APPENDIX A

Implementation of comparison between previous study and this stus is given below:

```
import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import nltk

from sklearn.preprocessing import LabelBinarizer

from nltk.corpus import stopwords

from nltk.stem.porter import PorterStemmer

from wordcloud import WordCloud,STOPWORDS

from nltk.stem import WordNetLemmatizer

from nltk.tokenize import word_tokenize,sent_tokenize

from bs4 import BeautifulSoup

import re,string,unicodedata

from keras.preprocessing import text, sequence

from sklearn.metrics import classification_report,confusion_matrix,accuracy_score

from sklearn.model_selection import train_test_split

from string import punctuation

import keras

from keras.models import Sequential

from keras.layers import Dense,Embedding,LSTM,Dropout,Bidirectional,GRU

import tensorflow as tf
```

```
from google.colab import drive
drive.mount('/content/drive')

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from keras.models import Model
from keras.layers import LSTM, Activation, Dense, Dropout, Input, Embedding
from keras.optimizers import RMSprop
from keras.optimizers import Adam
from keras.preprocessing.text import Tokenizer
from keras.preprocessing import sequence
from keras.utils import to_categorical

color=pd.read_csv("/content/drive/MyDrive/FurqanDurrani/color.csv")
size=pd.read_csv("/content/drive/MyDrive/FurqanDurrani/size.csv")
service=pd.read_csv("/content/drive/MyDrive/FurqanDurrani/services.csv")
weight=pd.read_csv("/content/drive/MyDrive/FurqanDurrani/weight.csv")
price=pd.read_csv("/content/drive/MyDrive/FurqanDurrani/price.csv")

dataF=color.append(size,ignore_index = True)
dataF=dataF.append(service,ignore_index = True)
dataF=dataF.append(weight,ignore_index = True)
dataF=dataF.append(price,ignore_index = True)
len(dataF)
```

```
color=dataF


color[:2]

data5=color.loc[color['sentiment'] == 0]

print(len(data5))

data4=color.loc[color['sentiment'] == 1]

print(len(data4))

data3=color.loc[color['sentiment'] == -1]

print(len(data3))


data5=data5[0:2380]

data4=data4[0:2380]

data3=data3[0:2380]


print(len(data5))

print(len(data4))

print(len(data3))


dataF=data4.append(data5,ignore_index = True)

dataF=dataF.append(data3,ignore_index = True)

len(dataF)


color["cleanText"]=color["cleanText"].astype(str)
```

```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer()

X_train_tf = vectorizer.fit_transform(color["cleanText"])

#X_test_tf = vectorizer.transform(X_test)

from imblearn.over_sampling import SMOTE

sm = SMOTE(random_state=2)

X_restf, y_restf = sm.fit_sample(X_train_tf, color["sentiment"].ravel())


print('After OverSampling, the shape of train_X: {}'.format(X_restf.shape))

print('After OverSampling, the shape of train_y: {} \n'.format(y_restf.shape))


from sklearn.model_selection import cross_val_score

from sklearn.ensemble import RandomForestClassifier

clf = RandomForestClassifier(n_estimators=200, max_depth=200)

scores = cross_val_score(clf, X_restf, y_restf, cv=5)

print("%0.2f accuracy with a standard deviation of %0.2f" % (scores.mean(), scores.std()))


from sklearn.model_selection import cross_val_score

from sklearn.naive_bayes import GaussianNB

clf = GaussianNB()


scores = cross_val_score(clf, X_restf.toarray(), y_restf, cv=5)

print("%0.2f accuracy with a standard deviation of %0.2f" % (scores.mean(), scores.std()))
```

```
from sklearn.model_selection import cross_val_score

from sklearn.ensemble import RandomForestClassifier

from sklearn.ensemble import RandomForestClassifier, VotingClassifier

from sklearn.linear_model import LogisticRegression

clf = RandomForestClassifier(n_estimators=200, max_depth=200)

lr = LogisticRegression(solver='saga',multi_class='multinomial',C=2.0)

vc=VotingClassifier(estimators=[('lr', lr), ('clf', clf),],voting='soft')

scores = cross_val_score(vc, X_restf, y_restf, cv=5)

print("%0.2f accuracy with a standard deviation of %0.2f" % (scores.mean(), scores.std()))


from keras import optimizers

import tensorflow.keras

!pip install -q keras_metrics

import nltk

nltk.download('stopwords')

import keras_metrics

tokenizer = tensorflow.keras.preprocessing.text.Tokenizer(num_words=5000,
lower=True,split=' ',filters='!"#$%&()*+,-./:;<=>?@[\\]^_`{|}~\t\n')

tokenizer.fit_on_texts(color["cleanText"].values)

#print(tokenizer.word_index)  # To see the dicstionary

X = tokenizer.texts_to_sequences(color["cleanText"].values)

X = tensorflow.keras.preprocessing.sequence.pad_sequences(X)

from sklearn.model_selection import train_test_split,KFold,cross_val_score

from keras.models import Sequential

from keras.layers import Activation, Dense,Flatten,LSTM,Dropout
```

```
kfold=KFold(2,True,7)

cvscores = []


from imblearn.over_sampling import SMOTE

sm = SMOTE(random_state=2)

X_restf, y_restf = sm.fit_sample(X, color["sentiment"].ravel())

Y = pd.get_dummies(y_restf).values

for train, test in kfold.split(X_restf, Y):


  X_train, X_test, y_train, y_test = train_test_split(X_restf, Y, test_size = 0.2, random_state = 0)

    #Deep Learning Network Structure

    model_conv = Sequential()

    model_conv.add(Embedding(5000,100, input_length=X.shape[1]))

    model_conv.add(Dropout(0.5))

    model_conv.add(LSTM(100))

    model_conv.add(Dense(3, activation='softmax'))

    model_conv.compile(loss="categorical_crossentropy", optimizer='adam',metrics =["mae"])

    model_conv.fit(X_train, y_train, epochs=10, batch_size=10, verbose=0,shuffle=False)

    scores = model_conv.evaluate(X_test, y_test, verbose=0)

    print("%s: %.2f%%" % (model_conv.metrics_names[1], scores[1]*100))

    cvscores.append(scores[1] * 100)

print("%.2f%% (+/- %.2f%%)" % (np.mean(cvscores), np.std(cvscores)))
```

```
 import nltk
  nltk.download('stopwords')
 >>> import nltk
 >>> nltk.download('wordnet')
import nltk
from nltk.tokenize import RegexpTokenizer
from nltk.stem import WordNetLemmatizer,PorterStemmer
from nltk.corpus import stopwords
import re
lemmatizer = WordNetLemmatizer()
stemmer = PorterStemmer()

def preprocess(sentence):
    sentence=str(sentence)
    sentence = sentence.lower()
    sentence=sentence.replace('{html}',"")
    cleanr = re.compile('<.*?>')
    cleantext = re.sub(cleanr, '', sentence)
    rem_url=re.sub(r'http\S+', '',cleantext)
    rem_num = re.sub('[0-9]+', '', rem_url)
    tokenizer = RegexpTokenizer(r'\w+')
    tokens = tokenizer.tokenize(rem_num)
    filtered_words = [w for w in tokens if len(w) > 2 if not w in stopwords.words('english')]
    stem_words=[stemmer.stem(w) for w in filtered_words]
```

```
   lemma_words=[lemmatizer.lemmatize(w) for w in stem_words]

   return " ".join(filtered_words)

color['cleanText']=color['Review Text'].map(lambda s:preprocess(s))

import nltk

from textblob import  TextBlob

color['Polarity Score']=""

color['sentiment']=""

#df2 = pd.DataFrame(columns=['text', 'sentiment', 'score'])

color['cleanText']=color['cleanText'].astype(str)

for i in range(len(color)):

   sentiment = TextBlob(color['cleanText'][i])

   a=sentiment.sentiment.polarity

   #df2.loc[i] = [data['cleanText'][i]]+[str(0)]+ [a]

   color["Polarity Score"][i]=a


for i in range(len(color)):

   if(color['Polarity Score'][i]>0):

      color['sentiment'][i]=1

   elif(color['Polarity Score'][i]==0):

      color['sentiment'][i]=0

   else:

      color['sentiment'][i]=-1


#color.to_csv("/content/drive/MyDrive/FurqanDurrani/price.csv")
```

```
color=dataF
color=color.dropna()
color=color.reset_index()

!pip install -q keras_metrics
import nltk
nltk.download('stopwords')
import keras_metrics

import nltk
