**Faculty Profiling Based On Students Feedbacks Using Sentiment Analysis**

A Dissertation Report Submitted to

Devi Ahilya Vishwavidhyalaya, Indore

towards partial fulfillment for the award of

**Master of Engineering (Computer Engineering) With**

**Specialization in Software Engineering**

**2018-2019**

**Supervised by: Submitted by:**

**Dr. Vaibhav Jain Neeraj Sharma**

**Department of Computer Engineering**

**Institute of Engineering &Technology**

**Devi Ahilya Vishwavidhyalaya, Indore**

**(**[**www.iet.dauniv.ac.in**](http://www.iet.dauniv.ac.in)**)**

**July 2019**

**Dissertation Approval Sheet**

The dissertation entitled **“Faculty Profiling Based On Student Feedbacks Using Sentiment Analysis”** submitted by **Neeraj Sharma** is approved as partial fulfillment for the award of the **Master of Engineering** **(Computer Engineering) Specialization in Software Engineering** degree by **Devi Ahilya Vishwavidhyalaya, Indore.**

**Internal Examiner External Examiner**

**Date:**

**Director**

**Institute of Engineering & Technology,**

**Devi Ahilya Vishwavidhyalaya, Indore**

**Recommendation**

The dissertation entitled “**Faculty Profiling Based On Student Feedbacks Using Sentiment Analysis**” Submitted by **Neeraj Sharma** is a satisfactory account of the bonafide work done under my supervision is recommended towards partial fulfillment for the award of the **Master of Engineering (Computer Engineering) Specialization in Software Engineering** degree by Devi Ahilya Vishwavidhyalaya, Indore.

**Dr. Vaibhav Jain**

Date:

Endorsed By

**Head**

**Department of Computer Engineering**

**Candidate Declaration**

I hereby declare that the work which is being presented in this project entitled **Faculty Profiling Based On Student Feedbacks Using Sentiment Analysis** in partial fulfillment of degree of **Master of Engineering** in **Computer Science Engineering(Specialization Software Engineering)**  is an authentic record of my own work carried out under the supervision and guidance of **Dr. Vaibhav Jain** in Department of **Computer Engineering**, Institute of Engineering and Technology, Devi Ahilya Vishwavidyalaya, Indore

I am fully responsible for the matter embodied in this project in case of any discrepancy found in the project and the project has not been submitted for the award of any other degree.

**Date:**

**Place: Indore**

<Signature of student >

**Acknowledgements**

I am highly indebted to my mentor Dr. Vaibhav Jain Sir (Associate Professor, Computer Engineering Department) for his guidance and constant supervision as well as for providing necessary information regarding the project & also for his support.

*Signature*

**Neeraj Sharma**

**Abstract**

Educational data Mining (EDM) deals with developing methods for exploring different varieties of data collected from the educational domain, and use those methods to understand students and the environment which they learn in. EDM facilitates the institution to discover useful patterns helping them to improve the decision making process. One way to attain quality in higher education system is by exploring the student feedback (comments) for improving the quality of instructors. We propose a system based on machine learning and sentiment analysis to help extract sentiments from the student feedback and generate instructor feedback accordingly.

**Table of Contents**

**Dissertation Approval sheet ……………………………………………………... I**

**Recommendation ……………………………………………………………..… II**

**Candidate declaration ……………………………………………………………III**

**Acknowledgements ……………………………………………………………. IV**

**Abstract ………………………………………………………………………….V**

**Table of Contents …………………………………………………………………VI**

1. **Introduction ………………………………………………………… Page no.**
   1. **Rationale**
   2. **Problem Definition and Proposed Solution**
   3. **System Synthesis**
   4. **Report Organization**
2. **Literature Survey ……………………………………………………Page no.**
3. **Preliminary Concepts ……………………………………………… Page no.**
   * 1. **Requirement Analysis**
     2. **Object Oriented Analysis**
     3. **Architectural Specification**
   1. **Design**
      1. **Use Case Diagram**
      2. **Use Case Specification**
      3. **Activity Diagram**
      4. **Sequence Diagram**
      5. **Subsystem Class Design ( Generic)**
      6. **Subsystem Class Design ( Implementation Specific )**
      7. **Data Model**

**4. Implementation and Testing**

* + 1. **Testing Strategy Adopted**
    2. **System Testing**
    3. **Component Testing**
    4. **Test Cases**

1. **Conclusion ………………………………………………… Page no.**
2. **Appendix A: Screen and Report Formats (Add any other relevant material)**
3. **References and Bibliography……………………………… Page no.**

**CHAPTER 1**

**INTRODUCTION**

**1.1 Introduction**

Educational Data Mining (EDM) is an application area of data mining that is developed to address problems in education. The Educational Data Mining community website, www.educationaldatamining.org, defines educational data mining as follows: “Educational Data Mining is an emerging discipline, concerned with developing methods for exploring the unique types of data that come from educational settings, and using those methods to better understand students, and the settings which they learn in.” Ryan S. Baker and Kalina Yacef [27] identified the following four goals of EDM: Predicting students' future learning behaviours, discovering or improving domain models, studying the effects of educational support and advancing scientific knowledge about learning and learners by building and incorporating student models, the field of EDM research and the technology and software used. These goals can be achieved by applying various data mining techniques like regression analysis, classification, clustering on the educational data. Achieving these goals can lead to helping students who need advice, removing and adding material to the unit according to student’s comprehension and students opinions about the course and faculty performance.

Castro et al. [29] suggests the following EDM subjects/tasks:

* Applications dealing with the assessment of the student’s learning performance
* Applications that provide course adaptation and learning recommendations based on the student’s learning behaviour,
* Approaches dealing with the evaluation of learning material and educational web-based courses,
* Applications that involve feedback to both teacher and students and its analysis,
* Developments for detection of atypical
* Students’ learning behaviours.

Raw Data

Interpretation

Pre-Processing

Educational

Environment

Model/Pattern

Relevant Data

EDM

Methods

Fig. 1.1 Educational data mining process.

Role of teachers have surprisingly changed over the years with the advent of Big Data. Teachers are now required not just to get accustomed to the usage of new tools but also keep pace with cutting edge technologies. Teachers are the skeleton of any educational system. Effectiveness of a teacher is not just measured by his/her academic qualiﬁcation but also by dedication, skill and commitment. The most effective way for a teacher to improve teaching methodologies is by taking timely feedback from students. Feedbacks can be open ended and/or close ended. Open ended textual feedbacks are difﬁcult to observe manually to draw conclusions as they are ﬁlled with observations and insights. This is the place where Educational Data Mining and Sentiment Analysis play an important role.

Education today has become one of the most important aspects that helps individuals build their social skills, enhances the problem solving and decision making skills. With the growing number of schools, colleges and universities around the globe, the use of technology in Education has increased to meet the need of providing up-to-date information. Education has taken altogether a new dimension. Quality is one of the imperative factors that advance the development of higher education, which has led to the emergence of data mining in education characterized as “Educational Data Mining”. Students’ academic performance is impacted by various factors which in turn affect the Quality of Education at various levels viz. schools, colleges and universities. This is where Educational Data Mining plays imperative function. As per the educational business construct outlook ‘Academic Analytics’, ‘Learning Analytics’ and ‘Educational Data-Mining’ are related and mapped together to meet the needs of Learners, Faculty, administrators, Funders, Education Authorities and various other stakeholders. With the role and importance of educational data mining being that clear the main issue is where and how to apply the educational data mining techniques to get the optimal outputs.

Sentiment Analysis can be used to extract useful information about the teaching methodology of a teacher and also towards the course curriculum. Sentiment Analysis identiﬁes students learning curve, understand students need, foresee their performances and make effective changes in the teaching style. The explosive data set of students must be mined efﬁciently with respect to only the required dataset. Results of Sentiment Analysis help the teachers and the organization to take corrective actions. Rewards and appreciation can presented to those teachers about whom student exhibit positive sentiments. Efﬁcient Sentiment Analysis of all the education data across different sources can help in making better informed policies thereby contributing holistically for the betterment of the education sector. Sentiment Analysis can be applied to the exponential volume of data for the target users such as student, teachers and the organization as a whole. Application of Sentiment Analysis techniques at the document, sentence or phase level enables efﬁcient business applications, helps to make better decisions and facilitates early prediction of trends.

Analytics Industry is all about obtaining the “Information” from the data. With the growing amount of data in recent years, that too mostly unstructured, it’s difﬁcult to obtain the relevant and desired information. But, technology has developed some powerful methods which can be used to mine through the data and fetch the information that we are looking for. One such technique in the ﬁeld of text mining is Text summarization. As the name suggests, it is a process to automatically summarizing a large body of text into a brief summary consisting of the important highlights of the context. Text summarization can be done in two ways; Extractive text summarization, and abstractive text summarization. Extractive summarization rely on extracting several parts, such as phrases and sentences, from a piece of text and stack them together to create a summary, i.e. attempt to summarize articles by selecting a subset of words that retain the most important points. On the other hand, abstractive summarization select words based on semantic understanding; even those words did not appear in the source documents. It aims at producing important material in a new way.

