**Mini Project Report on**



**Analyzing Sentiments in Students Reviews of Online Courses**



**Submitted in partial fulfilment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Student Name**  **University Roll No.**

**Dhruv Gupta 2021188**

***Under the Mentorship of***

**Dr. Amit Kumar**

**Assistant Professor**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**July-2024**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Analyzing Sentiments in Students Reviews of Online Courses”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of

**Dr. Amit Kumar, Assistant Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Name University Roll no

Dhruv Gupta 2021188

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | **1-2** |
| Chapter 2 | Literature Survey | **3-5** |
| Chapter 3 | Methodology | **6-9** |
| Chapter 4 | Result and Discussion | **10-14** |
| Chapter 5 | Conclusion and Future Work | **15-17** |
|  | References | **18** |

**Chapter 1**

**Introduction**

* 1. **Background**

In the contemporary digital landscape, the extraction of meaningful insights from textual data assumes a crucial role across various applications, be it in processing images, documents, or multimedia content. The identification and extraction of text serve as fundamental elements in the realms of data analysis and automation. This mini-project is dedicated to the implementation of Analyzing Sentiments in Students Reviews of Online Courses, aiming to create a versatile tool proficient in extracting valuable information from textual data for practical applications.

The achievement of this goal involved training the model on a preprocessed dataset utilizing diverse classification algorithms, such as Random Forest and Support Vector Machine (SVM), which categorize comments in alignment with the provided data. To ensure precise outcomes, data preprocessing is essential, involving the elimination of stop words and word tokenization.

* 1. **Context and Significance**

The context of this project lies within the realm of e-learning and the increasing reliance on online education platforms. As the demand for online courses continues to rise, understanding student feedback becomes vital for maintaining high educational standards and ensuring a positive learning experience.

The significance of this project is multi-faceted. By analyzing student sentiments, educators can identify areas of improvement in their courses. This can lead to enhanced course content, better instructional design, and more effective teaching strategies. Understanding student feedback helps in addressing their concerns and needs, thereby improving their overall satisfaction and retention rates. For e-learning platforms, analyzing sentiments can provide a competitive edge by highlighting the strengths and weaknesses of their offerings compared to competitors. Insights gained from sentiment analysis can be used to tailor learning experiences to individual student needs, promoting a more personalized and effective learning journey. The resulting text offers a correct interpretation of the inputted text according to the model. Additionally, a classification report provides further understanding of the model's performance, such as precision, recall, F1- score.

* 1. **Relevance of the Topic**

This project aims to contribute to the field of educational data mining and learning analytics by providing a comprehensive evaluation of sentiment analysis techniques applied to student reviews. By comparing the performance of SVM and Random Forest classifiers, the project seeks to identify the most effective method for sentiment classification, offering valuable insights for educators, researchers, and e-learning platforms.

**Chapter 2**

**Literature Survey**

**2.1 Supervised Learning**

Supervised learning, a subset of machine learning and artificial intelligence, utilizes labeled datasets to train algorithms for accurate data classification or outcome prediction. The model adjusts its weights as input data is fed in, achieving fitting through the cross-validation process. In supervised learning, a training set is employed to instruct models in producing the desired output. This set encompasses inputs and corresponding correct outputs, facilitating the model's gradual learning. The algorithm assesses its accuracy using a loss function, iteratively adjusting until the error is minimized. While supervised learning yields benefits like profound data insights and enhanced automation, it poses challenges in terms of requisite expertise for accurate model structuring, the time-intensive nature of model training, and the potential for human errors in datasets, leading to mislearning. Unlike unsupervised learning models, supervised learning lacks the inherent ability to independently cluster or classify data.

**2.2 Classification Algorithms:**

**Linear Support Vector Classifier** [1]

Support Vector Machines (SVM) is a class of supervised machine learning algorithms that can be used for classification and regression tasks. Specifically, a Linear Support Vector Classifier (Linear SVC) is a variant of SVM designed for binary classification problems.