from textblob import  TextBlob
color['Polarity Score1']=""
color['sentiment1']=""
#df2 = pd.DataFrame(columns=['text', 'sentiment', 'score'])
color['cleanText']=color['cleanText'].astype(str)
for i in range(len(color)):
   sentiment = TextBlob(color['cleanText'][i])
   a=sentiment.sentiment.polarity
   #df2.loc[i] = [data['cleanText'][i]]+[str(0)]+ [a]
   color["Polarity Score1"][i]=a

for i in range(len(color)):
   if(color['Polarity Score1'][i]>0):
```

```
        color['sentiment1'][i]=1
    elif(color['Polarity Score1'][i]==0):
        color['sentiment1'][i]=0
    else:
        color['sentiment1'][i]=-1
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(color["cleanText"],color["sentiment"].astype(int),test_size=0.20, random_state=200,shuffle=True)
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(color["cleanText"])
#y = vectorizer.transform(X_test)

from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer()
X1 = vectorizer.fit_transform(X_train)
y1 = vectorizer.transform(X_test)

from sklearn.ensemble import ExtraTreesClassifier
from sklearn.feature_selection import SelectFromModel
from sklearn.ensemble import RandomForestClassifier
extc1 = ExtraTreesClassifier(n_estimators=300, random_state=27, max_depth=50)
extc1.fit(X, color["sentiment"].astype(int))
sfm_extc = SelectFromModel(extc1,threshold=0.001)
```

```
# Train the selector

sfm_extc.fit(X,color["sentiment"].astype(int))

X_important_extc_train = sfm_extc.transform(X)

#X_important_extc_test = sfm_extc.transform(X_test_df)

from sklearn.svm import SVC

sv = SVC(kernel='linear', C=2.0, random_state=500)

sv.fit(X_important_extc_train, y_train)

# calculate accuracy of class predictions

y_pred_class = sv.predict(X_important_extc_test)

print(accuracy_score(y_test, y_pred_class))

print(classification_report(y_test, y_pred_class))

print(confusion_matrix(y_test, y_pred_class))


from sklearn.svm import SVC

clf = SVC(kernel='linear', C=2.0, random_state=500)

scores = cross_val_score(clf, X_important_extc_train, color["sentiment"], cv=5)

print("%0.2f accuracy with a standard deviation of %0.2f" % (scores.mean(), scores.std()))


from sklearn.neighbors import KNeighborsClassifier

sv = KNeighborsClassifier()

sv.fit(X, y_train)

# calculate accuracy of class predictions

y_pred_class = sv.predict(y)

print(accuracy_score(y_test, y_pred_class))
```

```
print(classification_report(y_test, y_pred_class))

print(confusion_matrix(y_test, y_pred_class))


from sklearn.neighbors import KNeighborsClassifier

clf = KNeighborsClassifier()

scores = cross_val_score(clf, X, color["sentiment"], cv=5)

print("%0.2f accuracy with a standard deviation of %0.2f" % (scores.mean(), scores.std()))


from sklearn.naive_bayes import GaussianNB

sv = GaussianNB()

sv.fit(X.toarray(), y_train)

# calculate accuracy of class predictions

y_pred_class = sv.predict(y.toarray())

print(accuracy_score(y_test, y_pred_class))

print(classification_report(y_test, y_pred_class))

print(confusion_matrix(y_test, y_pred_class))


from sklearn.naive_bayes import GaussianNB

clf = GaussianNB()

scores = cross_val_score(clf, X.toarray(), color["sentiment"], cv=5)

print("%0.2f accuracy with a standard deviation of %0.2f" % (scores.mean(), scores.std()))

from sklearn.svm import SVC

sv = SVC(kernel='linear', C=2.0, random_state=500)

sv.fit(X, y_train)
```

```
# calculate accuracy of class predictions

y_pred_class = sv.predict(y)

print(accuracy_score(y_test, y_pred_class))

print(classification_report(y_test, y_pred_class))

print(confusion_matrix(y_test, y_pred_class))


from sklearn.ensemble import RandomForestClassifier

rf=RandomForestClassifier(n_estimators=200, max_depth=200)

rf.fit(X_important_extc_train, y_train)

# calculate accuracy of class predictions

y_pred_class = rf.predict(X_important_extc_test)

print(accuracy_score(y_test, y_pred_class))

print(classification_report(y_test, y_pred_class))

print(confusion_matrix(y_test, y_pred_class))


from sklearn.ensemble import RandomForestClassifier

clf=RandomForestClassifier(n_estimators=200, max_depth=200)

scores = cross_val_score(clf, X.toarray(), color["sentiment"], cv=5)

print("%0.2f accuracy with a standard deviation of %0.2f" % (scores.mean(), scores.std()))

from sklearn.ensemble import RandomForestClassifier

rf=RandomForestClassifier(n_estimators=300, max_depth=300)

rf.fit(X, y_train)

# calculate accuracy of class predictions

y_pred_class = rf.predict(y)
```

```
print(accuracy_score(y_test, y_pred_class))

print(classification_report(y_test, y_pred_class))

print(confusion_matrix(y_test, y_pred_class))

from sklearn.linear_model import LogisticRegression

lr=LogisticRegression(random_state=1000, solver='liblinear',C=3.0)

lr.fit(X_important_extc_train, y_train)

# calculate accuracy of class predictions

y_pred_class = lr.predict(X_important_extc_test)

print(accuracy_score(y_test, y_pred_class))

print(classification_report(y_test, y_pred_class))

print(confusion_matrix(y_test, y_pred_class))

from sklearn.linear_model import LogisticRegression

lr=LogisticRegression(random_state=2,  multi_class="multinomial",C=3.0)

lr.fit(X, y_train)

# calculate accuracy of class predictions

y_pred_class = lr.predict(y)

print(accuracy_score(y_test, y_pred_class))

print(classification_report(y_test, y_pred_class))

print(confusion_matrix(y_test, y_pred_class))

from sklearn.linear_model import LogisticRegression

clf=LogisticRegression(random_state=2,  multi_class="multinomial",C=3.0)

scores = cross_val_score(clf, X, color["sentiment"], cv=5)

print("%0.2f accuracy with a standard deviation of %0.2f" % (scores.mean(), scores.std()))

from sklearn.model_selection import cross_val_score
```

```
from sklearn.ensemble import RandomForestClassifier

from sklearn.ensemble import RandomForestClassifier, VotingClassifier

from sklearn.linear_model import LogisticRegression

clf = RandomForestClassifier(n_estimators=300, max_depth=300)

lr = LogisticRegression(multi_class='multinomial',C=2.0)

vc=VotingClassifier(estimators=[('lr', lr), ('clf', clf),],voting='soft')

vc.fit(X_important_extc_train, y_train)

# calculate accuracy of class predictions

y_pred_class = vc.predict(X_important_extc_test)

print(accuracy_score(y_test, y_pred_class))

print(classification_report(y_test, y_pred_class))

print(confusion_matrix(y_test, y_pred_class))


from sklearn.model_selection import cross_val_score

from sklearn.ensemble import RandomForestClassifier

from sklearn.ensemble import RandomForestClassifier, VotingClassifier

from sklearn.linear_model import LogisticRegression

clf = RandomForestClassifier(n_estimators=300, max_depth=300)

lr = LogisticRegression(multi_class='multinomial',C=2.0)

vc=VotingClassifier(estimators=[('lr', lr), ('clf', clf),],voting='soft')

vc.fit(X, y_train)

# calculate accuracy of class predictions

y_pred_class = vc.predict(y)

print(accuracy_score(y_test, y_pred_class))
```

```
print(classification_report(y_test, y_pred_class))

print(confusion_matrix(y_test, y_pred_class))


from sklearn.model_selection import cross_val_score

from sklearn.ensemble import RandomForestClassifier

from sklearn.ensemble import RandomForestClassifier, VotingClassifier

from sklearn.linear_model import LogisticRegression

clf = RandomForestClassifier(n_estimators=300, max_depth=300)

lr = LogisticRegression(multi_class='multinomial',C=2.0)

vc=VotingClassifier(estimators=[('lr', lr), ('clf', clf),],voting='soft')

scores = cross_val_score(vc, X, color["sentiment"], cv=5)

print("%0.2f accuracy with a standard deviation of %0.2f" % (scores.mean(), scores.std()))
```

