**UNIVERSITY STUDENTS**

**COMPLAINT PRIORITIZATION**

*A project report submitted to ICT Academy of Kerala*

*in partial fulfillment of the requirements*

*for the certification of*

**CERTIFIED SPECIALIST**

**IN**

**Machine Learning and Artificial Intelligence**

submitted by

**Team 4**

**Mothe Thomas**

**NanduKrishna .S**

**Nitha S. Kumar**

**Sariga M .S**

**Soumya Babu**



**ICT ACADEMY OF KERALA**

**THIRUVANANTHAPURAM, KERALA, INDIA**

**August 2023**

**List of Figures**

| **Figure no.** | **Figure Name** | **Page no.** |
| --- | --- | --- |
| Fig 6.1 | Word Cloud of Complaints | 18 |
| Fig 6.2 | Scatter -3D of Genre column | 19 |
| Fig 6.4 | Subplot based on Genre and year | 20 |
| Fig 6.5 | Countplot of Gender Category | 20 |
| Fig 6.6 | Countplot - Frequency of Complaint of each topic | 21 |
| Fig 6.7 | Pairplot of features | 22 |
| Fig 7.1 | Accuracy Comparison Between Bag of Words and TF-IDF | 27 |
| Fig 7.1 | Hyper Parameter Tuning of Logistic Regression | 28 |
| Fig 7.2 | Top Prioritized Complaints Using BERT | 30 |
| Fig 8.1 | Screenshot of Model Deployment | 33 |
| Fig 9.1 | Accuracies from ML models | 33 |

**List of Abbreviations**

| **EDA** | Exploratory Data Analysis |
| --- | --- |
| **TF-IDF** | Term frequency-inverse document frequency |
| **SGDC** | Stochastic Gradient Descent Classifier |

**Table of Contents**

[**Abstract 6**](#_iu5447kk2afk)

[**1. Problem Definition 7**](#_lbuxx225cjrg)

[1.1 Overview 7](#_imhob0xlbpc3)

[1.2 Problem Statement 7](#_nuosdyreox1k)

[**2. Introduction 8**](#_ve15wes91hv1)

[**3. Literature Survey 9**](#_2iztiih0zqz)

[**4. Basic Description of the Dataset 12**](#_pssq5qy4lr10)

[**5. Preprocessing 14**](#_hqr88niuc9xq)

[Complaints to Lower-Cases 14](#_sg292glpu50i)

[Replacing Contractions 14](#_96qtbw2jvv9r)

[Removing the repetitive characters 15](#_jywgaew0iseu)

[Removing the punctuations 15](#_woxe4g2c0eaa)

[Removing numbers 15](#_a5iqi9ynhckb)

[Removing stopwords 15](#_i1q7tbnhl9by)

[Tokenization 15](#_89p9jvi57dp3)

[Lemmatization 16](#_ki6li53ceq2x)

[**6. EDA 17**](#_dhit8jlvig2u)

[Word Cloud 17](#_i9ehridzy267)

[Scatter\_3D 18](#_3wiz4bszrh3m)

[Subplots 19](#_4ybjo99xgrl)

[Countplot 20](#_7uzez5oga4q5)

[Pairplot 22](#_9y6hheupa80k)

[**7. Model Selection 24**](#_pyna3i8urp0a)

[Train-Test split 24](#_mn0kq6l52l51)

[TF-IDF 24](#_8mzq9um0d4ip)

[Bag of Words 24](#_o578glie8zq0)

[**Models 24**](#_ab9bwlab6vu0)

[Random Forest 24](#_vo1ththkartd)

[Logistics regression 25](#_3mb84sbr24y)

[MultNaive Bayes 25](#_zd2iqxlqrt1e)

[Support Vector Machine (SVM) 25](#_8951igyn0var)

[Stochastic Gradient Descent (SGD) 26](#_ci41zt7ez7sv)

[Extra Tree Classifier 26](#_275lwigb1w06)

[Accuracy Comparison 27](#_z2axlnds8x2)

[Hyper parameter tuning of Logistic regression 27](#_189hn6dwz9rr)

[Modeling using BERT 28](#_cklp6wulnyew)

[Complaint Prioritization using BERT 29](#_6btmpm5fjy86)

[**8. Model Deployment 31**](#_6cp80v6541n3)

[**9. Result 33**](#_87u8enbsl2og)

[**Conclusion 34**](#_t5l2o3ve73cz)

[**References 35**](#_9zybyinjb4js)

# 

# Abstract

Efficiently addressing student complaints is vital for educational institutions. Automated complaint management systems can prioritize complaints based on severity and priority levels, enabling universities to allocate resources effectively and provide timely resolutions. This project addresses the challenge of prioritizing university students’ complaints by leveraging Natural Language Processing (NLP) techniques. The dataset selected is called “Voice Heard” used for this study and consists of reports and complaints submitted by students in a university setting. From academic grievances to campus safety concerns, this dataset offers a rich trove of insights into the student experience, providing valuable feedback for university administrators and educators. This dataset provides a unique opportunity to gain a deeper understanding of the needs and concerns of students, as well as develop data-driven solutions to enhance the university experience.

The initial phase involves preprocessing “the university students complaint”(Voices Heard) dataset to cleanse and normalize the text data. Subsequently, NLP algorithms are implemented to extract essential features for complaint classification and sentiment analysis. Machine learning models and deep learning architectures are employed for these tasks. The project uses appropriate evaluation metrics such as accuracy, precision, recall, and F1 score to evaluate the system's performance. Additionally, cross-validation techniques are employed to ensure the robustness and generalizability of the developed models. Successful implementation of this project will provide educational institutions with a valuable tool for complaint management by automating the complaint prioritization process. Therefore, universities can enhance their responsiveness and improve student satisfaction.

# 1. Problem Definition

## 1.1 Overview

Managing and addressing customer feedback and grievances requires prioritizing complaints. By analyzing and categorizing incoming complaints, organizations can identify critical issues that require immediate attention and allocate resources more efficiently. Natural Language Processing (NLP) techniques play a vital role in automating this process, enabling the extraction of key information and sentiments from unstructured text data. Complaint prioritization models rely on keyword extraction, sentiment analysis, and topic modeling to identify the most pressing concerns and assign priority levels based on severity. Besides resolving urgent issues promptly, this helps organizations proactively address recurring problems and improve overall customer service.

## 1.2 Problem Statement

The project aims to create an automated complaint prioritization system for universities using NLP techniques. This system will analyze and classify complaints based on severity and urgency, improving resource allocation and student satisfaction.

