National College of Ireland

Network-based social media sentimental analysis on twitter

MSc Research Project
Artificial Intelligence

Phanisai kotha
Student ID: 23415754

School of Computing
National College of Ireland

Supervisor: Rejwanul Haque

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Phanisai kotha
Student ID:	23415754
Programme:	Artificial Intelligence
Year:	2025
Module:	MSc Research Project
Supervisor:	Rejwanul Haque
Submission Due Date:	28/01/2026
Project Title:	Network-based social media sentimental analysis on twitter
Word Count:	9150
Page Count:	25

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Phanisai kotha
Date:	28th January 2026

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Network-based social media sentimental analysis on twitter

Phanisai kotha
23415754

Abstract

The purpose of this research is to determine the level of efficiency of the classical machine-learning and the transformer-based models in large-scale Twitter sentiment analysis with the Sentiment140 Dataset. This project aims to address the issues pertaining to the lack of comparative studies focusing on the traditional and transformer models, the growing dependency on social media for analytics, and the understanding of the audience‘s opinion response. A thorough and robust preprocessing application was created, and subsequent to the development of Logistic Regression and the Support Vector Machine, the fine-tuned BERT model was applied. These models were tested and evaluated through metrics of efficiency and accuracy, precision, recall, F1 score, with confusion matrices and ROC/AUC analysis. BERT was proven to be the most accurate with a score of 0.84 and the most outstanding discrimination ability, though the classical models provided a more than competent baseline with a higher degree of efficiency. In order to offer real-time sentiment classification and support the exploration of an interactive knowledge graph for users, a Streamlit application was created. The findings of the research show a strong trade-off between accuracy and efficiency and should help users identify models to improve efficiency based on available resources and performance needs.

Keywords: Sentiment Analysis, Social Media, Twitter, Natural Language Processing(NLP), Term Frequency–Inverse Document Frequency (TF-IDF) , Bidirectional Encoder Representations from Transformers(BERT), Logistic Regression(LR), Support Vector Machine(SVM), Natural Language Toolkit(NLTK), Receiver Operating Characteristic / Area Under the Curve (ROC/AUC)

1 Introduction

Out of all online platforms that let users voice their opinions, social media is one of the largest in size and the quickest to react to events as they unfold in the real world. During the years that social media has existed, twitter has been one of the foremost social media platforms for users to voice their opinions and for businesses and other institutions to analyze. The informality of posts as well as the unique vocabulary and syntax, as well as the attitudes and sentiments of the posters, has been of interest to several branches of social science. The field that has processed these texts with the greatest volume and an enormous plurality of text to extract sentiment has been text mining. Training computational models to classify nuances in sentiment, has also been very interesting academically. Most notably, this interest has resulted in the creation of

the field of sentiment analysis in computer science. Unfortunately, as the years go by and more and more text is informally posted on platforms such as Twitter, the twitter data itself is evolving. And this is what continues to present challenges to the computational models that were created in response to the interest in sentiment analysis. For this reason, accurately classifying sentiment in several pieces of text, is especially challenging as time goes by.

The research is guided by the following primary question:

1.1 Research Question

Which sentiment classification method Logistic Regression, Support Vector Machines or a transformer-based BERT model provides the most effective balance between accuracy and computational efficiency when applied to Twitter sentiment analysis? To address this research question, the project establishes the following objectives:

- To preprocess and prepare the Sentiment140 dataset, including cleaning, normalisation and text representation.
- To train and evaluate multiple sentiment classification models Logistic Regression, SVM and BERT using consistent metrics.
- To compare model performance based on accuracy, precision, recall, F1-score and confusion matrices.
- To develop a simple, user-friendly Streamlit web application that enables real-time sentiment prediction.
- To provide recommendations on appropriate model selection based on performance cost trade-offs.

These objectives will be evaluated through quantitative testing, where each model is trained and assessed on the same test set. Success criteria include achieving reproducible performance metrics, clearly demonstrating differences between modelling approaches, and providing a justified conclusion regarding suitability for real-world applications.

2 Related Work

2.1 Classical Machine Learning Approaches to Twitter Sentiment Analysis

An analysis of sentiment on Twitter during its earlier stages utilized basic machine-learning techniques alongside engineering features such as TF-IDF and bag-of-words techniques. One of the first works in this field is Gautam and Yadav (2014), where traditional learning methods such as Naive Bayes, SVM, and Maximum Entropy are proven to achieve reasonably good results on well-processed Twitter datasets. His approach to sentiment detection was the extraction of semantic features, in this case adjectives, with the use of resources such as WordNet. This paper has contributed in that the models were able to capture sentiment in context due to the added semantic features; however, the use of handcrafted features immobilized the models by inefficient scaling, as the system

needed a lot of manual linguistic work as well as domain-specific resources. Ahmed et al. (2023) is another more recent classical work centered on testing several machine-learning techniques and TF-IDF on a Twitter dataset. Their study affirmed that the Logistic Regression and SVM are indeed effective if combined with good feature extraction and preprocessing. The classifiers were able to capture important lexical items due to TF-IDF as it helps in deweighting sentiment neutral words that tend to occur frequently. Although the study documented accurate results, the analysis did not include modelling the contextual dependencies of words, and this is a primary limitation of this research and all research relying on a TF-IDF (Term Frequency – Inverse Document Frequency) model. Words contained in a tweet have a sentiment, depending on context, were sarcastic or negated form. These characteristics traditional models seldom capture. It is this limitation that the authors acknowledge in the study, and recommend moving to contextual deep learning to inform other model systems to overcome the restrictions with linear classifiers. The contribution of classical machine-learning in this instance is to show that a simple, efficient method is often adequate, and that they have made a pragmatic, reasonable contribution in the study of Twitter data. Yet, the lack of contextual understanding, the reliance on feature-engineering, the difficulty of the model to generalise across problem domains all combine to show limitations, and contributed to the need to focus on deep learning. In particular, deep learning that is able to capture semantic and syntactic patterns is preferred.

2.2 Deep Learning and Hybrid Approaches: Overcoming Limitations of Classical Models

As the field of sentiment analysis grew, scholars began to implement deep learning models, which automatically pull out hidden features from within the text. Maurya and Jha (2024) made a remarkable contribution in such shift of trends, for example, by using a hybrid of TF–IDF based clustering, SVM classifiers, and deep neural networks. Their work proved that a hybrid of models (traditional and deep) could perform better, and be more accurate, than using classical models alone. The strength of this work is in the well structured experiments and showing that the use of unsupervised clustering could enhance supervised learning by finding the hidden sentiment in the noisy Twitter streams. Still, in spite of the increased accuracy, the models had great training times and new complexities that made the models less scalable to larger datasets like Sentiment140. The work of Sharma et al. (2025) is of the same move and focus, with the building block of a fusion model made of a BiGRU network with a DistilBERT encoder to perform sentiment analysis on COVID-19 tweets. Their model finesse is the use of both types of models, sequential and transformer and deep BiGRU for context and embeddings. The authors have shown how the integration of contextual representation with recurrent neural frameworks enhances the performance of sentiment classification tasks, especially those that are sarcastic, uncertain, and ambiguous. This study's reliance on Twitter data, especially Saudi tweets, showcases the versatility of applied deep learning. However, the integration of two different models also raised computational concerns that may hinder its real-time dev deployment and use in time-critical scenarios. Talukder et al. (2025) advanced this hybrid approach by incorporating class balancing and Differentially-Private deep learning frameworks to assess tweets related to COVID-19. This study addressed what has been considered the underlining challenges of sentiment analysis research. The employment of balancing strategies including SMOTE and cost-sensitive

learning significantly enhanced the predictive performance on the minority class of the sentiment. Despite the study showcasing well-founded methodology in the resolution of the imbalanced dataset, the use of over-sampling, especially the synthetic manner, has its risks which revolve around merging the dataset too tightly and may misrepresent the actual sentiments embedded within the data.

In hybrid models, the computational cost of the proposed models theory is significantly pronounced, however this is the models have been proposed might provide to out of the deep learning models has been layer based. Their most pronounced and strongest disadvantages include, but are not limited to, the system architecture that has been proposed. As a result of these limitations, transformer-based models of sentiment analysis have reached a state-of-the-art level.

