**PROJECT-II REPORT**

**On**

**Detection of violence on Social Media**

Submitted to Rajasthan Technical University

In partial fulfillment of the requirement for the award of the degree of

**B.TECH.**

**In**

**COMPUTER ENGINEERING**

**Submitted By**

**Mukul Kadel (PIET15CE066)**

**Aaditya Pareek (PIET15CE001)**

**Akshat Rajpurohit (PIET15CE012)**

**Under the Guidance of**

**Mr. Puneet Mathur**

at



**POORNIMA INSTITUTE OF ENGINEERING & TECHNOLOGY, JAIPUR**

**Rajasthan Technical University, KOTA**

**APRIL, 2018**

**CERTIFICATE**

This is to be certified that the project entitled “**Detection of violence on Social Media**” has been submitted for the Bachelor of Computer Science and Engineering, Poornima Institute Of Engineering & Technology, Jaipur during the academic year 2018-2019 is a bonafide piece of project work carried out by “ **Mukul Kadel, Aaditya Pareek & Akshat Rajpurohit**” towards the partial fulfillment for the award of the Degree (B.Tech.) under the guidance of “**Mr. Puneet Mathur**” and supervision and no part of thereof has been submitted by them for any degree or diploma.

Project Guide Project Coordinator Mr. Deepak Moud

Mr. Puneet Mathur Prof. (Dr.) Praveen Gupta (H.O.DCSE)

(Assistant Professor) (Professor)

**CANDIDATE’S DECLARATION**

We, **Mukul Kadel** **(PIET15CE066), Aaditya Pareek(PIET15CE001) & Akshat Rajpurohit(PIET15CE012)** B.Tech (Semester- VIII) of “**Poornima Institute Of Engineering & Technology, Jaipur”**

hereby declare that the Project Report entitled **“Detection of violence on Social Media”** is an original work and data provided in the study is authentic to the best of our knowledge.This report has not been submitted to any other Institute for the award of any other degree.

|  |  |  |
| --- | --- | --- |
| **Mukul kadel** | **Aaditya pareek** | **akshat rajpurohit** |
| **(piet15ce066)** | **(piet15ce001)** | **(piet15ce012)** |

|  |  |
| --- | --- |
| **Place: PIET, JAIPUR** |  |
| **Date:** |  |

**ACKNOWLEDGEMENT**

It is our pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced our thinking, behavior and acts during the course of study.

We express our sincere gratitude to ***Dr. O. P. Sharma,*** Director, PIET for providing us an opportunity to undergo this Major Project as the part of the curriculum.

We are thankful to ***Mr. Deepak Moud, HOD, CS*** for his support, cooperation, and motivation provided to us during the training for constant inspiration, presence and blessings.

We are thankful to ***Mr. Puneet Mathur*** for his support, cooperation, and motivation provided to us during the training for constant inspiration, presence and blessings.

We also extend our sincere appreciation to ***Prof. (Dr.) Praveen Gupta*** who provided his valuable suggestions and precious time in accomplishing our Project report.

Lastly, we would like to thank the almighty and our parents for their moral support and friends with whom we shared our day-to-day experience and received lots of suggestions that improved our quality of work.

|  |  |  |
| --- | --- | --- |
| **Mukul kadel** | **Aaditya pareek** | **akshat rajpurohit** |
| **(piet15ce066)** | **(piet15ce001)** | **(piet15ce012)** |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TOPICS** | **PAGE NO.** |
|  | TITLE PAGE | I |
|  | CERTIFICATE | II |
|  | CANDIDATE’S DECLARATION | III |
|  | ACKNOWLEDGEMENT | IV |
|  | TABLE OF CONTENTS | V |
|  | TABLE OF FIGURE | VI |
|  | ABSTRACT | VII |
| 1 | **INTRODUCTION TO PROJECT** | 1 |
|  | Project Aim and Objective  Problem Statement  Background of the Project (Literature Survey)  Software Requirements  Hardware Requirements |  |
| 2 | **PRODUCT BACKLOG** |  |
|  | 1. PRODUCT Backlog 2. Sprint Backlog-1 3. Sprint Backlog-2 4. Sprint Backlog-3 5. Sprint Backlog-4 |  |
|  |  |  |
| 3 | **TECHNOLOGY APPLIED AND PROJECT MANAGEMENT** |  |
|  | Brief Description of All technology Apply in the Project.  Project management  Agile  Relevance to Society  Ethics  Life Long Learning  Project Finance  Environment and Sustainability |  |
|  |  |  |
|  |  |  |
| 4 | **PROJECT IMPLEMENTATION** |  |
|  | Sprint Backlog-1  Sprint Backlog-2  Sprint Backlog-3  Sprint Backlog-4 |  |
|  |  |  |
| 5 | **CONCLUSION** |  |
|  | Results  Conclusion  Future Scope |  |
| 6 | ANNEXURES |  |
|  | References |  |
|  | APPENDIX/ANNEXURES |  |
|  | Research Paper ( if Presented and approved for publication) |  |
|  | CV |  |
|  |  |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **S. NO.** | **FIGURE** | **PAGE NO.** |
| 3.1 | Linear Regression of Machine learning | 28 |
| 3.2 | Handy machine learning Algorithm | 29 |
| 3.3 | Linear Regression learning model | 31 |
| 3.4 | Height vs Weight Linear Regression | 34 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **S. NO.** | **TABLES** | **PAGE NO.** |
| 1. |  |  |
| 2. |  |  |
| 3. |  |  |
| 4. |  |  |
| 5. |  |  |
| 6. |  |  |
| 7. |  |  |
| 8. |  |  |

**ABSTRACT**

Analysis of public information from social media could yield interesting results and insights into the world of public opinions about almost any product, service or personality. Social network data is one of the most effective and accurate indicators of public sentiment. The explosion of Web 2.0 has led to increased activity in Podcasting, Blogging, Tagging, Contributing to RSS, Social Bookmarking, and Social Networking. As a result there has been an eruption of interest in people to mine these vast resources of data for opinions. Sentiment Analysis or Opinion Mining is the computational treatment of opinions, sentiments and subjectivity of text. In this project we will be discussing a methodology which allows utilization and interpretation of twitter data to determine violence over social media. Developing a program for violent comment analysis is an approach to be used to computationally measure customer's perceptions. This project reports on the detection of a violence on social media analysis, extracting and training a vast amount of tweets. Results classify customer's perspective via tweets into positive and negative, which is represented in a pie chart, bar diagram, scatter plot.

**KEYWORDS:** Data mining, Natural language processing, Sentimental Analysis, Naive Bayes

**CHAPTER 1**

**INTRODUCTION TO PROJECT**

Sentiment is an attitude, thought, or judgment prompted by feeling which is also known as opinion mining, studies people’s sentiments towards certain entities. Internet is a resourceful place with respect to sentiment information. From a user’s perspective, people are able to post their own content through various social media, such as forums, micro-blogs, or online social networking sites. From a researcher’s perspective, many social media sites release their application programming interfaces (APIs), prompting data collection and analysis by researchers and developers. For instance, Twitter currently has three different versions of APIs available, namely the REST API, the Search API, and the Streaming API. With the REST API, developers are able to gather status data and user information; the Search API allows developers to query specific Twitter content, whereas the Streaming API is able to collect Twitter content in real time. Moreover, developers can mix those APIs to create their own applications. Hence, Detection of violence over twitter seems to be having a strong fundament with the support of massive online data. However, those types of online data have several flaws that potentially hinder the process of sentiments. The first flaw is that since people can freely post their own content, the quality of their opinions cannot be guaranteed. For example, instead of sharing topic-related opinions, online spammers post spam on forums. Some spam are meaningless at all, while others have irrelevant opinions also known as fake opinions. The second flaw is that ground truth of such online data is not always available. A ground truth is more like a tag of a certain opinion, indicating whether the opinion is positive, negative, or neutral. The Stanford Sentiment 140 Tweet Corpus is one of the datasets that has ground truth and is also public available. The corpus contains 1.6 million machine-tagged Twitter messages. Micro blogging websites have evolved to become a source of varied kind of information. This is due to nature of micro blogs on which people post real time messages about their opinions on a variety of topics, discuss current issues, 2 complain, and express positive sentiment for products they use in daily life. In fact, companies manufacturing such products have started to poll these micro blogs to get a sense of general sentiment for their product. Many time these companies study user reactions and reply to users on micro blogs. One challenge is to build technology to detect and summarize an overall sentiment. Our project Tweezers resembles the analyze of tweets by the peoples on certain products of companies or brands or performed by political leaders. In order to do this we analyzed tweets from Twitter. Tweets are a reliable source of information mainly because people tweet about anything and everything they do including buying new products and reviewing them. Besides, all tweets contain hash tags which make identifying relevant tweets a simple task. A number of research works has already been done on twitter data. Most of which mainly demonstrates how useful this information is to predict various outcomes. Our current research deals with outcome prediction and explores localized outcomes. We collected data using the Twitter public API which allows developers to extract tweets from twitter programmatically. The collected data, because of the random and casual nature of tweeting, need to be filtered to remove unnecessary information. Filtering out these and other problematic tweets such as redundant ones, and ones with no proper sentences was done next. As the preprocessing phase was done in certain extent it was possible to guarantee that analyzing these filtered tweets will give reliable results. Twitter does not provide the gender as a query parameter so it is not possible to obtain the gender of a user from his or her tweets. It turned out that twitter does not ask for user gender while opening an account so that information is seemingly unavailable.

* 1. Statement of the problem

The problem at hand consists of two subtasks:

**Phrase Level Content Analysis in Twitter :**

• Given a message containing a marked instance of a word or a phrase, determine whether that instance is positive, negative or neutral in that context.

**Sentence Level Content Analysis in Twitter:**

• Given a message, decide whether the message is of positive, negative, or neutral sentiment. For messages conveying both a positive and negative sentiment, whichever is the stronger sentiment should be chosen.

* 1. Aim & Objectives

The objectives of this project are:

* To implement an algorithm for automatic classification of text into positive and negative
* Content Analysis to determine the attitude of the mass is positive, negative or neutral towards the subject of interest
* Graphical representation of the sentiment in form of Pie-Chart, Bar Diagram and Scatter Plot.
  1. Scope of project

This project will be helpful to the companies, political parties as well as to the common people. It will be helpful to political party for reviewing about the program that they are going to do or the program that they have performed. Similarly companies also can get review about their new product on newly released hardwares or softwares. Also the movie maker can take review on the currently running movie. By analyzing the tweets analyzer can get result on how positive or negative or neutral are peoples about it.

