Next-Gen College CRM: Leveraging BERT for Student Feedback Classification in a Full Stack Framework

Dr D. Srinivas Rao[1], V. Manoj Kumar[2], P. Jogeeswara V. N. S[3], Adishankar Bhat[4], P. Renaiah[5], P. Pavan Kalyan[6]

[1] Department of Information Technology, dammavalam2@gmail.com ,VNRVJIET, Bachupally, Hyderabad, India,
[2] Department of Information Technology, manojkumarv2251@gmail.com , VNRVJIET, Bachupally, Hyderabad, India,
[3] Department of Information Technology, jogeeswarapuvvala@gmail.com , VNRVJIET, Bachupally, Hyderabad, India,
[4] Department of Information Technology, adishankara.bhat@gmail.com , VNRVJIET, Bachupally, Hyderabad, India
[5] Department of Information Technology, renaiah9381@gmail.com , VNRVJIET, Bachupally, Hyderabad, India
[6] Department of Information Technology, pavankalyan142211@gmail.com , VNRVJIET, Bachupally, Hyderabad, India

Abstract: CRM systems are designed to manage the interaction between the company and its customers in the business field. In the education field, CRM plays a key role in managing the interactions and communication between students and the college staff. This research aims to improve relations of key stakeholders, including students, faculty, and staff and satisfaction, by addressing the feedback given by the students more effectively through the incorporation of advanced technologies in NLP, like the BERT model for text classification to integrate in full stack framework. The ability of the BERT model to have a better understanding of the context and the granularity of classification makes it a stronger tool than traditional methods like automated content analysis (ACA) and conventional sentiment analysis tools that lose the contextual meaning of the sentences and find difficulties in analyzing complexities in the sentences. In this work, the comparison of the different types of transformer models for the classification of the complaints of students is presented. Some of the models include BERT base uncased, BERT large uncased, Distil BERT uncased, RoBERTa base, RoBERTa large, ALBERT base, and ALBERT large models. Evaluation of the performance of these models was done by using training accuracy and testing accuracy. In the given experiment, the BERT base uncased was found to give the highest validation accuracy of 100% on both 90-10 & 80-20 train-test splits thus making it the most effective in this classification task.

Keywords: BERT, Transformers, Customer Relation Management (CRM)

1. Introduction

CRM systems are required for the overall and enhanced management of students in educational institutions [1]. The students are customers of the institution, and therefore, their satisfaction plays a key role in the success of the institution. CRM helps institutions organize the interactions of their customers, more specifically students, throughout their educational lifecycle [1]. Collecting students' information is important in CRM systems because the information collected assists the institution in working closely with the students and offering them support and services to enhance student satisfaction and retention [1]. CRM facilitates understanding of students both as clients and customers by extending insight into the kind of services that students require and providing methods of reaching out to students, as well as offering interaction with alumni [1]. Alderman, L., Towers, S., & Bannah, S. (2012) [2] reviewed the existing literature on student feedback systems in institutions and identified trends in their design and usage. The literature says that the feedback given by the students is widely used in evaluating the effectiveness of teaching, but there are variations in how the feedback is collected, organized, and utilized [2]. They mentioned that the feedback is collected through surveys that can be online or in paper form and are collected during mid-semester and end-semester. They mentioned the challenges of a low response rate and the potential for bias in the feedback [2]. They provided an environmental scan that focused on the feedback systems used by various universities and examined the system implemented at Queensland University of Technology (QUT) in detail [2]. Berk, R. A. (2005) [3] provided an in-depth examination of 12 different strategies that can be used to evaluate the effectiveness of teaching in institutions [3]. The author mentioned the strengths and limitations of the 12 strategies, which are student ratings, peer ratings, self-evaluation, videos, student interviews, alumni ratings, employer ratings, administrator ratings, teaching scholarships, teaching awards, learning outcome measures, and teaching portfolios, aiming for a multi-source, multi-method approach for evaluating teaching in institutions [3]. The single method of evaluation is insufficient to evaluate the full scope of the effectiveness of the instructor [3]. Mandouit, Luke. (2018) [4] proposed an understanding of the impact of feedback on teaching practices by students. The author said that in relation to the task of establishing teaching objectives, students' views should be taken into account because they are a direct reflection of how the authors teach [4]. The study employed an action research methodology that consists of a cycle of steps, not linear but cyclical:

planning, acting, observing or controlling, and reflecting or evaluating, and they can be run repeatedly to make improvements based on the observation [4]. The feedback is gathered through surveys, interviews, and reflective diaries and refers to issues related to the clarity of instructions, engagement, and teaching methods used in the classroom [4]. The author fully supported student feedback as an innovative and useful tool in fashioning teaching strategies; the teachers that took their time to work with the feedback that the students have given them are able to change and justify their institutional methodologies, which would enhance the results of their students [4]. The author insisted on the fact that not only should the feedback be collected, but it should also be actively used in the analysis of the teaching [4]. T. Shaik et al. [5] proposed an extensive overview of how natural language processing (NLP) techniques are being utilized to analyze educational feedback and the challenges faced [5]. The study revealed the trends in NLP for educational feedback: automated analysis, which automates the analysis of large volumes of feedback; sentiment analysis, which represents the emotion of the student and helps in understanding student satisfaction; topic modeling, which extracts the key themes in the feedback and allows instructors to identify the repeating issues; and text classification, which classifies the feedback into predefined categories, i.e., positive, negative, and neutral, which are included in sentiment analysis [5]. The author highlighted the challenges faced in adopting NLP for feedback analysis: Data Quality and Quantity: The feedback varies in quality, making it difficult to analyze effectively. Context Understanding: NLP models may not understand the context in the feedback [5].

Recognizing the drawbacks in existing systems that involve the collection of feedback at mid-semester and the end-semester, our research proposes an efficient and effective text classification model based on the transformer architecture commonly known as BERT, which stands for Bidirectional Encoder Representations from Transformers, and its variants: DistilBERT uncased, BERT base uncased, BERT large uncased, RoBERTa base, RoBERTa large, ALBERT base v2, and ALBERT large v2 which preserves context and data quality as it is trained on large data.