# 2. Introduction

To have an effective educational system, some issues in the academic environment need to be properly addressed. An example is the issue of the complaints prioritization system in the university. This issue has created a lot of problems for academic growth in various aspects of the educational system in the past. To support this approach, this project identifies a range of options that can be used to manage and resolve academic complaints.

But it is not humanly possible for a particular person to sit and look at every single complaint available. It would simply be a waste of time. Machine Learning plays a significantly important part here in making this process easier. Generating statistical information regarding text out of the analysis of university students’ complaints, which can be used as an inference to understand how users feel thereby improving student experiences. The process of Complaint Prioritization falling under ML helps the system understand the Complaint of a particular label made and helps to derive the results more statistically than just by intuition. The system is built using several ML algorithms that can understand the nature of complaints or a set of the same.

Complaint Prioritization provides some answers to what the most important issues are, from the perspective of students, at least. Because complaint prioritization can be automated, decisions can be made based on a significant amount of data rather than plain intuition which isn’t always right. University students’ complaints can be categorized into different categories based on its label. The Complaint types of prioritization considered in this problem are based on their label. So, we can build a model and train them with the existing data so that in the future it can classify the Student's complaints into priority which will be of great help from a university perspective. So this is the main motivation for the project.

# 3. Literature Survey

Numerous studies have been conducted in the realm of NLP complaint prioritization systems, involving various authors who employed diverse data mining algorithms. Their efforts aimed to achieve effectiveness and precision in discerning priority levels for complaints. These endeavors encompassed datasets, algorithmic approaches, experimental outcomes, and proposed future enhancements to further optimize the system's efficacy in NLP-based complaint prioritization tasks.

Blümel and Zaki, in collaboration with Mohamed, conducted a comprehensive project focused on NLP-based complaint prioritization.Recent advancements in natural language processing have been shown to be very effective for different text mining tasks and thus have provided the opportunity to enhance service research. To improve the customer service experience, this paper compares several natural language processing approaches in order to automatically prioritize incoming customer complaints for service agents. This can help companies to reduce customers’ friction and enable effective resource allocations. Our paper uses state- of-the-art feature engineering techniques (e.g., term frequency, TF-IDF and Word2Vec) to identify key words that could enable machines to prioritize complainers. We experimented with many classical machine learning classification algorithms, such as Random Forests, Support Vector Machines, Decision Trees and Logistic Regression, as well as with deep learning-based classifiers, such as convolutional neural networks, bidirectional long short-term memory, and the pre-trained language model BERT to compare the model performance. Our findings show that the pre-trained language model BERT and TF- IDF in combination with Logistic Regression yields the highest macro averaged F1-score across the multiple classes and is therefore most capable of predicting the priority group of incoming customer complaints.

In their recent study titled "Cyber Complaint Automation System," A. V. Prabhu, M. Jefiya, J. D. Joseph, T. Sunny, and C. M. Abraham explored advanced computing and communication technologies for high-performance applications, presenting their findings at the 2023 ACCTHPA conference in Ernakulam, India.Due to the Internet's quick spread and the digitali-sation of commercial activity, cybercrime has risen dramatically. Computers or data are the intended victim of cybercrime. This paper is about “Cyber Complaint Automation System''. The current cyber cell of Kerala police has a time consuming approach. Complaints can be given in two ways. A written complaint can be given to the cyber police station. Second option is to use a cyber crime reporting portal. In this portal the user has to undergo four different steps to successfully register the complaint. These complaints are stored in a database and manually classified into respective departments. This methodology of a cyber complaint takes a great deal of effort and time. The time wasted on the current system helps the criminal in covering his track. As a result, the victim's time and money are squandered, and police receive criticism. We are working to create a software that will drastically shorten the time it takes to handle a complaint in an effort to remedy this scenario. Our system automates the classification process and prioritizes emergency complaints. Thus helps the police to apprehend the criminal faster.

In the Proceedings of the Second Workshop on Economics and Natural Language Processing, Filgueiras, Joao, and their team conducted a study focusing on complaint analysis and classification within the realms of economic and food safety concerns in 2019.Governmental institutions are employing artificial intelligence techniques to deal with their specific problems and exploit their huge amounts of both structured and unstructured information. In particular, natural language processing and machine learning techniques are being used to process citizen feedback. In this paper, we report on the use of such techniques for analyzing and classifying complaints, in the context of the Portuguese Economic and Food Safety Authority. Grounded in its operational process, we address three different classification problems: target economic activity, implied infarction severity level, and institutional competence. We show promising results obtained using feature-based approaches and traditional classifiers, with accuracy scores above 70%, and analyze the shortcomings of our current results and avenues for further improvement, taking into account the intended use of our classifiers in helping human officers to cope with thousands of yearly complaints.

Presented at the 4th International Conference on Advances in Science & Technology (ICAST2021), Wadkar, Pratik, Raorane, Atreya, and Shaikh Bushra introduced their innovative "AI-driven Complaint Management System" in 2021.Citizens Complaint is important information reflecting citizen's sound. Our main objective of the organization is to give valuable and productive feedback to the citizens. The design we proposed for an AI-driven logging portal will have the strength to minimize citizens' worry and additionally, it can inspire people to promote our country by logging complaints on our website. In this paper, we propose a new model that is an AI-driven logging portal where we try to improve communication between citizens and government, and we try to make transparent communication to make your country a better place to live. There are different services for different types of complaints in your web portal. These services are used by numerous citizens on the basis of their grievances. We have created a framework that can recognize citizens' problems and provide timely feedback to citizens. This system can recognize grievances by identifying and commenting on each complaint that has been raised. The concern of citizens is treated according to the priority in this portal. That is a problem depending on the seriousness of the situation that will be prioritized.

# 4. Basic Description of the Dataset

The dataset is publicly available on Kaggle website. The Voices Heard dataset is a comprehensive collection of reports and complaints submitted by students in a university setting. From academic grievances to campus safety concerns, this dataset offers a rich trove of insights into the student experience, providing valuable feedback for university administrators and educators. With its diverse range of feedback, "Voices Heard" offers a unique opportunity to gain a better understanding of the needs and concerns of students, and to develop data-driven solutions to enhance the university experience for all. The dataset contains 1006 rows and 7 features.