2. LITERATURE REVIEW

Sentiment analysis has been handled as a Natural Language Processing task at many levels of granularity. Starting from being a document level classification task ( Turney, 2002; Pang and Lee, 2004) , it has been handled at the sentence level (Hu and Liu [2], 2004; Kim and Hovy, 2004) and more recently at the phrase level (Wilson et al., 2005; Agarwal et al., 2009). Microblog data like Twitter, on which users post real time reactions to and opinions about “everything”, poses newer and different challenges. Some of the early and recent results on sentiment analysis of Twitter data are by Go et al. (2009), ( Bermingham and Smeaton, 2010) and Pak and Paroubek (2010). Go et al. (2009) use distant learning to acquire sentiment data. They use tweet sending in positive emotions like “:)” “:-)” as positive and negative emoticons like “:(” “:-(” as negative. They build models using Naive Bayes, Max Ent and Support Vector Machines (SVM), and they report SVM outperforms other classifiers. In terms of feature space, they try a Unigram, Bigram model in conjunction with parts-of-speech (POS) features. They note that the unigram model outperforms all other models. Specifically, bigrams and POS features do not help. Pak and Paroubek (2010) [3] collect data following a similar distant learning paradigm. They perform a different classification task though: subjective versus objective. For subjective data they collect the tweets ending with emoticons in the same manner as Go et al. (2009). For objective data they crawl twitter accounts of popular newspapers like “New York Times”, “Washington Posts” etc. They report that POS and bigrams both help (contrary to results presented by Go et al. (2009)). Both these approaches, however, are primarily based on ngram models. Moreover, the data they use for training and testing is collected by search queries and is therefore biased. In contrast, we present features that achieve a significant gain over a unigram baseline. In addition we explore a different method of data representation and report significant improvement over the unigram models. Another contribution of this paper is that we report results on manually annotated data that does not suffer from any known biases. Our data will be a random sample of streaming tweets unlike data collected by using specific queries. The size of our hand-labeled data 6 will allow us to perform cross validation experiments and check forth variance in performance of the classifier across folds. Another significant effort for sentiment classification on Twitter data is by Barbosa and Feng (2010). They use polarity predictions from three websites as noisy labels to train a model and use 1000 manually labeled tweets for tuning and another 1000 manually labeled tweets for testing. They however do not mention how they collect their test data. They propose the use of syntax features of tweets like retweet, hashtags, link, punctuation and exclamation marks in conjunction with features like prior polarity of words and POS of words. We extend their approach by using real valued prior polarity, and by combining prior polarity with POS. Our results show that the features that enhance the performance of our classifiers the most are features that combine prior polarity of words with their parts of speech. The tweet syntax features help but only marginally. Gamon (2004) perform sentiment analysis on feeadback data from Global Support Services survey. One aim of their paper is to analyze the role of linguistic features like POS tags. They perform extensive feature analysis and feature selection and demonstrate that abstract linguistic analysis features contributes to the classifier accuracy. In this paper we perform extensive feature analysis and show that the use of only 100 abstract linguistic features performs as well as a hard unigram baseline.

3. Software Requirements

In order to create a simple personal AI assistant, some programming abilities are required. In particular, such languages as Lisp, Java, Prolog and Python (being the most popular in this regard) are used for the creation of AI-based apps.

Python is used as a base for the most renowned AI-based software because of its flexibility, simplicity and longstanding reputation (it has been in existence for over 20 years).

To successfully develop a virtual assistant, even an experienced Python developer would need to advance the level of qualification from time to time, so topical literature will come in handy. We can recommend several useful tools to make the stages of AIA creation easier. Familiarize yourself with such libraries and tools as NumPy, Matplotlib, Pandas, Scikit, Theano, AIMA, pyDatalog, SimpleAI, EasyAi, PyBrain, MDP, scikit, PyML and others.

4.Hardware Requirements

As a general guide, these are some of the technical requirements that may be expected for a home-based call center. Each company's specific requirements will differ.

* **Desktop PC.** Laptop PCs or Macintosh computers are sometimes not allowed. Some *minimum*requirements for computers often are:
  + 1Ghz-2Ghz processor
  + Windows operating system - Which version varies but keep in mind that newer isn't always better; some companies are slow to adapt to changes.
  + 1GB of RAM
  + Sound card, speakers
  + 15" to 17" monitor
  + Virus and spyware protection software and a working firewall.
  + Software programs that may be required include Microsoft Office (Microsoft Word and Excel), and/or Adobe Acrobat reader.
* **Broadband internet connection.** DSL and cable are usually allowed but satellite, dial-up and wireless Internet connections are usually not. A wireless network inside the home is sometimes allowed, but many companies will require that computers are connected directly to the internet.
* **Landline phone service.** Cell, VOIP (i.e Vonage) and cable phone lines are not often not acceptable, though cable is becoming more common. However, there are some companies now who don't require any phone line because the connection is all done through the internet. Of the companies that do require a landline, many require it be a dedicated phone line separate from your home phone. Calling features on the phone line, such as call waiting, call blocking and voice mail, are often not allowed or must be disabled.
* **Corded telephone (with buttons not on the handset)**if a phone line is used.
* **Corded (not wireless) telephone headset with a noise-canceling microphone.**
* **Skype or another teleconferencing service.**Occasionally this is required but it is free.
* **Email account.** A few companies require specific provider.
* **Web browser.**

**CHAPTER 2**

**PRODUCT BACKLOG**

1. **PRODUCT Backlog**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **SPRINT BACKLOG** | **US ID** | **BACKLOG ITEM** | | | **PRIORITY** | | **AS A/AN** | **I WANT TO** | **SO THAT** | | 1 | US1/N1 | Developer | Understand the problem | I can search for relevant solutions | 1 | | 1 | US1/N2 | Developer | Understand user stories | I can search for dataset accordingly | 1 | | 1 | US1/N3 | Developer | Download module related to project | I can use them on project | 1 | | 1 | US1/N4 | Developer | Get authorization keys | I can fetch data for prediction | 1 | | 1 | US1/N5 | Developer | Create twitter api script | I can fetch real time data from twitter | 1 | | 1 | US1/N6 | Developer | Find website for dataset | I can use it | 1 | | 1 | US1/N7 | Developer | Download datasets from various sources | I can train model on it | 1 | | 1 | US1/N8 | Developer | Create list for positive and negative words | Find labels for dataset | 1 | | 1 | US1/N9 | Developer | Normalize and stem my data | I can simplify my dataset | 1 | | 1 | US1/N10 | Developer | Dictionary for my words |  | 1 | | 1 | US1/N11 | Developer | Do POS tagging on dataset | I can use them as features | 1 | | 1 | US1/N12 | Developer | Vectorize my processed data | I can train model on it | 1 | | 2 | US1/N13 | Developer | Train my data on naïve bayes model | I can test it's accuracy and performance | 2 | | 2 | US1/N14 | Developer | Train my data on logistic regression model | I can test it's accuracy and performance | 2 | | 2 | US1/N15 | Developer | Train my data on LDA model | I can test it's accuracy and performance | 2 | | 2 | US1/N16 | Developer | Train my data on Decision tree model | I can test it's accuracy and performance | 2 | | 2 | US1/N17 | Developer | Train my data on SVM model | I can test it's accuracy and performance | 2 | | 2 | US1/N18 | Developer | Train my data on random forest model | I can test it's accuracy and performance | 2 | | 2 | US1/N19 | Developer | Train my data on multi-layer perceptron | I can test it's accuracy and performance | 2 | | 2 | US1/N20 | Developer | Train my data on recurrent neural network | I can test it's accuracy and performance | 2 | | 2 | US1/N21 | Developer | Train my data on convolutional neural network | I can test it's accuracy and performance | 2 | | 3 | US1/N22 | Developer | Calculate accuracy, recall, fvalue and other performace measures | I can analyze and compare vairous models | 3 | | 3 | US1/N23 | Developer | Test model on different tradeoff values | I can select best model | 3 | | 3 | US1/N24 | Developer | Apply my trained model on user stories |  | 3 | | 3 | US1/N25 | Developer | Create GUI for the project | User can use system effortlessly | 3 | | 3 | US2/N1 | Standard User | Analyse the type of content by the people I follow | I can choose whether to still follow them without going through there feeds | 3 | | 3 | US2/N2 | Standard User | Identify hateful content from comments | I Can identify weather to reply it or not | 3 | | 3 | US2/N3 | Standard User | Check my content before posting it | I can analyse how it will affect the audience | 3 | | 4 | US3/N1 | Content Creater | Check the hatefull content form hashtags | I can see the reaction of public on the perticular topic(Hashtag) | 3 | | 4 | US3/N2 | Content Creater | Check for public openion on tweets | I can understand public reaction | 3 | | 4 | US3/N3 | Content Creater | See views of verified accounts | I can usderstand openien of celibrities | 3 | | 4 | US3/N4 | Content Creater | See supporters and constant haters | I can target my audience accordingly | 3 | | 4 | US4/N1 | Reviewer | Report undetected voilent content | User experience and project efficiency can be increased | 3 | | 4 | US1/N26 | Developer | Perform testing on project for vulnerabilities | we can prevent any data leaks | 4 | |