* 1. **Rationale**

Today a lot of the educational agencies like UGC and AICTE are focusing on students feedbacks. They require the affiliated colleges to upload the students’ feedback at the end of every semester on their website/portal for keeping a regular check on the faculty performance, as a measure to improve the services. Therefore, the motivation for sentiment analysis of feedback data is two-fold. Both students and teachers value “student’s opinion” about a faculty and how they teach. At the end of the course work when the students provide a feedback, it contains students’ opinions and suggestions for the course work and faculty. If the enormous feedback data is properly analyzed then this could help in improving course work, understanding student opinion about the faculty teaching methods, etc.

When taking up classes students tend to approach their senior students for the reviews about the course work and faculty. This review is incomplete since it is based on one or a bunch of students. A faculty profile that can be viewed can be of great help for the new coming students, comprising of the faculties profile as demonstrated on the college website. This is a tedious task if carried out using manual techniques. Therefore, to achieve both the goals firstly the feedback collections needs to be automated; followed by mining of the feedback for sentiments and opinions.

**1.2 Problem Definition**

Student feedback system is one of the important parts of educational field. It can help to establish strong association between student desires and teacher efforts. It can be good way to explore fine or minute scope of improvement. Feedback helps in knowing the view of user against any services or effort. Strong feedback systems can lead to improvement of the current education scenarios. For such reasons, feedbacks are taken from students at the end of every session. Once the feedback is taken, the major task is to analyze the data. This task can be tiresome and difficult to accomplish manually as the feedback data collected from the students is very large due to two reason, firstly, the total number of students in a college varies between few thousands and secondly, evaluating and combining of sentiment analysis as well as other techniques can be tough. Another way, manual feedback analysis can be affliction with inequitable decision or can’t be considered as transparent conclusion. Sometime it may face issue of fabrication or intentional exploitation. Subsequently, it cannot be considered as a strong medium on the basis of which changes are to be made in the current system.

The task of automation of an already existing manual feedback system is in itself a very tedious job. Here at IET-DAVV (Institute of Engineering and Technology, Indore) this has been already done with the efforts of the CS & IT departments of the college. The feedback provided by the students consists of nominal as well as free text. Feedback taken can be classified as objective or subjective. Nowadays, University or colleges explore hybrid pattern for feedback and club both technique into single process. Objective feedback system can give very fine result in terms of point/ marks with the representation of percentage. But it cannot be preferred to know the particular wish or desire of the student. Student may want to express some thought out from the feedback pattern. So, this feedback process always suffers with analysis approach and is not feasible to cover the wide scope of feedback. Subjective feedback can give open scope to express any view that can be complex to analyze and draw fine observation. Therefore, the combination of both and manually analyzing both can be challenging.

* 1. **Proposed Solution**

This work is carried out with the solution to prepare an automatic system to generate a faculty profile from the student feedback data collected using the online feedback system at IET-DAVV, Indore. This work observes the problem to view or analyze remark section of student feedback. (The remark section of student feedback consists of subjective feedback). Sentiment analysis can be an excellent source of information and can help to explore the thinking and desire of student. It can help in following prospective:

1. Determine teaching strategy
2. Improve understanding of student attitude.
3. Bridging the gap between student and teachers.
4. Generate teacher evaluation.

In this process we have built a prototype system takes in as input the student feedbacks coming in from the Student Feedback System at IET-DAVV, Indore. The feedbacks are than classified. The faculty profile is generated from the highly polarized feedback among the classified feedback.

Previous research works have shown that there are basically three techniques for sentiment analysis using classification, i.e., lexicon based, machine learning based and hybrid approach (Combination of the former two approaches). Therefore, we propose to carry out sentiment analysis using two classification techniques, .i.e., lexicon based and machine learning based. Though supervised machine learning techniques for sentiment analysis give most superior results for educational data, we have applied Sentiment analysis using various machine learning techniques along with the lexicon based technique on our feedback system. The machine learning techniques used for sentiment classification of student feedback are all supervised learning approaches as unsupervised approaches aren’t feasible in our scenario. We proposed to use two different classifiers, i.e., Naïve Bayes (multinomial), SVM (linear, radial basis function), among the lexicon based (Rule based approaches) we have used two approaches one is using the SentiWordNet dictionary and the other one is using the VADER (Valence Aware Dictionary and sEntiment Reasoner ) , for the purpose of evaluating which classifier works best with the educational feedback data. Once evaluated, the best among the four classifiers is further preferred to classify sentiment for the feedback data. Using the results of classification we generate a faculty profile.

* 1. **Report Organization**

This thesis work elaborates the brief idea, need and techniques for sentiment analysis when applied on the free text. The general outline of the thesis and the main contribution presented in different chapters are summarized below:

* **CHAPTER 2**: Literature review- consists of a brief history of the relevant research papers and survey papers related to the work.
* **CHAPTER 3**: Preliminary Concepts- consists of all the preliminary knowledge required for understanding of the techniques and modules used in the successful completion of the work.
* **CHAPTER 4**: Methodology- consists of the system architecture and the step by step processing at every level in the work.
* **CHAPTER 5**: Implementation and Results- consists of the elaborated coding implementation of the steps and the intermediate and end results produced.
* **CHAPTER 6**: Conclusions and Future Work- analysis of the results with the implementation limitations and the future scope of the work done.

**CHAPTER 2**

**LITERATURE REVIEW**

In this chapter we have surveyed the related work on sentiment analysis and text summarization techniques performed on educational data.

**Sentiment Analysis and Text summarization on Educational Data**

In the work by D.S. Chaplot et al. [9] have provided an algorithm that predicts and pinpoints the time when the student is going to drop out. They have used click stream logs and forum posts data from the Coursera MOOC. Their work allows the course instructor team to take necessary steps to prevent or reduce student attrition during the course. However, the amount of student feedbacks taken into account for carrying out the experiment is limited to only one course and hence the sample space is limited.

Miaomiao Wen et al.[11] have studied the correlation between the student sentiment ratio measured based on the daily forum post and the number of students droping out each day. They have used the MOOCs post course surveys to collect student opinions. Sentiment analysis is carried out in two phases in the post retrieval phase and opinion estimation phase in the post retrieval phase only the messages containing the topic keywords are used. From the results a keyword list for the topics is constructed. Opinion Estimation for each set of posts that are related to a topic, i.e. topic sentiment ratio on day t is the ratio of positive versus negative words used in that day’s post set. Students’ opinion about the course is measured by the number of students that drop out of the course each day. This work reﬂects on how sentiment analysis can be useful in a MOOC context on the course-level, and uses collective sentiment analysis, to explore the relation between opinions expressed by students and the students’ dropout rate. However, the positive and negative terms used in the sentiment lexicon are from [30], a word list containing about 2,000 and 4,800 words marked as positive and negative, respectively. We can say that the lexicon dictionary being used is of limited size.

In another work, Meishan Hu et al.[10] proposed two approaches to incorporate comments into document summarization. The ﬁrst approach scores document sentences based on keywords derived from comments, while the second approach scores document sentences and comments altogether. The latter approach showed significant improvement over the techniques using only the keywords. However, their work does not address the similarity between two sentences in a given piece of text which can prove to be an important aspect for text summarization.

Sentiment analysis can be done using three different techniques; lexicon based, machine learning based and hybrid which uses a combination of both lexicon based approach and machine learning based approach.

**Lexicon Based Sentiment Analysis On public reviews**

Regarding the evaluation of sentiment analysis for student feedback, Kim and Calvo [13], apply category based and dimension-based emotion prediction models. They used WordNet-affect as a linguistic lexical resource and two dimensionality reduction techniques are evaluated. Both models are inferred from textual and quantitative students’ responses to Unit of Study Evaluations questionnaires with the aim at providing a comprehensive understanding of the student experience.

Maite Toboado et.al.[3] presented a lexicon-based approach to extracting sentiment from text. The Semantic Orientation CALculator(SO-CAL) which uses dictionaries of words annotated with their semantic orientation (polarity and strength), and incorporates intensiﬁcation and negation. SO-CAL is applied to the polarity classiﬁcation task, the process of assigning a positive or negative label to a text that captures the text’s opinion towards its main subject matter. Efstratios Kentopoulos et.al.[4] Proposed the deployment of original ontology-based techniques towards a more efficient sentiment analysis of twitter posts. The posts are not simply characterized by a sentiment score as is the case with machine learning based classifiers, but instead receive a sentiment grade for each distinct notion in the post.

C.J. Hutto and Eric Gilbert[1] presented VADER, a simple rule-based model for general sentiment analysis, and compare its effectiveness to eleven typical state-of-practice benchmarks including LIWC, ANEW, the General Inquirer, SentiWordNet, and machine learning oriented techniques relying on Naive Bayes, Maximum Entropy, and Support Vector Machine (SVM) algorithms.

Pang & Lee [15] also came up with the approach based on sentence as being subjective or objective and then perform the sentiment classification on the subjective portion of sentence. But the results proved that it is not sufficient for predicting the sentiment of entities. Subjectivity and sentiment are both relevant properties of language. Subjectivity refers to linguistic expression of somebody’s opinion, beliefs, and speculations. Main task of subjectivity is to classify the contents in objective or subjective.