**Attribute Information**

* **Genre:** Genre column has 9 categories. I.e

1. Academic Support and Resources
2. Food and Cantines
3. Financial Support
4. Online learning
5. Career opportunities
6. International student experiences
7. Athletics and sports
8. Housing and Transportation
9. Health and Well-being Support
10. Activities and Travelling
11. Student Affairs

* **Reports:** All students complaint present in Reports column
* **Age:** Student age
* **Gpa:** Student mark
* **Year:** Course completion year
* **Count:** Count of complaints
* **Gender:** Gender of student
* The university students complaint dataset is taken for complaint prioritization.
* There are only int, float and object data types present in the dataset.
* The ‘Genre’ is the target of the dataset.
* Receiving the new input to ‘Report’ feature, automatically categorizes the feature ‘Genre’.
* The dataset didn’t have any missing values.

# 

# 5. Preprocessing

NLP (Natural Language Processing) preprocessing techniques are essential steps to clean, normalize, and transform raw text data into a format suitable for analysis and modeling. These techniques help to remove noise, irrelevant information, and standardize the text, making it easier for NLP algorithms to understand and process the data effectively. Here are some common NLP preprocessing techniques that are used to clean the dataset:

* Conversion to lower-case
* Replacing contractions
* Removing the repetitive characters
* Removing symbols
* Stop words removal
* Lemmatization
* Tokenization

### Complaints to Lower-Cases

The process of converting all text to lowercase is a common pre-processing step in . This step is performed to ensure that words that have different capitalization are treated as the same, which is important for correctly identifying sentiment. For example, the word "good" and "Good" should be considered the same sentiment-wise.

### Replacing Contractions

Contractions are often used in social media and text messaging, and they can be difficult for machine learning models to understand. Replacing abbreviations with their full words can help make the text more understandable and easier to analyze. For example, "can’t” can be replaced with "cannot".

### 

### Removing the repetitive characters

Sometimes the tweets contain words with consecutive repeated characters. For example, instead of typing ‘hi’, some may have typed ‘hiiiiii’. We have to remove this repeated sequence. For that we check whether the character occurs more than two times consecutively and replace them with the same word containing only two occurrences of that repeated character.

### Removing the punctuations

Punctuations are removed from the list of punctuation in the string library. Here a function is created which searches and removes the punctuation to the applied field. maketrans() method is used to create a mapping table which replaces or removes the characters. translate() method returns the changed value.

### Removing numbers

Numbers are removed using regular expression. A function is created to search and remove the numbers in the applied field. sub() method is used to find and remove numbers.

### Removing stopwords

Stop words are a set of commonly used words in a language. Examples of stop words in English are “a”, “the”, “is”, “are” and etc. Stop words are commonly used in Text Mining and Natural Language Processing (NLP) to eliminate words that are so commonly used that they carry very little useful information. To eliminate such words, a stop word package is downloaded. From that package english stopwords are selected. “Not” is then removed from the list because it can affect the bias of the data. Each sentence is then tokenized and removes the stop words by storing the values which are not present in stop words.

### Tokenization

A tokenizer breaks unstructured data and natural language text into chunks of information that can be considered as discrete elements. The token occurrences in a document can be used directly as a vector representing that document.

This immediately turns an unstructured string (text document) into a numerical data structure suitable for machine learning. They can also be used directly by a computer to trigger useful actions and responses. Or they might be used in a machine learning pipeline as features that trigger more complex decisions or behavior.[2] Tokenization can separate sentences, words, characters, or subwords. When we split the text into sentences, we call it sentence tokenization. For words, we call it word tokenization.

### Lemmatization

Lemmatization is the process of converting a word form to its meaningful base form. Example: better - good, caring - care. It is the same as stemming but lemmatization is preferred mostly because stemming converts the word form without knowing its context but lemmatization converts the text by knowing its word form.Here we have used the wordnet lemmatizer.WordNetLemmatizer is a library that is imported from nltk.stem which looks for lemmas of words from the WordNet Database.

# 

# 

# 

# 

# 

# 6. EDA

Exploratory Data Analysis (EDA) is a data analytics process to understand the data in depth and learn the different data characteristics, often with visual means. This allows you to get a better feel of your data and find useful patterns in it.

### Word Cloud

Here we have done word-cloud for EDA. Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word clouds are widely used for analyzing data from social network websites.The WordCloud function provides a lot of parameters that we can tweak according to our desire. Let us understand a few of them.

* width/height : To adjust the height and width of the wordcloud
* random\_state : To recreate the same plot every time we run the function. The random\_state parameter has to be an integer value.
* background\_color : To set a background\_color.
* colormap : To set up the color theme for the words.
* collocations : To include bigrams of two words when set to True. The default value is True
* stopwords : To set the list of words that needs to be eliminated. This list can include trivial words like this, that, is, was, the, etc. If this parameter is set to None, then function will consider a built-in list of STOPWORDS
* max\_font\_size : To set the maximum font size of the largest word.
* normalize\_plurals : To keep or remove the trailing 's' from the words
* max\_words: It specifies the maximum number of the word, default is 200

To display word cloud image .imshow() method of matplotlib.pyplot is used. In the above code, we are using two parameters:

* wordcloud: created in the above step
* interpolation=”bilinear”: used to display a smoother image.



Fig 6.1 - Wordcloud of Complaints

Based on this word-cloud, we can infer that most repeated words are University, Student, difficult, campus, feel, hard etc.

