Chap

Vvv

Vvv

Vvvv

CHAPTER 3

METHEDOLOGY

This study is carried out for the classification of sentiment with respect to the aspect of mobile with the help of different machine learning techniques. This study has been completed in five different phases (Literature Review and Problem Statement, Data Collection, Data Pre-processing, Classification, and Result and Discussion). The first phase- Literature Review and Problem Statement is further divided into two subsections, Literature review, and investigation of the research gaps. In the second phase- Data Collection of this study, the dataset is collected to conduct the research. In third phase- Data Pre-processing, data pre-processing is performed, and feature-related data is extracted. In the fourth phase- Classification, different machine learning models are applied on the data for the classification of sentiments. The final phase-Result and Discussion contains a detailed discussion of obtained results, evaluation, and conclusion. The research methodology is shown below in figure 3.1.

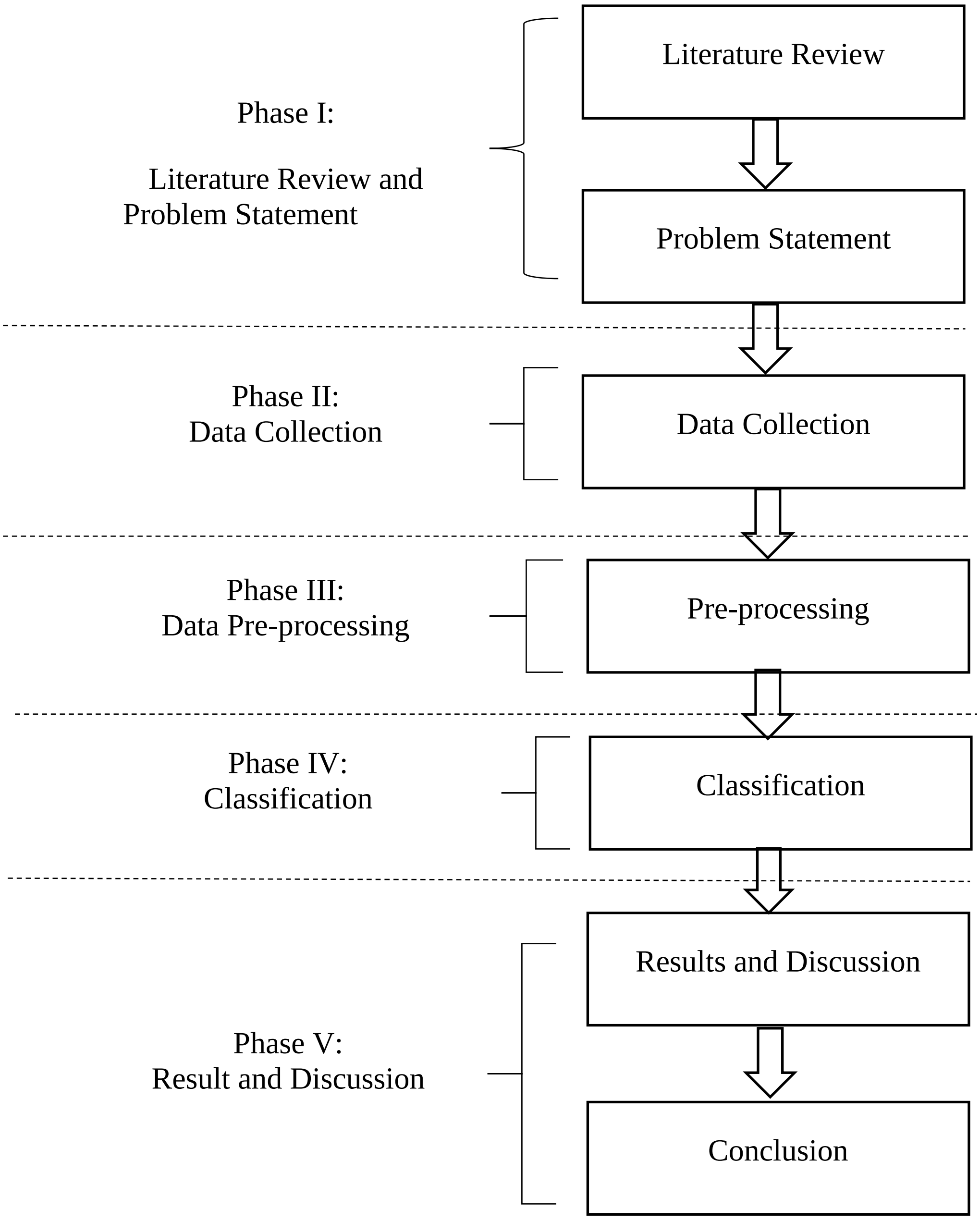

Figure 3.1: Proposed Methodology

3.1 Phase I: Literature Review and Problem identification

The first phase of the proposed methodology contains two phases. In the first sub phase, detailed literature review was conducted on existing researches and which helps to identify the research gap. In the second sub phase, the problem statement is established on the basis of literature review.

3.2 Phase II: Dataset Collection

In this study, a hybrid model technique has been proposed to improve the accuracy of aspect level sentiment analysis of e-commerce. This analysis has been performed on the user's experience which they have shared in form of reviews on ecommerce website. These reviews datasets have been collected from the following sources:

3.2.1 Dataset_1

The review dataset_1 has been downloaded from dataworld [117]. From the above data set, 4304 reviews have been extracted in which positive comments are 1455, negative reviews are 1455 and neutral reviews are 1394. The sample dataset_1 is as the mentioned in table 3.1:

Table 3.1: Sample Dataset_1

Product Name	Brand Name	Price	Rating	Reviews	Voting
Samsung Galaxy S Duos GT-S7562 GSM Unlocked	Samsung	95.88	5	Excellent, it was what I expected.	0

3.2.2 Dataset_2

The review dataset_1 has been downloaded from Kaggle [112]:

