ECE 421 – 6 Month Industrial Training Project

on

SPAM CLASSIFIER USING NLP & ML

Submitted in partial

fulfillment of the requirement

for the degree of

BACHELORS OF TECHNOLOGY

IN

ELECTRONICS & COMMUNICATION ENGINEERING

Submitted to

Jasdeep Mam

DEPARTMENT OF ELECTRONICS TECHNOLOGY

GURU NANAK DEV UNIVERISTY

AMRITSAR (PUNJAB)

Session: 2019-20

TUSHAR MAHAJAN

2016ECA1112

BTech. (ECE) Sem VIII

ACKNOWLEDGEMENT

It gives me immense pleasure to be associated with this project work. The project was a joyous learning process. The presentation of this report in the way required has been made possible by the way of contribution of various people.

Firstly, I would like to express our special thanks of gratitude to our teachers who gave me the golden opportunity to work along with the special thanks to INFOGAIN INDIA PVT. LTD. that has made me learn a lot of new technologies and trained me to them and also has helped me to work upon it and learn from my mistakes. This has helped me in doing a lot of research and I came to know about so many new things. I am very thankful for the advice, assistance & constant support throughout the preparation of this report. I cannot repay them for what they did for me, I cannot forget that and I am glad they helped me in doing this.

Secondly, we would also like to thank our parents and friends who helped us a lot in finalizing this project within the limited time frame.

- Tushar Mahajan

ABSTRACT

Human language is the most unstructured type of data, and yet we effortlessly parse and interpret it, and even generate our own. On the other hand, understanding everyday language is a significant challenge for machines; this is the focus of natural language processing (NLP)—the crossroads between linguistics and AI. In the 1950s, Alan Turing conjectured if a machine can fool a human into believing that he/she is speaking with another human, the machine exhibits intelligence—the iconic Turing test. Recently, a machine arguably passed the Turing test for the first time; the milestone was largely attributed to advances in NLP.

Natural Language Processing is a vital field of research having applications in different subjects. Text Classification is a part of NLP where the text is converted into a machine-readable form by performing various methods. Tokenizing, part-of-speech tagging, stemming, chunking are some of the text classification methods. Implementing these methods on our data gives us a classified data on which we will train the model to detect spam and ham messages using Scikit-Learn Classifiers.

I have proposed a model to solve the issue of classifying messages as spam or ham by experimenting and analyzing the relative strengths of several machine learning algorithms such as K-Nearest Neighbors (KNN), Naïve Bayes Classifier, Logistic Regression & Decision Tree to have a logical comparison of the performance measures of the methods we utilized in this research. The algorithm we proposed achieved a good accuracy on ‘SMS Spam Collection’ dataset.

LIST OF FIGURES

TABLE OF CONTENTS

ABOUT THE COMPANY

INFOGAIN is a Silicon Valley headquartered company with software platform engineering and deep domain expertise in the travel, retail, insurance and high technology industries. We accelerate the delivery of digital customer engagement systems using digital technologies such as cloud, microservices, robotic process automation and artificial intelligence to our clients.

Infogain delivers positive business outcomes for Fortune 500 companies and digital natives, using rapid prototyping and a solid foundation of DevSecOps-based software platform engineering that ensure high-quality and on-time delivery. A ChrysCapital portfolio company, Infogain has offices in California, Washington, Texas, London, Dubai, India and Singapore, with delivery centers in Austin, New Delhi, Bangalore, Pune, and Mumbai.

CORE VALUES

1. We believe that being helpful towards our peers and demonstrating cooperation, open-mindedness and compassion allows us to **Grow Together**.
2. We **Create Client Value**, ensuring that we thoroughly understand our client's needs and how to drive value for their business.
3. We aim at continuous **Innovation** by undertaking new challenges and learning from them, as well as by thinking outside the box.
4. We constantly strive to **Deliver Excellence** to our clients, shareholders and colleagues. Working as a dedicated team, Infogain strives to deliver exceptional quality of work.

INFOGAIN provides services in the domains like Retail, Insurance, Travel & Hospitality, Digital & High Technology and Healthcare. Some services are:

RETAIL: Our strategy combines digital transformation with retail value chain accelerators and automation to enhance the NextGen consumer experience. We offer stellar delivery capabilities in Merchandising, Retail POS and ORPOS, Retail Integration, omni-channel, and mobility giving retailers worldwide an edge over others.

Our solutions cover:

- Assessment, strategy, roadmap and evaluating ROI
- Mobile point-of-sale (mPOS) and personalization capabilities for a modern shopping experience
- Omni-commerce strategies and enterprise architecture for a consistent and intuitive shopping experiences through a variety of channels
- Frictionless consumer experience for creative Practice around UI/UX and web portals modernization
- Prescriptive methodologies for short order implementations on Stores and Merchandising solutions
- Shared services support model combined with Robotic Process Automation (RPA) at L2 level for higher operational efficiency
- Business Assurance (UAP™) and Test Automation services

INSURANCE: Digital Transformation, mobile and the Internet of Things (IoT) are major disruptors for the insurance industry. Insurers that address these disruptors will increase customer engagement, which will ultimately lead to future growth and profitability. Infogain provides customized digital and technology solutions for Auto Insurance, Workers Compensation, Personal and Commercial lines of business.

Benefits include:

- Efficient global knowledge management for better search accuracy
- Enhanced productivity with organized data integration and analytics
- Improved and seamless end-customer digital experience
- Improved Claim Processing efficiencies

TRAVEL & HOSPITALITY: It's a digital globe. Right now, 53 percent of all travel is booked online, and mobile is responsible for 94% of year-on-year growth in e-commerce travel traffic. According to a study by American Express, 83% of Millennials said they would let travel companies track their digital patterns, as long as it gave them a more personalized travel experience. We view digital transformation as business model transformation, not just an offering of a few unintegrated technology solutions like some vendors.

When you partner with us, you can expect:

- A larger share of the market
- Improved profit margins
- Increased customer loyalty
- Lower operational costs with optimized operations
- A superior track record of delivery

DIGITAL & HIGH TECHNOLOGY: Our Silicon Valley, CA headquartered company has been ‘engineering business outcomes’ for nearly 30 years. The digital & high technology practice offers software innovation, digital transformation and Oracle implementation expertise to privately held start-up “unicorn” enterprises to large technology conglomerates that operate globally. More recently, we have partnered with some members of the “FAMGA” (Facebook, Apple, Microsoft, Google, Amazon) group of companies. Extending beyond the Bay area, we deliver digital solutions to international enterprises in India, Asia Pacific, Europe and the Middle East.

When you partner with us, you can expect:

- Smart, user-centered, modernized software products—from strategy to deployment—based on real-world customer needs.
- Strong partnerships with digital firms specializing in RPA, AI, social, cloud, mobility and analytics as well as platform leaders Salesforce, Amazon Web Services, Google Cloud, Microsoft Azure and Oracle.
- Executive-level involvement from our leadership with average 20 plus years of experience delivering multi-million dollar engagements at Fortune 500 companies.
- Valuable knowledge, thanks to our strong track record of industry firsts—from the first-ever Oracle Knowledge Advanced implementation to the first implementation of a mobile Lithium platform.

HEALTHCARE: Serving as an extension of your healthcare IT team, our solutions lead to lower costs and improved patient outcomes. Our core Healthcare Services include Healthcare Application Development, Mobile Healthcare Apps, Patient Engagement Portals, and Healthcare Claims Analytics. Infogain optimizes business processes that improve accuracy, quality, outcomes and customer satisfaction.

Benefits to Business Include:

- Faster decision-making with analytics services that handle large volumes of information
- Faster time-to-market, leveraging established development frameworks
- Innovative digitalized ideas and transformational IT roadmaps to drive sustained, profitable growth
- Maintaining regulatory requirements
- Optimal efficiency

Various pillars of company include Revenue, Cost, Speed to Value, Risk & Innovation. Engineering Services include Digital Transformation, Quality Assurance, Package Implementation, Platform Engineering Solutions, Infrastructure Management and Data & Insight.

INTRODUCTION

This report features all the work that I have been doing in this past semester since January 2020. As I got placed with **Infogain India Pvt. Ltd.,** I got the opportunity to work with them and learn the things I always wanted to work upon but because of lack of skills and mentorship, I was not able to fulfil that work of mine. I always had keen interest in this Data Science & AI/ML field so I got the chance to choose between my earlier known language, JAVA or I can start to learn something new, staring from PYTHON to end up with my goal i.e. AI/ML. I chose that only and started the journey of learning the things I have listed below in my foundation training and after that I got my AI/ML core training and did learning there. I am working on NLP right now and hence I will be implementing that NLP to classify SMS as either Spam or Ham (not-spam). I did a few online courses and specialization that has helped me to clear more concepts than ever. So, this report is just the explanation and an analogy of what I did in these days. I will showcase them in this report.

The spam filtering among messages helps the mobile user to have a good visualization of the inbox. Unnecessary messages will be marked as spam so users need not waste their time reading them. In this paper, we propose to classify data in the messages as either spam (unwanted) or ham(wanted) messages. We devised our own spam detector. The introduction of spam filters has helped to aid this problem by classifying emails coming into a user's inbox as spam or ham (nonspam). Over the years, several techniques for filtering spam have been devised.

Some of the most popular techniques are described below:

- **Listbased spam filtering:**

This filtering method involves filtering email based on who sent the message. One method is to use a blacklist of all senders that are considered spam, and then categorize every message from these senders as spam. Another method is to use a whitelist of senders that are considered trusted, and only allow emails from senders on this whitelist, labelling the rest as spam.

- **Rulebased spam filtering:**

This filtering method uses a set of rules to evaluate between an email being spam or ham. Some rulebased systems might use a point system, assigning certain point scores to the satisfaction of certain rules. The final score can then be evaluated against some score threshold to determine whether the message is spam or ham.

- **Challengeresponse spam filtering:**

This simple method of spam filtering takes into account the fact that spam is sent out in bulk, usually through some sort of automation. In the challengeresponse spam system, whenever a message is received from a sender that has not been preapproved, a reply is sent back in which the recipient must perform some sort of task to have their message approved. Since spam is sent out many thousands of messages at a time, it would be extremely impractical for the spammer to perform all of these challenges.

Disadvantages of popular techniques

Although the methods listed above work in many circumstances, and can often times filter out a large majority of spam emails, they all have their shortcomings. For example, filtering engines that use whitelists are very restrictive in that new trusted senders usually must be added manually. Filters that use blacklists must continually be updated, as spammers can gain access to new sources to send from as fast as those sources of spam are added to the blacklist. With a rulebased system, if a spammer can find the list of rules that a spam engine uses to filters emails, they can simply design their spam message to avoid all of these rules. Challengeresponse spam systems can make it laborious for other users to send emails to someone with the system, and emails that are sent by automated systems, but that are not considered spam (like a mailing list or store receipt) will be erroneously blocked.

LEARNINGS DURING THE TRAINING

These are the various things learnt during the basic foundation course, like we had basic knowledge about all these topics and when going to the core side, we were made to learn and perform the things we learnt as our core training in deep.
Those topics are:

- **PYTHON**

Python is a programming language that lets you work quickly and integrate systems more effectively. It is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

We learnt the basics of python with their implementation and along with that the course also included various assignments to test our knowledge on how and what have we grasped till now and how much of it is implemented.

- **CLOUD COMPUTING**

Cloud computing is the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user. The term is generally used to describe data centers available to many users over the Internet. Cloud providers typically use a "pay-as-you-go" model, which can lead to unexpected operating expenses if administrators are not familiarized with cloud-pricing models. The availability of high-capacity networks, low-cost computers and storage devices as well as the widespread adoption of hardware virtualization, service-oriented architecture and autonomic and utility computing has led to growth in cloud computing.

Here, in this training, we learnt about the basics of cloud computing, various models and their implementation. Then we learnt about the top three cloud services namely Google Cloud Platform (GCP), Amazon Web Services (AWS) & Microsoft Azure. Learnt about the tools that used. Basically, learnt about the tools that present in them and what they do. We learnt all the tools using the Google Cloud Platform (GCP) and hence worked to make various virtual machines and hypervisor learning and all that.

- **MDM & DATA WAREHOUSE**

In business, master data management (MDM) is a method used to define and manage the critical data of an organization to provide, with data integration, a single point of reference. In computing, a master data management tool can be used to support master data management by removing duplicates, standardizing data (mass maintaining), and incorporating rules to eliminate incorrect data from entering the system in order to create an authoritative source of master data. The redundancy of party and account data is compounded in the front to back office life cycle, where the authoritative single source for the party, account and product data is needed but is often once again redundantly entered or augmented.

We learnt various ways of storing the data in warehouses, what are data marts, all the theoretical part was made understood for us. We came to know about the Golden Record and other things. We learnt about the various MDM approaches i.e. Registry, Consolidation, Coexistence and Centralized one.

- **PROBABILITY & STATISTICS**

Probability and Statistics form the basis of Data Science. The probability theory is very much helpful for making the prediction. Estimates and predictions form an important part of Data science. With the help of statistical methods, we make estimates for the further analysis. Thus, statistical methods are largely dependent on the theory of probability. And all of probability and statistics is dependent on Data.

In this probability and statistics training, we were given firstly the basic mean, variance, probability distribution and the basic probability theory. Then we worked out on the techniques, SD and their trends, Co-relation and Co-variance, all this theory was made understood to us.

Data is the collected information(observations) we have about something or facts and statistics collected together for reference or analysis. Data matters a lot nowadays as we can infer important information from it. Now let's delve into how data is categorized. Data can be of 2 types categorical and numerical data. For Example, in a bank, we have regions, occupation class, gender which follow categorical data as the data is within a fixed certain value and balance, credit score, age, tenure months follow numerical continuous distribution as data can follow an unlimited range of values.

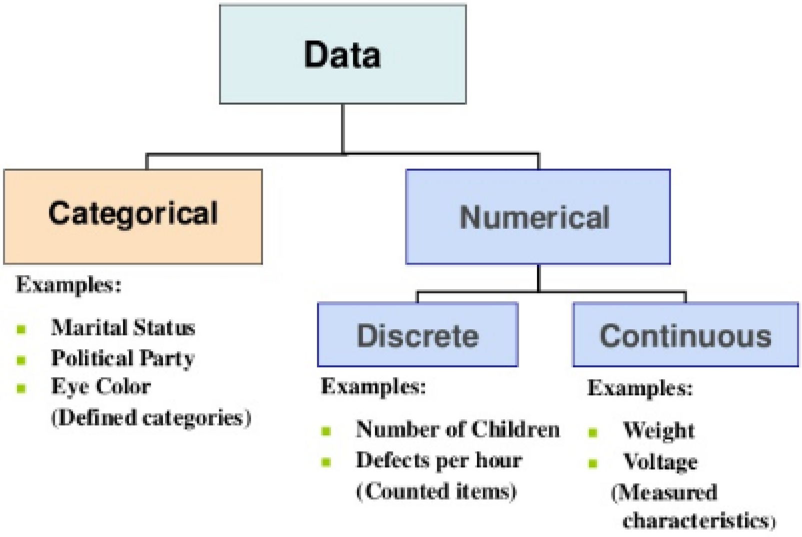


Figure 1: Types of Data

Note: Categorical Data can be visualized by Bar Plot, Pie Chart, Pareto Chart. Numerical Data can be visualized by Histogram, Line Plot, Scatter Plot

Descriptive Statistics: A descriptive statistic is a summary statistic that quantitatively describes or summarizes features of a collection of information. It helps us in knowing our data better. It is used to describe the characteristics of data.