Turney [17] used the lexicon-based approach that calculates the semantic orientation of words or phrases in the document to evaluate the document’s orientation. This is carried on by predicting the average semantic orientation of the adjectives and adverbs present in the phrases of the text. Any phrase is classified as positive if it as good associations and negative if it has bad associations. Turney proposed most challenging and effective model for sentiment classification which is based on document – level which involves two approaches: Term counting and machine learning approach. Term counting approach involves deriving a sentiment measure by calculating the positive and negative terms.

**Machine Learning based Sentiment Analysis on public reviews**

Altrabshes et al. [1] proposes a SA-E system architecture for sentiment analysis of student feedback data, which gives a summary from the techniques of collecting the data to pre-processing to machine learning and evaluations. It suggests the analysis of data using naïve bayes and SVM as they have been proven to work well with the educational data.

Neethu M.S. and Rajasree R. [5] analyzed the twitter posts about electronic products like mobiles, laptops etc using Machine Learning approach. They proposed that by doing sentiment analysis in a speciﬁc domain, it is possible to identify the effect of domain information in sentiment classiﬁcation.

Vapnik [18] proposed Support Vector Machine (SVM), which belongs to supervised learning method which classifies the data into two categories by constructing the N-dimensional hyper plane.

In the educational domain different researches on machine learning have given different results. Troussas el al. [23] found Naïve bayes to be the best method while Song et al.[24] found SVM to be best to do so.

Altrabshes et al. [11] proposed a study that evaluated the best model for sentiment analysis by considering four aspects: levels of pre-processing, features, machine learning techniques and the use of neutral class. It concluded that the unigrams performed well for almost all the models and SVM-linear was the best method with an accuracy of 95% followed by SVM radial basis kernel with 88% accuracy. Their results showed that SVM gave extremely high performance. SVM has been found to perform very well in varied domains, including movie reviews, customer feedback etc. They also observed better performance without including the neutral class.

**Text Summarization work on public reviews**

Text summarization can be carried out in two ways; extractive and abstractive. In extractive based text summarization, Rada Mihalcea[6]presents an innovative unsupervised method for automatic sentence extraction using graph based ranking algorithms. Rafael Ferreira et.al.[2] described and performed a quantitative and qualitative assessment of 15 algorithms available for sentence scoring in the literature. They evaluated three different datasets (including news, blogs and article contexts).

On the other hand, for abstractive based summarization, Ramesh Nallapati et.al.[7] modelled abstractive text summarization using Attentional Encoder Decoder Recurrent Neural Networks, and show that they achieve state-of-the-art performance on two different corpora.

Abigail See et.al.[8] proposed a novel architecture that augments the standard sequence-to-sequence attentional model in two orthogonal ways. First, they use a hybrid pointer-generator network that can copy words from the source text via pointing, which aids accurate reproduction of information, while retaining the ability to produce novel words through the generator. Second, we use coverage to keep track of what has been summarized, which discourages repetition.

After reviewing the related works we have found that almost none of them has addressed faculty profile generation by combining both sentiment analysis and text summarization.

**CHAPTER 3**

**PRELIMINARY CONCEPTS**

In this chapter, we have covered all the techniques and methodologies related to our task.

**3.1 Sentiment Analysis Overview**

A sentiment is a feeling, attitude, emotion or opinion of a user or opinion holder with respect to any particular topic. Sentiments are subjective impressions and have nothing to do with facts. Examining any text to identify these sentiments is called Sentiment Analysis.

Sentiment analysis is a novel technique that facilitates to investigate the thinking and thought of user. It can be defined as: “*Sentiment analysis, also called opinion mining, is the field of study that analyzes people’s opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes.*” Sentiment analysis refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials. Sentiment analysis is widely applied to reviews and social media for a variety of applications, ranging from marketing to customer service. Generally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document. The attitude may be his or her judgment or evaluation, affective state (the emotional state of the author when writing), or the intended emotional communication (that is to say, the emotional effect the author wishes to have on the reader). A basic task in sentiment analysis is classifying the polarity of a given text at the document, sentence, or feature/aspect level—whether the expressed opinion in a document, a sentence or an entity feature/aspect is positive, negative, or neutral. Advanced, "beyond polarity" sentiment classification looks, for instance, at emotional states such as "angry", "sad", and "happy".

In general, sentiment analysis has been investigated mainly at three levels:

* **Document level**: The task at this level is to classify whether a whole opinion document expresses a positive or negative sentiment. For example, given a product review, the system determines whether the review expresses an overall positive or negative opinion about the product. This task is commonly known as document-level sentiment classification. This level of analysis assumes that each document expresses opinions on a single entity (e.g., a single product). Thus, it is not applicable to documents which evaluate or compare multiple entities. [26] Have discussed the use of learning based approach for document level sentiment classification.
* **Sentence level**: The task at this level goes to the sentences and determines whether each sentence expressed a positive, negative, or neutral opinion. Neutral usually means no opinion. This level of analysis is closely related to subjectivity classification which distinguishes sentences (called objective sentences) that express factual information from sentences (called subjective sentences) that express subjective views and opinions. Works on sentence level and phrase level polarity classifications have been discussed in [16] and [20].
* **Entity and Aspect level**: Both the document level and the sentence level analyses do not discover what exactly people liked and did not like. Aspect level performs finer-grained analysis. Aspect level was earlier called feature level. Instead of looking at language constructs (documents, paragraphs, sentences, clauses or phrases), aspect level directly looks at the opinion itself. It is based on the idea that an opinion consists of a sentiment (positive or negative) and a target (of opinion). An opinion without its target being identified is of limited use. Realizing the importance of opinion targets also helps us understand the sentiment analysis problem better. For example, although the sentence “although the service is not that great, I still love this restaurant” clearly has a positive tone, we cannot say that this sentence is entirely positive. In fact, the sentence is positive about the restaurant (emphasized), but negative about its service (not emphasized). In many applications, opinion targets are described by entities and/or their different aspects. Thus, the goal of this level of analysis is to discover sentiments on entities and/or their aspects. [25] highlights work on aspect level sentiment analysis.

Sentiment Analysis has found applications and insights in various fields today like: Business Intelligence, Politics/Political Science, Law/Policy Making, Education, Sociology, and Psychology. Sentiment Analysis when combined with Natural Language processing, machine learning methods or statistics helps to extract, identify or otherwise characterize the sentimental contents of the text from any of the domains specified whereas above.

This work is based upon the sentiment analysis in educational field. We are required to perform sentiment analysis on the student feedback data given by students at the end of the course or at the end of the semester.

Train Test Split

Vectorization

Data Pre-Processing

Retrieve Data from Data Source

Evaluating The Model

Making Predictions on the Test Data

Training the Model

**Fig. 3.1 Machine learning based approach for sentiment analysis**

**Fig 3.2 Lexicon based approach for sentiment analysis**

The fig. 3.1 explains the flow diagram of sentiment analysis. As seen in the diagram the data

Classification

Dictionary Based Scoring

Tokens

Data from Data Source (Sentences)

Fig. 3.1 and 3.2 explain the flow diagram of sentiment analysis based on the machine learning approach and lexicon based approach for sentiment analysis respectively. As seen in the Fig. 3.1 for sentiment analysis based on the machine learning based approach the data is first retrieved and then pre-processed. Once pre-processed the next step is the vectorization of the data. In order for the college feedback data to make sense to the machine learning algorithms we will need to convert each feedback to a numeric representation. This process of converting textual data (student feedback in our case) to numerical representation is known as vectorization. Once vectorized the next step is data partitioning i.e. dividing the data into train and test data. The train data consists of the student feedbacks and their corresponding class labels (Positive/Negative). Once the model is trained then the next step is to make predictions for the class labels of the feedbacks of the test data. After making the predictions the next step is to evaluate the model on the basis of certain performance metrics the primary metric being the classifier accuracy.

While as seen in the fig. 3.2 for sentiment analysis based on the lexicon based approach, the first step is the tokenization of the feedback on the sentence and the word level. Once tokenized the words/sentences are assigned scores according to the lexicon polarity scores in the dictionary based on the method used (some methods assign polarity score to the entire sentence while some assign polarity score to one word of a sentence at a time) . The normalized score for the entire sentence is derived based on the normalization formula for specific method/algorithm used. The classification of the feedback based on the polarity scores is done as:

Polarity score<0 : Negative

Polarity score = 0 : Neutral

Polarity score>0 : Positive

**3.2 Classification Overview**

Classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts, for the purpose of being able to use the model to predict the class of objects whose class label is unknown. The derived model is based on the analysis of a set of training data (i.e., data objects whose class label is known).