It aims to find the hyperplane in the feature space that best separates instances of different classes. The term "linear" indicates that the decision boundary is a linear combination of the input features.

Hyperplane: The decision boundary that separates instances of different classes. In a binary classification problem, the hyperplane is defined as a linear combination of the input features:

wT ⋅x +b =0 , where w is the weight vector, x is the input feature vector, and b is the bias term.

Margin:

The margin is the distance between the hyperplane and the nearest instances (support vectors) from each class.

A larger margin implies a more robust classifier.

Support Vectors:

Support vectors are the instances that lie closest to the hyperplane and play a crucial role in determining the decision boundary.

Only the support vectors contribute to the determination of the hyperplane.

Application:

Linear SVC is widely used in various domains, including text classification, image recognition, and bioinformatics. Its effectiveness lies in its ability to handle high-dimensional data and find a hyperplane that separates classes in a linear manner.

Advantages:

Effective in high-dimensional spaces.

Suitable for problems with a clear margin of separation.

Limitations:

May not perform well on datasets with complex, non-linear relationships.

Sensitive to outliers.

Assumptions:

Linear SVC assumes that the data is linearly separable, meaning there exists a hyperplane that can perfectly separate instances from different classes. If perfect separation is not possible, the algorithm aims to find the hyperplane with the maximum margin while allowing for some misclassifications.

**Random Forests** [2]

Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges a.s. to a limit as the number of trees in the forest becomes large. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them. Using a random selection of features to split each node yields error rates that compare favorably to Adaboost (Freund and Schapire[1996]), but are more robust with respect to noise. Internal estimates monitor error, strength, and correlation and these are used to show the response to increasing the number of features used in the splitting. Internal estimates are also used to measure variable importance. These ideas are also applicable to regression.

**Chapter 3**

**Methodology**

**3.1 Data Collection**

**Dataset:**

Here we have implemented a datasets named ‘reviews’ having 100038 entries. Out of which 60% are positive, 20% are neutral and 20% negative comments having score -1, 1, 2 for the negative, neutral and positive comments respectively.

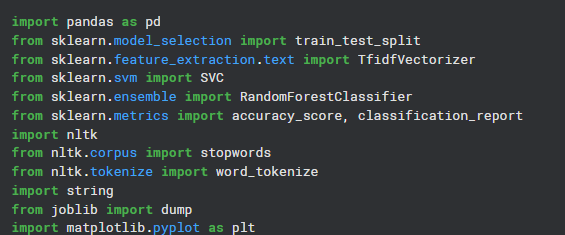
We have preprocessed the dataset to improve the accuracy of our determination and also to train our model on a variety of data

**3.2 Implementation Details**

**3.2.1 Programming Language and Libraries**

The execution was conducted using the Python programming language. The implementation involves the utilization of several Python libraries, including NumPy as np and Pandas as pd for data manipulation, Streamlit as st for creating interactive web applications, Matplotlib.pyplot as plt for data visualization, and Plotly Express as px for creating interactive plots. Additionally, scikit-learn's train\_test\_split is employed for splitting the dataset into training and testing sets. The creation of a machine learning pipeline is facilitated through scikit-learn's Pipeline class. For text feature extraction, the TfidfVectorizer from scikit-learn is utilized. The classification models incorporated in the implementation include Multinomial Naive Bayes (MultinomialNB), Complement Naive Bayes (ComplementNB), and Linear Support Vector Classifier (LinearSVC) from scikit-learn's machine learning module. Evaluation of the models is performed using metrics such as accuracy\_score and classification\_report, both available in scikit-learn's metrics module. Overall, these libraries collectively enable the development, analysis, and assessment of a machine learning model for the given application.