From the above data set, 13595 reviews has been extracted in which positive comments are 4916, negative reviews are 4296 and neutral reviews are 4383. The sample dataset_1 is as the mentioned in table 3.2:

Table 3.2: Sample dataset_2

category	Review Title	Review Content	Rating
Samsung Galaxy S Duos GT-S7562 GSM Unlocked	Samsung	This gaming headset ticks all the boxes # looks grate # built to last # excellent sound & mic # com...	5

3.2.3 Dataset_3

This study performs experiments on user reviews using a supervised machine learning approach. For this, eBay reviews were extracted using the Crawler. The sample of dataset is given in table 3.3. The dataset contains the product name, review date, review author, review title, reviews text, helpful vote, unhelpful vote, and review rating as all attribute descriptions are shown in Table. In this study, Review Text attribute is used for the experiments and then find the sentiment using the lexicon technique from text reviews.

Table 3.3: Sample Dataset

Product Name	**Review Date**	**Review Author**	**Review Title**	**Review Text**	**Helpful Vote**	**Unhelpful Vote**	**view Rating**
Apple iPhone 7 128GB Rose Gold (Unlocked) A1778 (GSM)	4-Sep 18	2006yushan	Volume control can be better	One thing that can be better is the …	0	2	5

The reviews data from the Amazon was also collected from the site on electronic devices. To increase the size of dataset, the dataset of eBay and Amazon is merged for the analysis. The Amazon dataset also contains the text reviews of the users on different amazon products from the electronics category. We extract the 252231 reviews from Amazon and 33324 reviews from eBay. The details of the dataset attribute is mentioned the table 3.4.

Table 3.4: Dataset attribute description

Attribute	**Description**
Product Name	This attribute contains the product name.
Review Date	This attribute contains the date for the review when the user posts it.
Review Author	This attribute contains the author name who posted the review.
Review Title	This attribute contains the short title for reviews.
Review Text	This attribute contains the text for reviews.
Helpful Vote	This attribute contains the vote for the product which is helpful for the users.
Unhelpful Vote	This attribute contains the vote for the product.
Review Rating	This attribute contains the rating for the product post by the reviewer.

