LONDON METROPOLITAN UNIVERSITY

Islington college
(इस्लिङ्टन कलेज)

FINAL YEAR PROJECT PROPOSAL

Sentiment Analysis of Hotel Reviews mined from Trip Advisor

2019-20 Autumn

Student Name	**:**	**Riya Shakya**
London Met ID	**:**	**17031225**
College ID	**:**	**np01cp4a170134**
Assignment Due Date	**:**	**January 7th, 2020**
Assignment Submission Date	**:**	**January 7th, 2020**
Word Count	**:**	**5920**
External Supervisor	**:**	**Ishwor Shrestha**
Internal Supervisor	**:**	**Subeksha Shrestha**

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Abstract

Recent studies have shown that Sentiment Analysis of reviews are being implemented across the globe as one of the few arising AI. However, even though being widely used worldwide the result of the Sentiment Analysis can be taken as final statement without the supervision of human as it is not yet accurate due to the use of sarcasm in reviews which is yet to be understood by the machine. It being a part of AI makes more accurate with practice. This report is about Sentiment Analysis on data mined from Trip advisor. The report covers over the introduction, background, development till day, progress obtained future work.

The report contains chapters, among which Introduction which gives brief summary of the project, Back ground gives conceptional understand the person related to their project topic, Development till date includes all the progress you have made in the project till date, Analyse of progress checks the process made by an individual and compare it to the Gantt chart, and Future work include all the work that need to be done for accomplish the goal.

Acknowledgement

I would like to thank my supervisor, Mr. Ishwor Shrestha as well as the second supervisor of the project, Mr. Subeksha Shrestha for guiding me through the project, pushing me every day to reach new limits of the project and helping me take important decisions along the way. I would also like to express my sincere gratitude towards London Metropolitan University for providing me with a platform to work on such project.

Table of Abbreviation

S. No.	Abbreviation	Full Forms
1	IoT	Internet of Things
2	ML	Machine Learning
3	AI	Artificial Learning
4	NLP	Natural Language Processing
5	NLTK	Natural Language Toolkit

Table of Contents

List of Figures:

List of Tables:

Chapter 1. Introduction

1.1 Preface

Berner-Lee's vision unexpectedly evolved into an explosion of data with the thinking of world beyond World Wide Web, Internet has spread world-wide over the years. It has connected the people around the globe, making it easier for people to open and freely share information and ideas. The data created by Internet users was forecasted 2.5 quintillion bytes by Forbes with the expectation of new connected device to tens of millions (Ash, 2020). The data can be used to provide real-time information by tracking materials, equipment and products through every step of decision making for manufacture companies to engage their customer and battle their competitors.

The data storage reached to a limit where storage of a single computer was not enough. The large dataset gathers from internet users gave birth to the concept of Big Data. It is a category of computing strategies and technologies that are used to handle large datasets (Ellingwood, 2016). The goal of big data system is to provide insights and connections from large volumes of data which is not possible using conventional methods. The big data processing categories are ingesting data into the system, persisting the data in storage, computing and visualizing the result. The Machine Learning is the process in which machine uses the data from big data. Big data provides different variety of data to machines to improve their performance in the task. A wide range of data gives more accurate result.

Machine Learning(ML) was created with the primary aim to allow the computers to learn automatically without human intervention or assistance and adjust actions accordingly (Expert Systems, 2017). There are different machine learning algorithms which are categorized as supervised and unsupervised. Artificial Intelligence(AI) is the study of how to train the computers so that computer can do things which at present human can do better (Anusha_Sharma, 2020). AI is applied based on machine learning as the machines are trained using machine learning algorithm. AI has multiple elements including ML, Natural Language Processing (NLP), Expert System, Speech Recognition, Planning, Robotics and Vision. Among the different element, NLP is one the rising part of AI and one of the difficult to achieve accurate result.

The sole purpose of NLP is to analyse text, it is used to retrieve information, extract information, machine translation, text simplification, sentiment analysis, text summarization, spam filter, auto-predict, auto correct, speech recognition, text generation, question answering, Classification, and Natural Language Generation (NLG) (Shetty, 2018). Sentiment Analysis deals with analysing text data and classifying opinion as negative, positive or neutral. It tracks the mood of the public about a product or topic. (G.Vinodhini*, 2012) It is also called opinion mining; it involves in building a system to collect and examine the comments of the users posted in tweets, blog post and reviews. It can be used to judge the success of a campaign, launch of new product, popularity of the product and with the likes and dislike of users along with the knowledge of the issues faced by the users in any products.

The emergence of Sentiment analysis was back in 1990's but only emerged in 2004 due to information management (G.Vinodhini*, 2012). The requirement for Sentiment Analysis is data source, data source can be blogs, reviews sites, dataset or microblogging such as tweets, sentiment classification which has machine learning, semantic orientation, role of negation, and feature based sentiment classification and where evaluation can be done by calculating various metrics like precision, recall and F-measure. With the marketing world controlled by internet, it is a necessity for business to keep in track of the reviews of the customers online. The increasing number of internet users makes it hard to manually go through the reviews. So, this report is about a system which present the sentiment report of the reviews of the customer of hotel in TripAdvisor. The system with list of the positive reviews and negative reviews percentage wiil be displayed to make it easier to have an overview of the reviews from the customer.

1.2 Present scenario and problem statement

Hotels are an imperative part of the tourism industry. The success of a hospitality business like a hotel depends on the client whether the client enjoys the hotel's services or not. The consumers' opinions on hotel facilities can be known through customer reviews. Trip Advisor is one of the largest online travel review sites with a strong network effect, where guest reviews can be found. It influences 40-50% of all online travel with an annual growth rate of 43% (Advisor, 2019). The reviews of Trip Advisor greatly impact the business as it is the foundation to rating, influence booking, and evolvement of business all over. Reviews provide a strong value for the hospitality business as more reviews is more engagement and the mistake made by many hospitality businesses is not actively collecting guest reviews on sites like Trip Advisor. Trip Advisor has more than 280 traveller reviews and opinion submitted to the site per minute which makes it difficult to go through the hotel reviews manually (Advisor, 2019). Therefore, it is hard for hotel business owner to determine what customer loves and hates based on huge data of reviews. (Reference: Appendix-B)

1.3 Project as a solutions

A web application will be developed to tackle the problem mentioned above. The application will have an input area where the user can input Trip Advisor's hotel link. The web application will analyse the reviews of Trip Advisor and give proper sentiment analysis of reviews in form of textual and virtualization. The website will help the company to know whether the facility is getting good feedbacks from the consumers or there is a weakness that needs to change or perhaps marketing policy is not practical and many other factors.

1.4 Project Aims and Objectives

The aims of this project are listed below:

• To gain knowledge about Natural Language Processing (NLP) and Machine Learning (ML) to create a web application.

• The NLP will be an effective tool to build a foundation for Part of Speech (POS) tagging and sentiment analysis.

• ML techniques will help to solve complex natural language processing tasks, such as understanding double meaning through automated training.

• The reviews of the customers are mainly unstructured which are difficult, time-consuming and expensive to analyse, understand and sort through but this project aims to make sense of the unstructured text by automating business processes, getting actionable insights, and saving hours of manual data.

The objective of the project is

• To develop a web application which will reflect the learning outcome of NLP and ML.

• To give a textual and virtual presentation of the reviews of Trip Advisor through the application which will give the hotel owner a clear idea of the customer review.

• The project will deliver a structured review of the clients to the hospitality business for the improvement of the hotel

1.5 Project Feature

- The system will give a visual representation of the overview of the reviews in a chart.
- It will list the features which are frequently appeared in the reviews.
- It will list the some of the reviews.
- It will store the result for 3 months of period.

1.6 Structure of report

Chapter 2 Background / Literature review

The main theme of this chapter is to introduce the subject of the report. It includes the similar system to this project with brief review on the libraries used to build the software and their knowledge gain through studying their system. It is also including the system architecture along with a brief description about the project.

Chapter 3 Development till date

The main theme of this chapter is to list of the achievements till date including Gantt chart, ERD, wireframe, use case, high-level use case and expanded use case along with the development of code.

Chapter 4 Analysis of progress

The main theme of this chapter is to list out the task which were supposed to be completed but due to some reason could not be completed.

Chapter 5 Future Work

The main theme of this chapter is to list out the task left according to Gantt chart soon.

Chapter 2. Background

2.1 Similar System Comparison and Analysis

2.1.1 Twitter Sentiment Analysis using Python

In this system, the sentiment analysis is done using libraries: Tweepy, Textblob and corpora of NLTK (Kumar, 2020). The library tweepy is open-sourced, hosted by GitHub which enables Python to communicate with Twitter platform and use its API (Novalić, 2013). TextBlob is a library for textual data which provides a simple API for diving into common NLP task such as sentiment analysis, classification, etc (TextBlob, 2020). The nltk.corpus package is a collection of corpus reader classes which is used to access the contents of a diverse set of corpora (nltk, Corpus Readers, 2020) .

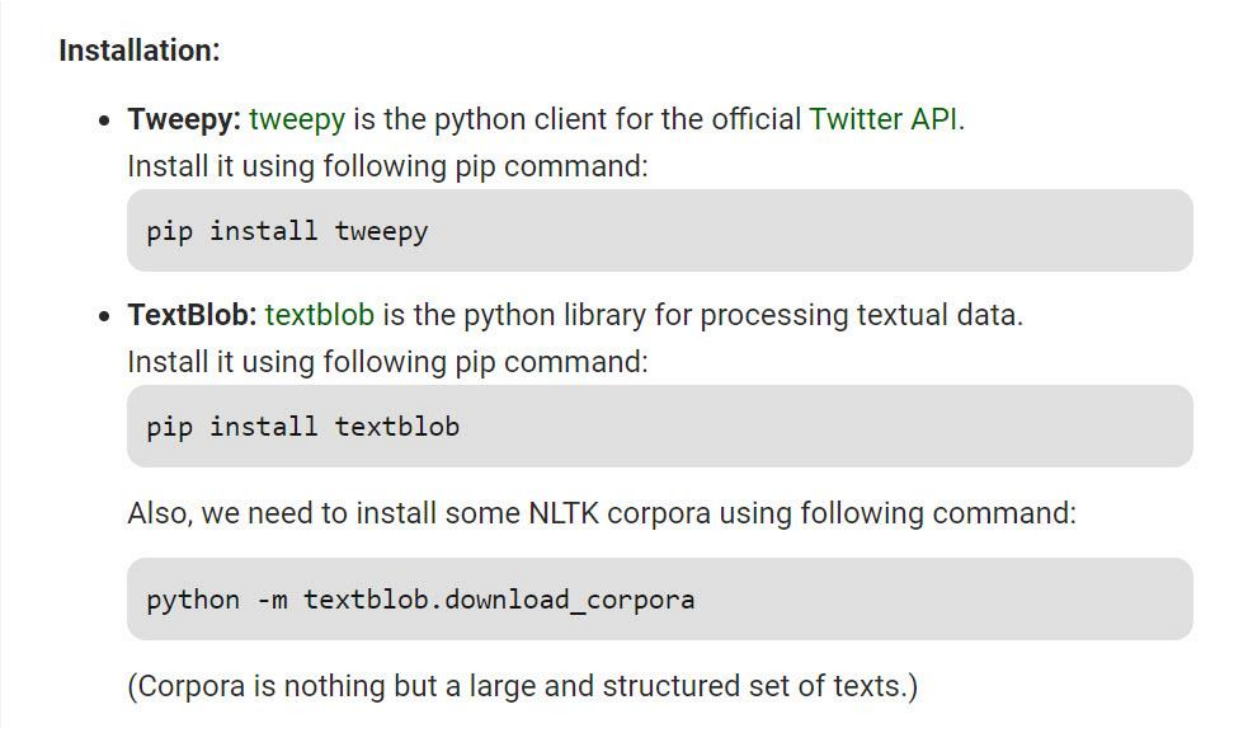


Figure 1. Twitter Sentiment Analysis using Python (a)

```python
import re
import tweepy
from tweepy import OAuthHandler
from textblob import TextBlob

class TwitterClient(object):
    '''
    Generic Twitter Class for sentiment analysis.
    '''
    def __init__(self):
        '''
        Class constructor or initialization method.
        '''
        # keys and tokens from the Twitter Dev Console
        consumer_key = 'XXXXXXXXXXXXXXXXXXXXXXXX'
        consumer_secret = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
        access_token = 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
        access_token_secret = 'XXXXXXXXXXXXXXXXXXXXXXXXX'

        # attempt authentication
        try:
            # create OAuthHandler object
            self.auth = OAuthHandler(consumer_key, consumer_secret)
            # set access token and secret
            self.auth.set_access_token(access_token, access_token_se
            # create tweepy API object to fetch tweets
            self.api = tweepy.API(self.auth)
        except:
            print("Error: Authentication Failed")

    def clean_tweet(self, tweet):
        '''
        Utility function to clean tweet text by removing links, spec
        using simple regex statements.
        '''
        return ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t])
                                    |(\w+:\/\/\S+)", " ", tweet).spl

    def get_tweet_sentiment(self, tweet):
        '''
        Utility function to classify sentiment of passed tweet
        using textblob's sentiment method
        '''
        # create TextBlob object of passed tweet text
        analysis = TextBlob(self.clean_tweet(tweet))
        # set sentiment
        if analysis.sentiment.polarity > 0:
            return 'positive'
        elif analysis.sentiment.polarity == 0:
            return 'neutral'
        else:
            return 'negative'

    def get_tweets(self, query, count = 10):
        '''
        Main function to fetch tweets and parse them.
        '''
        # empty list to store parsed tweets
        tweets = []

        try:
```

Figure 2. Twitter Sentiment Analysis using Python (b)

```
            tweets = []

            try:
                # call twitter api to fetch tweets
                fetched_tweets = self.api.search(q = query, count = coun

                # parsing tweets one by one
                for tweet in fetched_tweets:
                    # empty dictionary to store required params of a twe
                    parsed_tweet = {}

                    # saving text of tweet
                    parsed_tweet['text'] = tweet.text
                    # saving sentiment of tweet
                    parsed_tweet['sentiment'] = self.get_tweet_sentiment

                    # appending parsed tweet to tweets list
                    if tweet.retweet_count > 0:
                        # if tweet has retweets, ensure that it is appen
                        if parsed_tweet not in tweets:
                            tweets.append(parsed_tweet)
                    else:
                        tweets.append(parsed_tweet)

                # return parsed tweets
                return tweets

            except tweepy.TweepError as e:
                # print error (if any)
                print("Error : " + str(e))

def main():
    # creating object of TwitterClient Class
    api = TwitterClient()
    # calling function to get tweets
    tweets = api.get_tweets(query = 'Donald Trump', count = 200)

    # picking positive tweets from tweets
    ptweets = [tweet for tweet in tweets if tweet['sentiment'] == 'p
    # percentage of positive tweets
    print("Positive tweets percentage: {} %".format(100*len(ptweets),
    # picking negative tweets from tweets
    ntweets = [tweet for tweet in tweets if tweet['sentiment'] == 'n
    # percentage of negative tweets
    print("Negative tweets percentage: {} %".format(100*len(ntweets),
    # percentage of neutral tweets
    print("Neutral tweets percentage: {} % \
        ".format(100*len(tweets - ntweets - ptweets)/len(tweets)))

    # printing first 5 positive tweets
    print("\n\nPositive tweets:")
    for tweet in ptweets[:10]:
        print(tweet['text'])

    # printing first 5 negative tweets
    print("\n\nNegative tweets:")
    for tweet in ntweets[:10]:
        print(tweet['text'])

if __name__ == "__main__":
    # calling main function
    main()
```

Figure 3. Twitter Sentiment Analysis using Python (c)

2.1.2 Sentiment Analysis with Python using MonkeyLearn API

In this system, Sentiment Analysis is done using API of mokeylearn. The monkey learn provides the facility to train and create your own model with the option to choose the classification. It lets you choose the type of file to import, selection of tag, testing of model along with improvement of model (Stecanella, 2019).