Using following commands



**Figure 3.1 Loading Libraries**

**3.2.2 Text Classification Procedure**

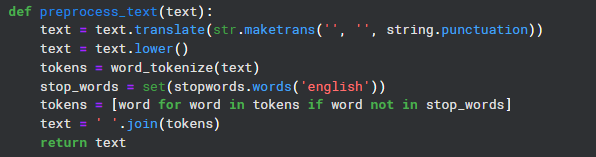
**Dataset loading: It loads the dataset for classification**

We will use pandas to read the dataset

****

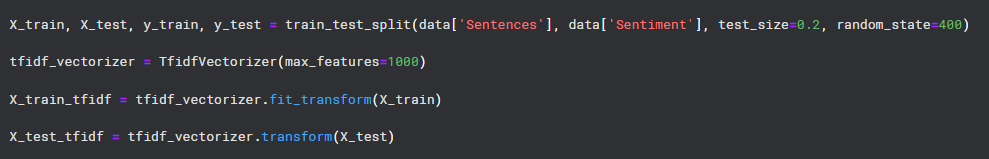
**Figure 3.2 Loading Dataset**

**Preprocessing the dataset**



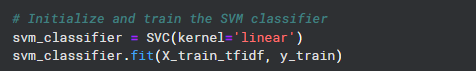
**Figure 3.3 Preprocessing of Dataset**

**Split the data into two parts for testing and training purposes**

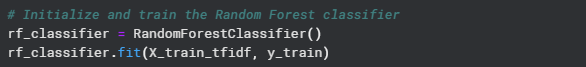


**Figure 3.4 Splitting of Dataset**

**Training the Model using all the algorithms**

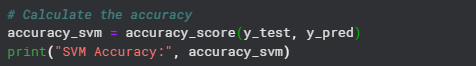
****

**Figure 3.5(a) Training SVM Classifier**

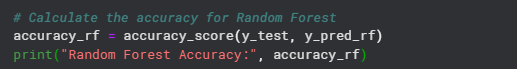
****

**Figure 3.5(b) Training Random Forest Classifier**

**Finding the Accuracies of each algorithm to determine the best one for our use case**



**Figure 3.6(a) SVM accuracy**



**Figure 3.6(b) Random Forest accuracy**

The final accuracies of each algorithm are as following:

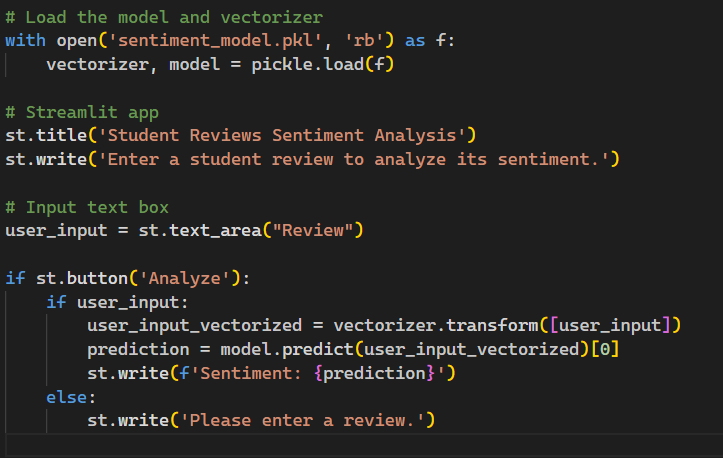
SVM: 0.9149

RF: 0.9078

Thus making SVM the most accurate for our use case and the one we’ll use for predicting whether the text is spam or ham.

**3.2.3 GUI Integration**

The implementation seamlessly integrates the sentiment analysis model using SVM with a user-friendly GUI using Streamlit . The GUI allows users to enter any text message and displays the result in a text box it also shows the accuracy and classification report of the model.