The derived model may be represented in various forms, such as classification (IF-THEN) rules, decision trees, mathematical formulae, or neural networks. A decision tree is a flow-chart-like tree structure, where each node denotes a test on an attribute value, each branch represents an outcome of the test, and tree leaves represent classes or class distributions. Decision trees can easily be converted to classification rules. A neural network, when used for classification, is typically a collection of neuron-like processing units with weighted connections between the units. There are many other methods for constructing classification models, such as naïve Bayesian classification, support vector machines, and k-nearest neighbour classification.

An example would be assigning a given email into "spam" or "non-spam" classes or assigning a diagnosis to a given patient as described by observed characteristics of the patient (gender, blood pressure, presence or absence of certain symptoms, etc.). Classification is an example of pattern recognition.

Data classification is a two-step process. In the first step, a classifier is built describing a predetermined set of data classes or concepts. This is the learning step (or training phase), where a classification algorithm builds the classifier by analyzing or “learning from” a training set made up of database tuples and their associated class labels. A tuple, X, is represented by an n-dimensional attribute vector, X = (x1, x2,..., xn), depicting n measurements made on the tuple from n database attributes, respectively, A1, A2,..., An. Each tuple, X, is assumed to belong to a predefined class as determined by another database attribute called the class label attribute. The class label attribute is discrete-valued and unordered. It is categorical in that each value serves as a category or class. The individual tuples making up the training set are referred to as training tuples and are selected from the database under analysis. In the context of classification, data tuples can be referred to as samples, examples, instances, data points, or objects.

Because the class label of each training tuple is provided, this step is also known as supervised learning (i.e., the learning of the classifier is “supervised” in that it is told to which class each training tuple belongs). It contrasts with unsupervised learning (or clustering), in which the class label of each training tuple is not known, and the number or set of classes to be learned may not be known in advance. For example, if we did not have the loan decision data available for the training set, we could use clustering to try to determine “groups of like tuples,” which may correspond to risk groups within the loan application data.

This first step of the classification process can also be viewed as the learning of a mapping or function, y = f(X), that can predict the associated class label y of a given tuple X. In this view, we wish to learn a mapping or function that separates the data classes. Typically, this mapping is represented in the form of classification rules, decision trees, or mathematical formulae. The rules can be used to categorize future data tuples, as well as provide deeper insight into the database contents. They also provide a compressed representation of the data.

In the second step, the model is used for classification. First, the predictive accuracy of the classifier is estimated. If we were to use the training set to measure the accuracy of the classifier, this estimate would likely be optimistic, because the classifier tends to overfit the data (i.e., during learning it may incorporate some particular anomalies of the training data that are not present in the general data set overall). Therefore, a test set is used, made up of test tuples and their associated class labels. These tuples are randomly selected from the general data set. They are independent of the training tuples, meaning that they are not used to construct the classifier. The accuracy of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier. The associated class label of each test tuple is compared with the learned classifier’s class prediction for that tuple. Section 6.13 describes several methods for estimating classifier accuracy. If the accuracy of the classifier is considered acceptable, the classifier can be used to classify future data tuples for which the class label is not known. (Such data are also referred to in the machine learning literature as “unknown” or “previously unseen” data.).

An algorithm that implements classification, especially in a concrete implementation, is known as a classifier. The term "classifier" sometimes also refers to the mathematical function, implemented by a classification algorithm that maps input data to a category.

Often, the individual observations are analyzed into a set of quantifiable properties, known variously as explanatory variables or features. These properties may variously be categorical (e.g. "A", "B", "AB" or "O", for blood type), ordinal (e.g. "large", "medium" or "small"), integer-valued (e.g. the number of occurrences of a particular word in an email) or real-valued (e.g. a measurement of blood pressure). Other classifiers work by comparing observations to previous observations by means of a similarity or distance function.

**3.3 Text Summarization Overview**

Automatic Text Summarization is a process of generating a concise and meaningful summary of text from multiple text resources such as books, news articles, blog posts, research papers, emails, and tweets.

Text summarization can broadly be divided into two categories — Extractive Summarization and Abstractive Summarization.

**3.3.1 Extractive Summarization**: These methods rely on extracting several parts, such as phrases and sentences, from a piece of text and stack them together to create a summary, i.e. attempt to summarize articles by selecting a subset of words that retain the most important points.

This approach weights the important part of sentences and uses the same to form the summary. Different algorithm and techniques are used to define weights for the sentences and further rank them based on importance and similarity among each other.

**Input document → sentences similarity → weight sentences → select sentences with higher rank.**

Therefore, identifying the right sentences for summarization is of utmost importance in an extractive method. Extractive summarization techniques involve methods such as word scoring, word frequency (TF-IDF), word co-occurrence (N-gram overlapping), graph scoring (Text Rank).

**3.3.2 Abstractive Summarization**: Abstractive methods select words based on semantic understanding; even those words did not appear in the source documents. It aims at producing important material in a new way. They interpret and examine the text using advanced natural language techniques in order to generate a new shorter text that conveys the most critical information from the original text. Abstractive text summarization techniques makes use of recurrent neural networks for generating new text for making the summaries look like more of a human generated summary.

It can be correlated to the way human reads a text article or blog post and then summarizes in their own word.

**Input document → understand context → semantics → create own summary.**

**3.4 Tools Used**

**3.4.1 Python**

Python is a widely used high-level programming language for general-purpose programming, created by Guido van Rossum and first released in 1991. Python features a dynamic type system and automatic memory management and supports multiple programming paradigms, including object-oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library. Large organizations that make use of Python include Wikipedia, Google, Yahoo!, CERN, NASA, and some smaller entities like ILM, and ITA. The social news networking site Reddit is written entirely in Python.

**Packages Used**

**(A) Scikit-Learn**

Licensed under a permissive simplified BSD license, Scikit-learn [34] is a machine learning software library freely available for python programming languages. Still under active development it was initially developed by David Cournapeau. Till date five official updates have been released. It is largely written in Python with some core algorithms done in Cython to achieve better performance. It provides a wide range of supervised as well as unsupervised machine learning algorithms. It is designed to work with the numerical and scientific libraries of Python like NumPy and SciPy. It features various classification, clustering and regression algorithms.

**(B) NLTK**

Natural Language Toolkit also known as NLTK [33], is a library to deal with natural language text for English written in Python programming language. NLTK was initially developed by Steven Bird and Edward Loper. I’ve used NLTK for pre-processing and removing unwanted data from our input file.

**IDE Used**

**(A) Anaconda**

Anaconda [32] is a free open source distribution of Python and R. It is used for scientific computing, large scale data processing and predictive analysis. Anaconda primarily aims to simplify the management and deployment of the various packages of python and R. This is achieved by its package management system known as conda. Anaconda basically provides an environment to use its various interactive services. The one used here is Jupyter. Jupyter [31] provides a browser based notebook that works on parallel and distributed computing. It connects to an IPython kernel. It supports code, text, mathematical expressions, inline plots along with support for interactive data visualization and GUI toolkits.

**CHAPTER 4**

**METHODOLOGY**

In our work the major emphasis has been put on sentiment analysis and text summarization of student feedbacks. Sentiment analysis is done using various techniques which are later compared using various performance parameters and the one with the best results is preferred and text summarization is done using the extractive technique based on Text Rank algorithm. Finally, a faculty profile is generated based on the student feedbacks.

* 1. **Proposed Architecture**

Our proposed solution consists of a combination of sentiment analysis and text summarization technique. The proposed architecture of the same is given in fig. 4.1. The architecture consists of various techniques applied at each step of the work in order to achieve the goal of generating a faculty profile using data mining techniques. The input to our work is the student feedback which is taken using an already existing online feedback system in IET, Indore. This system collects student’s feedback and relevant data can be extracted from this system for further analysis.

The various steps involved in the architecture are as follows:

1. Collection of feedback through the existing online feedback collection system.
2. Then we apply various lexicon based and machine learning based approaches sentiment analysis techniques on the student feedback data.
3. Evaluating all the sentiment analysis models.
4. Applying the optimal model for sentiment analysis obtained from the former step to the feedbacks.
5. Using the results of feedbacks classification for faculty profile generation through extractive summarization techniques.

The proposed architecture aims at performing sentiment analysis on the student feedbacks and generating faculty profile based on the student feedback.

Student Feedback System @ IET- DAVV, Indore

Machine Learning Based SA

Lexicon Based SA

VADER

SVM Classifier

Naïve Bayes Classifier

SVM Linear, RBF

Multinomial NB

Evaluate Sentiment Analysis Models

Feedback Classification

Separate Highly Polarized Feedbacks

Extractive Text Summarization

**Fig. 4.1 Proposed Architecture for Faculty Profiling Based On Student Feedbacks Using Sentiment Analysis**

**4.2 Existing Feedback Application**

The student feedback is collected at the end of every semester in IET, DAVV using the existing feedback application. The feedback is then stored in the database and relevant information is retrieved whenever required.

**4.3 Sentiment Analysis**

In the sentiment analysis phase we have experimented upon our dataset using three different approaches and evaluated the performance of those to take the results from the method having the best performance metrics. The different approaches followed for sentiment analysis are described below.

**4.3.1 Lexicon based classification**

The lexicon-based approach involves calculating orientation for a document from the semantic orientation of words or phrases in the document (Turney 2002) [17]. The lexicon based approach for sentiment analysis comprises of three steps described below.