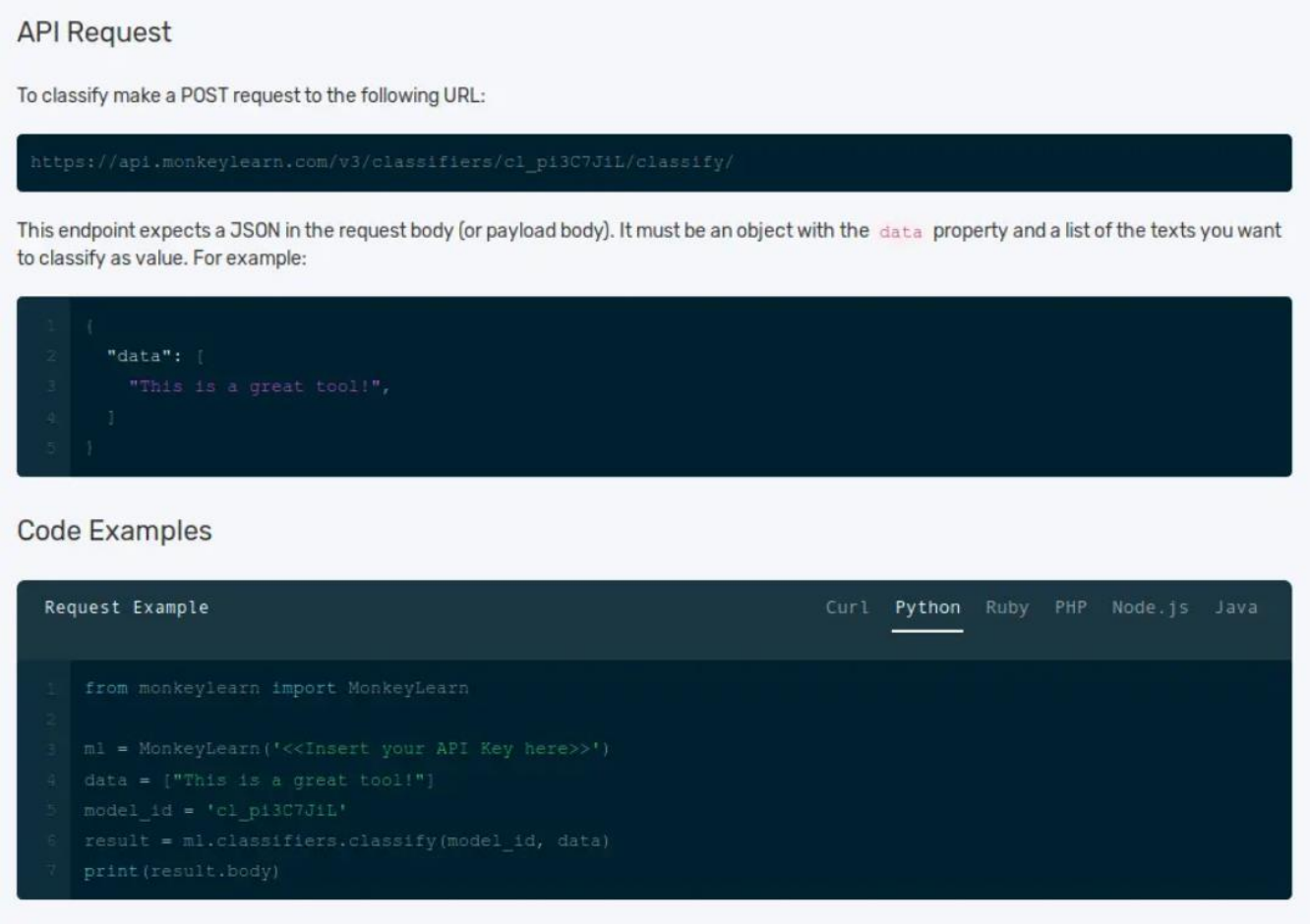


Figure 4. Sentiment Analysis with Python using MonkeyLearn API (a)

```
pip install monkeylearn
```

You can also clone the repository and run the setup.py script:

```
$ git clone git@github.com:monkeylearn/monkeylearn-python.git
$ cd monkylearn-python
$ python setup.py install
```

And that's it for setup.

You're ready to run a sentiment analysis on your texts with the following code:

```
from monkeylearn import MonkeyLearn

ml = MonkeyLearn('<<Your API key here>>')
data = ['The restaurant was great!', 'The curtains were disgusting']
model_id = 'cl_pi3C7JiL'
result = ml.classifiers.classify(model_id, data)

print(result.body)
```

The output will be a Python dict generated from the JSON sent by MonkeyLearn, and should look something like this:

```
[{
    'text': 'The restaurant was great!',
    'classifications': [{
        'tag_name': 'Positive',
        'confidence': 0.993,
        'tag_id': 33767179
    }],
    'error': False,
    'external_id': None
}, {
    'text': 'The curtains were disgusting',
    'classifications': [{
        'tag_name': 'Negative',
        'confidence': 0.979,
        'tag_id': 33767178
    }],
    'error': False,
    'external_id': None
}]
```

Figure 5. Sentiment Analysis with Python using MonkeyLearn API (b)

```python
from monkeylearn import MonkeyLearn

ml = MonkeyLearn('<<Your API key here>>')
data = ['The room was great!', 'The curtains were disgusting']
model_id = '<<Your model ID here>>'
result = ml.classifiers.classify(model_id, data)

print(result.body)

```

And the output for this code will be similar as well:

```
[{
    'text': 'The room was great!',
    'classifications': [{
        'tag_name': 'positive',
        'confidence': 0.836,
        'tag_id': 103237939
    }],
    'error': False,
    'external_id': None
}, {
    'text': 'The curtains were disgusting',
    'classifications': [{
        'tag_name': 'negative',
        'confidence': 0.924,
        'tag_id': 103237938
    }],
    'error': False,
    'external_id': None
}]

```

Figure 6. Sentiment Analysis with Python using MonkeyLearn API (c)

2.1.3 Sentiment Analysis in Python using NLTK

In this system, Sentiment Analysis is done using NLTK (Daityari, How To Perform Sentiment Analysis in Python 3 Using the Natural Language Toolkit (NLTK), 2019). NLTK is a leading platform with over 50 corpora and lexical resources with text processing libraries for classification, tokenization and such (nltk, Natural Language Toolkit, 2019).

```
from nltk.stem.wordnet import WordNetLemmatizer
from nltk.corpus import twitter_samples, stopwords
from nltk.tag import pos_tag
from nltk.tokenize import word_tokenize
from nltk import FreqDist, classify, NaiveBayesClassifier

import re, string, random

def remove_noise(tweet_tokens, stop_words = ()):

    cleaned_tokens = []

    for token, tag in pos_tag(tweet_tokens):
        token = re.sub('http[s]?://(?:[a-zA-Z]|[0-9]|[$-_@.&+#]|[!*\(\),]|'\
                       '(?:%[0-9a-fA-F][0-9a-fA-F]))+','', token)
        token = re.sub("(@[A-Za-z0-9_]+)","", token)

        if tag.startswith("NN"):
            pos = 'n'
        elif tag.startswith('VB'):
            pos = 'v'
        else:
            pos = 'a'

        lemmatizer = WordNetLemmatizer()
        token = lemmatizer.lemmatize(token, pos)

        if len(token) > 0 and token not in string.punctuation and token.lower() not in stop_wor
            cleaned_tokens.append(token.lower())
    return cleaned_tokens

def get_all_words(cleaned_tokens_list):
    for tokens in cleaned_tokens_list:
        for token in tokens:
            yield token

def get_tweets_for_model(cleaned_tokens_list):
    for tweet_tokens in cleaned_tokens_list:
        yield dict([token, True] for token in tweet_tokens)
```

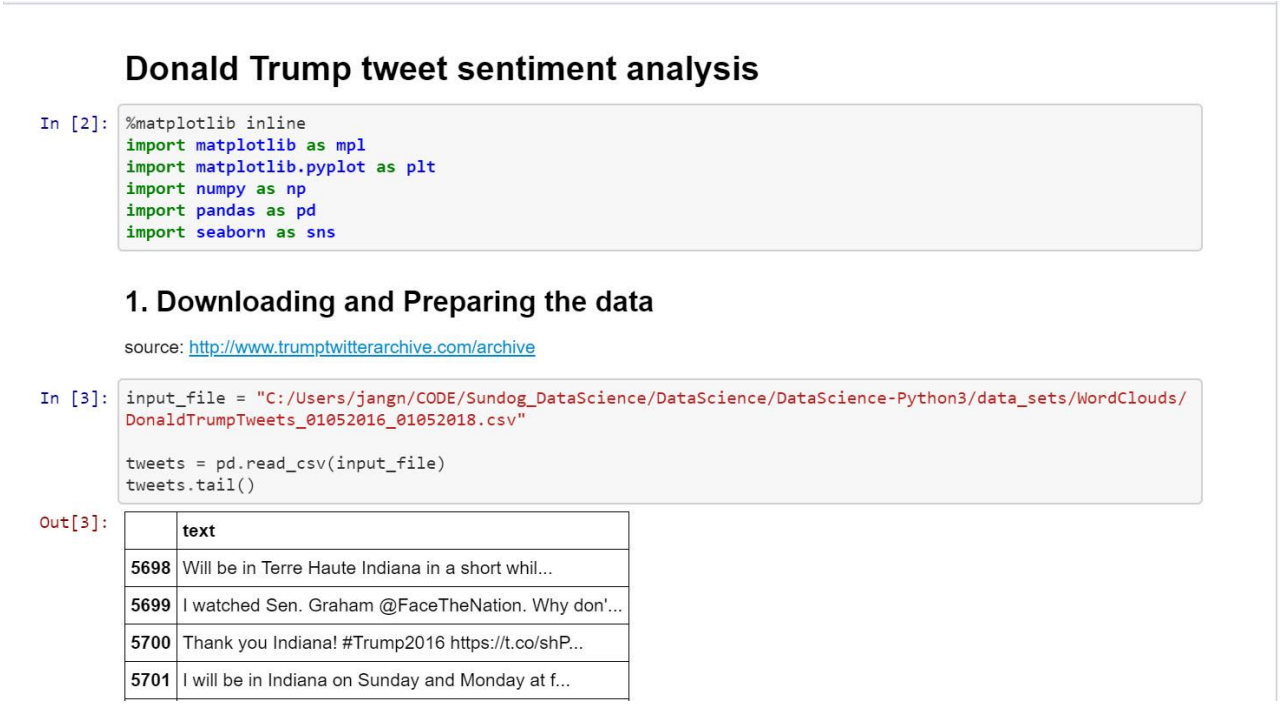
Figure 7. Sentiment Analysis in Python using NLTK (a)

```
if __name__ == "__main__":

    positive_tweets = twitter_samples.strings('positive_tweets.json')
    negative_tweets = twitter_samples.strings('negative_tweets.json')
    text = twitter_samples.strings('tweets.20150430-223406.json')
    tweet_tokens = twitter_samples.tokenized('positive_tweets.json')[0]

    stop_words = stopwords.words('english')

    positive_tweet_tokens = twitter_samples.tokenized('positive_tweets.json')
    negative_tweet_tokens = twitter_samples.tokenized('negative_tweets.json')

    positive_cleaned_tokens_list = []
    negative_cleaned_tokens_list = []

    for tokens in positive_tweet_tokens:
        positive_cleaned_tokens_list.append(remove_noise(tokens, stop_words))

    for tokens in negative_tweet_tokens:
        negative_cleaned_tokens_list.append(remove_noise(tokens, stop_words))

    all_pos_words = get_all_words(positive_cleaned_tokens_list)

    freq_dist_pos = FreqDist(all_pos_words)
    print(freq_dist_pos.most_common(10))

    positive_tokens_for_model = get_tweets_for_model(positive_cleaned_tokens_list)
    negative_tokens_for_model = get_tweets_for_model(negative_cleaned_tokens_list)

    positive_dataset = [(tweet_dict, "Positive")
                        for tweet_dict in positive_tokens_for_model]

    negative_dataset = [(tweet_dict, "Negative")
                        for tweet_dict in negative_tokens_for_model]

    dataset = positive_dataset + negative_dataset

    random.shuffle(dataset)

    train_data = dataset[:7000]
    test_data = dataset[7000:]

    classifier = NaiveBayesClassifier.train(train_data)

    print("Accuracy is:", classify.accuracy(classifier, test_data))

    print(classifier.show_most_informative_features(10))

    custom_tweet = "I ordered just once from TerribleCo, they screwed up, never used the app ag

    custom_tokens = remove_noise(word_tokenize(custom_tweet))

    print(custom_tweet, classifier.classify(dict([token, True] for token in custom_tokens)))
```

Figure 8. Sentiment Analysis in Python using NLTK (b)

2.1.4 Sentiment Analysis on Donald Trump's tweets

In this system, Sentiment Analysis is done using libraries such as matplotlib, numpy, pandas and seaborn (Nordin, 2018). The matplotlib library is a python 2D plotting library that generates quality figures for publications across platforms in different hard copy formats and interactive environments (matplotlib, 2020). The numpy library is the fundamental package for scientific computing with python containing powerful N-dimensional array object, useful linear algebra, random number capabilities and such (numpy, 2019). The pandas library is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for python (pandas, 2019). The seaborn library is a python data visualization library based on matplotlib which provides a high-level interface for drawing attractive and informative statistical graphics (seaborn, 2018).



Donald Trump tweet sentiment analysis

In [2]:
```
%matplotlib inline
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
```

1. Downloading and Preparing the data

source: http://www.trumptwitterarchive.com/archive

In [3]:
```
input_file = "C:/Users/jangn/CODE/Sundog_DataScience/DataScience/DataScience-Python3/data_sets/WordClouds/
DonaldTrumpTweets_01052016_01052018.csv"

tweets = pd.read_csv(input_file)
tweets.tail()
```

Out[3]:

	text
5698	Will be in Terre Haute Indiana in a short whil...
5699	I watched Sen. Graham @FaceTheNation. Why don'...
5700	Thank you Indiana! #Trump2016 https://t.co/shP...
5701	I will be in Indiana on Sunday and Monday at f...