1. **Sprint Backlog-1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **US ID** | **USER STORY** | **TASK ID** | **TASKS** | **TM** | **STATUS (NOT STARTED / IN PROGRESS / COMPLETED)** |
| SB1/US1 | Understand the problem | SB1/D1/T1 | Breaking the problem and understanding each component | MK+AP+AR | Completed |
| SB1/D1/T2 | Searching for relavent solutions | MK+AP+AR | Completed |
| SB1/D1/T3 | Searching alternative approches | MK+AP+AR | Completed |
| SB1/D1/T4 | Analysing similar projects | MK+AP+AR | Completed |
| SB1/US2 | Understand user stories | SB1/D2/T1 | Looking for various use cases | MK+AP+AR | Completed |
| SB1/D2/T2 | Discussion about user stories | MK+AP+AR | Completed |
| SB1/D2/T3 | Searching for type of data required | MK+AP+AR | Completed |
| SB1/US3 | Download module related to project | SB1/D3/T1 | Finding modules required | MK+AP+AR | Completed |
| SB1/D3/T2 | Downloading and Installing the modules | MK+AP+AR | Completed |
| SB1/US6 | Get authorization keys | SB1/D4/T1 | Registering application on social media developer portal | MK+AP+AR | Completed |
| SB1/US7 | Create twitter api script | SB1/D5/T1 | Creating basic twitter script | MK+AP+AR | Completed |
| SB1/D5/T2 | Adding additional features and filters in the script | MK+AP+AR | Completed |
| SB1/US27 | Find website for dataset | SB1/D6/T1 | Searching for reliable data sources | MK+AP+AR | Completed |
| SB1/US27 | Download datasets from various sources | SB1/D7/T2 | Downloading datasets from github | MK | Completed |
| SB1/D7/T3 | Downloading datasets from kaggle | AP | Completed |
| SB1/D7/T4 | Downloading datasets from popular dataset sites | AR | Completed |
| SB1/D7/T5 | Downloading datasets from other sources | MK+AP+AR | Completed |
| SB1/US150 | Create list for positive and negative words | SB1/D8/T1 | Categorizing words in dataset | MK+AP+AR | Completed |
| SB1/D8/T2 | Downloading lists from internet sources | MK+AP+AR | Completed |
| SB1/US28 | Normalize and stem my data | SB1/D9/T1 | Applying normalization and stemming on data | MK+AP+AR | Completed |
| SB1/US50 | Dictionary for my words | SB1/D10/T1 | Creating list of words with their synonyms | MK+AP+AR | Completed |
| SB1/US53 | Do POS tagging on dataset | SB1/D11/T2 | Studing POS tagging | MK+AP+AR |  |
| SB1/D11/T3 | Applying POS on data in distributed form | MK+AP+AR |  |
| SB1/US55 | Vectorize my processed data | SB1/D12/T1 | Studing Vectorizing methods | MK+AP+AR |  |
| SB1/D12/T2 | Applying and comparing various vectorizing methods | MK+AP+AR | Completed |
| SB1/D12/T3 | Selecting best method | MK+AP+AR | Completed |

1. **Sprint Backlog-2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **US ID** | **USER STORY** | **TASK ID** | **TASKS** | **TM** | **STATUS (NOT STARTED / IN PROGRESS / COMPLETED)** |
| SB2/US1 | Train my data on naïve bayes model | SB2/D1/T1 | Studing naïve bayes model | MK+AP+AR | Completed |
| SB2/D1/T2 | Implementing naïve bayes model | MK+AP+AR | Completed |
| SB2/D1/T3 | Implementing naïve bayes model on project data | MK+AP+AR | Completed |
| SB2/US2 | Train my data on logistic regression model | SB2/D2/T1 | Studing logistic regression model | MK+AP+AR |  |
| SB2/D2/T2 | Implementing logistic regression model | MK+AP+AR | Completed |
| SB2/D2/T3 | Implementing logistic regression model on project data | MK+AP+AR | Completed |
| SB2/US3 | Train my data on LDA model | SB2/D3/T1 | Studing LDA model | MK+AP+AR | Completed |
| SB2/D3/T2 | Implementing LDA model | MK+AP+AR | Completed |
| SB2/D3/T3 | Implementing LDA model on project data | MK+AP+AR | Completed |
| SB2/US4 | Train my data on Decision tree model | SB2/D4/T1 | Studing decision tree model | MK+AP+AR | Completed |
| SB2/D4/T2 | Implementing decision tree model | MK+AP+AR | Completed |
| SB2/D4/T3 | Implementing decision tree model on project data | MK+AP+AR | Completed |
| SB2/US5 | Train my data on SVM model | SB2/D5/T1 | Studing SVM model | MK+AP+AR | Completed |
| SB2/D5/T2 | Implementing SVM model | MK+AP+AR | Completed |
| SB2/D5/T3 | Implementing SVM model on project data | MK+AP+AR | Completed |
| SB2/US6 | Train my data on random forest model | SB2/D6/T1 | Studing random forest model | MK+AP+AR | Completed |
|  |  | SB2/D6/T2 | Implementing random forest model | MK+AP+AR | Completed |
| SB2/D6/T3 | Implementing random forest model on project data | MK+AP+AR | Completed |
| SB2/US7 | Train my data on multi-layer perceptron | SB2/D7/T1 | Studing multi-level perceptron | MK+AP+AR | Completed |
| SB2/D7/T2 | Implementing multi-level perceptron | MK+AP+AR | Completed |
| SB2/D7/T3 | Implementing multi-level perceptron on project data | MK+AP+AR | Completed |
| SB2/US8 | Train my data on recurrent neural network | SB2/D8/T1 | Studing recurrent neural network | MK+AP+AR |  |
| SB2/D8/T2 | Implementing recurrent neural network | MK+AP+AR |  |
| SB2/D8/T3 | Implementing recurrent neural network on project data | MK+AP+AR |  |
| SB2/US9 | Train my data on convolutional neural network | SB2/D8/T1 | Studing convolutional neural network | MK+AP+AR |  |
| SB2/D8/T2 | Implementing convolutional neural network | MK+AP+AR |  |

1. **Sprint Backlog-3**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **US ID** | **USER STORY** | **TASK ID** | **TASKS** | **TM** | **STATUS (NOT STARTED / IN PROGRESS / COMPLETED)** |
| SB3/US1 | Calculate accuracy, recall, fvalue and other performace measures | SB3/D1/T1 | Calculating accuracy, recall and fvalue | MK+AP+AR |  |
| SB3/D1/T2 | Searching for other performance measures and calculating them | MK+AP+AR |  |
| SB3/US2 | Test model on different tradeoff values | SB3/D2/T1 | Testing models with different parameters | MK+AP+AR |  |
| SB3/D2/T2 | Testing Neural networks | MK |  |
| SB3/D2/T3 | Testing SVM, decision tree and random forest models | AP |  |
| SB3/D2/T4 | Testing Logistic regression, LDA and naïve bayes models | AR |  |
| SB3/US3 | Apply my trained model on user stories | SB3/D3/T1 | Looking for various ways to apply trained model | MK+AP+AR |  |
| SB3/US4 | Create GUI for the project | SB3/D4/T1 | Designing GUI for project | MK+AP+AR |  |
| SB3/D4/T2 | Finalizing GUI for project | MK+AP+AR |  |
| SB3/D4/T3 | Creating basic structure of GUI | MK+AP+AR |  |
| SB3/US5 | Analyse the type of content by the people I follow | SB3/D5/T1 | Creating modifying twitter script to fetch this data | MK |  |
| SB3/D5/T2 | Creating GUI for user | AP |  |
| SB3/D5/T3 | Joining script with GUI | AR |  |
| SB3/US6 | Identify hateful content from comments | SB3/D6/T1 | Creating modifying twitter script to fetch this data | MK |  |
| SB3/D6/T2 | Creating GUI for user | AP |  |
| SB3/D6/T3 | Joining script with GUI | AR |  |
| SB3/US7 | Check my content before posting it | SB3/D7/T1 | Creating modifying twitter script to fetch this data | MK |  |
| SB3/D7/T2 | Creating GUI for user | AP |  |

1. **Sprint Backlog-4**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **US ID** | **USER STORY** | **TASK ID** | **TASKS** | **TM** | **STATUS (NOT STARTED / IN PROGRESS / COMPLETED)** |
| 1 | Check the hatefull content form hashtags | S4/D1/T1 | Creating modifying twitter script to fetch this data | MK |  |
| S4/D1/T2 | Creating GUI for user | AP |  |
| S4/D1/T3 | Joining script with GUI | AR |  |
| 2 | Check for public openion on tweets | S4/D2/T1 | Creating modifying twitter script to fetch this data | MK |  |
| S4/D2/T2 | Creating GUI for user | AP |  |
| S4/D2/T3 | Joining script with GUI | AR |  |
| 3 | See views of verified accounts | S4/D3/T1 | Creating modifying twitter script to fetch this data | MK |  |
| S4/D3/T2 | Creating GUI for user | AP |  |
| S4/D3/T3 | Joining script with GUI | AR |  |
| 4 | See supporters and constant haters | S4/D4/T1 | Creating modifying twitter script to fetch this data | MK |  |
| S4/D4/T2 | Creating GUI for user | AP |  |
| S4/D4/T3 | Joining script with GUI | AR |  |
| 5 | Report undetected voilent content | S4/D5/T1 | Creating GUI for user | AR |  |
| S4/D5/T2 | Creating admin panel for reviewing reports | AP |  |
| S4/D5/T3 | Creating script to monitor reported content | MK |  |
| 6 | Perform testing on project for vulnerabilities | S4/D6/T1 | Testing various aspects of website | MK+AP+AR |  |
| S4/D6/T2 | Creating test cases | MK |  |

**CHAPTER 3**

**TECHNOLOGY APPLIED AND PROJECT MANAGEMENT**

**Supervised Machine Learning**

Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output.

Y = f(X)

The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data.

It is called supervised learning because the process of an algorithm learning from the training dataset can be thought of as a teacher supervising the learning process. We know the correct answers, the algorithm iteratively makes predictions on the training data and is corrected by the teacher. Learning stops when the algorithm achieves an acceptable level of performance.

Supervised learning problems can be further grouped into regression and classification problems.

* **Classification**: A classification problem is when the output variable is a category, such as “red” or “blue” or “disease” and “no disease”.
* **Regression**: A regression problem is when the output variable is a real value, such as “dollars” or “weight”.

Some common types of problems built on top of classification and regression include recommendation and time series prediction respectively.

Some popular examples of supervised machine learning algorithms are:

* Linear regression for regression problems.
* Random forest for classification and regression problems.
* Support vector machines for classification problems.

**Naive Bayes for Machine Learning**

Naive Bayes is a simple but surprisingly powerful algorithm for predictive modeling.

In this post you will discover the Naive Bayes algorithm for classification. After reading this post, you will know:

The representation used by naive Bayes that is actually stored when a model is written to a file.

How a learned model can be used to make predictions.

How you can learn a naive Bayes model from training data.

How to best prepare your data for the naive Bayes algorithm.

Where to go for more information on naive Bayes.

This post is written for developers and does not assume any background in statistics or probability, although knowing a little probability wouldn’t hurt.

In machine learning we are often interested in selecting the best hypothesis (h) given data (d).

In a classification problem, our hypothesis (h) may be the class to assign for a new data instance (d).

One of the easiest ways of selecting the most probable hypothesis given the data that we have that we can use as our prior knowledge about the problem. Bayes’ Theorem provides a way that we can calculate the probability of a hypothesis given our prior knowledge.