1. Tokenization
2. Dictionary based scoring
3. Classification

**(i) Tokenization**

Tokenization is the process of splitting the text into tokens. These tokens could be paragraphs, sentences, or individual words. Tokenization needs to be done because amongst different lexicon based approaches some of the methods perform sentence level sentiment scoring while, the others perform word level sentiment scoring.

**(ii) Dictionary based scoring**

Each token (word/sentence based on the approach used) is assigned a score based on the scores of the lexicons in the dictionary then the normalized sentiment score for the sentence is determined.

**(iii) Classification**

Once the sentiment scores of the sentences are normalized the sentences whose polarity is found to be less than are classified as negative, those with the polarity equivalent to zero are classified as neutral and sentences with the polarity greater than zero are classified as positive.

**4.3.2 Machine Learning Based Sentiment Analysis**

The machine learning based approach of sentiment analysis typically involves three steps.

1. Text Pre-Processing
2. Vectorization
3. Feedback Classification

**(i) Text Pre-Processing**

Pre-processing is the process of cleaning and organizing the data and preparing the text for classification. Free text usually contains lots of noise and unnecessary parts that need to be discarded. Pre-processing the data reduces the noise which helps to improve the performance of the classifier. One of the advantages of pre-processing is that it speeds up the classification process and improves the performance to a great extent. Many researchers have shown that appropriate text pre-processing including data transformations and filtering can significantly improve the performance.

**(ii) Vectorization**

Vectorization is the process of dividing the data and then converting it into numerical form. The feedbacks are extracted from the input file and stored in a feature matrix and then it is count vectorized using a vectorizer, .i.e., it is converted to a document-term matrix. The score for each suggestion is stored in a response vector and used later on during the time of evaluations. Therefore, the output of vectorization is feature matrix and response vector, which are fed into the feedback classification module and aspect identification module.

**(iii) Feedback Classification**

Data classification is the process of organizing data into categories for its most effective and efficient use. It is a data mining function that assigns items in a collection to target categories or classes, i.e., it is used to predict categorical class labels. The goal of classification is to accurately predict the target class for each case in the data (newly available data) based on training set and class labels. Classification procedure is recognized method for repeatedly making such decisions in new situations. For instance in our case the free text that is provided by the students of IET in the form of feedback suggestions needs to be assigned to either of the two classes, i.e., positive or negative.

In machine learning, classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations whose category membership is known. In their works [19] and [21] have discussed about the use of machine learning techniques for sentiment analysis using text classification. An example of text classification would be assigning a given email into "spam" or "non-spam" classes or assigning a diagnosis to a given patient as described by observed characteristics of the patient (gender, blood pressure, presence or absence of certain symptoms, etc.). Classification is an example of pattern recognition. In the terminology of machine learning, classification is considered an instance of supervised learning, i.e. learning where a training set of correctly identified observations is available. An algorithm that implements classification, especially in a concrete implementation, is known as a classifier. The term "classifier" sometimes also refers to the mathematical function, implemented by a classification algorithm, which maps input data to a category.

The following classifiers are used when classifying using machine Learning:

* Naïve Bayes- Multinomial Naïve Bayes, Complimentary Naïve Bayes.
* SVM- Linear, Radial basis function.

**4.3.2.1 Naïve Bayes**

Naïve Bayes is a machine learning algorithm for classification problems. Based on the Bayes’ probability theorem, it is not only the most simple classification model but the most affective too. Primarily used for text classification, it involves high dimensional training data sets. A few examples are spam filtration, sentimental analysis, and classifying news articles. Naïve Bayes algorithm is based on the Bayes’ theorem with an assumption of independence among predictors. In simple terms, a Naïve Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Bayes’ theorem provides for the calculation of posterior probability, .i.e., the probability of an item x being present in the class c based on the previous knowledge available on the events.

***P(c|x) = P(x|c) P(c)***

***P(x)***

* P(c|x) : posterior probability of class (c, target) given predictor (x, attributes).
* P(c) : prior probability of class c.
* P(x|c) : likelihood which is the probability of predictor given class.
* P(x) : prior probability of predictor.

**Multinomial Naïve Bayes**

The variant of Naïve Bayes algorithm used when the data is distributed multinomial, and is one of the two classic naïve Bayes variants used in text classification (where the data are typically represented as word vector counts, although tf-idf vectors are also known to work well in practice). It is used for discrete counts. Multinomial naïve bayes  implements the naive Bayes algorithm for multinomially distributed data, and is one of the two classic naive Bayes variants used in text classification (where the data are typically represented as word vector counts, although tf-idf vectors are also known to work well in practice). The distribution is parametrized by vectors θy=(θy1,…,θyn) for each class y, where n is the number of features (in text classification, the size of the vocabulary) and θyi is the probability P(xi∣y) of feature i appearing in a sample belonging to class y.

**4.3.2.2 SVM**

Yielding the highest performance and accuracy results in text classification problems, the next classifier that is used is the SVM. SVMs function by making a hyperplane with maximum Euclidean distance to the closest training examples, i.e., it maps examples as points in a high-dimensional space where separate classes are divided by an as wide as possible tangential distance to the hyperplane. New set of examples are mapped on to either side of the hyperplane depending on which side of the hyperplane or which class it is predicted to belong. The hyperplane is determined by a small subset of the training data called the support vector and the rest of the training data has no effect on the trained classifier.

Maximum Margin Hyperplane

**Fig. 4.3 Maximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors.**

**SVM-linear**

SVM-linear is an extremely fast linearly separable machine learning algorithm for solving multiclass problems. SVM-linear is a linearly scalable routine meaning that it creates an SVM model in a CPU time which scales linearly with the size of the training data set. It is efficient in dealing with extra-large data sets (say, several millions training data pairs), provides solution of multiclass classification problems with any number of classes, works effortlessly with high dimensional data (thousands of features, attributes) in both sparse and dense format, doesn’t need expensive computing resources (personal computer is a standard platform), is ideal for contemporary applications in digital advertisement, e-commerce, web page categorization, text classification, bioinformatics, proteomics, banking services and many other areas.

**SVM-Radial Basis Function**

Problems that are non-solvable by Linear SVM or are non-linear are solved using Non-linear SVM by using kernel function. As shown in fig. 4.4 it separates the non-linearly separable using a kernel by mapping it onto a linear feature space.

Feature Space

Input Space



**4.4 Evaluating Classification Models**

Classifier performance depends greatly on the characteristics of the data to be classified. There is no single classifier that works best on all given. Various empirical tests have been performed to compare classifier performance and to find the characteristics of data that determine classifier performance. Determining a suitable classifier for a given problem is however still more an art than a science.

**4.4.1 Classification Accuracy**

When a model is built for a classification problem the most naïve way of evaluating the performance is by calculating the accuracy of the model. Therefore, it is always necessary to look at the accuracy of that model. This is the classification accuracy.

Classification accuracy refers to the ability of the model to correctly predict the class label of new or previously unseen data.

*Accuracy = Correctly classified data*

*Total Data*

**Accuracy paradox**

Accuracy is often the starting point for analyzing the quality of a classification model, as well as an obvious criterion for prediction. Accuracy measures the ratio of correct predictions to the total number of cases evaluated. It may seem obvious that the ratio of correct predictions

to cases should be a key metric. A classification model may have high accuracy, but be useless.

In the field of data science and specifically the problem of classification, a confusion matrix, a special kind of contingency table, is a specific table layout that allows visualization of the performance of a classification model, with two dimensions ("actual" and "predicted") i.e., each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class.

|  |  |  |
| --- | --- | --- |
|  | Classified Positive | Classified Negative |
| Positive Class | *True Positive*  *(TP)* | *False Negative*  *(FN)* |
| Negative Class | *False Positive*  *(FP)* | *True Negative*  *(TN)* |

Fig. 4.4 Confusion Matrix for a classification problem.

The different terminology of the Confusion matrix is defined as follows:

* True Positives (TP): Number of examples correctly labelled as positive, i.e.,that were actually positive and also labelled positive.
* False Positives (FP): Number of examples wrongly labelled as positive examples, i.e., that were actually negative but were labelled as positive.
* True Negatives (TN): Number of examples correctly labelled as negative, i.e., that were actually negative and also labelled negative.
* False Negatives (FN): Number of examples wrongly labelled negative examples, i.e., those were actually positive but were labelled as negative.

When we calculate Accuracy using the confusion matrix the formula becomes:

*Accuracy = (TP + TN)*

*(TP + TN + FP + FN)*

**4.4.2 Precision and Recall**

Precision is the ratio of the number of positive examples that were correctly classified to the total number of examples that were classified as positive by the model. It is usually expressed as a percentage.

*Precision = TP*

*(TP+FP)*

Recall is the ratio of the number of positive examples that were correctly classified to the total number of actually positive examples in the dataset. It is usually expressed as a percentage.