Figure 9. Sentiment Analysis on Donald Trump's tweets (a)

Most liked

In [7]:
```
fav_max = np.max(df_tweets['favorite_count'])

fav = df_tweets[df_tweets.favorite_count == fav_max].index[0] # print(fav): 3574

print("The tweet with most likes/favourite counts is: \n{}".format(df_tweets['text'][fav]))
print("Number of likes: {}".format(fav_max))
```

```
The tweet with most likes/favourite counts is:
Such a beautiful and important evening! The forgotten man and woman will never be forgotten again. We will
all come together as never before
Number of likes: 633253
```

In [8]:
```
# second highest liked
fav3 = df_tweets['favorite_count'].nlargest(3)
print(df_tweets['text'][fav3])
#print("The tweet with most likes/favourite counts is: \n{}".format(df_tweets['text'][fav3]))
#print("Number of likes: {}".format(fav_max))
```

```
favorite_count
633253    NaN
616217    NaN
605098    NaN
Name: text, dtype: object
```

Most retweeted

In [9]:
```
rt_max  = np.max(df_tweets['retweet_count'])

rt  = df_tweets[df_tweets.retweet_count == rt_max].index[0]

print("The tweet with most retweets is: \n{}".format(df_tweets['text'][rt]))
print("Number of retweets: {}".format(rt_max))
```

Figure 10. Sentiment Analysis on Donald Trump's tweets (b)

Out[64]:
```
14671.238967838444
```

In [65]:
```
#Avg Negative Favorited
negat_avg_fav = negative_tweets['favorite_count'].mean()
negat_avg_fav
```

Out[65]:
```
62713.656829100895
```

In [66]:
```
#Avg Negative Retweeted
negat_avg_rt = negative_tweets['retweet_count'].mean()
negat_avg_rt
```

Out[66]:
```
17962.862731640358
```

In [67]:
```
f, ax = plt.subplots(figsize=(16, 4))
values=[neut_avg_fav,neut_avg_rt, posit_avg_fav, posit_avg_rt, negat_avg_fav, negat_avg_rt ]
#colors = ['y','y','g','g', 'r', 'r']
colors =[ '#FFD700', '#FFD700', 'g', 'g', '#FF6347','#FF6347']
plt.bar(range(0,6), values, color=colors)
plt.show()
```

Figure 11. Sentiment Analysis on Donald Trump's tweets (c)

2.2 Finding of the Research

The system mentioned above are few examples of sentiment analysis applied in the data collect in internet. The above mention system has used NLP specialized library NLTK and API such as monkeylearn where one can create their own model and train them. Some examples did not used any specialised library nor any APIs but create their own system with the help of other libraries. The research was information as the knowledge of different libraries and their function known through the research.

The libraries which seems to be import were numpy, pandas, scikit learn and such. The numpy library is used for scientific computing with python. The pandas library provides high-performance, easy-to-use data structures and data analysis tools for python. And scikit learn has machine learning algorithm which can be used to understand the workflow of the machine learning algorithm. The research also gave insight on the specification of sentiment analysis. The text such as post, or tweets have their own library to tackle such problem and give more accurate. So, the research also pointed out to use proper libraries to use to handle data.

2.3 Resource required

- **Hardware requirements:** Laptop with Internet Connection
- **Software requirements:** Python, Django and PostgreSQL.

  Python is a high-level, interpreted and object-oriented programming language with dynamic semantics. It has high-level built in data structures, combined with dynamic typing binding (Python, 2020). It is easy to learn low maintenance cost. It supports modules and packages encouraging program modularity and code re-use.
- **Front-end:** The front-end will be created with the help of Django framework.

  It is a high-level web framework of Python which will be compatible with the code of Python in sentiment analysis. It encourages rapid development and clean, pragmatic design. It is fat, secure and flexible.

- **Database:** The PostgreSQL is used for the database of this project. It is a powerful, open source with object relational database system. It is also compatible with Django framework.

  The PostgreSQL is an opensource object-relational database that uses and extends SQL language. There are many features that include developers building applications, data integrity protection administrators, build-tolerant environments, data management regardless of size (PostgreSQL, 2020).
- **Back-end:** The sentiment analysis will be carried as the backend where the required data will be extracted, the data goes through tokenization, then stopping words will be removed, the text will be vectorized, the text will go through training algorithm, and lastly the prediction of result will be displayed.
- **Libraries:** The libraries which will be used in the numpy, pandas, word2vector and so on for sentiment analysis which are explained in brief in the finding.

2.4 Methodology

The framework used to structure, plan, and control the process of developing an information system is a software development methodology. There are numerous ways to organise the process of developing and writing code. There are many methodologies such as agile software development, lean development, waterfall, prototype, rapid application development, spiral model, rational unified process, devops deployment and so on (Software, 2020).

2.4.1 Possible Methodologies

- **Prototype Methodology**

Prototype methodology is a specialized software development technique that initiates developers to make a simple resolution to validate their functional nature to the customer and make important improvements as time passed by the customer's request before the true final solution is developed (K, 2018).

It gives clear idea of workflow of the process of the software which reduces the risk of failure in a solution and it assists well in requirement gathering and analysis of overall system. It requires excessive involvement of client and has chance of extension in management cost due to many changes occur during development phase.

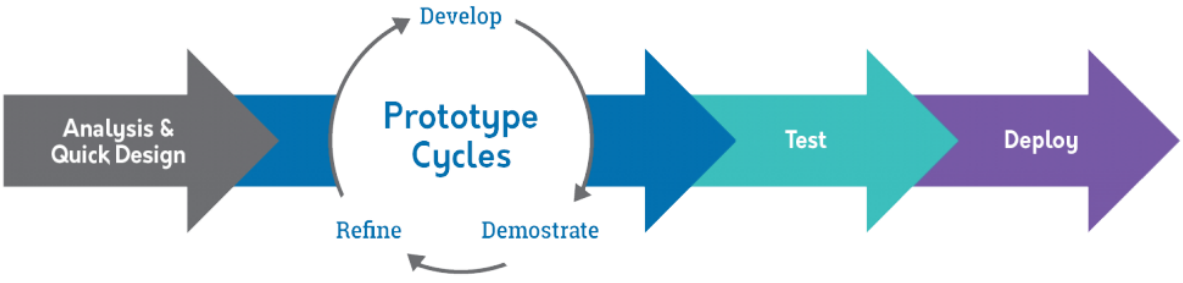


Figure 12. Prototype methodology

- **Rational Unified Process Methodology**

This methodology powers software using rational tools, it divides the development process into four different stages which includes business modelling, scrutiny and design, enactment, testing and disposition. It assists software developer by stating guidelines, templates, and specimens for all feature and stages of software development (K, 2018).

It focuses on precise documentation, removes risks in the project linked with client, and less requirement with the client. It requires involvement of expert software developer, complicated methodology, and integration is also very complicated to understand.

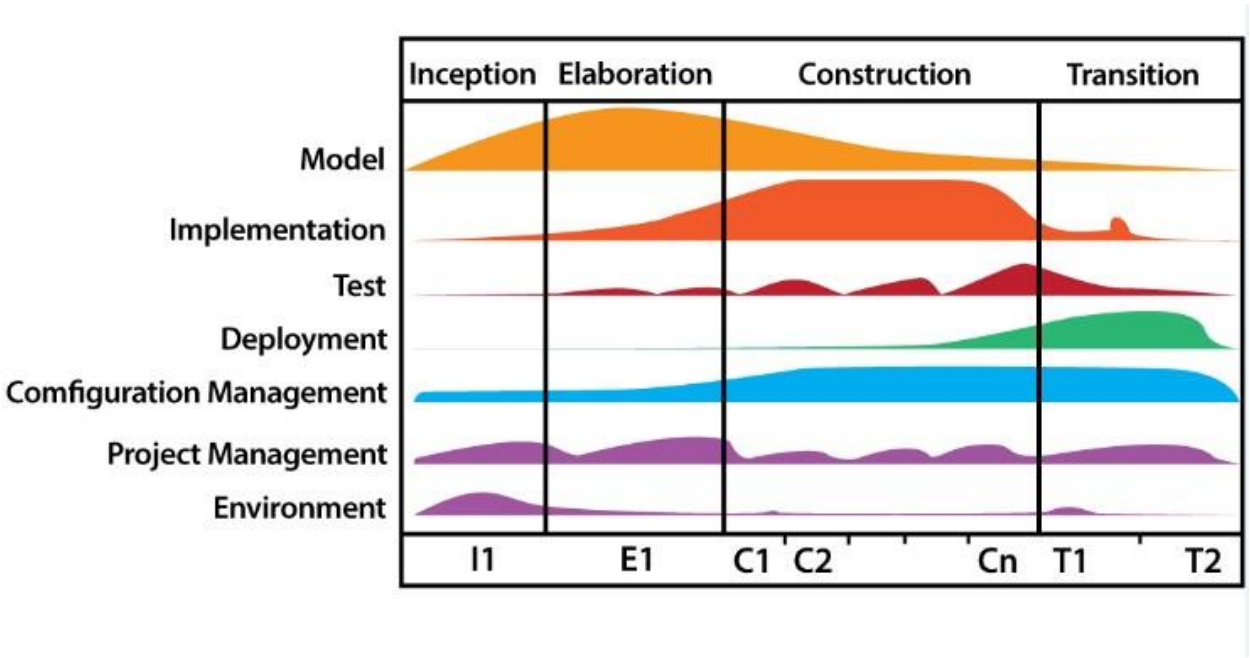


Figure 13. Rational Unified Process methodology

- **Iterative methodology**

The iterative methodology does not require full specification of requirement to initialize the software (Ghahrai, 2018). It can be started with specifying and implementing a part of the software and reviewed to identify further requirement. It has repetition of process multiple times to produce a new version of the software for each cycle of the model. The phases of the cycle include requirement phase, design phase, implementation and test phase and review phase.

The key success of this methodology is rigorous validation of requirements. Then test each version of the software against these criteria in each mode process. It generates quick software during the software life cycle, flexible, less cost for changes, easy testing and debugging, easy to manage risk and each iteration is easy managed milestone. The phases are rigid and do not overlap. There may be problem in system architecture as the requirement gathering is not completed in early stage.

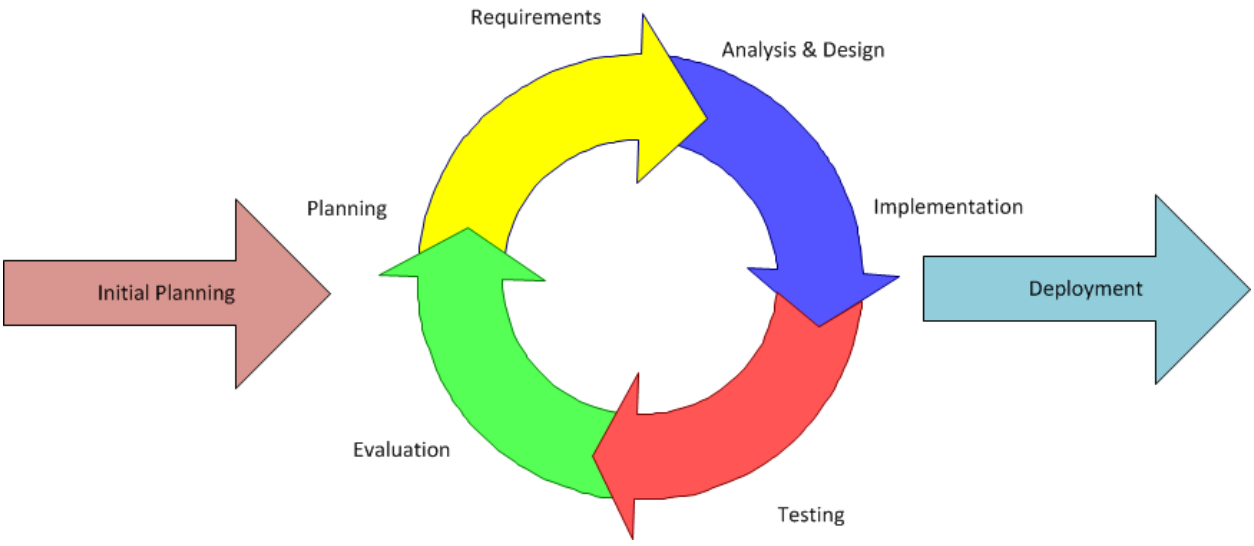


Figure 14. Iterative methodology

2.4.2 Selected Methodology

The methodology chosen for this project is Iterative methodology. It is an implementation of a software development life cycle (SDLC) that focuses on initial and simplified implementation, then further progresses to gain more complexity and border feature set until the completion of the system. The methodology is based on the concept of incremental development, which is often used liberally and interchangeably, it explains the incremental alterations made during the design and implementation of each new iteration. (Powell-Morse, 2016)

The first step is the extraction of data from Trip advisor which will create a list of reviews. The next step is data processing which is to convert the text in lower case and remove punctuation. The next step is tokenization which will create a vocab to Int mapping dictionary, encode the words and encode the labels. The next step is training, validation, test dataset split, the next step is data loaders and batching. The next step is defining the model class. The last step is testing on test data or user-generated data. The reason to select iterative methodology for this project is the repetition of the filtering for better performance as there are terms as bigrams and trigrams which plays a vital role in the sentiment score as the result of this project.