2.3 Transformer-Based Models: The State of the Art in Sentiment Analysis

Transformer architectures, it is now possible to do NLP while understanding contextual relationships between different parts in the text. As shown by Bello et al. (2023), the positive impacts of fine-tuning BERT for Twitter sentiment classification are considerable when compared to baselines in CNN, RNN, and BiLSTM. The self-attention mechanisms of the classical models allow the self-attention mechanisms of the classical models. Capture long-range and subtle criticisms the classical models miss. One major impact of this study was to compare the sentiment analysis models of classical neural architectures to self-attention neural architectures and frame the former as superior. The fine-tuning of BERT was a big impact in relation to costs. The authors are not as self-sustained as automated large-scale training. He (2024) has proposed fine-tuning RoBERTa and BERT models in parallel as a joint dual classifier. The positive impact of this paired approach is to leverage the full underlying self-attention mechanism of both models. The impact of model accuracy increased to a considerable number compared to the fine-tuned models evaluated in isolation. The approach is methodologically exhaustive in experimental validation across multiple Twitter datasets. The impact is it has significant resource consumption. Relatively low processing hardware limitations. Completing the above is Sahoo et al. (2023) the increased demand for social media sentiment analysis. The success of multilingual BERT implementations and domain-specific ones like BERTweet and RoBERTuito has been assessed in one of the most thorough surveys of BERT variants done most recently in 2023. He showed that in the case of Tweets, domain-adapted models beat the more generalized ones. The most important aspect of this study is cross-domain comparisons, and it postures the architectural design of the models and the corpus in pretraining to have major impacts on performance. He, however, is model-specific and it cannot be said that domain-specific BERT models generalize to contexts outside the training corpus, which is a limitation. Further insight has been provided by Enagandula (2024), who examined unsupervised BERT models and their performance on general text sentiment tasks. Enagandula's study acknowledged the contextual understanding by BERT while also noting that there are criticisms that are associated with sensitivity to decisions made on pre processing and overfitting especially on smaller sets. Although the study defends the positive aspects that BERT has, it is lacking more sophisticated comparisons with other recently released variants of transformers. Generally, across most of the literature there is a consensus that within the literature, the libraries of models also demonstrate superior performance especially with regards to alternatives that are

more classical and also deep learning. The major limiting factors such models present are the long training time and the highly specialized hardware they need, which makes the analysis of the trade-offs between efficiencies and accuracy relevant to this project.

2.4 Synthesis of Literature and Identification of Research Gaps

Evolution can be seen in most literature from classical modelling to deep learning, culminating in transformer designs. Logistic Regression and SVM, still classical Moul data, are still important and possess efficiency and interpretability as valuable assets, even if they are limited in context and sentiment. Complex deep learning integrations tend to perform better, however, they tend to be complex hyperparameter tuning of high-computational models and overfitting. Transformers are the best models, as they are the most accurate and provide bidirectional context, however, they are highly computationally expensive and take a long time to train and they generalise poorly, sometimes, with regard to the the information domain of the tasks. Several gaps still exist in the literature even with these models. Many literature is devoid of the equivalent, equivalent and equivalent supervised, controlled study, of classical models, and hybrid models, as well as transformer models, all trained, processed using the same pipelines, datasets and evaluation metrics. Transformer fine tuning is highly domain specific and this makes cross modelling studies allusion. Furthermore, the computational models highly paradigm shifts of each modelling stream are underspecified, although they are highly important, in practice. Lastly, very few studies exist, regarding the use of a very practical model evaluation, in the form of a real-time prediction interface, to assess the usability of the models in practice. All these factors from the literature to the modelling paradigms form the baseline of a comparative modelling study to evaluate the paradigm modelling stream in equalised terms and a standardised, large twitter stream dataset. This research continues by trained and compared systematically the Logistic Regression, SVM, and BERT models on the Sentiment140 dataset with an identical experimental method while also analysing the computational resource efficiency and the deployment of the models in Streamlit for practicality assessment.

Table 1: Summary of Recent Studies

Study	**Dataset**	**Models**	**Key Findings**	**Strengths**	**Limitations**
Jonnala et al. (2025)	Large-scale Twitter dataset	Hybrid deep learning + machine learning model	Hybrid architecture significantly improved sentiment classification performance	Demonstrates synergy between ML and DL; strong validation on large dataset	High computational cost; complex model design
Sharma et al. (2025)	COVID-19 tweets	BiGRU + DistilBERT fusion model	Fusion model achieved superior accuracy by combining contextual embeddings with sequential learning	Strong contextual understanding; effective on domain-specific tweets	Expensive training; heavy computational overhead

Maurya and Jha (2024)	Twitter data	Hybrid lexicon + TF-IDF + clustering + SVM	Hybrid pipeline improves accuracy over classical models alone	Integrates multiple representations; good noise handling	High complexity; longer training process
He (2024)	Twitter (multi-topic datasets)	Dual-Joint RoBERTa–BERT classifier	Joint transformer architecture outperformed single-model baselines	Innovative architecture; robust cross-dataset performance	Requires high GPU resources; limited real-time feasibility
Sahoo et al. (2023)	Multiple multilingual Twitter datasets	Comparative study of BERT, RoBERTuito, mBERT	Domain-specific BERT variants performed best	Comprehensive cross-model evaluation	Generalisability varies across pretrained domains
Talukder et al. (2025)	COVID-19 sentiment dataset	Hybrid DL + class balancing techniques	Balanced model improved performance on minority classes	Addresses dataset imbalance; high accuracy	Synthetic balancing methods may introduce noise
Ahmed et al. (2023)	Twitter dataset	TF–IDF + LR, SVM, XGBoost	Logistic Regression and SVM performed best with TF–IDF	Strong classical baseline; computationally efficient	No contextual modelling; limited accuracy ceiling
Bello et al. (2023)	Twitter sentiment	Fine-tuned BERT; compared with CNN, RNN	BERT outperformed all traditional DL models	Strong evidence for transformer dominance	Expensive training; requires careful preprocessing
Enagandula (2024)	General text sentiment	BERT pretrained model	BERT captures contextual cues better than ML baselines	Shows BERT robustness across domains	Limited cross-domain evaluation; simplified dataset
Singh et al. (2022)	Twitter data	TF-IDF + classical ML models	SVM best performer among classical approaches	Validates TF–IDF baselines; simple and fast	No deep learning comparison; lacks contextual insight
Srivastava et al. (2019)	General Twitter datasets	Lexicon-based + ML techniques	Demonstrated usefulness of lexicons in early Twitter sentiment	Good early groundwork; highlights challenges	Outdated compared to transformers; weaker accuracy
Gautam and Yadav (2014)	Twitter dataset	SVM, Naïve Bayes + semantic analysis	WordNet features improved sentiment performance	Shows benefit of semantic features	Hand-crafted features do not scale; older dataset

3 Methodology

3.1 Dataset Acquisition and Sampling Strategy

To obtain a sense of the sentiment behind the tweets, we first acquired the Sentiment140 dataset, the largest available dataset with over 1.6 million tweets and the sentiment behind them classified as positive, negative and neutral. It is also the most widely used dataset in the literature due to it being large and having a real-world Twitter data collection Gautam and Yadav (2014). The dataset was fetched with Latin 1 encoding and a data cleaning process was performed where we remove duplicate rows and rows where the sentiment was missing. After removing roughly 82,567 duplicates, we were left with a dataset of around 1.51 million rows. Following the sentiment classification literature, we removed neutral tweets Singh et al. (2022). We were left with positive and negative tweets . To ensure we would not overfit to the small dataset we also applied a strategy where we sampled the dataset without replacement. We took a sample of 20% of the data in each sentiment as to ensure we had the same sentiment composition in our sample as we had in the complete dataset. In total, we had roughly 303,487 tweets in our sample dataset after applying the stratified sampling. The three models like Logistic Regression, Support Vector Machine, and BERT, all underwent training and assessment only with this particular stratified segment, which was further split into 80% for training and 20% for validation and using a pre-determined random number seed. This choice was influenced by studies showing that transformer-based models like BERT have high computational costs and sometimes lack the infrastructure for fine-tuning on complete Twitter datasets Bello et al. (2023). All models using the same stratified subsets meant that comparisons between classical and transformer models were as fair and as transparent as possible.