Bayes’ Theorem is stated as:

P(h|d) = (P(d|h) \* P(h)) / P(d)

Where

**P(h|d)** is the probability of hypothesis h given the data d. This is called the posterior probability.

**P(d|h)** is the probability of data d given that the hypothesis h was true.

**P(h)** is the probability of hypothesis h being true (regardless of the data). This is called the prior probability of h.

**P(d)** is the probability of the data (regardless of the hypothesis).

You can see that we are interested in calculating the posterior probability of P(h|d) from the prior probability p(h) with P(D) and P(d|h).

After calculating the posterior probability for a number of different hypotheses, you can select the hypothesis with the highest probability. This is the maximum probable hypothesis and may formally be called the [maximum a posteriori](https://en.wikipedia.org/wiki/Maximum_a_posteriori_estimation) (MAP) hypothesis.

This can be written as:

MAP(h) = max(P(h|d))

or

MAP(h) = max((P(d|h) \* P(h)) / P(d))

or

MAP(h) = max(P(d|h) \* P(h))

The P(d) is a normalizing term which allows us to calculate the probability. We can drop it when we are interested in the most probable hypothesis as it is constant and only used to normalize.

Back to classification, if we have an even number of instances in each class in our training data, then the probability of each class (e.g. P(h)) will be equal. Again, this would be a constant term in our equation and we could drop it so that we end up with:

MAP(h) = max(P(d|h))

This is a useful exercise, because when reading up further on Naive Bayes you may see all of these forms of the theorem.

**Linear Regression for Machine Learning**

Linear regression is perhaps one of the most well known and well understood algorithms in statistics and machine learning.

In this post you will discover the linear regression algorithm, how it works and how you can best use it in on your machine learning projects. In this post you will learn:

Why linear regression belongs to both statistics and machine learning.

The many names by which linear regression is known.

The representation and learning algorithms used to create a linear regression model.

How to best prepare your data when modeling using linear regression.

You do not need to know any statistics or linear algebra to understand linear regression. This is a gentle high-level introduction to the technique to give you enough background to be able to use it effectively on your own problems.

Before we dive into the details of linear regression, you may be asking yourself why we are looking at this algorithm.

Isn’t it a technique from statistics?

Machine learning, more specifically the field of predictive modeling is primarily concerned with minimizing the error of a model or making the most accurate predictions possible, at the expense of explainability. In applied machine learning we will borrow, reuse and steal algorithms from many different fields, including statistics and use them towards these ends.

As such, linear regression was developed in the field of statistics and is studied as a model for understanding the relationship between input and output numerical variables, but has been borrowed by machine learning. It is both a statistical algorithm and a machine learning algorithm.

Next, let’s review some of the common names used to refer to a linear regression model.

Linear Regression for Machine Learning

Linear regression is perhaps one of the most well known and well understood algorithms in statistics and machine learning.

In this post you will discover the linear regression algorithm, how it works and how you can best use it in on your machine learning projects. In this post you will learn:

Why linear regression belongs to both statistics and machine learning.

The many names by which linear regression is known.

The representation and learning algorithms used to create a linear regression model.

How to best prepare your data when modeling using linear regression.

You do not need to know any statistics or linear algebra to understand linear regression. This is a gentle high-level introduction to the technique to give you enough background to be able to use it effectively on your own problems.

Let’s get started.

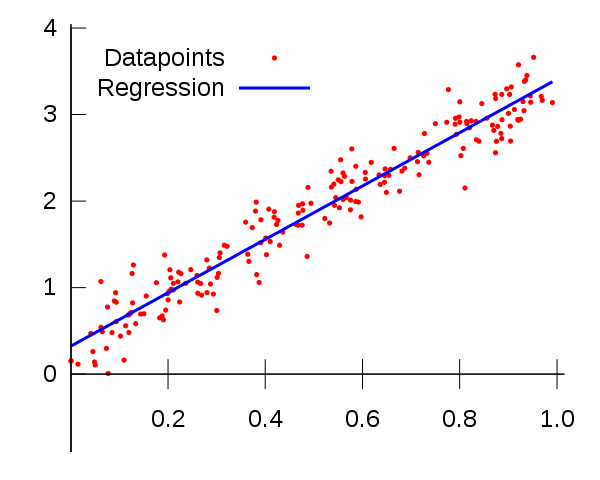


Fig 3.1 Linear Regression for Machine Learning

Machine learning, more specifically the field of predictive modeling is primarily concerned with minimizing the error of a model or making the most accurate predictions possible, at the expense of explainability. In applied machine learning we will borrow, reuse and steal algorithms from many different fields, including statistics and use them towards these ends.

As such, linear regression was developed in the field of statistics and is studied as a model for understanding the relationship between input and output numerical variables, but has been borrowed by machine learning. It is both a statistical algorithm and a machine learning algorithm.

Next, let’s review some of the common names used to refer to a linear regression model.

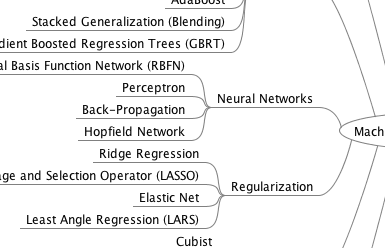


Fig 3.2 Sample of the handy machine learning algorithms mind map.

 When you start looking into linear regression, things can get very confusing.

The reason is because linear regression has been around for so long (more than 200 years). It has been studied from every possible angle and often each angle has a new and different name.

Linear regression is a **linear model**, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x).

When there is a single input variable (x), the method is referred to as **simple linear regression**. When there are **multiple input variables**, literature from statistics often refers to the method as multiple linear regression.

Different techniques can be used to prepare or train the linear regression equation from data, the most common of which is called **Ordinary Least Squares**. It is common to therefore refer to a model prepared this way as Ordinary Least Squares Linear Regression or just Least Squares Regression.

Now that we know some names used to describe linear regression, let’s take a closer look at the representation used.

Linear Regression Model Representation

[Linear regression](https://en.wikipedia.org/wiki/Linear_regression) is an attractive model because the representation is so simple.

The representation is a linear equation that combines a specific set of input values (x) the solution to which is the predicted output for that set of input values (y). As such, both the input values (x) and the output value are numeric.

The linear equation assigns one scale factor to each input value or column, called a coefficient and represented by the capital Greek letter Beta (B). One additional coefficient is also added, giving the line an additional degree of freedom (e.g. moving up and down on a two-dimensional plot) and is often called the intercept or the bias coefficient.

For example, in a simple regression problem (a single x and a single y), the form of the model would be:

y = B0 + B1\*x

In higher dimensions when we have more than one input (x), the line is called a plane or a hyper-plane. The representation therefore is the form of the equation and the specific values used for the coefficients (e.g. B0 and B1 in the above example).

It is common to talk about the complexity of a regression model like linear regression. This refers to the number of coefficients used in the model.

When a coefficient becomes zero, it effectively removes the influence of the input variable on the model and therefore from the prediction made from the model (0 \* x = 0). This becomes  relevant if you look at regularization methods that change the learning algorithm to reduce the complexity of regression models by putting pressure on the absolute size of the coefficients, driving some to zero.

Now that we understand the representation used for a linear regression model, let’s review some ways that we can learn this representation from data.

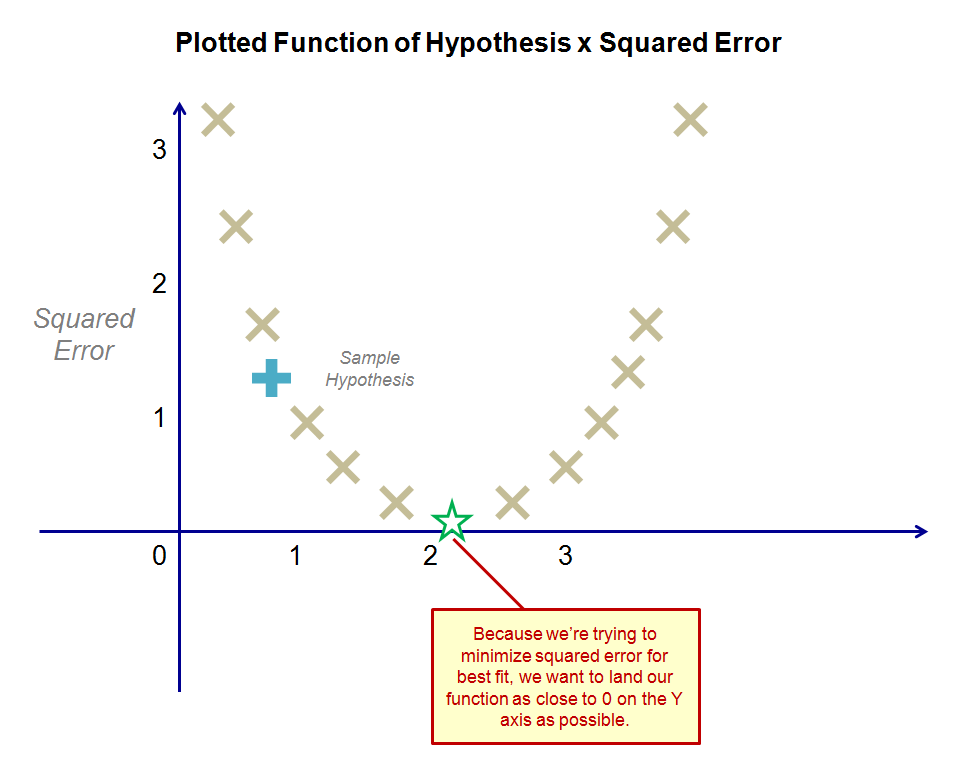


Fig 3.3 Linear Regression Learning the Model

Learning a linear regression model means estimating the values of the coefficients used in the representation with the data that we have available.

In this section we will take a brief look at four techniques to prepare a linear regression model. This is not enough information to implement them from scratch, but enough to get a flavor of the computation and trade-offs involved.

There are many more techniques because the model is so well studied. Take note of Ordinary Least Squares because it is the most common method used in general. Also take note of Gradient Descent as it is the most common technique taught in machine learning classes.

**1. Simple Linear Regression**

With simple linear regression when we have a single input, we can use statistics to estimate the coefficients.

This requires that you calculate statistical properties from the data such as means, standard deviations, correlations and covariance. All of the data must be available to traverse and calculate statistics.

This is fun as an exercise in excel, but not really useful in practice.