sentimental words with their polarities [28][13] [12]. Generally, they collect sentimental words from phrases.

Based on scattered information like strength and polarities of sentimental words, they classify the sentences in classes of sentiments with help of polarities[29] [30]. Moreover, the lexicon-based models are efficient and simple but manually sentiment lexicon creation is a time-consuming and labour-intensive job. Secondly, already static polarity is required for every sentence. For this solution, some kinds of models that automatically generate sentiment lexicons have been proposed [31][32].

Like the sentence "Sound quality of Techno mobile is not so good.". In this sense of lexicon-based approach, this sentence expresses the negative behaviour. But in the sentence "Sound quality of Samsung mobile is good." the good expresses the positive sentiment towards the sound quality of Samsung mobile. For this solution, some kind of machine learning-based models are still available.

To understand this aspect level problem, some kinds of reviews have multiple useful meanings. Like this sentence "Samsung is a good brand of mobile" in this specific sentence, clear positive opinion can be extracted and for example “techno brand of mobile is not a good brand”. In this sentence, we can extract the negative review of the consumer. But what in the case when the user shares the opinion like “Samsung is a good brand but Techno is not a good brand in the same sentence”. In this sample sentence, both negative and positive opinions are extracted.

This kind of problem can be resolved by classifying the sentiment analysis in the following techniques mentioned in figure 1.2 [33].

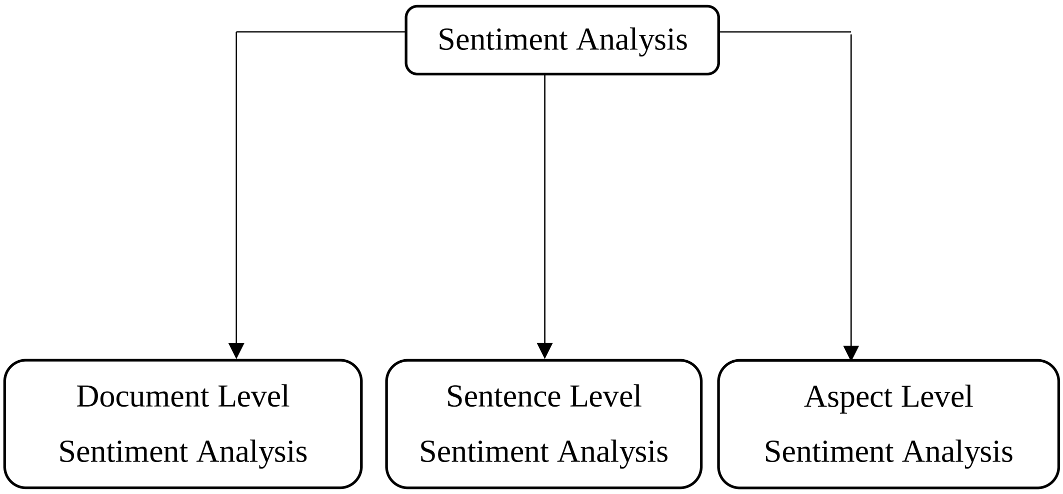

Figure 1.2 Types of Sentiment Analysis

1.2.1 Document Level Sentiment Analysis

Document-level sentiment analysis is said to be the analysis of the whole document. In this approach, the complete document is considered as a single entity and it is analysed at once. The opinion about the whole document is considered as positive or negative. However, this is not a good approach because there may be e a positive specific review that has a great importance but the overall sentiment score of the document is negative and vice versa [28] [34].

1.2.2 Sentence Level Sentiment Analysis

Sentiment analysis at the sentence level is considered as the calculation of sentiment of each of the sentences in the document. In this approach, the document is divided into sentences, and every sentence is considered as an entity. This is a better way to find the sentiment clarity as compared to the whole document because in this technique every sentence is analysed separately. Anyhow this is also not the best case to find the sentiment because referring to the above example Samsung is a good brand but that techno is not a good brand. In the above examples, we can extract the multiple meanings.

To overcome this issue the aspect level sentiment analysis has been proposed [35][36][37].

1.2.3 Aspect Level Sentiment Analysis

Aspect level sentiment analysis is said to be the analysis in which every feature or aspect is considered as an entity like price, size, and weight of mobile. A feature is said to be the instance or attribute of anything. In this approach, the main focus is to find out the feature of an entity and to find out the sentiment according to the feature. Aspect level sentiment analysis has been performed in many fields so far like explain in [38][39] researches.

1.3 Machine Learning

In 1997 researcher defines machine learning as the feature of computer science that aims to gain knowledge from data [25]. Machine learning is used to improve the efficiency of different analyses for example in applied Health Care and Emotion Detection etc. This is used to automate the process of flexibility and efficiency that identifies the trends from Complex data sets [40].

There are multiple steps involved to determine when ML is being used. The first step is that the machine learning technique can be used to answer the research question. In research [41], the researcher defines the three types of research problems i.e. Descriptive research, Explanatory research, and Predictive research which can be resolved through machine learning. Machine learning has been performed in many other fields and it is verified by the statistical methods which are sufficient in some cases and sometimes the questions validate the results. With respect to the working, Machine learning is divided into the following three types mentioned in figure 1.3 [33].

