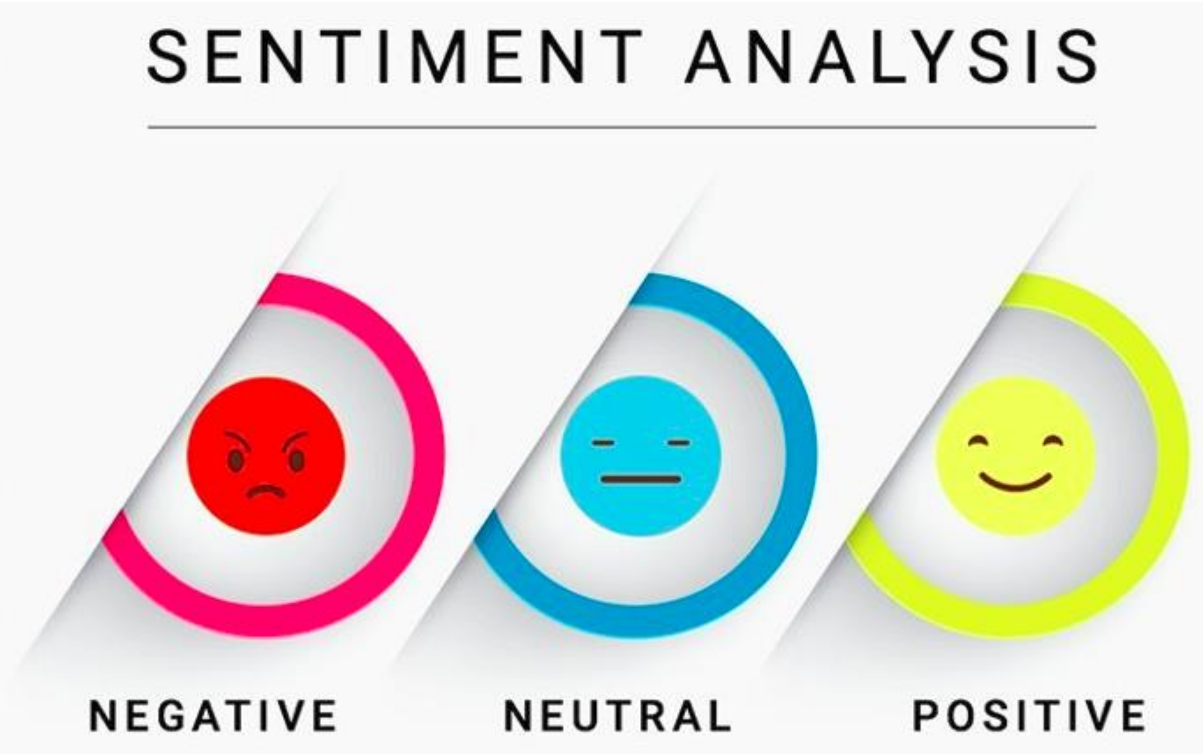
UNVEILING STUDENT SENTIMENTS: A FEEDBACK ANALYSIS STUDY



A PROJECT BY

TUSHAR DEEPAK KSHIRSAGAR

M.Sc. (Statistics)

EXL-Certified Data Analyst

INDEX

ABSTRACT

In the dynamic landscape of education, understanding student sentiment is pivotal for institutions seeking to enhance the learning experience and optimize educational outcomes. This project delves into a comprehensive analysis of student sentiment, employing both descriptive and predictive statistical techniques. By dissecting the intricate interplay of factors that influence student sentiment, this endeavor seeks to empower educational institutions with actionable insights for continuous improvement.

The project commences with data collection, compiling sentiments expressed by students through surveys, feedback forms, and written responses. This rich dataset encapsulates a broad spectrum of student sentiments, ranging from satisfaction with course content to concerns about assessment methods.

Descriptive statistical analysis is deployed to unravel the underlying patterns and trends within the data. Key objectives include categorizing sentiments into positive, negative, or neutral categories, identifying recurring themes and sentiment drivers, and examining sentiment fluctuations over time.

The predictive aspect of the project harnesses the power of machine learning to develop models capable of forecasting student sentiment trends. Using historical sentiment data, these models offer a glimpse into future sentiment trajectories, enabling institutions to proactively address issues and enhance student experiences.

In conclusion, this project seeks to provide educational institutions with a multifaceted understanding of student sentiment, encompassing both descriptive insights and predictive capabilities. By embracing these insights, institutions can tailor their educational strategies, curriculum design, and support services to foster a more positive and enriching learning environment. Ultimately, this project represents a significant step toward elevating the quality of education by placing student sentiment at the forefront of decision-making and continuous improvement efforts.

KEYWORDS: Wordcloud, Students Feedback, Sentiment Analysis, Support Vector Machine.

INTRODUCTION

In the ever-evolving landscape of education, the voice of the student has taken center stage as a catalyst for transformation and improvement. Understanding and responding to student sentiment has become paramount for educational institutions striving to deliver exceptional learning experiences. To this end, we embark on a project that blends the realms of descriptive and predictive statistical analysis to gain comprehensive insights into the sentiments of students.

The project's primary objective is twofold: to describe the current state of student sentiment and, in doing so, lay the groundwork for predicting future trends. By harnessing the power of data, advanced statistical techniques, and machine learning, this endeavor seeks to unravel the intricate web of factors that shape student sentiments, from the classroom to the virtual learning environment.

Modern educational institutions, driven by a commitment to continuous improvement, recognize the value of capturing and interpreting student sentiments. Beyond traditional measures of academic performance, sentiment analysis offers a unique lens through which to view the student experience. It allows us to delve into the emotions, opinions, and perspectives of learners, providing invaluable insights into their engagement, satisfaction, and overall well-being.

This project's dual focus on descriptive and predictive statistical analysis represents a comprehensive effort to unravel the intricate dynamics of student sentiment. The descriptive component seeks to unveil the current landscape of student emotions, categorizing sentiments, identifying recurring themes, and shedding light on the factors that drive these sentiments. Through a retrospective lens, we gain an in-depth understanding of the present state of student sentiment within the educational institution.

However, the project goes beyond mere observation, venturing into the realm of prediction. By harnessing the power of predictive modeling and machine learning, we aspire to peer into the future of student sentiment. This predictive capability allows us to anticipate shifts, trends, and potential challenges in student sentiment, empowering educational institutions to be proactive in addressing issues and cultivating a more positive learning environment.

The journey we undertake in this project is guided by a profound commitment to the enhancement of education. By understanding and acting upon the sentiments of students, we aim to equip educational institutions with the tools and insights needed to make data-driven decisions, refine pedagogical strategies, and ultimately, provide a more fulfilling and enriching educational experience for all learners. Through this project, we endeavor to bridge the gap between student voices and institutional responsiveness, forging a path toward educational excellence and student success.

MOTIVATION

In the realm of education, the pursuit of excellence and continuous improvement is an unceasing endeavor. Today, more than ever, the educational landscape is undergoing profound changes, driven by digital transformation, evolving pedagogical methods, and the imperatives of a rapidly changing world. Amidst these shifts, the voice of the student emerges as a beacon, guiding educational institutions toward a brighter and more responsive future.