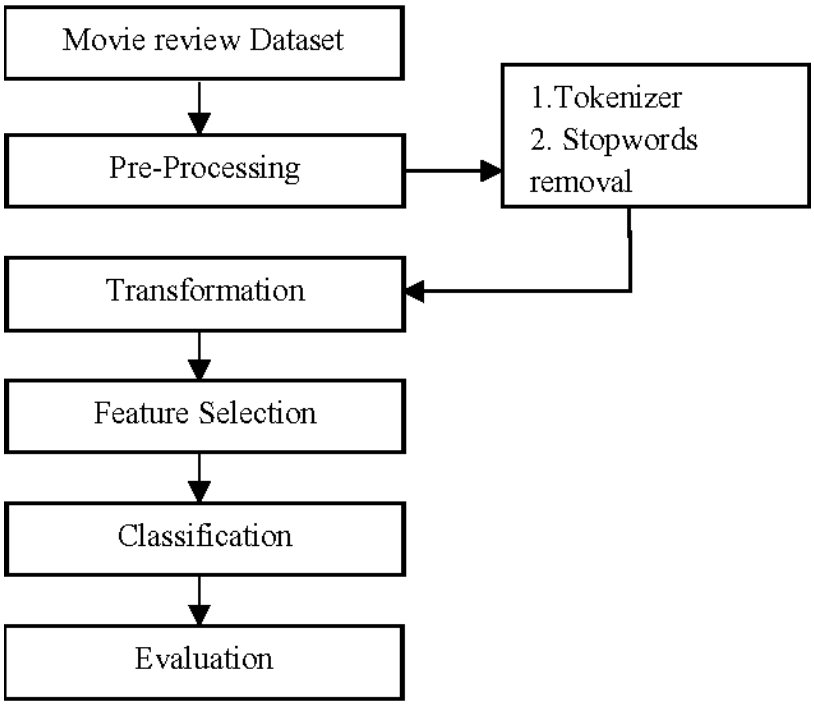


Figure 15. Stages in Sentiment Analysis

Chapter 3. Development till date

3.1 ERD

3.1.1 Initial ERD

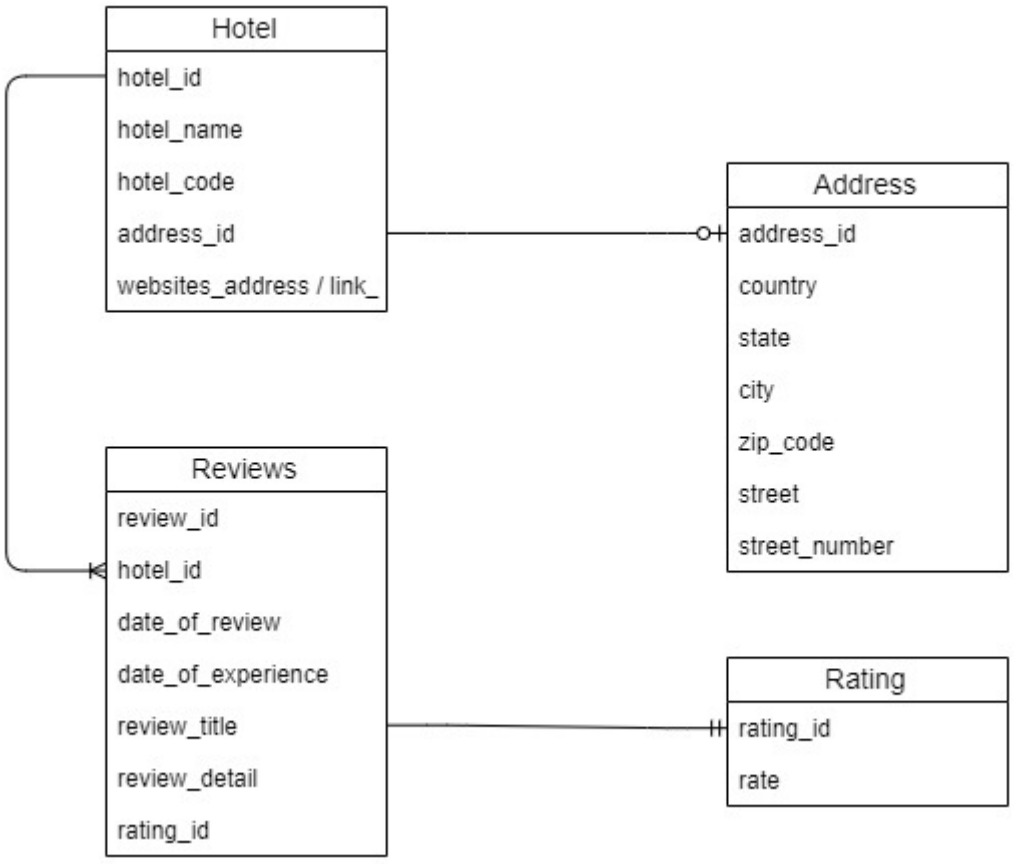


Figure 16. Initial ERD of the system

Assumptions:

- A hotel has single address
- A hotel can have multiple reviews
- A review will only have one rating

3.1.2 Normalization

UNF

Hotel = (hotel_code, hotel_name, address_id, country, state, city, zip_code, street, street_no, { review_id, review_title, review_detail, date_of_experience, date_of_review, rating_id, rate })

1NF

Hotel = (**hotel_code**, hotel_name, address_id, country, state, city, zip_code, street, street_no)

Hotel_Review = (**hotel_code*,** **review_id**, review_title, review_detail, date_of_experience, date_of_review, rating_id, rate)

2NF

Checking for partial dependency of Review,

hotel_code = X

review_id = review_title, review_detail, date_of_review, rating_id, rate

hotel_code, review_id = date_of_experience

Final 2NF

Hotel = (**hotel_code**, hotel_name, address_id, country, state, city, zip_code, street, street_no)

Hotel_Review = (**hotel_code***, **review_id*,** date_of_experience)

Review = (**review_id**, review_title, review_detail, date_of_review, rating_id, rate)

3NF

Checking for transitive dependency of Rating,

address_id = country, state, city, zip_code, street, street_no

rating_id = rating_id, rate

street_no =street

Final 3NF

Hotel = (**hotel_code**, hotel_name, **address_id***)

Address = (**address_id**, country, state, city, zip_code, **street_no***)

Review = (**review_id**, review_title, review_detail, date_of_review, **rating_id***)

Hotel_Review = (**hotel_code***, **review_id*,** date_of_experience, **rating_id***)

Rating = (**rating_id**, rate)

Street = (**street_no,** street)

3.1.3 Final ERD

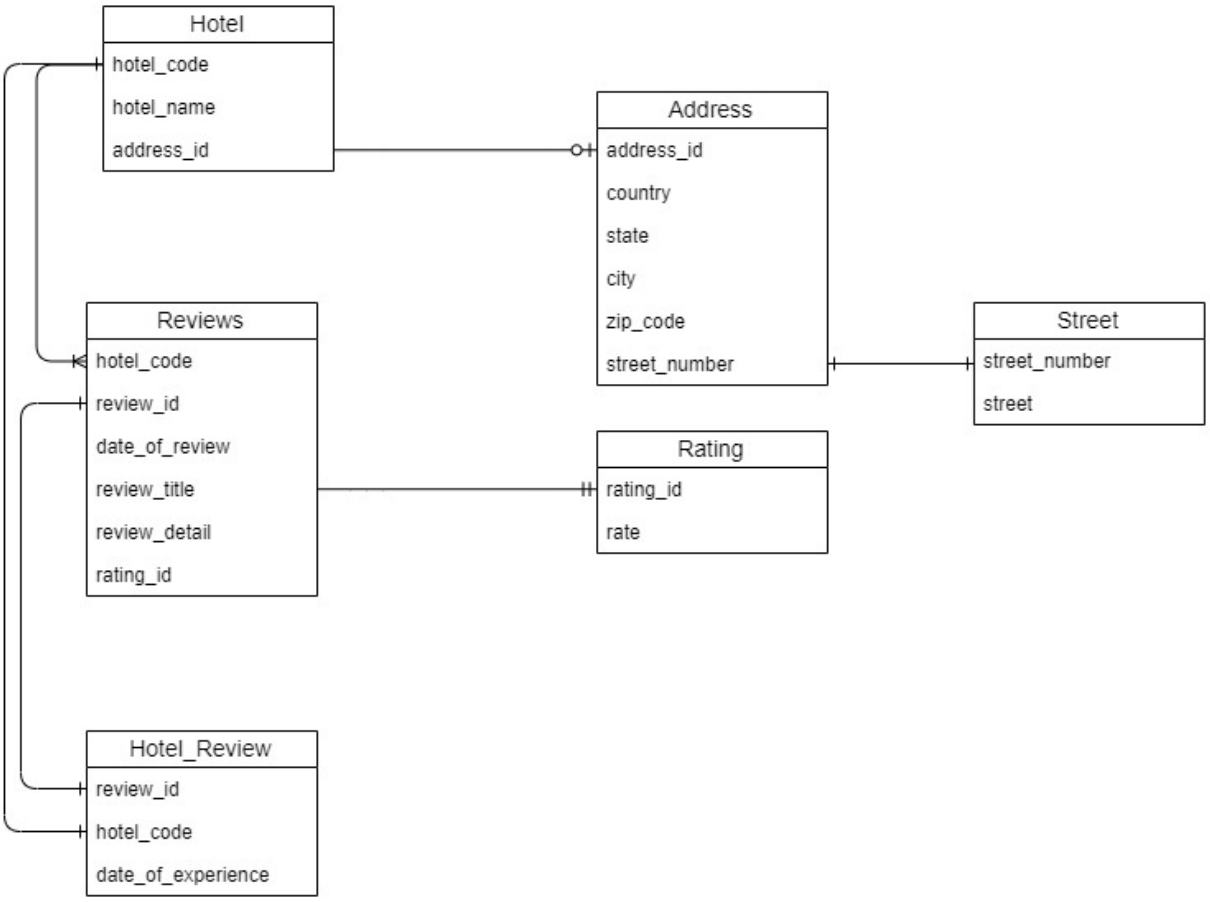


Figure 17. Final ERD of the system

3.2 Wireframe

The wireframe of the system is simple as the system has more work in code then in the website. The website has a text field for users to drop the link of the hotel of trip advisor. The system will take the link and extract the reviews of the hotel and display the sentiment analysis result in the result page in visual form such as chart and points.

Figure 18. Wireframe of the system (a)

Figure 19. Wireframe of the system (b)

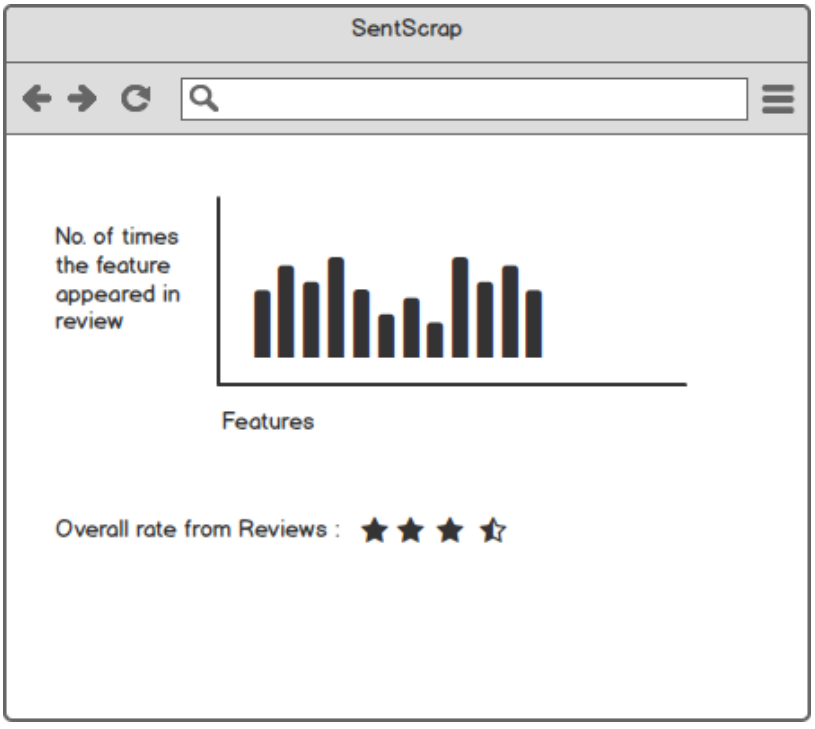


Figure 20. Wireframe of the system (c)

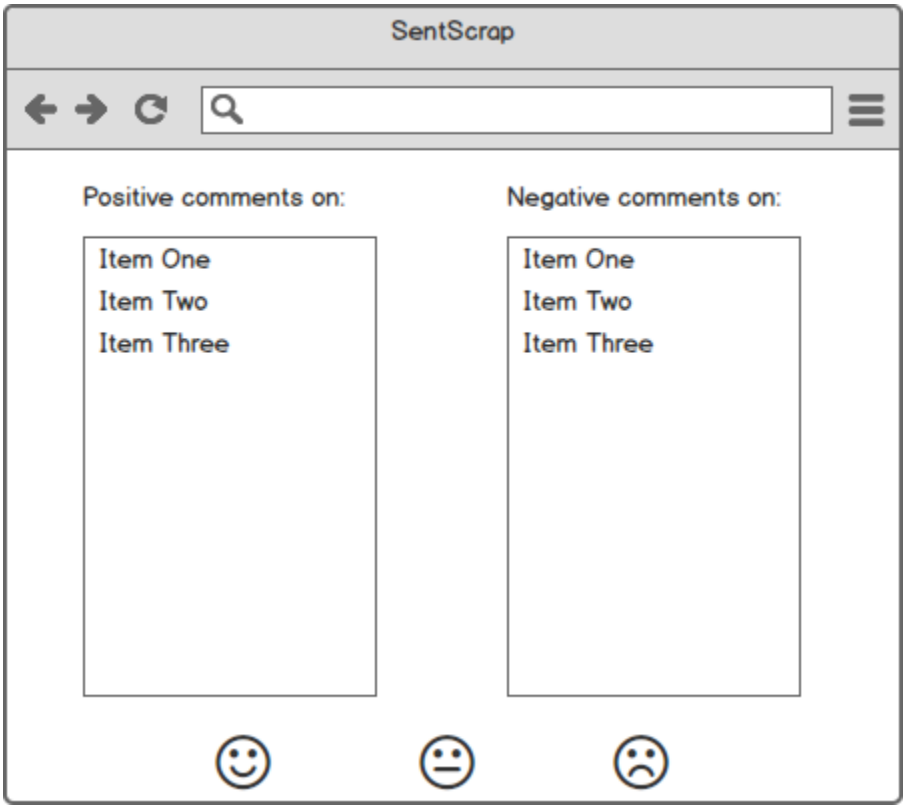


Figure 21. Wireframe of the system (d)

3.3 Use Case

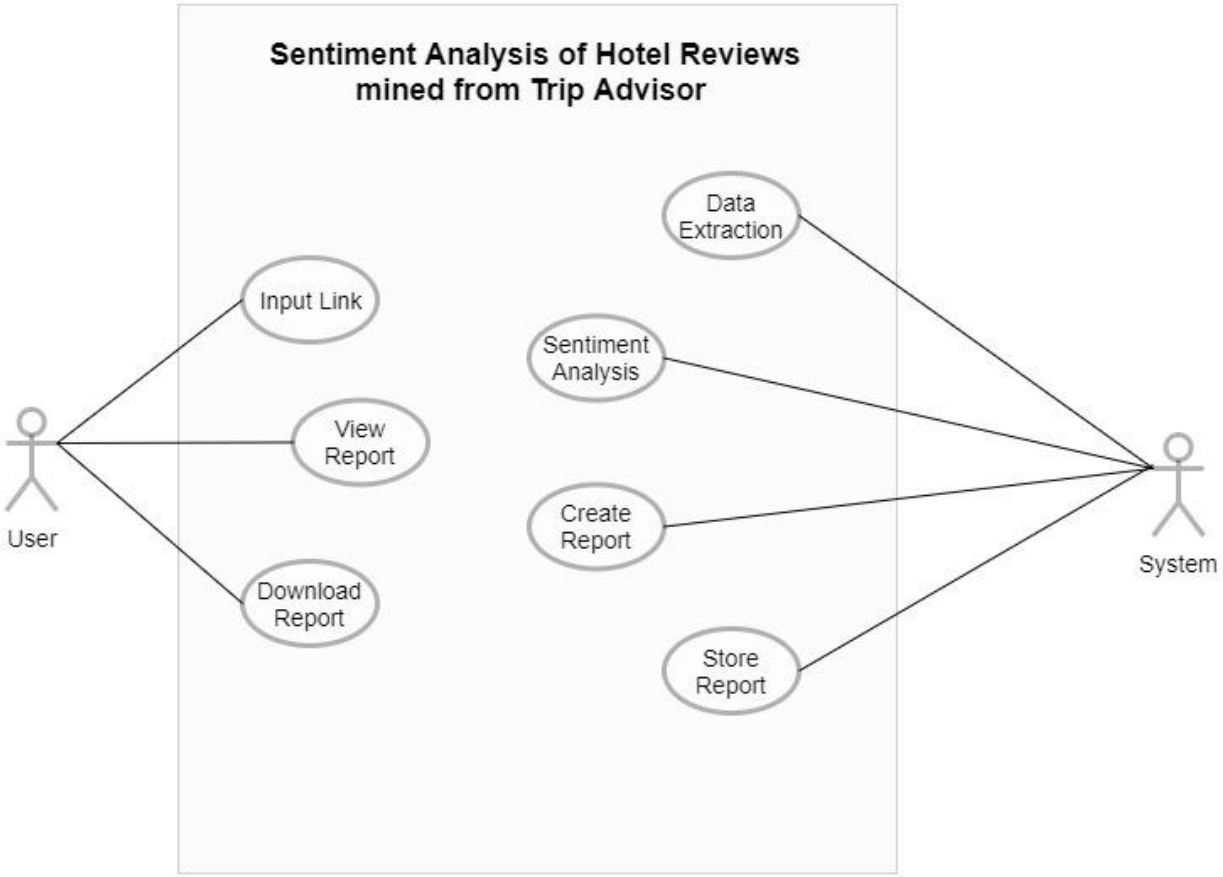


Figure 22. Use Case of the system.