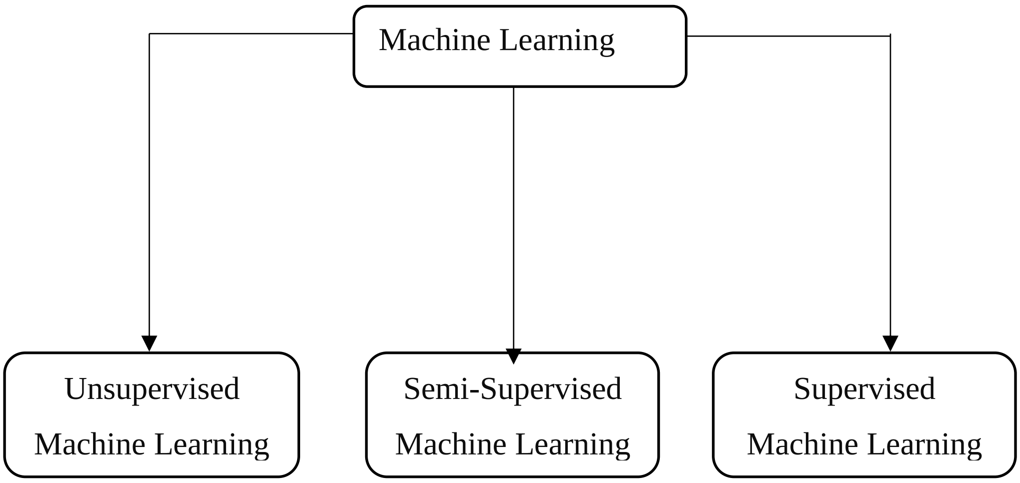

Figure 1.3 Types of Machine Learning

1.3.1 Unsupervised Machine Learning

Unsupervised machine learning is specifically helpful for descriptive research because this research aims to find the relationship between the data structure without knowing the target outcomes. This methodology is referred to unsupervised learning because we don't have any target variable that could be happened [42].

The main purpose of unsupervised learning is to identify or analyse the dimensions of the component's trajectories for clusters from the dataset. Multiple approaches for unsupervised learning are used i.e. Factor analysis, mixture modelling, and component analysis.

Unsupervised learning is used to find trends from the dataset. There are two main types of unsupervised learning which are commonly used, principal component analysis and cluster analysis. The cluster analysis is used to achieve different qualitative groups of individuals. Principal component analysis can be used to learn the large numbers of neurons. This approach is often used as pre-processed data or to reduce the size of the forecaster from big data.

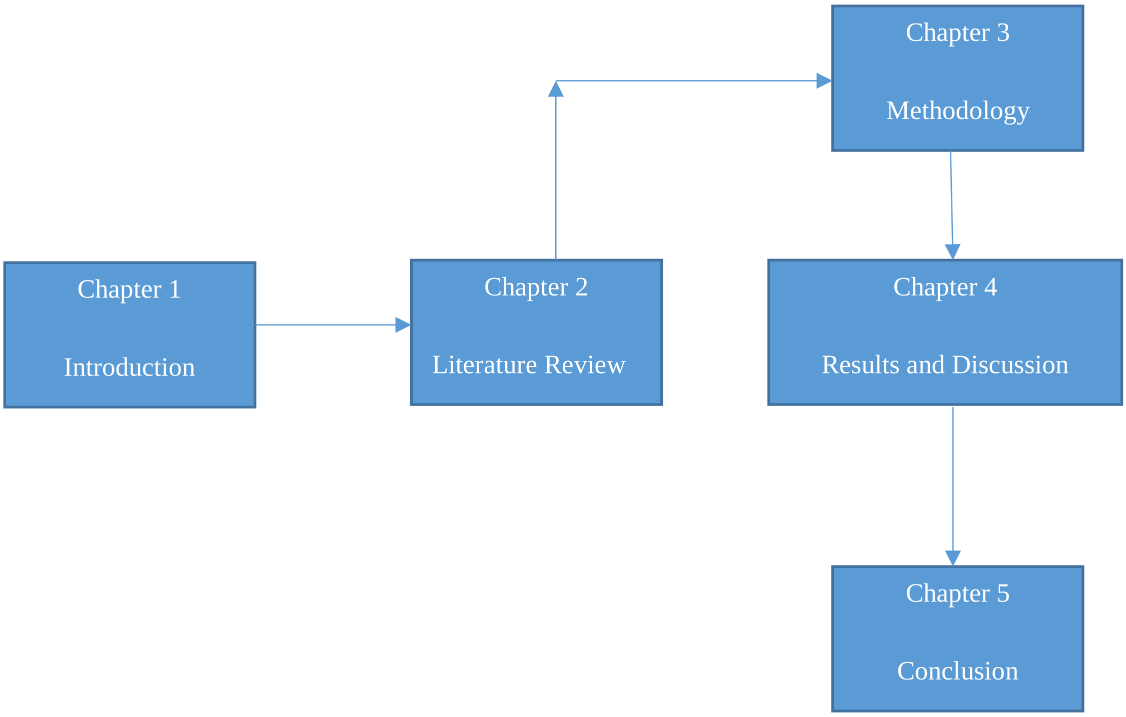

Figure 1.4 Thesis Organization

A recall is the ratio of correctly predicted positive variables by the total variables of an actual class. It can also be called, the rate of true positive in the total number of positive samples. In binary classification, recall is the sensitivity of the classifier.

Recall refers to the percentage of the total relevant results that were correctly classified by the algorithm.

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \qquad \text{Eq. 4.3}$$

4.1.4 F-Measure

F-score is the weighted harmonic mean of precision and recall. It reaches the best value which means perfect precision and recall.

$$F\ measure = 2 * \frac{\mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \qquad \text{Eq. 4.4}$$

4.2 Logistic Regression Results

This section contains the results for the Logistic Regression with the under sampling, oversampling, and without sampling technique. Performance of Logistic Regression improves with the oversampling technique because oversampling generates more features for the learning of Logistic Regression while the performance of Logistic Regression on imbalanced dataset show poor performance because the ratio of target classes are not equal and Logistic Regression gets overfitting on majority class data and show poor performance on minority class data. Under sampling reduces the performance of LR because under sampling randomly deletes the records for data which causes the reduction of features and model get under-fitted and reduce the accuracy. Table 4.1

Table 4.2 shows the results using the 10-fold cross-validation after oversampling which shows the significance of Logistic Regression. With 10-fold cross-validation after oversampling, LR acquired 0.95 validation scores on the size aspect, 0.94 validation score on the color aspect, 0.95 scores on the price aspect, 0.94 and 0.95 validation scores on service and weight aspects respectively. This shows the robustness and effectiveness of Logistic Regression with oversampling since oversampling provides a larger feature set for Logistic Regression to extract information from this performing with better accuracy results.

4.3 Random Forest Results

Random Forest is an ensemble model that uses different decision trees to make a final prediction and uses the bootstrap method to train each decision tree in a random forest. Bootstrap method randomly selects the records from dataset to make a subsample dataset for decision tree so in this way random forest can be good somehow on the imbalanced and small size of datasets.