The motivation behind our project, “Unveiling Student Sentiments: A Feedback Analysis Study” is rooted in the recognition that the sentiments and experiences of students are not just invaluable feedback; they are the keys to unlocking the full potential of education. By venturing into the intricate realm of student sentiment, we embark on a journey that aligns with the core values of education: empathy, adaptation, and continuous improvement.

Moreover, the digital age has ushered in an era of data abundance. As educational institutions collect vast troves of data, the question arises: How can this data be harnessed to serve students better? The motivation behind our project lies in harnessing the power of data science, advanced statistics, and predictive modeling to transform data into actionable insights. By doing so, we bridge the gap between data and meaningful decision-making, ensuring that the information collected serves a purpose beyond mere data points.

In conclusion, the motivation for our project is grounded in the conviction that education is a dynamic and responsive field that thrives on the voices and sentiments of its primary stakeholders: the students. By embarking on this journey of descriptive and predictive statistical analysis of student sentiment, we aspire to illuminate the path toward a more student-centered, adaptive, and enriching educational landscape. This project is driven by the belief that education, when guided by data-informed decisions and the experiences of its learners, has the power to transform lives and create a brighter future.

OBJECTIVES

- To propose an innovative machine learning model to classify students' sentiments towards examinations into categories such as positive, negative, or neutral based on their comments and feedback.
- To create visualizations, such as sentiment heatmaps or sentiment graphs, to present the findings in an easily understandable format for educational stakeholders
- To investigate ways to integrate sentiment analysis findings into the educational system, such as using student feedback to inform instructional design or exam format improvements.

METHODOLOGY

Descriptive statistics involve the summarization and presentation of data to provide a clear and concise overview of its main characteristics. The primary goals of descriptive statistics are to simplify complex data, identify patterns, and communicate key features effectively. Predictive statistical analysis goes beyond describing data and aims to make informed predictions or forecasts based on historical data patterns. This analysis is particularly valuable when there's a need to anticipate future events, outcomes, or trends.

Commenced the project by gathering data related to student sentiment. This data is collected through various channels such as student surveys, feedback forms, online forums, and written comments. Ensured that the data encompasses a wide range of sentiment expressions. Prepared the collected data for analysis by conducting data preprocessing. This phase involved cleaning the data, handling missing values, and standardizing or encoding categorical variables to ensure data quality and consistency. Performed descriptive sentiment analysis to gain an initial understanding of student sentiment. This analysis includes:

- Categorizing sentiments into positive, negative, or neutral classes.
- Generating summary statistics, including sentiment distribution and frequency.

Employed topic modeling techniques to identify recurring themes or topics within student comments or feedback. This step provides deeper insights into the specific issues or areas driving sentiment. Developed predictive models using machine learning techniques to forecast student sentiment. Steps include:

- Model selection: Experiment with various machine learning algorithms to build predictive models.
- Train and validate models using historical sentiment data.

Evaluated the predictive models using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score. Ensured that the models generalized well to new data to make accurate predictions. Documented the entire methodology and results comprehensively in a project report. Included clear explanations of data preprocessing steps, descriptive and predictive analysis techniques, model details, and actionable recommendations. Utilized visualizations and tables to enhance data interpretation.

By following this methodology, the project aims to provide a holistic view of student sentiment, from descriptive insights to predictive capabilities, and to equip educational institutions with valuable information to enhance the student experience and foster a more responsive and adaptive educational environment.

MATERIALS AND METHODS

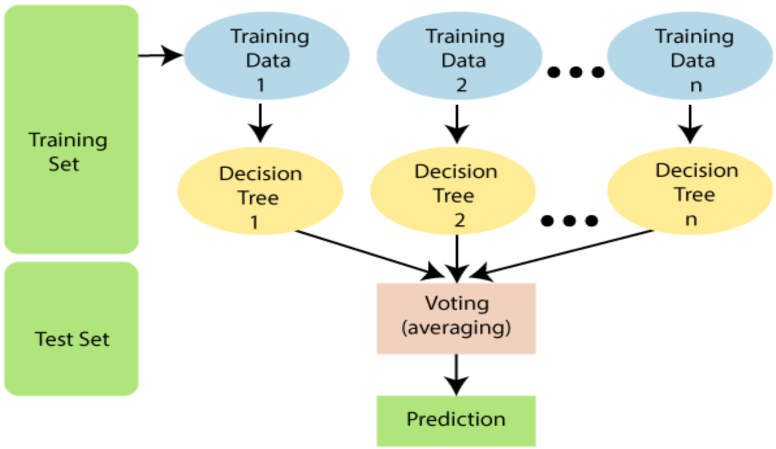
Software, Libraries and Packages Used:

Google Colab	1) Descriptive and Predictive analysis and Exploratory Data Analysis (EDA) is performed using Python language. 2) Basically, Google Colab is an application which provides us with an environment where we can write and execute python code. 3) Importing data in python and performing data pre-processing methods on it, so that data becomes ready for analysis and we get more accurate analysis. 4) After performing pre-processing on the data, we can also implement different machine learning algorithms in order to classify the sentiments. 5) And finally, Google Colab will also make it easier to visualise and analyse the data. 6) Code scripts, research-related text, visualisation plots and graphs, machine learning models, and other materials may all be kept in one Notebook or document and shared effortlessly across several platforms. 7) These are all important uses of the Google Colab and these are the reasons behind choosing Google Colab.
Pandas	The data which was collected from the Hospitals was stored in CSV files or formats. So, in order to work with CSV files and import them in the python we will require a Special library of python which is PANDAS LIBRARY. So, basically when we want to work with the datasets, we will be using Pandas library. It is not just used to import the CSV files but there are many other uses such as: I. This library can clean up unreadable and irrelevant data sets. Relevant data is most important thing for a Data Analyst. II. This library can be used to answer some basic questions about data such as, • If there are two columns, so is there any correlation between these two columns? • What is an average value of a particular column? • What is the minimum or maximum value that is occurring in a dataset?
Sklearn	Sklearn basically stands for Scikit Learn Library. As we are aware of the fact that there are many Machine Learning Libraries used in Python. Scikit learn is one of them, in fact this library is one the best known. This library is responsible to support both of the machine learning approaches, Supervised and Unsupervised. This library also provides

	many different algorithms for the classification, dimensionality reduction, clustering and regression purposes. The combination of two libraries i.e., NumPy and SciPy is Sklearn library. Additionally, it functions nicely with other libraries like Pandas and Seaborn. We can use this library for pre-processing such as feature encoding, feature extraction and we can also use this library to split the data into train and test, then we can use this library to implement different machine learning models and finally in order to check the accuracy of these models we can use this library.
WordCloud	When we have textual data, and you have no idea how to visualise it then "WordCloud" library is the most important library which will be used. The magnitude of each word in a word cloud, shows its frequency or relevance. That is, the word which is mostly repeated will be displayed most bold and large in size. In the image below is a WordCloud. By looking at that image we can understand that words like **"Antibiotic", "inhibitor"** are bolder and larger in size, this is because frequency of these words in the whole text is highest. Python modules such as matplotlib, pandas, and WordCloud are required for creating word clouds.
NLTK	The Natural Language Toolkit, often referred to as NLTK, is a powerful Python library for working with human language data, primarily for **natural language processing (NLP)** and text analysis tasks. It is one of the most widely used libraries for NLP in the Python ecosystem. NLTK provides a wide range of **text processing** utilities and algorithms. NLTK can be used for **sentiment analysis**, which involves determining the sentiment or emotional tone of a piece of text. It includes sentiment analysis datasets and tools for building sentiment classifiers.
Pickle	The pickle module implements binary protocols for serializing and de-serializing a Python object structure. "**Pickling**" is the process whereby a Python object hierarchy is converted into a byte stream and "**unpickling**" is the inverse operation, whereby a byte stream (from a binary file or bytes-like object) is converted back into an object hierarchy. Pickling is alternatively known as "serialization", "**marshalling**," 1 or "flattening"; however, to avoid confusion, the terms used here are "**pickling**" and "**unpickling**".

