Configuration Manual

1. Introduction

The following configuration manual illustrates the requirements for implementing the system which was designed for detecting the fake product reviews by using the Deep Learning models and NLP (Natural Language Processing) techniques. Further, the manual will thoroughly explain the software and hardware requirements that were used for the successful implementation of the project.

2. System Configuration

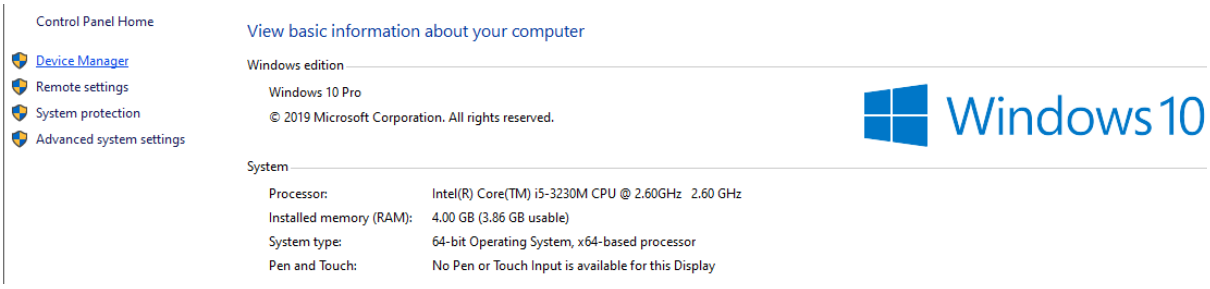
Following are the hardware and software configuration which were used for the implementation of this Project.

The hardware configurations used for implementation are as follows:

2.1. Hardware Requirement

Hardware	**Configurations**
System	Lenovo Z580 Idea pad
Operating System	Windows 10 (64 Bits) Pro
RAM	4 GB
Hard Disk	1 TB
Graphics Card	NVIDIA RTX 2060 (6 GB)
Processor	Intel Core i5-3230M

Table 1: Hardware requirement



View basic information about your computer

Windows edition

Windows 10 Pro

© 2019 Microsoft Corporation. All rights reserved.

System

Processor: Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz 2.60 GHz

Installed memory (RAM): 4.00 GB (3.86 GB usable)

System type: 64-bit Operating System, x64-based processor

Pen and Touch: No Pen or Touch Input is available for this Display

Figure 1: Operating System Configurations

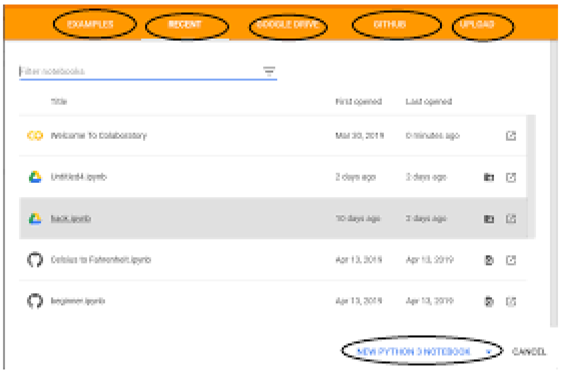
2.2. Software Requirement

The software configurations used for implementation are as follows:

Software	**Version**
Python	3.8 (64 Bits)
Google Colab Community	2021.2 (64 Bits)

Table 2: Software Requirements

Figure 2: Jupyter notebook and Google Colab integration



Title	First opened	Last opened
Welcome To Colaboratory	Mar 30, 2019	0 minutes ago
Untitled4.ipynb	2 days ago	2 days ago
back.ipynb	10 days ago	2 days ago
Celsius to Fahrenheit.ipynb	Apr 13, 2019	Apr 13, 2019
beginner.ipynb	Apr 13, 2019	Apr 13, 2019

Figure 3: Loading the jupyter notebook in Google Colab

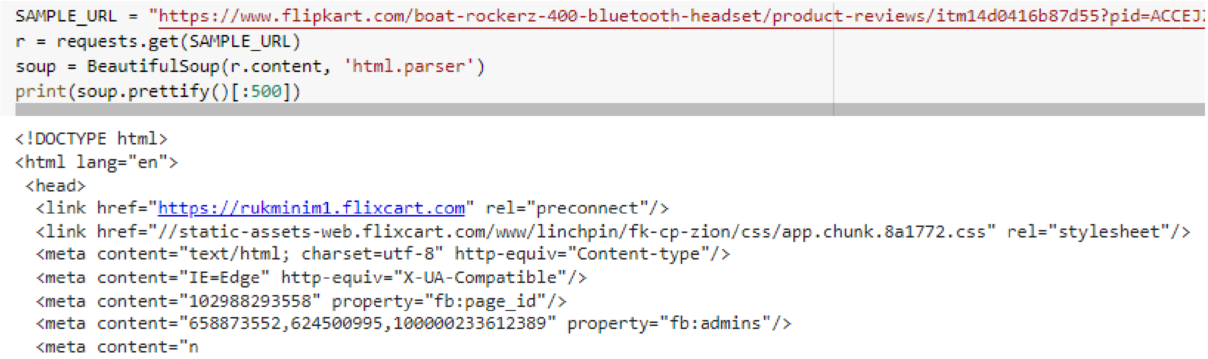
3. Project Implementation

3.1. Data summary

The list below depicts the data column summary and the data column description of the dataset which is scrapped from the online shopping website called Flipkart.