Random Forest achieved the highest accuracy of 0.97 with oversampling on all aspects except size and service. Random Forest shows a better result as compared to the Logistic Regression. The lowest accuracy is achieved on the color aspect dataset using Random Forest and under sampling. Table 4.3 shows the results of Random Forest for aspect-based sentiment analysis on review datasets.

From the results obtained by Random Forest on the size aspect of the review dataset, Random Forest achieved the highest accuracy of 0.96 with similar scores for precision, recall, and f1-score with oversampling. Whereas, it achieved 0.78 accuracy without sampling and under sampling. Random Forest follows the splitting of a random subset of a feature set thus introducing correlation among the trees. With oversampling the increase in the size of the feature set is leveraged by RF to produce better results.

REFERENCES

1. C. Campbell, L. F. Pitt, M. Parent, and P. R. Berthon, 'Understanding consumer conversations around ads in a Web 2.0 world', *J. Advert.*, vol. 40, no. 1, pp. 87–102. 2011
2. M. Tichindelean and S. Vinerean,'Disclosing the Promising Power of Social Media – an Important Digital Marketing Tool', *Stud. Bus. Econ.*, vol. 6, no. 1, pp. 37–45. 2011
3. S. F. Bernritter, P. W. J. Verlegh, and E. G. Smit,'Why Nonprofits Are Easier to Endorse on Social Media: The Roles of Warmth and Brand Symbolism', *J. Interact. Mark.*, vol. 33, pp. 27–42, 2016.
4. E. Maslowska, E. C. Malthouse, and S. F. Bernritter,'Too good to be true: The role of online reviews' features in probability to buy', *Int. J. Advert.*, vol. 36, no. 1, pp. 142–163, 2017.
5. G. Cui, H. K. Lui, and X. Guo,'The effect of online consumer reviews on new product sales', *Int. J. Electron. Commer.*, vol. 17, no. 1, pp. 39–58, 2012 .
6. J. Lee, D. H. Park, and I. Han,'The different effects of online consumer reviews on consumers' purchase intentions depending on trust in online shopping malls: An advertising perspective', *Internet Res.*, vol. 21, no. 2, pp. 187–206, 2011.
7. Sulin Ba and Paul A. Pavlou,'Evidence of the Effect of Trust Building Technology in Electronic Markets: Price Premiums and Buyer Behavior', vol. 26, no. 3, pp. 243–268, 2002.
8. T. Ahmad and M. N. Doja, 'Ranking System for Opinion Mining of Features from Review Documents', *Int. J. Comput. Sci. Issues*, vol. 9, no. 4, pp. 440–447, 2012
9. S. Aravindan, Feature Extraction and Opinion Mining in Online Product Reviews'. 2014
10. H. Xia, Y. Yang, X. Pan, Z. Zhang, and W. An,'Sentiment analysis for online reviews using conditional random fields and support vector machines', *Electron. Commer. Res.*, vol. 20, no. 2, pp. 343–360, 2020
11. G. Choi, S. Oh, and H. Kim,'Improving document-level sentiment classification using importance of sentences', *Entropy*, vol. 22, no. 12, pp. 1–11. 2020

12. S. Baccianella, A. Esuli, and F. Sebastiani,‘SENTIWORDNET 3.0: An enhanced lexical resource for sentiment analysis and opinion mining’, *Proc. 7th Int. Conf. Lang. Resour. Eval. Lr. 2010*, vol. 0, pp. 2200–2204. 2010

13. M. Hu and B. Liu, 2004 ‘Mining and summarizing customer reviews’, *KDD-Proc. Tenth ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 168–177. *2004*

14. A. A. K. Author and P. P. A. D. Co-author,‘Review Analyzer Analysis of Product reviews on WEKA Classifiers’. 2015 .

15. A. Ceron, L. Curini, S. M. Iacus, and G. Porro,‘Every tweet counts? How sentiment analysis of social media can improve our knowledge of citizens’ political preferences with an application to Italy and France’, *New Media Soc.*, vol. 16, no. 2, pp. 340–358. 2014.

16. A. K. Swain and R. Q. Cao,‘Using sentiment analysis to improve supply chain intelligence’, *Inf. Syst. Front.*, vol. 21, no. 2, pp. 469–484. 2019

17. Q. A. Al-Radaideh and G. Y. Al-Qudah,‘Application of Rough Set-Based Feature Selection for Arabic Sentiment Analysis’, *Cognit. Comput.*, vol. 9, no. 4, pp. 436–445. 2017.

18. E. S. Tellez, S. Miranda-Jiménez, M. Graff, D. Moctezuma, O. S. Siordia, and E. A. Villaseñor,‘A case study of Spanish text transformations for twitter sentiment analysis’, *Expert Syst. Appl.*, vol. 81, pp. 457–471. 2017.

19. S. K. Sahu, P. Behera, D. P. Mohapatra, and R. C. Balabantaray,‘Sentiment analysis for Odia language using supervised classifier: an information retrieval in Indian language initiative’, *CSI Trans. ICT*, vol. 4, no. 2–4, pp. 111–115. 2016

20. Z. Nasim and S. Haider,‘ABSA Toolkit: An Open Source Tool for Aspect Based Sentiment Analysis’, *Int. J. Artif. Intell. Tools*, vol. 26, no. 6. 2018

21. M. Al-Smadi, M. Al-Ayyoub, H. Al-Sarhan, and Y. Jararweh,‘Using Aspect-Based Sentiment Analysis to Evaluate Arabic News Affect on Readers’, *Proc. - 2015 IEEE/ACM 8th Int. Conf. Util. Cloud Comput. UCC 2015*, pp. 436–441. 2015.

22. T. H. Nguyen, K. Shirai, and J. Velcin,‘Sentiment analysis on social media for stock movement prediction’, *Expert Syst. Appl.*, vol. 42, no. 24, pp. 9603–9611. 2015.

23. K. Ravi and V. Ravi, *A survey on opinion mining and sentiment analysis: Tasks,*

approaches and applications, vol. 89, no. June. Elsevier B.V. 2015.