STATISTICAL TERMS

RANDOM FOREST - A random forest is a Machine Learning technique that is used to solve regression and classification problems. It utilizes ensemble learning technique, that combines many classifiers to provide solutions to complex problems. A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms. The (random forest) algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome.



SUPPORT VECTOR MACHINE-

A Support Vector Machine (SVM) is a powerful supervised machine learning algorithm used for classification and regression tasks. It's particularly effective in separating data points into different classes when the data is not linearly separable in the original feature space. SVM accomplishes this by finding an optimal hyperplane that maximizes the margin between the classes while minimizing classification errors. Here's an explanation of how SVM works,

- **Linear Seperation:**
  In a simplified two-dimensional space (2D), there is a binary classification problem where we have two classes. The goal is to find a hyperplane that separates these two classes with the maximum margin.
- **Non-Linear Seperations:**
  In many real-world scenarios, data is not linearly separable in the original feature space. SVM can handle this by mapping the data into a higher-dimensional space using a kernel function.
- **Multi-Class Classification**:

  SVM can be extended to handle multi-class classification problems using techniques like one-vs-one or one-vs-all. Multiple binary classifiers are trained, one for each class, and combined to make predictions.

- **SVM in Higher Dimensions:**

  SVM's effectiveness increases as the dimensionality of the feature space grows. It excels in problems with many features, such as text classification, image recognition, and gene expression analysis.

NAIVE BAYES-

Naïve Bayes is a popular and simple probabilistic machine learning algorithm used for classification and text analysis tasks. It is based on Bayes' theorem and assumes that the features used for classification are conditionally independent, which is where the "naive" in its name comes from. Despite this simplifying assumption, Naive Bayes often performs remarkably well in practice, especially for tasks like text classification and spam email detection. Here's an overview of how Naive Bayes works:

Bayes' Theorem: At the core of Naive Bayes is Bayes' theorem, which describes the probability of an event, based on prior knowledge of conditions that might be related to the event. In the context of classification:

$P(A|B) = P(B|A).P(A)/ P(B)$

$P(A|B)$: Probability of class A given the observation B.

$P(B|A)$: Probability of observation B given class A.

$P(A)$: Prior probability of class A.

$P(B)$: Probability of observation B.

In the case of Naive Bayes, we are interested in computing $P(A|B)$, which represents the probability of a data point belonging to a particular class given the observed features.

GRADIENT BOOSTING – Gradient boosting is popular boosting algorithm. The main idea behind this is to build models sequentially and these subsequent models try to reduce the errors of the previous model. Gradient boosting is a method standing out for its prediction speed and accuracy, particularly with large and complex datasets. When the target column is continuous, we use Gradient Boosting Regressor whereas when it is a classification problem, we use Gradient Boosting Classifier.

ADA BOOSTING- AdaBoost, also called Adaptive Boosting, is a technique in Machine Learning used as an Ensemble Method. It is called Adaptive Boosting as the weights are reassigned to each instance, with higher weights assigned to incorrectly classified instances. Boosting is used to reduce bias as well as variance for supervised learning. It works on the principal of learners growing sequentially. In simple words, weak learners are converted into strong ones. The AdaBoost algorithm works on the same principle as boosting with a slight difference.

REPRESENTATION OF DATASET

	ID	Polarity	Comment
0	1	Neutral	No issue in it, just little bit of old fashioned
1	2	Neutral	Exam pattern and how it is conducted is really...
2	3	Negative	Not upto the mark.
3	4	Negative	not good improve it
4	5	Negative	very difficults exams
...	...	...	...
215	216	Positive	exam pattern is good and marks distribution is...
216	217	Neutral	all are good
217	218	Positive	The examination pattern is good .But time is n...
218	219	Neutral	MCQ pattern is quite good and efficient way fo...
219	220	Neutral	PAPER CHECKING IS VERY HARD REMAINING IS GOOD