3.4 High-level Use Case

Use Case	:	Input Link
Actors	:	User
Description	:	The user puts the link of the hotel from which the user wants to view the sentiment analysis report in the text field of the main page.

Use Case	:	Data Extraction
Actors	:	System
Description	:	The system extracts the reviews of the hotel from the link provided by the user.

Use Case	:	Sentiment Analysis
Actors	:	System
Description	:	The extracted data goes through the process of sentiment analysis and result is presented in the form of report.

Use Case	:	View Report
Actors	:	User
Description	:	The report prepared after the sentiment analysis is presented to user which is viewed by the user.

Use Case	:	Store Report
Actors	:	System
Description	:	The report generated by the system will be stored in the database.

3.5 Expanded Use Case

Use Case : Input Link

Actors : User

Description : The user puts the link of the hotel from which the user wants to view the sentiment analysis report in the text field of the main page.

Typical Course of Events:

User	System
1. A user copies the link of a hotel from trip advisor.	
2. The copied link is inserted in the text field of the websites.	
	3. The system checks if the link provided by the user belongs to trip advisor or not.
	4. If the link provided by the user exists, the data extraction proceeded and if the link does not exist the system asks the user to re-insert the link.

Use Case : Data Extraction

Actors : System

Description : The system extracts the reviews of the hotel from the link provided by the user.

Typical Course of Events:

User	System
	1. The reviews from the link provided by user is extracted.
	2. The extracted data is stored in CSV file.

Use Case : Sentiment Analysis

Actors : System

Description : The extracted data goes through the process of sentiment analysis and result is presented in the form of report.

Typical Course of Events:

User	System
	1. The process of sentiment analysis is implemented on the csv file. 2. The result of sentiment analysis is stored in database. 3. The data stored in database is displayed as result to the user.

Use Case : View Report

Actors : User

Description : The report prepared after the sentiment analysis is presented to user which is viewed by the user.

Typical Course of Events:

User	System
1. The result displayed by the system is viewed by the user.	

Use Case	:	Store Report
Actors	:	System
Description	:	The report generated by the system will be stored in the database.

Typical Course of Events:

User	System
	1. The report generated by the system is stored in the database.

3.6 Front-end Development

The website home page is shown in the figure below, figure 23, the home page is simple with a welcome slogan and functional search box is created with the help of jQuery. The orange circle will transform in a text field when the mouse hovers the circle which is shown in the next figure, figure 24. The text field is created for user to drop the link of hotel review mined from trip advisor.

In figure 25, the database created for this is shown. The database was crested using Django framework and PostgreSQL..

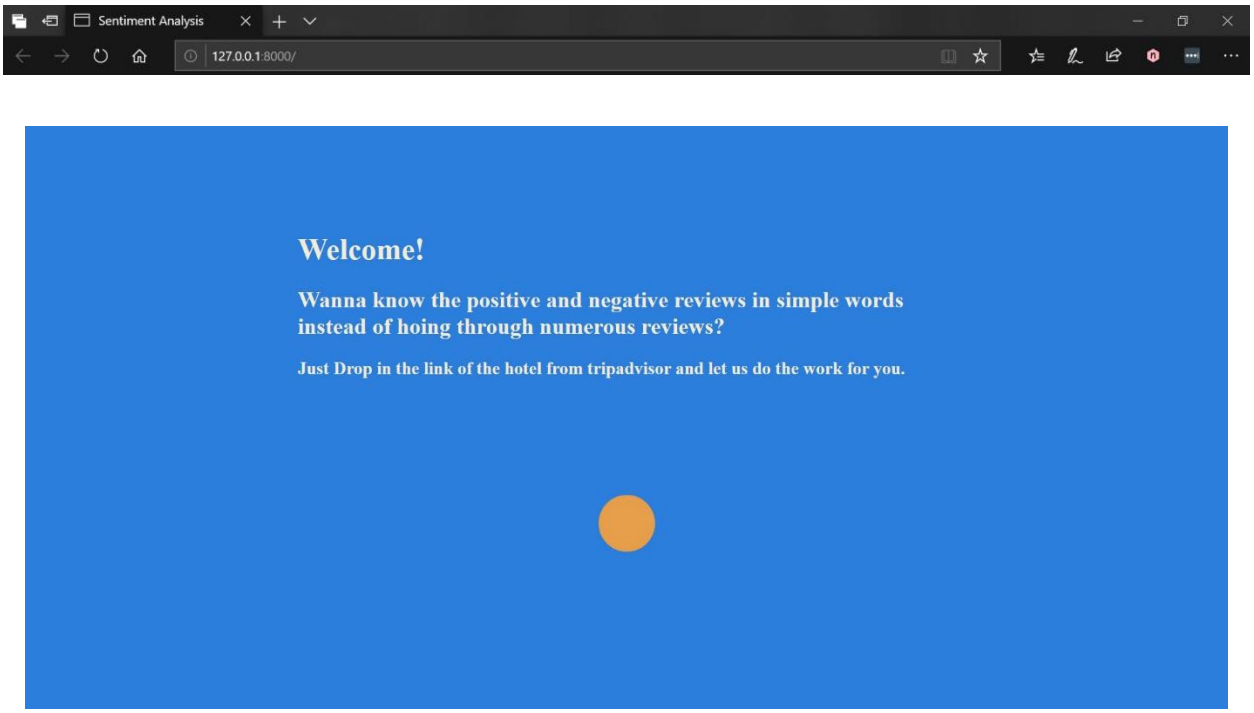


Figure 23. Front-end Development (a)

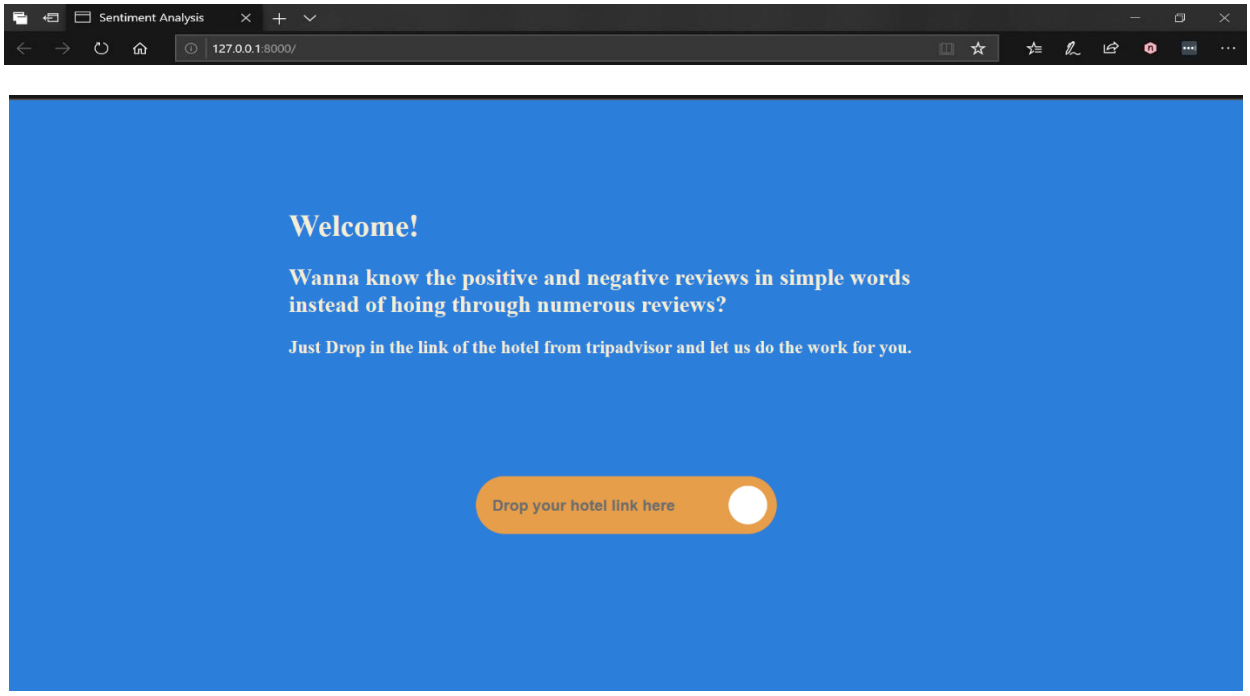


Figure 24. Front-end Development (b)

3.7 Back-end Development

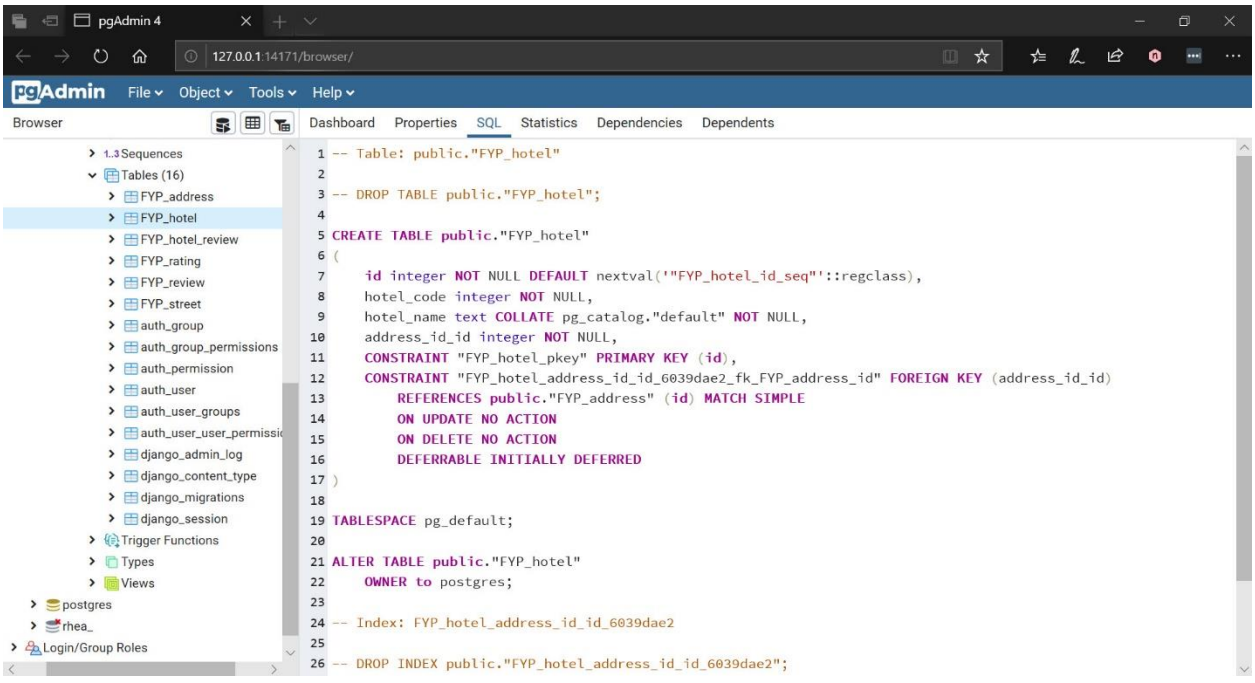


Figure 25. Back-end Development

Chapter 4. Analysis of progress

4.1 Assignment of Student Table

S.no.	Task	Sub-Task	Start Date	End Date	Status
1.	Analyse	Analyse of Data Extraction	5-11-2019	12-11-2019	Completed
		Analyse of NLP	7-11-2019	13-112019	Completed
		Analyse of Sentiment Analysis	10-11-2019	20-11-2019	Completed
2.	Requirements	Data Acquisition	21-11-2019	11-12-2019	On-going
		Text pre-processing	7-12-2019	16-12-2019	On-going
3.	Analysis	Feature selection	17-12-2019	25-12-2019	Completed
		Feature extraction	22-12-2019	3-1-2020	Completed
4.	Design	Sentiment Classification	4-1-2020	30-1-2020	On-going

Table 1. Table of Analyze progress

4.2 Progress Detail

Research on sentiment analysis: The main task of this project was to gain knowledge as much as possible in order to start the project. Likewise, the first task in Gantt chart was to gather knowledge regarding data extraction, NLP and Sentiment analysis (Reference appendix-A). This stage was used as information gathering stage on related topics so there will be no problem in the future related to such topic. During this process, many libraries and API were discovered along with their work function which the clear view of the idea about the system should be created with specific libraries.

Requirement (data extraction): This phase was allocated for data extraction, but it was used for implementing different libraries for sentiment analysis. As the main task of the project was of sentiment analysis, most of the time was spent on choosing the best library for sentiment analysis instead of data extraction. The workflow of data extraction was studied but not implemented. The extracted data was collected online and the sentiment analysis was implemented so the lack of time for data extraction will be recovered by spending some less time on sentiment analysis library and more od data extraction so the time track of the Gantt chart stays intact.