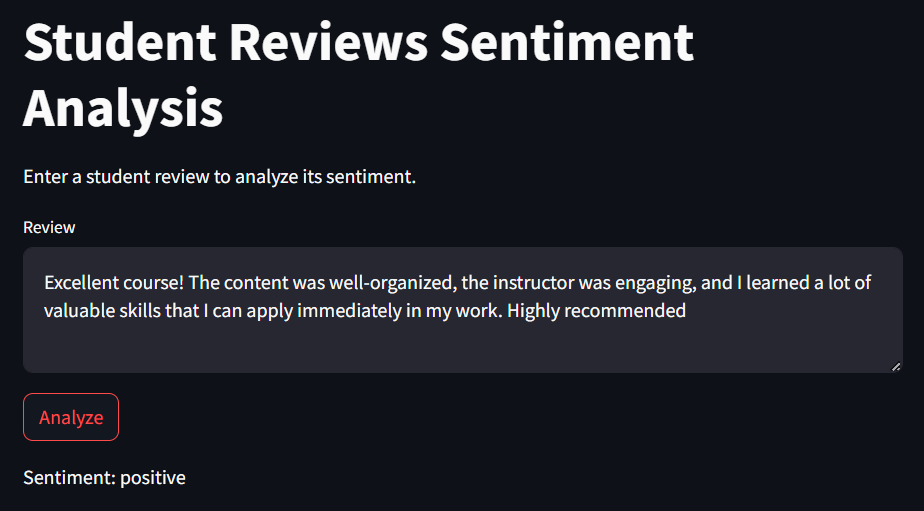
****

**Figure 3.7 GUI Integration using Streamlit**

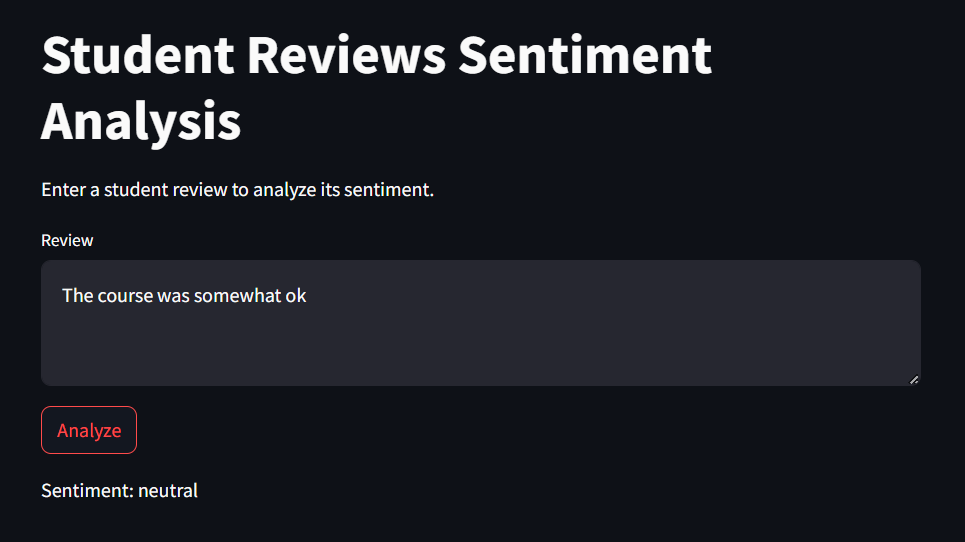
**Chapter 4**

**Result and Discussion**

**4.1 Sample Text and Output**

****

**Figure 4.1 Positive Review**

****

**Figure 4.2 Neutral Review**

****

**Figure 4.3 Negative Review**

The model successfully determines whether the entered review is positive, neutral or negative based on the SVM Algorithm.

**4.3 Graphs**

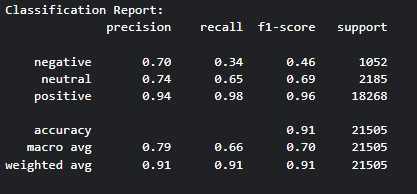
**4.3.1 Comparison Between different Classifiers/Algorithms**

****

**Figure 4.4 Comparison of Classifiers**

As we can see from the Graph the Accuracy of the Linear SVM is the most and will give the best output when used for text classification purposes

**4.3.2 Classification Report**



**Figure 4.5 Classification Report of the model**

Recall (Sensitivity or True Positive Rate): [3]

Recall is the proportion of accurately predicted positive observations in relation to the total number of actual positive instances. It emphasizes the ability of the model to capture all the relevant instances of a positive class.