Understanding of Data

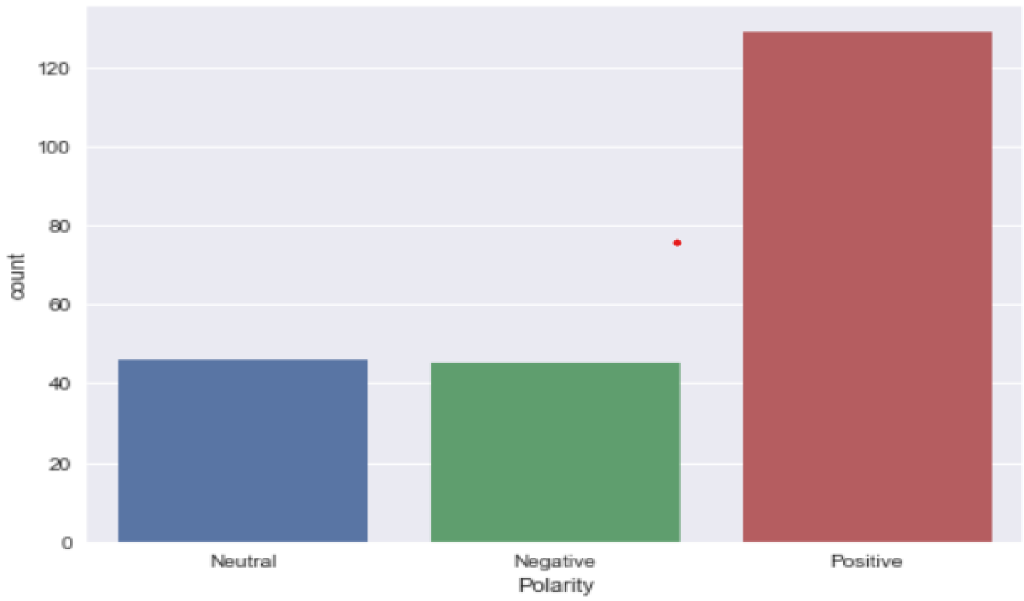
The information of the dataset gives us the basic idea about the data, that is number of columns in the data, name of the columns, non-null count of the data along with data type of each and every column in the data. By which we can make our roadmap of analysis and decide which technique we can use to get required results.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 220 entries, 0 to 219
Data columns (total 3 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Polarity        220 non-null    object
 1   Comment         220 non-null    object
 2   Polarity_label  220 non-null    int64
dtypes: int64(1), object(2)
memory usage: 5.3+ KB
```

EXPLORATORY DATA ANALYSIS

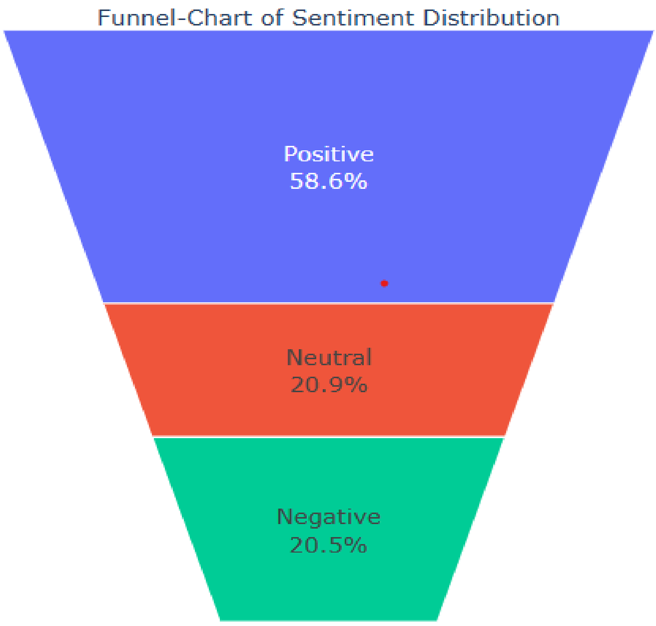
1) Descriptive Statistical Model:

COUNT PLOT OF TARGET VARIABLE:



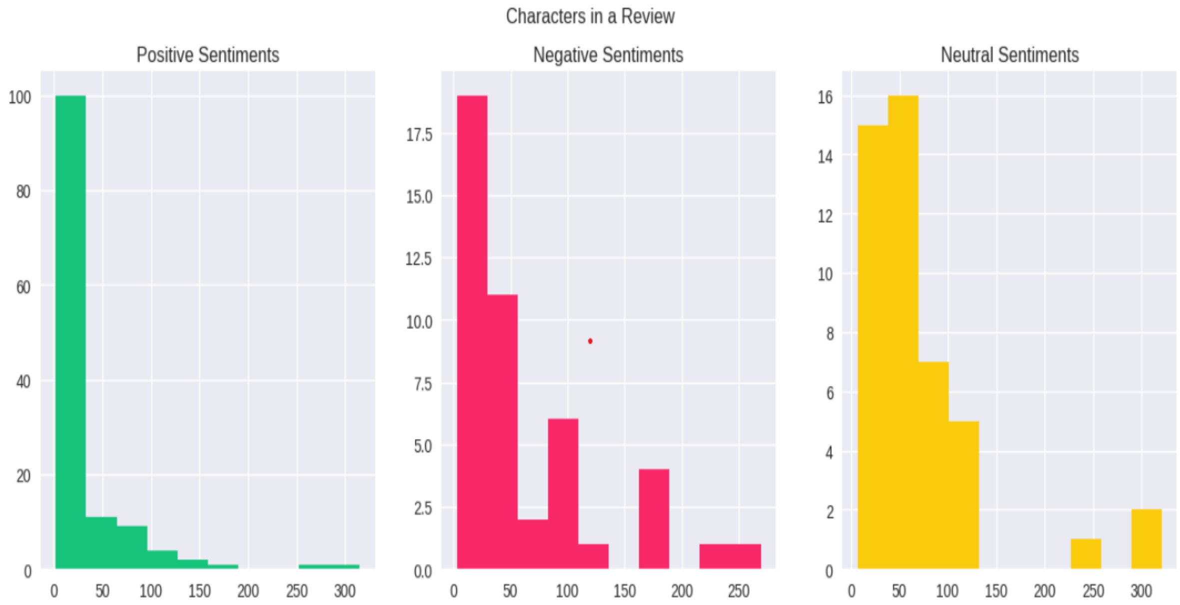
Conclusions: The above Count plot shows that target labels are highly imbalanced.

FUNNEL-CHART OF SENTIMENT DISTRIBUTION:



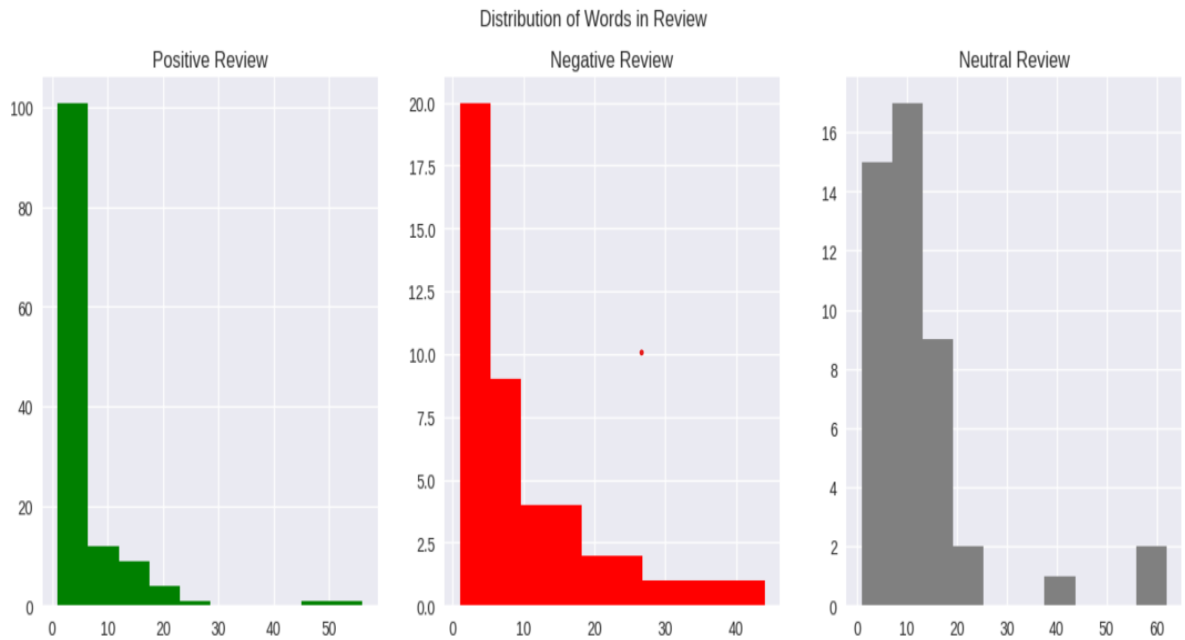
Conclusions: There is an uneven distribution of sentiments in the data with the largest portion belongs to Positive (58.6%) followed by Negative (20.5%) and Neutral(20.9%).

HISTOGRAM OF CHARACTERS IN REVIEW:



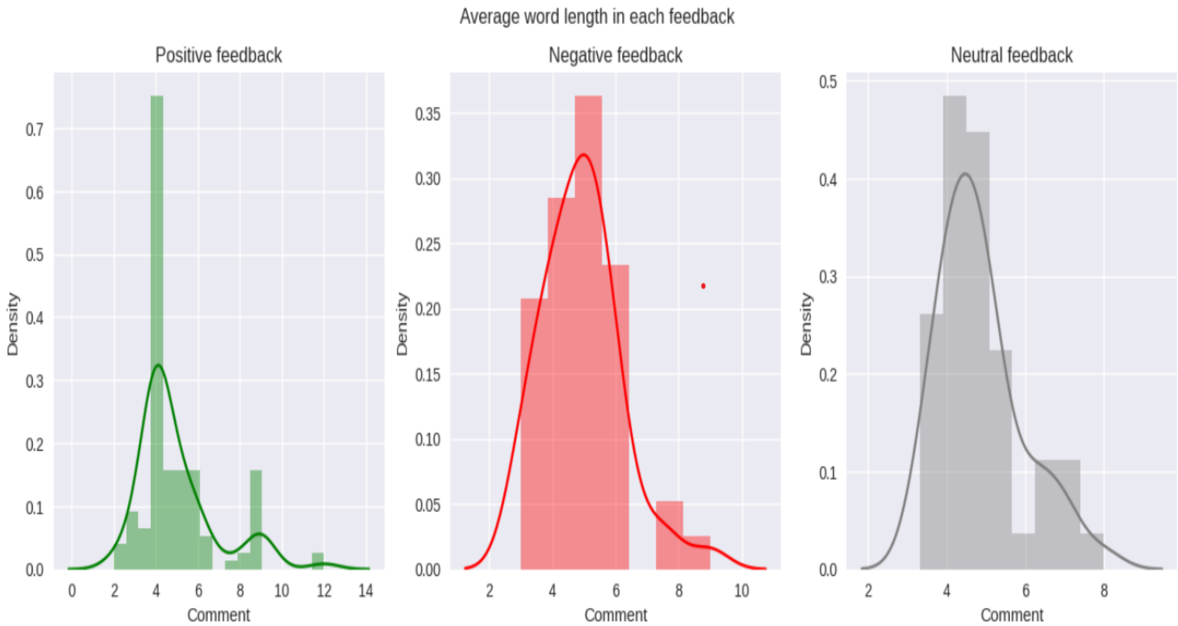
Conclusions: From above Histogram we can see that, the frequency of number of characters in all type of reviews Positive, Negative and Neutral.

HISTOGRAM OF DISTRIBUTION OF WORDS IN REVIEW:



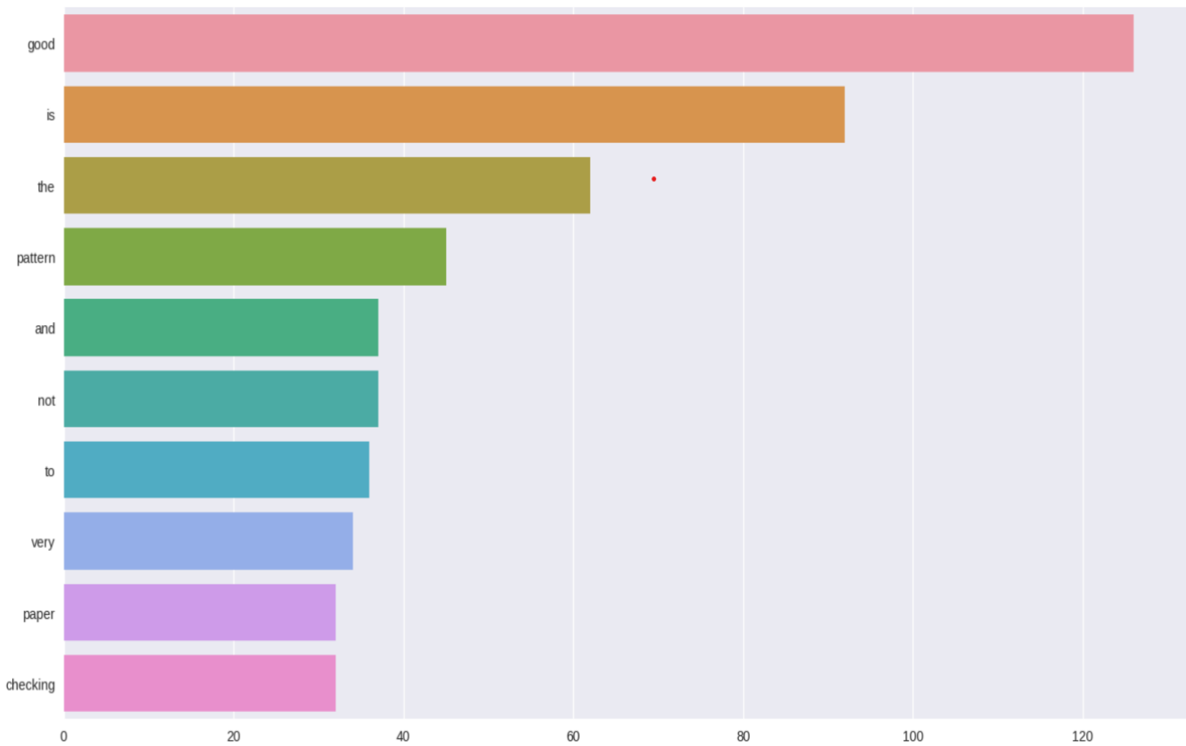
Conclusions: From this Histogram we can see that, the distribution of words in all type of reviews Positive, Negative and Neutral.

DISTRIBUTION PLOT OF AVERAGE WORD LENGTH IN EACH FEEDBACK:



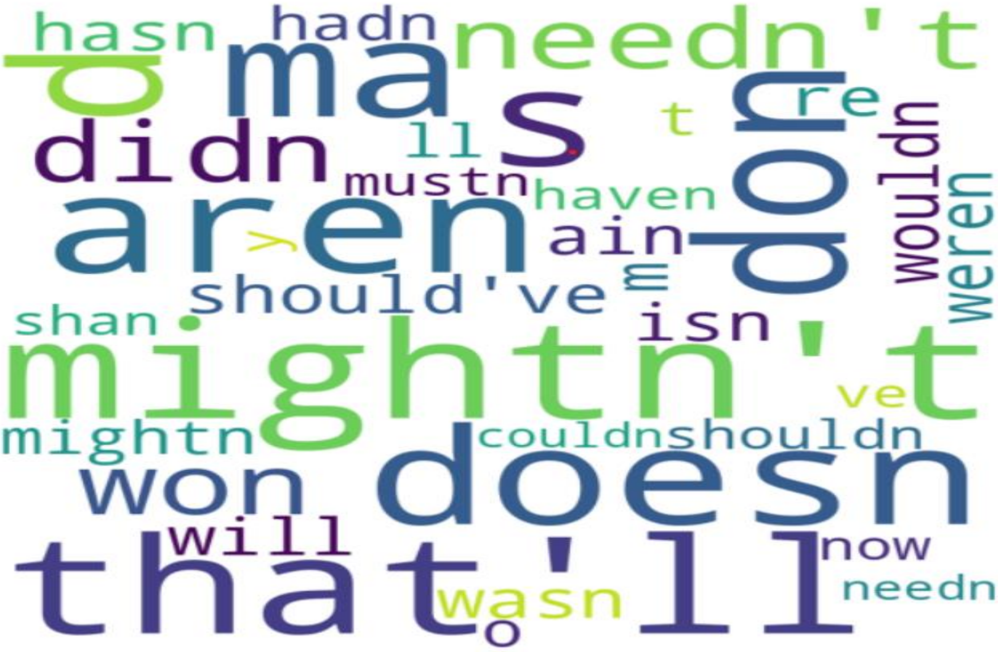
Conclusions: From above Distribution Plot we can conclude that, average word length of Positive, Negative and Neutral feedback is 4, 5, 5 respectively.

BAR PLOT OF MOST COMMON UNIT:



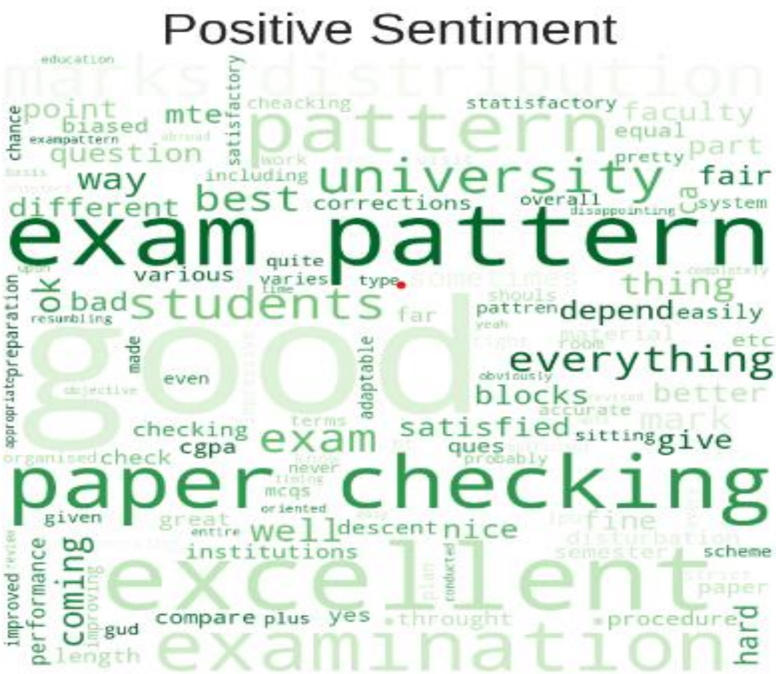