Further, we have introduced a unique complaints system where a student can give complaints and feedback at any time, not at fixed time intervals. This system exists as a module of a full-stack application where students, faculty, and alumni can engage with one another. Every user makes use of a login, which is recorded in the database to allow for safe and individualized use. Regarding the complaint, once a student provides it, it is sent to our proposed BERT-based model, where it is categorized into one of the identified categories and thereafter stored in the database. The received complaints are next combined to form a Word cloud, which visualizes the most frequent issues and can be used for constant performance enhancement.

2. Related works

Cardona, Tatiana & Cudney, Elizabeth & Hoerl, Roger & Snyder, Jennifer. (2020) [6] provided a comprehensive review of applying data mining techniques and ML models that can be used for predicting student retention. The authors focused on various techniques, such as classification, regression, clustering, and ensemble methods, that encourage the usage of ML in higher education [6]. K. Z. Aung and N. N. Myo [7] employed lexicon-based sentiment analysis to categorize student feedback as positive, negative, or neutral and found it useful to have an automatic classification of student comments regarding their educational experience [7]. The tool applied in the study is the VADER (Valence Aware Dictionary and Sentiment Reasoner), which is one of the best for the analysis of text from social media [7]. L. Balachandran and A. Kirupananda [8] proposed a system that is an aspect-based sentiment analysis that assesses the online review of a higher education institution to enable the student to make the right decision about which institution to join through the use of social media-unstructured data about the institution [8]. The presented system mainly tends to know specific aspects of the institutions, like faculty, infrastructure, courses, etc., from the user reviews and finds out whether they are positive, negative, or neutral [8]. Berardinelli, Nabeela & Gaber, Mohamed & Haig, Ella. (2013) [9] explored how sentiment analysis techniques can be applied in educational institutions for analyzing student feedback. The authors introduced a system called SA-E (Sentiment Analysis for Education), which is designed to extract and evaluate sentiments from the student's feedback, which are either positive, negative, or neutral [9]. Wook, Muslihah, et al. [10] developed a system called OMFeedback that employs opinion mining techniques for feedback analysis in the case of students. OMFeedback is a feedback analysis system that is premised on the lexicon-based sentiment analysis of the students' feedback [10]. They rely on a set of lexicon words and phrases that are associated with sentiment and determine the sentiment of the feedback as positive or negative or neutral [10]. Dake, Delali Kwasi, and Esther Gyimah [11] have presented sentiment analysis techniques for qualitative responses given by students. The authors used four classifiers: Naïve Bayes, Support Vector Machine, J48 Decision Tree, and Random Forest, out of which SVM achieved maximum accuracy compared to the others and scored 63.79% accuracy [11]. Shaik, Thanveer, et al. [12] provided an extensive review of sentiment analysis and opinion mining techniques that can be applied to educational data. The study provides the traditional methods, machine learning methods, and deep learning methods that are used to analyze feedback from students and also provides the effectiveness as well as limitations of implementing these approaches [12]. The authors reviewed the methodologies that can be used to perform sentiment analysis: traditional methods: lexicon baes; machine learning approaches: support vector machines (SVM); naive bayes; decision trees; deep learning methods: LSTMs; transformer-BERT models [12]. The study demonstrates the importance of integrating these techniques to improve the effectiveness and accuracy of sentiment analysis in education [12]. Shah, Mahsood & Pabel, Anja. (2019) [13] proposed how qualitative student feedback can be utilized to enhance the student experience and satisfaction. The study explores the use of a text analytics tool called Leximancer to analyze qualitative feedback [13]. The methodology discussed involves segmenting feedback into categories: on-campus and online experiences, and further into the best aspects and aspects that need improvement. Qualitative data is transformed into quantitative insights by Leximancer to enhance student experience [13]. Abaidullah, Anwar Muhammad, Naseer Ahmed, and Edriss Ali [14] worked on clustering analysis to identify the hidden pattern in the student feedback. For the analysis of the student feedback, the study uses the K-mean clustering algorithm [14]. K-means is one of the clustering methods that are used in unsupervised learning techniques to group the data points based on their similarities. The feedback data is partitioned into clusters, and each cluster, on its part, is a group of feedback with similar characteristics, which facilitates the comparison and identification of similar patterns in the feedback [14]. Altrabsheh, N., Cocea, M., Fallahkhair, S. (2014) [15] suggested applying sentiment analysis as a method for real-time responses to students' feedback. The study used four different algorithms for sentiment classification, which include Naive Bayes, Complement Naive Bayes, Maximum Entropy, and SVM, to distinguish the feedback as 'Positive', 'Negative', or 'Neutral' [15]. The SVM classifier implemented yielded the highest level of accuracy of 94%, while Complement Naive Bayes was at 84%. These classifiers turned out to be helpful in determining the sentiment of the feedback [15]. The system was meant to analyze sentiment in real time; the feedback gathered was required by the classifier, and the results were immediate [15].

3. Methodology

3.1 Dataset Description

The first process of text classification is to obtain text data fit for the area of interest of the research. For this work, a data set with 1006 records that have been classified into eleven complaint categories available on Kaggle have been used.

Table-1: Distribution of Complaints in the dataset and oversampled values.

Category	Value counts (Initial)	Oversampled
Academic Support and Resources	236	200
Food and Cantines	138	200
Financial Support	91	200
Online learning	90	200
Career opportunities	89	200
International student experiences	86	200
Athletics and sports	85	200
Housing and Transportation	64	200
Health and Well-being Support	53	200
Activities and travelling	40	200
Student Affairs	33	200

(a) (b)

