CHAPTER 1

INTRODUCTION

1.1 Introduction

For our final project, we have chosen to analyze a movie review. What we will be extracting from the data set is the significance of attributes that result in a large gross revenue of a movie. The goal of analyzing this data set is to successfully figure out which attributes are the most significant when determining future success of a movie title before it is released. Critics and human instinct, when it comes to movies, is sometimes unreliable. We want to be able to accurately predict what attributes in fluence movie success based on several characteristics in specific areas such as the IMDB website.

1.2 Motivation and Problem Statement

An article published in a newspaper or magazine that describes and evaluates a movie. Reviews are typically written by journalists giving their opinion of the movie. For many of us, reviews are like one written by our friends on Facebook, are important in making our decision to watch a movie.

Similarly, these reviews are available to movie production companies which helps them to understand sentiment and check the popularity of their films to figure out new marketing strategies and future directions. Human mind can read and understand whether a review is positive but for movie studios it is difficult to hire employees to simply read and judge movie opinions. So here comes Machine Learning to rescue - to process, reliably extract and classify the sentiment of unstructured movie reviews. Regarding this scenario, sentiment analysis is a brilliant term of artificial intelligence for understanding the user or audience's sentiment in the basis of polarity and subjectivity.

IMDb Movie has a lot of users and the reviews of a movie users are increasing day by day. "Sentiment Analysis on IMDb Movie Reviews" scraps the review data as data set and completes the classification of review texts by subjectivity or objectivity and negative or positive attitude of users. This project presents and empirical study of efficacy of classifying movie review.

Thus, this is a completely different approach by removing the unstructured data and then classifying reviews employing Natural Language Processing.

1.3 Project Scope

This section will describe the scope of the project, and briefly outline the preconditions and constraints that narrows down the focus area of this project.

Our first experiments, uses a dataset based on positive, negative and neutral movie reviews from IMDb. We gathered reviews from IMDb website and other available resources. Our dataset contains 75000 reviews. It is intended as a larger benchmark dataset for sentiment classification, and consists of movie reviews. Our second experiments, uses a dataset based on sarcastic and non-sarcastic movie reviews from IMDb. We gathered reviews from IMDb

website and other available resources. Our dataset contains 50000 review. The age of the datasets, does not concern us, since the theories tested in this thesis, can be applied to newer datasets without any problems. The trustworthiness of the documents and data sources will not be a focus of this project.

Sentiment analysis methods till now have been used to detect the polarity in the thoughts and opinions of all the users that access social media. Researchers and Businesses are very interested to understand the thoughts of people and how they respond to everything happening around them. Movies Industries can use this project to evaluate their movies and services and to improve their movies and services. Sentiment analysis by this project has more possibility and accuracy rather than manual analysis of sentiment data virtually as we have included sarcasm detection along with sentiment analysis to improve the accuracy.

Sarcasm detection can help us understand the polarity of a review correctly. The users sometimes tend to express his/her sentiment in the form of sarcastic utterances. The sarcastic utterance usually shifts the polarity of text from negative to positive and likewise. There is a lot of scope in analyzing the users review on the web. Now-a-days, there are lots of website which post the reviews of different movies provided by different reputed movies industries. Sentiment analysis will have to pace up with this change. "Sentiment Analysis on IMDb movie reviews" can help the movie industries to change strategies based on user feedback.

1.4 Project Objective

The objective of this project is to show how sentimental analysis can help improving the user experience over a system interface. The learning algorithm will learn what users' emotions are from statistical data then determine the sentiment and if there is any utterances of sarcastic review. Reviews will be classified into positive, negative and neutral sentiments. "Sentiment Analysis on IMDb Movie Reviews" is the process of user review data mining, analyzing of user's feedback and display a graph of the result which presents the polarity values basis of positivity and negativity.

1.5 Impact, Significant and Contribution

We've put impact, Significant and contribution in the center of our feedback cycle. The ability to understand positive feelings, or negative feelings has been difficult, for machines that lack feelings. Movie industries want to accommodate the sentiment analysis tools into areas of user feedback, marketing. Although, by having this project, the industries will able to track the users and fulfill their parameter. The industries can easily detect the sentiment of the users by analyzing their review data.

1.6 Background Analysis

Creating a predictive model for this data set is not vital to human existence, however it would be useful for some movie- goers. This analyzation pertains to the entertainment/movie industry. It can help producers, actors, actresses, directors, film investors, and movie-goers determine how successful the proposed movie will be. Without

the predictive modeling, there would only be gut decisions/personal preferences about how a movie will turn out. Not everyone thinks that a certain actor or actress is amazing, therefore saying the entirety of the movie is amazing. Putting it in terms of analytical processing makes the prediction more stable and unbiased. This project would be deemed significant to this group of people mentioned previously because it will be an unbiased predictive data set that will be utilized to determine gross revenue. Every producer and director believe their movie will be one of the greatest, and they will do everything in their power to make it the greatest. However, majority of the time, this turns out to be false. They can take this data set and implement it into their thought process when planning their movie. On the flipside, I hear a lot of the time that people will go see a movie and say I just wasted x amount of money to see that horrible film!". Movie-goers can use this data set to make the same predictions once the movie is announced with primary and supporting actors/actresses. It could possibly save movie-goers money when debating on whether to go see a movie or not.