Conclusions: From above Bar Plot we can observe that, the words "good", "is", "the" and "pattern" are most common units having frequency more than rest of the words.

WORDCLOUD OF COMMON STOPWORDS IN TEXT:



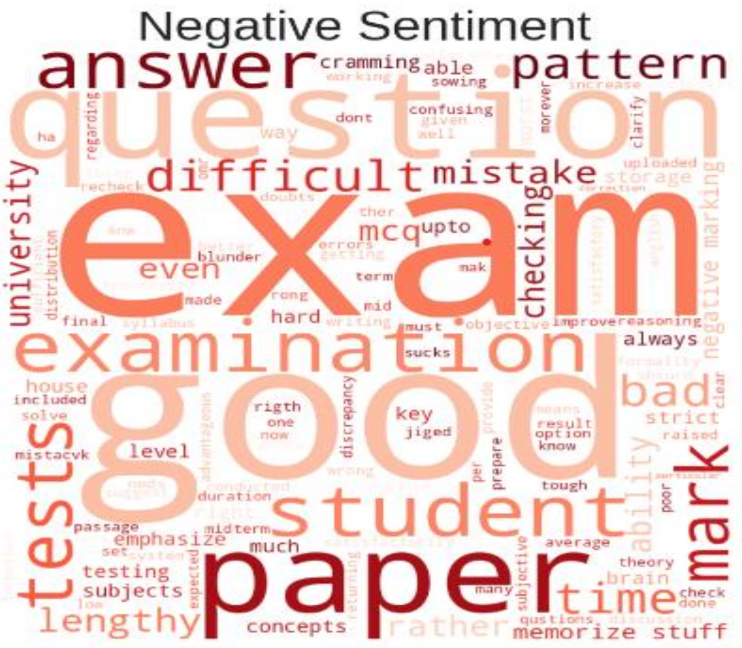
Conclusions: From the above Word Cloud we can observe that, most commonly appeared stopwords in the text are "mightn't", "doesn't", "aren't", "don't" etc.

WORDCLOUD OF POSITIVE SENTIMENT:



Conclusions: From the above Word Cloud we can observe that, most of the positive sentiments contains words "good", "exam pattern", "excellent", "paper checking", "examination" etc.

WORDCLOUD OF NEGATIVE SENTIMENT:



Conclusions: From the above Word Cloud we can observe that, most of the negative sentiments contains words "exam", "question paper", "difficult", "lengthy", "examination" etc.

WORDCLOUD OF NEUTRAL SENTIMENT:

Conclusions: From the above Word Cloud we can observe that, most of the neutral sentiments contains words "good", "pattern", "paper checking", "exam", "improved" etc.

DATA PREPROCESSING

Original Data:

	Polarity	Comment	Polarity_label
0	Neutral	No issue in it, just little bit of old fashioned	0
1	Neutral	Exam pattern and how it is conducted is really...	0
2	Negative	Not upto the mark.	-1
3	Negative	not good improve it	-1
4	Negative	very difficults exams	-1
5	Neutral	Fine but still can be improved	0
6	Negative	Worst examination pattern	-1
7	Negative	Again the university tests students of their a...	-1
8	Negative	The examination pattern is good but the class ...	-1
9	Neutral	Exams came too quickly, 20 days between minors...	0

Features Selected for Model Building:

	Polarity_label	Comment
0	0	No issue in it, just little bit of old fashioned
1	0	Exam pattern and how it is conducted is really...
2	-1	Not upto the mark.
3	-1	not good improve it
4	-1	very difficults exams
5	0	Fine but still can be improved
6	-1	Worst examination pattern
7	-1	Again the university tests students of their a...
8	-1	The examination pattern is good but the class ...
9	0	Exams came too quickly, 20 days between minors...

Preprocesssing:

1. Removal of URLs and HTML Links