**2. Ordinary Least Squares**

When we have more than one input we can use Ordinary Least Squares to estimate the values of the coefficients.

The [Ordinary Least Squares](https://en.wikipedia.org/wiki/Ordinary_least_squares) procedure seeks to minimize the sum of the squared residuals. This means that given a regression line through the data we calculate the distance from each data point to the regression line, square it, and sum all of the squared errors together. This is the quantity that ordinary least squares seeks to minimize.

This approach treats the data as a matrix and uses linear algebra operations to estimate the optimal values for the coefficients. It means that all of the data must be available and you must have enough memory to fit the data and perform matrix operations.

It is unusual to implement the Ordinary Least Squares procedure yourself unless as an exercise in linear algebra. It is more likely that you will call a procedure in a linear algebra library. This procedure is very fast to calculate.

**3. Gradient Descent**

When there are one or more inputs you can use a process of optimizing the values of the coefficients by iteratively minimizing the error of the model on your training data.

This operation is called [Gradient Descent](https://en.wikipedia.org/wiki/Gradient_descent) and works by starting with random values for each coefficient. The sum of the squared errors are calculated for each pair of input and output values. A learning rate is used as a scale factor and the coefficients are updated in the direction towards minimizing the error. The process is repeated until a minimum sum squared error is achieved or no further improvement is possible.

When using this method, you must select a learning rate (alpha) parameter that determines the size of the improvement step to take on each iteration of the procedure.

Gradient descent is often taught using a linear regression model because it is relatively straightforward to understand. In practice, it is useful when you have a very large dataset either in the number of rows or the number of columns that may not fit into memory.

**4. Regularization**

There are extensions of the training of the linear model called regularization methods. These seek to both minimize the sum of the squared error of the model on the training data (using ordinary least squares) but also to reduce the complexity of the model (like the number or absolute size of the sum of all coefficients in the model).

Two popular examples of regularization procedures for linear regression are:

[Lasso Regression](https://en.wikipedia.org/wiki/Lasso_(statistics)): where Ordinary Least Squares is modified to also minimize the absolute sum of the coefficients (called L1 regularization).

[Ridge Regression](https://en.wikipedia.org/wiki/Tikhonov_regularization): where Ordinary Least Squares is modified to also minimize the squared absolute sum of the coefficients (called L2 regularization).

These methods are effective to use when there is collinearity in your input values and ordinary least squares would overfit the training data.

Now that you know some techniques to learn the coefficients in a linear regression model, let’s look at how we can use a model to make predictions on new data.

Making Predictions with Linear Regression

Given the representation is a linear equation, making predictions is as simple as solving the equation for a specific set of inputs.

Let’s make this concrete with an example. Imagine we are predicting weight (y) from height (x). Our linear regression model representation for this problem would be:

y = B0 + B1 \* x1

or

weight =B0 +B1 \* height

Where B0 is the bias coefficient and B1 is the coefficient for the height column. We use a learning technique to find a good set of coefficient values. Once found, we can plug in different height values to predict the weight.

For example, lets use B0 = 0.1 and B1 = 0.5. Let’s plug them in and calculate the weight (in kilograms) for a person with the height of 182 centimeters.

weight = 0.1 + 0.05 \* 182

weight = 91.1

You can see that the above equation could be plotted as a line in two-dimensions. The B0 is our starting point regardless of what height we have. We can run through a bunch of heights from 100 to 250 centimeters and plug them to the equation and get weight values, creating our line.

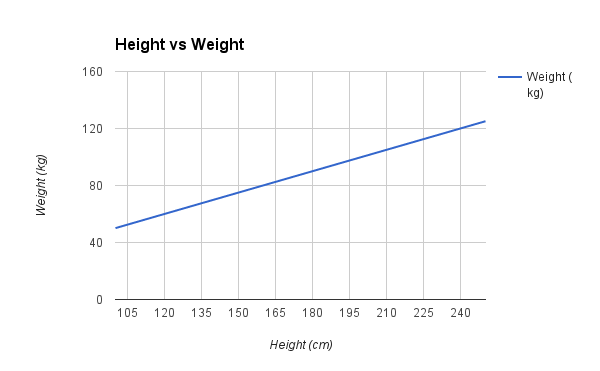


Fig 3.4 Sample Height vs Weight Linear Regression

Now that we know how to make predictions given a learned linear regression model, let’s look at some rules of thumb for preparing our data to make the most of this type of model.

Preparing Data For Linear Regression

Linear regression is been studied at great length, and there is a lot of literature on how your data must be structured to make best use of the model.

As such, there is a lot of sophistication when talking about these requirements and expectations which can be intimidating. In practice, you can uses these rules more as rules of thumb when using Ordinary Least Squares Regression, the most common implementation of linear regression.

Try different preparations of your data using these heuristics and see what works best for your problem.

**Linear Assumption**. Linear regression assumes that the relationship between your input and output is linear. It does not support anything else. This may be obvious, but it is good to remember when you have a lot of attributes. You may need to transform data to make the relationship linear (e.g. log transform for an exponential relationship).

**Remove Noise**. Linear regression assumes that your input and output variables are not noisy. Consider using data cleaning operations that let you better expose and clarify the signal in your data. This is most important for the output variable and you want to remove outliers in the output variable (y) if possible.

**Remove Collinearity**. Linear regression will over-fit your data when you have highly correlated input variables. Consider calculating pairwise correlations for your input data and removing the most correlated.

**Gaussian Distributions**. Linear regression will make more reliable predictions if your input and output variables have a Gaussian distribution. You may get some benefit using transforms (e.g. log or BoxCox) on you variables to make their distribution more Gaussian looking.

**Rescale Inputs**: Linear regression will often make more reliable predictions if you rescale input variables using standardization or normalization.

**Project management :**

Project management is the application of processes, methods, knowledge, skills and experience to achieve the project objectives. General. A project is a unique, transient endeavor, undertaken to achieve planned objectives, which could be defined in terms of outputs, outcomes or benefits.

Project management is the practise of initiating, planning, executing, controlling, and closing the [work](https://en.wikipedia.org/wiki/Work_(project_management)) of a [team](https://en.wikipedia.org/wiki/Project_team) to achieve specific goals and meet specific success criteria at the specified time. A [project](https://en.wikipedia.org/wiki/Project) is a temporary endeavor designed to produce a unique product, service or result with a defined beginning and end undertaken to meet unique goals and objectives, typically to bring about beneficial change or added value. The temporary nature of projects stands in contrast with [business as usual](https://en.wikipedia.org/wiki/Business_operations), which are repetitive, permanent, or semi-permanent functional activities to produce products or services. In practice, the [management](https://en.wikipedia.org/wiki/Management) of such distinct production approaches requires the development of distinct technical skills and management strategies.

**Software project management**

Software project management is the art and science of planning and leading software projects. It is a sub-discipline of [project management](https://en.wikipedia.org/wiki/Project_management) in which [software](https://en.wikipedia.org/wiki/Software) projects are planned, implemented, monitored and controlled.

The job pattern of an IT company engaged in software development can be seen split in two parts:

* Software Creation
* Software Project Management

A project is well-defined task, which is a collection of several operations done in order to achieve a goal (for example, software development and delivery). A Project can be characterized as:

* Every project may have a unique and distinct goal.
* Project is not routine activity or day-to-day operations.
* Project comes with a start time and end time.
* Project ends when its goal is achieved hence it is a temporary phase in the lifetime of an organization.
* Project needs adequate resources in terms of time, manpower, finance, material and knowledge-bank.

**Software Project**

A Software Project is the complete procedure of software development from requirement gathering to testing and maintenance, carried out according to the execution methodologies, in a specified period of time to achieve intended software product.

**Need of software project management**

Software is said to be an intangible product. Software development is a kind of all new stream in world business and there’s very little experience in building software products. Most software products are tailor made to fit client’s requirements. The most important is that the underlying technology changes and advances so frequently and rapidly that experience of one product may not be applied to the other one. All such business and environmental constraints bring risk in software development hence it is essential to manage software projects efficiently.



The image above shows triple constraints for software projects. It is an essential part of software organization to deliver quality product, keeping the cost within client’s budget constrain and deliver the project as per scheduled. There are several factors, both internal and external, which may impact this triple constrain triangle. Any of three factor can severely impact the other two.

Therefore, software project management is essential to incorporate user requirements along with budget and time constraints.

**Software Project Manager**

A software project manager is a person who undertakes the responsibility of executing the software project. Software project manager is thoroughly aware of all the phases of SDLC that the software would go through. Project manager may never directly involve in producing the end product but he controls and manages the activities involved in production.

A project manager closely monitors the development process, prepares and executes various plans, arranges necessary and adequate resources, maintains communication among all team members in order to address issues of cost, budget, resources, time, quality and customer satisfaction.

Let us see few responsibilities that a project manager shoulders -

**Managing People**

* Act as project leader
* Liaison with stakeholders
* Managing human resources
* Setting up reporting hierarchy etc.

**Managing Project**

* Defining and setting up project scope
* Managing project management activities
* Monitoring progress and performance
* Risk analysis at every phase
* Take necessary step to avoid or come out of problems
* Act as project spokesperson

**Software Management Activities**

Software project management comprises of a number of activities, which contains planning of project, deciding scope of software product, estimation of cost in various terms, scheduling of tasks and events, and resource management. Project management activities may include:

* **Project Planning**
* **Scope Management**
* **Project Estimation**

**Project Planning**

Software project planning is task, which is performed before the production of software actually starts. It is there for the software production but involves no concrete activity that has any direction connection with software production; rather it is a set of multiple processes, which facilitates software production. Project planning may include the following:

**Scope Management**

It defines the scope of project; this includes all the activities, process need to be done in order to make a deliverable software product. Scope management is essential because it creates boundaries of the project by clearly defining what would be done in the project and what would not be done. This makes project to contain limited and quantifiable tasks, which can easily be documented and in turn avoids cost and time overrun.

During Project Scope management, it is necessary to -

* Define the scope
* Decide its verification and control
* Divide the project into various smaller parts for ease of management.
* Verify the scope
* Control the scope by incorporating changes to the scope

**Project Estimation**

For an effective management accurate estimation of various measures is a must. With correct estimation managers can manage and control the project more efficiently and effectively.

Project estimation may involve the following:

* **Software size estimation**

Software size may be estimated either in terms of KLOC (Kilo Line of Code) or by calculating number of function points in the software. Lines of code depend upon coding practices and Function points vary according to the user or software requirement.