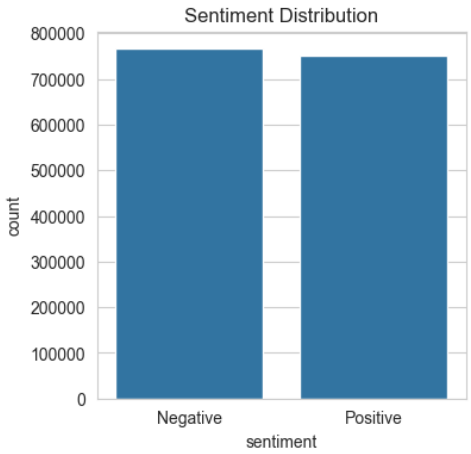


Figure 1: Sentiment distribution

3.2 Text Preprocessing and Data Transformation

The first step was to transform the initial set of tweets to a uniform and analyzable form. In this step the focus should be placed on the identification and removal of noisy data since it has proven to be damaging to the model results Ahmed et al. (2023). A tailor custom built Python regular expression function was created to complete the following cleaning tasks: removal of URLs, mentions of users, hashtags, digits, emojis, any and

all punctuation, and other extraneous characters. To further standardize the data, all text was converted to lower case and multiple consecutive whitespace characters were replaced with single spaces. The final product of this step was a single column containing the now normalised text which was to serve the purpose of preliminary data analysis and model training. The contribution of the NLTK library of Python to the work presented in this paper constitutes the implementation of explodatory data analysis and feature-engineering. The other components of the work including the training of ML models do not depend on NLTK tokens. Features demanded the utilization of NLTK tokenization to produce various mathematical statistics around the tokens such as frequencies and ratios of the overall count of words in the text and various distributions of the tokens around the count of stop words including the tokens. The classical models themselves make use of the NLTK library as the stopword removal process is handled through the inbuilt functionality of the Scikit learn library and in particular its TF IDF vectoriser. The BERT model does not utilize NLTK tokens since it's WordPiece tokenizer works directly on the cleaned text to generated subword units that are suitable for being processed by the transformer. In addition to the aforementioned lexical cleaning, other linguistic and behavioral features were intentionally added to the base analytical data table. These features were character count, word count, average word length, hashtag count, mention count, unique word ratio, as well as temporal features like hour, weekday, and month of posting derived from the timestamp metadata. These features help characterize and cluster communication and user behaviour and have previously been demonstrated to aid in a more sophisticated interpretation of sentiment phenomena in the Twitter data Srivastava et al. (2019). These transformations resulted in a dataset that was not only clean and expressive, but one that could be subjected to traditional TF IDF vectorisation, as well as to tokenisation through the use of transformers.

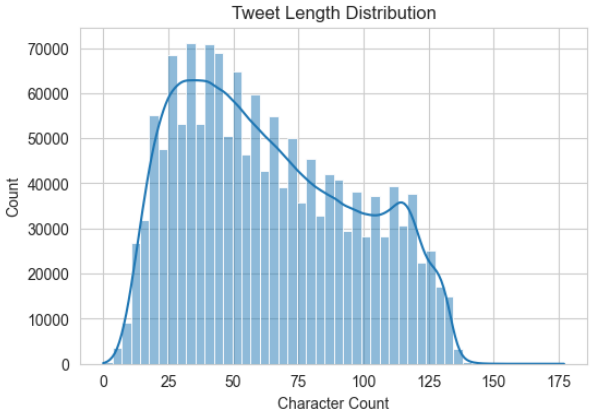
3.3 Exploratory Data Analysis Framework

In order to understand the structural, lexical, and behavioral characteristics of the dataset, a thorough exploratory data analysis framework was utilized. To understand the distribution of the lengths of tweets and the number of words per tweet, histograms were created to visualize the data. This was to ensure that the tweets were well within the upper limit of the maximum sequence lengths for the BERT tokenizer, as well as the boundaries for the classical text models. In terms of assessing the strategy for the sample that was collected, the class distribution has verified that the stratified sample has positive and negative sentiments of approximately equal scale. This was to confirm that the strategy for the sample collection was valid. Further analysis was performed using n-grams, and with the help of the Scikit learn CountVectorizer, the most recurring unigrams, bigrams, and trigrams were calculated. This was to aid in the identification of the sentiment bearing expressions that are typical of tweets and are similar to those that were used researched in Maurya and Jha (2024). Themes were uncovered using the Latent Dirichlet Allocation, and were able to produce understandable subjects that were about different experiences, relationships, and emotions. Separate word clouds for each of the sentiments were created to visualize the lexical items that were most frequented, and positive and negative classes were created for the sentiments. Temporal engagement of the users was examined through tweet volumes over different hours of the week.

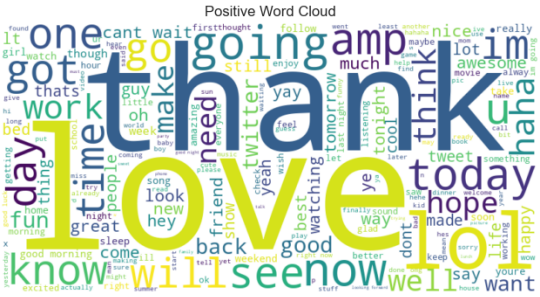
3.3.1 Exploratory Data Analysis results

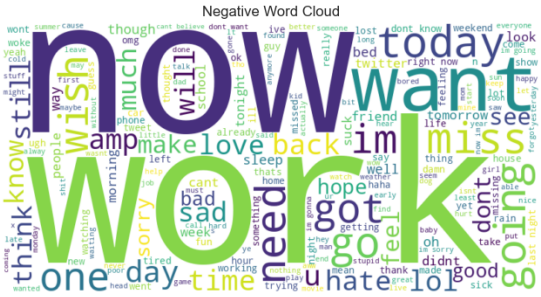
These are some of impotant EDA results

Tweet Length Distribution This plot displays the length of the tweets in the dataset. The length of most tweets is short, as is fitting of Twitter, and this aids in proving that our models should perform well with short text.

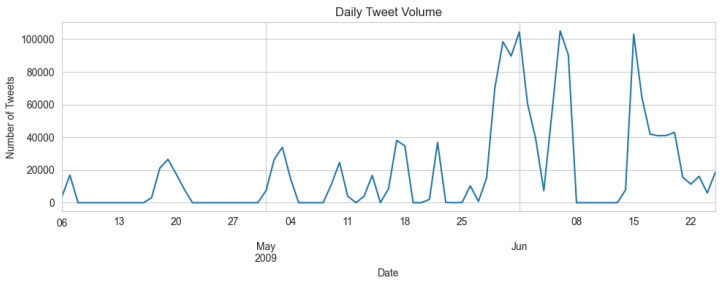


Word Clouds (Positive & Negative) These word clouds indicate the highest frequency of words used in positive and negative tweets. They provide a brief visual interpretation of the vocabulary that is related to each feeling.

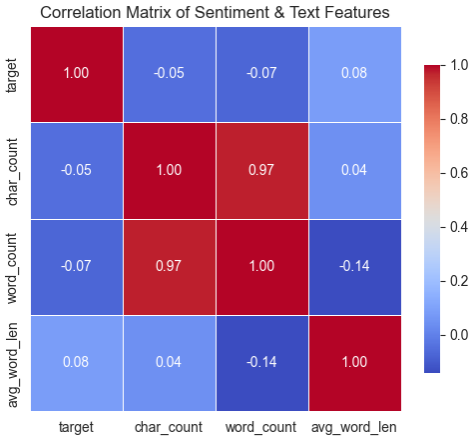




Daily Tweet Volume The time-series plot of the day reveals the variation of the usage of tweets in time. It assists in ensuring that there are no trends, spikes or anomalies in the data.



Correlation Heatmap The heatmap indicates the relationship between simple numeric characteristics such as character count and word-count and sentiment. It gives a brief summary of the existence of noticeable features of any simple features.



3.4 Classical Machine Learning Modelling Strategy

Modelling strategy one incorporated basic approaches in machine-learning such as Support Vector Machines and Logistic Regression, both of which served as decent starting points for text classification problems. These algorithms are considered standard for Twitter sentiment analysis and have shown consistent, above-average results when used with TF-IDF representations Gautam and Yadav (2014); Ahmed et al. (2023). Scikit learn Pipelines that combined text vectorisation and classification into one object were used in both classical models. The TF-IDF vectoriser created sparse feature matrices from the cleaned tweets, English stopwords were removed, and an n gram configuration was used which was determined through preliminary experiments. There was an increased maximum number of iterations for the Logistic Regression in order to ensure convergence for the large feature space. LinearSVC, which is a popular text classification SVM, was used and further refined with a fivefold GridSearchCV that looked at varying C values and n gram levels. All training made use of a stratified 20% sample with an 80-20 split for training and validation. The pure pipeline design made it easy to do a side-by-side comparison for both classical models with regard to reproducible preprocessing and modelling.