24. J. Serrano-Guerrero, J. A. Olivas, F. P. Romero, and E. Herrera-Viedma, ‘Sentiment analysis: A review and comparative analysis of web services’, *Inf. Sci. (Ny* , vol. 311, pp. 18–38. 2015
25. F. H. Khan, U. Qamar, and S. Bashir, ‘A semi-supervised approach to sentiment analysis using revised sentiment strength based on SentiWordNet’, *Knowl. Inf. Syst.*, vol. 51, no. 3, pp. 851–872. 2017
26. M. Bavakhani and A. O. Mining, ‘A Deep Learning Approach for Extracting Polarity from Customers ’ Reviews’, *2019 5th Int. Conf. Web Res.*, pp. 276–280. 2019
27. A. Bhatti, H. Akram, H. M. Basit, A. U. Khan, and S. M. Raza, ‘E-commerce trends during COVID-19 Pandemic’, vol. 13, no. 2, pp. 1449–1452. 2020
28. P. Hoffmann,‘Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis’, no. October, pp. 347–354. 2005
29. R. Passonneau, ‘Sentiment Analysis of Twitter Data’, no. June, pp. 30–38, 2011.
30. C. Zhang, D. Zeng, and F. Wang,‘Sentiment Analysis of Chinese Documents : From’, vol. 60, no. 12, pp. 2474–2487. 2009
31. F. Alqasemi, A. Abdelwahab, and H. Abdelkader,‘Constructing automatic domain-specific sentiment lexicon using KNN search via terms discrimination vectors’, *Int. J. Comput. Appl.*, vol. 7074, no. December, pp. 1–11. 2017
32. M. Song, H. Park, and K. Shin,‘Attention-based long short-term memory network using sentiment lexicon embedding for aspect-level sentiment analysis in Korean’, *Inf. Process. Manag.*, vol. 56, no. 3, pp. 637–653. 2019
33. S. Vanaja, 2018 ‘Aspect-Level Sentiment Analysis on E-Commerce Data’,*Int. Conf. Inven. Res. Comput. Appl.*, no. Icirca, pp. 1275–1279. *2018*
34. D. Mohey and E. M. Hussein,‘A survey on sentiment analysis challenges’, *J. KING SAUD Univ. - Eng. Sci.*, no. April. 2016
35. A. Fölling, J. Lepping, A. Fölling, and J. Lepping, ‘grids To cite this version : Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery’. 2012
36. A. Shoukry and A. Rafea,‘Sentence-Level Arabic Sentiment Analysis’, pp. 546–550. 2012
37. Z. Bošnjak, ‘Sentiment Analysis of Customer Data’, vol. 23, no. 3, pp. 38–49. 2018
38. K. Schouten and F. Frasincar, ‘Survey on Aspect-Level Sentiment Analysis’, vol.

4347, no. c, pp. 1–20. 2015

39. T. M. Mitchell, 2010 'Introduction to Machine Learning'.

40. E. L. Hamaker, J. D. Mulder, and M. H. Van Ijzendoorn,'Developmental Cognitive Neuroscience Description , prediction and causation : Methodological challenges of studying child and adolescent development', *Dev. Cogn. Neurosci.*, vol. 46, no. September, p. 100867. 2020

41. X Zhu, AB Goldberg, *Introduction to Semi-Supervised Learning*. . 2009

42. G. James, *An Introduction to Statistical Learning*. 2013

43. I. A. Silva, A. Boukerche, T. R. M. B. Silva, L. B. Ruiz, E. Cerqueira, and A. A. F. Loureiro,'Vehicular Networks : A New Challenge for Content-Delivery-Based', vol. 49, no. 1. 2016

44. I. R. Galatzer-levy and I. R. Galatzer-levy,'Applications of Latent Growth Mixture Modeling and allied methods to posttraumatic stress response data Applications of Latent Growth Mixture Modeling and allied methods to posttraumatic stress response data', vol. 8198, no. September, pp. 11–13. 2017

45. R. C. Kessler, R. M. Bossarte, A. Luedtke, A. M. Zaslavsky, and J. R. Zubizarreta, 'Behaviour Research and Therapy Machine learning methods for developing precision treatment rules with observational data', *Behav. Res. Ther.*, vol. 120, no. February, p. 103412. 2019

46. J. L. Gradus *et al.*,(X 'Prediction of Sex-Specific Suicide Risk Using Machine Learning and Single-Payer Health Care Registry Data From Denmark', vol. 02118, pp. 1–10. 2019

47. W. Medhat, A. Hassan, and H. Korashy,'Sentiment analysis algorithms and applications: A survey', *Ain Shams Eng. J.*, vol. 5, no. 4, pp. 1093–1113. 2014

48. hi-Hua Zhou, brief introduction to weakly supervised learning. 2017

49. S. B. K. I. D. Z. P. E. Pintelas,'Machine learning : a review of classification and combining techniques', no. 2006, pp. 159–190, 2007

50. J. R. Quinlan,'Induction of Decision Trees', pp. 81–106. 2007

51. D. Grossman and P. Domingos,'Learning Bayesian Network Classifiers by Maximizing Conditional Likelihood'. 2004

52. S. R. Safavian and D. Landgrebe,'A Survey of Decision Tree Classifier Methodology', *IEEE Trans. Syst. Man Cybern.*, vol. 21, no. 3, pp. 660–674. 1991

53. H. Bhavsar and A. Ganatra,'A Comparative Study of Training Algorithms for Supervised Machine Learning', no. January 2012. 2016

54. KP Soman, S Diwakar, V Ajay,'Insight into Data Mining Theory and Practice'. 2006
55. K. N. Stevens, T. M. Cover, and P. E. Hart,'Nearest Neighbor', vol. I. 1967
56. D. Mining, K. Discovery, B. Laboratories, and L. Technologies, 'A Tutorial on Support Vector Machines for Pattern Recognition', vol. 167, pp. 121–167. 1998
57. M. L. Zhang and Z. H. Zhou, 'ML-KNN: A lazy learning approach to multi-label learning', *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, 2007, doi: 10.1016/j.patcog.2006.12.019.
58. Q. T. Ain *et al.*,'Sentiment Analysis Using Deep Learning Techniques : A Review', vol. 8, no. 6. 2017
59. P. Vateekul and T. Koomsubha,'A Study of Sentiment Analysis Using Deep Learning Techniques on Thai Twitter Data'. 2016
60. W. S. Noble and P. Street,'What is a support vector machine ?', vol. 24, no. 12, pp. 1565–1567. 2006
61. V. Y. Kulkarni, P. K. Sinha, and A. E. Classifiers,'Pruning of Random Forest Classifiers : A Survey and Future Directions', pp. 64–68. 2012
62. F. Rustam, A. Mehmood, M. Ahmad, S. Ullah, D. M. Khan, and G. Y. U. S. Choi, 'Classification of Shopify App User Reviews Using Novel Multi Text Features', vol. 8. 2020
63. F. Y. Osisanwo, J. E. T. Akinsola, O. Awodele, J. O. Hinmikaiye, O. Olakanmi, and J. Akinjobi,'Supervised Machine Learning Algorithms : Classification and Comparison', no. July,. 2017
64. E. Ogunfiditimi, F. and Oguntade, 'Logistic Regression : A Paradigm for Dichotomous Response Data', *Int. J. Engeneeing Sci.*, vol. 3, no. 6, pp. 1–5. 2014
65. D. Srilatha, D. P. Rao, C. Nikhil, and J. S. Tejumade,'Classifying Fake News Articles Using Machine Learning Techniques', pp. 162–167. 2020
66. B. Kovács, F. Tinya, C. Németh, and P. Ódor,'Unfolding the effects of different forestry treatments on microclimate in oak forests: results of a 4-yr experiment', *Ecol. Appl.*, vol. 30, no. 2, pp. 321–357. 2020
67. A. Nawaz, A. Ahmed, A. Tariq, A. Muhammad, and R. Rashid,'Product ' s behaviour recommendations using free text : an aspect based sentiment analysis approach', *Cluster Comput.*, vol. 23, no. 2, pp. 1267–1279. 2020
68. A. S. Manek, P. D. Shenoy, and M. C. M. V. K. R,'Aspect term extraction for sentiment analysis in large movie reviews using Gini Index feature selection