Goals:

There are a couple of goals that We wish to achieve with this data set. The goals We wish to achieve are:

- Assist directors and producers in maximizing their potential revenue of a proposed film
- Save money or spend money wisely when debating on seeing a new film
- Gain practice in using multiple linear regression
- Develop more skill in pre-processing techniques such as data partitioning and handling missing data
- Learn more about post processing technique sensitivity analysis

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The literature review is an important part of a Project. It should be thorough and accurate. This project going to review on the article, book and internal resources to study about the "Sentiment Analysis on IMDb movie reviews".

- An analysis of temporal multivariate networks derived from IMDB – Used methods such as (p, q)-core and 4-ring to identify subgraphs and short cycles
- An analysis of how an individual's movie preferences correlated with Oscar winning titles – Used Linear Regression
- An analysis a movie dataset using regression and k-nearest neighbor methods Literature Review
- An analysis a movie dataset using regression and k-nearest neighbor methods

2.1.1 Natural Language Processing

Natural language processing (NLP) is the ability of a computer program to understand human language as it is spoken. NLP is a component of artificial intelligence (AI). **[1]** The development of NLP applications is challenging because computers traditionally require humans to "speak" to them in a programming language that is precise, unambiguous and highly structured, or through a limited number of clearly enunciated voice commands. Human speech, however, is not always precise -- it is often ambiguous and the linguistic structure can depend on many complex variables, including slang, regional dialects and social context.

2.1.2 Natural Language Toolkit (NLTK)

The Natural Language Toolkit (NLTK) is a platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP). [2] it provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries.

NLTK has been called "a wonderful tool for teaching, and working in, computational linguistics using Python," and "an amazing library to play with natural language". It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning. It also includes graphical demonstrations and sample data sets as well as accompanied by a cook book and a book which explains the principles behind the underlying language processing tasks that NLTK supports.

The Good:[17]

- Makes NLP easier and more accessible

- ❖ Python (great learning language)
- ❖ Lots of documentation
- ❖ Designed for training custom models
- ❖ Includes many training corpora
- ❖ Many algorithms to experiment with

The Bad:[17]

- ❖ Few out-of-the-box solutions (Pattern)
- ❖ Not designed for big-data (Mahout)
- ❖ Doesn't have latest algorithms (Scikits-Learn)
- ❖ No online or active learning algorithms

2.1.3 Stop Words

Stop words are generally thought to be a "single set of words". It really can mean different things to different applications. For example, in some applications removing all stop words right from determiners (e.g. the, a, an) to prepositions (e.g. above, across, before) to some adjectives (e.g. good, nice) can be an appropriate stop word list. To some applications however, this can be detrimental. [3] For instance, in sentiment analysis removing adjective terms such as 'good' and 'nice' as well as negations such as 'not' can throw algorithms off their tracks. In such cases, one can choose to use a minimal stop list consisting of just determiners or determiners with prepositions or just coordinating conjunctions depending on the needs of the application. Examples of minimal stop word lists that can use:

- ❖ **Determiners** –Determiners tend to mark nouns where a determiner usually will be followed by a noun. Examples: the, a, an, another
- ❖ **Coordinating conjunctions** – Coordinating conjunctions connect words, phrases, and clauses.
- Examples: for, an, nor, but, or, yet, so
- ❖ **Prepositions** – Prepositions express temporal or spatial relations.
- Examples: in, under, towards, before

In some domain specific cases, such as clinical texts, we may want a whole different set of stop words. For example, terms like "mcg" "dr" and "patient" may have less discriminating power in building intelligent applications compared to terms such as 'heart' 'failure' and 'diabetes'. In such cases, we can also construct domain specific stop words as opposed to using a published stop word list.

2.1.4 NLTK Stop Words

Natural Language Processing with Python Natural language processing (NLP) is a research field that presents many challenges such as natural language understanding. Text may contain stop words like ‘the’, ‘is’, ‘are’. Stop words can be filtered from the text to be processed. There is no universal list of stop words in NLP research, however the NLTK module contains a list of stop words. [4]

2.1.5 Lemmatization

Lemmatization is an important aspect of natural language understanding (NLU) and natural language processing (NLP) and plays an important role in big data analytics and artificial intelligence (AI). Complex algorithms use the rules of linguistic morphology, in context with a particular language's vocabulary, to group words used in speech and writing by inflected forms. Deep learning is used to analyze and understand the grouping as a whole, so when any inflectional form of a word is mentioned, the base term's entire lemmatization is included.

In computational linguistics, lemmatization is the algorithmic process of determining the lemma of a word based on its intended meaning. [5] Unlike stemming, lemmatization depends on correctly identifying the intended part of speech and meaning of a word in a sentence, as well as within the larger context surrounding that sentence, such as neighboring sentences or even an entire document. As a result, developing efficient lemmatization algorithms is an open area of research. In many situations, it seems as if it would be useful for a search for one of these words to return documents that contain another word in the set. The goal of lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form.