3.5 Transformer-Based Modelling Using BERT

The second approach employed a transformer. This was BERT base uncased, which has performed we well in sentiment analysis studies and has bidirectional attention and can capture deep contextual relations between tokens Sahoo et al. (2023). The tweets that had already been cleaned were first tokenised for the BERT WordPiece tokeniser, which had a maximum sequence of 64 for truncation and padding, which is a common configuration in sentiment analysis for Twitter in large-scale studies to achieve a good trade-off in computational cost for representational richness Enagandula (2024). The tokenisation then produced attention masks and input IDs, which were then converted to PyTorch TensorDatasets. For the model, the fine tuning was done with the PyTorch library and the Hugging Face Transformers library. A classification head that does binary

sentiment prediction was added on top of BERT. The output pooled was then used. The AdamW optimiser was used for training with a learning 2e-5 and a linear sched to decay the learning rate without a warm-up step. Given that the model was run for a single epoch, which was a result of execution constrainment (i.e., CPU only), it still took a good 13.5 hours to execute on the stratified subset. The loss for each batch during each epoch was computed and averaged resulting in a training loss of 0.38, indicating a convergence and consistent training. Validation accuracy and ROC AUC were calculated using softmax just before the end of each epoch in the holdout validation set. At the end of the training cycle, the model and tokenizer were stored using the `save_pretrained` function, resulting in a directory that can be used later for deployment and that contains the model configuration, weights, and a vocabulary.

3.6 Quantitative Evaluation and Statistical Analysis

Sentiment analysis research employs binary classification for model evaluation using a purely quantitative methodology. For each model, a review of classification quality was provided by computing the metrics of accuracy, precision, recall, and F1 score. Metrics in this case are standard points of reference for sentiment analysis research, which allows comparison by other research in the field Bello et al. (2023). Class-specific performance and patterns of frequent misclassification were examined and explained using confusion matrices in which distribution of true positives, true negatives, false positives, and false negatives were utilized. To measure the discriminatory strength of each model for various decision thresholds, Receiver Operating Characteristic (ROC) curves were constructed and Areas Under the Curve (AUC) values were determined. Classical models had ROC AUC values of about 0.85, while the BERT model was best, achieving about 0.92. Along with the predictive metrics, the time taken for each classifier to make and receive predictions for the validation set was also recorded. The trade-offs were determined to measure the relative computational efficiency, taking into account the low resources of Logistic Regression and SVM in contrast to the high resources of future transformer fine-tuning He (2024).

3.7 Application Level Validation and Real World Integration

The last step was to see how useful the new Streamlit application would be in the real world and to incorporate the trained models into the application. It can perform two functions. One feature is a Tweet Sentiment Classifier that allows users to input a single tweet or upload a CSV document ($<$ 200mb) for bulk predictions. Cleaning and TF IDF or tokenisation processing are applied to the input as during training, and the model selected provides a sentiment label and associated confidence values. Probability bar charts and confidence gauges portray model outputs in a user-friendly manner. The other feature is a Knowledge Graph Explorer which constructs and shows interactive graphs with NetworkX and PyVis that illustrate the relationships between users and mentions or between keywords and user accounts. Sentiment labels on the edges help users see how sentiment spreads in a local interaction network. A prediction request with the timestamp, model chosen, sentiment predicted, and confidence values is saved for each prediction in a CSV file. This system auditing functionality demonstrates a shift in research focus toward interactive, interpretable, and deployable sentiment analysis tools, as opposed to models that are studied in isolation Jonnala et al. (2025).

4 Design Specification

4.1 Overall System Architecture

The whole structure is split into 4 main flows of information. The data management section is responsible storing the Sentiment140 dataset in CSV format and the intermediate artefacts like the cleaned datasets and the engineered feature evaluation outputs. The processing section performs text cleaning and normalisation feature engineering to convert the unprocessed tweets to a structured format that can be used for modelling. The modelling section sits for fitting and evaluation of classical and transformer models which includes TF-IDF plus Logistic Regression and Support Vector Machine, and fine-tuned BERT. The application section contains the Streamlit interface, the prediction engine and the knowledge graph which gives real-time access of the trained models to the end users. The structure is designed for separation of data preparation, modelling and deployment. This is a requirement for a machine-learning system and is a part of best practices for modularity on which the system can be maintained and extended in the future Sahoo et al. (2023).

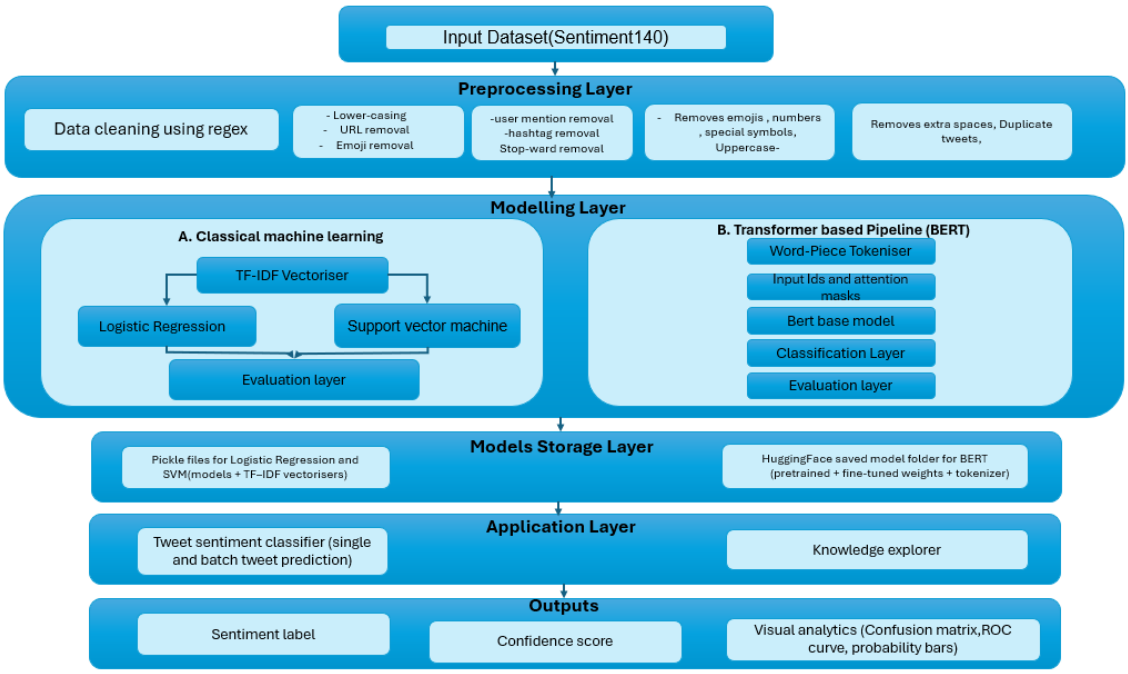


Figure 2: System Architecture Diagram

Figure 2 presents the overall architecture of the sentiment analysis system, illustrating the flow from raw Twitter data ingestion, through preprocessing and feature-engineering, into the classical and transformer-based model pipelines, and finally into the Streamlit application for real-time prediction and knowledge-graph exploration.

4.2 Data Processing Pipeline Design

The process consisting of loading, validating and parsing Latin 1 encoded CSV files is the starting point of the data processing pipeline that prepares tweets in their original,

unstructured format for downstream processing and analysis. The system runs a function based on regular expressions that strip the text of URLs, user mentions, hashtags, digits, emojis and other punctuation, and that also lowercases the text and normalises whitespace. Following Ahmed et al. (2023), we recognised that noise removal and lexical normalisation are critical to effective sentiment modelling and adopted these practices. Cleaning is followed by initial tokenisation and stopword processing in NLTK for quick exploratory analyses. Training pipelines will use TF IDF in Scikit learn for the more classical models, and the WordPiece tokenizer in the case of the transform model. A number of linguistic and behavioural indicators are appended to the base table for enriched user and communication pattern analysis. These indicators include word and character counts, unique word ratios, counts for hashtags and mentions, plus a range of other temporal features. In sentiment modelling research, these constructed indicators aligned with the field standards, and also demonstrated how sentiment varied with changes in the text and time made. Srivastava et al. (2019). .