*Recall = TP*

*(TP+FN)*

**4.4.3 F-Score**

It’s the harmonic mean of precision and recall. F1 score is an 'average' of both precision and recall. We use the harmonic mean because it is the appropriate way to average ratios (while arithmetic mean is appropriate when it conceptually makes sense to add things up).

*F-score = 2 \* (PRECISION x RECALL)*

*(PRECISION + RECALL)*

**4.5 Feedback Classification**

We have evaluated all the machine learning models trained on the student feedback data and also the lexicon based methods based on the above parameters, accuracy, precision, recall and F1- Score and finally, we chose the best model for further analysis.

**4.6 Separating Highly Polarized Feedbacks**

Classified feedbacks obtained from the previous phase are further analyzed based on their polarity scores and highly polarized feedbacks i.e. those feedbacks which are either highly positive or highly negative are separated for the purpose of faculty profile generation.

**4.7 Text Summarization (Faculty Profile Generation)**

In this phase the highly polarized feedbacks with respect to each faculty are summarized using the Text Rank algorithm which is an extractive approach for text summarization to obtain a brief profile of the faculty.

**CHAPTER 5**

**IMPLEMENTATION & RESULT ANALYSIS**

**5.1 Implementation:**

The implementation of the work is done in Python. Python is used due its rich open source libraries which provide ease over other languages like java and matlab. Since the major use of python is used in data mining therefore it is apt for performing sentiment analysis as it is a part of data mining. The step by step implementation is as follows:

* 1. **Major tasks in our Work**

**I Feedback Collection from existing feedback application and feedback pre- processing:**

The student feedback is collected at the end of every semester in IET-DAVV, Indore using the existing feedback application. The online feedback form contains a mix of free-response and quantitative (also called Likert scale) questions. The Likert Scale questions are easy to evaluate as the answer to these questions are scores only. The free responses in the feedback form contain three questions asking about the strengths, weaknesses and suggestions about the respective faculty for whom the feedback is being taken.

**Section A** Very poor Poor Good Average Excellent

01 Teacher’s subject knowledge 0 0 0 0 0

02 Compliments and coverage of course 0 0 0 0 0

03 Compliments theory with practical examples 0 0 0 0 0

04 Teacher’s computer/IT skills 0 0 0 0 0

06 Teacher’s overall performance 0 0 0 0 0

**Section B**

1 Result of test declared within 0 Yes 0 No 0 No comments

two weeks of it being conducted

2 Adequate number of assignments/ 0 Yes 0 No 0 No comments

cases given

**Section C**

a. What are the strengths of the teacher \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

b. What are the areas of weaknesses in teacher\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

c. Any other suggestions (regarding curriculum/subject(s), faculty)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Fig. 5.1 Student’s feedback form IET-DAVV, Indore**

The collected student feedback is stored in the database in SQL format. The feedback data is needed to be extracted from the database into a format that can be read into python for further processing. For our work we have exported the SQL data to CSV format. The following SQL commands were used to accomplish the task of extracting the feedback.

>sqlite3 C:\sqlite\feedback.db

sqlite>.headers on

sqlite>.output feedback\_data.csv

sqlite>SELECT strengths,

weaknesses,

suggestions

from feedback\_table;

sqlite>quit

**Fig 5.2 Exporting feedbacks from Database to CSV format**

Once the feedback is collected, it is stored in the database. Now, our task is to perform sentiment analysis, this task cannot be achieved using the quantitative questions as they already rate particular features, therefore we extract the suggestions provided by the students from the database to achieve our goals. This retrieved information is the raw data that may contain discrepancies which will be tackled within the next step.

Some distortion or unwanted text, html tags, links and stop words etc. are included in raw data. This distortion or noise may not be suited when we want to feed the data to the machine learning algorithms for text classification. This module cleans such unwanted data from the feedback. Other than the student feedbacks from the IET-DAVV student feedbacks system we have also taken into account the feedbacks from the MOOC course feedbacks.

communicative lectures lengthy,

interactive

good clearing concepts

brings conceptual clarity best way

good teaching

0 communicative but lectures are lengthy.

1 Interactive.

2 good at clearing the concepts.

3 brings conceptual clarity in the best way.

4 good at teaching.

**Fig. 5.3 Raw feedback data Fig. 5.4 Feedback data after preprocessing**

**II Applying Sentiment Analysis on student feedbacks**

Once the data is collected from the Student Feedback System at IET-DAVV and pre-processed, the data is then subjected to sentiment analysis. In our work we have applied both lexicon based approach and machine learning based approach for sentiment analysis, the one with the best performance was considered for further working. The work flow of both the approaches during the experimentation was as described below.

**II A Machine learning based approach for sentiment analysis**

The machine learning based approach to sentiment analysis is comprised of the following steps as described below.

1. **Vectorization:** For textual data to make sense to the machine learning algorithms we need to convert textual data to a numeric representation. The process of converting textual data to numeric representation is known as vectorization. In our work for feeding the student feedback data to the machine learning algorithms for sentiment classification we have converted the feedback data to numeric representation with the sklearn module vectorize() method.

Feedback:

good at clearing the concepts interactive with students deep knowledge of subject practical knowledge. best teacher for c++. makes us understand things very nicely. subject knowledge is excellent little bit angry. good conceptual clarity. his knowledge about programming is excellent and he deliver tiny to tiny concepts to students. Sir is the best to teach computer programming they know very minute and important things regarding subject.

Vectorizer Vocabulary:

{'good': 11, 'clearing': 4, 'concepts': 6, 'interactive': 13, 'students': 25, 'deep': 8, 'knowledge': 15, 'subject': 26, 'practicle': 21, 'best': 1, 'teacher': 28, 'makes': 17, 'us': 32, 'understand': 31, 'things': 29, 'nicely': 20, 'excellent': 10, 'little': 16, 'bit': 2, 'angry': 0, 'conceptual': 7, 'clarity': 3, 'programming': 22, 'deliver': 9, 'tiny': 30, 'manoj': 18, 'sir': 24, 'teach': 27, 'computer': 5, 'know': 14, 'minute': 19, 'important': 12, 'regarding': 23}

Matrix representation:

[[0.1118034 0.2236068 0.1118034 0.1118034 0.1118034 0.1118034 0.2236068

0.1118034 0.1118034 0.1118034 0.2236068 0.2236068 0.1118034 0.1118034

0.1118034 0.4472136 0.1118034 0.1118034 0.1118034 0.1118034 0.1118034

0.1118034 0.2236068 0.1118034 0.1118034 0.2236068 0.3354102 0.1118034

0.1118034 0.2236068 0.2236068 0.1118034 0.1118034]]

**Fig. 5.5 Vectoriztion of feedback data and matrix representation**

1. **Data Partitioning:** Once vectorized the data is to be partitioned into the train and test data for the purpose of training and cross validation. The test data consists of a fragment of data which is completely different from the data in the training set.In our experiments we have partitioned the data using the train\_test\_split() method of the scikit learn module. With three different data fragments 1:2, 6:4 and 7:3 respectively.
2. **Defining and Training the Model:** This is the process of defining the machine algorithm that is to be used for the purpose of classification and then training the model on the train data set obtained by the data partitioning in the former step. Different algorithms used in our experiments are Multinomial naïve bayes and support vector machine with linear and radial basis function kernel respectively.

x\_train,x\_test,y\_train,y\_test = (X,y,test\_size=0.3)

from sklearn import naïve\_bayes

clf = naïve\_bayes.MultinomialNB()

clf.fit(x\_train,y\_train)

**Fig. 5.6(a) Data partitioning and training NB model**

x\_train,x\_test,y\_train,y\_test = (X,y,test\_size=0.3)

from sklearn import svm

clf = svm.SVC(kernel = ”linear”)

clf.fit(x\_train,y\_train)

**Fig 5.6(b) Data partitioning and training SVM model**

1. **Making Predictions:** This process involves making predictions on the test data based on the learnings from the training data.
2. **Evaluating the Model:** This process involves assessment of the metrics that evaluate the performance of the model. The metrics that we have used for assessing the performance of the machine learning algorithms are accuracy, precision, recall and f-score.

precision recall f1-score support

0 1.00 0.07 0.13 148

1 0.80 1.00 0.89 538

avg / total 0.84 0.80 0.72 686

precision recall f1-score support

0 0.90 0.37 0.52 98

1 0.84 0.99 0.91 331

avg / total 0.85 0.85 0.82 429

**Fig. 5.7 Evaluation of NB and SVM models based on the performance metrics**

**II B Lexicon Based Approach for Sentiment Analysis: Valence Aware Dictionary for sEntiment Reasoning (VADER)**

VADER sentiment analysis calculates the sentiment score of an input text. It combines a dictionary, which maps lexical features to emotion intensity, and five simple heuristics, which encode how contextual elements increment, decrement, or negate the sentiment of text. The greatest thing about VADER sentiment analysis is that the colloquialisms i.e. emoticons like☺, acronyms like LOL, and slang like “meh” also get mapped to intensity values as well. Emotion intensity or sentiment score is measured on a scale from -4 to +4, where -4 is the most negative and +4 is the most positive. The midpoint 0 represents a neutral sentiment.