* **Effort estimation**

The managers estimate efforts in terms of personnel requirement and man-hour required to produce the software. For effort estimation software size should be known. This can either be derived by managers’ experience, organization’s historical data or software size can be converted into efforts by using some standard formulae.

* **Time estimation**

Once size and efforts are estimated, the time required to produce the software can be estimated. An effort required is segregated into sub categories as per the requirement specifications and interdependency of various components of software. Software tasks are divided into smaller tasks, activities or events by Work Breakthrough Structure (WBS). The tasks are scheduled on day-to-day basis or in calendar months.

The sum of time required to complete all tasks in hours or days is the total time invested to complete the project.

* **Cost estimation**

This might be considered as the most difficult of all because it depends on more elements than any of the previous ones. For estimating project cost, it is required to consider -

* + Size of software
  + Software quality
  + Hardware
  + Additional software or tools, licenses etc.
  + Skilled personnel with task-specific skills
  + Travel involved
  + Communication
  + Training and support

**Project Estimation Techniques**

We discussed various parameters involving project estimation such as size, effort, time and cost.Project manager can estimate the listed factors using two broadly recognized techniques

**Decomposition Technique**

This technique assumes the software as a product of various compositions.

There are two main models -

* **Line of Code** Estimation is done on behalf of number of line of codes in the software product.
* **Function Points** Estimation is done on behalf of number of function points in the software product.

**Empirical Estimation Technique**

This technique uses empirically derived formulae to make estimation.These formulae are based on LOC or FPs.

* **Putnam Model**

This model is made by Lawrence H. Putnam, which is based on Norden’s frequency distribution (Rayleigh curve). Putnam model maps time and efforts required with software size.

* **COCOMO**

COCOMO stands for COnstructiveCOstMOdel, developed by Barry W. Boehm. It divides the software product into three categories of software: organic, semi-detached and embedded.

**Project Scheduling**

Project Scheduling in a project refers to roadmap of all activities to be done with specified order and within time slot allotted to each activity. Project managers tend to define various tasks, and project milestones and they arrange them keeping various factors in mind. They look for tasks lie in critical path in the schedule, which are necessary to complete in specific manner and strictly within the time allocated. Arrangement of tasks which lies out of critical path are less likely to impact over all schedule of the project.

For scheduling a project, it is necessary to -

* Break down the project tasks into smaller, manageable form
* Find out various tasks and correlate them
* Estimate time frame required for each task
* Divide time into work-units
* Assign adequate number of work-units for each task
* Calculate total time required for the project from start to finish

**Resource management**

All elements used to develop a software product may be assumed as resource for that project. This may include human resource, productive tools and software libraries.

The resources are available in limited quantity and stay in the organization as a pool of assets. The shortage of resources hampers the development of project and it can lag behind the schedule. Allocating extra resources increases development cost in the end. It is therefore necessary to estimate and allocate adequate resources for the project.

Resource management includes -

* Defining proper organization project by creating a project team and allocating responsibilities to each team member
* Determining resources required at a particular stage and their availability
* Manage Resources by generating resource request when they are required and de-allocating them when they are no more needed.

**Project Risk Management**

Risk management involves all activities pertaining to identification, analysing and making provision for predictable and non-predictable risks in the project. Risk may include the following:

* Experienced staff leaving the project and new staff coming in.
* Change in organizational management.
* Requirement change or misinterpreting requirement.
* Under-estimation of required time and resources.
* Technological changes, environmental changes, business competition.

**Risk Management Process**

There are following activities involved in risk management process:

* **Identification -** Make note of all possible risks, which may occur in the project.
* **Categorize -** Categorize known risks into high, medium and low risk intensity as per their possible impact on the project.
* **Manage -** Analyze the probability of occurrence of risks at various phases. Make plan to avoid or face risks. Attempt to minimize their side-effects.
* **Monitor -** Closely monitor the potential risks and their early symptoms. Also monitor the effects of steps taken to mitigate or avoid them.

**Project Execution & Monitoring**

In this phase, the tasks described in project plans are executed according to their schedules.

Execution needs monitoring in order to check whether everything is going according to the plan. Monitoring is observing to check the probability of risk and taking measures to address the risk or report the status of various tasks.

These measures include -

* **Activity Monitoring -** All activities scheduled within some task can be monitored on day-to-day basis. When all activities in a task are completed, it is considered as complete.
* **Status Reports -** The reports contain status of activities and tasks completed within a given time frame, generally a week. Status can be marked as finished, pending or work-in-progress etc.
* **Milestones Checklist -** Every project is divided into multiple phases where major tasks are performed (milestones) based on the phases of SDLC. This milestone checklist is prepared once every few weeks and reports the status of milestones.

**Project Communication Management**

Effective communication plays vital role in the success of a project. It bridges gaps between client and the organization, among the team members as well as other stake holders in the project such as hardware suppliers.

Communication can be oral or written. Communication management process may have the following steps:

* **Planning** - This step includes the identifications of all the stakeholders in the project and the mode of communication among them. It also considers if any additional communication facilities are required.
* **Sharing** - After determining various aspects of planning, manager focuses on sharing correct information with the correct person on correct time. This keeps every one involved the project up to date with project progress and its status.
* **Feedback** - Project managers use various measures and feedback mechanism and create status and performance reports. This mechanism ensures that input from various stakeholders is coming to the project manager as their feedback.
* **Closure** - At the end of each major event, end of a phase of SDLC or end of the project itself, administrative closure is formally announced to update every stakeholder by sending email, by distributing a hardcopy of document or by other mean of effective communication.

After closure, the team moves to next phase or project.

**Configuration Management**

Configuration management is a process of tracking and controlling the changes in software in terms of the requirements, design, functions and development of the product.

IEEE defines it as “the process of identifying and defining the items in the system, controlling the change of these items throughout their life cycle, recording and reporting the status of items and change requests, and verifying the completeness and correctness of items”.

Generally, once the SRS is finalized there is less chance of requirement of changes from user. If they occur, the changes are addressed only with prior approval of higher management, as there is a possibility of cost and time overrun.

**Project management Tools:**

Project management required tools to manage the work , time and resources. At present many of the software are available for project management. Some of the popular software tools are as follows.

### 01. [Trello](http://send.getapp.com/aff_c?offer_id=677&aff_id=1371)

Trello is an project management tool, instead this app is a free visual way to to glance at the entire project with a single view. With Trello you can organise cards, these cards can be your thoughts, conversations and to-do lists and be placed on a board for everyone to collaborate on.

### 02. [Basecamp](http://send.getapp.com/aff_c?offer_id=637&aff_id=1371)

Basecamp is the granddaddy of project management apps. Basecamp is considered the leading project management tool around. It boost a simple and easy to use interface to collaborate with your team and client. It allows you to create multiple projects and setup discussions, write to-do lists, manage files, create and share documents, and organise dates for scheduling.

### 03. [Teamwork Projects](http://send.getapp.com/aff_c?offer_id=947&aff_id=1371)

Teamwork Projects is the ultimate productivity tool to manage projects with your team. Teamwork allows you to keep all your projects, tasks and files all in one place and easily collaborate with a team. Teamwork helps you to visualise the entire project through a marked calendar and gantt chart and setup reporting. Teamwork supports file management with Google Drive, Box.com and Dropbox. As well as integration with leading apps such as third party accounting software and customer support apps.

### 04. [Resource Guru](https://resourceguruapp.com/)

Billed as the "simple way to schedule people, equipment and other resources", Resource Guru is a streamlined resource scheduling and leave management tool that’s designed to keep your projects on track. You can plan your team's workloads, receive daily booking reminders, report on KPIs, and more. Apple, Saatchi & Saatchi and Deloitte are among some of the cloud-based team calendar’s heavyweight customers.

### 05. [ActiveCollab](http://send.getapp.com/aff_c?offer_id=949&aff_id=1371)

ActiveCollab recently released its new version 5.0. The new revamped app is now more powerful and focused project management tool. It offers team collaborating features, task management, time tracking and importing expenses. One of the biggest asset of ActiveCollab is it offers invoicing features. You are able to track payments and expenses and have invoices paid directly within ActiveCollab with PayPal, and other credit card payments.

### 06. [Zoho Projects](http://send.appdoubler.com/aff_c?offer_id=101&aff_id=1371)

Zoho offers a wide range of business software including Projects. Zoho Projects is an proficient tool to project plan and project coordinator from start to finish. It boost all the features you need for project management with some advance features including reporting, integration with Google Apps and Dropbox, bug tracking, setup Wiki Pages to build a repository of information, forums and more.

### 07. [Jira](http://send.getapp.com/aff_c?offer_id=281&aff_id=1371)

Jira is specifically targeted for software development teams. Jira offers abilities to raise issues and bugs. Jira makes it real easy to track bugs and see which issues are still outstanding and how much time was spent on each task. Jira offer other products including Confluence a document collaboration tool, and HipChat a team chat and video and file sharing platform and other products.

### 08. [Asana](http://send.getapp.com/aff_c?offer_id=587&aff_id=1371)

Asana is the easiest way for teams to track their work so everyone knows who's doing what, by when. With tasks, projects, conversations and dashboards, Asana keeps your work organized, and teammates accountable so you can move work forward faster. Asana also lets you keep track of your work wherever you are with mobile apps for both iOS and Android.

### 09. [Podio](http://send.getapp.com/aff_c?offer_id=951&aff_id=1371)

Podio is a ever growing tool to organise and communication tool for any business. Podio allows you to personalise this platform to fit your business needs. Besides being able to communicate with a team, setup task management, use as a file storage system, like a traditional project management app, Podio can be an internal intranet for all your colleagues and departments to interact.

### 10. [Freedcamp](https://freedcamp.com/)

Whatever your project may be, either setting up an event, a web project or organising a wedding, Freedcamp helps you organise and plan effectively. Freedcamp has an organised dashboard to view the entire project at a glance. You can easily setup tasks, use sticky notes to visually setup tasks and organise them into the calendar. Freedcamp provides advance add-ons for high level business use including CRM, invoicing, issue tracking and setting up wiki pages.

### 11. [Wrike](http://send.getapp.com/aff_c?offer_id=239&aff_id=1371)

Wrike is advance application to help you work smarter. By making sure you are always staying on track and ensure you have the adequate resources to finish on time and on budget.Setting up tasks, engage your team and integrate with your business tools including Google Apps, Microsoft Excel, Dropbox and many more is so easy with Wrike.