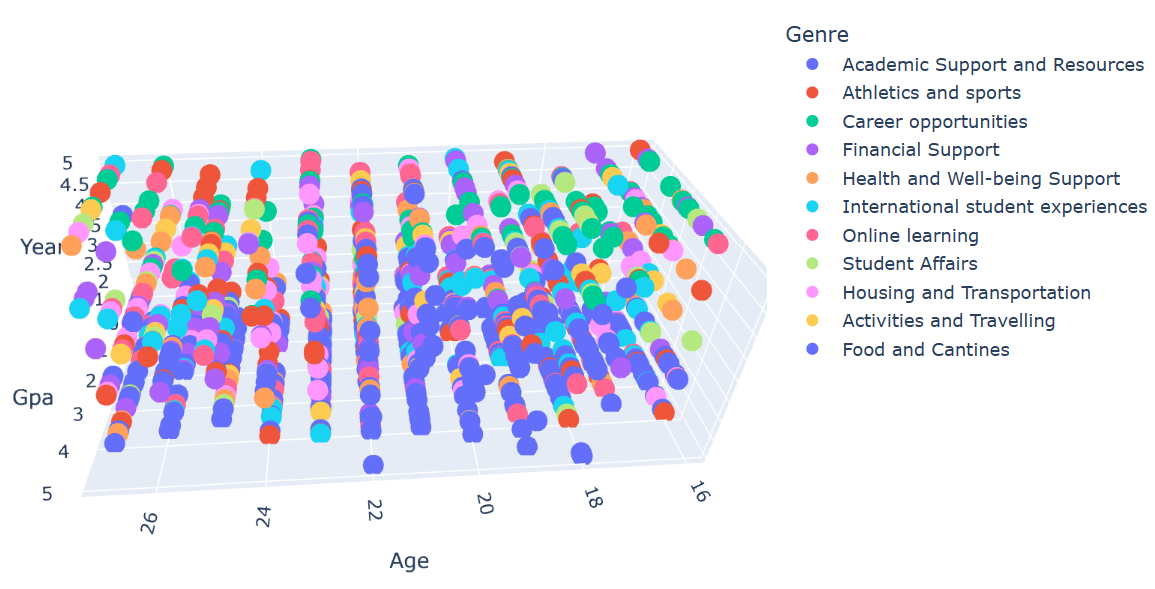
### Scatter\_3D

A 3D scatter plot (or 3D scatter chart) is a type of data visualization that displays data points in a three-dimensional space. It is an extension of the traditional scatter plot, which is a two-dimensional representation of data points using Cartesian coordinates (x, y).

In a 3D scatter plot, each data point is represented by three numerical values, usually corresponding to three different variables. These variables are usually denoted as x, y, and z, which represent the three dimensions of the plot. The x, y, and z coordinates determine the position of each data point in the 3D space.

We have a DataFrame df with columns 'Age'(x), 'Gpa'(y), 'Year'(z), and 'Genre’(color).

This code will create an interactive 3D scatter plot using Plotly Express, where 'Age', 'Gpa', and 'Year' will be represented along the x, y, and z-axes, respectively. The points will be colored based on the 'Genre' column in the DataFrame.



### 

Fig 6.2 - Scatter -3D of Genre column

### Subplots

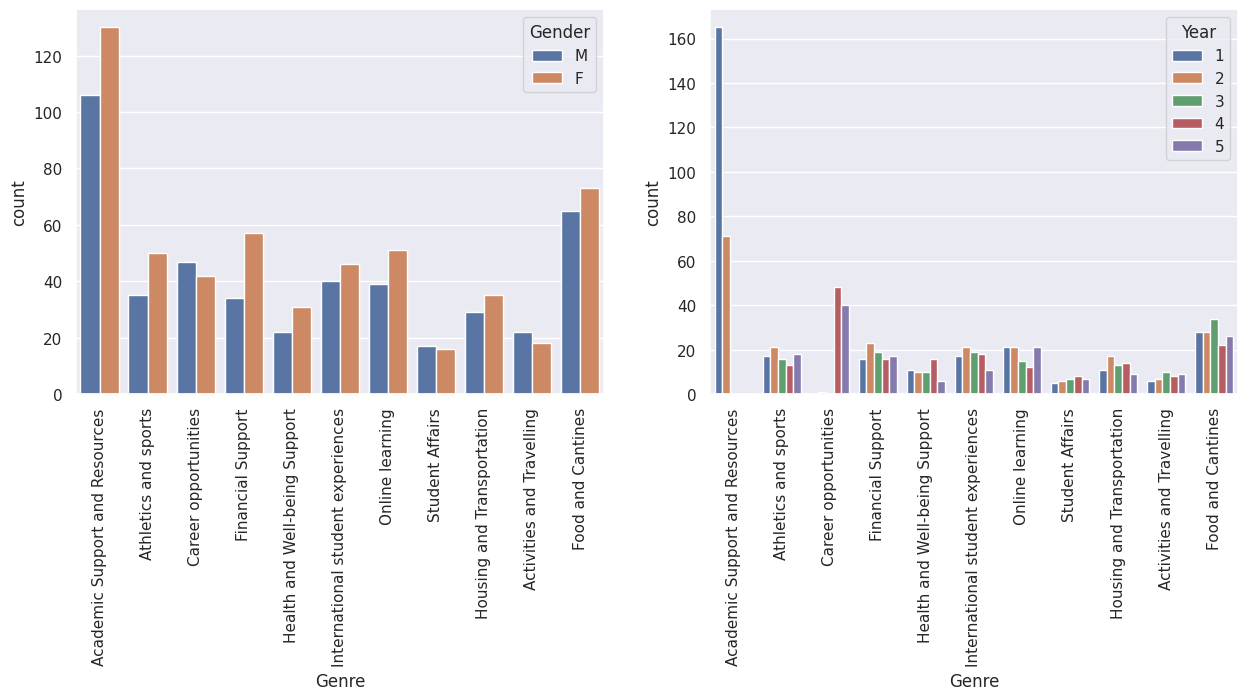
sns.countplot(data=df, x = df['Genre'], hue=df['Gender'])

The plt.subplot function in Matplotlib is used to create a grid of subplots within a single figure. It allows you to arrange multiple plots in rows and columns.

This count plot is helpful to visualize how different genres are distributed among different genders in your dataset. For instance, you can quickly see which genres are more popular among males and females or any other patterns related to genre distribution based on gender.

sns.countplot(data=df, x = df['Genre'], hue=df['Year'])

This count plot is helpful to visualize how different genres are distributed across different years in your dataset. For instance, you can quickly see how the popularity of different genres evolves over the years or any other patterns related to genre distribution based on the year.



**Fig 6.3 -Subplot based Genre and Count Fig 6.4 - Subplot based on Genre and year**

### Countplot

**sns.countplot(x='Gender',data=df)**

This code will generate a count plot with the 'Gender' column on the x-axis, showing the number of occurrences of each gender category.