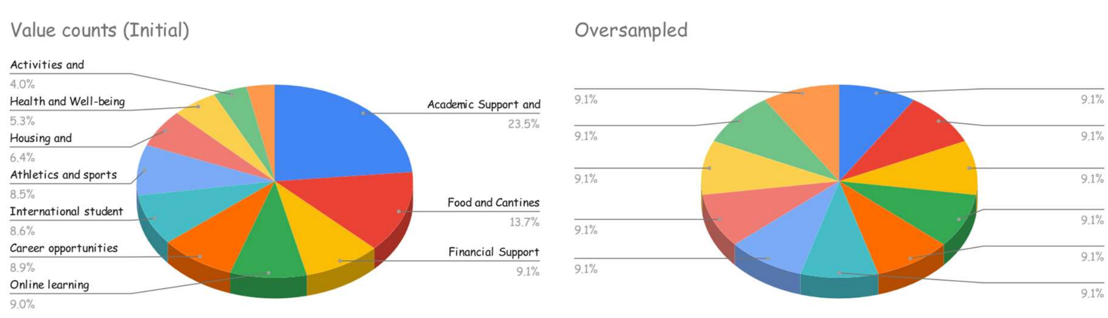


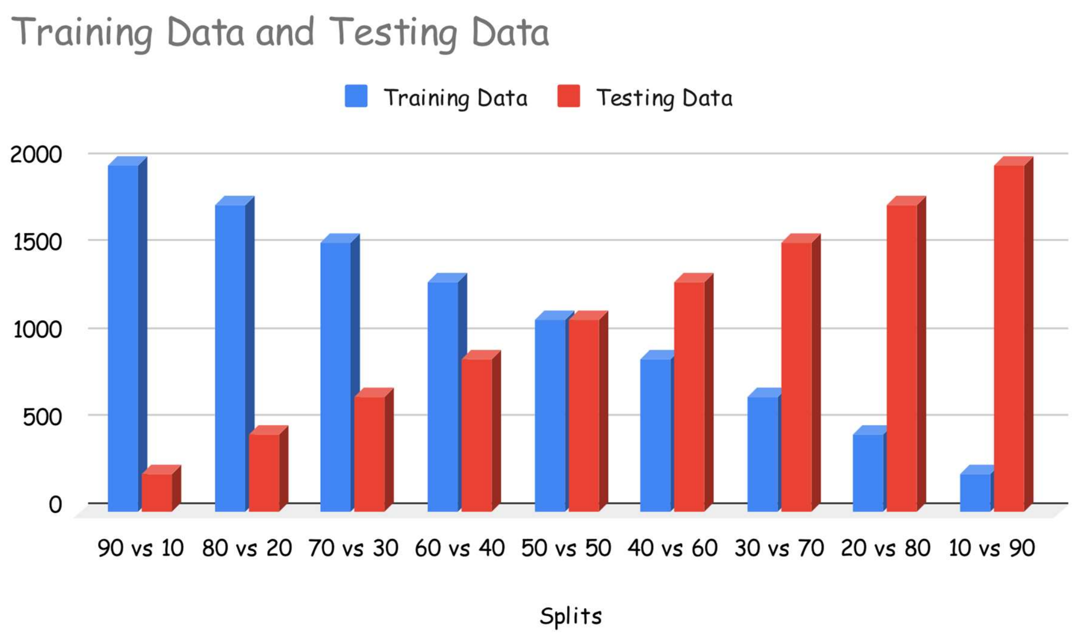
Figure-1: (a) Pie Chart Representing Value counts in the Dataset – Unbalanced (b) Pie Chart Representing Oversampled Values – Balanced – 2200 rows

Table-1 describes the distribution of complaints in the dataset that are evaluated in the proposed model for training and testing, with samples ranging from 90 vs 10 to 10 vs 90.

Table-1: Distribution of Complaints into Training and Testing

Splits	Training Data	Testing Data
90 vs 10	1980	220
80 vs 20	1760	440
70 vs 30	1540	660
60 vs 40	1320	880
50 vs 50	1100	1100
40 vs 60	880	1320
30 vs 70	660	1540
20 vs 80	440	1760
10 vs 90	220	1980

Figure-2: Bar Graph Representing Distribution of Complaints into Training and Testing



3.2 Transformers – BERT Models

BERT

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. (2018) [16] proposed BERT (Bidirectional Encoder Representations from Transformers), a language model that obtains state-of-the-art performance in a variety of NLP problems. The authors' claim is that while traditional language models are uni-directional, they do not have two-way functionality, which is useful when analyzing context in language. BERT is a bidirectional language model [16]. BERT is constructed on a multi-layer bidirectional transformer encoder. BERT employs masked language modeling, which makes it possible for the model to be bidirectional in its approach to languages [16]. It is composed of an encoder and a pooling layer. The encoder receives and processes input tokens, and the output is a sequence of vectors, while the pooler takes the output of the encoder and produces a fixed-sized vector representation of the sequence [16]. BERT is pre-trained for two objectives: Masked Language Modeling (MLM): Some input tokens are randomly replaced by a [MASK] token, and the model is trained to predict the actual token. Next Sentence Prediction (NSP): Two sentences are given as input, and the model is trained to predict if the second sentence is really following the first sentence if it is in the same text [16]. BERT is pre-trained on a large corpus of text. Book Corpus with 800 million words, English Wikipedia with 2.5 billion words, and Books and Wikipedia with 3. 3 billion words [16].
The authors also experimented with two variants of BERT: cased and uncased [16].

Cased BERT:

- The cased version of BERT is trained on the original case-sensitive text data and is used for our experiments [16].
- This means that the model is capable of processing text cases; hence, it will give different outputs between uppercase and lowercase texts [16].
- For instance, the model can be trained in such a way that it understands that 'Apple' with the capital 'A' is a proper noun, whereas 'apple' with a small 'a' is a common noun.

Uncased BERT

- The uncased version of BERT is pre-trained with text data in which all characters are in lower case [16].
- This means that the model does not differentiate between the case of the input text, i.e., uppercase and lowercase letters are considered to be the same [16].
- For instance, the model would consider "Apple" as the same token as "apple.".

BERT Architectures are two types

- BERT Base
  Base version of BERT contains 12 layers ,768 hidden size, 12 attention heads. The Parameters are 110M [16].
- BERT Large
  Large version of BERT contains 24 layers ,1024 hidden size, 16 attention heads. The Parameters are 340M [16].

In the case of BERT based sequence classification, the first thing is to load the tokenizer from the Hugging Face library's AutoTokenizer. A convenient tool that helps effectively pre-process the textual data so that it can be used to train the BERT model. Regarding the model, it is loaded as TFAutoModelForSequenceClassification. All of them are designed particularly for 11 out-put labels that are associated with various types of students' complaints.