**PO and Their Relevance to project**

**PO1: Engineering knowledge:**Apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

In this project creation process engineering knowledge of the software engineering and Electronics engineering have been applied. we have used software engineering , HTML,xml, java , android , java script , php , j2ee, data base , oracle , my sql , mango and other programming language and database to the project. We have applied all above engineering subjects in our projects.

**PO2: Problem analysis:**Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

In our projects we have identified an problem , once verified by the client we have worked to identify the solution using all of our theoretical and practical knowledge.

**PO3: Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

**PO4: Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**PO5: Modern tool usage:**Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

In the project development we have applied Integrated Development Environment IDE for the rapid development of the code, used web server for the software development.

**PO6: The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

In 1961 , the Conference of Engineering Societies of Western Europe and the United States of America defined "professional engineer" as follows.

A professional engineer is competent by virtue of his/her fundamental education and training to apply the scientific method and outlook to the analysis and solution of engineering problems. He/she is able to assume personal responsibility for the development and application of engineering science and knowledge, notably in research, design, construction, manufacturing, superintending, managing and in the education of the engineer. His/her work is predominantly intellectual and varied and not of a routine mental or physical character. It requires the exercise of original thought and judgement and the ability to supervise the technical and administrative work of others. His/her education will have been such as to make him/her capable of closely and continuously following progress in his/her branch of engineering science by consulting newly published works on a worldwide basis, assimilating such information and applying it independently. He/she is thus placed in a position to make contributions to the development of engineering science or its applications. His/her education and training will have been such that he/she will have acquired a broad and general appreciation of the engineering sciences as well as thorough insight into the special features of his/her own branch. In due time he/she will be able to give authoritative technical advice and to assume responsibility for the direction of important tasks in his/her branch.

**PO7: Environment and sustainability:** Understand the impact of the professional engineering solutions in and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

Sustainability is the ability to continue a defined behavior indefinitely. Sometimes environmental, social and economic are termed to be the three pillars of sustainability.

**PO8: Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice

The ethics of engineers and the fundamental principles for Engineers are as follows.

Engineers uphold and advance the integrity, honor and dignity of the engineering profession by:

I. using their knowledge and skill for the enhancement of human welfare;

II.being honest and impartial, and servicing with fidelity the public, their employers and clients;

III. Striving to increase the competence and prestige of the engineering profession; and

IV. Supporting the professional and technical societies of their disciplines.   
  
  
**PO9. Individual and team work**: Function effectively as an individual and as a member or leader in diverse teams, and in multidisciplinary settings.  
  
  
To work successful in team a team member must have following capabilities.

**1. The Ability to Listen**

it is important to listen to one another's ideas. Too often in a business setting, you have a group of people simply waiting for their turn to speak, not paying one iota of attention to the persons on their left or right. So it is a good teamwork skill to have the ability to listen

**2. Check Your Ego**

This isn't saying abandon your ego all together, because that isn't healthy. But leaving your ego at the door temporarily is a very important team work skill. The reason this is so essential is because there is always someone better than you at something, no matter how brilliant you are.

**3. Critique**

By critique, I mean constructive criticism. Be able to give others constructive criticism and be able to listen to others critique your ideas and work. There shouldn't be any offense taken to constructive criticism. You all want to succeed, and this is a vital step in doing so.

**4. Delegation**

The mentality must be applied to teamwork. Delegate roles to those who do them best.

**5. Show Respect**

If you and another person happen to be paired up and can't stand each other, you can still put that aside for a couple of hours, treat each other civilly, and complete the tasks at hand. You may even overcome the dislike toward one another.

**6. Be Helpful**

This is simple.If one of your teammates does not understand an idea, discussion, or task that is being completed, take the necessary time to explain it to them and work with them. There are no weak links when everyone helps one another. Some take longer to learn than others, but that doesn't mean that they are of less intelligence. If in a meeting someone asks a question because they don't understand, don't frown at them. Just answer the questions patiently and concisely.

**7. Question One Another**

If someone brings up a topic of discussion and a solution to this topic, question them. Respectfully question, don't badger. Rather, ask them how it will work, why it will work over the long-run, and how everyone else can implement the idea.

**8. Participation**

Have the entire team encourage shy people to engage in the topics of discussion. Don't demand it, but make them realize that you really want to hear their ideas.

**9. Rational Debate**

Bad ideas are bad for teams. Spirited, friendly, rational debate is where facts come forward, ideas are born, and quality rises to the top.

**10. Set The Right Environment**

Try to make the space in which your team is assembled as comfortable, relaxing, and inviting as possible. You do not want your team to be tense and with frayed nerves.

**PO 10: Communication:**Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**PO11: Project management and finance:**Demonstrate knowledge and understanding of the engineering management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

Project management is the application of processes, methods, knowledge, skills and experience to achieve the project objectives. In general project is a unique, transient endeavour, undertaken to achieve planned objectives, which could be defined in terms of outputs, outcomes or benefits.

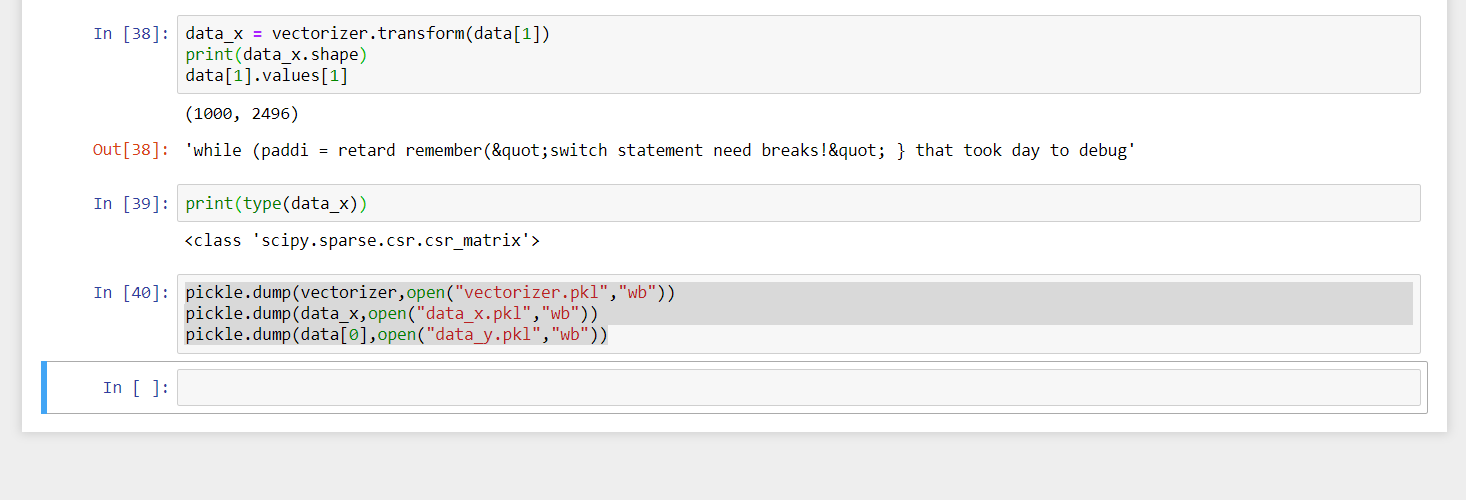
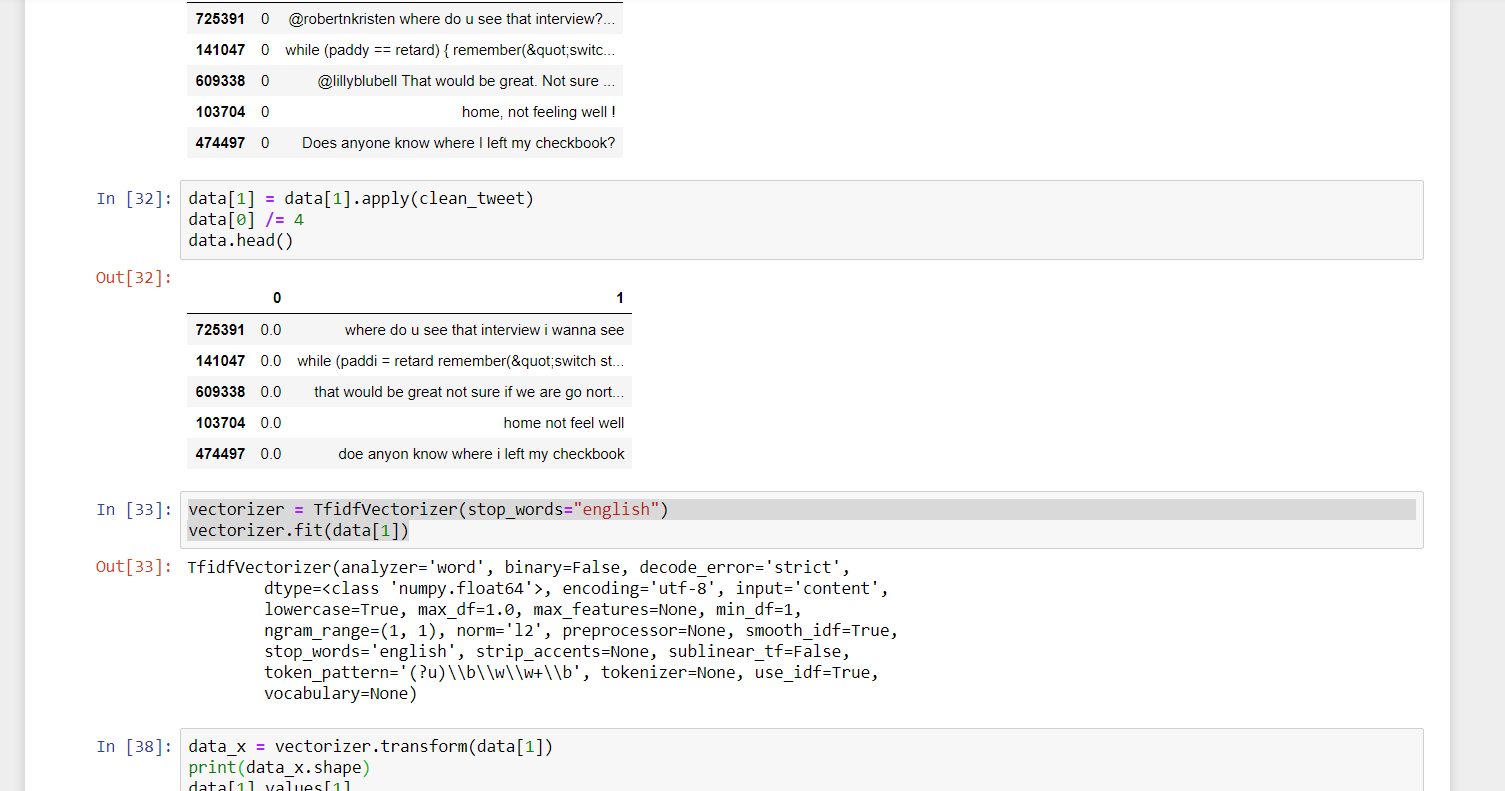
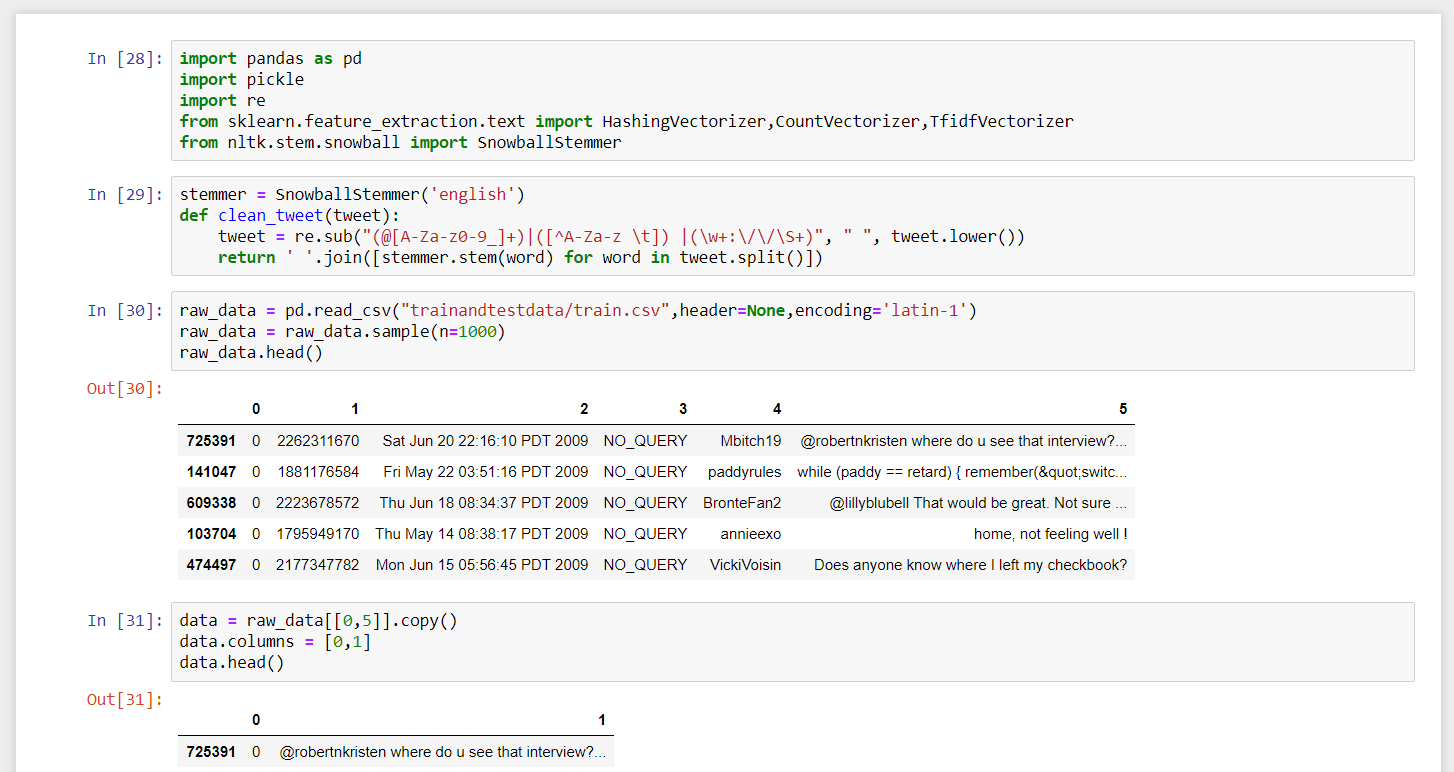
**PO12: Life-long learning**: Recognize the need for and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