4.3 Classical Machine Learning Pipeline Specification

In classical machine-learning, we focus on TF IDF vectorization and two linear classifiers: Logistic Regression, and Support Vector Machine. Using TF IDF vectorization, we convert each one of the cleaned tweets to high dimensional sparse vectors, which takes into account the relative importance of each of the terms in the corpus. TF IDF has proven to work really well in short text sentiment classification Singh et al. (2022). For Logistic Regression, we combine the TF IDF vectorizer with Logistic Regression, raising the maximum iteration limit to ensure we converge. For Support Vector Machine, TF IDF vectorization is combined with a LinearSVC classifier, which works well for large, sparse feature spaces. Using GridSearchCV with 5-fold cross-validation, we tune the hyperparameters on different values of C and n-grams. This configuration allows a fine search for the right hyperparameter settings to have the best trade-off from a generalization and performance point of view and also allows for a fair and strong baseline comparision with the model that is based on transformers Gautam and Yadav (2014).

4.4 Transformer-Based Model Architecture

The focus of this track is BERT base uncased, a bidirectional encoder that is able to capture relationships between tokens within the self-attention mechanism across the entire input sequence. BERT was chosen because, within the scope of transformer architectures, it is able to represent the sentiment analysis of a given models task and capture semantic nuance and long-range dependencies. A task that sentiment analysis models struggle to achieve. Bello et al. (2023) The WordPiece tokenizer takes each tweet and breaks it down to subword units, creating input IDs and attention masks, that flag which tokens are to be attended to and which are padding. Tokens are processed and contextualized into embeddings by multiple transformer layers and attention heads. A binary classification head is attached to the pooled output vector and the complete network is fine-tuned on the stratified subset. Stable fine-tuning of BERT in text classification tasks is reflected via the use of the AdamW optimizer and a linear learning rate scheduler. Enagandula (2024) The model architecture itself still remains the same as the standard BERT configurations to avoid deviations and maintain the same functionality, even when training is limited to CPU.

Figure 3 and Figure 4 illustrates the end to end model flow. The Figure 3 shows the classical pipeline, from cleaned text through TF IDF vectorisation to Logistic Regression or SVM classification. The Figure 4 shows the BERT transformer pipeline, from cleaned text through WordPiece tokenisation, transformer encoding and final classification output.

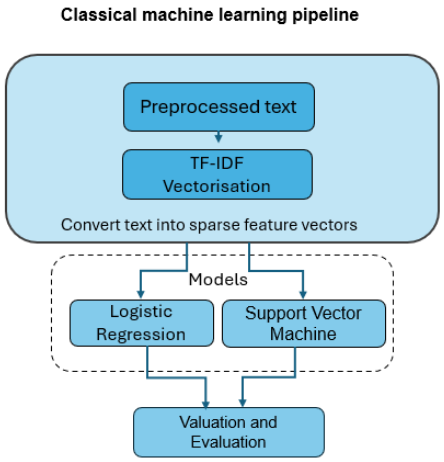


Figure 3: Classical machine learning pipeline

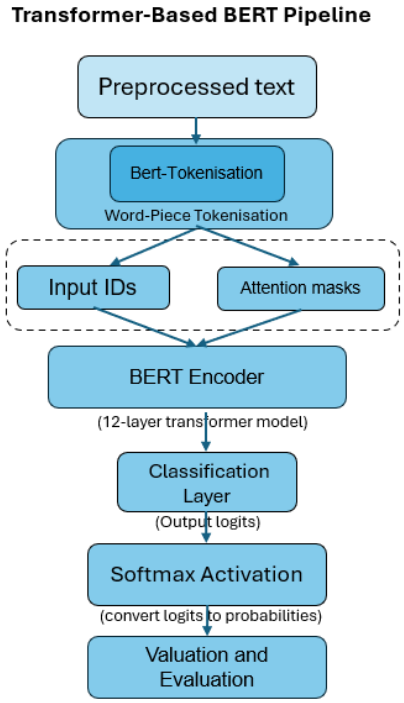


Figure 4: Bert pipeline

4.5 Application Architecture and Components

This application architecture has been done in Streamlit and has two main components, the Tweet Sentiment Classifier and the Knowledge Graph Explorer. The Tweet Sentiment Classifier has an interface where users can type in a tweet or upload a CSV file. The interface allows the user to choose one of the models and runs the necessary preprocessing steps and vectoriser or tokenizer and predicts the sentiment and gives numerically confidence predictions. There are elements like probability bar charts and gauges to make it easy for non-experts to understand the model output. In batch mode, the application also calculates accuracy, precision, recall, and F1 score, and gives confusion and ROC curve plots, when true labels are available. This design supports recommendations in the literature where sentiment systems are designed with flexible, user-oriented mechanisms Jonnala et al. (2025). The Knowledge Graph Explorer module supports network centric analysis by building graphs of user mentions or keyword user relationships with NetworkX and rendering them interactively using PyVis. Sentiment annotations for each node can be added to see the distribution of sentiment across the interaction network. The combination of natural language processing with analytic tools is a culmination of modern methodologies which look to grasp the essence of emotion and the manner in which it is organized in relation to social networks Talukder et al. (2025).

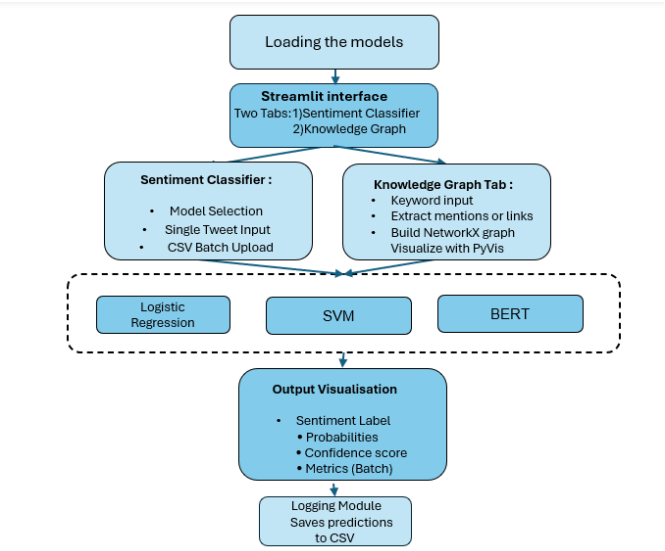


Figure 5: Application Architecture Diagram

5 Implementation

5.1 Data Cleaning and Dataset Preparation Outputs

Preparation of the Sentiment140 dataset had already been implemented before major dataset preparation activities were initiated. All of the raw corpus, in the form of CSV files, were archived and imported using latin 1 character encoding. All duplicables were removed, and 1.43 million tweets were a result of corpus cleaning. Since we were only interested in binary classification of polarity, the neutral class was also removed. In the interest of preserving the original sentiment distribution, a stratified twenty percent sample of 303,487 tweets was obtained. This sample also reduced the computational burden. All of the cleaning methodologies in the sample were executed, for the URLs, mentions, hashtags, digits, emojis and punctuations were removed, the text was lowercased, and whitespace was normalised. Before the sampling, exploratory analyses were done using NLTK for tokenisation and stopword analyses, though they weren't used in the modelling pipeline downstream. Various features were also provided in the form of engineered metrics based based on the tweets, such as text length, word count, average length of words, hashtags and mentions, unique words, and time features like year, month, day of the month, week, hour and many other time period elements. The merged attributes were combined into an analytical base table, allowing for the consideration of both text and behaviour for the purposes of exploratory research and model assessment Srivastava et al. (2019). The dataset was cleaned, enriched, saved, and loaded onto the disk to be the foundation for all further modelling.

5.2 Machine Learning Model Development and Training Outputs

The trained classical machine-learning models were the next outputs from the implementation. Both models, Logistic Regression and Support Vector Machine, were executed as

Scikit learn Pipelines, which, in this case, are abstractions where the TF IDF vectoriser is bundled together with the classifier, forming a single object that includes all the preprocessing and prediction functionalities. In the case of Logistic Regression, the TF IDF vectoriser created features at the unigram level after English stopwords had been removed, and the classifier was set at a maximum of 1000 iterations in order to converge on the highly dimensional feature space. Stratified training of the subset took about 9.2 seconds, and validation prediction took 1.2 seconds. This Logistic Regression model had an accuracy of approximately 0.77, a macro F1 of 0.78, 0.84 of the ROC AUC, which was in line with what has previously been documented with TF IDF in combination with Logistic Regression as the baseline in Twitter sentiment Singh et al. (2022). As for SVM model, the pipeline used the same TF IDF vectoriser with a LinearSVC classifier. Hyperparameter values were adjusted with the help of GridSearchCV, with a 5-fold cross-validation tuning for different n-gram ranges and C values. The most effective configuration employed a lower C value along with a mix of both unigrams and bigrams for n gram range, resulting in obtaining 0.77 accuracy, 0.77 F1 score and around 0.85 ROC AUC for validation. Training SVM on the subset took about 186 seconds, while the validation set took around 2.5 seconds for prediction. Because LinearSVC does not generate probabilities, the ROC AUC and ROC curve are computed using decision score details. Retroactively, both classical pipelines along with their TF IDF vectorisers were saved as pickle files so that they could be loaded back in without the need for re-training within the deployment ecosystem. Retaining both model weights and precise feature mapping historically is crucial for dependable inference and is a standard machine-learning deployment best practice Gautam and Yadav (2014).