method and SVM classifier', pp. 135–154,2017

69. B. R. Aditya,'Measuring e-Commerce service quality from online customer review using sentiment analysis Measuring e-Commerce service quality from online customer review using sentiment analysis'. 2018
70. M. S. Elli and Y. Wang, 'Amazon Reviews, business analytics with sentiment analysis', *Acta Archaeol.*, vol. 77, no. 1–2, pp. 62–90. 2006
71. Y. Xu, X. Wu, and Q. Wang,'Sentiment Analysis of Yelp's Ratings Based on Text Reviews', *2015 17th Int. Symp. Symb. Numer. Algorithms Sci. Comput.* 2015
72. A. Bhatt, A. Patel, H. Chheda, and K. Gawande,'Amazon Review Classification and Sentiment Analysis', *Int. J. Comput. Sci. Inf. Technol.*, vol. 6, no. 6, pp. 5107–5110. 2015
73. B. Lewis, I. Smith, M. Fowler, and J. Licato,'The robot mafia: A test environment for deceptive robots', *28th Mod. Artif. Intell. Cogn. Sci. Conf. MAICS 2017*, pp. 189–190. 2017
74. T. Shaikh and D. Deshpande,'Feature Selection Methods in Sentiment Analysis and Sentiment Classification of Amazon Product Reviews', *Int. J. Comput. Trends Technol.*, vol. 36, no. 4, pp. 225–230, 2016
75. E. M. S. Mona Mohamed Nasr and Ahmed Mostafa Hafez,'Building Sentiment analysis Model using Graphlab', *Int. J. Sci. Eng. Res.*, vol. 8, no. 6, pp. 1155–1160. 2017
76. M. Wang, 'Text Mining for Yelp Dataset Challenge', pp. 1–5.
77. Y. Liu and X. Zhu,'Short text sentiment classification based on feature extension and ensemble classifier', *AIP Conf. Proc.*, vol. 1967. 2018
78. A. R. Nabhan and K. Shaalan,'Text Graphlet Patterns', vol. 2, no. June, pp. 152–161.
79. L. Xu, J. Liu, L. Wang, and C. Yin,'Aspect based sentiment analysis for online reviews', *Lect. Notes Electr. Eng.*, vol. 474, pp. 475–480. 2018
80. D. Anand and D. Naorem,'Semi-supervised Aspect Based Sentiment Analysis for Movies Using Review Filtering', *Procedia Comput. Sci.*, vol. 84, pp. 86–93. 2016
81. M. Soleymani, M. N. Caro, E. M. Schmidt, C. Y. Sha, and Y. H. Yang,'1000 Songs for Emotional Analysis of Music', *CrowdMM 2013 - Proc. 2nd ACM Int. Work. Crowdsourcing Multimed.*, pp. 1–6, 2013
82. G. M. M. Sarria, J. Diaz, and C. Arce-Lopera,'Analyzing and Extending the Salsa Music Dataset', *2019 22nd Symp. Image, Signal Process. Artif. Vision, STSIVA*

2019 - Conf. Proc., pp. 1–5. 2019

83. J. Hendler,‘Data integration for heterogenous datasets’, *Big Data*, vol. 2, no. 4, pp. 205–215. 2014
84. E. Triantafillou *et al.*,‘L EARNING TO L EARN FROM F EW E XAMPLES’. 2020
85. B. Gopal, P. Dipankar, and S. Bandyopadhyay,‘Multimodal mood classification of Hindi and Western songs’. 2018
86. M. Bilal, H. Israr, M. Shahid, and A. Khan,‘Sentiment classification of Roman-Urdu opinions using Navie Baysian, Decision Tree and KNN classification techniques’, *J. KING SAUD Univ. - Comput. Inf. Sci.* 2015
87. W. P. Wong, M. Lo, and T. Ramayah,‘THE EFFECTS OF TECHNOLOGY ACCEPTANCE FACTORS ON CUSTOMER E-LOYALTY AND E-SATISFACTION IN MALAYSIA’, vol. 15, no. 3, pp. 477–502. 2014
88. P. Patil and P. Yalagi,‘Sentiment Analysis using Aspect Level Classification’, vol. 2, no. 4, pp. 23–27. 2016
89. A. Rextin, A. Kakakhel, and M. Nasim,‘Analyzing Emergent Users ’ Text Messages Data and Exploring Its Benefits’, *IEEE Access*, vol. 7, pp. 2870–2879. 2019
90. G. Z. Nargis and N. Jamil,‘Generating an Emotion Ontology for Roman Urdu Text’, vol. 7, no. 3, pp. 83–91. 2016
91. S. Vijayarani, M. J. Ilamathi, and M. Nithya,‘Preprocessing Techniques for Text Mining - An Overview Preprocessing Techniques for Text Mining - An Overview’, no. February 2015. 2020
92. I. Javed and H. Afzal,‘Opinion Analysis of Bi-Lingual Event Data from Social Networks’. 2013
93. B. R. Aditya, A. R. Sulthana, A. K. Jaithunbi, and L. Sai,‘Aspect level sentiment analysis using machine learning’. 2017
94. D. Kalita,‘Supervised and Unsupervised Document Classification-A survey’, vol. 6, no. 2, pp. 1971–1974. 2015
95. M. P. Abraham and K. R. Udaya Kumar Reddy,‘Feature based sentiment analysis of mobile product reviews using machine learning techniques’, *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 2, pp. 2289–2296. 2020
96. A. S. Manek, P. D. Shenoy, M. C. Mohan, and K. R. Venugopal,‘Aspect term extraction for sentiment analysis in large movie reviews using Gini Index feature

selection method and SVM classifier', *World Wide Web,* vol. 20, no. 2, pp. 135–154. 2017

97. P. Pankaj, P. Pandey, M. Muskan, and N. Soni,'Sentiment Analysis on Customer Feedback Data: Amazon Product Reviews', *Proc. Int. Conf. Mach. Learn. Big Data, Cloud Parallel Comput. Trends, Prespectives Prospect. Com. 2019*, pp. 320–322. 2019

98. X. Wang, X. Chen, M. Tang, T. Yang, and Z. Wang,'Aspect-Level Sentiment Analysis Based on Position Features Using Multilevel Interactive Bidirectional GRU and Attention Mechanism', vol. 2020. 2020

99. P. K. Sari, A. Alamsyah, and S. Wibowo,'Measuring e-Commerce service quality from online customer review using sentiment analysis', *J. Phys. Conf. Ser.,* vol. 971, no. 1. 2019

100. A. K and V. J,'Aspect Based Sentiment Analysis for E-Commerce Using Classification Techniques', *SSRN Electron. J.*, vol. 1, no. 1, pp. 35–42. 2019

101. C. Huang and G. Liu,'Sentiment analysis of network comments based on GCNN', *ACM Int. Conf. Proceeding Ser.*, vol. 7, no. 6, pp. 409–413. 2018

102. Q. Lu, Z. Zhu, D. Zhang, W. Wu, and Q. Guo,'Interactive Rule Attention Network for Aspect-Level Sentiment Analysis', *IEEE Access*, vol. 8, pp. 52505–52516. 2020

103. F. Xu, Z. Pan, and R. Xia,'E-commerce product review sentiment classification based on a naïve Bayes continuous learning framework', *Inf. Process. Manag.*, vol. 57, no. 5, p. 102221. 2020

104. Y. Han, M. Liu, and W. Jing,'Aspect-Level Drug Reviews Sentiment Analysis Based on Double BiGRU and Knowledge Transfer', *IEEE Access,* vol. 8, pp. 21314–21325, 2020.