3.3 Phase III: Data Pre-processing

Pre-processing of the obtained data is one of the significant tasks that need to be performed. Through data pre-processing, we generally manage to transform the unstructured data into an organized and structural format [105]. The basic purpose of pre-processing is to enhance the quality of the input data by reducing its quantity so that the machine can understand the patterns from data that further helps to extract more useful and relevant features from the pre-processed data. It aids the machine to learn more accurate patterns from data which also improves the performance of the machine learning classifier in terms of accuracy. Input data should be delivered in the required format, amount, and structure that is appropriate to the required task. Unfavorably, real-world data is vastly inclined by the inappropriate factors, the performance of the analysis being performed depends on the quality of data ultimately the low-quality data will provide low quality performance[106]. Several existing data pre-processing techniques are being used to attain structured data from unorganized data. These techniques are used to remove the least useful and unnecessary data that devours machine process time and power.

As it helps to attain only the data that is useful and important to make further analysis, pre-processed data acts a vital role in the decision-making of the machine learning model [107]. Usually, initially obtained text data contains the combination of lower-case and upper-case letters, numbers, stop words, punctuations, and various forms of words that have no importance in the classification and rather take a lot of processing time which further leads to misclassification of data. Such types of data have no meaning to the text and have no role in the decision-making process of the machine. As it is conclusive that the initial format of obtained data is inappropriate to the classifier, so that removing such patterns from data will not cause any harm and no important information will be lost, it will become more valuable for classification tasks instead.

It is the initial processing of data to prepare it for further major processing or analysis. There are several steps required to prepare data for further processing. Some of the core applied steps are discussed below:

3.3.1 Tokenization

This is the main part of pre-processing technique in which the raw data is converted into small tokens by splitting the long strings of text. In tokenization of the textual data, a sentence is transformed into an set of words or terms [108]. Example of tokenization is discussed in the following table 3.5.

Table 3.5: Data before and after tokenization

Before Tokenization	After Tokenization
I would like to introduce Mr. David as new Sales Manager, he’ll start his job on Oct 01, 2021.	‘I’, ‘would’, ‘like’, ‘to’, ‘introduce’, ‘Mr.’, ‘David’, ‘as’, ‘new’, ‘Sales’, ‘Manager’, ‘,’, ‘he’ll’, ‘start’, ‘his’, ‘job’, ‘from’, ‘Oct’, ‘01’ ‘,’, ‘2021’.

3.3.2 Punctuation Removal

Punctuation removal is another step of data pre-processing that aims to remove the punctuations including “!’;’.&? -_,” from the data. Punctuations are removed from data because they do not have any impact on the data as they are meaningless to the machine. It also reduces the capability of a machine to discriminate between other characters and punctuation [109].

Table 3.6: Data before and after punctuation removal

Before Punctuation Removal	After Punctuation Removal
I , would , like , to , introduce , Mr. , David , as , new , Sales , Manager , , , he’ll , start , his , job , from , Oct , 01 , , 2021 .	I , would , like , to , introduce , Mr , David as , new , Sales , Manager , he, ll , start , his , job , from , Oct , 01 , 2021

3.3.3 Numeric Removal

In this step, numeric values are removed from the data to improve its quality. Since in the text data where the numeric values like digits are not of any use in the decision-making process, which rather trouble the machine in the feature extraction process.

Usually, the values containing numbers do not contribute to the classification of data [110]. When working with reviews or textual data that is not concerned with the digits, then it is necessary to pre-process data to remove the numeric values. The same applies to the null values since the null values do not add to the performance of the model.