The qualitative and quantitative data is very much similar to the above categorical and numerical data.

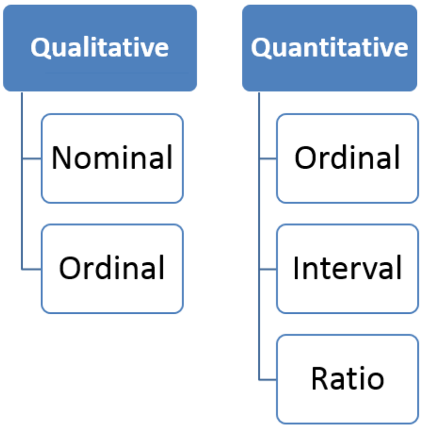


Figure 2: Measurement of Data

Nominal: Data at this level is categorized using names, labels or qualities. eg: Brand Name, ZipCode, Gender.
Ordinal: Data at this level can be arranged in order or ranked and can be compared. eg: Grades, Star Reviews, Position in Race, Date.

Interval: Data at this level can be ordered as it is in a range of values and meaningful differences between the data points can be calculated. eg: Temperature in Celsius, Year of Birth.

Ratio: Data at this level is similar to interval level with added property of an inherent zero. Mathematical calculations can be performed on these data points. eg: Height, Age, Weight.

Population or Sample Data

Before performing any analysis of data, we should determine if the data we're dealing with is population or sample.

Population: Collection of all items (N) and it includes each and every unit of our study. It is hard to define and the measure of characteristic such as mean, mode is called parameter.

Sample: Subset of the population (n) and it includes only a handful units of the population. It is selected at random and the measure of the characteristic is called as statistics.

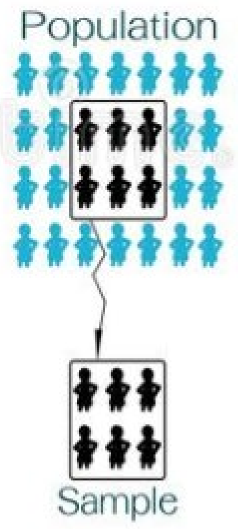


Figure 3: Population v/s Sample Data

Measures of Central Tendency:

The measure of central tendency is a single value that attempts to describe a set of data by identifying the central position within that set of data. As such, measures of central tendency are sometimes called measures of central location. They are also classed as summary statistics.

Mean: The mean is equal to the sum of all the values in the data set divided by the number of values in the data set i.e the calculated average. It susceptible to outliers when unusual values are added it gets skewed i.e. deviates from the typical central value.

Median: The median is the middle value for a dataset that has been arranged in order of magnitude. Median is a better alternative to mean as it is less affected by outliers and skewness of the data. The median value is much closer than the typical central value.

If the total number of values is odd then

$$\text{Median} = \left(\frac{n+1}{2}\right)^{th} \text{term}$$

If the total number of values is even then

$$\text{Median} = \left(\frac{\left(\frac{n}{2}\right)^{th} term + \left(\frac{n}{2}+1\right)^{th} term}{2}\right)^{\text{th}} term$$

Mode: The mode is the most commonly occurring value in the dataset. The mode can, therefore sometimes consider the mode as being the most popular option.

Measures of Asymmetry

Skewness: Skewness is the asymmetry in a statistical distribution, in which the curve appears distorted or skewed towards to the left or to the right. Skewness indicates whether the data is concentrated on one side.

Positive Skewness: Positive Skewness is when the mean>median>mode. The outliers are skewed to the right i.e. the tail is skewed to the right.

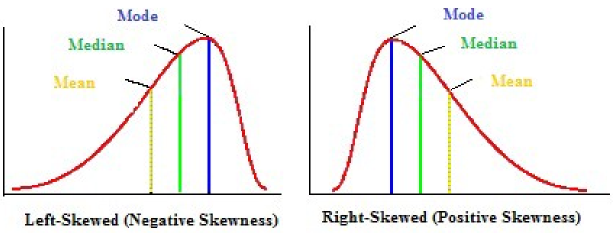


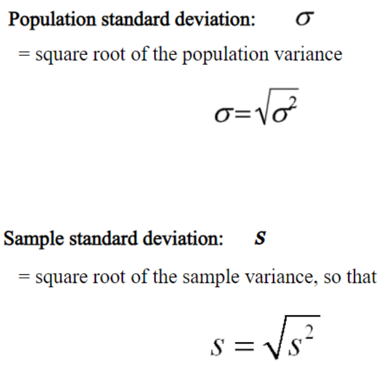
Figure 4: Left v/s Right Skewed

Negative Skewness: Negative Skewness is when the mean<median<mode. The outliers are skewed to the left i.e. the tail is skewed to the left.

Skewness is important as it tells us about where the data is distributed.

Measures of Variability (Dispersion)

The measure of central tendency gives a single value that represents the whole value; however, the central tendency cannot describe the observation fully. The measure of dispersion helps us to study the variability of the items i.e. the spread of data.



Population standard deviation: σ

= square root of the population variance

$$\sigma = \sqrt{\sigma^2}$$

Sample standard deviation: s

= square root of the sample variance, so that

$$s = \sqrt{s^2}$$

Figure 5: Population & Sample SD

Remember: Population Data has N data points and Sample Data has (n-1) data points. (n-1) is called Bessel's Correction and it is used to reduce bias.

Range: The difference between the largest and the smallest value of a data, is termed as the range of the distribution. Range does not consider all the values of a series, i.e. it takes only the extreme items and middle items are not considered significant.

Variance: Variance measures how far is the sum of squared distances from each point to the mean i.e. the dispersion around the mean.
Variance is the average of all squared deviations.
Note: The units of values and variance is not equal so we use another variability measure.

$$\sigma^2 = \frac{\sum (x_i - \mu)^2}{n} \text{ for populations}$$

$$s^2 = \frac{\sum (x_i - \bar{x})^2}{n-1} \text{ for samples}$$

Standard Deviation: As Variance suffers from unit difference so standard deviation is used. The square root of the variance is the standard deviation. It tells about the concentration of the data around the mean of the data set.

Coefficient of Variation(CV): It is also called as the relative standard deviation. It is the ratio of standard deviation to the mean of the dataset.

Standard deviation is the variability of a single dataset. Whereas the coefficient of variance can be used for comparing 2 datasets.

CV for a population:

$$CV = \frac{\sigma}{\mu} * 100\%$$

CV for a sample:

$$CV = \frac{s}{\bar{x}} * 100\%$$

Figure 6: Coeff. of Variation

Measures of Relationship
Measures of relationship are used to find the comparison between 2 variables.

Covariance: Covariance is a measure of the relationship between the variability of 2 variables i.e. It measures the degree of change in the variables, when one variable changes, will there be the same/a similar change in the other variable. Covariance does not give effective information about the relation between 2 variables as it is not normalized.

Correlation: Correlation gives a better understanding of covariance (normalized covariance). Correlation tells us how correlated the variables are to each other. It is also called as *Pearson Correlation Coefficient.*

$$Correlation = \rho = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}$$

The value of correlation ranges from -1 to 1. -1 indicates negative correlation i.e with an increase in 1 variable independent there is a decrease in the other dependent variable.1 indicates positive correlation i.e. with an increase in 1 variable independent there is an increase in the other dependent variable.0 indicates that the variables are independent of each other.

- **SQL**

SQL is a database computer language designed for the retrieval and management of data in a relational database. SQL stands for Structured Query Language. This tutorial will give you a quick start to SQL. It covers most of the topics required for a basic understanding of SQL and to get a feel of how it works.

SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

Also, they are using different dialects, such as −

- MS SQL Server using T-SQL,
- Oracle using PL/SQL,
- MS Access version of SQL is called JET SQL (native format) etc.

Applications of SQL

As mentioned before, SQL is one of the most widely used query language over the databases. I'm going to list few of them here:

- Allows users to access data in the relational database management systems.
- Allows users to describe the data.
- Allows users to define the data in a database and manipulate that data.
- Allows to embed within other languages using SQL modules, libraries & pre-compilers.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database.
- Allows users to set permissions on tables, procedures and views.

Here, we were learnt this SQL so that we can further use those queries to work on SQL Server and also prepare us further so that we can work upon the Big Data and Hadoop Systems.

- **BIG DATA/ HADOOP**

Apache Hadoop is open-source software that facilitates a network of computers to solve problems that require massive datasets and computation power. Hadoop is highly scalable, that is designed to accommodate computation ranging from a single server to a cluster of thousands of machines. While Hadoop is written in Java, you can program in Hadoop using multiple languages like Python, C++, Perl, Ruby etc.

There are three main components of Hadoop –

Hadoop Distributed Filesystem – It is the storage component of Hadoop. Hadoop is a collection of master-slave networks. In HDFS there are two daemons – namenode and datanode that run on the master and slave nodes respectively.

Map-Reduce – This part of Hadoop is responsible for high-level data processing. It facilitates processing of a large amount of data over the cluster of nodes.

YARN – It is used for resource management and job scheduling. In a multi-node cluster, it is difficult to manage, allocate and release the resources.

Data Science is a vast field. It stems from multiple interdisciplinary fields like mathematics, statistics, and programming. It is about finding patterns in data. Data Scientists are trained for extracting, analyzing and generating predictions from the data. It is an umbrella term that incorporates almost every technology that involves the use of data.

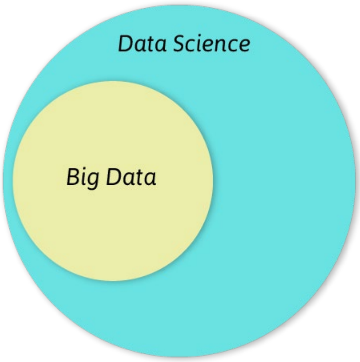


Figure 7: Data Science v/s Big Data

The main functionality of Hadoop is storage of Big Data. It also allows the users to store all forms of data, that is, both structured data and unstructured data. Hadoop also provides modules like Pig and Hive for analysis of large-scale data. However, the difference between data science and big data is that the former is a discipline that involves all the data operations. As a result, Big Data is a part of Data Science. Since Data Science contains a sea of information, it is not necessary to know Big Data. A Data Scientist needs to be inclusive about all the data related operations. Therefore, having expertise at Big Data and Hadoop will allow you to develop a comprehensive architecture analyzes a colossal amount of data.

Anatomy of Hadoop

Some of the major components of Hadoop are –

- Hadoop Distributed File System (HDFS)
- MapReduce
- YARN
- Hive
- Pig
- HBase

Over the past few years, Hadoop has been increasingly used for implementing data science tools in the industries. With the assimilation of big data and data science, industries have been able to fully leverage data science.

Traditionally, machine learning engineers had to deal with a limited amount of data, which ultimately resulted in the low performance of their models. However, with the help of Hadoop ecosystem that provides linear scalable storage, you can store all the data in RAW format.

- **MACHINE LEARNING**

The origin of machine learning can be traced back to a series of profound events in the 1950s in which pioneering research established computers' ability to learn. In 1950, the famous *"Turing Test"* was developed by the English mathematician Alan Turing to determine if a machine exhibits intelligent behavior equal or similar to a human. In 1952, the data scientist *Arthur Lee Samuel* managed to teach an IBM computer program to not only learn the game of checkers but to improve the more it played.

Then in 1957, the world's first neural network for computers was designed by the American psychologist Frank Rosenblatt. From there, experimentation escalated. In the 1960s, Bayesian methods for probabilistic interference in machine learning were introduced.

Hardware for efficient processing. The next momentous event that helped enable machine learning as we know it today is hardware advancements which occurred in the early 2000s. Graphics processing units (GPUs) were developed that could not only speed up algorithm training significantly—from weeks to days— but could also be used in embedded systems. In 2009, Nvidia's GPUs were used by the famous Google Brain to create capable deep neural networks that could learn to recognize unlabeled pictures of cats from YouTube.

- **TENSORFLOW**

TensorFlow is a free and open - source library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google. TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache License 2.0 on November 9, 2015.

TensorFlow can train and run deep neural networks for handwritten digit classification, image recognition, word embeddings, recurrent neural networks, sequence-to-sequence models for machine translation, natural language processing, and PDE (partial differential equation) based simulations. Best of all, TensorFlow supports production prediction at scale, with the same models used for training.

TensorFlow provides all of this for the programmer by way of the Python language. Python is easy to learn and work with, and provides convenient ways to express how high-level abstractions can be coupled together. Nodes and tensors in TensorFlow are Python objects, and TensorFlow applications are themselves Python applications.

The actual math operations, however, are not performed in Python. The libraries of transformations that are available through TensorFlow are written as high-performance C++ binaries. Python just directs traffic between the pieces, and provides high-level programming abstractions to hook them together.

- **DATA ANALYSIS**

Data analysis is defined as a process of cleaning, transforming, and modeling data to discover useful information for business decision-making. The purpose of Data Analysis is to extract useful information from data and taking the decision based upon the data analysis.
Whenever we take any decision in our day-to-day life is by thinking about what happened last time or what will happen by choosing that particular decision. This is nothing but analyzing our past or future and making decisions based on it. For that, we gather memories of our past or dreams of our future. So that is nothing but data analysis. Now same thing analyst does for business purposes, is called Data Analysis.

Why Data Analysis?

To grow your business even to grow in your life, sometimes all you need to do is Analysis! If your business is not growing, then you have to look back and acknowledge your mistakes and make a plan again without repeating those mistakes. And even if your business is growing, then you have to look forward to making the business to grow more. All you need to do is analyze your business data and business processes.

Data Analysis Tools:

There are a lot many tools that are used for data analysis and some of them are listed below:

- Lumify
- R Programming
- Apache Spark
- IDEA
- Xplenty

There are several types of data analysis techniques that exist based on business and technology. The major types of data analysis are:

Text Analysis is also referred to as Data Mining. It is a method to discover a pattern in large data sets using databases or data mining tools. It used to transform raw data into business information. Business Intelligence tools are present in the market which is used to take strategic business decisions.