When it comes to data collation, we use DataCollatorWithPadding from the Transformers library. This means that the sequences that are tokenized should be batched with padding meaning that this really drives the training forward.

bert-base-uncased

Table-2: Performance Metrics of BERT Base Uncased Model Across Various Train-Test Splits

Split	Training		Testing	
	Loss	Accuracy	Loss	Accuracy
90 vs 10	**0.1263**	**99.95**	**0.0721**	**100**
80 vs 20	**0.1980**	**99.83**	**0.1065**	**100**
70 vs 30	0.2090	99.87	0.1285	99.70
60 vs 40	0.2981	99.47	0.1941	98.75
50 vs 50	0.5804	97.61	0.3933	98.82
40 vs 60	0.8368	96.70	0.6081	97.80
30 vs 70	1.1552	96.34	0.9371	95.71
20 vs 80	1.7091	80.09	1.5470	84.72
10 vs 90	2.0490	50.00	2.0427	44.44

(a) (b)

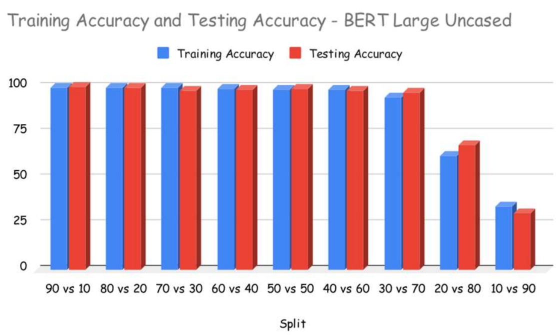
Figure-3: (a) Bar Graph Representing Training Accuracy and Testing Accuracy (b) Line Graph Representing Training Loss and Testing Loss of BERT Base Uncased Model Across Various Train-Test Splits

bert-large-uncased

Table-3: Performance Metrics of BERT Large Uncased Model Across Various Train-Test Splits

Split	Training		Testing	
	Loss	Accuracy	Loss	Accuracy
90 vs 10	**0.0405**	**99.85**	**0.0184**	**100**
80 vs 20	0.0479	99.72	0.0276	99.55
70 vs 30	0.0863	99.54	0.0679	98.18
60 vs 40	0.1934	99.31	0.1287	98.75
50 vs 50	0.2360	98.71	0.1272	99.00
40 vs 60	0.6262	98.30	0.3584	98.03
30 vs 70	0.7820	94.36	0.5096	96.82
20 vs 80	1.6940	62.27	1.6042	68.07
10 vs 90	2.0369	34.62	2.0879	31.26

(a) (b)



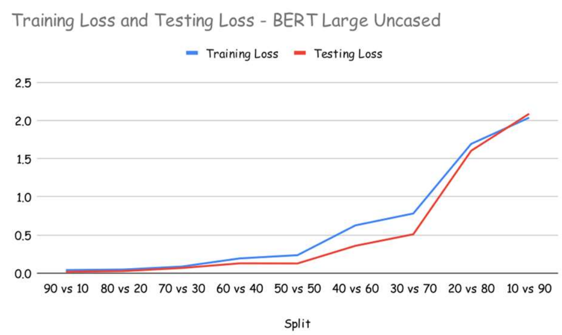


Figure-4: (a) Bar Graph Representing Training Accuracy and Testing Accuracy (b) Line Graph Representing Training Loss and Testing Loss of BERT Large Uncased Model Across Various Train-Test Splits

DistilBERT

Victor Sanh, Lysandre Debut, Julien Chaumond, Thomas Wolf. 2019 [17] proposed DistilBERT which is a much compact and fast version of BERT that has been highly adopted in many applications. Specifically, the authors trained DistilBERT with a distillation technique in which DistilBERT is trained to predict the output probabilities that BERT does on a large scale with help of both the MLM and NSP [17]. DistilBERT has the same small architecture as BERT, but with fewer layers (6 against 12) and a smaller hidden size (768 against 1024). DistilBERT being a lightweight version of BERT, is pre-trained on Book Corpus with 800 million words and English Wikipedia with 2.5 billion words [17].
DistilBERT contains 6 layers ,768 hidden size, 12 attention heads. The Parameters are 66M [17].

distilbert-base-uncased

Table-4: Performance Metrics of DistilBERT Base Uncased Model Across Various Train-Test Splits

Split	Training		Testing	
	Loss	Accuracy	Loss	Accuracy
90 vs 10	**0.1422**	**99.85**	**0.0899**	**100**
80 vs 20	0.1766	99.66	0.1131	99.55
70 vs 30	0.2638	99.54	0.1845	98.64
60 vs 40	0.3268	99.31	0.2290	99.09
50 vs 50	0.5585	99.08	0.4005	98.55
40 vs 60	0.8593	96.93	0.6593	96.06
30 vs 70	1.1567	95.88	0.9484	94.81
20 vs 80	1.7405	86.11	1.5898	88.41
10 vs 90	2.2207	49.52	2.2173	42.98

(a) (b)

Figure-5: (a) Bar Graph Representing Training Accuracy and Testing Accuracy (b) Line Graph Representing Training Loss and Testing Loss of DistilBERT Base Uncased Model Across Various Train-Test Splits

RoBERTa

Yinhan Liu, Myle Ott., [18] proposed RoBERTa, which is a variant of the popular BERT language model. RoBERTa changes with BERT: Dynamic Masking: This is different from the static manner of masking, in which the entire mask is applied throughout the process of pretraining; in RoBERTa, the mask used is a random one that changes from one sample to the next [18]. Full-sentence input: In BERT, the sentence is truncated to a certain length, but in RoBERTa, full-sentence input is used instead of a truncated sentence, which enables the model to capture long-range dependencies [18]. Larger Batch Size: RoBERTa uses a larger batch size than BERT.RoBERTa uses 8192 batch sizes, but BERT uses only 512 [18]. Longer Training: RoBERTa undergoes training for many more steps (125K) than BERT (50K) [18]. Different Optimizer: RoBERTa uses the same optimizers as Adam and a distinct learning rate schedule as BERT [18]. RoBERTa is pre-trained on a large corpus of text. Book Corpus with 800 million words, English Wikipedia with 2.5 billion words, OpenWebText with 38 billion words and CC-News with 63 billon words [18].
RoBERTa Architectures are two types