Table 3.7: Data before and after numeric removal

Before Numeric Removal	After Numeric Removal
I , would , like , to , introduce , Mr , David as , new , Sales , Manager , hell , start , his , job , from , Oct , 01 , 2021	I , would , like , to , introduce , Mr , David , as , new , Sales , Manager , hell , start , his , job , from , Oct

3.3.4 Lowercase Conversion

In another step, all the letters are converted to lowercase. Machine learning models are case sensitive so that this conversion has major importance[111]. For example, if the conversion is not applied to data the model will count the existence of “Sales” and “sales” as two different words. Sometimes in an informal or formal record, a blend of upper-case and lower-case letters is used to give significant attention to the specific words.

Table 3.8: Data before and after converting to lower case

Before Lowercase Conversion	After Lowercase Conversion
I , would ,like , to , introduce , Mr , David , as , new , Sales , Manager , hell , start , his , job , from , Oct	i , would , like , to , introduce , mr , david , as , new , sales , manager , hell , start , his , job , from , oct

3.3.5 Stemming

Stemming is performed on the input data to convert all the words or terms used in the data into their first form. In other words, it is a preprocessing technique that has been used to transform words into their root form to improve machine learning model performance [112]. For instance, the word “records”, “recording”, “recorded” are different forms of the same word that might confuse the classification model. Therefore, after stemming techniques these forms of words will be converted to their root form “record”.

Table 3.9: Data before and after stemming

Before Stemming	After Stemming
i , would , like , to , introduce , mr , david , as , new , sales , manager , hell , start , his , job , from , Oct	i , will , like , to , introduce , mr , david , as , new , sale , manager , hell , start , his , job , from , oct

3.3.6 Stop words Removal

One of the major tasks in pre-processing is to remove the data that has no use in the classification. Stop words are the words that are worthless for the model to make the decision. In this step of pre-processing the stop words are removed from the dataset. It is the most vital task in pre-processing that removes the useless data for further processing of the data. Stop words are the words used to form a sentence that has no use in text classification and are meaningless to the machine learning models [9]. Stop words include words like is, am, i, the, to, are, that, they, etc.

Table 3.10: Data before and after removing stop words

Before Stop words Removal	After Stop words Removal
i , will , like , to , introduce , mr , david , as , new , sale , manager , he,ll , start , his , job , from , Oct	like , introduce , david , new , sale , manager , he,ll , start , job , oct

3.3.7 Words Removal

There are many meaningless words generated during pre-processing which do not have any specific meaning and can spoil the results. These words are not even the part of sentiment dictionary. In this way, the words which have length of equal or less than length 2 is removed.

By following all the above pre-processing steps, review text has been changed in the following format mentioned in the table 3.11.

Table 3.11: Data before and after pre-processing

Before Preprocessing	After Preprocessing
I would like to introduce Mr. David as new Sales Manager, he’ll start his job on Oct 01, 2021.	like introduce david new sale manager he start job oct

3.3.8 Data Balancing

An imbalanced class distribution will have one or more classes with few examples (the minority classes) and one or more classes with many examples (the majority classes). It is best understood in the context of a binary (two-class) classification problem where class 0 is the majority class and class 1 is the minority class. There are two techniques to handle the imbalanced data.

3.3.8.1 Under Sampling

Undersampling techniques remove examples from the training dataset that belong to the majority class in order to better balance the class distribution, such as reducing the skew from a 1:100 to a 1:10, 1:2, or even a 1:1 class distribution. The simplest undersampling technique involves randomly selecting examples from the majority class and deleting them from the training dataset. This is referred to as random undersampling. Although simple and effective, a limitation of this technique is that examples are removed without any concern for how useful or important they might be in determining the decision boundary between the classes.

3.3.8.2 Over Sampling

One way to solve this problem is to oversample the examples in the minority class. This can be achieved by simply duplicating examples from the minority class in the training dataset prior to fitting a model. This can balance the class distribution but does not provide any additional information to the model.

An improvement on duplicating examples from the minority class is to synthesize new examples from the minority class. This is a type of data augmentation for tabular data and can be very effective.