Analysis (feature section): This phase was allocated for analysis or feature selection. This phase was used make ERD, wireframe, use case, high-level use case and expanded use case to have mind-set to carry out the stages of development of the software. The initial ERD was made which normalized to give final ERD to have a well-managed database to store the results of sentiment analysis. The wireframe is the base for the website creation, it lets us arrange the placement of elements of a web so we can make a proper website. The use case simply explains the right and job of the user, all the work and the response is all listed out which makes programming easy as we know what is required.

Design (sentiment classification): This phase where the selected libraries are placed together to form a model to train the model for sentiment classification. Alongside creation of model, the home page is created for the website where one can drop the link for the data extraction. After the completion of ERD, database was created to store the result of the sentiment analysis and Django framework was connected to PostgreSQL.

Chapter 5. Future work

Task	Sub-Task	Start Date	End Date	Duration (Days)
Testing	Create model	22-1-2020	14-2-2020	18
Implementation	Polarity detection	7-2-2020	3-3-2020	18
Review	Validation and evaluation	29-2-2020	9-3-2020	7
Phase 2: Iteration 2				
Design	Classification for bigrams	10-3-2020	18-3-2020	7
Testing	Create new model with new data for bigrams	13-3-2020	23-3-2020	7
Implementation	Polarity detection for bigrams	16-3-2020	25-3-2020	8
Review	Validation and evaluation	22-3-2020	27-3-2020	6
Phase 3: Iteration				
Design	Classification for trigrams	28-3-2020	9-4-2020	10
Testing	Create model with new data for trigrams	30-3-2020	6-4-2020	6
Implementation	Polarity detection for trigrams	1-4-2020	10-4-2020	9
Review	Integration and Test Support	5-4-2020	14-4-2020	9
Deployment	Plan Deployment	15-4-2020	22-4-2020	7
	Develop Support Material	18-4-2020	27-4-2020	9
	Produce Deployment Unit	20-4-2020	28-4-2020	7
Maintenance	Manage Acceptance test (At Development Site)	22-4-2020	30-4-2020	7
	Manage Acceptance test (At Development Site)	24-4-2020	4-5-2020	7
Documentation	Interim Report	10-11-2019	7-1-2019	43
	Final Report	10-1-2020	4-5-2020	82

Table 2. Table of Future work

References

Anusha_Sharma. (2020, 1 5). *Difference between Machine learning and Artificial Intelligence*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/difference-between-machine-learning-and-artificial-intelligence/

Ash, J. C. (2020, 1 2). *How the World Wide Web gave birth to the Internet of Things*. Retrieved from Medium: https://medium.com/dataseries/how-the-world-wide-web-gave-birth-to-the-internet-of-things-57c808d112f

Daityari, S. (2019, 0 26). *How To Perform Sentiment Analysis in Python 3 Using the Natural Language Toolkit (NLTK)*. Retrieved from Community: https://www.digitalocean.com/community/tutorials/how-to-perform-sentiment-analysis-in-python-3-using-the-natural-language-toolkit-nltk

Daityari, S. (2020, January 5). *How To Perform Sentiment Analysis in Python 3 Using the Natural Language Toolkit (NLTK)*. Retrieved from Community: https://www.digitalocean.com/community/tutorials/how-to-perform-sentiment-analysis-in-python-3-using-the-natural-language-toolkit-nltk

Ellingwood, J. (2016, 9 28). *An Introduction to Big Data Concepts and Terminology*. Retrieved from Community: https://www.digitalocean.com/community/tutorials/an-introduction-to-big-data-concepts-and-terminology

Expert Systems. (2017, 3 7). *What is Machine Learning? A definition*. Retrieved from Expert Systems: https://expertsystem.com/machine-learning-definition/

G.Vinodhini*, R. (2012). *International Journal of Advanced Research in.* Annamalai Nagar: ijarcsse.

Ghahrai, A. (2018, December 2). *Iterative Model*. Retrieved from Testing Excellence: https://www.testingexcellence.com/iterative-model/

K, J. (2018, june 1). *12 BEST SOFTWARE DEVELOPMENT METHODOLOGIES WITH PROS AND CONS*. Retrieved from acodez: 12 BEST SOFTWARE DEVELOPMENT METHODOLOGIES WITH PROS AND CONS

Kumar, N. (2020, 1 5). *Twitter Sentiment Analysis using Python*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/

matplotlib. (2020, 1 5). *matplotlib*. Retrieved from matplotlib: https://matplotlib.org/

nltk. (2019). *Natural Language Toolkit*. Retrieved from nltk: https://www.nltk.org/

nltk. (2020, 1 7). *Corpus Readers*. Retrieved from nltk: http://www.nltk.org/howto/corpus.html

Nordin, J. (2018, 5 26). *Donald Trump tweet sentiment analysis*. Retrieved from github: https://github.com/LJANGN/Sentiment-Analysis-on-Donald-Trump-s-tweets/blob/master/DT_SA.ipynb

Novalić, A. (2013, 1 23). *Introduction to tweepy, Twitter for Python*. Retrieved from Python Central: https://www.pythoncentral.io/introduction-to-tweepy-twitter-for-python/

numpy. (2019). *numpy*. Retrieved from numpy: https://numpy.org/

pandas. (2019, 11 9). *pandas: powerful Python data analysis toolkit*. Retrieved from pandas: https://pandas.pydata.org/pandas-docs/stable/

PostgreSQL. (2020). *What is PostgreSQL*. Retrieved from PostgreSQL: https://www.postgresql.org/about/

Python. (2020). *Python*. Retrieved from Python: https://www.python.org/doc/essays/blurb/

seaborn. (2018). *seaborn: statistical data visualization*. Retrieved from seaborn: https://seaborn.pydata.org/

Shetty, B. (2018, 11 25). *Natural Language Processing(NLP) for Machine Learning*. Retrieved from towards data science: https://towardsdatascience.com/natural-language-processing-nlp-for-machine-learning-d44498845d5b

Software, A. (2020). *An Introduction To Software Development Methodologies*. Retrieved from Alliance Software: https://www.alliancesoftware.com.au/introduction-software-development-methodologies/

Stecanella, B. (2019, 5 8). *Sentiment Analysis with Python*. Retrieved from monkeylearn: https://monkeylearn.com/blog/sentiment-analysis-with-python/

TextBlob. (2020, 1 7). *TextBlob*. Retrieved from TextBlob: https://textblob.readthedocs.io/en/dev/

Bibliography

https://www.guru99.com/what-is-big-data.html

https://towardsdatascience.com/mach ine-learning-vs-big-data-lets-find-the-relationship-between-them-e55c9c861311

https://theappsolutions.com/blog/development/machine-learning-and-big-data/

https://ieeexplore.ieee.org/abstract/document/7219856

https://patents.google.com/patent/US8838633B2/en

https://monkeylearn.com/sentiment-analysis/

https://becominghuman.ai/connection-between-data-science-ml-and-ai-d1c18d89b0bd

https://books.google.com.np/books?hl=en&lr=&id=lSkIewOw2WoC&oi=fnd&pg=PR5&dq=machine+learning+and+artificial+intelligence+relation&ots=T1EMP-egn0&sig=_g0XKi7AqOOowJy_51FsLtchi8&redir_esc=y#v=onepage&q=machine%20learning%20and%20artificial%20intelligence%20relation&f=false

https://www.ibm.com/developerworks/library/ba-sentiment-analysis-big-data/index.html

https://towardsdatascience.com/fine-grained-sentiment-analysis-in-python-part-1-2697bb111ed4

https://www.researchgate.net/publication/312176414_Sentiment_Analysis_in_Python_using_NLTK

https://tweepy.readthedocs.io/en/latest/

https://www.analyticsvidhya.com/blog/2018/02/natural-language-processing-for-beginners-using-textblob/

https://stackabuse.com/python-for-nlp-introduction-to-the-textblob-library/

https://www.nltk.org/book/ch02.html

https://www.nltk.org/api/nltk.corpus.html

https://www.nltk.org/book/

https://pythonspot.com/category/nltk/

https://realpython.com/python-matplotlib-guide/

https://www.geeksforgeeks.org/python-introduction-matplotlib/

http://cs231n.github.io/python-numpy-tutorial/

https://pandas.pydata.org/pandas-docs/version/0.15/tutorials.html

https://pandas.pydata.org/

https://python-graph-gallery.com/seaborn/

https://en.wikibooks.org/wiki/Introduction_to_Software_Engineering/Process/Methodology

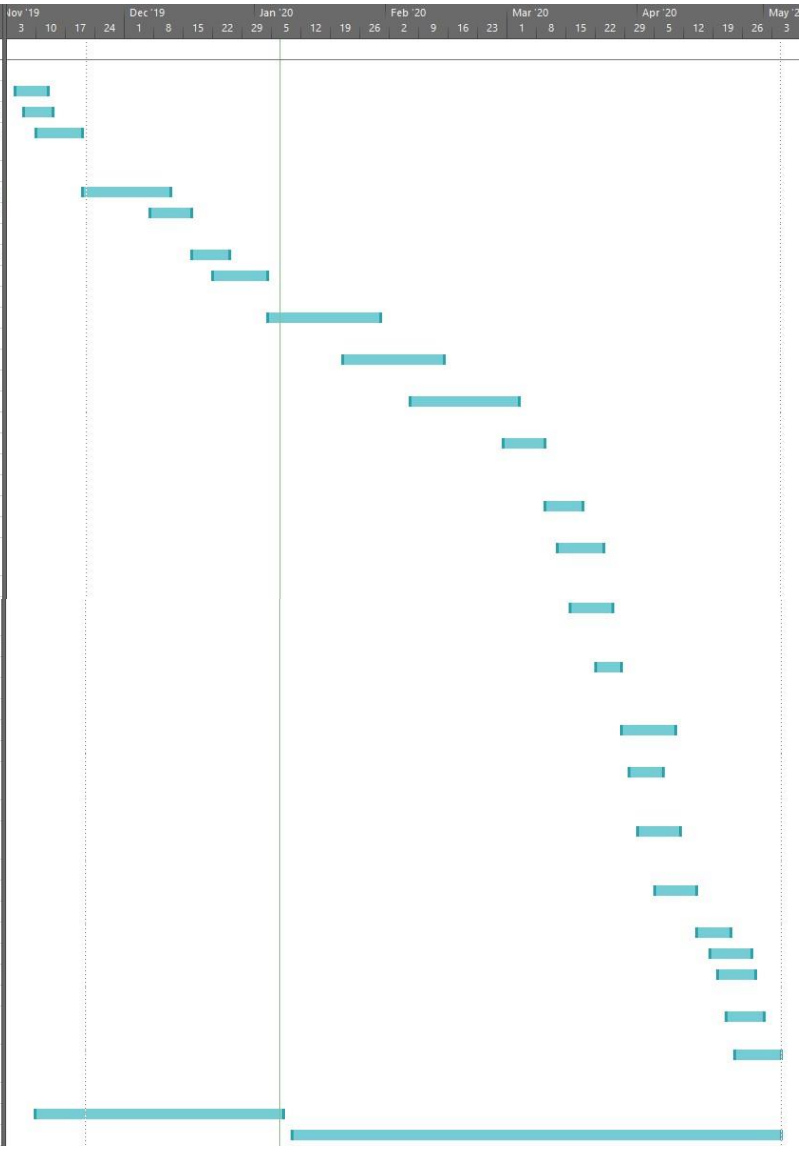
http://www.professionalqa.com/iterative-model