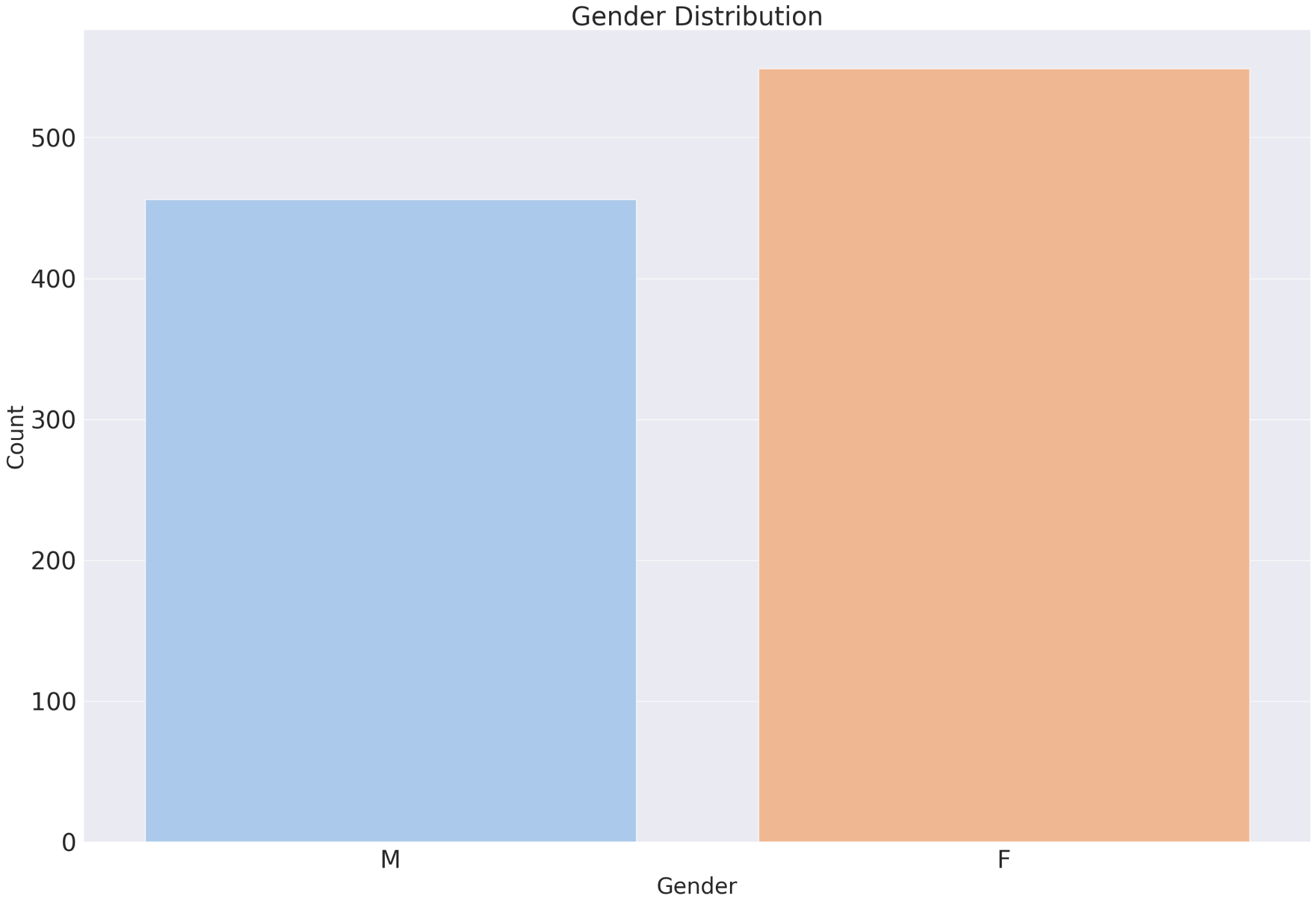


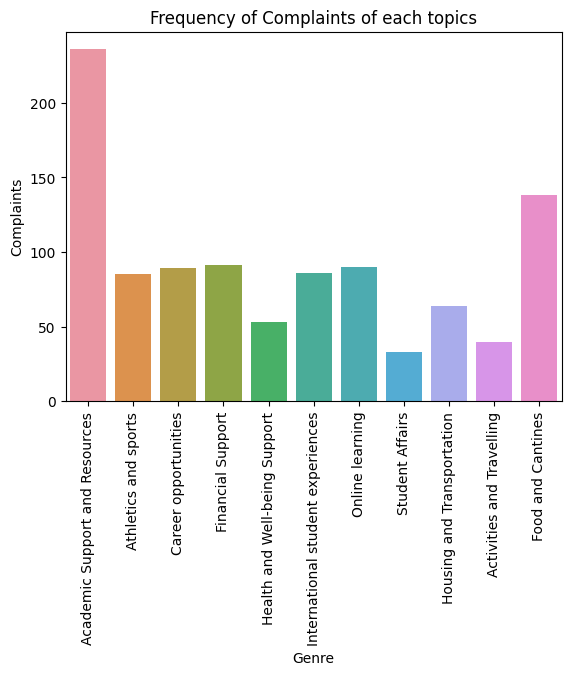
Fig 6.5 - Countplot of Gender Category

As per graph, it is visible that females are raised more complaints than males.

**sns.countplot(x='Genre',data=df)**

plt.title('Frequency of Complaints of each topics')

This code will generate a count plot with the 'Genre' column on the x-axis, showing the number of occurrences of each genre category.

 Fig 6.6 - Countplot - Frequency of Complaint of each topic

Based on the above graph, we can infer that Most complaints are from Academic Support and Resources and then Food and Cantines.

### Pairplot

sns.pairplot(df)