1. product_id: This data attribute describes the product identification number (ID)
2. product_title: This data attribute describes the product title displayed on the flipkart website
3. rating: This data attributes describes the rating of the product out of 5
4. summary: This data attributes is the summarization and description for the product.
5. Review: This data attribute consists of the review regarding the product
6. Location: This data attribute describes the location of the user or the reviewer
7. Date: This data attribute is the date of the review posted on the website
8. Upvotes: This attribute consists of the upvote (Positive) attribution of the product
9. Downvotes: This attribute consists of the downvote (Negative) attribution of the product

3.2. Data Preparation

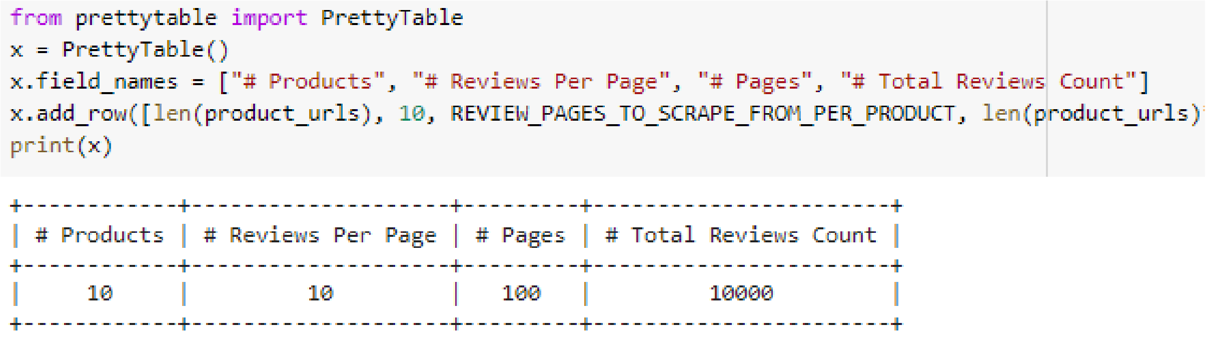


```
SAMPLE_URL = "https://www.flipkart.com/boat-rockerz-400-bluetooth-headset/product-reviews/itm14d0416b87d55?pid=ACCEJ
r = requests.get(SAMPLE_URL)
soup = BeautifulSoup(r.content, 'html.parser')
print(soup.prettify()[:500])
```

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <link href="https://rukminim1.flixcart.com" rel="preconnect"/>
  <link href="//static-assets-web.flixcart.com/www/linchpin/fk-cp-zion/css/app.chunk.8a1772.css" rel="stylesheet"/>
  <meta content="text/html; charset=utf-8" http-equiv="Content-type"/>
  <meta content="IE=Edge" http-equiv="X-UA-Compatible"/>
  <meta content="102988293558" property="fb:page_id"/>
  <meta content="658873552,624500995,100000233612389" property="fb:admins"/>
  <meta content="n
```

Figure 4: HTML component scrapping from the website's webpage

The figure above depicts the HTML component scrapping from the online shopping web page of Flipkart.

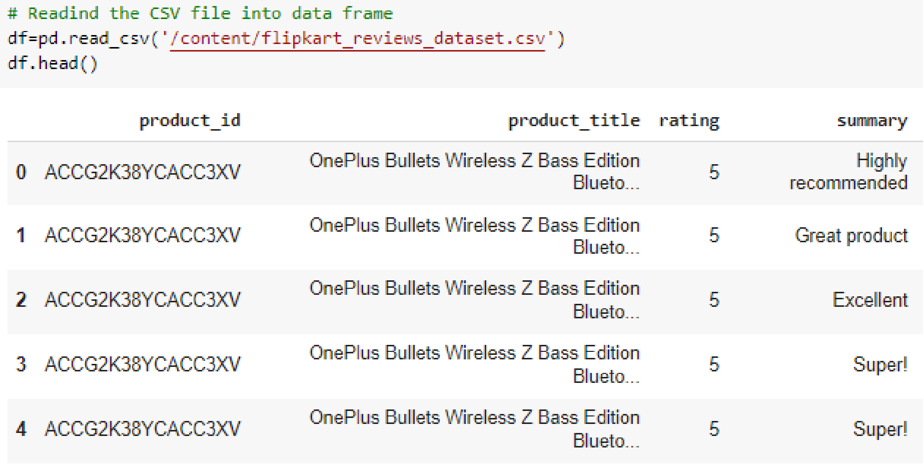