https://searchsoftwarequality.techtarget.com/definition/iterative-development

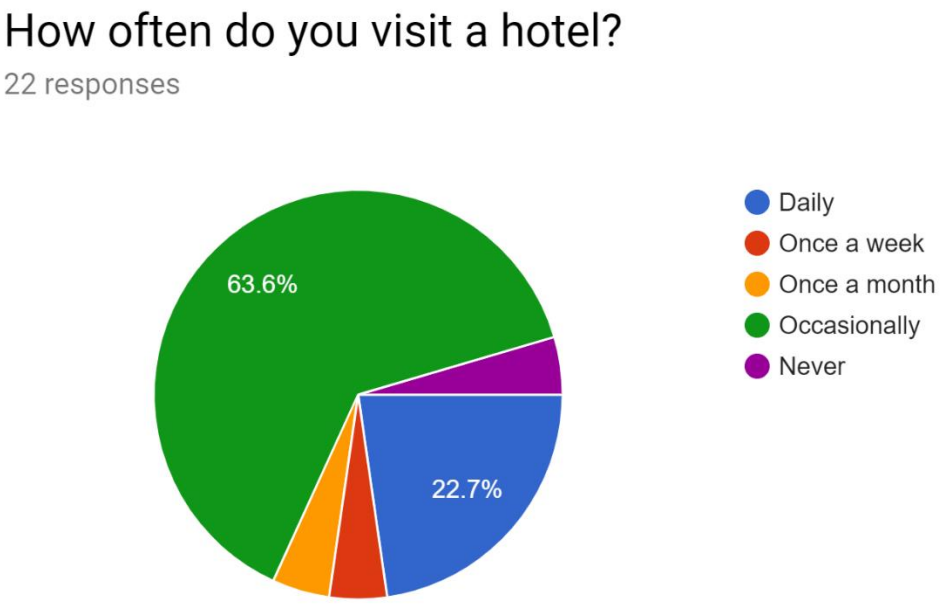
Appendix

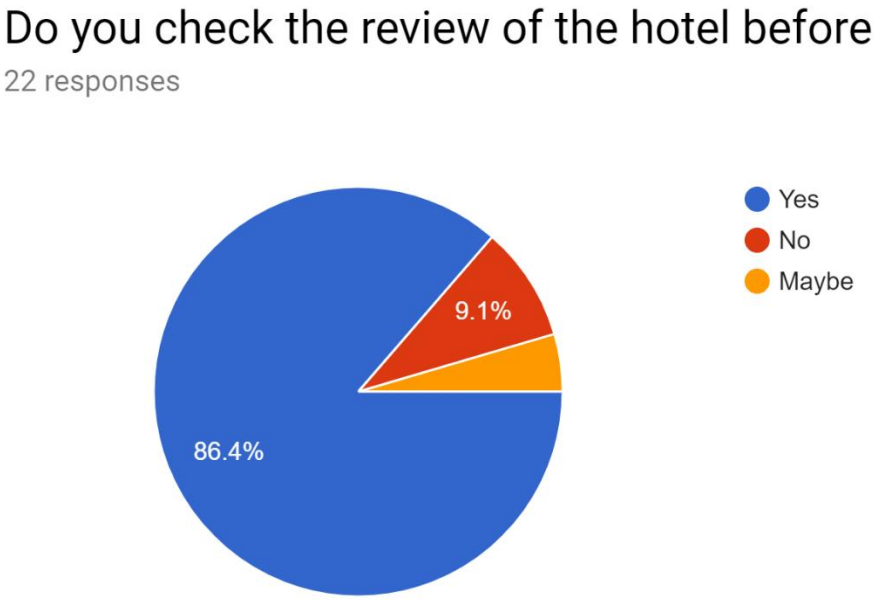
1. Appendix A – Gantt Chart

	Task Name	Duratio	Start	Finish
1	**Phase 1: Iteration 1**			
2	**Analyze**			
3	Analyze of Data Extraction	6 days	Tue 11/5/19	Tue 11/12/19
4	Analyze of NLP	5 days	Thu 11/7/19	Wed 11/13/19
5	Analyze of Sentiment Analysis	9 days	Sun 11/10/19	Wed 11/20/19
6	**Requirements**			
7	Data Acquistition	15 days	Thu 11/21/19	Wed 12/11/19
8	Text preprocessing	7 days	Sat 12/7/19	Mon 12/16/19
9	**Analysis**			
10	Feature selection	7 days	Tue 12/17/19	Wed 12/25/19
11	Feature extraction	11 days	Sun 12/22/19	Fri 1/3/20
12	**Design**			
13	Sentiment Classification	20 days	Sat 1/4/20	Thu 1/30/20
14	**Testing**			
15	Create model	18 days	Wed 1/22/20	Fri 2/14/20
16	**Implementation**			
17	Polarity detection	18 days	Fri 2/7/20	Tue 3/3/20
18	**Review**			
19	Validation and evaluation	7 days	Sat 2/29/20	Mon 3/9/20
20	**Phase 2: Iteration 2**			
21	**Design**			
22	Classification for bigrams	7 days	Tue 3/10/20	Wed 3/18/20
23	**Testing**			
24	Create new model with new data for bigrams	7 days	Fri 3/13/20	Mon 3/23/20
25	**Implementation**			
26	Polarity detection for bigrams	8 days	Mon 3/16/20	Wed 3/25/20
27	**Review**			
28	Validation and evaluation	6 days	Sun 3/22/20	Fri 3/27/20
29	**Phase 3: Iteration 3**			
30	**Design**			
31	Classification for trigrams	10 days	Sat 3/28/20	Thu 4/9/20
32	**Testing**			
33	Create model with new data for trigrams	6 days	Mon 3/30/20	Mon 4/6/20
34	**Implementation**			
35	Polarity detection for trigrams	8 days	Wed 4/1/20	Fri 4/10/20
36	**Review**			
37	Integration and Test Support	8 days	Sun 4/5/20	Tue 4/14/20
38	**Deployment**			
39	Plan Deployment	6 days	Wed 4/15/20	Wed 4/22/20
40	Develop Support Material	7 days	Sat 4/18/20	Mon 4/27/20
41	Produce Deployment Unit	7 days	Mon 4/20/20	Tue 4/28/20
42	**Maintenance**			
43	Manage Acceptance test (At Development Site)	7 days	Wed 4/22/20	Thu 4/30/20
44	Manage Acceptance test (At Installation Site)	7 days	Fri 4/24/20	Mon 5/4/20
45	**Documentation**			
46	Interim Report	43 days	Sun 11/10/19	Tue 1/7/20
47	Final Report	82 days	Fri 1/10/20	Mon 5/4/20



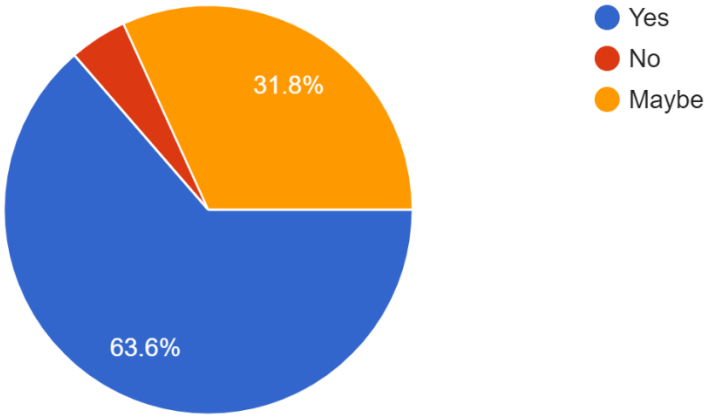
2. Appendix B – Survey Result





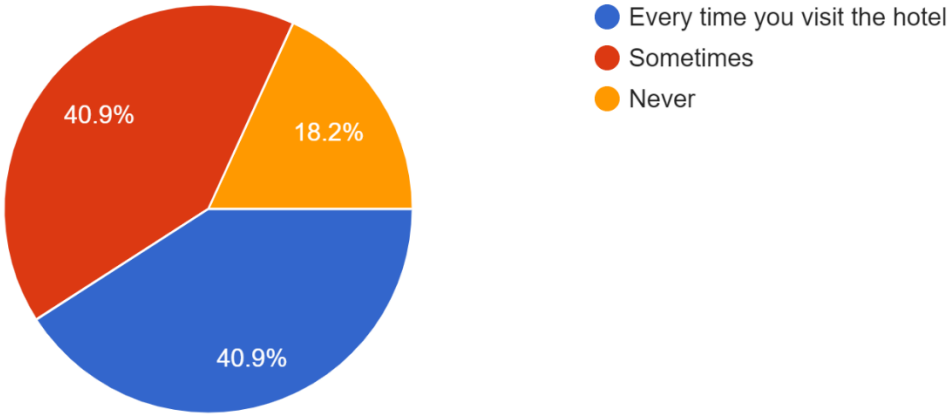
Does your opinion change about a hotel after reading the reviews?

22 responses



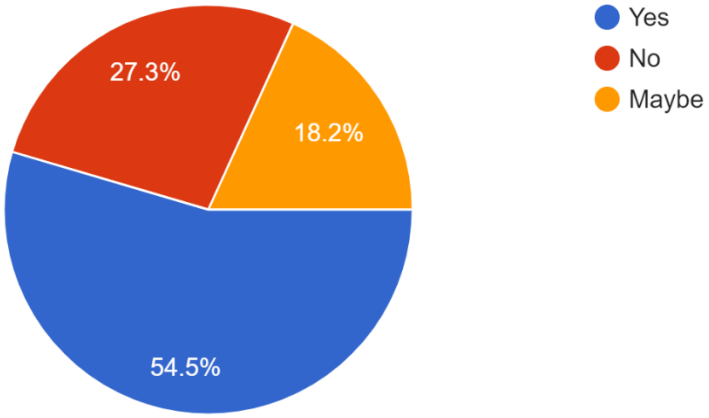
How often you leave feedback for a hotel online?

22 responses



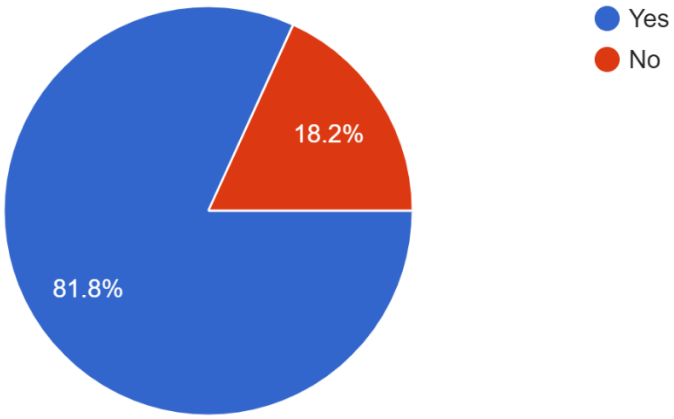
Have you ever heard about about Sentiment Analysis?

22 responses



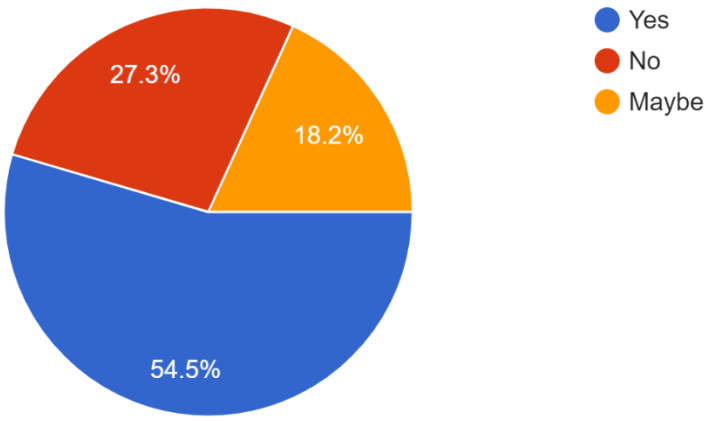
Do you post review if you didn't like anything about the hotel?

22 responses



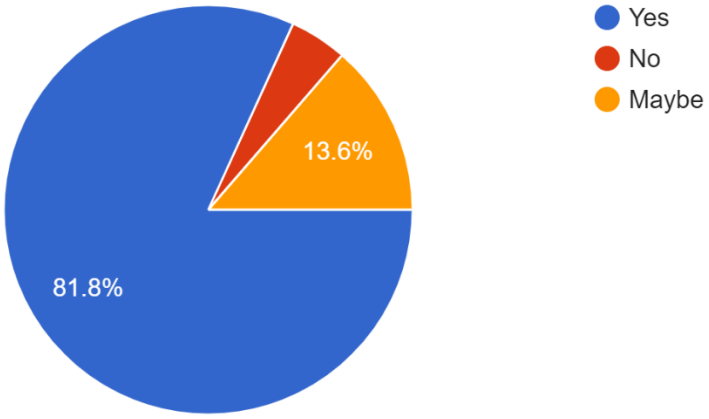
Have you ever heard about about Sentiment Analysis?

22 responses



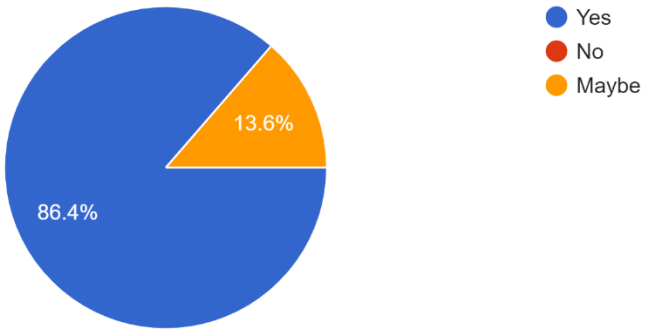
Do you think a machine can differentiate between a negative and a positive comment?

22 responses



Do you think analyzing the reviews will help hotels give better customer services.

22 responses



3. Appendix C – Code

3.1 Front End Code

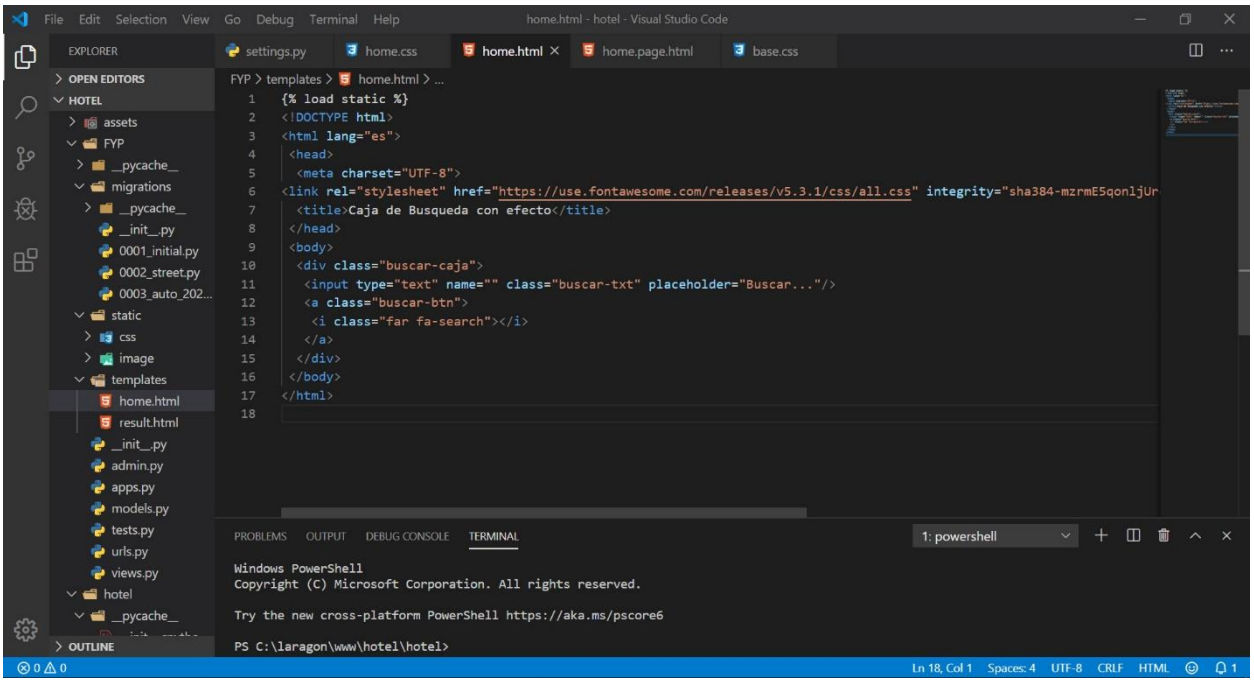


Figure 27. Front -End Code of the project - home.page

3.2 Back End Code

```
# Database
# https://docs.djangoproject.com/en/2.1/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'hoteldb',
        'USER': 'postgres',
        'PASSWORD': 'root',
        'HOST': '127.0.0.1',
        'PORT': '5432',
    }
}
```

Figure 28. Back-end Code - database connection

FYP > models.py > Hotel_Review

```
from django.db import models

# Create your models here.

class Rating(models.Model):
    rating_id = models.IntegerField()
    rate = models.CharField(max_length=10)


class Street(models.Model):
    street_no = models.IntegerField()
    street = models.CharField(max_length=100)


class Address(models.Model):
    address_id = models.IntegerField()
    country = models.TextField()
    state = models.TextField()
    city = models.TextField()
    zip_code = models.IntegerField()
    street_no = models.ForeignKey(Street, on_delete=models.CASCADE)

```

Figure 29. Back-end Code - Model creation (Tables) (a)