```
0      No issue in it, just little bit of old fashioned
1     Exam pattern and how it is conducted is really...
2                                   Not upto the mark.
3                                  not good improve it
4                                 very difficults exams
5                          Fine but still can be improved
6                              Worst examination pattern
7     Again the university tests students of their a...
8     The examination pattern is good but the class ...
9     Exams came too quickly, 20 days between minors...
```

Conclusion: In order to preprocess the data removal of URLs and HTML Links from the text of column comment is done.

2. Lower Casing

```
0      no issue in it, just little bit of old fashioned
1     exam pattern and how it is conducted is really...
2                                   not upto the mark.
3                                  not good improve it
4                                 very difficults exams
5                          fine but still can be improved
6                              worst examination pattern
7     again the university tests students of their a...
8     the examination pattern is good but the class ...
9     exams came too quickly, 20 days between minors...
```

Conclusion: In this section of preprocessing, lower casing of text of column comment is done.

3. Removal of Numbers

```
0      no issue in it, just little bit of old fashioned
1     exam pattern and how it is conducted is really...
2                                   not upto the mark.
3                                  not good improve it
4                                 very difficults exams
5                          fine but still can be improved
6                              worst examination pattern
7     again the university tests students of their a...
8     the examination pattern is good but the class ...
9     exams came too quickly,  days between minors a...
```

Conclusion: In this section of preprocessing, removal of numbers from the text of column comment is done.

4. Removal of Stopwords and Punctuations