After assigning sentiment score to each and every word ranging from -4 to +4 VADER sentiment intensity analyzer normalizes the sentiment score of the sentence in the range -1 to +1 using the Hutto’s normalization formula:

*X*

Where,

X = sum of the sentiment scores of all the constituent words of a given feedback.

= Normalization parameter set to 15.

The sentences are classified into positive and negative according to the intervals of the sentiment score, score<0: Negative, score=0: Neutral, score>0: Positive. When the performance of the machine learning based approaches and lexicon based approaches for sentiment analysis are compared, the performance of the lexicon based approach is found to be far better and therefore, the results of lexicon based classification were carried forward for the further work.

from nltk.sentiment.vader import SentimentIntensityAnalyzer

def sentiment\_value(paragraph):

analyser = SentimentIntensityAnalyzer()

new\_words = {

'knowledge':4,

'irregular':-4,

'Inappropriate':-4,

'dont':-4,

'poor':-4,

'excellent':4,

'depth':4,

'case':4,

'Industrial':4,

'no':-4,

'degraded':-4,

'thorough':4,

'without':-4,

'not':-4,

'Insufficient':-4,

'cant':-4,

'complete':4,

'lack':-4,

'goes':-4,

'relates':4,

'trimendous':4,

'vast':4,

'dedication':4,

'wastes':-4

}

analyser.lexicon.update(new\_words)

result = analyser.polarity\_scores(paragraph)

score = result['compound']

return round(score,1)

good subjective knowledge.

Sentiment:

0.8

good conceptual skills and knowledge for subject.

Sentiment:

0.8

**Fig. 5.8 Sentiment Analysis using VADER**

The *lexicon.update()* method was used for adding some of the domain specific lexicons to the sentiment intensity analyzer to make the VADER sentiment intensity analyzer to make it fit for sentiment analysis on educational domain.

**II C Evaluating the best method for sentiment analysis:**

The lexicon based method and the 3 machine learning classifiers are compared based on various performance parameters. The performance parameters we used for our classification problem were accuracies, precision, recall and F-score.

Accuracies, precision, recall and F-score were calculated after the training and testing process for all the techniques, as the predicted and actual values need to be compared for the calculation of the performance parameters.

Firstly, the accuracy was calculated. Classification accuracy refers to the ability of the model to correctly predict the class label of new or previously unseen data. Then due to the accuracy paradox theory, confusion matrix was created using the true positive, true negative, false positive and false negatives. Using this it became very easy to calculate the other two performance parameters, i.e., precision and recall (or specificity and sensitivity). These two were calculated based on the true positives in the classification process. Precision is the ratio of the number of positive examples that were correctly classified to the total number of examples that were classified as positive by the model. Recall is the ratio of the number of positive examples that were correctly classified to the total number of actually positive examples in the dataset. These two are represented in percentage.

Now, comparing the 4 techniques based on 2 parameters can be difficult, therefore, another performance measure was brought into light, the F-score. F-score, the harmonic mean of specificity and sensitivity gave a perfect measure and evaluation criterion to compare the 4 techniques.

Therefore, for a simplified understanding we grouped the accuracies of the 4 techniques into one table while the precision, recall and F-scores into another table so as to make it easier to make correspondence between the precision, recall and F-scores.

**II D Separating highly polarized feedbacks for faculty profile generation**

According to the sentiment scores obtained from the VADER, the feedbacks were classified into five different categories namely positive, negative, neutral, very positive, very negative according to different intervals of sentiment scores so as to take only the highly polarized feedbacks (only the very positive, very negative ones) to the next phase which is text summarization for accomplishing the purpose of faculty profiling. The intervals used for feedback classification were

SENTIMENT\_VALUE = []

SENTIMENT = []

If(sentiment<=1 and sentiment>=0.5):

SENTIMENT.append(“Very Positive”)

SENTIMENT\_VALUE.append(5)

elif(sentiment<0.5 and sentiment>0):

SENTIMENT.append(“Positive”)

SENTIMENT\_VALUE.append(4)

elif(sentiment==0):

SENTIMENT.append(“Neutral”)

SENTIMENT\_VALUE.append(3)

elif(sentiment<0 and sentiment>=-0.5):

SENTIMENT.append(“Negative”)

SENTIMENT\_VALUE.append(2)

else:

SENTIMENT.append(“Very Negative”)

SENTIMENT\_VALUE.append(1)

**Fig. 5.9 Assigning different sentiment classes to the feedbacks according to their polarity scores**

|  |  |  |
| --- | --- | --- |
| **comments** | **SENTIMENT\_VALUE** | **SENTIMENT** |
| brings conceptual clarity in the best way. | 5 | Very Positive |
| good subjective knowledge | 5 | Very Positive |
| good at clearing the concepts | 5 | Very Positive |
| does not show test copies | 1 | Very Negative |
| course coverage is not up to the mark | 1 | Very Negative |
| does not explain the concepts | 1 | Very Negative |

**Fig 5.10 Highly polarized feedbacks separated for faculty profile generation**

**III Faculty profile generation**

Once the highly polarized feedbacks are obtained, the faculty profile generation is done using the *genism* summarizer based on the text rank algorithm, which is an extractive summarization technique.

import pandas as pd

from genism.summarization import summarize

df = pd.read\_csv(“clean\_feedbacks\_data.csv”)

print(“Feedback”)

print(df[“comments”][2])

print(“Faculty Profile”)

print(summarize(df[“comments”][2]))

**Fig. 5.11 Summarizing student feedbacks to generate faculty profile**

Feedback

good at clearing the concepts interactive with students deep knowledge of subject practical knowledge. best teacher for c++. makes us understand things very nicely. subject knowledge is excellent little bit angry. good conceptual clarity. his knowledge about programming is excellent and he deliver tiny to tiny concepts to students. sir is the best to teach computer programming they know very minute and important things regarding subject.

Faculty Profile

his knowledge about programming is excellent and he deliver tiny to tiny concepts to students.

**Fig. 5.12 Faculty profile obtained from student feedbacks**

**5.3 Results and Analysis**

All the experiments in our work have been performed on a computer with 4 gigabytes RAM and intel i3 4th generation processor.

|  |  |  |  |
| --- | --- | --- | --- |
| **Classification Technique** | **Accuracy** | | |
| **Training=50%**  **Testing=50%** | **Training=60%**  **Testing=40%** | **Training=70%**  **Testing=30%** |
| *Multinomial Naïve Bayes* | **82.4%** | **80.3%** | **83.07%** |
| *Linear-SVM* | 77.3% | 79.5% | 80.3% |
| *SVM-RBF* | 78.9% | 77.25% | 77.23% |

**Table 1 Accuracies of various Machine Learning Based Classification techniques on the IET-DAVV student feedback dataset**

|  |  |  |  |
| --- | --- | --- | --- |
| **Classification Technique** | **Accuracy** | | |
| **Training=50%**  **Testing=50%** | **Training=60%**  **Testing=40%** | **Training=70%**  **Testing=30%** |
| *Multinomial Naïve Bayes* | 91.14% | 91.02% | 90.98% |
| *Linear-SVM* | 91.14% | 91.20% | 91.29% |
| *SVM-RBF* | 91.02% | 91.14% | 91.22% |

Table 2 Accuracies of various Machine Learning Based Classification techniques

**on the Coursera Mooc student feedback dataset**

While accuracy is not always considered as the best evaluation parameter, therefore a look at the F-score would give precise and finer results. These results are shown in Table 2.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Classification Technique** | **Data Distribution** | | | | | | | | |
| **Training=50%**  **Testing=50%** | | | **Training=60%**  **Testing=40%** | | | **Training=70%**  **Testing=30%** | | |
| **Precision** | **Recall** | **F-score** | **Precision** | **Recall** | **F-score** | **Precision** | **Recall** | **F-score** |
| *Multinomial Naïve Bayes* | 84% | 82% | 79% | 83% | 80% | 76% | 84% | 83% | 80% |
| *SVM-Linear* | 60% | 77% | 67% | 84% | 80% | 72% | 84% | 80% | 73% |
| *SVM-RBF* | 62% | 79% | 70% | 60% | 77% | 67% | 60% | 77% | 67% |

**Table 3 Experimental Results showing Precision, Recall, F-score for the various Machine Learning Based Classification techniques on the IET-DAVV student Feedback dataset**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Classification Technique** | **Data Distribution** | | | | | | | | |
| **Training=50%**  **Testing=50%** | | | **Training=60%**  **Testing=40%** | | | **Training=70%**  **Testing=30%** | | |
| **Precision** | **Recall** | **F-score** | **Precision** | **Recall** | **F-score** | **Precision** | **Recall** | **F-score** |
| *Multinomial Naïve Bayes* | 84% | 91% | 87% | 84% | 91% | 87% | 84% | 91% | 87% |
| *SVM-Linear* | 83% | 91% | 87% | 83% | 91% | 87% | 83% | 91% | 87% |
| *SVM-RBF* | 83% | 91% | 87% | 83% | 91% | 87% | 83% | 91% | 87% |

**Table 4 Experimental Results showing Precision, Recall, F-score for the various Machine Learning Based Classification techniques on the Coursera mooc student feedback dataset**

|  |  |  |  |
| --- | --- | --- | --- |
| Accuracy | Precision | Recall | F-Score |
| 93% | 100% | 92% | 95% |

**Table 5 Experimental Results showing Accuracy, Precision, Recall, F-score for VADER on the IET-DAVV Student Feedbacks Dataset**

**CHAPTER 6**

**CONCLUSION AND FUTURE SCOPE**

**6.1 Conclusion**

We started our work with the motive of generating the faculty profile based on the student feedbacks by using sentiment analysis and text summarization and we have successfully accomplished the task.