For instance: am, are, is

- be car, cars, car's, cars'
- car

The result of this mapping of text will be something like:

the boy's cars are different colors ⇒ the boy car be different color

2.1.6 Scikit-learn

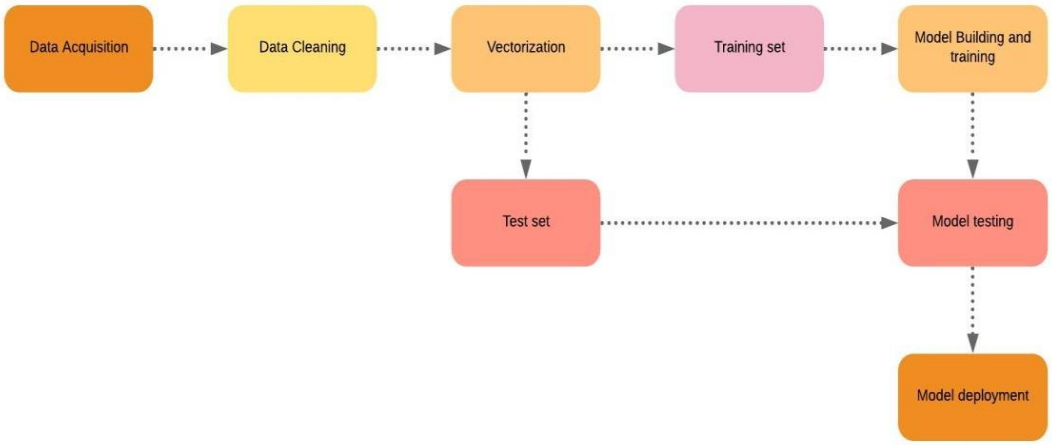
Scikit-learn (formerly scikits-learn and also known as sk-learn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including naïve Bayes, logistic regression, support vector machine, k-means etc, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. [6]

We have used naïve bayes and logistic regression from scikit-learn for classifying our sentiment and sarcasm data.

2.2 Methodology

- **Data acquisition:** We will collect reviews from IMDb website and prepare a dataset.

- **Data pre-processing:** The data gathered will be in a raw format and this data won't be feasible for the analysis. Therefore, certain steps will be executed to convert the data into a small clean data set.



- **Vectorization:** Pre-processed data will be converted into numbers using Vectorization model e.g. Bag-of-Words, TF-IDF vectorizer.

- **Splitting data into training set and testing set:** Data will be split into training and testing set (80-20%).

- **Model building, training and testing:** An important point to note is that during training the classifier only the training set is available. The test data set won't be used during training the classifier. The test set will only be available during testing the classifier.

CHAPTER 3

SYSTEM IMPLEMENT PROCESS

3.1 Process

The steps of our system model. Those are:

- The first step is data acquisition from imdb using a script to scrape review as dataset.
- The second step is analyzing data to sort out the number of words, characters and other attributes to be needed for next steps.
- The third step is stop words count for natural language processing.
- The fourth step is data cleaning for reducing of unwanted data\symbols or special characters from the dataset.
- The fifth step is lemmatization of the words.
- The sixth step is to classify the reviews on the basis of polarity.
- The seventh step is to detect sarcasm in the reviews.
- The final step is graphical representation of confusion matrix.

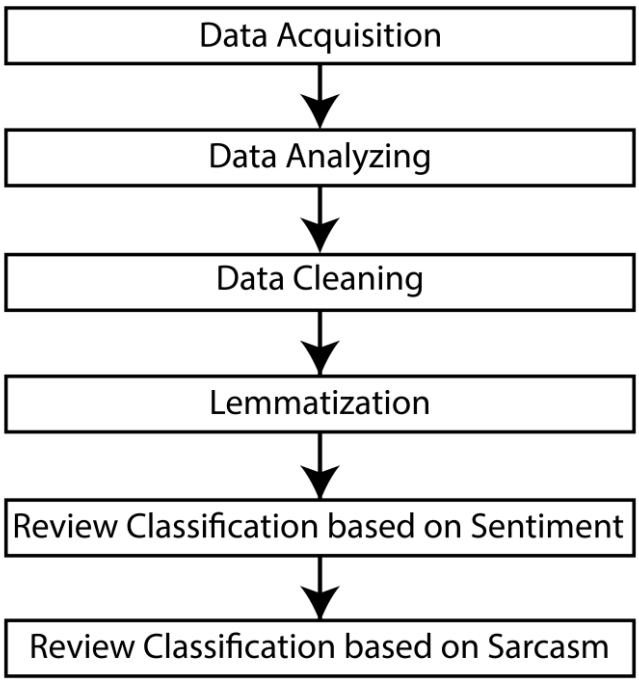


Figure-3.1: Process of system Implementation

3.2 Tools

The main tools used to develop this project is python (OOP Language), Python library, package and module: NumPy, matplotlib, pandas, re, nltk.corpus and nltk.

Python: Python 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. [7] Python is dynamically typed and garbage collected. It supports multiple programming paradigms, including procedural, object oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library. The primary language of our project is python.

Matplotlib: Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+. [8] There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. We have used it for graphical representation of polarity and irony of reviews.

Pandas: Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. [9] The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.

Regular Expression (RE): Regular expression is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern. [10] Usually such patterns are used by string searching algorithms for "find" or "find and replace" operations on strings, or for input validation. Regular expressions are widely used in UNIX world. The Python module re provides full support for Perl-like regular expressions in Python. We have used it for removing unwanted symbols and characters from the reviews.