Statistical Analysis shows "What happen?" by using past data in the form of dashboards. Statistical Analysis includes collection, Analysis, interpretation, presentation, and modeling of data. It analyses a set of data or a sample of data.

There are two categories of this type of Analysis - Descriptive and Inferential Analysis.

- **Descriptive Analysis** analyses complete data or a sample of summarized numerical data. It shows mean and deviation for continuous data whereas percentage and frequency for categorical data.
- **Inferential Analysis** analyses sample from complete data. In this type of Analysis, you can find different conclusions from the same data by selecting different samples.

Diagnostic Analysis shows "Why did it happen?" by finding the cause from the insight found in Statistical Analysis. This Analysis is useful to identify behavior patterns of data. If a new problem arrives in your business process, then you can look into this Analysis to find similar patterns of that problem.

Predictive Analysis shows "what is likely to happen" by using previous data. The simplest example is like if last year I bought two dresses based on my savings and if this year my salary is increasing double then I can buy four dresses. So here, this Analysis makes predictions about future outcomes based on current or past data. Its accuracy is based on how much detailed information you have and how much you dig in it.

Prescriptive Analysis combines the insight from all previous Analysis to determine which action to take in a current problem or decision. Most data-driven companies are utilizing Prescriptive Analysis because predictive and descriptive Analysis are not enough to improve data performance. Based on current situations and problems, they analyze the data and make decisions.

- **DATA ANALYSIS PROCESS**

Data Analysis Process is nothing but gathering information by using proper application or tool which allows you to explore the data and find a pattern in it. Based on that, you can take decisions, or you can get ultimate conclusions.

Data Analysis consists of the following phases:

- Data Requirement Gathering
- Data Collection
- Data Cleaning
- Data Analysis
- Data Interpretation
- Data Visualization
- Data Requirement Gathering

First of all, you have to think about why do you want to do this data analysis? All you need to find out the purpose or aim of doing the Analysis.

Data Collection

After requirement gathering, you will get a clear idea about what things you have to measure and what should be your findings. Now it's time to collect your data based on requirements. Once you collect your data, remember that the collected data must be processed or organized for Analysis. As you collected data from various sources, you must have to keep a log with a collection date and source of the data.

Data Cleaning

Now whatever data is collected may not be useful or irrelevant to your aim of Analysis, hence it should be cleaned. The data which is collected may contain duplicate records, white spaces or errors. This phase must be done before Analysis because based on data cleaning, your output of Analysis will be closer to your expected outcome.

Data Analysis

Once the data is collected, cleaned, and processed, it is ready for Analysis. As you manipulate data, you may find you have the exact information you need, or you might need to collect more data. During this phase, you can use data analysis tools and software which will help you to understand, interpret, and derive conclusions based on the requirements.

Data Interpretation

After analyzing your data, it's finally time to interpret your results. You can choose the way to express or communicate your data analysis either you can use simply in words or maybe a table or chart. Then use the results of your data analysis process to decide your best course of action.

Data Visualization

Data visualization is very common in your day to day life; they often appear in the form of charts and graphs. In other words, data shown graphically so that it will be easier for the human brain to understand and process it. By observing relationships and comparing datasets, you can find a way to find out meaningful information.

All these are the process that are used during data analysis and it implements what all I have learnt till now in ML i.e. the Exploratory Data Analysis, Use of ML Tools to predict things, apply various ML Algorithms to do and calculate answers for tedious work.

- **ML ALGORITHMS**

There are different ways an algorithm can model a problem based on its interaction with the experience or environment or whatever we want to call the input data. It is popular in machine learning and artificial intelligence textbooks to first consider the learning styles that an algorithm can adopt.

There are only a few main learning styles or learning models that an algorithm can have and we'll go through them here with a few examples of algorithms and problem types that they suit. This taxonomy or way of organizing machine learning algorithms is useful.

Let's take a look at three different learning styles in machine learning algorithms:

Supervised Learning

Input data is called training data and has a known label or result such as spam/not-spam or a stock price at a time.
A model is prepared through a training process in which it is required to make predictions and is corrected when those predictions are wrong. The training process continues until the model achieves a desired level of accuracy on the training data.
Example problems are classification and regression.
Example algorithms include: Logistic Regression and the Back Propagation Neural Network.

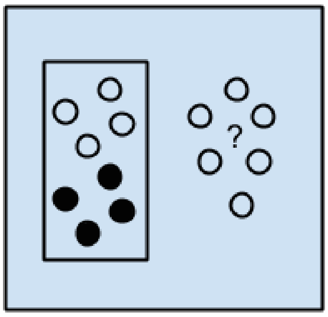


Figure 8: Supervised Learning

Unsupervised Learning

Input data is not labeled and does not have a known result.
A model is prepared by deducing structures present in the input data. This may be to extract general rules. It may be through a mathematical process to systematically reduce redundancy, or it may be to organize data by similarity.
Example problems are clustering, dimensionality reduction and association rule learning.
Example algorithms include: the Apriori algorithm and K-Means.

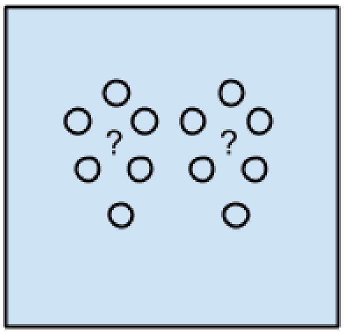


Figure 9: Unsupervised Learning Algorithms

Semi-Supervised Learning

Input data is a mixture of labeled and unlabelled examples. There is a desired prediction problem but the model must learn the structures to organize the data as well as make predictions. Example problems are classification and regression. Example algorithms are extensions to other flexible methods that make assumptions about how to model the unlabeled data.

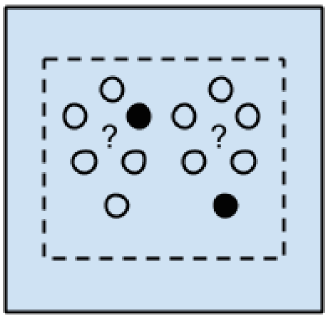


Figure 10: Semi-supervised Learning Algorithms

Algorithms Grouped By Similarity

Algorithms are often grouped by similarity in terms of their function (how they work). For example, tree-based methods, and neural network inspired methods. There are also categories that have the same name that describe the problem and the class of algorithm such as Regression and Clustering.

Regression Algorithms

Regression is concerned with modeling the relationship between variables that is iteratively refined using a measure of error in the predictions made by the model.

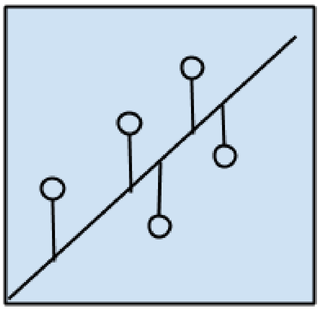


Figure 11: Regression Algorithms

This may be confusing because we can use regression to refer to the class of problem and the class of algorithm. Really, regression is a process.

The most popular regression algorithms are:

- Linear Regression
- Logistic Regression
- Stepwise Regression
- Multivariate Adaptive Regression Splines (MARS)

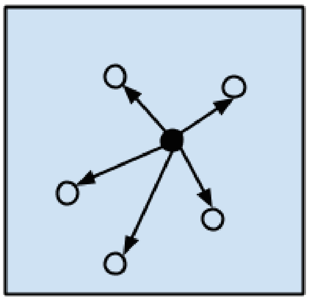


Figure 12: Instance Based Learning

Instance-based learning model

It is a decision problem with instances or examples of training data that are deemed important or required to the model. Such methods typically build up a database of example data and compare new data to the database using a similarity measure in order to find the best match.

The most popular instance-based algorithms are:

- k-Nearest Neighbor (kNN)
- Learning Vector Quantization (LVQ)
- Support Vector Machines (SVM)

Decision Tree Algorithms

Decision tree methods construct a model of decisions made based on actual values of attributes in the data. Decisions fork in tree structures until a prediction decision is made for a given record. Decision trees are trained on data for classification and regression problems. Decision trees are often fast and accurate and a big favorite in machine learning.

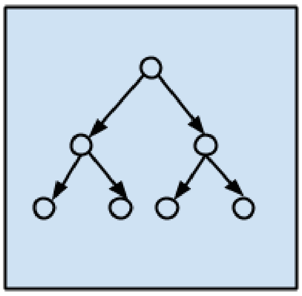


Figure 13: Decision Tree Algorithms

The most popular decision tree algorithms are:

- Classification and Regression Tree (CART)
- Conditional Decision Trees

Bayesian Algorithms

Bayesian methods are those that explicitly apply Bayes' Theorem for problems such as classification and regression.

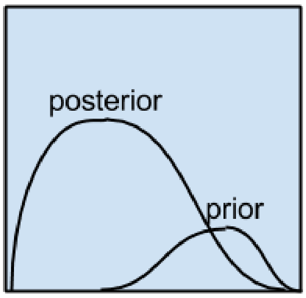


Figure 14: Bayesian Algorithms

The most popular Bayesian algorithms are:

- Naive Bayes
- Gaussian Naive Bayes

Clustering Algorithms

Clustering, like regression, describes the class of problem and the class of methods. Clustering methods are typically organized by the modeling approaches such as centroid-based and hierarchal. All methods are concerned with using the inherent structures in the data to best organize the data into groups of maximum commonalities.

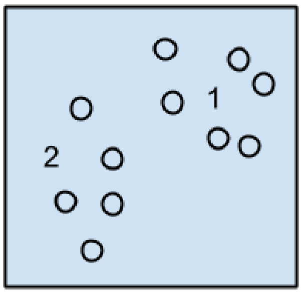


Figure 15: Clustering Algorithms

The most popular clustering algorithms are:

- k-Means
- Hierarchical Clustering

Artificial Neural Network Algorithms

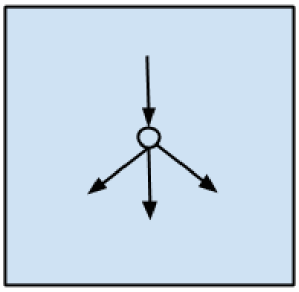


Figure 16: ANN Algorithms

Artificial Neural Networks are models that are inspired by the structure and/or function of biological neural networks. They are a class of pattern matching that are commonly used for regression and classification problems but are really an enormous subfield comprised of hundreds of algorithms and variations for all manner of problem types.

The most popular artificial neural network algorithms are:

- Perceptron
- Multilayer Perceptrons (MLP)
- Back-Propagation
- Stochastic Gradient Descent

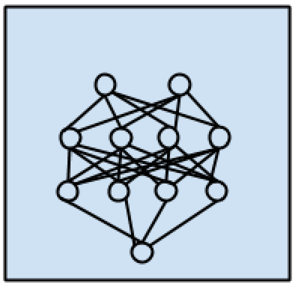


Figure 17: Deep Learning Algorithms

Deep Learning Algorithms

Deep Learning methods are a modern update to Artificial Neural Networks that exploit abundant cheap computation. They are concerned with building much larger and more complex neural networks and, as commented on above, many methods are concerned with very large datasets of labelled analog data, such as image, text. audio, and video.

The most popular deep learning algorithms are:

- Convolutional Neural Network (CNN)
- Recurrent Neural Networks (RNNs)
- Long Short-Term Memory Networks (LSTMs)

Dimensionality Reduction Algorithms

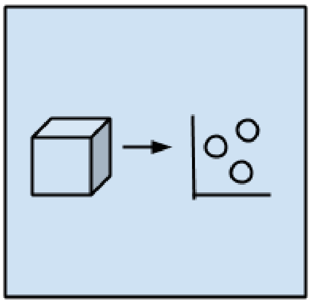


Figure 18: Dimensionality Reduction Algorithms

Like clustering methods, dimensionality reduction seek and exploit the inherent structure in the data, but in this case in an unsupervised manner or order to summarize or describe data using less information. This can be useful to visualize dimensional data or to simplify data which can then be used in a supervised learning method.

Many of these methods can be adapted for use in classification and regression.

- Principal Component Analysis (PCA)
- Principal Component Regression (PCR)

Ensemble Algorithms

Ensemble methods are models composed of multiple weaker models that are independently trained and whose predictions are combined in some way to make the overall prediction.Much effort is put into what types of weak learners to combine and the ways in which to combine them. This is a very powerful class of techniques and as such is very popular.

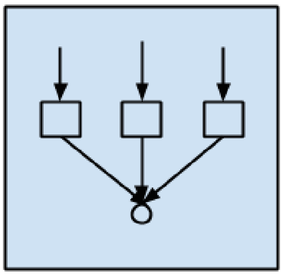


Figure 19: Ensemble Algorithms

Some algorithms are:

- Boosting
- Bootstrapped Aggregation (Bagging)
- Random Forest

- **NEURAL NETWORKS & DEEP LEARNING**

Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must be translated.

Neural networks help us cluster and classify. You can think of them as a clustering and classification layer on top of the data you store and manage. They help to group unlabeled data according to similarities among the example inputs, and they classify data when they have a labeled dataset to train on.

Here are a few examples of what deep learning can do:

Classification

All classification tasks depend upon labeled datasets; that is, humans must transfer their knowledge to the dataset in order for a neural network to learn the correlation between labels and data. This is known as supervised learning.

- Detect faces, identify people in images, recognize facial expressions (angry, joyful)
- Identify objects in images (stop signs, pedestrians, lane markers…)
- Recognize gestures in video
- Detect voices, identify speakers, transcribe speech to text, recognize sentiment in voices
- Classify text as spam (in emails), or fraudulent (in insurance claims); recognize sentiment in text (customer feedback)
- Any labels that humans can generate, any outcomes that you care about and which correlate to data, can be used to train a neural network.

Clustering

Clustering or grouping is the detection of similarities. Deep learning does not require labels to detect similarities. Learning without labels is called unsupervised learning. Unlabeled data is the majority of data in the world. One law of machine learning is: the more data an algorithm can train on, the more accurate it will be. Therefore, unsupervised learning has the potential to produce highly accurate models.

- **Search:** Comparing documents, images or sounds to surface similar items.
- **Anomaly detection:** The flipside of detecting similarities is detecting anomalies, or unusual behavior. In many cases, unusual behavior correlates highly with things you want to detect and prevent, such as fraud.

Neural Network Elements

Deep learning is the name we use for "stacked neural networks"; that is, networks composed of several layers.