```
from prettytable import PrettyTable
x = PrettyTable()
x.field_names = ["# Products", "# Reviews Per Page", "# Pages", "# Total Reviews Count"]
x.add_row([len(product_urls), 10, REVIEW_PAGES_TO_SCRAPE_FROM_PER_PRODUCT, len(product_urls)
print(x)
```

```
+------------+--------------------+---------+-----------------------+
| # Products | # Reviews Per Page | # Pages | # Total Reviews Count |
+------------+--------------------+---------+-----------------------+
|     10     |         10         |   100   |         10000         |
+------------+--------------------+---------+-----------------------+
```

Figure 5: Review analysis table format

The figure above depicts the review analysis in the form of table format using the pretty table library of Python 3.8.

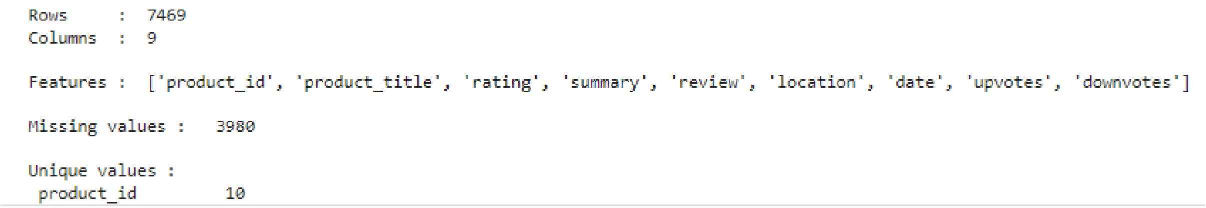


```
# Readind the CSV file into data frame
df=pd.read_csv('/content/flipkart_reviews_dataset.csv')
df.head()
```

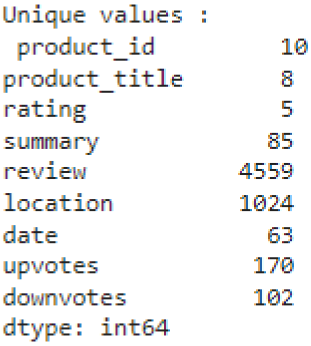
	product_id	product_title	rating	summary
0	ACCG2K38YCACC3XV	OnePlus Bullets Wireless Z Bass Edition Blueto...	5	Highly recommended
1	ACCG2K38YCACC3XV	OnePlus Bullets Wireless Z Bass Edition Blueto...	5	Great product
2	ACCG2K38YCACC3XV	OnePlus Bullets Wireless Z Bass Edition Blueto...	5	Excellent
3	ACCG2K38YCACC3XV	OnePlus Bullets Wireless Z Bass Edition Blueto...	5	Super!
4	ACCG2K38YCACC3XV	OnePlus Bullets Wireless Z Bass Edition Blueto...	5	Super!

Figure 6: Loading the main dataset into the data frame

The figure above depicts the loading of the main dataset which is scrapped from the online shopping website called Flipkart into the data frame foe further analysis.



```
Rows     :  7469
Columns  :  9

Features :  ['product_id', 'product_title', 'rating', 'summary', 'review', 'location', 'date', 'upvotes', 'downvotes']

Missing values :   3980

Unique values :
 product_id        10
```

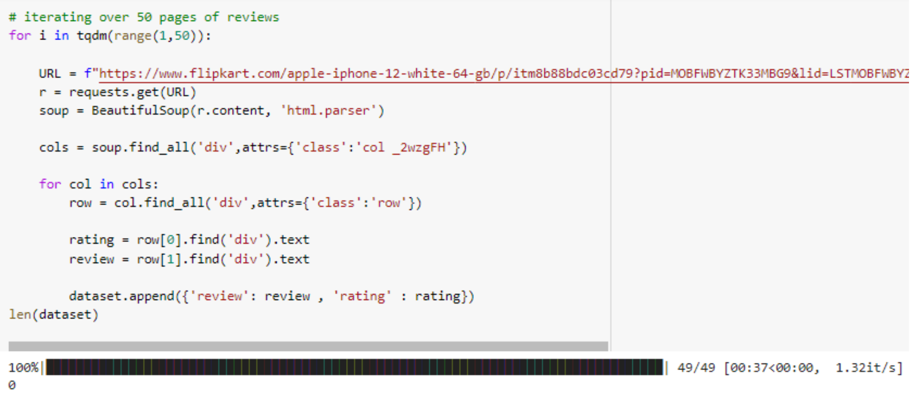