Following conclusions can be drawn on the basis of the experiments in our work:

The lexicon based sentiment analysis technique we have used (VADER) has an accuracy of 93% and an F1- score of 95%, which is higher when compared to the machine learning based methods. Simultaneously, the precision, recall and F-score are also higher. This is to say that the overall performance of lexicon based sentiment analysis in our study is very high as compared to the machine learning based sentiment analysis.

In the learning based approach we have considered different supervised machine learning techniques. Naïve bayes (multinomial), SVM-Linear, SVM-RBF (radial basis function),. We then considered three different schemes for data distribution, i.e., 1:2, 3:2, 7:3. When different combinations of machine learning classifiers and data distribution schemes were tested, we found out that the performance of Naïve Bayes (multinomial) and SVM-linear is comparatively lesser as compared to VADER the lexicon based approach for sentiment analysis.

F-score which combines precision and recall into a single value, With the F-score of 0.95 the VADER gives a precision of 1.0 and recall of 0.92. This means that of all the feedbacks we classified as positive 100% are actually correct/positive and of all the actually positive feedbacks in the system the classifier could only classify 92% feedbacks a positive.

The accuracy of multinomial Naïve bayes evaluated to 84% and was therefore found to be very less as compared to VADER’s 93% for the sentiment analysis of student feedback data.

Therefore, we conclude that the lexicon based approach VADER outperforms both Naïve bayes as well as the SVM for mining educational data.

**6.2 Future Work**

There is a scope of future work in the following areas:

**6.2.1 Sentiment analysis**

With sentiment analysis being performed using the lexicon based approach and the learning based approach respectively. Hybrid approaches can also be tried and tested for accomplishing the task. Hybrid approaches tend to provide better performance as they apply the combination of both of the former techniques.

**6.2.2 Text Summarization**

Text summarization can also be achieved with the help of abstractive summarization techniques. Abstractive techniques make use of the encoder-decoder based neural network architecture. The summaries generated using the abstractive techniques sometimes also includes words that are not present in the source text, those words are generated by the model itself on the basis of the context of the source text.

**6.2.3 Based on the platform available**

We agree that we have used limited computational power for our experiment work. In future we would like to use Graphical Processing Unit (GPU) and cloud platforms as well for analyzing our results.

**REFRENCES**

[1] Hutto, Clayton J., and Eric Gilbert. "Vader: A parsimonious rule-based model for sentiment analysis of social media text." Eighth international AAAI conference on weblogs and social media. 2014.

[2] Ferreira, Rafael, et al. "Assessing sentence scoring techniques for extractive text summarization." Expert systems with applications 40.14 (2013): 5755-5764.

[3] Taboada M. Sentiment analysis: an overview from linguistics. Annual Review of Linguistics. 2016 Jan 14;2:325-47.

[4] Kontopoulos, Efstratios, et al. "Ontology-based sentiment analysis of twitter posts." Expert systems with applications 40.10 (2013): 4065-4074.

[5] Neethu, M. S., and R. Rajasree. "Sentiment analysis in twitter using machine learning techniques." 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT). IEEE, 2013.

[6] Mihalcea, Rada. "Graph-based ranking algorithms for sentence extraction, applied to text summarization." Proceedings of the ACL Interactive Poster and Demonstration Sessions. 2004.

[7] Nallapati, Ramesh, et al. "Abstractive text summarization using sequence-to-sequence rnns and beyond." arXiv preprint arXiv:1602.06023 (2016).

[8] See, Abigail, Peter J. Liu, and Christopher D. Manning. "Get to the point: Summarization with pointer-generator networks." arXiv preprint arXiv:1704.04368 (2017).

[9] Chaplot, Devendra Singh, Eunhee Rhim, and Jihie Kim. "Predicting Student Attrition in MOOCs using Sentiment Analysis and Neural Networks." AIED Workshops. Vol. 53. 2015.

[10] Hu, Meishan, Aixin Sun, and Ee-Peng Lim. "Comments-oriented document summarization: understanding documents with readers' feedback." Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2008.

[11] Wen, Miaomiao, Diyi Yang, and Carolyn Rose. "Sentiment Analysis in MOOC Discussion Forums: What does it tell us?." Educational data mining 2014. 2014.

language technologies 5.1 (2012): 1-167.

[12] Altrabsheh, N., Gaber, M. M., & Cocea, M. , “SA-E : Sentiment Analysis for Education”, 5th KES International Conference on Intelligent Decision Technologies, Portugal, pp. 353–362, 2013

[13] Kim, S. M., & Calvo, R. A., “Sentiment analysis in student experiences of learning”, In Proceedings of the 3rd international conference on educational data mining, pp. 111–120, 2010.

[14] Li, S., Wang, Z., Zhou, G., & Lee, S.Y.M. , “Semi-supervised learning for imbalanced sentiment classification”, In Proceedings of international joint conference on artificial intelligence, pp. 1826–1831, 2012.

[15] Pang, B., & Lee, L., “A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts”, In Proceedings of the association for computational linguistics, pp. 271–278, 2004.

[16] Theresa Wilson, JanyceWiebe, Paul Hoffmann, “Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis”, In Proceedings Of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), Vancouver, pp. 347–354 2004.

[17] Turney, “Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews”, In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, pp. 417-424, 2002.

[18] Vapnik, V., “The Nature of Statistical Learning Theory”, Springer- Verlag, pp. 863-884, 2000.

[19] Yang. Y, X. Liu, “A re-examination of text categorization methods”, In Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), ACM, New York, NY, USA, pp. 42–49,1999.

[20] Yi, J., Nasukawa, T., Niblack,W., &Bunescu, R. , “Sentiment analyzer: extracting sentiments about a given topic using natural language processing techniques”, In Proceedings of the 3rd IEEE international conference on data mining (ICDM 2003), USA, pp. 427– 434, 2003.

[21] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan., “Thumbs up?: sentiment classification using machine learning techniques”, In Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10 (EMNLP '02), Vol. 10. Association for Computational Linguistics, Stroudsburg, PA, USA, 79-86, 2002.

[22]  N. Altrabsheh, M. Cocea, S. Fallahkhair, “Sentiment analysis: towards a tool for analysing real-time students feedback”, 2014 IEEE 26th international conference on tools with artificial intelligence, pp. 419-423, 2014.

[23] C. Troussas, M. Virvou, K. Espinosa, K. Llaguno and J. Caro, “Sentiment analysis of Facebook statuses using Naive Bayes classifier for language learning”, IISA, 2013.

[24] D. Song, H. Lin, and Z. Yang, “Opinion Mining in e-Learning System”, in Network and Parallel Computing Workshops, 2007, pp. 788-792.

[25] Whitelaw, Casey, Navendu Garg, and Shlomo Argamon, “Using appraisal groups for sentiment analysis”, In Proceedings of the 14th ACM international conference on Information and knowledge management, ACM, 2005.

[26] Ms. K. Nirmala Devi, Ms. K. Mouthami, Dr. V. Murali Bhaskaran, “Sentiment Analysis and Classification Based on Textual Reviews”, 2012.

[27] Baker, R.S.; Yacef, K (2009), “The state of educational data mining in 2009: A review and future visions”, JEDM-Journal of Educational Data Mining. 1 (1): 2017.

[28] C. Romero, S. Ventura., “Educational Data Mining: A Review of the State-of-the-Art”, IEEE Transaction on Systems, Man, and Cybernetics, Part C: Applications and Reviews. 40(6), 601-618, 2010.

[29] Castro, F., Vellido, A., Nebot, A. Mugica, F., “Applying Data Mining Techniques to e-Learning Problems”, In: Jain, L.C., Tedman, R. and Tedman, D. (eds.) “Evolution of Teaching and Learning Paradigms in Intelligent Environment”, Studies in Computational Intelligence, 62,Springer-Verlag, 183-221, 2002.

[30] Liu, Bing. "Sentiment analysis and opinion mining." *Synthesis lectures on human language technologies* 5.1 (2012): 1-167.

[31] Jupyter <https://jupyter.org/index.html> The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text.

[32] Anaconda <https://www.anaconda.com/distribution/> The open-source [Anaconda Distribution](https://docs.anaconda.com/anaconda/) is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X.

[33] Natural Language Tool Kit (NLTK) <https://www.nltk.org/> NLTK is a leading platform for building Python programs to work with human language data.

[34] Scikit- learn <https://scikit-learn.org/stable/>