Perhaps the most widely used approach to synthesizing new examples is called the **Synthetic Minority Oversampling TEchnique**, or SMOTE for short. This technique was described by Nitesh Chawla. in their 2002 paper named for the technique

SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line.

Specifically, a random example from the minority class is first chosen. Then *k* of the nearest neighbors for that example are found (typically *k=5*). A randomly selected neighbor is chosen and a synthetic example is created at a randomly selected point between the two examples in feature space.

This procedure can be used to create as many synthetic examples for the minority class as are required. As described in the paper, it suggests first using random undersampling to trim the number of examples in the majority class, then use SMOTE to oversample the minority class to balance the class distribution.

3.4 Aspect Based Data Extraction

In [68], researchers explained the technique of aspect based data extraction from the data corpus. POS tagging is an important technique to find sentiment word and aspect from the textual data. We also used the same technique to extract the feature-based data from the large dataset. In POS tagging, the nouns phrase and noun represent the aspect term in the textual sentence. Parts of speech tags are presented in the following table:

Table 3.12: Parts of speech tags.

Description	Parts of Speech Tags	Indication
Noun	NN	Aspect
Adjective	JJ	Sentiment
VB	Verb	Sentiment
RB	Adverb	Sentiment

In the experiment approach first, we collect the data using a crawler from the eBay site. In data collection, we collect data contain the reviews of eBay users. We also extract the reviews from Amazon related to the same electronics products and merge them with the eBay data to increase the size of the data. After that, we separate the reviews based on aspects such as Color, Size, Weight, Service, and Price with the help of POS tagging.

Table 3.13: Aspects based reviews count

Aspect	**Count**
Size	6969
Price	32139
Service	6901
Weight	2068
Color	3074

For the classification purpose, we extract the target label for reviews as positive, negative, and neutral. For that, we used text blob library. Before passing the data to text blob we have done pre-processing of text reviews to clean text reviews. Pre-processing removes all raw data such as punctuation, numbers, stop words, and then we pass this clean data to the text blob to extract the sentiment from the data as shown in Table 11. The dataset sample after finding the sentiment is shown in Table 13 and the ratio of sentiment is shown in Table 3.14.

Table 3.14: Sample of data after labeling

ID	Sentences	Polarity Score	Sentiment
1	One thing definitely better volume control.	0.26	Positive
2	Apple within week phone said hacked phone process	0	Neutral
3	started broken microphone problems reach frequent hangs	-0.45	Negative

Table 3.15: Sentiment count in the dataset

Aspect	Count	Positive	Negative	Neutral
Size	6969	5262	1358	345
Price	32139	24105	6592	1417
Service	6901	5023	1515	360
Weight	2068	1524	467	75
Color	3074	2239	680	153

The feature extraction technique was used to extract the feature after extracting the sentiment and for that, we used the TF-IDF features extract technique which gives the weighted features which are more suitable for the learning of the model (see Section Feature Extraction). The used dataset is imbalanced because the ratio of the data for each sentiment is not equal which can be cause for the overfitting of the model for majority class data. To solve this problem, SMOTE technique is used to make the dataset balanced. The data count after balancing the dataset is shown in Table 3.16. SMOTE technique generates mock data to create a balance between the target class ratios. An argument is passed to the SMOTE algorithm to set a threshold value for mock data to balance minority and majority classes. In this technique, SMOTE chooses comparative records and alters those records one column at a time by a random value within the difference to the adjacent records. We get a 1:1 ratio of each class negative, positive and negative examples by using SMOTE techniques in this experiment as shown 3.16.

Table 3.16: Sentiment count in the dataset after oversampling

Aspect	Total Count	Positive	Negative	Neutral
Size	15786	5262	5262	5262
Price	72315	24105	24105	24105
Service	15069	5023	5023	5023
Weight	4572	1524	1524	1524
Color	6717	2239	2239	2239

After that, we split the dataset into training and testing sets with the ratio of 80 and 20. The 80 percent of data used for the training of machine learning models and 20 percent of data used for the testing of models. Random forest, logistic regression, support vector machine, k nearest neighbour, and decision tree are gets fitted on the training set and after that evaluation is done using the test data. For the evaluation, accuracy, precision, recall, and F1 score are used and on an imbalanced dataset F1 score can be preferred.