```
Unique values :
 product_id        10
product_title       8
rating              5
summary            85
review           4559
location         1024
date               63
upvotes           170
downvotes         102
dtype: int64
```

Figure 7: Unique data attributes analysis

The figure above depicts the Unique data attributes analysis from the main data frame.

3.3. Data Pre-processing

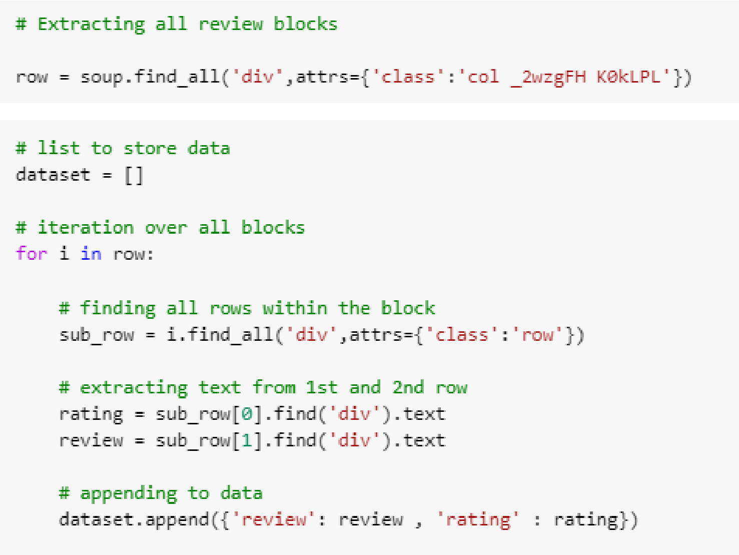


```
# iterating over 50 pages of reviews
for i in tqdm(range(1,50)):

    URL = f"https://www.flipkart.com/apple-iphone-12-white-64-gb/p/itm8b88bdc03cd79?pid=MOBFWBYZTK33MBG9&lid=LSTMOBFWBYZ
    r = requests.get(URL)
    soup = BeautifulSoup(r.content, 'html.parser')

    cols = soup.find_all('div',attrs={'class':'col _2wzgFH'})

    for col in cols:
        row = col.find_all('div',attrs={'class':'row'})

        rating = row[0].find('div').text
        review = row[1].find('div').text

        dataset.append({'review': review , 'rating' : rating})
len(dataset)
```

```
100%|██████████████████████████████████████████████████████████| 49/49 [00:37<00:00,  1.32it/s]
0
```

Figure 8: Data Scrapping from the Flipkart website for product review purposes

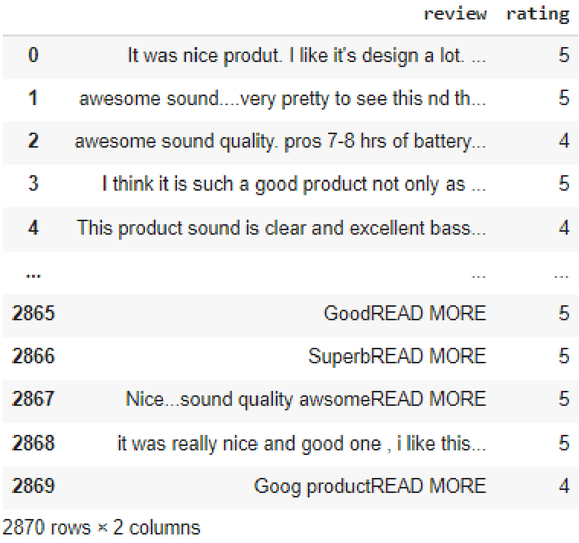
The figure above depicts the data scrapping from the Flipkart webpage for the product reviews purposes.



```
# Extracting all review blocks

row = soup.find_all('div',attrs={'class':'col _2wzgFH K0kLPL'})
```

```
# list to store data
dataset = []