5.3 Transformer-Based BERT Fine Tuning Outputs

A key outcome from the plans put into action was the fine-tuned BERT base uncased model. From the Hugging Face repository, the model and the tokenizer were downloaded and modified for binary sentiment analysis by appending a classification head. The cleaned tweets and the labels from the stratified subset were transformed into Hugging Face Datasets, and these were processed by the tokenizer with truncating and padding to a length of sixty-four tokens. The resulting input IDs, attention masks, and labels were converted into PyTorch TensorDatasets and served through DataLoaders set to a batch size of thirty-two with shuffle set to true for the training loader. In the training procedure, the AdamW optimiser with a learning rate of 2e to the power of minus 5, and a linear learning rate scheduler was used, and with no warmup steps to conform to standard practice for BERT fine-tuning for text classification Bello et al. (2023). Due to the CPU-only constraints, training was done for a single epoch. Cross-entropy loss was computed, and for this epoch, a total of 0.38 was registered, meaning that training was stable and divergent into the loss. With every training epoch, the model was assessed based on the validationset using softmax based probability to compute the accuracy, precision, recall, F1 score, and ROC AUC. For the fine-tuned model, the validation accuracy was approximately 0.84 and the ROC AUC was 0.92, which serve to show the advantage of using contextual transformer representations over classical baselines. The defend components of the trained model impact, the configuration file, vocabulary, and weight tensors, using the `save_pretrained` method, resulting in a model.safetensors file, config.json, vocab.txt, and other tokenizer metadata in a pre-defined directory structure. This structure is the basic transformer artefact used in deployment, which was copied

into the application folder.

5.4 Analytical and Evaluation Outputs

5.4 The stage of implementation also resulted in the generation of a wide range of analytical visualisations and evaluation artefacts. For the purpose of exploratory data analysis, distributions of sentiment, length of tweets, and word count were visualised, and n-gram frequency graphs were created. There were also time series graphs of the volume of tweets by hour and by day of the week. Latent Dirichlet Allocation (LDA) was used to derive topics which were then word cloud visualisations for the positive and negative tweets to make the dominant lexical patterns more accessible. There was evidence of deeper engagement with the data beyond just training a single model. In the course of model evaluation confusion matrices were created for the Logistic Regression, SVM and BERT models to visualise the distribution of correct and incorrect predictions across the different sentiment classes. For each of the models ROC curves were generated and the AUC computed, where BERT was clearly superior to the classical models. To visualise the difference in performance across the models, comparative bar charts were created from the F1 scores and ROC AUC values. Furthermore, a table outlining the training and prediction times of each of the three models was created which illustrated the excessive computational resources required by transformer models for fine tuning in comparison to classical models. These outputs, both visual and tabular, support the assertions discussed in the Evaluation chapter and respond to the growing literature demand for thorough, multiperspective evaluations of sentiment models He (2024).

5.5 Streamlit Application Development Outputs

The last step of implementing functionality of Streamlit is complete and operationalised the trained models for interactive use. The functionality of the application is divided into two main parts. The first is the Tweet Sentiment Classifier where users can enter information manually and upload a CSV file for batch prediction. The application will access the respective classical model's pickle file or BERT model directory and execute the same training cleaning and vectorization. The results of the prediction along with the respective confidence scores are listed. The application will determine and report the respective accuracy, precision, recall, and F1 score for batch inputs that contain a ground truth file along with the confusion matrix and the ROC curves for situational awareness evaluation feedback provided on the spot. The second application functionality is labelled the Knowledge Graph Explorer. A subset of tweets is transformed into an interactive representation of a network built using networkX where the nodes are users or words and the relationships, like mentions or co-occurrence, are the edges. The resulting PyVis network can be zoomed, dragged, and inspected so that the sentiment for each node can be visually explored. This functionality transforms the system from a basic classifier into a module for examining the distribution of sentiment in an interaction network. This illustrates the increasing interest in integrating sentiment and network analysis in social media studies Talukder et al. (2025).

- The following figures show outputs from the Streamlit application.