Mathematically, Recall is calculated as:

Recall= (True Positives+False Negatives)/True Positives

Accuracy: [3]

Accuracy is a measure of the overall correctness of the model and represents the ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances.

Mathematically, Accuracy is calculated as:

​Accuracy= (True Positives+True Negatives)/Total Instances

F1 Score: [3]

The F1 score is the balanced average of precision and recall, calculated using the harmonic mean. It provides a balance between precision and recall, especially in situations where there is an imbalance between the number of positive and negative instances.

Mathematically, F1 Score is calculated as:

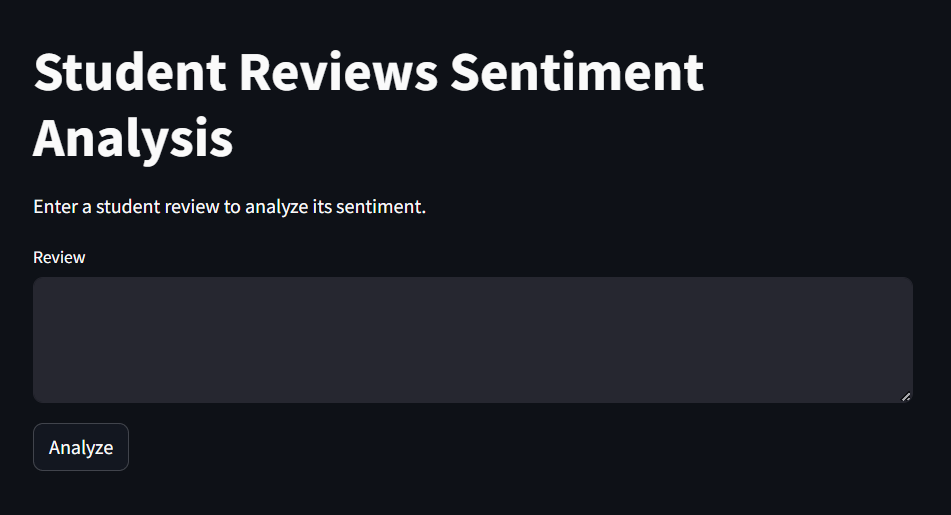
F1= 2⋅Precision⋅Recall/Precision+Recall

Precision: [3]

Precision is the proportion of accurately predicted positive observations relative to the total instances predicted as positive

**4.4 GUI(Graphical User Interface)**

The graphical user interface (GUI) implementation in this code facilitates user interaction with the "Student Reviews Sentiment Analysis." The GUI is designed using Streamlit, as indicated by the use of st.sidebar.title, st.title, st.checkbox, st.text\_area, st.button, st.success, st.warning, and various other Streamlit functions. The app presents a user-friendly interface where the user can view the first and last few rows of the dataset, input a message for analyzing the review, and initiate the prediction process by clicking the "Analyze" button. Streamlit simplifies the creation of an interactive web app, allowing users to engage with the spam detection model seamlessly through a straightforward and visually intuitive interface.

****

**Figure 4.6 the GUI of the Model**

**4.5 Conclusion-**

In conclusion, the Sentiment Analysis App successfully leverages machine learning, specifically the Support Vector Machine (SVM) model, to accurately identify sentiment of the courses. The app's graphical user interface streamlines user interaction, allowing for easy input of messages and real-time detection. Through the model training and evaluation phases, the SVM, incorporating TF-IDF vectorization, demonstrates strong performance with a high accuracy rate. The effectiveness of the sentiment analysis model is showcased not only through its successful predictions but also through the transparent presentation of evaluation metrics, including accuracy and a detailed classification report.