# iteration over all blocks
for i in row:

    # finding all rows within the block
    sub_row = i.find_all('div',attrs={'class':'row'})

    # extracting text from 1st and 2nd row
    rating = sub_row[0].find('div').text
    review = sub_row[1].find('div').text

    # appending to data
    dataset.append({'review': review , 'rating' : rating})
```

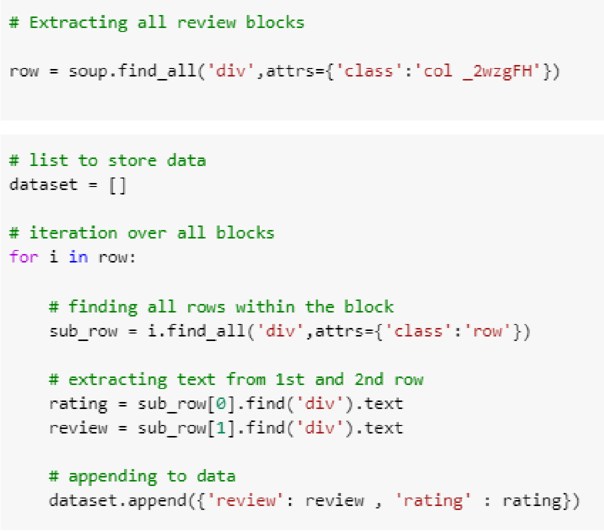
Figure 9: Reviews Extraction for product 1



	review	rating
0	It was nice produt. I like it's design a lot. ...	5
1	awesome sound....very pretty to see this nd th...	5
2	awesome sound quality. pros 7-8 hrs of battery...	4
3	I think it is such a good product not only as ...	5
4	This product sound is clear and excellent bass...	4
...	...	...
2865	GoodREAD MORE	5
2866	SuperbREAD MORE	5
2867	Nice...sound quality awsomeREAD MORE	5
2868	it was really nice and good one , i like this...	5
2869	Goog productREAD MORE	4

2870 rows × 2 columns

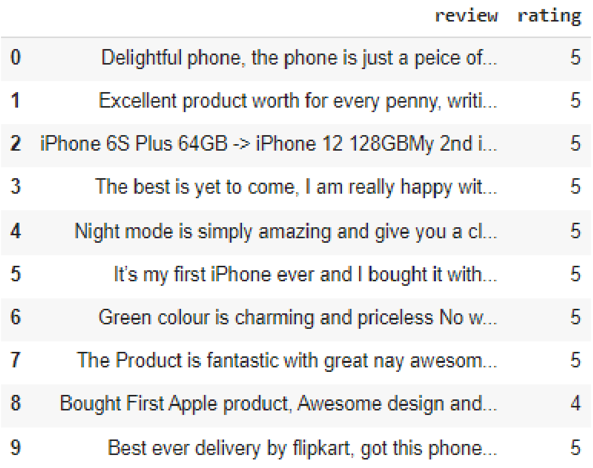
Figure 10: Review data frame for product 1



```
# Extracting all review blocks

row = soup.find_all('div',attrs={'class':'col _2wzgFH'})
```

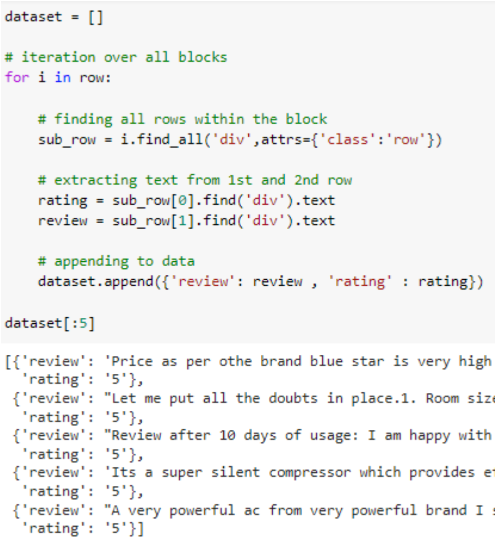
```
# list to store data
dataset = []

# iteration over all blocks
for i in row:

    # finding all rows within the block
    sub_row = i.find_all('div',attrs={'class':'row'})

    # extracting text from 1st and 2nd row
    rating = sub_row[0].find('div').text
    review = sub_row[1].find('div').text

    # appending to data
    dataset.append({'review': review , 'rating' : rating})
```

Figure 11: Reviews Extraction for product 2



	review	rating
0	Delightful phone, the phone is just a peice of...	5
1	Excellent product worth for every penny, writi...	5
2	iPhone 6S Plus 64GB -> iPhone 12 128GBMy 2nd i...	5
3	The best is yet to come, I am really happy wit...	5
4	Night mode is simply amazing and give you a cl...	5
5	It's my first iPhone ever and I bought it with...	5
6	Green colour is charming and priceless No w...	5
7	The Product is fantastic with great nay awesom...	5
8	Bought First Apple product, Awesome design and...	4
9	Best ever delivery by flipkart, got this phone...	5

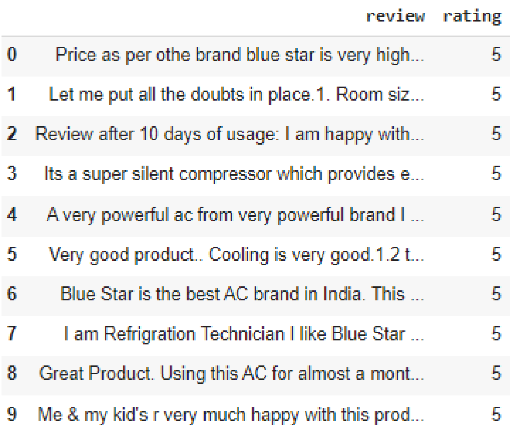
Figure 12: Review data frame for product 2