```
0                          issue little bit old fashioned
1    exam pattern conducted really seems good first...
2                                           upto mark
3                                        good improve
4                                    difficults exams
5                                 fine still improved
6                            worst examination pattern
7    university tests students ability memorize stu...
8    examination pattern good class testsmidterm te...
9    exams came quickly days minors minor major tim...
```

Conclusion: In this section of preprocessing, removal of stopwords and punctuations from the text of column comment is done.

5. Removal of Extra White Space

```
0                          issue little bit old fashioned
1    exam pattern conducted really seems good first...
2                                           upto mark
3                                        good improve
4                                    difficults exams
5                                 fine still improved
6                            worst examination pattern
7    university tests students ability memorize stu...
8    examination pattern good class testsmidterm te...
9    exams came quickly days minors minor major tim...
```

Conclusion: In this section of preprocessing, removal of extra white spaces from the text of column comment is done.

6. Lemmatization

```
0                          issue little bit old fashioned
1    exam pattern conducted really seems good first...
2                                           upto mark
3                                        good improve
4                                     difficults exam
5                                 fine still improved
6                            worst examination pattern
7    university test student ability memorize stuff...
8    examination pattern good class testsmidterm te...
9    exam came quickly day minor minor major time c...
```

Conclusion: In this section of preprocessing, lemmatization of text of column comment is done.

In this way, the data is preprocessed. Now, the data is ready for model building. This data is used for predictive modelling.

2) PREDICTIVE STATISTICAL MODEL

Splitting Train & Test Set:

Taking randomly 80% data set as train dataset and 20% dataset as validation dataset. Again, from validation data taking randomly 50% dataset as test dataset and remaining 50% dataset as validation dataset.

Model Building:

Various Machine Learning models are built on training dataset with TF-IDF Technique. Test dataset is used further for evaluation of model performance.

Model Selection:

- **Naïve Bayes Classifier -**

  Output: Accuracy of Naïve Bayes Classifier is **0.68.**

- **Support Vector Classifier -**

  Output: Accuracy of Support Vector Classifier is **0.73.**

- **Random Forest Classifier -**

  Output: Accuracy of Random Forest Classifier is **0.55.**

- **Ada boost -**

  Output: Accuracy of Ada Boost Classifier is **0.64.**

- **Gradient Boosting -**

  Output: Accuracy of Gradient Boosting Classifier is **0.55.**

Conclusion:

Support Vector Classifier has greater accuracy than the other models. So, Support Vector Classifier does best amongst the models to be the most accurate.
Hence, we fit Support Vector Classifier for predictions.

SUPPORT VECTOR CLASSIFIER

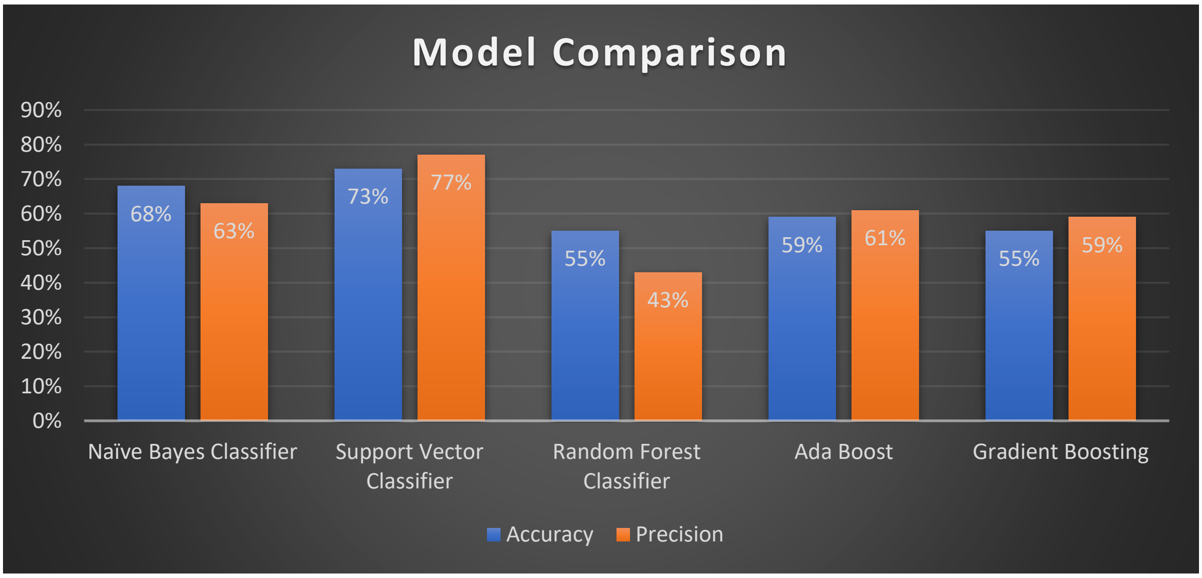
Model Summary:

```
********* Support Vector Classifier *********
    Accuracy     :  0.7272727272727273
    Precision    :  0.7651515151515151
    Recall       :  0.7272727272727273
    F1 Score     :  0.7384615384615384
```

Conclusion: From above summary, we can conclude that the accuracy of fitted model on test set is 73%.

MODEL COMPARISON

Model	Accuracy	Precision	Recall	F1 score
Naïve Bayes Classifier	68%	63%	68%	60%
Support Vector Classifier	73%	77%	73%	74%
Random Forest Classifier	55%	43%	55%	48%
Ada Boost	59%	61%	59%	60%
Gradient Boosting	55%	59%	55%	56%



Interpretation: Above graph also shows that Support Vector Classifier performs best among all the other models, with greater accuracy and precision.

CONCLUSION

- From the Funnel chart of Sentiment Distribution, there are 58.6% positive sentiments, 20.5% negative sentiments and 20.9% neutral sentiments.

- From the Distribution Plot of Average Word Length, average word length of Positive, Negative and Neutral feedback is 4, 5, 5 respectively.

- From the Bar Plot of Most Common Units, the words “good”, “is”, “the” and “pattern” are most common units in the review having frequency more than rest of the words.

- From the Word Clouds of Positive, Negative and Neutral Sentiments, most of the positive sentiments contains words “good”, “exam pattern”, “excellent”, “paper checking”, “examination” etc., most of the negative sentiments contains words “exam”, “question paper”, “difficult”, “lengthy”, “examination” etc., most of the neutral sentiments contains words “good”, “pattern”, “paper checking”, “exam”, “improved” etc.

- Support Vactor Classifier has greater accuracy and precision than the other models and it does best amongst the models to be the most accurate.

- The accuracy and precision of Support Vector Classifier on test set is 73% and 77% respectively.

DISCUSSION

The project “Unveiling Student Sentiments: A Feedback Analysis Study” has provided valuable insights into the complex realm of student sentiment within the educational context. Through a comprehensive exploration of descriptive and predictive statistical techniques, this study has shed light on various aspects of student sentiment, paving the way for data-driven decision-making in educational institutions. The descriptive statistical analysis component of this project played a crucial role in understanding the current state of student sentiment. By categorizing sentiments into positive, negative, and neutral classes, we gained a holistic view of the emotional responses of students towards different aspects of their educational journey. The predictive aspect of this project, facilitated by machine learning techniques, empowered us to forecast student sentiment trends. By utilizing historical sentiment data, we can now anticipate changes in student sentiment, enabling educational institutions to proactively address issues and optimize the student experience.

In conclusion, the project on “Unveiling Student Sentiments: A Feedback Analysis Study” has provided a comprehensive understanding of student sentiment within an educational context. By combining descriptive insights with predictive capabilities, this study equips educational institutions with valuable information to improve the student experience, foster a responsive and adaptive environment, and empower students to actively participate in shaping their educational journey. The project represents a significant step toward data-informed decision-making in education and underscores the pivotal role of student sentiment in enhancing learning outcomes and student satisfaction.

SUGGESTIONS

On the basis of this study, we suggest educational institutions to identify areas of concern or dissatisfaction within specific courses or programs. This study can inform curriculum enhancements, adjustments in course content, and improvements in teaching methodologies to better meet students' needs and expectations.

By this study, we suggest educational institutions to use sentiment analysis as part of quality assurance processes. It can help assess the overall quality of education and student services, allowing institutions to maintain high standards and address areas that require improvement.

From this study, we suggest educational policy makers to make informed choices by understanding student sentiments about educational policies and reforms.

SCOPE & LIMITATIONS

SCOPE

- In this study, we have proposed an innovative machine learning model which can be very helpful for all educational institutions, so they can assess the overall quality of education and student services, allowing institutions to maintain high standards and address areas that require improvement.
- This study can also be very helpful for educational policy makers. From this study, educational policy makers can make their policies by understanding student sentiments about educational policies and reforms.

LIMITATIONS

- In this study we have collected the data from students of a particular university but which is not enough for being this study as global solution for educational policy making.
- This study is based on handful of courses and data is collected from a university belonging to specific region.

REFERNCES

- Sentiment Analysis in higher education: a systematic mapping review, International Conference on Applied and Practical Sciences (R Baragash, H Aldowah)

- Sentiment Analysis of Students' Feedback with NLP and Deep Learning: A Systematic Mapping Study, International Journal of Applied Science (Zenun Kastrati et., al.)

- Data Mining by Mathew A. Russell

APPENDIX

#Code

Basic statistical Discription