The layers are made of nodes. A node is just a place where computation happens, loosely patterned on a neuron in the human brain, which fires when it encounters sufficient stimuli. A node combines input from the data with a set of coefficients, or weights, that either amplify or dampen that input, thereby assigning significance to inputs with regard to the task the algorithm is trying to learn. These input-weight products are summed and then the sum is passed through a node's so-called activation function, to determine whether and to what extent that signal should progress further through the network to affect the ultimate outcome, say, an act of classification. If the signals passes through, the neuron has been "activated."

Here's a diagram of what one node might look like.

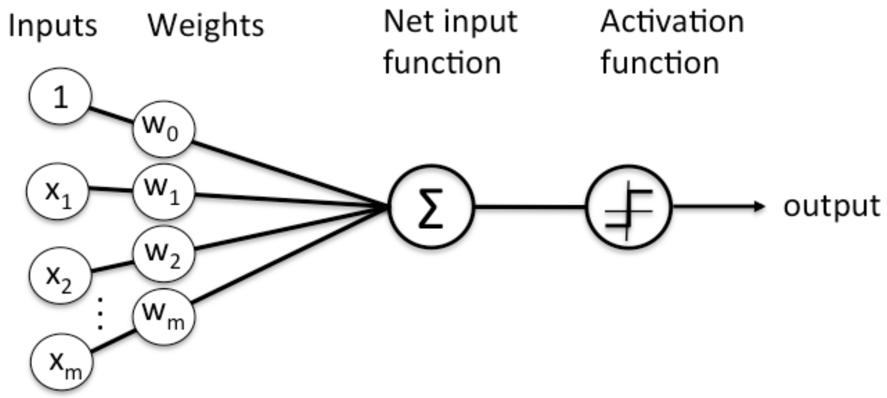
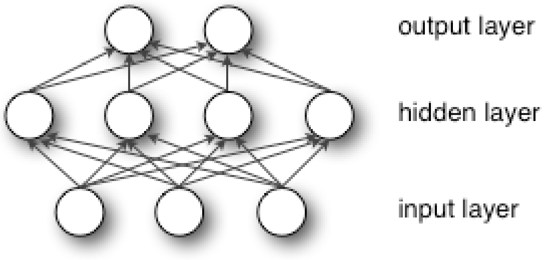


Figure 20: A basic neural network

A node layer is a row of those neuron-like switches that turn on or off as the input is fed through the net. Each layer's output is simultaneously the subsequent layer's input, starting from an initial input layer receiving your data.

Pairing the model's adjustable weights with input features is how we assign significance to those features with regard to how the neural network classifies and clusters input.

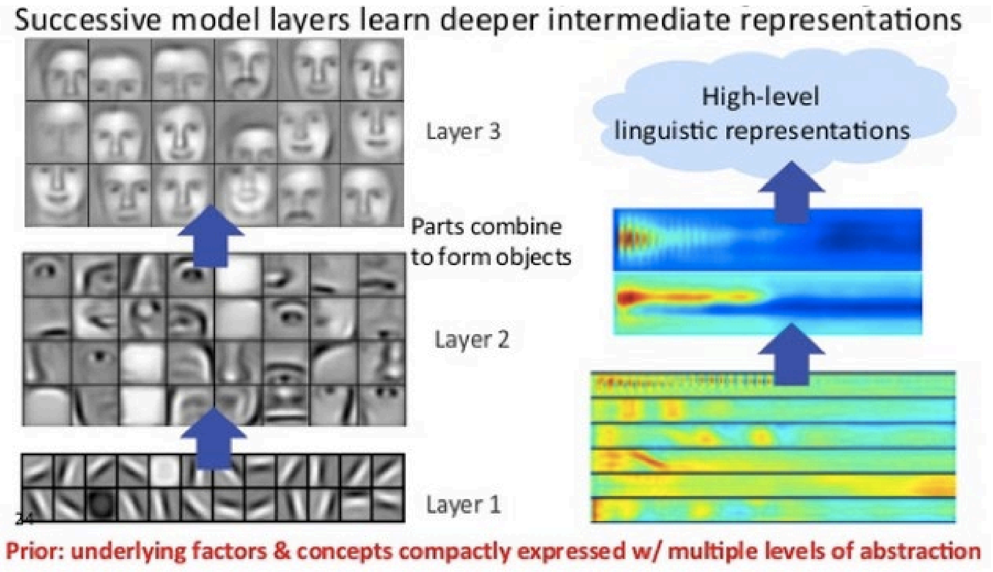


Key Concepts of Deep Neural Networks

Deep-learning networks are distinguished from the more commonplace single-hidden-layer neural networks by their depth; that is, the number of node layers through which data must pass in a multistep process of pattern recognition.

Earlier versions of neural networks such as the first perceptrons were shallow, composed of one input and one output layer, and at most one hidden layer in between. More than three layers (including input and output) qualifies as "deep" learning.

In deep-learning networks, each layer of nodes trains on a distinct set of features based on the previous layer's output. The further you advance into the neural net, the more complex the features your nodes can recognize, since they aggregate and recombine features from the previous layer.


Successive model layers learn deeper intermediate representations
Layer 3
High-level
linguistic representations
Parts combine
to form objects
Layer 2
Layer 1
Prior: underlying factors & concepts compactly expressed w/ multiple levels of abstraction

This is known as feature hierarchy, and it is a hierarchy of increasing complexity and abstraction. It makes deep-learning networks capable of handling very large, high-dimensional data sets with billions of parameters that pass through nonlinear functions.

Above all, these neural nets are capable of discovering latent structures within unlabeled, unstructured data, which is the vast majority of data in the world. Another word for unstructured data is raw media; i.e. pictures, texts, video and audio recordings. Therefore, one of the problems deep learning solves best is in processing and clustering the world's raw, unlabeled media, discerning similarities and anomalies in data that no human has organized in a relational database or ever put a name to.

A deep-learning network trained on labeled data can then be applied to unstructured data, giving it access to much more input than machine-learning nets. Deep-learning networks end in an output layer: a logistic, or softmax, classifier that assigns a likelihood to a particular outcome or label. We call that predictive, but it is predictive in a broad sense.

Gradient Descent

The name for one commonly used optimization function that adjusts weights according to the error they caused is called “gradient descent.” Gradient is another word for slope, and slope, in its typical form on an x-y graph, represents how two variables relate to each other: rise over run, the change in money over the change in time, etc.

As a neural network learns, it slowly adjusts many weights so that they can map signal to meaning correctly. The relationship between network Error and each of those weights is a derivative, *dE/dw*, that measures the degree to which a slight change in a weight causes a slight change in the error.

Each weight is just one factor in a deep network that involves many transforms; the signal of the weight passes through activations and sums over several layers, so we use the chain rule of calculus to march back through the networks activations and outputs and finally arrive at the weight in question, and its relationship to overall error.

The chain rule in calculus states that

$$\frac{dz}{dx} = \frac{dz}{dy} \cdot \frac{dy}{dx}.$$

In a feedforward network, the relationship between the net’s error and a single weight will look something like this:

$$\frac{dError}{dweight} = \frac{dError}{dactivation} * \frac{dactivation}{dweight}$$

That is, given two variables, Error and weight, that are mediated by a third variable, activation, through which the weight is passed, you can calculate how a change in weight affects a change in Error by first calculating how a change in activation affects a change in Error, and how a change in weight affects a change in activation. The essence of learning in deep learning is nothing more than that: adjusting a model’s weights in response to the error it produces, until you can’t reduce the error any more.

Some examples of **optimization algorithms** include *AdaDelta, Adam, None, RMSProp, SGD & line gradient descent.*

Some examples of **activation function** include *identity, LeakyReLu, RelationalTanh, ReLu, Sigmoid, Softmax &Tanh.*

- **NATURAL LANGUAGE PROCESSING**

Natural language processing (NLP) is a subfield of linguistics, computer science, information engineering, and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of natural language data. It is a branch of artificial intelligence that deals with the interaction between computers and humans using the natural language. It is basically the processing done to the data so that the machine can understand the language we use in our day to day life.

What is NLP used for?

- Natural Language Processing is the driving force behind the following common applications:
- Language translation applications such as Google Translate
- Word Processors such as Microsoft Word and Grammarly that employ NLP to check grammatical accuracy of texts.

In this, we learnt various processes of natural language processing i.e. the steps that are involved in Preprocessing. **Preprocessing** is defined as the process that is usually followed to process the data before it is fed to the machine for further analysis. Many methods are used for doing preprocessing.

They are:

- Lowercase the data
- Filter out the string (like removing the HTML, emails, unnecessary data, auto-generated texts, ASCII values, contractions, accented characters etc.)
- Remove the punctuations.
- Remove the stop words.
- Tokenization.
- Next is either Stemming or Lemmatization.
    - The *Lemmatization* is used to keep the '*Lemma'* i.e. the root word behind every word and it is always a dictionary word whereas *Stemming* is a rigorous approach which just slashes the last alphabet(s) and hence fail to achieve the dictionary meaning. Both have their own use based on requirement occurred.
- Now we apply various methodologies like *BagOfWords(BOW), Word2Vec, TF-IDF* etc. to do the further analysis.

TOOLS USED

I have made a small project using python notebooks using the **JUPYTER NOTEBOOK** and software named **ANACONDA** by creating an environment there**.** It basically has the implementation of various topics learnt during this training till now. Using that knowledge, I will be making a **Spam SMS Classifier** and use different algorithms on it and get the output along with the accuracy score. Also, at the end, I have share with you the certificates which show the completion of extra courses during the training time.

There are a few software and programming languages that are used here:

I. Python

Further the libraries that are used in this project include:

i) Pandas
ii) Numpy
iii) NLTK
iv) Seaborn
v) Matplotlib

II. Anaconda Environment

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS.

III. Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

SPAM CLASSIFIER USING NLP AND APPLYING DIFFERENT ML MODELS

This spam classifier programme uses the dataset `train.csv.` We then see the data summaries and describe the data components which is usually known as ‘knowing your data’. After that, we will firstly view a few visualizations to see the data components and then we will train the data using the algorithms and at the end will test the model we created on our test data and then we will calculate the accuracy and hence the confusion matrix is generated. *Confusion Matrix* is a table that is often used to describe the performance of a classification model (or “classifier”) on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm.

The four algorithms we will be using are listed below with their explanations:

i. Naïve Bayes
ii. Decision Tree
iii. Logistic Regression
iv. K- Nearest Neighbor

We will explain each and every model/algorithm here.

ALGORITHMS USED:

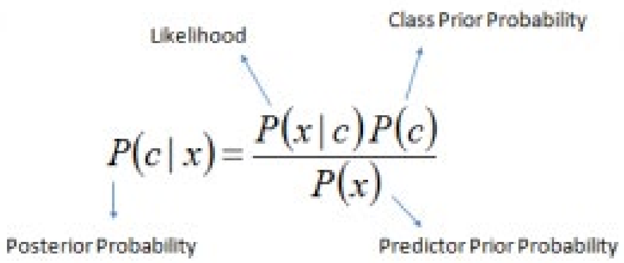
Naïve Bayes

It is a classification technique based on Bayes’ Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as ‘Naive’.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:



$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Figure 21: Naive Bayes

Above,

P(c|x) is the posterior probability of class (c, target) given predictor (x, attributes).
P(c) is the prior probability of class.
P(x|c) is the likelihood which is the probability of predictor given class.
P(x) is the prior probability of predictor.

Again, scikit learn (python library) will help here to build a Naive Bayes model in Python. There are three types of Naive Bayes model under the scikit-learn library:

- **Gaussian:** It is used in classification and it assumes that features follow a normal distribution.

- **Multinomial:** It is used for discrete counts. For example, let’s say, we have a text classification problem. Here we can consider Bernoulli trials which is one step further and instead of “word occurring in the document”, we have “count how often word occurs in the document”, you can think of it as “number of times outcome number x_i is observed over the n trials”.

- **Bernoulli:** The binomial model is useful if your feature vectors are binary (i.e. zeros and ones). One application would be text classification with ‘bag of words’ model where the 1s & 0s are “word occurs in the document” and “word does not occur in the document” respectively.

Decision Tree

Decision Tree is one of the easiest and popular classification algorithms to understand and interpret. Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too. The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data (training data).

In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

Types of Decision Trees

Types of decision trees are based on the type of target variable we have. It can be of two types:

- **Categorical Variable Decision Tree:** Decision Tree which has a categorical target variable then it called a Categorical variable decision tree.

- **Continuous Variable Decision Tree:** Decision Tree has a continuous target variable then it is called Continuous Variable Decision Tree.

Important Terminology related to Decision Trees

- **Root Node:** It represents the entire population or sample and this further gets divided into two or more homogeneous sets.
- **Splitting:** It is a process of dividing a node into two or more sub-nodes.
- **Decision Node:** When a sub-node splits into further sub-nodes, then it is called the decision node.
- **Leaf / Terminal Node:** Nodes do not split is called Leaf or Terminal node.
- **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can say the opposite process of splitting.
- **Branch / Sub-Tree:** A subsection of the entire tree is called branch or sub-tree.
- **Parent and Child Node:** A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.

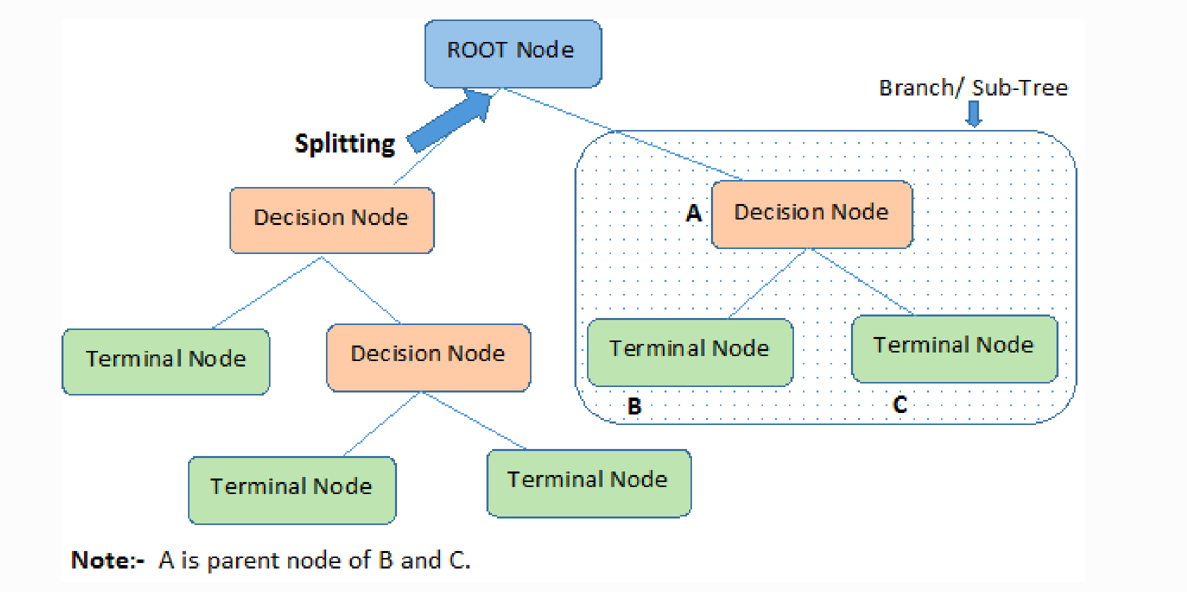


Figure 22: A sample Decision Tree

Decision trees classify the examples by sorting them down the tree from the root to some leaf/terminal node, with the leaf/terminal node providing the classification of the example. This process is recursive in nature and is repeated for every subtree rooted at the new node.