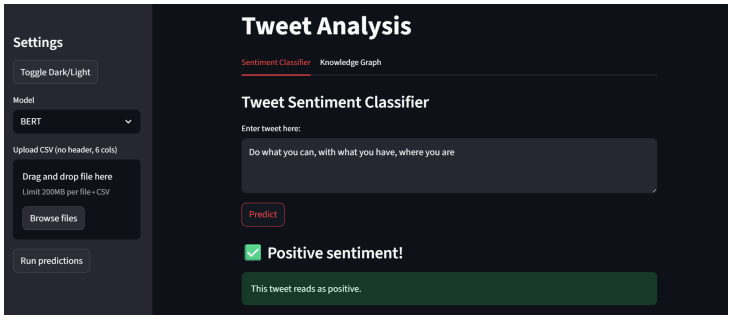


Figure 6: Input Field UI

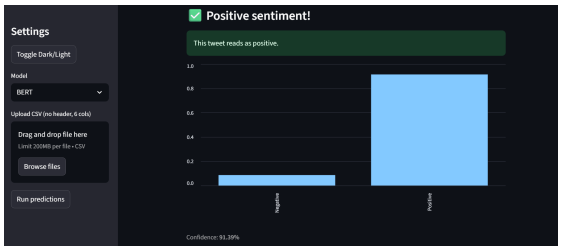


Figure 7: Predicted sentiment bar chart

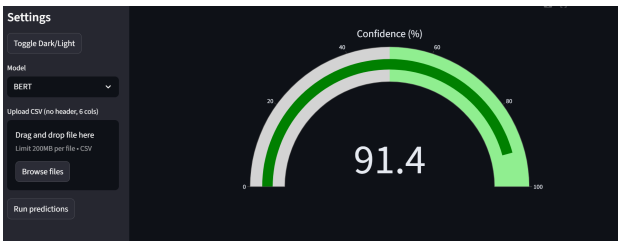


Figure 8: Confidence Score

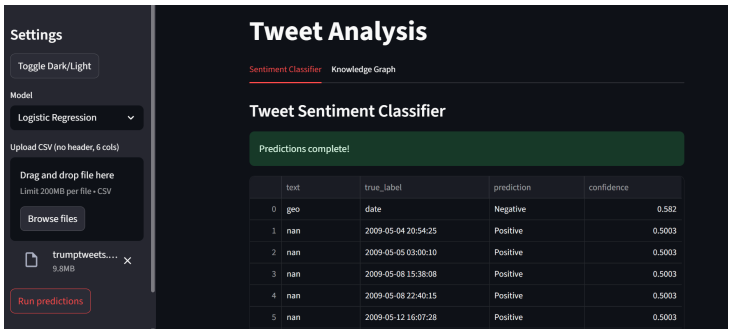


Figure 9: Batch prediction UI

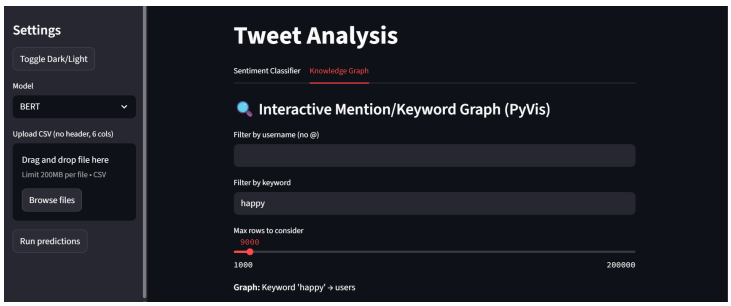


Figure 10: Knowledge Graph Tab UI

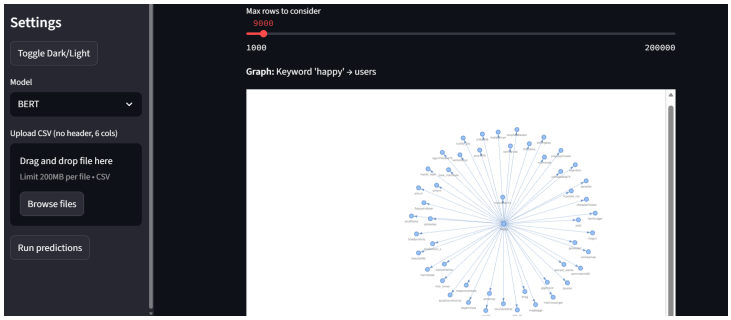


Figure 11: Knowledge Graph

5.6 Tools, Libraries and Technologies Implemented

The implementation made use of the available support of data science and machine-learning within the Python ecosystem. For data manipulation, the use of Pandas and NumPy was preferred. Text cleaning and exploratory analysis of the data was made easier through regular expressions and NLTK. The Scikit-Learn library was used for TF – IDF vectorisation, Logistic Regression, LinearSVC, evaluation metrics and GridSearchCV. For exploratory data analysis and evaluation, Matplotlib and Seaborn were used to create static visualisations. The Hugging Face Transformers library and PyTorch were used for BERT fine tuning, including, but not limited to, tokenisation, model definition, training and evaluation **?**. For graph-based visualisation within the knowledge graph module, PyVis and NetworkX were utilized, and Streamlit was used to build the interactive web app that offers real-time sentiment prediction. The power of the tools used and their accessibility constitute what can be termed as a modern sentiment analysis stack, making the project extensible of replicable by other researchers and practitioners.

6 Evaluation

6.1 Introduction to the Evaluation Framework

This project included the Evaluation stage wherein focus was placed on testing the performance, robustness and usability of the three models developed in previous chapters: Logistic Regression and Support Vector Machine, and the BERT Model. Structured evaluation methodology is critical in sentiment analysis because of the peculiarities in Twitter data which is noisy, contextual, and highly variable in terms of its natural language form, and this can affect the behaviour of the classifier. Evaluating models in this field of work involves the computing of several herein mentioned metrics which include accuracy, precision, recall, F1-score, and patterns of confusion matrices and ROC-AUC, which is standard practice. A multi-faceted evaluation of the models was carried out, which is in line with previous authors' recommendations on the avoidance of one-metric evaluations, owing to the high probability of such evaluations missing some classifier shortcomings in text of low balance and ambiguity. Suffice it to note that assessing computational efficiency proves to be critical in justifying the balance of predictive efficiency and resource exhaustiveness, which is a common consideration in most current transformer-based models. The evaluation was however not entirely qualitative, since it included the testing of

the Streamlit application to assess its potential for real-world interaction in terms of usability, agility and consistency.

6.2 Performance Evaluation of Logistic Regression

Logistic Regression was the first base model in this research as well as in the other models that were investigated in this research. This model was consistent across the different measures as well. In this regard, the model managed to attain an accuracy of 77%, which corroborates other research findings that employed TF-IDF for vectorisation process, in digital sentiment classification for tweets. This also confirms that linear decision boundaries do in fact work for sparsely populated high dimensional text features Singh et al. (2022). The models predicted F1 score was 78%, which indicates that the model was well capable of predicting the positive sentiment and as well as the negative sentiment. Notwithstanding the presence of false positive, false negative, and true negative, the confusion matrix shows that the Logistic Regression model failed to capture the fine-grained sentiment, especially when it comes to the use of sarcasm and other contextual tokens Gautam and Yadav (2014) which is indeed one of the weaknesses of linear classifiers. On the other hand, the model's AUC value of (0.84) illustrates the model 2019s discriminative strength ability. This infers that Logistic Regression is better than a random classifier at discriminating among the sentiment classes. However, the curve does show a somewhat gentle separation compared to the BERT model. Logistic Regression is a reliable baseline model, but because it does not have the ability to model the non-linear semantic relationships that characterize the unique contextual patterns of the Twitter language Ahmed et al. (2023), it may be subpar.

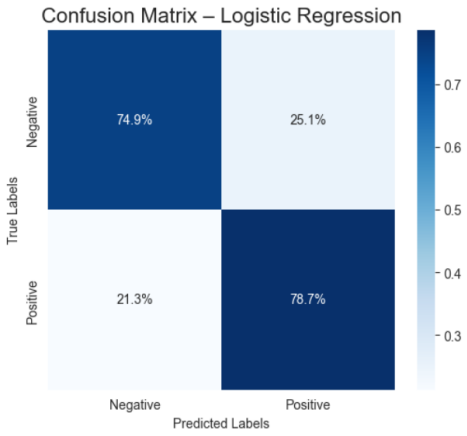


Figure 12: Confusion matrix - LR

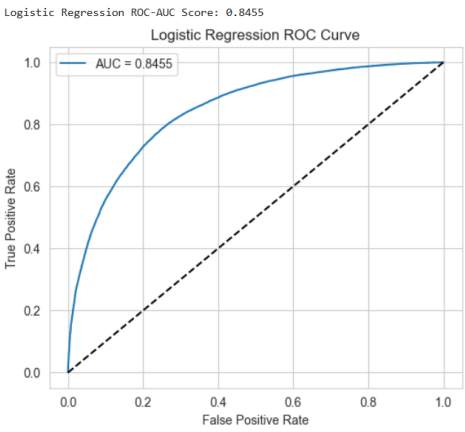


Figure 13: ROC LR

6.3 Performance Evaluation of Support Vector Machine

The Support Vector Machine model had the same outcome as Logistic Regression, recording an accuracy rate and F1-score of 0.77. Support Vector Machines are often reported to be one of the top performers for tasks of sentiment classification as they are able to achieve and maintain large decision margins in complex hyperspaces, particularly with the aid of TF-IDF features Maurya and Jha (2024). The SVM confusion matrix exhibits a distribution of such erroneous classifications resembling to Logistic Regression and, SVM also exhibits a slightly stronger command in the area of false positive. Such a trend

indicates the SVM margin maximisation aspect led to more enhanced class separability for the sentiment classification tasks with clearer lexical clues.

The ROC curve indicates that SVM reached an AUC of around 0.87, only marginally surpassing what was achieved with Logistic Regression. This indicates SVM's steady performance with rich in sparse features datasets. This is consistent with previous studies from Srivastava et al. (2019) analysing classical models ins attention related corpora on Twitter. The SVM continues to provide robust levels of performance but is still bounded by the classical models major limitation of patterning only surface level lexicon.

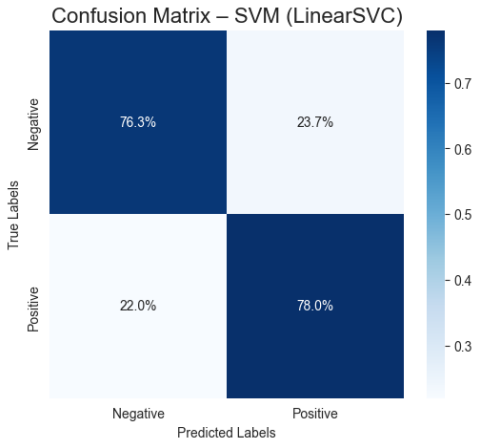


Figure 14: Confusion matrix SVM

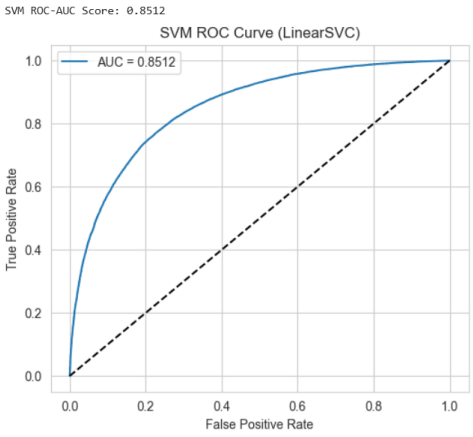


Figure 15: ROC SVM

6.4 Performance Evaluation of the BERT Transformer Model

Out of all the models tested, BERT continues to outperform the rest, which corroborates with the new studies done by Sahoo et al. (2023) studies which highlight the utility of contextual embedding transformer models in the informal text message studies. Sure, BERT got an accuracy of 84 which outperformed classical models not to mention achieved a strong F1 accuracy. This is evidence of the improvements that can be achieved with bidirectional encoder models as it enables the model to learn contextual dependencies of words, and not just word frequency. The confusion matrix BERT shows that there is a lower amount of misclassifications in classical model when looking at underlying positive and negative insights. This model definitely has lower falsepositive/negatives which demonstrates its ability to understand complex linguistic processes such as negation and its idiomatic and emotion-laden expressions whereas the traditional TF-IDF models just fall short He (2024).