NLTK: The Natural Language Toolkit (NLTK) is a platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP). it provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries.[11]

3.3 Data Acquisition:

The goal of data acquisition is to find datasets that can be used to train machine learning models. There are largely three approaches in the literature: data discovery, data augmentation, and data generation. Data discovery is necessary when one wants to share or search for new datasets and has become important as more datasets are available on the Web and corporate data lakes.[15] Data augmentation complements data discovery where

existing datasets are enhanced by adding more external data. Data generation can be used when there is no available external dataset, but it is possible to generate crowdsourced or synthetic datasets instead.

3.4 Data Labeling

Once enough data has been acquired, the next step is to label individual examples. For instance, given an image dataset of industrial components in a smart factory application, workers can start annotating if there are any defects in the components. In many cases, data acquisition is done along with data labeling. When extracting facts from the Web and constructing a knowledge base, then each fact is assumed to be correct and thus implicitly labeled as true. When discussing the data labeling literature, it is easier to separate it from data acquisition as the techniques can be quite different.[15]
We believe the following categories provide a reasonable view of understanding the data labeling landscape:

• **Use existing labels:** An early idea of data labeling is to exploit any labels that already exist. There is an extensive literature on semi-supervised learning where the idea is to learn from the labels to predict the rest of the labels.

• **Crowd-based:** The next set of techniques are based on crowdsourcing. A simple approach is to label individual examples. A more advanced technique is to use active learning where questions to ask are more carefully selected. More recently, many crowdsourcing techniques have been proposed to help workers become more effective in labeling.

• **Weak labels:** While it is desirable to generate correct labels all the time, this process may be too expensive an alternative approach is to newly generate less than perfect labels (i.e., weak labels), but in large quantities to compensate for the lower quality. Recently, the latter approach is gaining more popularity as labeled data is scarce in many new applications.

3.5 Implementation Issues and Challenges

Sentiment analysis classifies text as positive, negative or else neutral, so it can be thought as text classification task. Text classification has many classes as there are many topics but sentiment analysis has only three classes. However, there are many factors that make sentiment analysis difficult compared to traditional text classification. There are several defined elements in a piece of text that factor into sentiment analysis: the object, the attributes, the opinion holder, the opinion orientation, and the opinion strength.[16]

- **Object**: The product, service, individual, organization, event or topic being analyzed. Example: iPhone
- **Attributes**: The specific components and properties of the object Component examples: battery, touch screen, headphone jack Property examples: size, weight, processing speed
- **Opinion holder**: The person or organization who's expressing the sentiment

Example: the person who purchased the iPhone

- **Opinion orientation (polarity)**: The general position of the opinion
  Examples: positive, negative or neutral
- **Opinion strength**: The level, scale or intensity of the opinion Examples:
  ecstatic > joyous > happy > contented

To obtain complete, accurate and actionable information from a piece of text, it's important to not only identify each of these five elements individually but to also understand how they work together to provide the full context and sentiment.

However, challenges faced in sarcasm detection are far more complex. The current approaches, types and features used for sarcasm detection also encounter issues that are handled in different ways. In this section we focus on several important issues such as issues with data, issues with features, issues with classification techniques.[13]

i. **Issues with Data:**
   a. Although labeled sentences with hashtag provide clear revelation of data, the quality of dataset may become ambiguous and doubtful. For example: 'I love bland food. #not'. Sarcasm is expressed through #not. If we remove #not and consider only 'I love bland food', then it may not have sarcastic interpretation.

ii. **Issues with Features:**
   a. Sarcastic sentences deceive the sentiment classifier and hence accuracy to classify the text may be reduced. Sentiment can be used as feature for classifier and it requires ground polarity of sentence. Therefore, new features should be explored and used with the combination of existing features for better accuracy.

iii. **Issues with Classification Techniques:**
   a. Sometimes researchers use small set of data, sometimes large. But it is not necessary that the dataset is fairly distributed for classification, which makes dataset balanced and imbalanced. So, the accurate classification technique should be applied on dataset for accurate classification of sentences into sarcastic and non-sarcastic.

iv. **Difficulty in Sarcasm Detection from Text:**
   a. In spoken interaction, sarcasm can be recognized using facial expression whereas in written communication, there is no facial expression in text. Therefore, detection of sarcasm from text is challenging and requires much deeper study. 2044 This full-text paper was peer-reviewed and accepted to be presented at the IEEE WiSPNET 2017 conference.

v. **Negative Sentiment Using Positive Words:**
   a. Sarcastic sentences express a negative opinion using only positive words or intensified positive words. So, a simple bag-of-words cannot be used for Sentiment Analysis on such sentences. They require additional features such as features related to author, audience, semantic, etc.

vi. **Use of Short Text:**
   a. The detection of sarcasm in short and noisy contextless text becomes very

challenging as they do not provide more features.

vii. **Integration of World Knowledge:**

a. Integration of world knowledge is required in some cases and it is itself a big task. For example, 'Love the cover (book, amazon)'. If we consider the expression "do not judge a book by its cover", we realize that it is actually a sarcastic sentence

3.6 Timeline

This project is estimated to be complete in a period of around 8 months. The following Grant Chart will show the timeline for each phases of the project. There are nine phases in this project and take different duration to complete. Requirement analysis will take around30 days to complete. Planning and prototyping will take 10 days and 20 days respectively. Then implementation will take 80 days and, training and testing are estimated to take 30 days each. Debug and review will be done for around 20 days each. At the end, we will look for opportunities for enhancement.

