74.68
	SVM	64	69.48	85.71
	Decision tree	100	100	100
Heart-Disease	k-NN	87	87.66	89
	SVM	80.32	81.97	92.5
	Decision tree	70.04	79.35	100
Hepatitis B	k-NN	63	70.8	70
	SVM	60	78.10	63
	Decision tree	100	100	100
Lung-Cancer	k-NN	54	58.04	57
	SVM	75.7	87.5	82.4
	Decision tree	100	100	100
Dermatology	k-NN	63.01	64.38	64.38
	SVM	60.27	60.27	60.27
	Decision tree	100	100	100
Lymphography	k-NN	72.88	82.66	79.66
	SVM	79.66	84.35	81.36
	Decision tree	68.88	77.03	82.22
Breast- cancer	k-NN	91.47	95.61	95.61
	SVM	95.28	97.27	97.27

	Decision tree	95.1	95.7	95.7
Arrhythmia	k-NN	62.22	66.9	77.78
	SVM	64.40	84	81.11
	Decision tree	95.58	97.13	96.61
Abalone	k-NN	80.14	80.14	82.89
	SVM	51.59	51.59	68.77
	Decision tree	83.82	83.82	84.08
Parkinsons	k-NN	53.27	67.62	62.5
	SVM	52.5	63.14	55
	Decision tree	98.86	99.83	99.12

Table 8.2 Accuracy of ten different dataset in different classifier

Conclusion

Our approach for feature selection is using fully data-driven techniques such as ANOVA Testing and Granger Causality Hypothesis test becomes effective. Using these technique, we enhance the accuracy of the model by eliminating trivial attributes present in the dataset.

We have increased the classification performance in the dataset using the ANOVA (F-Test) and Granger Causality Test in KNN (ANOVA- 79.22%, Granger Causality- 74.68%) and many other classifiers.

Future Scope

The future improvement from this study is to design a feature selection algorithm for high dimensional multiclass dataset with considerable improvement in accuracy with less space and time requirement.

we will also use factor analysis technique to reduce a large number of variables into fewer number of factors. By using this technique, we extract maximum common variance from all variables and put them into a common score. As an index of all variables, we will use this score for further analysis.

References

1) Sankar Narayan Patra, Amrita Prasad, Soumya Roy, Gautam Bhattacharya, Subhash Chandra Panja, Koushik Ghosh : Causality Analysis between Solar Irradiance and Forbush Decrease Indices
2) Granger Causality : http://www.scholarpedia.org/article/Granger_causality
3) Andreas Lindholm, Niklas Wahlström,Fredrik Lindsten, Thomas B. Schön : “ Supervised Machine Learning “, http://www.it.uu.se/edu/course/homepage/sml/literature/lecture_notes.pdf
4) Breast Cancer Dataset : https://www.kaggle.com/uciml/breast-cancer-wisconsin-data
5) Tingquan Deng and Jinhong Yang :- An Improved Semisupervised Outlier Detection Algorithm Based on Adaptive Feature Weighted Clustering
6) https://towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561
7) https://www.cs.cmu.edu/~kdeng/thesis/feature.pdf
8) https://raw.githubusercontent.com/chrisalbon/sim_data/master/titanic.csv
9) https://livebook.manning.com/book/real-world-machine-learning/chapter-3/1

10) http://pages.stern.nyu.edu/~fprovost/Papers/selective_data_acq.pdf

11) https://towardsdatascience.com/data-preprocessing-concepts-fa946d11c825