105. B. Krawczyk and F. Herrera,'Neurocomputing A survey on data preprocessing for data stream mining : Current status and future directions', vol. 239, pp. 39–57.

106. E. Acuna,'Probit analysis', no. January 2011. 2017, 2019

107. S. Alasadi,'Review of Data Preprocessing Techniques in Data Mining', no. September. 2017

108. L. A. Mullen, K. Benoit, O. Keyes, D. Selivanov, and J. Arnold,'Fast , Consistent Tokenization of Natural Language Text', vol. 2017, pp. 2017–2019. 2018

109. T. Astuti, R. Wahyudi, Z. Rifai, and R. A. Pambudi,'An Experimental Study of Text Preprocessing Techniques for Automatic Short Answer Grading in

Indonesian', *2018 3rd Int. Conf. Inf. Technol. Inf. Syst. Electr. Eng.*, pp. 230–234. 2018.

110. S. Xu, Y. Qian, and R. Q. Hu,'A Data-driven Preprocessing Scheme on Anomaly Detection in Big Data Applications', pp. 814–819. 2017

111. S. Yang and H. Zhang,'Text Mining of Twitter Data Using a Latent Dirichlet Allocation Topic Model and Sentiment Analysis', no. August. 2019

112. S. Al-saqqa, A. Awajan, and S. Ghoul,'Stemming Effects on Sentiment Analysis using Large Arabic Multi-Domain Resources', *2019 Sixth Int. Conf. Soc. Networks Anal. Manag. Secur.*, pp. 211–216. 2019

113. B. Yu and I. Science,'An evaluation of text classification methods for literary study', vol. 23, no. 3, pp. 327–343. 2008

114. A. Companies,'Tweets Classification on the Base of Sentiments for US Airline Companies', pp. 1–22. 2019

115. W. Zhang, T. Yoshida, and X. Tang,'Expert Systems with Applications A comparative study of TF Ã IDF , LSI and multi-words for text classification', *Expert Syst. Appl.*, vol. 38, no. 3, pp. 2758–2765, 2011

116. S. Robertson,'Understanding inverse document frequency : on theoretical arguments for IDF', vol. 60, no. 5, pp. 503–520. 2004

Approval for Examination

Scholar's Name: Furqan Durrani Registration No: 60609

Program of Study: Master of Science in Computer Science

Thesis Title: Aspect Level Sentiment Analysis of E-Commerce: A Case Study of eBay and Amazon

It is to certify that the above student's thesis has been completed to my satisfaction and, to my belief, its standard is appropriate for submission for evaluation. I have also conducted a plagiarism test of this thesis using HEC prescribed software and found a similarity index at 10 %that is within the permissible limit set by the HEC for the MS/MPhil degree thesis. I have also found the thesis in a format recognized by the BU for the MS/MPhil thesis.

Principal Supervisor's Signature: ______________________

Date: 12-3-2021 Name: Muhammad Aasim Qureshi

Author's Declaration

I, Furqan Durrani _ hereby state that my MS thesis titled "Aspect Level Sentiment Analysis of E-Commerce: A Case Study of eBay and Amazon" is my own work and has not been submitted previously by me for taking any degree from this university Bahria University Lahore or anywhere else in the country/world.

At any time if my statement is found to be incorrect even after my graduation, the University has the right to withdraw/cancel my MS degree.

Name of Scholar: Furqan Durrani

Date: March 12, 2021

Plagiarism Undertaking

I, solemnly declare that research work presented in the thesis title "Aspect Level Sentiment Analysis of E-Commerce: A Case Study of eBay and Amazon" is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been duly acknowledged and that complete thesis has been written by me.

I understand the zero-tolerance policy of the HEC and Bahria University towards plagiarism. Therefore, I as an author of the above-titled thesis declare that no portion of my thesis has been plagiarized and any material used as a reference is properly referred/ cited.

I undertake that if I am found guilty of any formal plagiarism in the above-titled thesis even after awarding of MS degree, the university reserves the right to withdraw/revoke my MS degree and that HEC and the University have the right to publish my name on the HEC / University website on which names of students are placed who submitted plagiarized thesis.

Scholar / Author's Sign: ____________________

Name of the Scholar: Furqan Durrani

I would like to dedicate my thesis to ALLAH Almighty, who constantly supported me and motivates me. My parents who teach me to have faith in ALLAH and to remain determined and confident.

ACKNOWLEDGMENT

I would like to thank ALLAH Almighty, who gave me the strength and courage to accomplish this research and make me able to contribute my little efforts in the field of data science.

This thesis is the first-fruit of the training and efforts, which I have been receiving form my respected professor, and as a gratitude, I primary dedicate my thesis to my professor Dr. M. Aasim Qureshi. Without his constant guidance, it would not be possible for me to achieve my goal. I would like to thank all my teachers, who contributed to my academic career.

I am grateful to my father Aurangzaib Khan and mother Sanjida Baigum who have provided me utmost support in all aspects of my life. I am also grateful to my siblings and friends who have played a key role in the accomplishment of this achievement.

ABSTRACT

E-Commerce is the trading of services and goods through the internet. E-commerce has a very great influence on the world's economy. With the advancement of technology, consumers prefer to buy their desired products from online stores and e-commerce websites nowadays. Consumers also share their opinion about the product. In this case, a large amount of textual data are also generated in the form of feedbacks, suggestions, comments, and tweets. These reviews data help the organizations to understand customer expectations, provide a better shopping experience, and to increase sales. Sentiment Analysis can be used to identify positive, negative, and neutral sentiments from the customer reviews. Sentiment analysis a NLP technique that is used to analyze the data whether it is negative, neutral, or positive. Existing machine learning models provide a useful account of how to judge sentiment polarity. However, the accuracy of aspect-related information for the target terms is still required to be done. Hence, this study proposed the model which contains the combinations of two machine learning models Logistic Regression and Random Forest under the architecture of voting classifiers. For this study, data is collected from e-commerce websites eBay and Amazon. After pre-processing, subsets of data have been extracted with respect to the aspects of price, color, size, weight, and service from data, and the proposed model and different machine learning models (Naive Bayes, KNN, SVM, Random Forest, and RNN) are applied. It is observed that accuracy has been improved by using the proposed methodology which is 97%.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CNN	-	Convolutional Neural Network
LR	-	Logistic regression
SVM	-	Support Vector Machine
RF	-	Random Forest
VC	-	Voting Classifier
SA	-	Sentiment Analysis
DLSA	-	Document Level Sentiment Analysis
SLSA	-	Sentence Level Sentiment Analysis
ALSA	-	Aspect Level Sentiment Analysis
IEEE	-	Institute of Electrical and Electronics Engineers
ANN	-	Artificial Neural network
RNN	-	Recurrent Neural Network
ML	-	Machine Learning
SMOTE	-	**Synthetic Minority Oversampling Technique**

LIST OF APPENDICES

ASPECT LEVEL CENTIMENT ANALYSIS OF ECOMRESE: A CASE STUDY OF E-BAY AND AMAZON

FURQAN DURANI
03-243182-026

A thesis submitted in the fulfilment of the
requirements for the award of the degree of
Master of Science (Computer Science)

S

Department of Computer Sciences

BAHRIA UNIVERSITY LAHORE

MARCH 2021