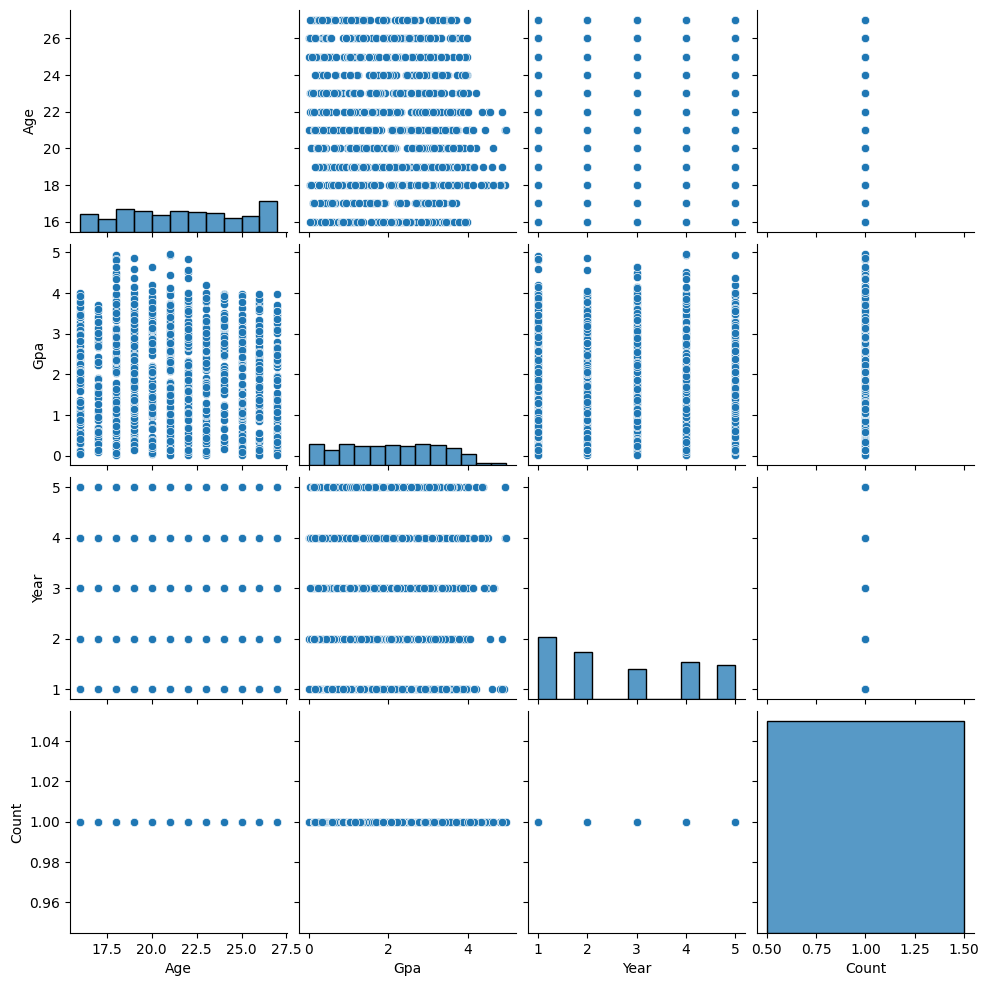


Fig 6.7 - Pairplot of features

This plot is particularly useful for exploring correlations and patterns between variables in your dataset. It can help you identify potential relationships, trends, and outliers, giving you a visual overview of the data distribution and associations between variables.

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 

# 7. Model Selection

### 

### Train-Test split

The dataset is split into train and test using train test split. The training set is used to train the model and the testing set is used to test the model.

Here we used two feature engineering methods - TF-IDF & Bag of Words

### TF-IDF

TF-IDF stands for “Term Frequency — Inverse Document Frequency”. This is a technique to quantify words in a set of documents. We generally compute a score for each word to signify its importance in the document and corpus. This method is a widely used technique in Information Retrieval and Text Mining.

### Bag of Words

The bag-of-words (BOW) model is a representation that turns arbitrary text into fixed-length vectors by counting how many times each word appears. This process is often referred to as vectorization.It’s an algorithm that transforms the text into fixed-length vectors. This is possible by counting the number of times the word is present in a document. The word occurrences allow to compare different documents and evaluate their similarities for applications, such as search, document classification, and topic modeling.

## Models

### 

### Random Forest

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. We got the following results.

Accuracy with TF-IDF as 0.930348

Accuracy with Bag of Words as 0.940299

### Logistics regression

Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on. We got the following results.

Accuracy with TF-IDF as 0.895522

Accuracy with Bag of Words as 0.940299

### MultNaive Bayes

Multinomial Naive Bayes is a probabilistic machine learning algorithm widely used for text classification tasks, especially when dealing with discrete feature data, such as word frequencies in documents. It is a variant of the Naive Bayes algorithm, and it assumes that features are conditionally independent given the class label. In the context of text classification, it models the occurrence counts of words or features within each class. During the training phase, the algorithm estimates the probability distributions of feature occurrences for each class. Then, during prediction, it calculates the likelihood of a document belonging to each class based on the observed word frequencies and selects the class with the highest likelihood as the predicted class for the input document.

Accuracy with TF-IDF as 0.741294

Accuracy with Bag of Words as 0.920398

### **Support Vector Machine (SVM)**

Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification and regression tasks. It works by finding the optimal hyperplane that best separates different classes in a high-dimensional feature space. The main goal of SVM is to maximize the margin between classes, leading to better generalization and improved performance on unseen data.

Accuracy with TF-IDF as 0.910448

Accuracy with Bag of Words as 0.930348

### Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (SGD) is a simple yet very efficient approach to discriminative learning of linear classifiers under convex loss functions such as (linear) Support Vector Machines and Logistic Regression. We got

Accuracy with TF-IDF as 0.935323

Accuracy with Bag of Words as 0.915423

## Extra Tree Classifier

The Extra Trees Classifier is an ensemble machine learning algorithm used for classification tasks. It is an extension of the Random Forest algorithm and works by constructing multiple decision trees during the training process. However, unlike Random Forest, Extra Trees introduces additional randomness by randomly selecting feature subsets and using random thresholds to split nodes. This approach promotes diversity among the individual decision trees and reduces overfitting. During the prediction phase, the algorithm aggregates the predictions of all trees to make the final classification decision. Extra Trees is computationally efficient and can handle high-dimensional datasets, making it a powerful choice for classification problems in various domains.

Accuracy with TF-IDF as 0.955224

Accuracy with Bag of Words as 0.950249

## 

## 

## 

## Accuracy Comparison

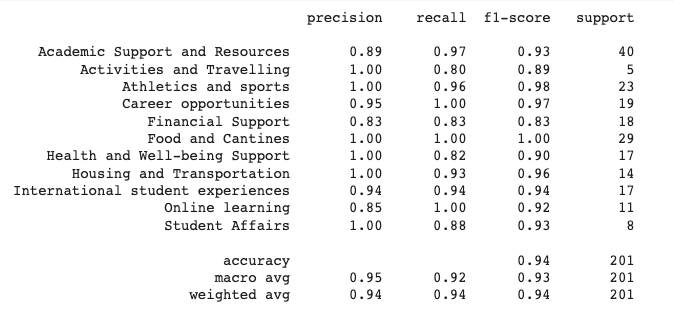
# 

Fig 7.1 - Accuracy Comparison Between Bag of Words and TF-IDF

### Hyper parameter tuning of Logistic regression

Hyperparameter tuning is a crucial step in optimizing the performance of the ExtraTreeClassifier. The initialized ExtraTreeClassifier will be fine-tuned using the defined hyperparameter grid to explore various combinations of hyperparameters. This tuning process aims to discover the best set of hyperparameter values that yield the highest accuracy or other performance metrics for the particular classification problem addressed in your project. By experimenting with different values for n\_estimators, max\_depth, min\_samples\_split, and min\_samples\_leaf, the optimal configuration will be identified to strike a balance between model complexity and generalization.