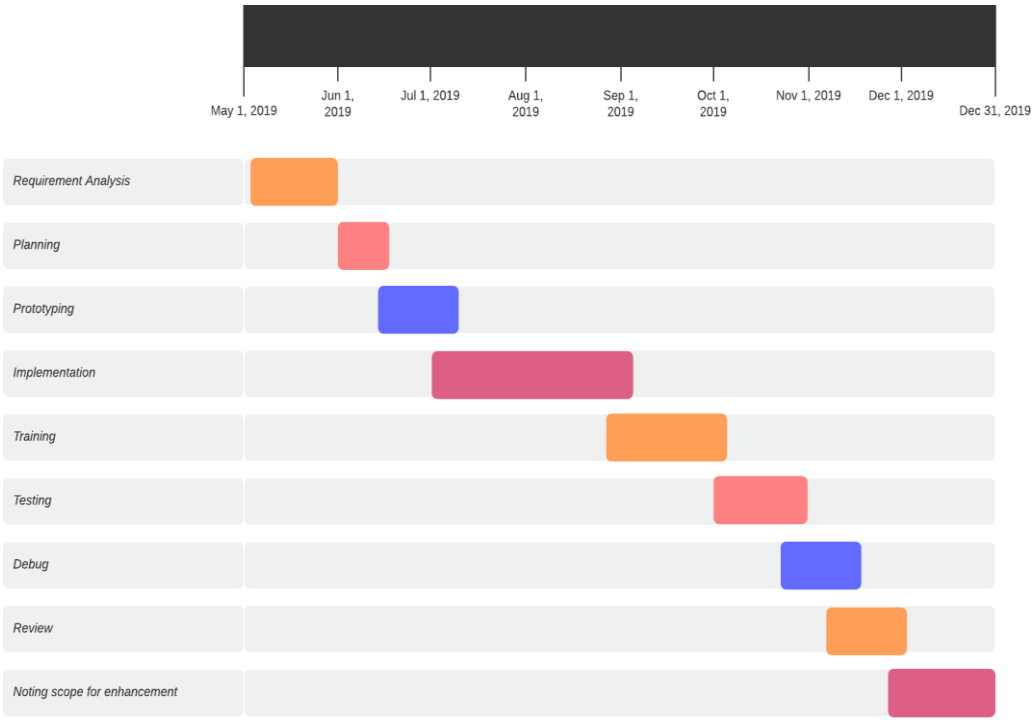


Table-3.4: Gantt chart of system Implementation

3.7 Requirement Specification

Requirement specification of the project is given below:

Data Collection: The raw IMDb dataset is structured in such a way that most of its attributes and information is organized and stored separately in compressed plain text files. For instance, all of the roughly 75,000 movie ratings from the dataset is stored in

the csv file ratings for sentiment analysis and 50000 movie ratings are stored in a separate csv file for sarcasm detection. List (e.g. ratings.list.gz), which includes textual information about the data as well as a table of film rank, the number of votes and film titles. Thus, some sort of cleaning, integration and preprocessing is likely to be required in order to make good use of the data for the purpose of data mining through supervised machine learning techniques.

Data Analysis: The dataset is divided into training dataset and test dataset which contains the classes like Hit, Flop and Average and predicting variables like actor, actress, composer, genre, director producer and music director k-means clustering is used to analyze the training dataset to develop models which can be used for test dataset for analysis decision tree algorithm is used for predicting which factors.

Project Output: Project output should be denoted by the accuracy of the models which will be the main objective of this study.

Polarity Scale: Polarity of the reviews are classified into three categories: positive, negative and neutral. If the polarity value is 1, it will be treated as positive review, if the polarity value is 0, it will be treated as negative review and if the polarity value is 2 then it will be treated as neutral review.

Irony Scale: Irony /Sarcasm of the reviews are classified into two categories: sarcastic and non-sarcastic. If the value of the sentence is 1, it will be treated as a sarcastic review. And if the value of the sentence is 0, it will be treated as non-sarcastic review.

CHAPTER 4

SYSTEM IMPLEMENTATION

4.1 System Implementation

4.1.1 Data Acquisition

We have collected IMDb movie reviews and from other available sources. Then we have prepared two datasets. One of them is for sentiment analysis and the other one is for sarcasm detection. Both datasets consist of two columns: reviews and labels. The reviews are not preprocessed or cleaned.[15]

	review	sentiment
0	One of the other reviewers has mentioned that ...	1
1	A wonderful little production. The...	1
2	I thought this was a wonderful way to spend ti...	1
3	Basically there's a family where a little boy ...	0
4	Petter Mattei's "Love in the Time of Money" is...	1
5	Probably my all-time favorite movie, a story o...	1
6	I sure would like to see a resurrection of a u...	1
7	This show was an amazing, fresh & innovative i...	0
8	Encouraged by the positive comments about this...	0
9	If you like original gut wrenching laughter yo...	1
10	Phil the Alien is one of those quirky films wh...	0
11	I saw this movie when I was about 12 when it c...	0
12	So im not a big fan of Boll's work but then ag...	0
13	The cast played Shakespeare. Shakes...	0
14	This a fantastic movie of three prisoners who ...	1

Fig-4.1.1(a): Sentiment Analysis Dataset

In the sentiment analysis dataset, reviews are labeled as positive (1), negative (0) and neutral (2).