Logistic Regression

Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability.

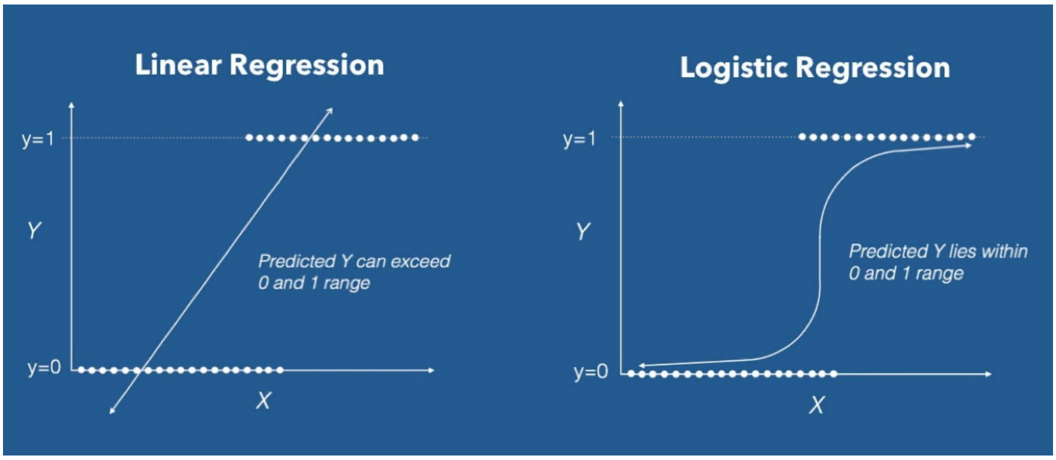


Figure 23: Linear v/s Logistic Regression

We can call a Logistic Regression a Linear Regression model but the Logistic Regression uses a more complex cost function, this cost function can be defined as the ‘Sigmoid function’ or also known as the ‘logistic function’ instead of a linear function.

The hypothesis of logistic regression tends it to limit the cost function between 0 and 1. Therefore linear functions fail to represent it as it can have a value greater than 1 or less than 0 which is not possible as per the hypothesis of logistic regression.

$$0 \leq h_\theta(x) \leq 1$$

What is the Sigmoid Function?

In order to map predicted values to probabilities, we use the Sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.

$$f(x) = \frac{1}{1+\mathrm{e}^{-(x)}}$$

Figure 25: Formula for Sigmoid Function

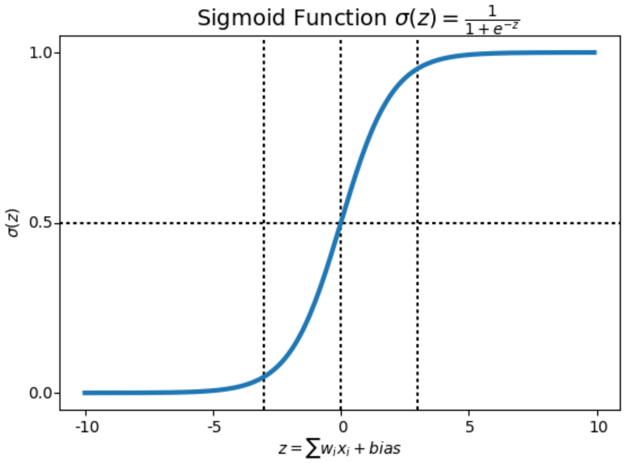


Figure 24: Sigmoid Function

Hypothesis Representation

When using linear regression, we used a formula of the hypothesis i.e.

hΘ(x) = $\beta_0 + \beta_1$X

For logistic regression we are going to modify it a little bit i.e.

σ(Z) = σ($\beta_0 + \beta_1$X)

We have expected that our hypothesis will give values between 0 and 1.

Z = $\beta_0 + \beta_1$X

hΘ(x) = sigmoid(Z)

i.e. hΘ(x) = 1/(1 + e^-($\beta_0 + \beta_1$X)

$$h\theta(X) = \frac{1}{1 + e^{-\,(\beta_0 + \beta_1 X)}}$$

Figure 26: Hypothesis Function

K-Nearest Neighbor

KNN algorithm is one of the simplest classification algorithms and it is one of the most used learning algorithms. KNN is a non-parametric, lazy learning algorithm. Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point.

KNN Algorithm is based on 'feature similarity': How closely out-of-sample features resemble our training set determines how we classify a given data point:

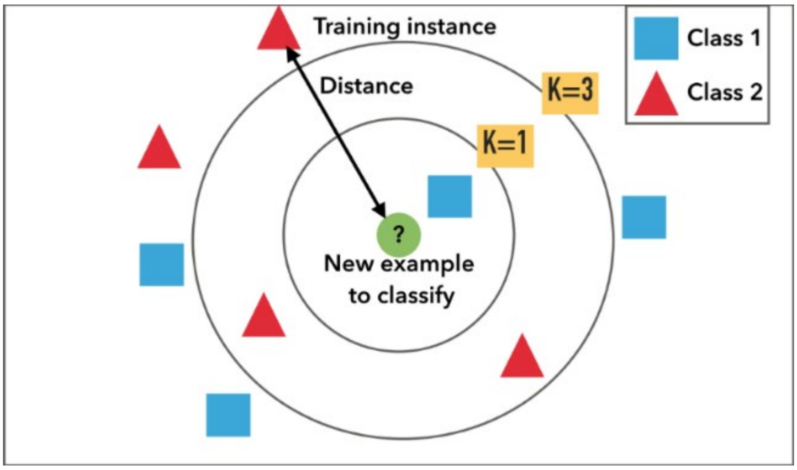


Figure 27: KNN Example

KNN can be used for classification — the output is a class membership (predicts a class — a discrete value). An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. It can also be used for regression — output is the value for the object (predicts continuous values). This value is the average (or median) of the values of its k nearest neighbors.

How does the KNN algorithm work?

Let's take a simple case to understand this algorithm. Following is a spread of red circles (RC) and green squares (GS):

You intend to find out the class of the blue star (BS). BS can either be RC or GS and nothing else. The "K" is KNN algorithm is the nearest neighbor we wish to take the vote from. Let's say K = 3. Hence, we will now make a circle with BS as the center just as big as to enclose only three datapoints on the plane.

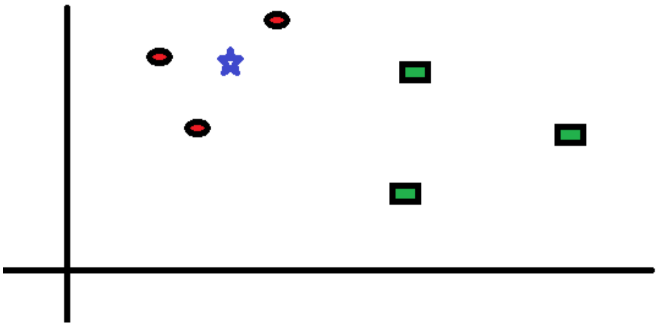


Figure 28: KNN Algorithm Explained -1

The three closest points to BS is all RC. Hence, with a good confidence level, we can say that the BS should belong to the class RC. Here, the choice became very obvious as all three votes from the closest neighbor went to RC. The choice of the parameter K is very crucial in this algorithm.

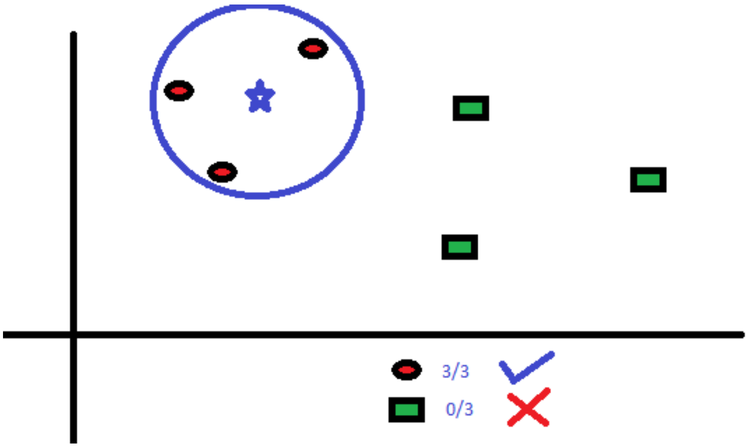


Figure 29: KNN Algorithm Explained -2

How do we choose the factor K?

First let us try to understand what exactly does K influence in the algorithm. If we see the last example, given that all the 6-training observation remain constant, with a given K value we can make boundaries of each class. These boundaries will segregate RC from GS. In the same way, let's try to see the effect of value "K" on the class boundaries. The following are the different boundaries separating the two classes with different values of K.

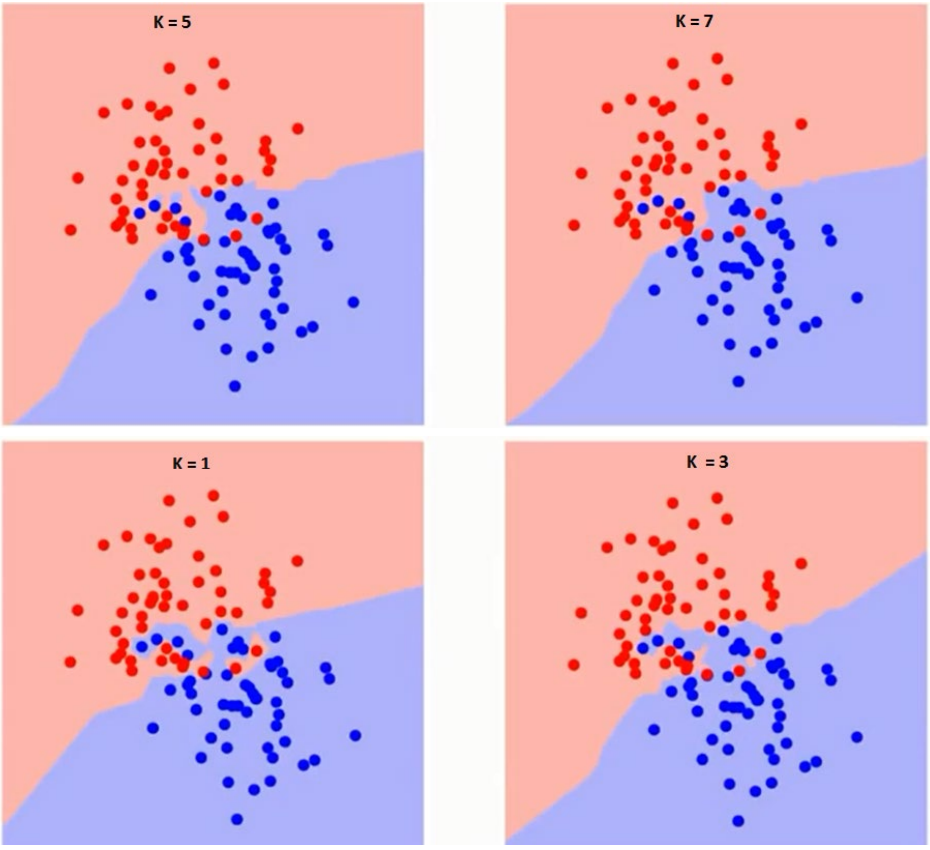


Figure 30: Effect of K on the output

If you watch carefully, you can see that the boundary becomes smoother with increasing value of K. With K increasing to infinity it finally becomes all blue or all red depending on the total majority. The training error rate and the validation error rate are two parameters we need to access different K-value. Following is the curve for the training error rate with a varying value of K:

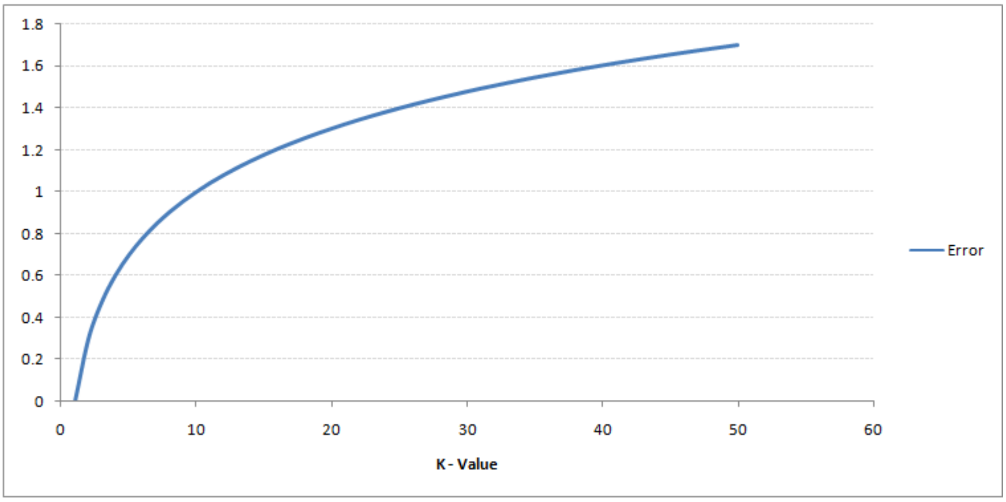


Figure 31: Value of K v/s Error

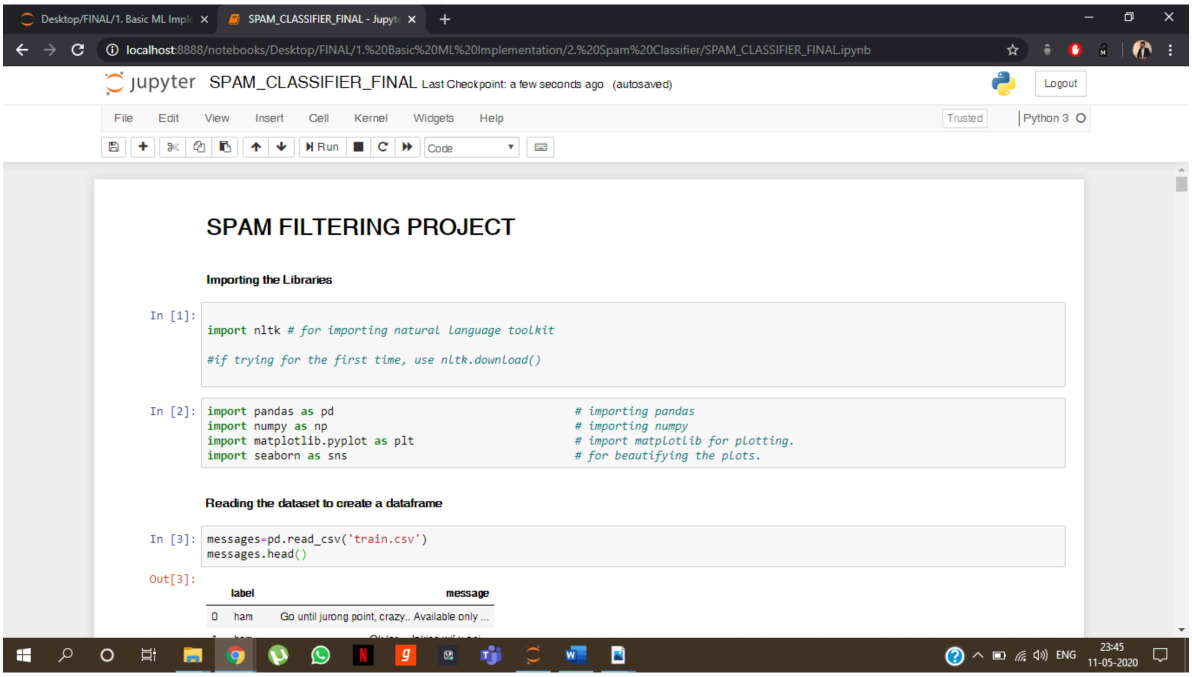
As you can see, the error rate at K=1 is always zero for the training sample. This is because the closest point to any training data point is itself. Hence the prediction is always accurate with K=1. If validation error curve would have been similar, our choice of K would have been 1.