```
dataset = []

# iteration over all blocks
for i in row:

    # finding all rows within the block
    sub_row = i.find_all('div',attrs={'class':'row'})

    # extracting text from 1st and 2nd row
    rating = sub_row[0].find('div').text
    review = sub_row[1].find('div').text

    # appending to data
    dataset.append({'review': review , 'rating' : rating})

dataset[:5]
```

```
[{'review': 'Price as per othe brand blue star is very high
  'rating': '5'},
 {'review': "Let me put all the doubts in place.1. Room siz
  'rating': '5'},
 {'review': "Review after 10 days of usage: I am happy with
  'rating': '5'},
 {'review': 'Its a super silent compressor which provides e
  'rating': '5'},
 {'review': "A very powerful ac from very powerful brand I
  'rating': '5'}]
```

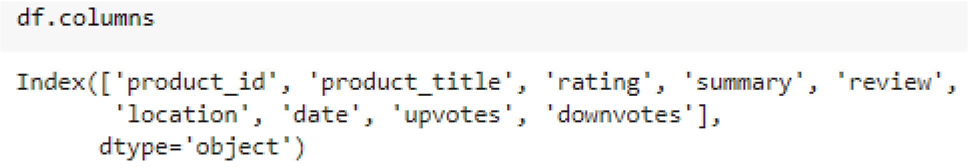
Figure 13: Reviews Extraction for product 3



	review	rating
0	Price as per othe brand blue star is very high...	5
1	Let me put all the doubts in place.1. Room siz...	5
2	Review after 10 days of usage: I am happy with...	5
3	Its a super silent compressor which provides e...	5
4	A very powerful ac from very powerful brand I ...	5
5	Very good product.. Cooling is very good.1.2 t...	5
6	Blue Star is the best AC brand in India. This ...	5
7	I am Refrigration Technician I like Blue Star ...	5
8	Great Product. Using this AC for almost a mont...	5
9	Me & my kid's r very much happy with this prod...	5

Figure 14: Review data frame for product 3

The figures above depicts the review of the data frame for product 1, 2 and 3 respectively along with reviews extraction code snippet.