- RoBERTa Base
  Base version of RoBERTa contains 12 layers ,768 hidden size, 12 attention heads. The Parameters are 335M [18].
- RoBERTa Large
  Large version of RoBERTa contains 24 layers ,1024 hidden size, 16 attention heads. The Parameters are 550M [18].

roberta-base

Table-5: Performance Metrics of RoBERTa Base Model Across Various Train-Test Splits

Split	Training		Testing	
	Loss	Accuracy	Loss	Accuracy
90 vs 10	0.0514	99.85	0.0677	98.64
80 vs 20	**0.0629**	**99.72**	**0.0446**	**99.55**
70 vs 30	0.0678	99.87	0.0433	99.55
60 vs 40	0.0916	99.92	0.0665	99.43
50 vs 50	0.1297	99.63	0.0749	99.82
40 vs 60	0.2160	98.75	0.1314	98.48
30 vs 70	0.4072	98.63	0.2741	96.88
20 vs 80	1.0861	93.52	0.7323	95.06
10 vs 90	2.1887	41.83	1.9992	59.04

(a) (b)

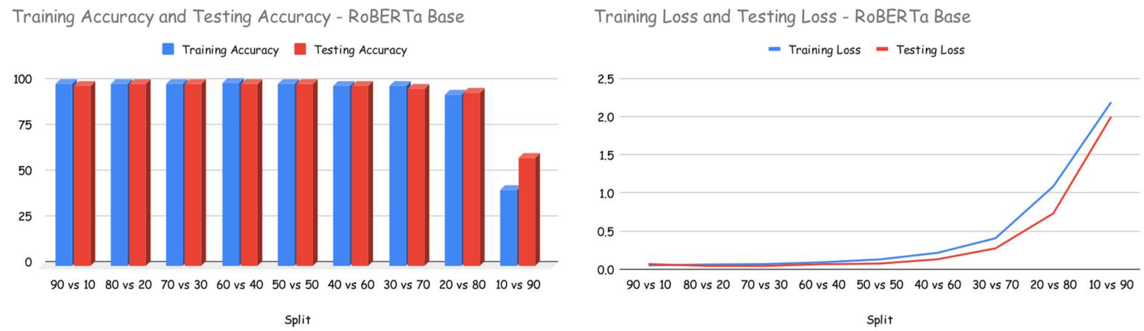


Figure-6: (a) Bar Graph Representing Training Accuracy and Testing Accuracy (b) Line Graph Representing Training Loss and Testing Loss of RoBERTa Base Model Across Various Train-Test Splits

roberta-large

Table-6: Performance Metrics of RoBERTa Large Model Across Various Train-Test Splits

Split	Training		Testing	
	Loss	Accuracy	Loss	Accuracy
90 vs 10	**0.0288**	**99.75**	**0.0074**	**100**
80 vs 20	0.0410	99.26	0.0412	98.86
70 vs 30	0.0277	99.67	0.0418	99.24
60 vs 40	0.0348	99.77	0.0328	99.09
50 vs 50	0.0606	99.36	0.0341	99.00
40 vs 60	0.0815	98.98	0.0688	98.71
30 vs 70	0.1432	98.32	0.0973	98.12
20 vs 80	0.5218	95.14	0.2261	96.99
10 vs 90	2.3375	17.31	2.2509	28.89

(a) (b)

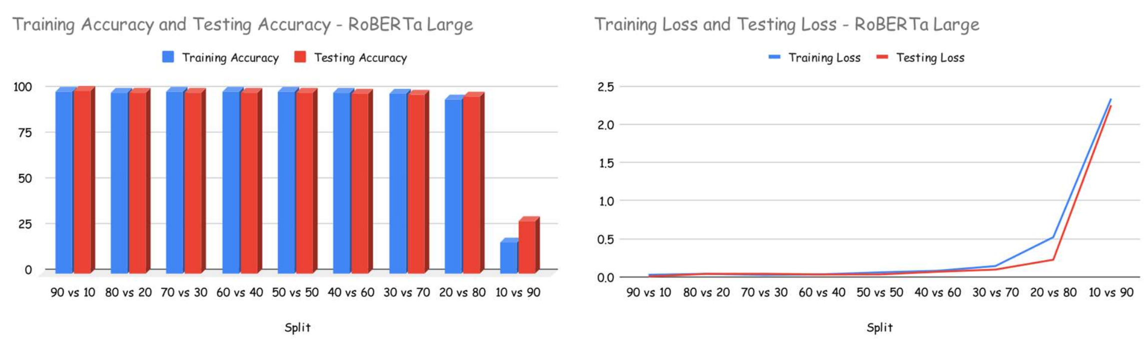


Figure-7: (a) Bar Graph Representing Training Accuracy and Testing Accuracy (b) Line Graph Representing Training Loss and Testing Loss of RoBERTa Large Model Across Various Train-Test Splits

ALBERT

Zhenzhong Lan, Mingda Chen.,[19] presented ALBERT, which is a version of the language model BERT that has similar performance as BERT but uses fewer parameters [19]. ALBERT changes with BERT: Factorized Embedding Parameterization: ALBERT also employs Factorized Embedding Parameterization for better generalization abilities of the model in question by reducing the number of parameters present in the embedding layer [19]. Cross-layer Parameter Sharing: ALBERT shares parameters across layers, which results in a larger number of parameters, making the model more efficient [19]. Sentence Order Prediction (SOP): ALBERT employs a novel pretraining task called Sentence Order Prediction (SOP), which involves the prediction of the order of two successive sentences [19]. Reduced Model Size: ALBERT is a lighter model than the BERT, with 12 layers and a hidden size of 128 [19]. Albert is pre-trained on a large corpus of text. Book Corpus with 800 million words, English Wikipedia with 2.5 billion words, OpenWebText with 38 billion words, and CC-News with 63 billion words [19].
ALBERT Architectures are two types

- ALBERT Base
  Base version of ALBERT contains 12 layers ,768 hidden size, 12 attention heads. The Parameters are 117M [19].