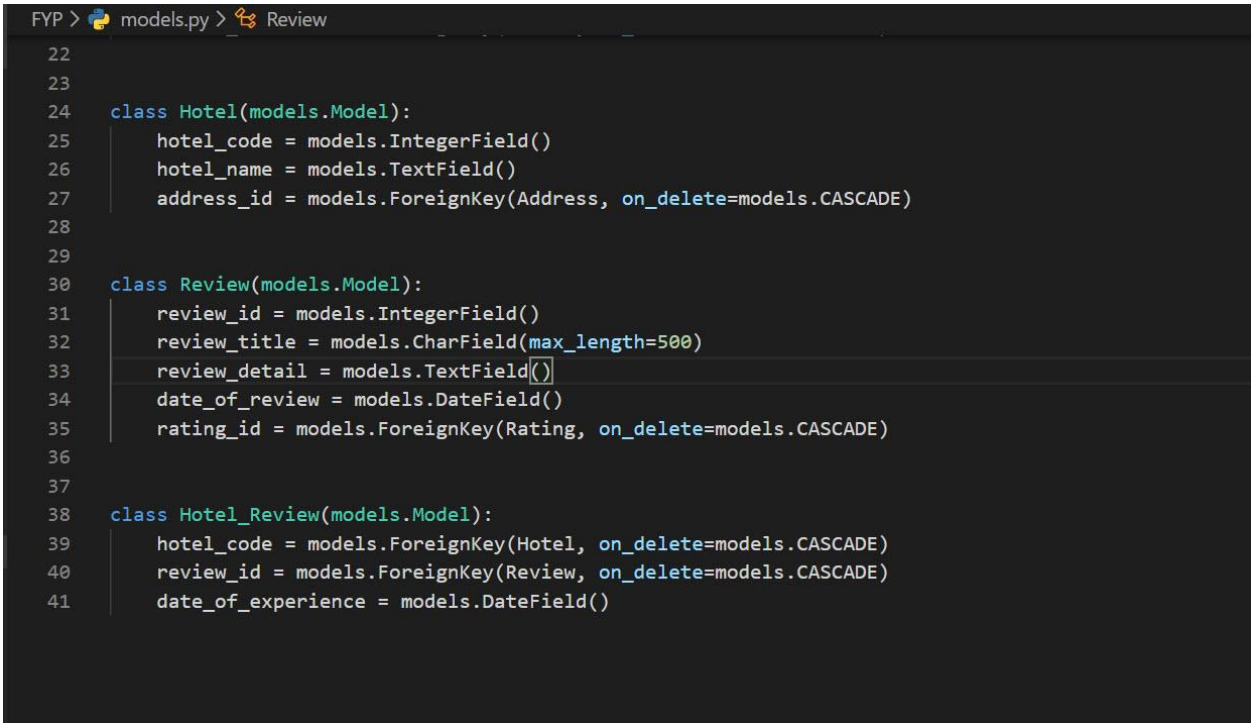


Figure 30. Back-end Code - Model creation (Tables) (b)

3.3 Sentiment Analysis Code

- Sentiment Analysis Code using Json, numpy, pandas and Scikit library.

```
In [ ]:
import json as j
import pandas as pd
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.feature_extraction.text import TfidfTransformer


# --- read and transform json file
json_data = None
with open('untitled1.txt') as data_file:
    lines = data_file.readlines()
    joined_lines = "[" + ",".join(lines) + "]"

    json_data = j.loads(joined_lines)

data = pd.DataFrame(json_data)
print(data.head())


# --- prepare the data

data = data[data.stars != 3]
data['sentiment'] = data['stars'] >= 4
print(data.head())

# --- build the model

X_train, X_test, y_train, y_test = train_test_split(data, data.sentiment, test_size=0.2)

# -
count = CountVectorizer()
temp = count.fit_transform(X_train.text)

tdif = TfidfTransformer()
temp2 = tdif.fit_transform(temp)
```

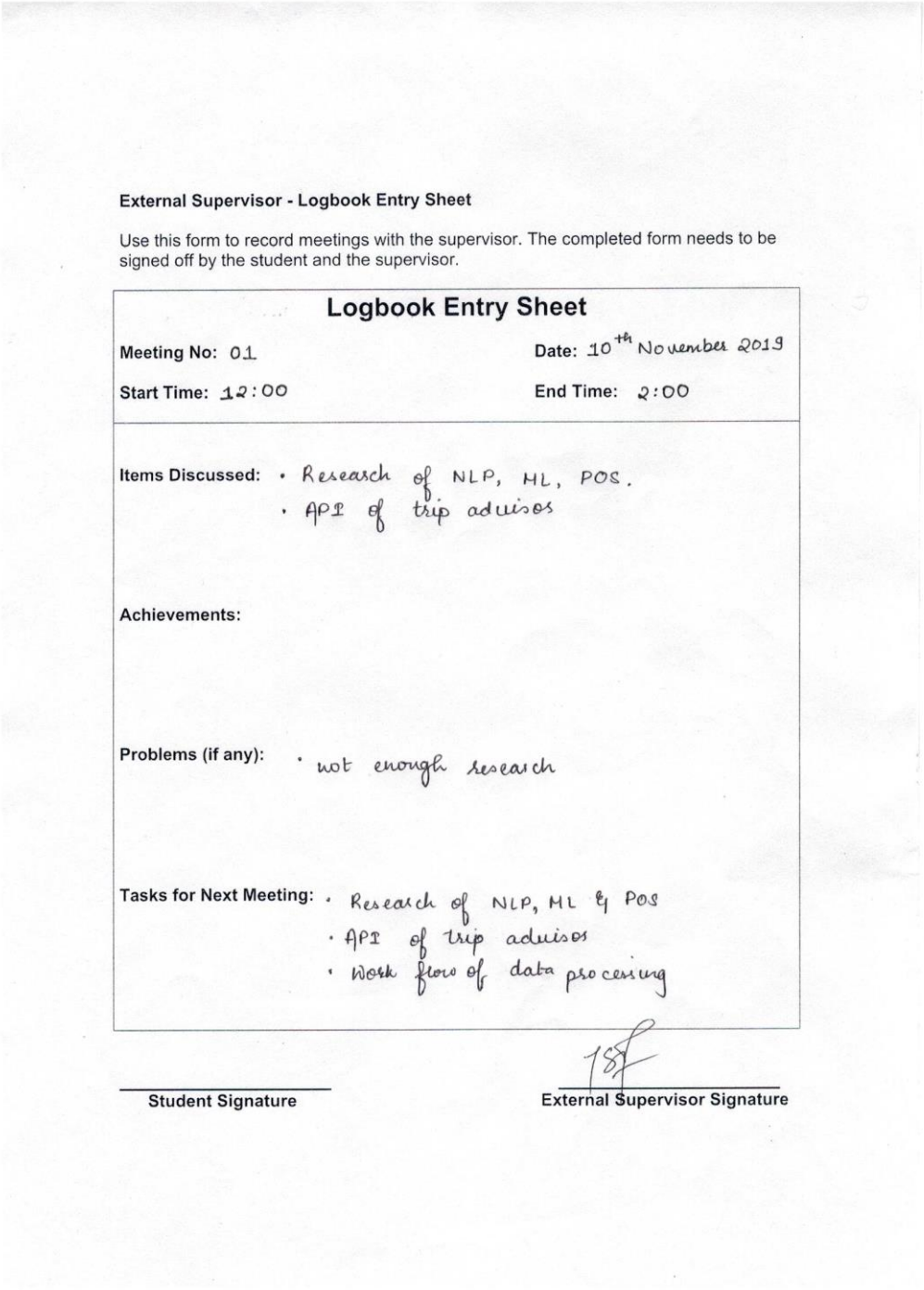
Figure 31. Sentiment Analysis Code (a)

```
temp2 = tdif.fit_transform(temp)

text_regression = LogisticRegression()
model = text_regression.fit(temp2, y_train)

prediction_data = tdif.transform(count.transform(X_test.text))

predicted = model.predict(prediction_data)

# instead of doing all this steps above one could also use Pipeline
# this is a more compact way of writing the code above...
# it also has the benefit that there is no need to perform the transformations on the test data
#
#
#from sklearn.pipeline import Pipeline
#text_regression = Pipeline([('count', CountVectorizer()), ('tfidf', TfidfTransformer()),('reg', LogisticRegression())])
#
#model = text_regression.fit(X_train.text, y_train)
#predicted = model.predict(X_test.text)


# --- make predictions


print(np.mean(predicted == y_test))

# --- have some fun with the model

print(model.predict(tdif.transform(count.transform(["this product was a great video game"]))))
```

```
In [9]:
print(model.predict(tdif.transform(count.transform(["this product was a great video game"]))))

[ True]
```

Figure 32. Sentiment Analysis Code (b)

4. Appendix D – Logbook Entry Sheet



External Supervisor - Logbook Entry Sheet

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

Logbook Entry Sheet

Meeting No: 01 **Date:** 10th November 2019

Start Time: 12:00 **End Time:** 2:00

Items Discussed:
- Research of NLP, ML, POS.
- API of trip advisor

Achievements:

Problems (if any):
- not enough research

Tasks for Next Meeting:
- Research of NLP, ML & POS
- API of trip advisor
- Work flow of data processing

Student Signature **External Supervisor Signature**

Figure 33. Logbook 1 (a)

Internal Supervisor - Logbook Entry Sheet

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

Logbook Entry Sheet

Meeting No: 01 **Date:** 10th November 2019

Start Time: 12:00 **End Time:** 2:00

Items Discussed:
- Research regarding NLP, ML, POS,
- API regarding trip Advisor

Achievements:

Problems (if any):
- not enough search

Tasks for Next Meeting:
- Research of NLP, ML, POS.
- API of trip Advisor
- work flow of data processing
- reasons for selection of methodology.

Student Signature **Internal Supervisor Signature**

Figure 34. Logbook 1 (b)

External Supervisor - Logbook Entry Sheet

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

Logbook Entry Sheet

Meeting No: 02 **Date:** 17th Novembes 2019

Start Time: 12:00pm **End Time:** 2:00 pm

Items Discussed:
1) NLP Theoritical knowledge
2) ML, Web Scraping, Methodology
3) Basic idea of sentiment analysis break down

Achievements:
1) Theoritical knowledge of NLP
2) ML & NLP relation
3) Basic idea of sentiment analysis
4) Use of methodology
5) Web Scraping

Problems (if any):
1) Gantt Chart
2) More Theoritical concept
3) Lack of implementation

Tasks for Next Meeting:
1) NLP implementation on simple text
2) Web Scraping of simple data.

Student Signature

External Supervisor Signature

Figure 35. Logbook 2 (a)

Internal Supervisor - Logbook Entry Sheet

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

Logbook Entry Sheet

Meeting No: 02 **Date:** 17th November, 2019

Start Time: 12:00 pm **End Time:** 2:00 pm

Items Discussed:
1) NLP Implementation (theoritical knowledge)
2) ML, Web scraping, methodology
3) Basic idea of sentiment analysis break down

Achievements:
1) Theoritical knowledge of NLP
2) ML, Web & NLP relation
3) Basic idea of sentiment analysis
4) Use of methodology
5) Web Scraping

Problems (if any):
1) Gantt chart
2) More theoritical concept
3) Lack of implementation

Tasks for Next Meeting:
1) NLP implementation on simple text
2) Web scraping of simple data.

Student Signature

Internal Supervisor Signature

Figure 36. Logbook 2 (b)

Internal Supervisor - Logbook Entry Sheet

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

Logbook Entry Sheet

Meeting No: 03 **Date:** 24th November 2019

Start Time: 12:00 pm **End Time:** 2:00pm

Items Discussed: 1) Wireframe

Achievements: 1) Wireframe finalized

Problems (if any): 1) Wireframe
- graph unclear.
- result not shown properly.

Tasks for Next Meeting:
1) Sentiment analysis on simple text
2) Use Case
3) ERD
4) Extended Use case

Student Signature

Internal Supervisor Signature

Figure 37. Logbook 3 (a)

External Supervisor - Logbook Entry Sheet

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

Logbook Entry Sheet

Meeting No: 03 **Date:** 24th November 2019

Start Time: 12:00 pm **End Time:** 2:00 pm

Items Discussed: 1) Wireframe

Achievements: 1) Wireframe

Problems (if any): 1) Wireframe
- graph uncleas
- result not shown properly

Tasks for Next Meeting: 1) Use Case
2) Extended Use Case
3) ERD
4) Sentiment analysis on simple text

Student Signature **External Supervisor Signature**

Figure 38. Logbook 3 (b)

Internal Supervisor - Logbook Entry Sheet

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

Logbook Entry Sheet

Meeting No: 4 **Date:** 8th December 2019

Start Time: 12:00 **End Time:** 2:00

Items Discussed:
1) Database
2) Comparing extracted & new extracted data
3) Libraries for development
4) Download option

Achievements:
1) Use Case
2) ERD

Problems (if any): 1) Libraries for development

Tasks for Next Meeting:
1) Download option addition
2) Development of sentiment analysis.

Student Signature **Internal Supervisor Signature**

Figure 39. Logbook 4 (a)

External Supervisor - Logbook Entry Sheet

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

Logbook Entry Sheet

Meeting No: 04 **Date:** 8th December 2019

Start Time: 12:00 **End Time:** 2:00

Items Discussed: 1) Database
2) Comparing extracted & new extracted data
3) Libraries for development
4) Download option

Achievements: 1) Use Case
2) ERD

Problems (if any): 1) Libraries for development

Tasks for Next Meeting: 1) Add Download option
2) Development of sentiment analysis

Student Signature **External Supervisor Signature**

Figure 40. Logbook 4 (b)

Internal Supervisor - Logbook Entry Sheet

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

Logbook Entry Sheet

Meeting No: 05 **Date:** 22nd December 2019

Start Time: 12:00 pm **End Time:** 2:00 pm

Items Discussed:
- Final ERD
- Libraries for sentiment analysis

Achievements:
- ERD

Problems (if any):
- use of libraries

Tasks for Next Meeting:
- Website (dashboard)
- High level use case
- Interim report

Student Signature

Internal Supervisor Signature

Figure 41. Logbook 5 (a)

External Supervisor - Logbook Entry Sheet

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

Logbook Entry Sheet

Meeting No: 05 **Date:** 22nd December, 2019

Start Time: 12:00 pm **End Time:** 2:00 pm

Items Discussed:
- Finally ERD
- Libraries for sentiment analysis

Achievements:
- ERD

Problems (if any):
- use of libraries

Tasks for Next Meeting:
- High level use case

Student Signature **External Supervisor Signature**

Figure 42. Logbook 5 (b)

External Supervisor - Logbook Entry Sheet

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

Logbook Entry Sheet

Meeting No: 06 **Date:** 5th Jan, 2020

Start Time: 12:00 **End Time:** 2:00

Items Discussed:
- Database
- Interim Report
- Normalization

Achievements:
- Normalization

Problems (if any):
- DataBase

Tasks for Next Meeting:
- Interim Report Completion
- Database

Student Signature

External Supervisor Signature

Figure 43. Logbook 6 (a)

Internal Supervisor - Logbook Entry Sheet

Use this form to record meetings with the supervisor. The completed form needs to be signed off by the student and the supervisor.

Logbook Entry Sheet

Meeting No: 06 **Date:** 5th Jan, 2020

Start Time: 12:00 **End Time:** 2:00

Items Discussed:
- Database
- Interim Report
- Normalization

Achievements:
- Normalization

Problems (if any):
- Database

Tasks for Next Meeting:
- Interim Report completion
- Database

Student Signature

Internal Supervisor Signature

Figure 44. Logbook 6 (b)