PROCEDURE:

- The dataset is loaded and hence the basic summary is displayed using the commands.
- Then various plots are done on it so as to get the idea about how the data is displayed and is present in it.
- Then the data is cleaned using various NLP preprocessing techniques mentioned above and then the BagOfWords (BoW) is made out of it.
- The cleaned data is then label encoded so that it can be sent to the machine for processing and there is no error on it.
- The clean and processed dataset is then split into test & train.
- The model is created and then it is trained using the training data provided to us.
- Then that model is used to predict the output on the test data.
- Both the outputs i.e. the predicted and the original one is used to generate the accuracy score and the confusion matrix is generated out of it.

- Similarly, these steps are repeated for the rest of the models i.e. KNN, Decision Tree & Logistic Regression.
- The output of accuracy from different models is then compared using the visualization techniques.

SNAPSHOTS:


Desktop/FINAL/1. Basic ML Imple
SPAM_CLASSIFIER_FINAL - Jupyt
localhost:8888/notebooks/Desktop/FINAL/1.%20Basic%20ML%20Implementation/2.%20Spam%20Classifier/SPAM_CLASSIFIER_FINAL.ipynb
jupyter
SPAM_CLASSIFIER_FINAL Last Checkpoint: a few seconds ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
ENG
23:45
11-05-2020

SPAM FILTERING PROJECT

Importing the Libraries

```
In [1]: import nltk # for importing natural language toolkit

        #if trying for the first time, use nltk.download()
```

```
In [2]: import pandas as pd                                    # importing pandas
        import numpy as np                                     # importing numpy
        import matplotlib.pyplot as plt                        # import matplotlib for plotting.
        import seaborn as sns                                  # for beautifying the plots.
```

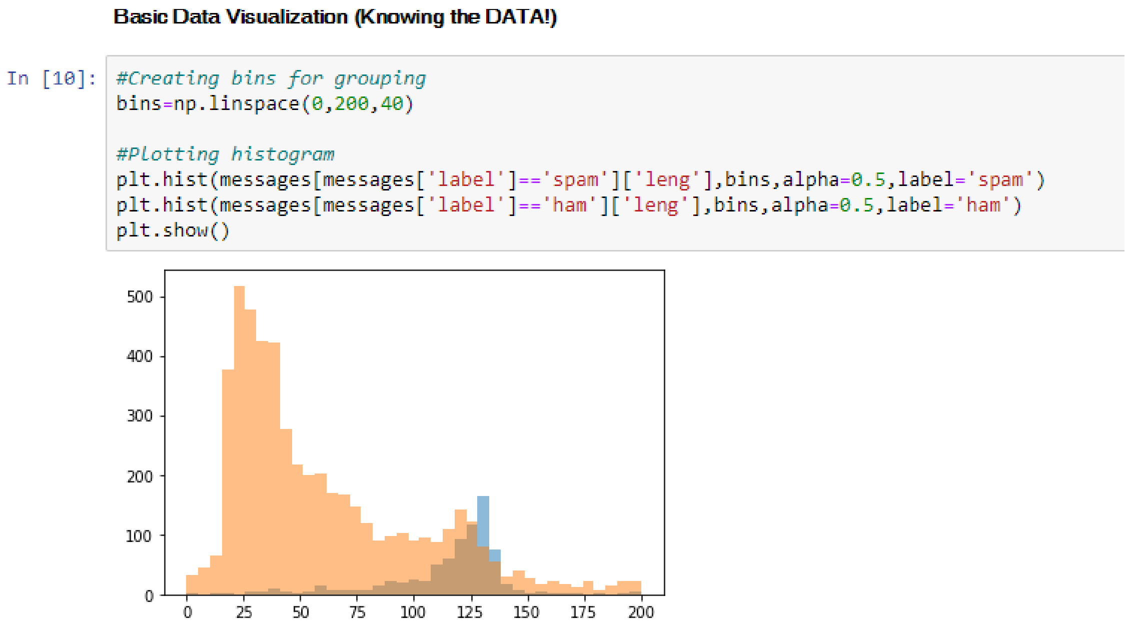
Reading the dataset to create a dataframe

```
In [3]: messages=pd.read_csv('train.csv')
        messages.head()
Out[3]:
```

	label	message
0	ham	Go until jurong point, crazy.. Available only ...

Figure 32: Basic Jupyter Notebook Interface

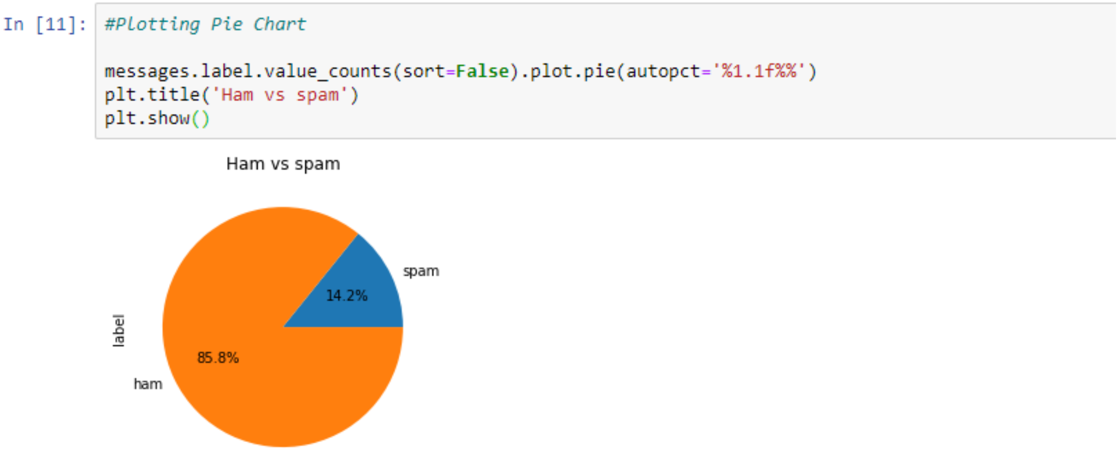
A few visualizations for knowing the data:



Basic Data Visualization (Knowing the DATA!)

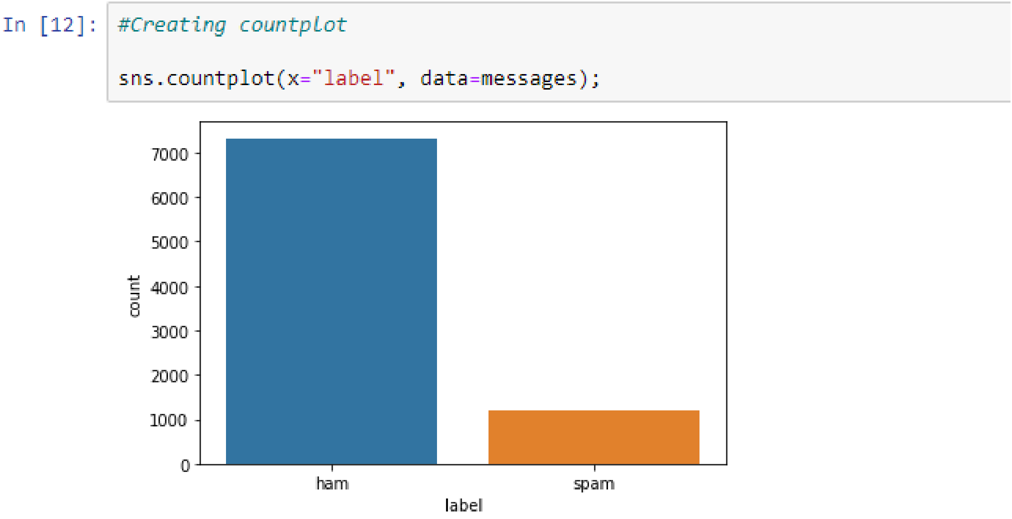
```
In [10]: #Creating bins for grouping
         bins=np.linspace(0,200,40)

         #Plotting histogram
         plt.hist(messages[messages['label']=='spam']['leng'],bins,alpha=0.5,label='spam')
         plt.hist(messages[messages['label']=='ham']['leng'],bins,alpha=0.5,label='ham')
         plt.show()
```

Figure 33: A Histogram



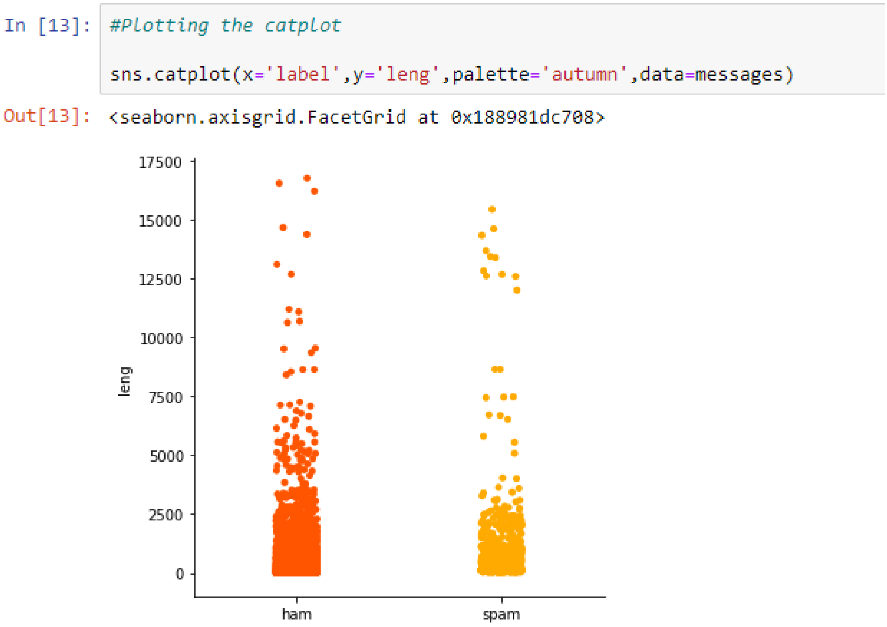
```
In [11]: #Plotting Pie Chart

         messages.label.value_counts(sort=False).plot.pie(autopct='%1.1f%%')
         plt.title('Ham vs spam')
         plt.show()
```

Figure 36: A Pie chart



```
In [12]: #Creating countplot

         sns.countplot(x="label", data=messages);
```

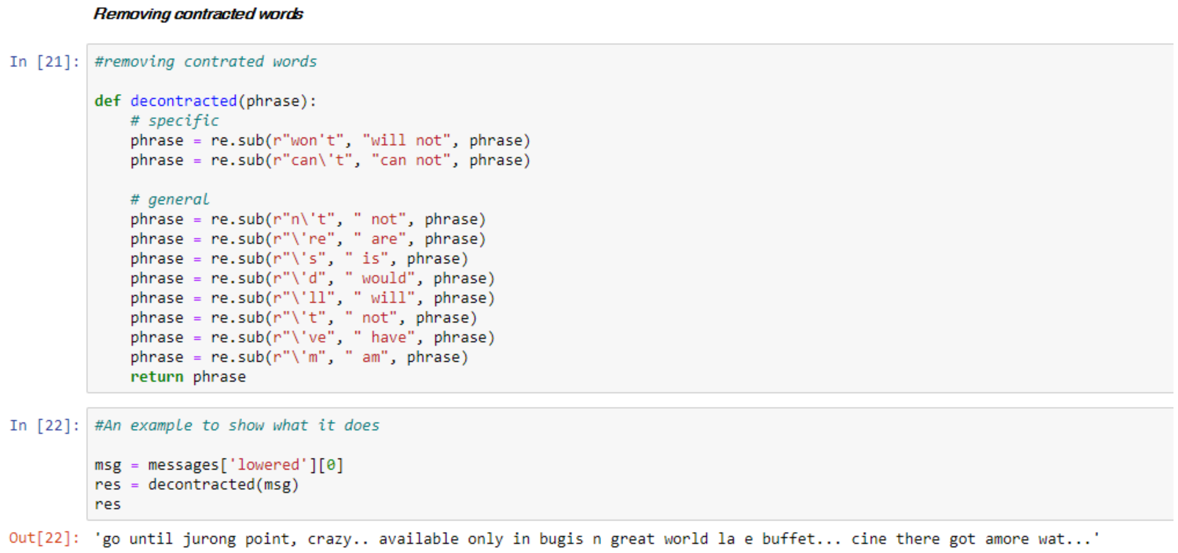
Figure 35: Count plot



```
In [13]: #Plotting the catplot

         sns.catplot(x='label',y='leng',palette='autumn',data=messages)

Out[13]: <seaborn.axisgrid.FacetGrid at 0x188981dc708>
```