# 

 Fig 7.1 - Hyper Parameter Tuning of Logistic Regression

### Modeling using BERT

BERT (Bidirectional Encoder Representations from Transformers) is a revolutionary natural language processing (NLP) model introduced by Google AI researchers. It redefined how pre-trained language representations are learned and utilized. The transformative aspect of BERT lies in its bidirectional approach to understanding context within language. Unlike previous models that processed text in a left-to-right or right-to-left manner, BERT considers both directions simultaneously. This means that each word's representation is influenced by its surrounding words, enabling BERT to grasp the intricacies of language semantics and nuances more effectively.

The architecture of BERT consists of an encoder that employs transformer layers. Transformers are essential for BERT's performance, as they facilitate the model's contextual understanding by allowing it to focus on different parts of the input text while accounting for the relationships between words. The pre-training phase involves unsupervised learning on massive corpora, where BERT learns to predict missing words in sentences. This process imbues the model with a deep understanding of context and contextually appropriate word usage. After pre-training, BERT can be fine-tuned for specific tasks, which involves training on task-specific datasets with labeled examples. This fine-tuning process adapts the general language understanding capabilities of BERT to more specific NLP tasks, such as sentiment analysis, question answering, and more.

The impact of BERT on the NLP landscape has been substantial. It has raised the bar for performance across a wide range of tasks and benchmark datasets. Additionally, the success of BERT has inspired the development of various other transformer-based models, each tailored to specific aspects of language understanding and generation. Models like GPT-3, RoBERTa, and T5 have pushed the boundaries of what is possible in NLP and have led to remarkable advances in machine understanding of human language. BERT's innovative approach to capturing contextual information has left a lasting imprint on the field of NLP, paving the way for more sophisticated and accurate language models.

## Complaint Prioritization using BERT

The process begins by encoding each complaint using the BERT (Bidirectional Encoder Representations from Transformers) model, which captures intricate semantic relationships and contextual information within the text. This results in contextualized embeddings that capture the essence of each complaint in a high-dimensional vector space.

Subsequently, pairwise distance is computed between these embeddings, quantifying the dissimilarity or similarity between pairs of complaints. Lower pairwise distances indicate higher textual similarity, implying that complaints are closely related in content. On the other hand, higher pairwise distances reflect greater dissimilarity, signifying unique issues raised by students.

These computed pairwise distances are then used to calculate priority scores for each complaint. Priority is assigned inversely proportional to the pairwise distance, where smaller distances correspond to higher priorities. Complaints with smaller distances signify similar issues faced by multiple students, suggesting potential widespread impact. In contrast, complaints with larger distances imply distinct concerns raised by individuals.

Finally, the prioritized list of complaints is generated by sorting them based on their priority scores. This list presents complaints in descending order of urgency, facilitating administrative action to address the most pressing issues affecting the student body. By combining the power of BERT embeddings and pairwise distance, the system effectively identifies and ranks complaints, enabling efficient allocation of resources to enhance the student experience.

Below figure represents the Visualization of the top-prioritized complaints.

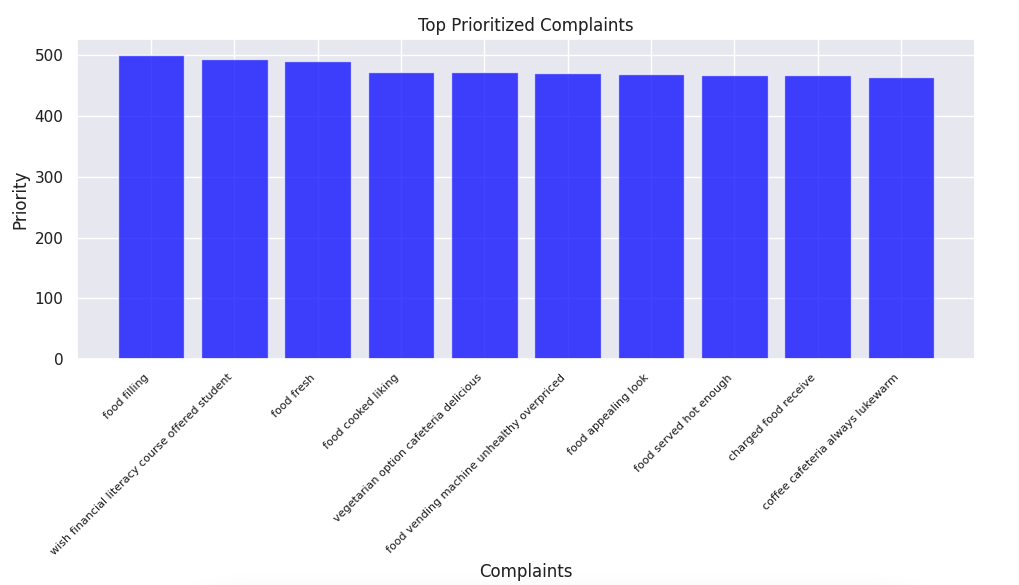


Fig 7.2 - Top Prioritized Complaints Using BERT

# 8. Model Deployment

Streamlit is chosen for model deployment of this project. It is an open-source Python library that simplifies the process of creating interactive web applications for data science and machine learning projects. With its intuitive and user-friendly interface, developers can transform data scripts into interactive web apps without extensive web development knowledge. Streamlit offers a variety of widgets to visualize data, create charts, and showcase machine learning models, enabling easy customization of the app's layout and functionality. This library has gained popularity for its ability to streamline the deployment of data-driven applications, making it an efficient tool for presenting and sharing insights from data analyses and machine learning experiments