- ALBERT Large
  Large version of ALBERT contains 24 layers ,1024 hidden size, 16 attention heads. The Parameters are 223M [19].

albert-base-v2

Table-7: Performance Metrics of ALBERT Base v2 Model Across Various Train-Test Splits

Split	Training		Testing	
	Loss	Accuracy	Loss	Accuracy
90 vs 10	**0.1207**	**99.49**	**0.0880**	**100**
80 vs 20	0.0929	99.43	0.0989	99.09
70 vs 30	0.2317	99.15	0.1830	98.03
60 vs 40	0.2737	98.93	0.2131	98.07
50 vs 50	0.2807	98.44	0.2549	96.91
40 vs 60	0.8535	91.93	0.6859	94.47
30 vs 70	1.2533	82.62	1.1429	83.77
20 vs 80	1.6707	62.27	1.6156	67.44
10 vs 90	2.0208	36.06	2.1169	34.34

(a) (b)

Figure-8: (a) Bar Graph Representing Training Accuracy and Testing Accuracy (b) Line Graph Representing Training Loss and Testing Loss of ALBERT Base Model Across Various Train-Test Splits

albert-large-v2

Table-8: Performance Metrics of ALBERT Large v2 Model Across Various Train-Test Splits

Split	Training		Testing	
	Loss	Accuracy	Loss	Accuracy
90 vs 10	**0.0820**	**99.59**	**0.0503**	**100**
80 vs 20	0.0876	99.20	0.0912	98.18
70 vs 30	0.1798	98.96	0.1369	99.39
60 vs 40	0.1411	99.47	0.1232	98.75
50 vs 50	1.6319	62.78	1.4588	70.36
40 vs 60	1.1830	74.55	0.9443	84.62
30 vs 70	0.8470	91.46	0.6857	92.08
20 vs 80	1.9113	43.06	1.7981	45.74
10 vs 90	2.1443	37.02	2.1529	36.11

(a) (b)

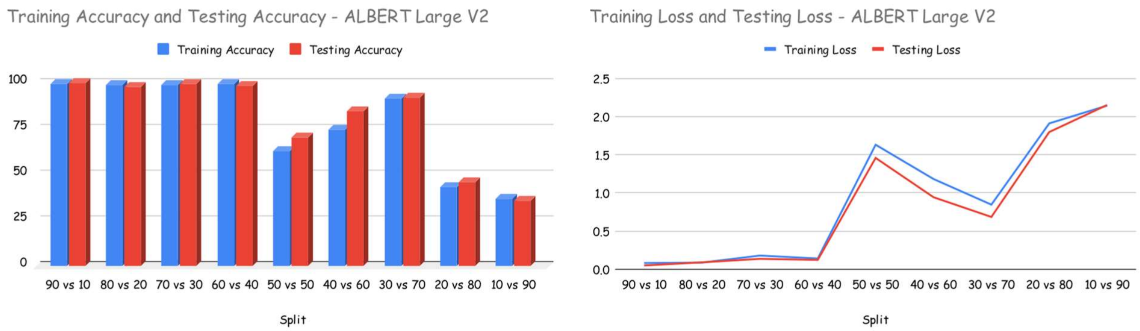


Figure-9: (a) Bar Graph Representing Training Accuracy and Testing Accuracy (b) Line Graph Representing Training Loss and Testing Loss of ALBERT Large Model Across Various Train-Test Splits

3.3 Full Stack Implementation

Frontend and Backend Development:

The home page, admission page, departments page, admin page, student page, and alumni page of the system were designed under the Frontend Framework (ReactJS).

Python Flask framework was used for backend logic to optimize the flow of data between the frontend and backend components as well as correct authentication of the company interactions existing in between.

Database Design and Implementation:

The detailed student, alumni, department, semester results complaint, and message database were structured with the help of a relational database model. Data management, which is a performance criterion in our case, was implemented with SQL.

User authentication and authorization:

For administrators, students, and alumni, user authentication and authorization were successfully incorporated. This also helped to keep the information secure, and the access rights were properly managed depending on the roles of the users.

Integration of BERT model in full stack framework

Only students who log in with the correct credential related to the college that is connected to the database can submit the complaint. The frontend collects the student's name, roll number, and department, and any complaint or feedback that is to be submitted by the student is passed to the backend. In the backend, the revised complaint or feedback is given to the BERT model, which categorizes it into one of the predetermined categories, and the details with the label, i.e., the category given by the model, are stored in the database. This student can see it under the My Complaints tab.

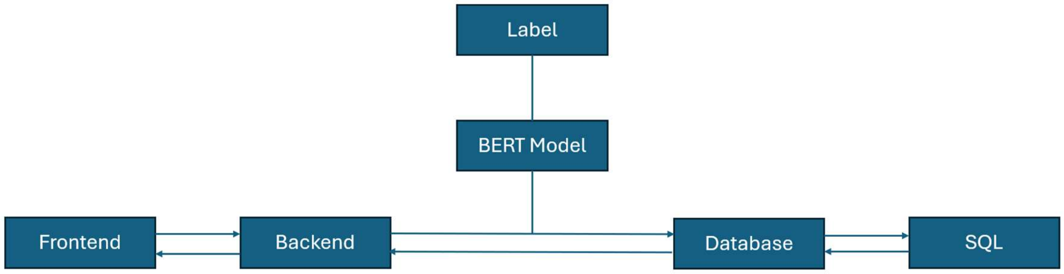
Word Cloud in Full Stack Framework

F. Heimerl, S. Lohmann, S. Lange and T. Ertl, [20] presented Word Cloud Explorer, an additive instrument to work with textual data employing word clouds. Word clouds are data sets that are portrayed in the form of pictures, with the size of the word in the picture being directly proportional to the importance of the word in the data set. Through word clouds, for instance, the tool enables users to navigate through the textual data in an interactive way [20].

The complaints so far collected are retrieved from the database to form the word cloud.