And in the sarcasm detection dataset, reviews are labeled as sarcastic (1) and non-sarcastic (0).

	comment	label
0	"Administer of the pain, just finished huffing...	0
1	"Admits", Stiglitz is a very well known Euro c...	0
2	"adopt(ing) private insurance" is not a fair w...	0
3	"Adopted".	1
4	"Adorable"	0
5	"advancing"	0
6	"Aegis sov is such a success, the player engag...	1
7	"AFAIK" - that covers your back no end .... fo...	0
8	"Afk for a bit, I need to go iron my dog"	0
9	"Africa is the only country".. What?	0
10	"African American"	1
11	"Afro," Walter?	0
12	"after he was released from jail for IDENTICAL...	1
13	"After nine years in development, hopefully it...	0
14	"after our species trekked out of Africa to po...	0

Fig-4.1.1(b): Sentiment Analysis Dataset

4.1.2 Data Analyzing

In short, data analysis is the automated process that allows machines to extract and classify information from data or text. Here, we try to count several parameters of the dataset. Row count, Unique values, Top value in a column, frequency of that top value etc.

	review	sentiment
count	75001	75001
unique	74556	3
top	#NAME?	2
freq	28	25001

Fig-4.1.2: Description of Sentiment Analysis Dataset

4.1.3 Natural language Processing (StopWords)

In computing, **stop words** are words which are filtered out before processing of natural language data (text). Stop words are generally the most common words in a language; there is no single universal list of stop words used by all-natural language processing tools, and indeed not all tools even use such a list. Some tools avoid removing stop words to support phrase search.[4]

Any set of words can be chosen as the stop words for a given purpose. For some search engines, these are some of the most common, short function words, such as the, is, at, which, and on. In this case, stop words can cause problems when searching for phrases that include them, particularly in names such as "The Who", "The The", or "Take That". Other search engines remove some of the most common words— including lexical words, such as "want"—from a query in order to improve performance.

As we have used nltk stopwords , let us see the words included in nltk.corpus library.

4.1.4 Data Cleaning

Text is the form of data which has existed for millenniums throughout the human history. All the sacred texts influencing all the religions, all the compositions of poets and authors, all the scientific explanations by the brightest minds of their times, all the political documents which define our history and our future, and all kind of explicit human communication, these "All"s define the importance of data available in the form of what we call text.[15]
Text is just a sequence of characters. But when we usually deal with natural language processing, we are more concerned about the words as a whole, instead of just worrying about character-level depth of our text data. One reason behind that is, that in the natural language processing, individual characters don't have a lot of "context". Characters like 'b', 'a', 'l' don't hold any context individually, but when rearranged in the form of a word, they might generate the word "ball", which might explain a object we often use to play some sort of sports. We need to follow some steps to get our data in a desired format.

4.1.4.1 Removing unwanted characters

The is a primary step in the process of text cleaning. When we collect data from different source on internet, they contain lot of tags, HTML entities, punctuation, non-alphabets, and any other kind of characters which might not be a part of the language. We need to get rid of all them. The general methods of such cleaning involve regular expressions, which can be used to filter out most of the unwanted texts.

There are some systems where important English characters like the full-stops, question-marks, exclamation symbols, etc are retained. Consider an example where you want to perform some sentiment analysis on human generated tweets, and you want to classify the tweets are very angry, angry, neutral, happy, and very happy. Simple sentiment analysis might find it hard to differentiate between a happy, and a very happy sentiment, because there can be some moments only words are not able to explain.

4.1.4.2 Tokenization

Tokenization is just theprocess of splitting a sentence into words. So according to the need of the model we need to process the text further. That is why we need to tokenize the sentences.

4.1.4.3 De-capitalization

Capitalization and De-capitalization is again, dependent on what the application is going to be. If we're only concerned with the terms, and not their "intensities of presence", then all the terms with lowercase should do fine, but if we want to differentiate between any sentiments, then something written in uppercase might mean something different than something written in lowercase.

For our model we don't need to be concerned with the intensities of presence of the words so we have de-capitalized the words.

4.1.4.4 Removing Punctuation, Numbers

This step can vary for model to model. It totally depends on what kind of analysis one is trying to perform. Below is the reasoning of performing this step for our model:

Getting rid of the punctuation characters, because they behave as the noise inside the text data because they hold no specific semantic meaning.

Getting rid of the numbers and numerals, because we want to perform a qualitative analysis (positive, negative or neutral) instead of any sort of quantitative analysis involving numbers.

Removing all the words with less than or equal to three characters, because such words can either be stop words, or they can be the words acting as slang terms.

4.1.4.5 Removing Stopwords

Removing stopwords like and, or, of, the, is, am, had, etc. Stopwords are so common, and hold so little semantic information, that their removal is favorable because it not only reduces the dimensionality of the vector-space model, but also increases accuracy in classification in most cases. There are two common approaches of removing the stopwords, and both are fairly straightforward. One way is to count all the word occurrences, and providing a threshold value on the count, and getting rid of all the terms/words occurring more than the specified threshold value.

The other way is to have a predetermined list of stopwords, which can be removed from the list of tokens/tokenized sentences. Some human expressions, like "hahaha, lol, lmfao, brb, wtf" can also be a valuable information when working on systems based on semantic/sentiment analysis, but for the systems requiring a more formal kind of an application, these expressions might also get removed.

This cleaning step also depends on what we will eventually be doing with the data after preprocessing.