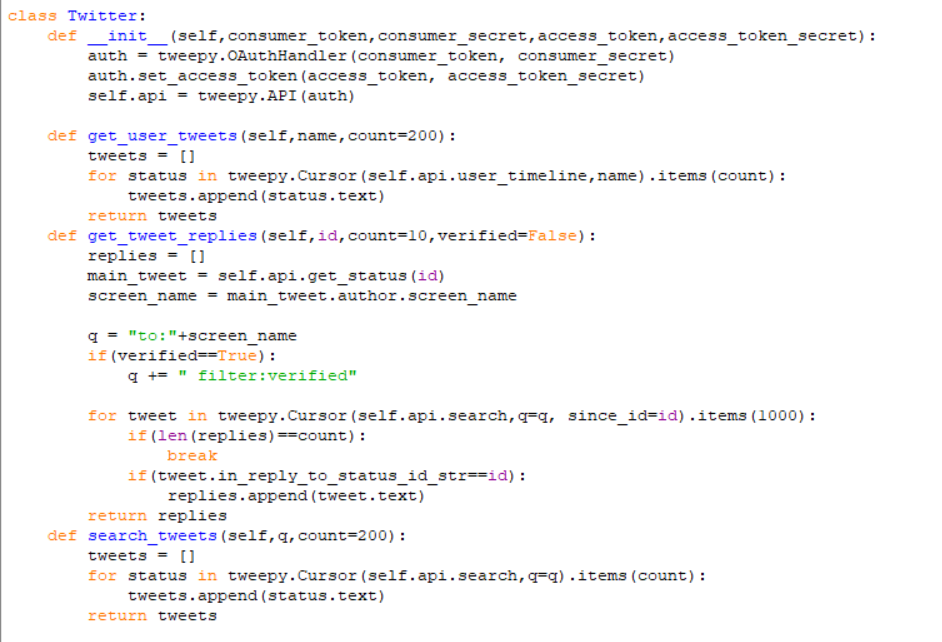
Life Long Learning means is the provision or use of both formal and informal learning opportunities throughout people's lives in order to foster the continuous development and improvement of the knowledge and skills needed for employment and personal fulfillment

**CHAPTER 4**

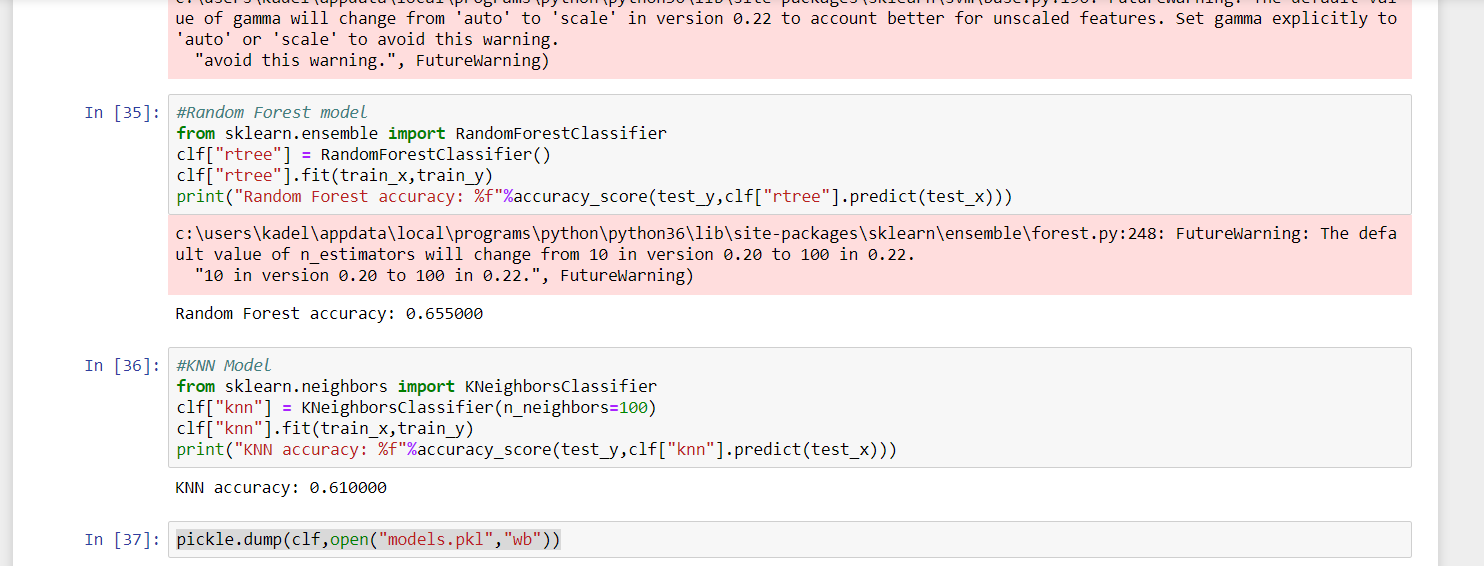
**PROJECT IMPLEMENTATION**

1. Sprint Backlog-1

Preprocessing.ipynb  


Twitter.py

1. Sprint Backlog-2



1. Sprint Backlog-3

( ***Project Design , Algo ,Database Design , ERD , Project Code ( Sample code only Limited to Max of 2-3 Page if needed , Project Screen Shots*** )

1. Sprint Backlog-4  
     
   ( ***Project Design , Algo ,Database Design , ERD , Project Code ( Sample code only Limited to Max of 2-3 Page if needed , Project Screen Shots*** )

**CHAPTER 5**

**CONCLUSION**

( 3 to 5 page)

Results

Conclusion

Future Scope

**ANNEXURES**

References

Research Paper ( if Presented and approved for publication)

DST Document presented for grants.

CV

**Instructions**

For Chapter Heading

Font: Times New Roman

Font Size: 14, Bold

Alignment: Center

Line Spacing 1.5

For Paragraph Heading

Font: Times New Roman

Font Size: 12, Bold

Alignment: Left

Line Spacing 1.5

For paragraph

Font: Times New Roman

Font Size: 12

Line Spacing 1.5

Alignment: Justify

Line Spacing 1.5