```
df.columns
```

```
Index(['product_id', 'product_title', 'rating', 'summary', 'review',
       'location', 'date', 'upvotes', 'downvotes'],
      dtype='object')
```

Figure 15: Data Column analysis

The figure above depicts the data columns of the dataset namely, product ID, product title, rating, summary, review, location, date, upvotes and downvotes.

4. Model Building

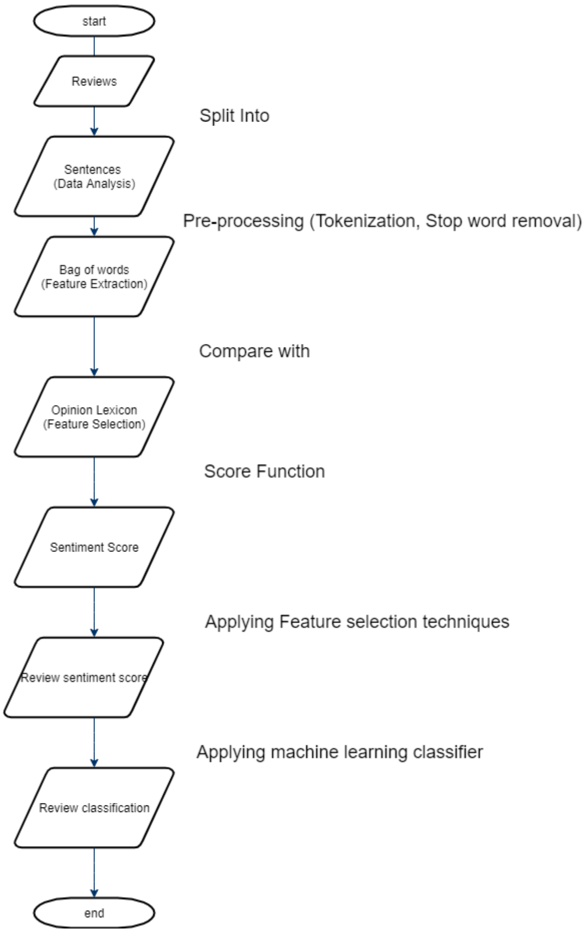


Figure 16:Research Methodology diagram

The figure above represents the research workflow that is followed for the research analysis completion.

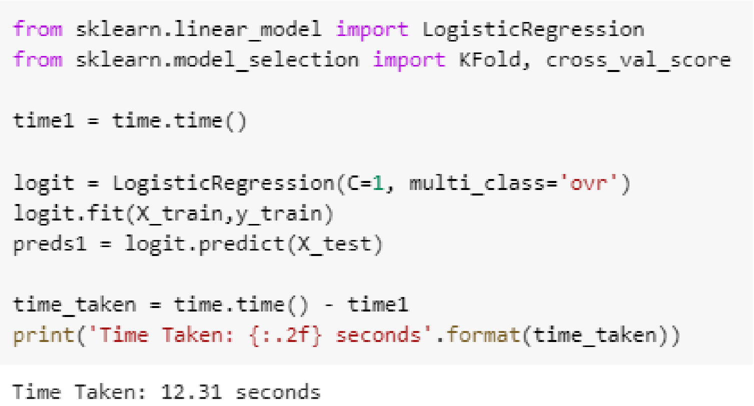
The workflow consists of scrapping the data in the form of product reviews from the online shopping website called Flipkart.

The reviews are scrapped using the beautiful soup API using the python language.

The reviews are gathered and collected in the comma separated file (CSV). Sentences, words and characters are reviewed from the dataset.

Data features are extracted in the form of bags of words and data feature selection is carried out using the opinion lexicon analysis.

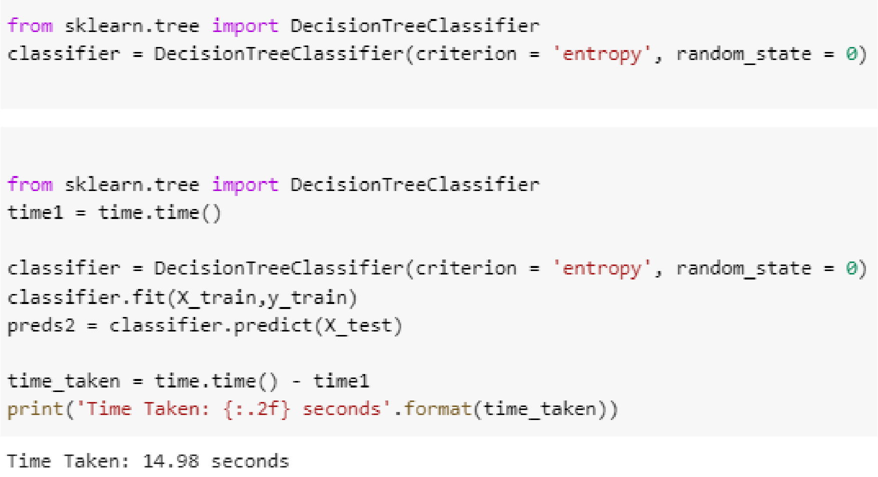
The score function is defined to compare and score the sentiment of the product reviews provided by the product users. Product reviews ratings are analyzed with the application of feature selection techniques. Finally, the reviews are classified using the machine learning classifiers which include supervised learning approach, regression approach and the deep learning approach using the neural network.



```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import KFold, cross_val_score

time1 = time.time()

logit = LogisticRegression(C=1, multi_class='ovr')
logit.fit(X_train,y_train)
preds1 = logit.predict(X_test)

time_taken = time.time() - time1
print('Time Taken: {:.2f} seconds'.format(time_taken))
```

```
Time Taken: 12.31 seconds
```

Figure 17: Model training with Logistic regression algorithm

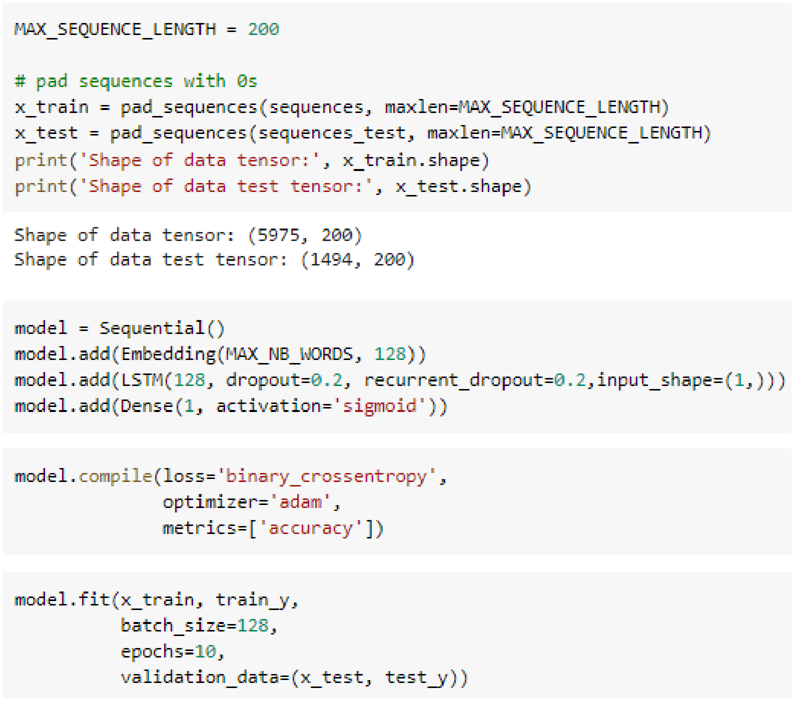