3.5 Features Extraction

This study used TF-IDF for feature extraction. TF-IDF is an abbreviation of Term Frequency (TF) and Inverse Document Frequency (IDF). TF-IDF is a counting measure technique that is typically reproduced in information retrieval (IR) and clarification. It is assumed that TF-IDF will show how a term is represented in the analysis. TF and IDF are used in feature extraction techniques [113]. This is a very common algorithm for converting text into a meaningful number representation that is used to suit the prediction machine algorithm. The words are calculated to be more important with higher frequency ratings. TF-IDF is different from the BoW method because the BoW is the basic word count in a script, but the TF-IDF finds weighted text data features such that machine learning models can train themselves to increase their accuracy on important aspects [114]. The frequency of words shows us how often a word is used in a script. The huge volume document may contain many term frequency for a word so there is a better possibility that a term will be available for more time in the large documents [115].

Term frequency can be calculated as:

$$TF(t) = \frac{\mathrm{N}}{\mathrm{T}} \qquad \text{Eq. 3.1}$$

Here the N represents the number of times a term occurs in a document whereas the T represents the total terms in a document.

While the frequency of the inverse document (IDF) shows us how continually a word appears in a corpus document. If a more frequent word in a document having a low score of inverse document frequency. Then the stop words in the dataset also have low IDF that indicates the feature's low value [116].

Inverse document frequency can be calculated as

$$\mathrm{IDF(t) = log\ (N / (1 + D_t))} \qquad \text{Eq. 3.2}$$

Here D_T is the number of documents where term t appears when the term frequency function satisfies $TF \neq 0$ then 1 will be added into the formula to avoid zero-division.

So, the complete TF-IDF can be defined as:

$$\text{Tf} - \text{IDF} = \text{if}(t) * \text{IDF}(t) \qquad \text{Eq. 3.3}$$

3.6 Phase IV: Classification

In phase IV, different machine learning models such as Support Vector Machine, Naïve Bayes, K-Nearest Neighbour, Gausian Naïve Bayes, Random Forest, Logistc Regression, Recurrent Neural Network and Voting Classifier were implemented on customer reviews for aspect level sentiment analysis. Details of phase IV is discussed in chapter 4.

3.6.1 Voting Classifier

After the performance evaluation of the different well known and commonly used classifiers like Support Vector Machine, Naïve Bayes, K-Nearest Neighbour, Gausian Naïve Bayes, Random Forest, Logistic Regression, Recurrent Neural Network, the top three best classifiers have been identified based on the performance. Then these top three classifiers were utilized for the next step to ensemble. Ensemble Voting Classifier: For the Ensemble Classifier, this article considered the approach of Voting Classifier. Selected top three classifiers were utilized for these Voting Classification to get the best performance and output. Results: In the last step, the performance of the Voting Classifier will be assessed based on execution measurements likewise test score, ROC score, precision score, recall value and F1. The results will then be compared with other relevant works for evaluating the results.

3.6.1.1 Voting Classifier Algorithm

The Ensemble Voting Classifier [16] is a meta classifier for consolidating comparative or adroitly extraordinary machine learning classifiers for classification and detection. The Ensemble Voting Classifier executes “hard” and “soft” voting.

Hard Voting: Hard ensemble voting is the easiest case of majority voting. Here, the class label Y is determined through majority voting of each classifier Cj :

$$Y = \text{mode}\ \{C1(x), C2(x), \ldots , Cm(x)\} \qquad [j = 1, 2, 3, \ldots, m]$$

Soft Voting: In soft ensemble voting, the class names are anticipated depending on the anticipated probabilities Pij of each instance ‘i’ classifier. This methodology is usually prescribed if the classifiers are very much aligned.

$$Y = \text{arg maxi} \sum m\ j{=}1\ \ WjPij \qquad [j = 1, 2, 3, \ldots, m;\ i = 1, 2, 3, \ldots, n]$$

where Wj is the load that can be distributed with the jth classifier.