Figure 34: Catplot

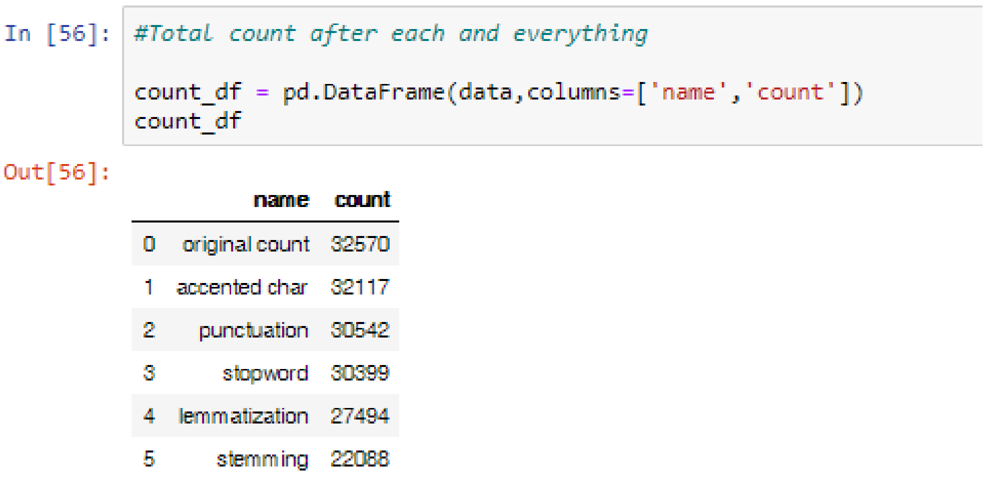
Creating a function and using it on dataset:



Removing contracted words

```
In [21]: #removing contrated words

         def decontracted(phrase):
             # specific
             phrase = re.sub(r"won't", "will not", phrase)
             phrase = re.sub(r"can\'t", "can not", phrase)

             # general
             phrase = re.sub(r"n\'t", " not", phrase)
             phrase = re.sub(r"\'re", " are", phrase)
             phrase = re.sub(r"\'s", " is", phrase)
             phrase = re.sub(r"\'d", " would", phrase)
             phrase = re.sub(r"\'ll", " will", phrase)
             phrase = re.sub(r"\'t", " not", phrase)
             phrase = re.sub(r"\'ve", " have", phrase)
             phrase = re.sub(r"\'m", " am", phrase)
             return phrase

In [22]: #An example to show what it does

         msg = messages['lowered'][0]
         res = decontracted(msg)
         res

Out[22]: 'go until jurong point, crazy.. available only in bugis n great world la e buffet... cine there got amore wat...'
```

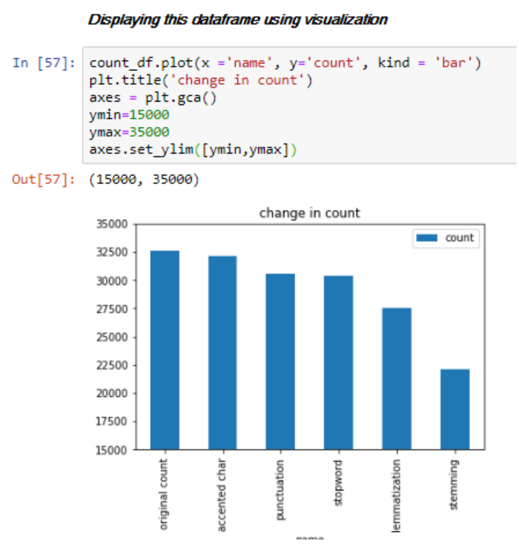
Figure 40:Defining a function and using it



```
In [56]: #Total count after each and everything

         count_df = pd.DataFrame(data,columns=['name','count'])
         count_df

Out[56]:
```

	name	count
0	original count	32570
1	accented char	32117
2	punctuation	30542
3	stopword	30399
4	lemmatization	27494
5	stemming	22088

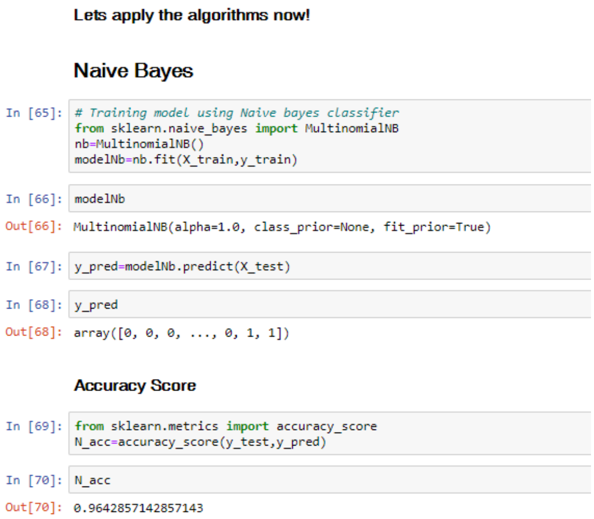
Figure 38: Total words after each step of preprocessing:



Displaying this dataframe using visualization

```
In [57]: count_df.plot(x ='name', y='count', kind = 'bar')
         plt.title('change in count')
         axes = plt.gca()
         ymin=15000
         ymax=35000
         axes.set_ylim([ymin,ymax])

Out[57]: (15000, 35000)
```

Figure 39: Viewing using visualization



Lets apply the algorithms now!

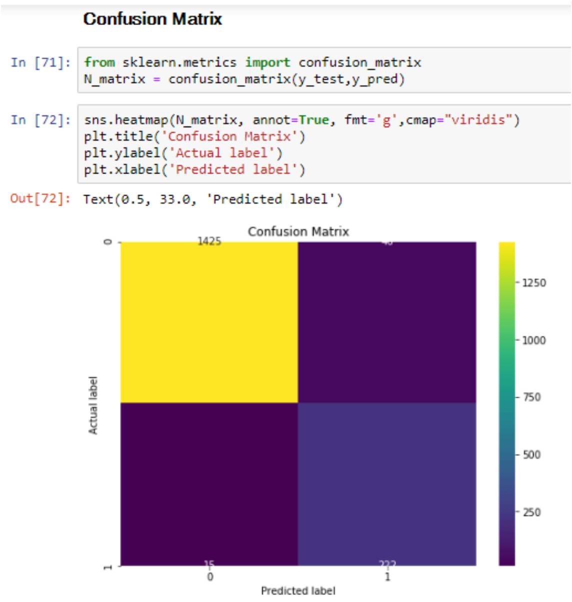
Naive Bayes

```
In [65]: # Training model using Naive bayes classifier
         from sklearn.naive_bayes import MultinomialNB
         nb=MultinomialNB()
         modelNb=nb.fit(X_train,y_train)

In [66]: modelNb

Out[66]: MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)

In [67]: y_pred=modelNb.predict(X_test)

In [68]: y_pred

Out[68]: array([0, 0, 0, ..., 0, 1, 1])
```

Accuracy Score

```
In [69]: from sklearn.metrics import accuracy_score
         N_acc=accuracy_score(y_test,y_pred)

In [70]: N_acc

Out[70]: 0.9642857142857143
```

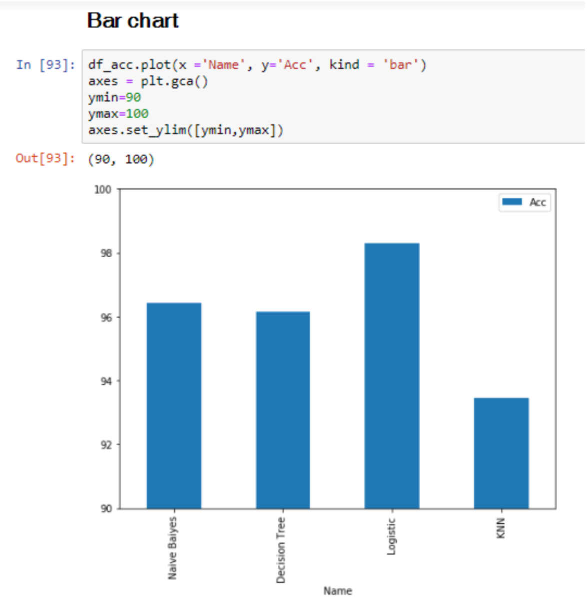
Figure 37: Applying the model to view the accuracy



Confusion Matrix

```
In [71]: from sklearn.metrics import confusion_matrix
         N_matrix = confusion_matrix(y_test,y_pred)

In [72]: sns.heatmap(N_matrix, annot=True, fmt='g',cmap="viridis")
         plt.title('Confusion Matrix')
         plt.ylabel('Actual label')
         plt.xlabel('Predicted label')

Out[72]: Text(0.5, 33.0, 'Predicted label')
```

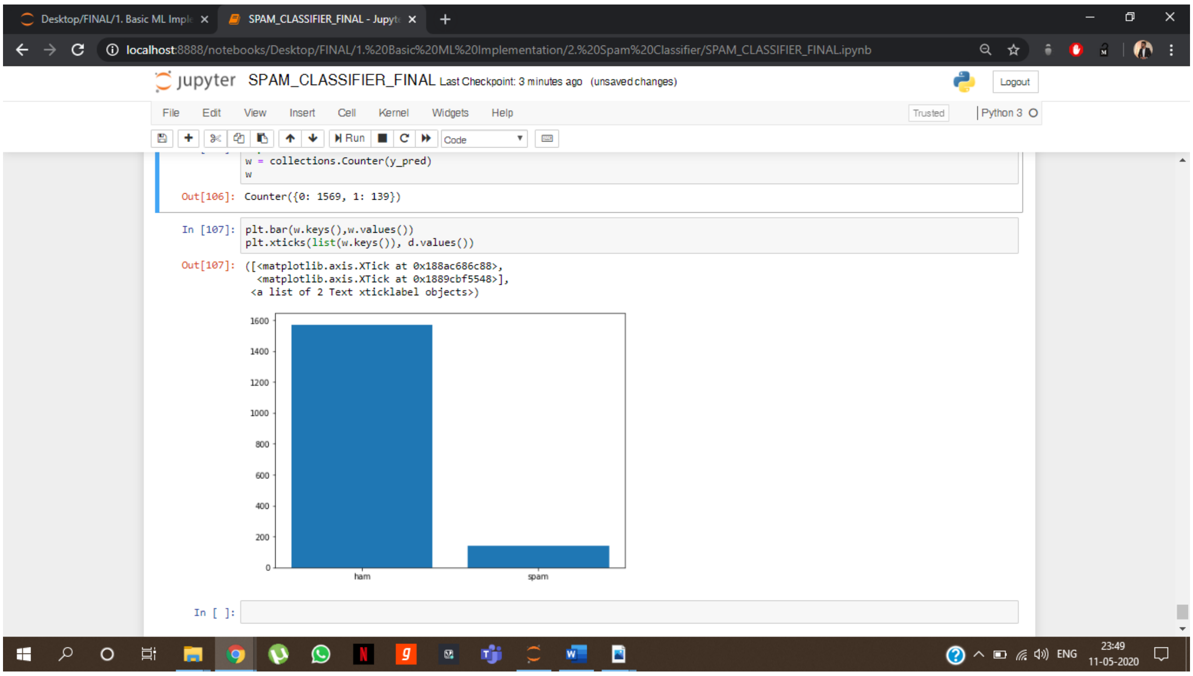
Figure 42: Confusion matrix for the above model



Bar chart

```
In [93]: df_acc.plot(x ='Name', y='Acc', kind = 'bar')
         axes = plt.gca()
         ymin=90
         ymax=100
         axes.set_ylim([ymin,ymax])

Out[93]: (90, 100)
```

Figure 41: Bar chart for models v/s Accuracy



Desktop/FINAL/1. Basic ML Impl
SPAM_CLASSIFIER_FINAL - Jupyt
localhost:8888/notebooks/Desktop/FINAL/1.%20Basic%20ML%20Implementation/2.%20Spam%20Classifier/SPAM_CLASSIFIER_FINAL.ipynb
jupyter SPAM_CLASSIFIER_FINAL Last Checkpoint: 3 minutes ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 ○
Run
Code

```
         w = collections.Counter(y_pred)
         w

Out[106]: Counter({0: 1569, 1: 139})

In [107]: plt.bar(w.keys(),w.values())
          plt.xticks(list(w.keys()), d.values())

Out[107]: ([<matplotlib.axis.XTick at 0x188ac686c88>,
            <matplotlib.axis.XTick at 0x1889cbf5548>],
           <a list of 2 Text xticklabel objects>)
```

1600
1400
1200
1000
800
600
400
200
0
ham
spam

```
In [ ]:
```

Figure 43: the most accurate model used on the test data.

EXTRA CERTIFICATIONS DONE

During the period of lockdown, to increase my understanding on various topics learnt during the training, I did some extra courses online which enhanced my knowledge and made me clearer on the thoughts and help me think in a better way while solving the problems.

- **MACHINE LEARNING COURSE – COURSERA (STANDORD ONLINE)**

This was the course that was provided by Stanford University in collaboration with Coursera and was taught by Andrew Ng, who is a great Data Scientist, Co-founder of Coursera and also a professor at Stanford University.