The Workflow

Figure-10: Describes the work flow the proposed model in integration with full stack framework



4. Results

4.1 Results of Experimentation - Text Classification - BERT:

We have evaluated multiple transformer models for the task of text classification, which involves categorizing student complaints. The proposed models are BERT base uncased, BERT large uncased, DistilBERT uncased, RoBERTa base, RoBERTa large, ALBERT base, and ALBERT large. The performance of the models is evaluated based on training accuracy and testing accuracy. The results are illustrated in the below table.

Table-8: Accuracy Comparison of Various Transformer Models on 90-10 and 80-20 Train-Test Splits

Model	Accuracy			
	90 vs 10 Split		80 vs 20 Split	
	Training	Testing	Training	Testing
BERT Base Uncased	**99.95**	**100**	**99.83**	**100**
BERT Large Uncased	99.85	100	99.72	99.55
DistilBERT Base Uncased	99.85	100	99.66	99.55
RoBERTa Base	99.85	98.64	99.72	99.55
RoBERTa Large	99.75	100	99.26	98.86
ALBERT Base v2	99.49	100	99.43	99.09
ALBERT Large v2	99.59	100	99.20	98.18

(a) (b)

Figure-10: (a) Bar Graph Representing Accuracy Comparison of Various Transformer Models on 90-10 Train-Test Split (b) Bar Graph Representing Accuracy Comparison of Various Transformer Models on 80-20 Train-Test Split

Observation: BERT Base Uncased Model was found to give the highest validation accuracy of 100% on both 90-10 & 80-20 train-test splits thus making it the most effective in this classification task. So we integrate this model in the full stack framework to provide real time processing.

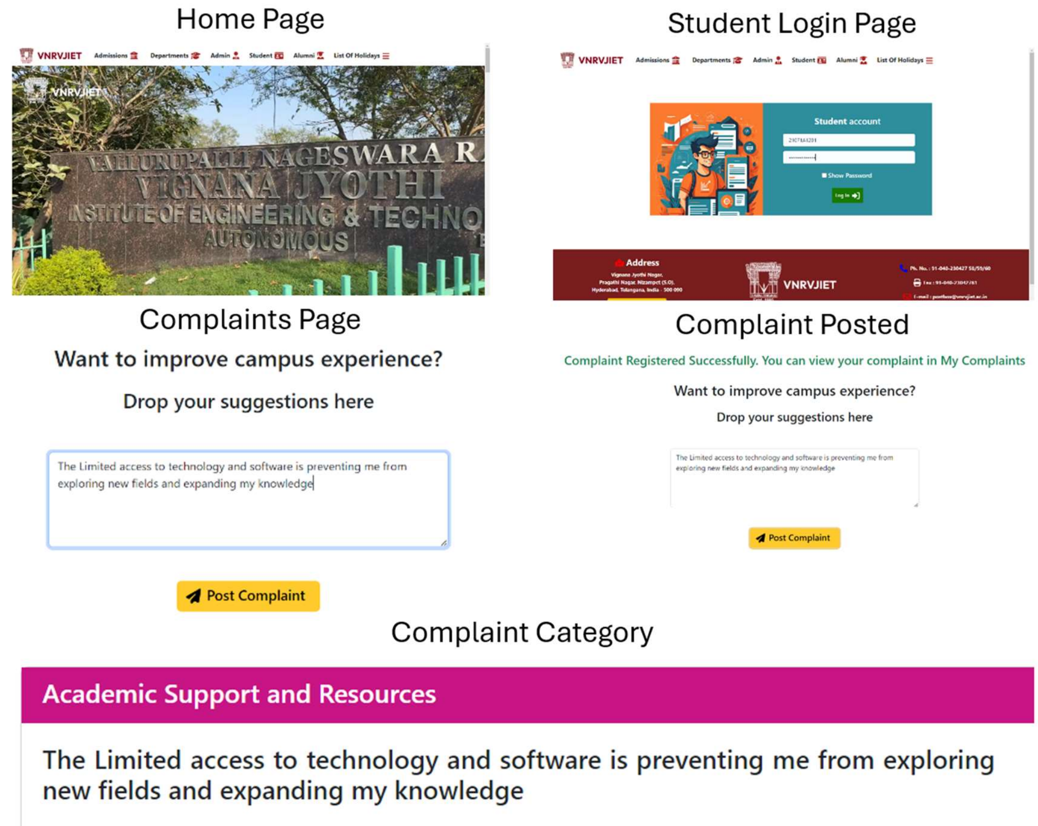


Figure-11: Home Page – The Landing Page of the CRM Website. Student Login Page – The Login Page where students can login using valid credentials. Complaints Page – The Complaints Page that can be accessed after successful login. Complaint Posted – Student getting Acknowledgement that the complaint registered successfully. Complaint Category – The Label Which is given by the proposed BERT model

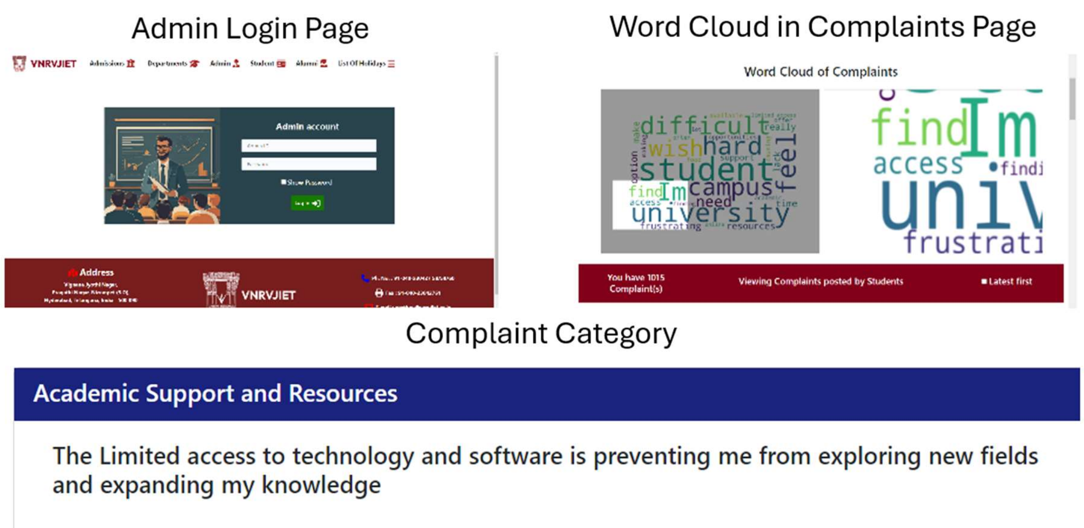


Figure-11: Admin Login Page – The Login Page where admins can login using valid credentials.
Word Cloud in Complaints Page – The Word Cloud in Complaints Page that can be accessed after successful login. The admin can view all the complaints that are stored in the database along with Compliant Categories.
Complaint Category – The Label Which is given by the proposed BERT model.