```
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
```

```
from sklearn.tree import DecisionTreeClassifier
time1 = time.time()

classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
classifier.fit(X_train,y_train)
preds2 = classifier.predict(X_test)

time_taken = time.time() - time1
print('Time Taken: {:.2f} seconds'.format(time_taken))
```

```
Time Taken: 14.98 seconds
```

Figure 18: Model training with Decision tree algorithm



```
MAX_SEQUENCE_LENGTH = 200

# pad sequences with 0s
x_train = pad_sequences(sequences, maxlen=MAX_SEQUENCE_LENGTH)
x_test = pad_sequences(sequences_test, maxlen=MAX_SEQUENCE_LENGTH)
print('Shape of data tensor:', x_train.shape)
print('Shape of data test tensor:', x_test.shape)
```

```
Shape of data tensor: (5975, 200)
Shape of data test tensor: (1494, 200)
```

```
model = Sequential()
model.add(Embedding(MAX_NB_WORDS, 128))
model.add(LSTM(128, dropout=0.2, recurrent_dropout=0.2,input_shape=(1,)))
model.add(Dense(1, activation='sigmoid'))
```

```
model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

```
model.fit(x_train, train_y,
          batch_size=128,
          epochs=10,
          validation_data=(x_test, test_y))
```

Figure 19: Model training with Neural network layers

4.1. Comparative Analysis

MODELS	ACCURACY VALUE
LOGISTIC REGRESSION MODEL	0.71
DECISION TREE MODEL	0.70
NEURAL NETWORK MODEL	0.97

From our work we have come to the conclusion that finding spam ideas in large amounts of unstructured data has become an important research problem. Although, some of the algorithms used in the spam analysis of ideas give good results, but still no algorithm can solve all the challenges and difficulties faced by today's generation. It is very important to consider specific quality standards such as usefulness, helpfulness and usability while analyzing each review. In literature research there are many complex explanations that describe the analysis of emotions in relation to various aspects. Our app that will help the user to pay for the right product without getting into any scams. Our work performed the analysis and map the genuine review to genuine product.

And the user can be sure about the product availability through the review prediction process. In the future we will try to improve the way we calculate sentimental feedback score. We will also try to update our data dictionary containing the sentiment words. We can try to add more words to our dictionary and revise the weights given to those words in order to get the most accurate counting points for updates. Sentimental analysis or opinion can be applied to any new classifier that follow the rules of data mining. Guide to future research is system utilization and performance evaluation using the proposed method for various measurement data sets. The main purpose of our work is to create a system that will receive spam and unwanted updates and filter them so that the user can understand the product information. The aim of our project is to improve customer satisfaction and make online shopping more reliable. The project will detect the fake reviews by incorporating mining algorithms like logistic regression classifier, Decision tree classifier and neural network classifier.

4.2. Error Analysis

Error analysis helps to isolate, verify and confirm erroneous ML estimates, thereby helping to understand the high and low performance of the model. The neural network gave the accuracy score of 97% but it varies in subgroup of the data so the model performance changes if the input conditions are varied leading to the failure of the overall model performance.

References

[1] Bist, J., Hulsurkar, N., Bhalerao, S., and Narkhede, D., 2020. Comment Sentiment Analysis and Fake Product Review Detection.Available on:https://www.academia.edu/download/64555513/IRJET-V7I594.pdf

[2] Danish, N.M., Tanzeel, S.M., Usama, N., Muhammad, A., Martinez-Enriquez, A.M. and Muhammad, A., 2019, September. Intelligent interface for fake product review monitoring and removal. In 2019 16th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE) (pp. 1-6). IEEE.Available on:https://ieeexplore.ieee.org/abstract/document/8884529/

[3] Sinha, A., Arora, N., Singh, S., Cheema, M., and Nazir, A., 2018. Fake product review monitoring using opinion mining. International Journal of Pure and Applied Mathematics, 119(12), pp.13203-13209.Available on:https://www.acadpubl.eu/hub/2018-119-12/articles/5/1203.pdf

[4] Wahyuni, E.D. and Djunaidy, A., 2016. Fake review detection from a product review using a modified method of iterative computation framework. In MATEC Web of conferences (Vol. 58, p. 03003). EDP Sciences.Available on:https://www.matec-conferences.org/articles/matecconf/abs/2016/21/matecconf_bisstech2016_03003/matecconf_bisstech2016_03003.html

[5] Boutaba, R., Salahuddin, M.A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F. and Caicedo, O.M., 2018. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *Journal of Internet Services and Applications*, *9*(1), pp.1-99.