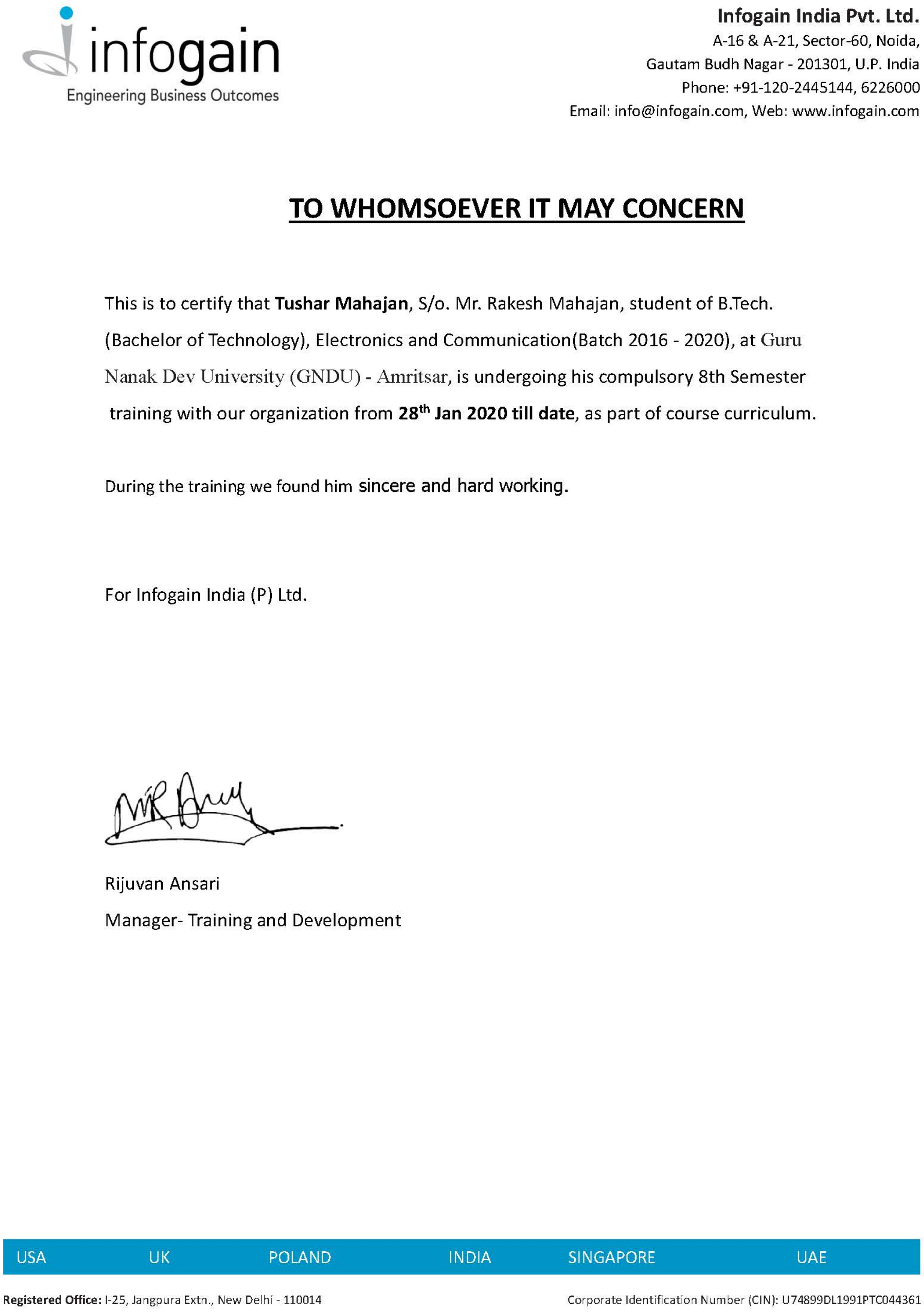
He taught all the things including from various problems in Machine Learning along with the mathematics behind it. It was a comprehensive as well as quite informative course.

- **DEEP LEARNING SPECIALIZATION – COURSERA (DEEPLEARNIG.AI)**

This was the course provided by John Hopkins University in collaboration with Coursera and was taught by few of the lecturers.

This was a 5-course certification which was starting initially with basic neural network, then there was hyperparameter tuning, then structuring the machine learning projects, then convolution neural network and then sequencing the models. All this has helped me in clearing the concepts in a better way

CERTIFICATES



Infogain India Pvt. Ltd.
A-16 & A-21, Sector-60, Noida,
Gautam Budh Nagar - 201301, U.P. India
Phone: +91-120-2445144, 6226000
Email: info@infogain.com, Web: www.infogain.com

infogain
Engineering Business Outcomes

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Tushar Mahajan**, S/o. Mr. Rakesh Mahajan, student of B.Tech. (Bachelor of Technology), Electronics and Communication(Batch 2016 - 2020), at Guru Nanak Dev University (GNDU) - Amritsar, is undergoing his compulsory 8th Semester training with our organization from **28th Jan 2020 till date**, as part of course curriculum.

During the training we found him sincere and hard working.

For Infogain India (P) Ltd.

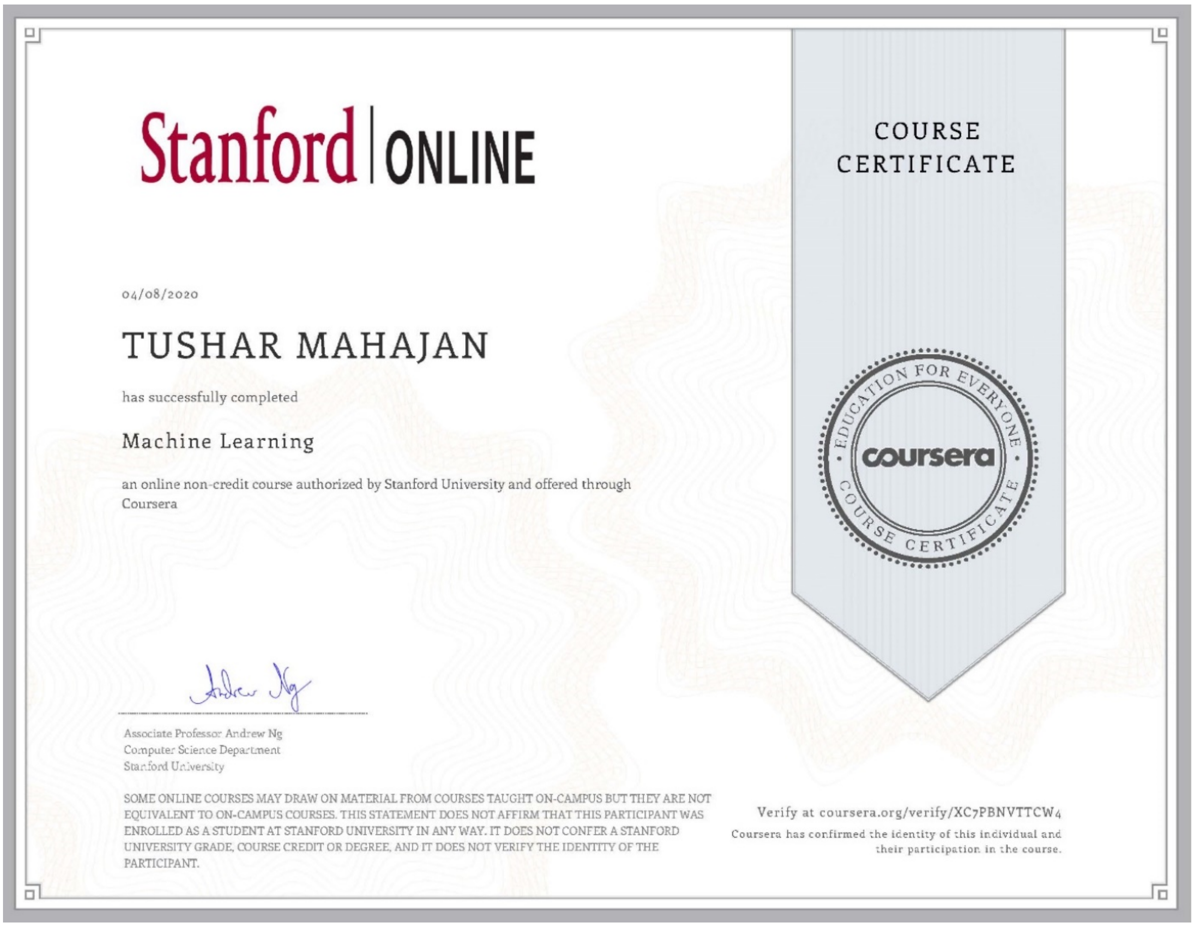
Rijuvan Ansari
Manager- Training and Development

USA UK POLAND INDIA SINGAPORE UAE

Registered Office: I-25, Jangpura Extn., New Delhi - 110014

Corporate Identification Number (CIN): U74899DL1991PTC044361

Figure 44: Training Certificate



Stanford | ONLINE

COURSE CERTIFICATE

04/08/2020

TUSHAR MAHAJAN

has successfully completed

Machine Learning

an online non-credit course authorized by Stanford University and offered through Coursera

EDUCATION FOR EVERYONE

coursera

COURSE CERTIFICATE

Andrew Ng

Associate Professor Andrew Ng
Computer Science Department
Stanford University

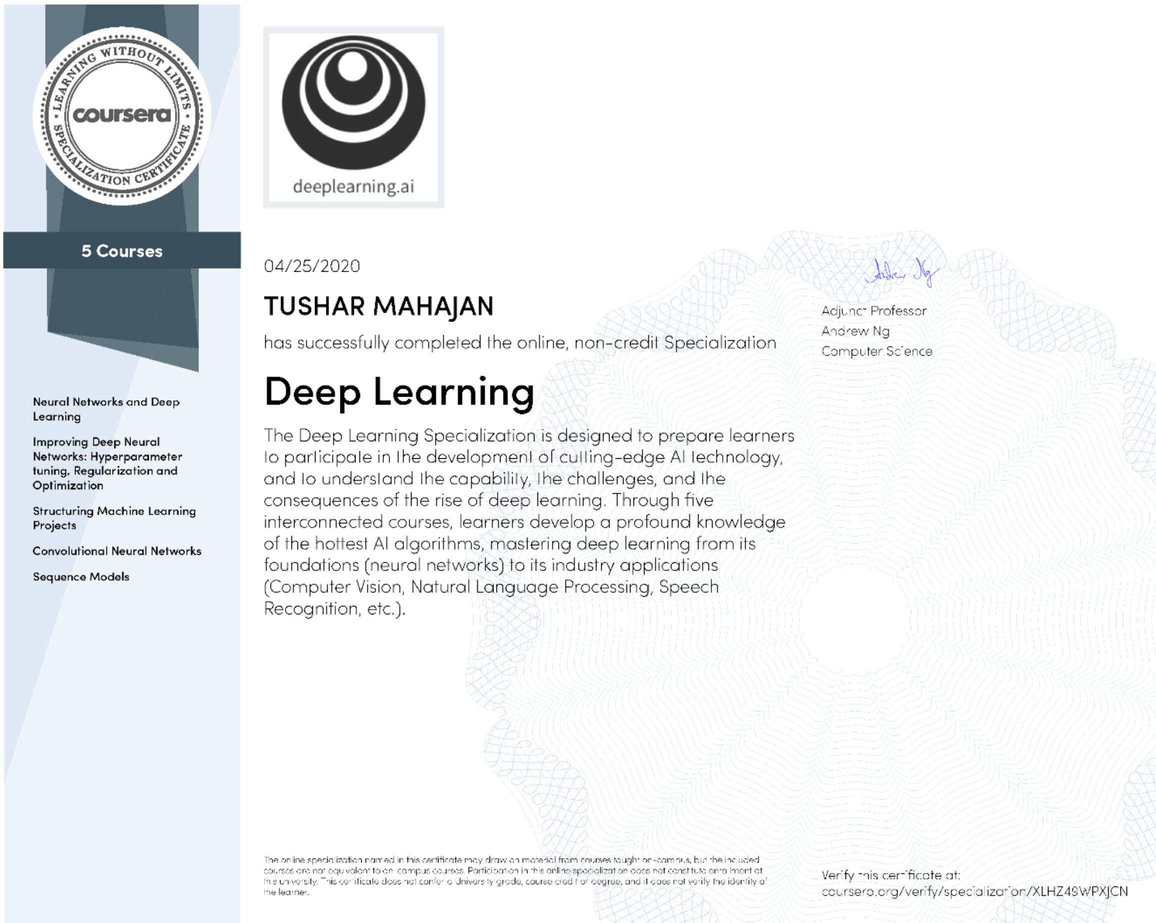
SOME ONLINE COURSES MAY DRAW ON MATERIAL FROM COURSES TAUGHT ON-CAMPUS BUT THEY ARE NOT EQUIVALENT TO ON-CAMPUS COURSES. THIS STATEMENT DOES NOT AFFIRM THAT THIS PARTICIPANT WAS ENROLLED AS A STUDENT AT STANFORD UNIVERSITY IN ANY WAY. IT DOES NOT CONFER A STANFORD UNIVERSITY GRADE, COURSE CREDIT OR DEGREE, AND IT DOES NOT VERIFY THE IDENTITY OF THE PARTICIPANT.

Verify at coursera.org/verify/XC7PBNVTTCW4

Coursera has confirmed the identity of this individual and their participation in the course.

Figure 45: Andrew Ng ML Coursera Certificate

Verify at coursera.org/verify/XC7PBNVTTCW4



LEARNING WITHOUT LIMITS
coursera
SPECIALIZATION CERTIFICATE
deeplearning.ai

5 Courses

Neural Networks and Deep Learning

Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization

Structuring Machine Learning Projects

Convolutional Neural Networks

Sequence Models

04/25/2020

TUSHAR MAHAJAN

has successfully completed the online, non-credit Specialization

Deep Learning

The Deep Learning Specialization is designed to prepare learners to participate in the development of cutting-edge AI technology, and to understand the capability, the challenges, and the consequences of the rise of deep learning. Through five interconnected courses, learners develop a profound knowledge of the hottest AI algorithms, mastering deep learning from its foundations (neural networks) to its industry applications (Computer Vision, Natural Language Processing, Speech Recognition, etc.).

Andrew Ng

Adjunct Professor
Andrew Ng
Computer Science

The online specialization named in this certificate may draw on material from courses taught on-campus, but the included courses are not equivalent to on-campus courses. Participation in this online specialization does not constitute enrollment at this university. This certificate does not confer a University grade, course credit or degree, and it does not verify the identity of the learner.

Verify this certificate at:
coursera.org/verify/specialization/XLHZ49WPXJCN

Figure 46: Deep Learning Specialization Certificate

Verify this certificate at: coursera.org/verify/specialization/XLHZ49WPXJCN

FUTURE ASPECTS

All the models that were used performed well but one was outshining this dataset. The model i.e. Logistic Regression is the best among the ones used on this dataset. The data right now is not that big and hence many algorithms perform very well if they are trained on large dataset and hence the future aspects can be the ones that has the dataset can be increased and hyperparameter optimization should be used so that the ones having very low accuracy should also reach good score. This will eventually make our spam classifier a big success and also our algorithm will be highly optimized after that.

Till now, I have had a great time learning these things and will not settle for less and will continue to learn and work more so that I can get better projects to work upon and shape my future in a better way and also help the organization to grow and excel.

CONCLUSION

Spam mails are a serious concern to and a major annoyance for many Internet users. The mode proposed as a solution in this paper is highly beneficial because it introduces a threshold counter which helps overcome congestion on the web server and also maintain the spam filter efficiency but at the same time, it also requires overhead storage space for the databases.

Since NLP is a relatively underdeveloped area for research, further enhancements can be made in the field of spam detection for online security using Natural Language Processing in future. This will surely help our future generations and also our generation to generate better models and hence aim for a better future with highly optimized machine learning algorithms that can help and ease the load of humans.

REFERENCES

- https://www.w3schools.com/
- https://www.tensorflow.org/
- https://www.infogain.com/
- https://www.wikipedia.org/
- https://towardsdatascience.com/
- https://www.coursera.org/
- https://www.slideshare.net/
- http://neuralnetworksanddeeplearning.com/
- http://dataversity.net/
- https://jupyter.org/
- https://www.tutorialspoint.com/
- https://www.python.org/
- https://www.youtube.com/