4.1.4.6 De-tokenization

This process is exactly the opposite of the tokenization. After cleaning the sentences, we try to join the words and restore them in form of sentences.

4.1.5 Lemmatization

Lemmatization is one important phase, because it reduces the words or terms to their roots. For example, the term "faster" gets converted into "fast", "recommended" to "recommend", "eating" to "eat", etc. This helps in retaining the semantic meaning of the sentence while simplifying the repetition. By definition it is a Text Normalization (or sometimes called Word Normalization) techniques in the field of Natural Language Processing that are used to prepare text, words, and documents for further processing. There is another process which performs the same task named stemming. The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form.
However, the two words differ in their flavor. Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes. Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma.
As lemmatization prefers to do thing properly, so in our project we will be using lemmatization.

After completing all the steps of data cleaning and lemmatization, we get data in our desired form.

Index	Type	Size	Value
0	str	1	one reviewer mentioned watching oz episode hooked right exactly happen ...
1	str	1	wonderful little production br br filming technique unassuming old tim ...
2	str	1	thought wonderful way spend time hot summer weekend sitting air condit ...
3	str	1	basically family little boy jake think zombie closet parent fighting t ...
4	str	1	petter mattei love time money visually stunning film watch mr mattei o ...
5	str	1	probably time favorite movie story selflessness sacrifice dedication n ...
6	str	1	sure would like see resurrection dated seahunt series tech today would ...
7	str	1	show amazing fresh innovative idea first aired first year brilliant th ...
8	str	1	encouraged positive comment film looking forward watching film bad mis ...
9	str	1	like original gut wrenching laughter like movie young old love movie h ...
10	str	1	phil alien one quirky film humour based around oddness everything rath ...

Fig-4.1.4: After Data Cleaning and Lemmatization

4.1.6 Vectorization

Vectorization is just a method to convert words into long lists of numbers, which might hold some sort of complex structuring, only to be understood by a computer using some sort of machine learning, or data mining algorithm. It is an important decision, because converting English words into vectors in such a way that their semantics can be represented through numerical vectors of very high dimensions is somewhat a tricky process. For our specific problem, we had the options to use TF-IDF features. These kinds of features are preferred when we are going to develop some information retrieval systems, where not only the existence, but also the number of times a term exists, and the number of documents in which the term (word) exists also matters. TF-IDF features basically provide higher values for the rare words, and lesser weightage values for the common terms. The features we'll be calculating are called the BoW (Bag of Word) features, and the idea behind them is that we consider a document (in our case, a single review) as a collection, or say, a "Bag" of different terms (words). Either a word is present inside a document, or not. The entire concept of BoW is based on very large binary vectors. Each word represents a unique dimension, therefore if there are say, W unique words in the entire collection of D number of documents, then each document will be represented as a W-Dimensional vector. Consider a vocabulary of four words, say "alpha, beta, charlie, and delta". Now, each word, or term will have an index assigned to it. Say that these indexes are assigned using an alphabetically sorted order. In this case, the term "alpha" will be on index 0, "beta" on index 1, and so on. Now, if we want to represent a document which says "alpha beta delta delta", then it can be represented as a 4- Dimensional vector as [1, 1, 0, 1].

4.1.7 Sentiment Analysis

Sentiment analysis as the name suggests is all about emotions. If the customer likes a product or not; if the customer likes a specific service or not. On a whole it's all about the customers opinions about a product or a service.[18]
Theoretically and technically sentiment analysis refers to the use of natural language processing, text analysis, computational linguistics, and bio-metrics to systematically identify, extract, quantify, and study effective states and subjective information. Sentiment analysis is widely applied to voice of the customer materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine.
The process of extracting insights from reviews, opinions, survey's is widely adopted by many organizations across the world. Some of the tasks it's used for are:

1. How good a movie is?
2. How is a new product doing?
3. What is going to be the result of an upcoming election?
4. What opinions do people have/ what side the people are going to pick on a trending issue or a political issue?

When people comment, review or provide a feedback, the information in the piece of text is invaluable. We use that invaluable information, and tell out if the comment, review or a feed back is positive or negative.

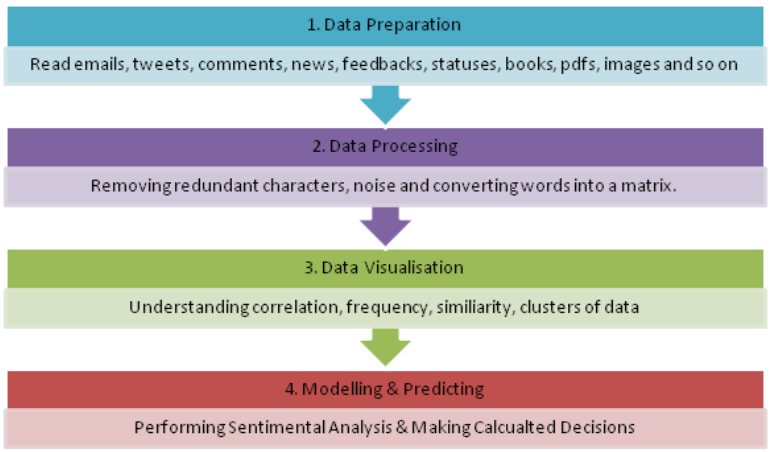


Fig-4.1.7(a): Process

After preprocessing, lemmatization and vectorization, we split both the dataset for training and testing the classifier. The ratio we have used to split the dataset is to 80/20.