Figure-12: The Word Cloud that is visible to the admins.

5. Conclusion

In this work, we focused on the use of modern NLP tools and especially BERT-based models for the classification of the students' complaints in a college setting. Therefore, due to the usage of the BERT model, which provides deep contextual analysis of the feedback, the performance of the system is remarkably higher compared to the traditional one, which includes ACA and sentiment analysis of the students' feedback with the complexity of the recognition.

In our experiments, we showed that the BERT Base Uncased model had the highest validation accuracy of 100% for both 90-10 and 80-20 splits in training and testing sets, which indicates that this model has the highest effectiveness of the classification among all the analyzed transformer models. The integration of this model into a full-stack application gives a strong and flexible complaint management system through which students can submit complaints anytime, not just at the end of the week or month, greatly addressing a major weakness of current systems.

The use of the described system in a real-life learning environment improves not only the institutions' timeliness and efficiency in addressing students' issues but also contributes to the satisfaction level of learners. The word cloud derived and categorized from the complaints contains useful data to understand the persisting concerns and to adapt teaching approaches and student support further.

Finally, there is also the role of adaptability in the proposed system as part of the whole mechanism of integrating state-of-the-art NLP models into CRM systems as a as a means of enriching the spheres of communication management in educational institutions, student relations management, and, as a result, the promotion of effective implementation of educational activities. The future works can be continued with other kinds of feedback, can be tried with modern transformer models, and the system can be implemented in a larger number of institutions.

6. References

[1]. Seeman, Elaine D., and Margaret O'Hara. "Customer relationship management in higher education: using information systems to improve the student-school relationship." Campus-wide information systems 23.1 (2006): 24-34.

[2]. Alderman, L., Towers, S., & Bannah, S. (2012). Student feedback systems in higher education: a focused literature review and environmental scan. Quality in Higher Education, 18(3), 261–280. https://doi.org/10.1080/13538322.2012.730714

[3]. Berk, R. A. (2005) Survey of 12 Strategies to Measure Teaching Effectiveness. International Journal of Teaching and Learning in Higher Education, 17, 48-62.

[4]. Mandouit, Luke. (2018). Using student feedback to improve teaching. Educational Action Research. 26. 1-15. 10.1080/09650792.2018.1426470.

[5]. T. Shaik et al., "A Review of the Trends and Challenges in Adopting Natural Language Processing Methods for Education Feedback Analysis," in IEEE Access, vol. 10, pp. 56720-56739, 2022, doi: 10.1109/ACCESS.2022.3177752.

[6]. Cardona, Tatiana & Cudney, Elizabeth & Hoerl, Roger & Snyder, Jennifer. (2020). Data Mining and Machine Learning Retention Models in Higher Education. Journal of College Student Retention: Research, Theory & Practice. 25. 152102512096492. 10.1177/1521025120964920.

[7]. K. Z. Aung and N. N. Myo, "Sentiment analysis of students' comment using lexicon based approach," 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), Wuhan, China, 2017, pp. 149-154, doi: 10.1109/ICIS.2017.7959985.

[8]. L. Balachandran and A. Kirupananda, "Online reviews evaluation system for higher education institution: An aspect based sentiment analysis tool," 2017 11th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), Malabe, Sri Lanka, 2017, pp. 1-7, doi: 10.1109/SKIMA.2017.8294118.

[9]. Berardinelli, Nabeela & Gaber, Mohamed & Haig, Ella. (2013). SA-E: Sentiment Analysis for Education. Frontiers in Artificial Intelligence and Applications. 255. 10.3233/978-1-61499-264-6-353.

[10]. Wook, Muslihah, et al. "Opinion mining technique for developing student feedback analysis system using lexicon-based approach (OMFeedback)." Education and Information Technologies 25 (2020): 2549-2560.

[11]. Dake, Delali Kwasi, and Esther Gyimah. "Using sentiment analysis to evaluate qualitative students' responses." Education and Information Technologies 28.4 (2023): 4629-4647.

[12]. Shaik, Thanveer, et al. "Sentiment analysis and opinion mining on educational data: A survey." Natural Language Processing Journal 2 (2023): 100003.

[13]. Shah, Mahsood & Pabel, Anja. (2019). Making the student voice count: using qualitative student feedback to enhance the student experience. Journal of Applied Research in Higher Education. ahead-of-print. 10.1108/JARHE-02-2019-0030.

[14]. Abaidullah, Anwar Muhammad, Naseer Ahmed, and Edriss Ali. "Identifying Hidden Patterns in Students" Feedback through Cluster Analysis." International Journal of Computer Theory and Engineering 7.1 (2015): 16.

[15]. Altrabsheh, N., Cocea, M., Fallahkhair, S. (2014). Learning Sentiment from Students' Feedback for Real-Time Interventions in Classrooms. In: Bouchachia, A. (eds) Adaptive and Intelligent Systems. ICAIS 2014. Lecture Notes in Computer Science(), vol 8779. Springer, Cham. https://doi.org/10.1007/978-3-319-11298-5_5

[16]. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:1810.04805.

[17]. Victor Sanh, Lysandre Debut, Julien Chaumond, Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108

[18]. Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692

[19]. Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut. 2019. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. arXiv preprint arXiv:1909.11942

[20]. F. Heimerl, S. Lohmann, S. Lange and T. Ertl, "Word Cloud Explorer: Text Analytics Based on Word Clouds," 2014 47th Hawaii International Conference on System Sciences, Waikoloa, HI, USA, 2014, pp. 1833-1842, doi: 10.1109/HICSS.2014.231.