**Chapter 5**

**Conclusion and Future Work**

**5.1 Summary of Findings**

While working on this project we learn about the high effectiveness of the implemented Support Vector Machine (SVM) model in accurately classifying sentiment as positive, negative or neutral. The user-friendly graphical interface developed with Streamlit facilitates seamless interaction, enabling users to input messages for real-time spam detection. Rigorous model evaluation, including accuracy, precision, recall, and F1-score metrics, demonstrates the model's proficiency in distinguishing between sentiments of the reviews. The transparent presentation of evaluation metrics within the graphical interface enhances user understanding and trust in the system. Overall, the project establishes a practical and accessible solution for analyzing the sentiments of the courses, suggesting significant potential for future enhancements and real-world applications.

**5.2 Current Limitations**

While the "Analyzing Reviews in Student Reviews of Online Courses" project has demonstrated effectiveness, it is important to acknowledge certain limitations in the current approach**:**

**5.2.1 Feature Limitations:**

The model relies primarily on the textual content of reviews for classification. Introducing additional features such as email metadata or sender reputation could enhance the model's discriminatory power**.**

**5.2.2 Model Generalization:**

The model's performance may be influenced by the characteristics of the training data. Ensuring robust generalization to diverse dataset and evolving analyzing tactics remains a challenge.

**5.2.3 Scalability Concerns:**

The scalability of the model to handle large datasets or increased user loads has not been extensively tested. Evaluating its performance under varying data volumes is necessary for real-world deployment.

**5.2.4 User Feedback Integration:**

The current approach lacks a mechanism for user feedback integration, hindering the model's ability to adapt and improve based on user interactions and evolving spam patterns.

**5.2.5 Dependency on User Input:**

The model's effectiveness is contingent on user input, and its performance may vary based on the quality and nature of the messages provided. Investigating ways to handle diverse user inputs is crucial.

**5.2.6 Limited Real-Time Updates:**

The model does not currently support real-time updates, potentially causing it to become less effective over sarcastic comments. Implementing mechanisms for continuous learning and adaptation is a future consideration.

**5.2.7 Interpretability Challenges:**

The SVM model's lack of inherent interpretability may pose challenges in understanding the specific features contributing to its decisions. Incorporating model interpretability techniques could address this limitation**.**

**5.3 Future Improvements**

To enhance the performance of our object detection system, consider the following suggestions for future improvements:

Increase Data Diversity:

Incorporate a broader range of diverse and high-quality data into our training set. This expansion aims to equip the model with the ability to effectively handle various scenarios and enhance its overall adaptability.

Ensemble Methods Integration:

Incorporate ensemble learning techniques, such as combining the predictions of multiple models like LinearSVC or Gradient Boosting, to potentially enhance the overall accuracy and robustness of the spam detection system.

Dynamic Feature Engineering:

Explore advanced feature engineering strategies, including the incorporation of email metadata, sender behavior analysis, and linguistic patterns, to capture a broader range of characteristics and improve the model's ability to discern between actual and sarcastic reviews.

Real-Time Model Updates:

Implement a mechanism for real-time updates to the model to adapt to evolving spam tactics. Regularly retrain the model with new data and patterns to ensure it remains effective over time.

User Feedback Mechanism:

Introduce a user feedback mechanism within the web app to collect user inputs on model predictions. This information can be utilized to continuously refine and improve the model, addressing potential misclassifications and adapting to user-specific preferences.

Ethical Considerations Framework:

Develop a comprehensive framework to address ethical considerations, including user privacy, fairness, and potential biases in the training data. Ensure that the model deployment aligns with ethical standards and guidelines to promote responsible and transparent use.

# **References**

|  |  |
| --- | --- |
| [1] | Sci-Kit, "Linear SVC," Sci-kit learn.org, [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html. [Accessed 8 January 2024]. |
| [2] | L. Breiman, "Random Forests," in *Machine Learning*, The Netherlands, Kluwer Academic Publishers, 2001, pp. 5-32. |
| [3] | analyticsvidhya, "Precision and Recall | Essential Metrics for Machine Learning," analyticsvidhya, [Online]. Available: https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/. [Accessed 8 January 2024]. |