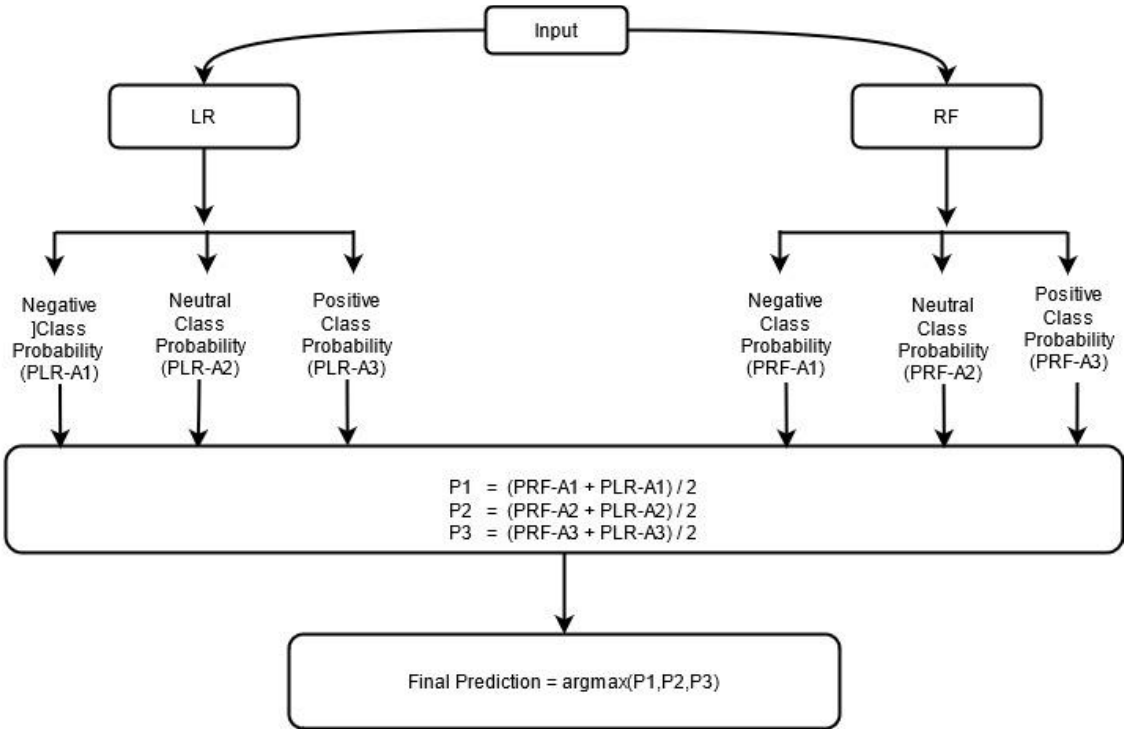
Input
LR
RF
Negative]Class Probability (PLR-A1)
Neutral Class Probability (PLR-A2)
Positive Class Probability (PLR-A3)
Negative Class Probability (PRF-A1)
Neutral Class Probability (PRF-A2)
Positive Class Probability (PRF-A3)
P1 = (PRF-A1 + PLR-A1) / 2
P2 = (PRF-A2 + PLR-A2) / 2
P3 = (PRF-A3 + PLR-A3) / 2
Final Prediction = argmax(P1,P2,P3)

Figure 3.2: Voting Classifier Architecture

3.7 Phase V: Results and Discussion

In phase v, different machine learning algorithms were applied on aspect-based data sets and results were generated and evaluated in term of accuracy, precision, recall and F! score. Then study is concluded with the progress that will be made in future. Details can be found in Chapter 4.