The screen shot of the website is,

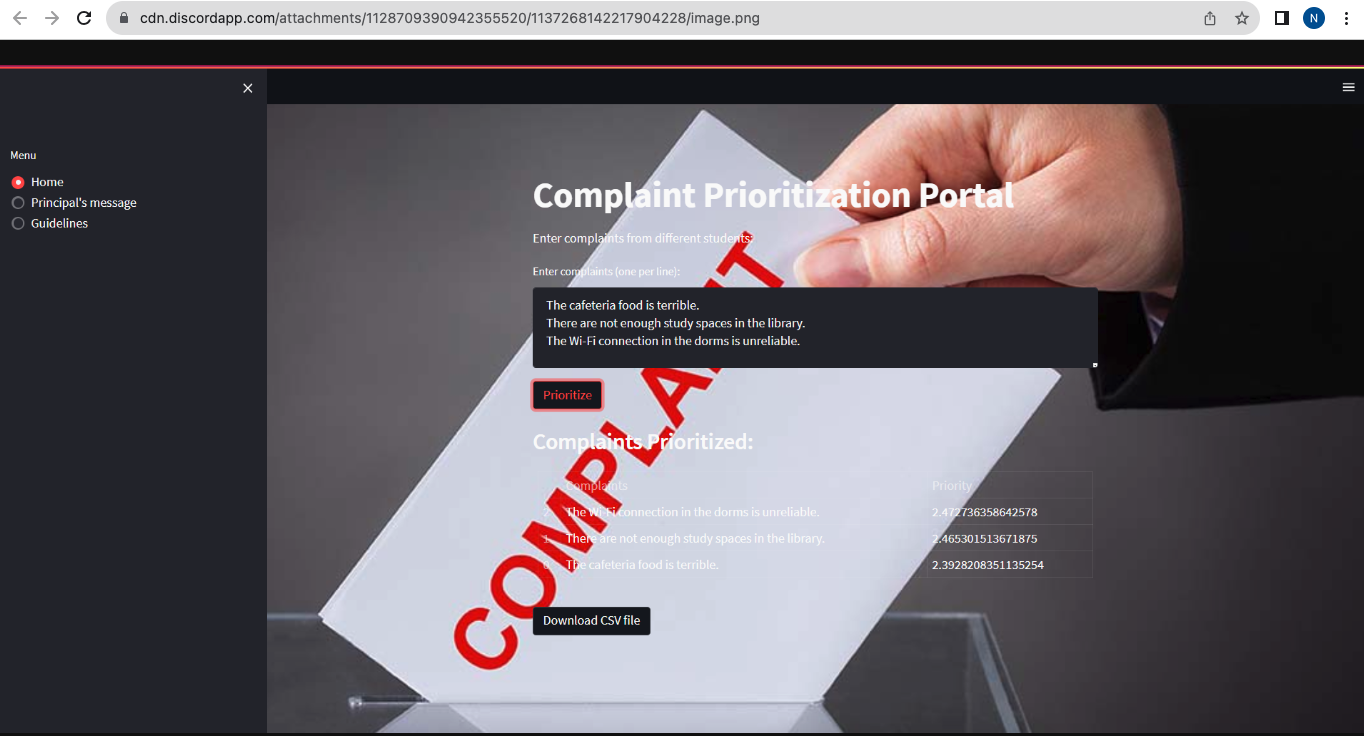


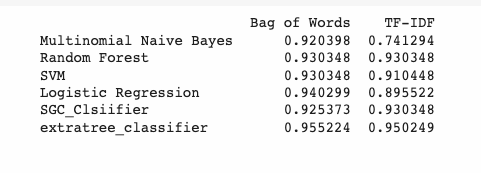
Fig 8.1 -Screenshot of Model Deployment

Link of the website is,

<https://complaintspriorityappict.streamlit.app/>

# 9. Result

The University Students Complaint dataset contains 1006 rows and 7 features. which were initially preprocessed and applied with the various ML models . The ML models showed the below accuracies :



# 

Fig 9.1 - Accuracies from ML models

In order to try a new model for complaint prioritization, BERT is used for further steps. In the results phase, the application of the BERT model for encoding complaints has proven successful in capturing intricate semantic relationships and contextual nuances within the text, generating comprehensive embeddings in a high-dimensional vector space. The subsequent calculation of pairwise distances has facilitated the quantification of textual similarity or dissimilarity between complaints. The model's capacity to distinguish lower pairwise distances as indicating related content and higher distances as reflective of unique concerns has been instrumental in effectively assigning priority scores. This prioritization process has led to the generation of a well-ordered list of complaints, ranked based on their priority scores. This system, integrating BERT embeddings and pairwise distance computation, presents a robust framework for identifying and ranking complaints, thus optimizing resource allocation and enhancing the overall student experience.

# Conclusion

In the pursuit of enhancing University complaint management, the project "University Complaint Prioritization'' embarked on a transformative journey, leveraging advanced Natural Language Processing (NLP) techniques. The foundation of this project was the Voiceheard dataset sourced from Kaggle, providing a comprehensive repository of student complaints. Our primary objective was to harness the power of the revolutionary BERT model to intelligently categorize and score these grievances based on their priority levels, ultimately fostering a more responsive and efficient complaint resolution process.

The process initiates with encoding each complaint using the BERT (Bidirectional Encoder Representations from Transformers) model, which aptly captures intricate semantic relationships and contextual nuances within the text. The outcome is a collection of contextualized embeddings, encapsulating the core essence of each complaint within a multi-dimensional vector framework. Following this, the calculation of pairwise distances between these embeddings ensues, offering insights into the textual affinity or dissimilarity between complaint pairs. Smaller pairwise distances signify textual resemblance, indicative of closely aligned content, while larger distances underscore distinct concerns. These computed distances translate into priority scores for each complaint, inversely related to their distance. This prioritization model not only streamlines administrative action but also optimizes resource allocation, addressing urgent concerns and enhancing the holistic student experience.

In conclusion, the "University Complaint Prioritization'' project offered a compelling demonstration of the power of BERT in the realm of complaint management. Rooted in the Voiceheard dataset, our model exhibited promising potential for revolutionizing the way universities address student concerns. As we celebrate our project's achievements, we eagerly anticipate the continued evolution of this field through enhancements in vocabulary, dynamic learning, and the pursuit of ever-increasing accuracy. This project represents a significant step towards creating a more effective and student-centric complaint resolution system within the realm of higher education.

# References

<https://www.researchgate.net/publication/369146797_Prediction_and_Analysis_of_Customer_Complaints_Using_Machine_Learning_Techniques>

<https://www.igi-global.com/article/prediction-and-analysis-of-customer-complaints-using-machine-learning-techniques/319716>

<https://www.kaggle.com/kaggle/us-consumer-finance-complaints>

<https://www.kaggle.com/datasets/ashwinik/consumer-complaints-financial-products>

<https://www.consumerfinance.gov/data-research/consumer-complaints/search/?date_received_max=2023-07-10&date_received_min=2023-04-10&has_narrative=true&page=1&searchField=all&size=25&sort=created_date_desc&tab=List>

<https://www.kaggle.com/datasets/selener/consumer-complaint-database/code>

<https://monkeylearn.com/blog/customer-complaint-classification/>

<https://www.kaggle.com/datasets/shashwatwork/consume-complaints-dataset-fo-nlp?resource=download>

<https://www.kaggle.com/code/saurabhsawhney/nlp-complaints-classification>

<https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html>