We feed the classifier with respective dataset and train them. It is a time-consuming process

as we have a large number of datasets.[13][16]

After training is completed, we test both classifiers with remaining 20% of the data. Accuracy of the sentiment analysis model is 86%.

```
accuracy_score(y_test,sentiment)
```

```
0.864875674955003
```

Fig-4.1.7(b): Accuracy Result of Sentiment

4.1.7.1 Observation

We can see that the accuracy of the models falls marginally when it is combined with the sarcasm detection model.

By that we can conclude that the accuracy of sarcasm detection wasn't really up to the mark.

It is to be noted that while detecting sarcasm we didn't really played attention to the context and structure of the sentences which resulted in low accuracy of the model.

4.1.8 Sarcasm Detection

Sarcasm**,** which is both positively funny and negatively nasty, plays an important part in human social interaction.[19]
But sometimes it is difficult to detect whether someone is making fun of us with some irony. So to make it easy we built something which helps us in detecting sarcastic text.

After preprocessing, lemmatization and vectorization, we split both the dataset for training and testing the classifier. The ratio we have used to split the dataset is to 80/20.

We feed the classifier with respective dataset and train them. It is a time-consuming process as we have a large number of datasets.[13][16]

After training is completed, we test both classifiers with remaining 20% of the data. Accuracy of the sarcasm detection model is 76%.

```
accuracy_score(y_test_s,sarcasm)
```

```
0.7635716959940098
```

Fig-4.1.8: Accuracy Result of Sarcasm Detection Model

CHAPTER 5

RESULT ANALYSIS

5.1 Result Analysis

In the sentiment analysis and sarcasm detection step we have passed the vectorized values of the sentences with labels describing their polarity and irony respectively.

The polarity classes for sentiment analysis was 1,0,2. If the polarity was 1, then the sentence was positive. it the polarity was 0, then the sentence was negative and finally if the polarity was 2, then the sentence was neutral.

- Positive (1)
- Negative (0)
- Neutral (2)

Similarly, the categories for sarcasm detection was 1, 0. If the irony value was 1, then the sentence was sarcastic and if the irony value was 0, then the sentence was non-sarcastic.

- Sarcastic (1)
- Non-sarcastic (0)

CHAPTER 6

CONCLUSION & FUTURE ENHANCEMENT

6.1 Conclusion

In this paper, we have proposed a analysis system containing multiple models which can help companies, producers and regular users of "IMDb" website. This can be widely used in the future to get understanding of sentiments of viewers of a particular movie. This project can help both producers and viewers. The producers will be able to understand the business aspect of their previous movies or movies of their contemporaries. And take decisions which will help them enhance their business. At the same time, it will help the viewers to get a understanding of the movies based on their reviews and choose the best one depending on the requirement of the user.

Tis project is reliable as the rating is based on customer opinion and this rating will reflect the actual state of the product and services provided by "IMDb website". Also, it will help companies to provide with their product or service to the viewers.

6.2 Future Enhancement

We have implemented this project on IMDb movie reviews. Our initial plan was to implement the project for classifying the user reviews based on their polarity. To get the correct polarity result possible we have also implemented sarcasm detection. In future we wish to improve the sarcasm detection model that can detect irony in a sentence based on the context and the structure of the sentence. We also want to integrate this to a website or software so that it becomes more user friendly.

We look forward to ease the process of decision making for both makers and viewers of a movie or series. This project would be just a few steps towards digitalization. We belief it will allow everyone to see an accurate review of any movie or series.

REFERENCES

[1] https://searchbusinessanalytics.techtarget.com/definition/natural-language-processing-NLP

[2] https://www.nltk.org/

[3] https://en.wikipedia.org/wiki/Stop_words

[4] https://pythonspot.com/nltk-stop-words/

[5] https://en.wikipedia.org/wiki/Lemmatisation

[6] https://en.wikipedia.org/wiki/Scikit-learn

[7] https://en.wikipedia.org/wiki/Python_(programming_language)

[8] https://en.wikipedia.org/wiki/Matplotlib

[9] https://en.wikipedia.org/wiki/Pandas_(software)

[10] https://en.wikipedia.org/wiki/Regular_expression

[11] https://www.nltk.org/

[12] S.K. Bharti et al., "Sarcastic sentiment detection in tweets streamed in real time: a big data approach", Digital Communications and Networks, vol. 2, no. 3, pp. 108-121, 2016.

[13] Mondher Bouazizi, Tomoaki Otsuki, "A pattern-based approach for sarcasm detection on twitter", IEEE Transl., 2016.

[14] S. Rossano, J. Paloma, T. Joel, "Detecting sarcasm in multimodal social platforms", Proceedings of the 2016 ACM on Multimedia Conference, 2016.

[15] Yuji Roh, Geon Heo, Steven Euijong Whang,"A Survey on Data Collection for Machine Learning", IEEE, 2019

[16] Shahnawaz,Parmanand Astya,"Sentiment analysis: Approaches and open issues", ICCCA,2017

[17] https://www.slideshare.net/japerk/nltk-the-good-the-bad-the-awesome-85569

[18] https://medium.com/analytics-vidhya/sentiment-analysis-using-nltk-d520f043fc0

[19] https://towardsdatascience.com/sarcasm-detection-step-towards-sentiment-analysis-84cb013bb6db