```
df.info()
df.describe()
```

Checking null values

```
df.isnull().sum()
```

Aspect mapping

```
df = df[['Polarity','Comment']]
df.Polarity.unique()
aspect_mapping = {'Positive': 1 ,'Negative': 2 , 'Neutral': 0}
df['Polarity_label'] = df['Polarity'].map(aspect_mapping)
```

#Exploratory Data Analysis

Count Plot

```
ax = sns.countplot(df.Polarity)
```

Word Cloud

```
comment_words = ''
stopwords = set(STOPWORDS)


for val in stop:

    # typecaste each val to string
    val = str(val)

    # split the value
    tokens = val.split()

    # Converts each token into lowercase
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    comment_words += " ".join(tokens)+" "


wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = stopwords,
                min_font_size = 10).generate(comment_words)

# plot the WordCloud image
plt.figure(figsize = (8, 5), facecolor = "white")
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

Distribution Plot

```
fig,(ax1,ax2,ax3)=plt.subplots(1,3,figsize=(15,5))
word=df[df['Polarity_label']==1]['Comment'].str.split().apply(lambda x : [len(i) for i in x])
sns.distplot(word.map(lambda x: np.mean(x)),ax=ax1,color='green')
ax1.set_title('Positive feedback')
word=df[df['Polarity_label']==-1]['Comment'].str.split().apply(lambda x : [len(i) for i in x])
sns.distplot(word.map(lambda x: np.mean(x)),ax=ax2,color='red')
ax2.set_title('Negative feedback')
word=df[df['Polarity_label']==0]['Comment'].str.split().apply(lambda x : [len(i) for i in x])
sns.distplot(word.map(lambda x: np.mean(x)),ax=ax3,color='grey')
ax3.set_title('Neutral feedback')
fig.suptitle('Average word length in each feedback')
```

#Model Building

Splitting the Data

```
from sklearn.model_selection import train_test_split, GridSearchCV
training_sentences, validation_sentences, training_labels, validation_labels = train_test_split(reviews, labels, test_size=.2,random_state = 23)
validation_sentences, test_sentences, validation_labels, test_labels = train_test_split(validation_sentences, validation_labels, test_size=.5,random_state = 23)
```

Naïve Bayes Classifier

```
# Model 1 - default parameter
from sklearn.metrics import classification_report
from sklearn.naive_bayes import MultinomialNB

from sklearn.pipeline import Pipeline

nb_classifier1 = MultinomialNB()

# define the stages of the pipeline
pipeline = Pipeline(steps= [('tfidf', tfidf),
                            ('model',nb_classifier1) ])

pipeline.fit(training_sentences,training_labels)


pred1 = pipeline.predict(test_sentences)

print(classification_report(test_labels,pred1, target_names = ['Positive','Negative','Neutral']))
```

```
[ ] precision = precision_score(test_labels, pred1,average='weighted')
    print('Precision: %f' % precision)
    recall = recall_score(test_labels, pred1,average='weighted')
    print('Recall: %f' % recall)
    f1 = f1_score(test_labels, pred1,average='weighted')
    print('F1 Score: %f' % f1)
```

Support Vector Classifier

```
# Model 2
from sklearn.svm import SVC

svc_model1 = SVC(C=1, kernel='linear', gamma= 1)

# define the stages of the pipeline
pipeline = Pipeline(steps= [('tfidf', tfidf),
                            ('model',svc_model1) ])

pipeline.fit(training_sentences,training_labels)

pred1 = pipeline.predict(test_sentences)

print(classification_report(test_labels, pred1, target_names = ['Positive','Negative','Neutral']))
```

```
acccuracy = accuracy_score(test_labels, pred1)
precision = precision_score(test_labels, pred1,average='weighted')
recall = recall_score(test_labels, pred1,average='weighted')
f1_score = f1_score(test_labels, pred1,average='weighted')

print("********* Support Vector Classifier *********")
print("\tAccuracy    : ", acccuracy)
print("\tPrecision   : ", precision)
print("\tRecall      : ", recall)
print("\tF1 Score    : ", f1_score)
```

#Table of all models

```
# table of all models

Pipelines = [Pipeline(steps= [('tfidf', tfidf),('model',nb_classifier1) ]), Pipeline(steps= [('tfidf', tfidf),('model',svc_model1) ]),
             Pipeline(steps= [('tfidf', tfidf),('model',rf) ]), Pipeline(steps= [('tfidf', tfidf),('model',abc) ]),
             Pipeline(steps= [('tfidf', tfidf),('model',gbc) ])]

model_names = ['Naive Bayes', 'SVC', 'Random Forest', 'Ada Boost', 'Gradient Boosting']

accuracy = []
precision = []

for i in Pipelines:
    i.fit(training_sentences,training_labels)
    y_pred = i.predict(test_sentences)
    accuracy.append(accuracy_score(test_labels, y_pred))
    precision.append(precision_score(test_labels, y_pred,average='weighted'))

model_comparison = pd.DataFrame({'Model': model_names, 'Accuracy': accuracy, 'Precision': precision})
model_comparison.sort_values(by='Accuracy', ascending=False)
```