The ROC curve further supports BERT's superior discriminative power, with an AUC of approximately 0.92, the highest among all three classifiers. These results demonstrate the advantage of transformer models in sentiment analysis tasks where contextual information is crucial, although this comes with a notable increase in computational cost.

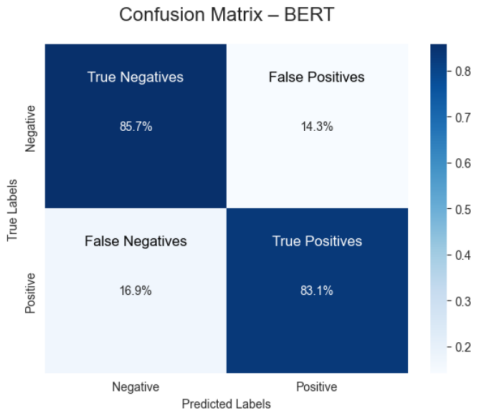


Figure 16: Confusion matrix BERT

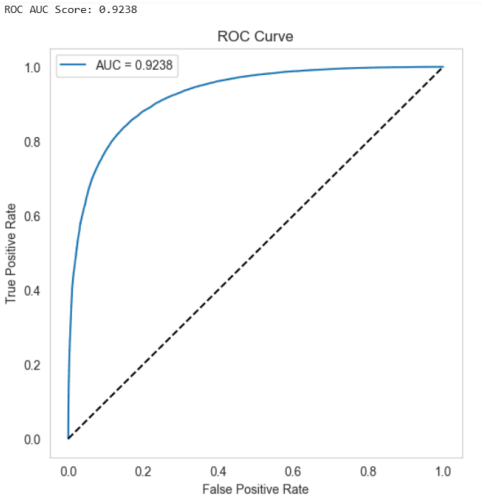


Figure 17: ROC BERT

6.5 Comparative Analysis of All Models

The analysis of the models shows how BERT is better than Logistic Regression and SVM in relation to the F1 score and the ROC-AUC score. The comparative F1 score chart shows BERT F1 score is the highest. This is a reflection of the deep representation learning that BERT is capable of. A similar pattern can be observed in the ROC-AUC score. BERT is also the highest in the ROC score with SVM and Logit Regression following. This is consistent with previous research that shows Transformers to outperform all other models in Social Media Sentiment Datasets because of their semantic modelling capabilities Sharma et al. (2025).

Model Performance Comparison

	Model	Accuracy	Precision	Recall	F1 Score	ROC-AUC
0	Logistic Regression	0.7677	0.7545	0.7869	0.7704	0.8455
1	SVM (LinearSVC)	0.7715	0.7637	0.7798	0.7717	0.8512
2	BERT	0.8444	0.8512	0.8310	0.8410	0.9238

Figure 18: Model Performance Comparison

The comparative analysis clearly indicates that although classical models provide competitive baselines, the transformer model meaningfully surpasses them across all performance indicators.

6.6 Discussion

The classical ml model results replicate what is obtainable from previous studies. There is no difference in accuracy (around 0.77) and F1 score (0.77-0.78) from logistic regression and svms. Results support past findings that suggest strong baseline accuracy is

obtainable from linear models with tf-idf vectorisation when applied to noisy short texts of tweets. Ahmed et al. (2023); Singh et al. (2022) confirms that classical learners are relevant for large-scale sentiment tasks where efficiency and interpretability are required. Results obtained also show social media models are poor at contextualization, which is one of the major and inherent limitations in classical ml models. Unlike classical ml models, the transformer ml models which are BERT based outperformed the classical ml models with metrics 0.84 accuracy and auc of 0.92. There is overwhelming literature support for the metrics obtained Bello et al. (2023); Sahoo et al. (2023) that indicates informal contextual languages are best served/ precision is guaranteed when such texts are processed using contemporary embedding architectures. Recognizing sentiment signals such as negation and idiot phrase negation is something BERT is slowly working towards. The consensus thus BERT works best is sentiment microblog analysis especially on data containing a variety of linguistic features. Using Streamlit, people were able to interact with the sentiment analysis system and this gave an indicative visualization of confidence prediction and prediction output Streamlit functionality shows real-time prediction on both single and multi-text prediction. The accomplishment of this project is demonstrated by the results. BERT vs Classical Model Comparison shows BERT has the best accuracy while classical has the best efficiency; this, thus answers the research question where BERT lies on the accuracy/cost computational scale. Gaps that were discussed in the literature are in fact addressed.

7 Conclusion and Future Work

7.1 Revisiting the Research Aim and Objectives

The focus of the research was on systematically and carefully comparing traditional ML techniques and transformer models on the sentiment analysis of huge volumes of Twitter messages. This was because prior works were reviewed, and it was found that there have been little to no attempts to compare the different models/ methodologies under the same conditions with the same datasets and do a proper head-to-head comparison. The research question was which sentiment analysis models among Logistic Regression, Support Vector Machines, and BERT were the most accurate and computationally cheaper in Twitter sentiment classification analysis? To answer this question, a meticulous procedure was designed with detailed data cleaning, feature preparation, data exploration, classical modelling with TF-IDF, finetuning a transformer of BERT, proper evaluation of the model, and building a Streamlit application to answer the model for the user/ audience. The model comparison was designed for consistency to ensure the fairness of the comparison in the ML research. The research objectives were met in total, including preparation of the Sentiment140 datasets, execution of different ML pipelines, assessment of the ML models with rule-based evaluations and application evaluations, and verifiable and transferable outcome.

7.2 Summary of Findings

For the evaluation phase, the analysis reveals a definite ranking of the effectiveness of the models. The Logistic Regression and SVM classical models achieved positive, stable results with an accuracy of nearly 0.77, validating the prior results supporting the fact

that linear models and TF-IDF together form a strong baseline model for text classification Ahmed et al. (2023); Singh et al. (2022). While these models had difficulties with contextual, ambiguous, or informal issues, the efficiency of these models along with their low computational costs is advantageous for situations where speedy predictions on low storage hardware are required. However, the results were higher for the accuracy (0.84) and the ROC-AUC (0.92) of the BERT transformer model. This performance corroborates the literature describing the game-changing potential wrapped in contextual embeddings and self-attention mechanisms Bello et al. (2023); Sahoo et al. (2023). BERT was the one that managed to eliminate a lot of false positives and false negatives, showing that this model was particularly effective on a lot of the more subtle sentiment analysis cases that the older models had a lot of trouble with. This is in parallel to the other studies in this domain that sentiment analysis is moving towards transformer models. While this formulation is straightforward, this study also clearly highlights the trade-off between accuracy and efficiency, which is a major contribution of this study. Despite BERT yielding the best predictive performance, extensive computational power and long training durations on the CPU were needed. In contrast, logistic regression and SVM models offered instant training and inference, albeit with less accuracy. Hence, the observations made are of great interest to system designers, researchers, and practitioners who are weighing the trade-offs between classical and transformer-based methodologies considering their operational constraints and designs. The Streamlit application showcased the integration of these results into a functional interface, confirming the models' developed utility. The platform offered real-time predictions with probability visualizations, interactive batch processing, a knowledge graph interface, and advanced analytics to facilitate and streamline broader analytical processing. This part evidenced the work's applied contribution by exposing the means to operationalize research outputs beyond the experimental framework

7.3 Future Work

Future research must investigate training of the transformer models on advanced and domain-specific architectures. For example, BERTweet, RoBERTa and DistilBERT are tailored for social media texts and often surpass generic BERT on sentiment analysis tasks for Twitter Sahoo et al. (2023). Furthermore, the profound adjustments of the training's processing unit chips (GPUs) will allow noticeable improvements of the training and performance, conditioned on batch and training epoch size, to be suggested. More research could focus on the processing techniques that focus on keeping sentiment detecting symbols (emojis and hashtags) as well as improving tweet performance by incorporating emoji embeddings, hashtags and contextual augmentation. Furthermore, detecting models that focus on the irony (or sarcasm) common to Twitter posts could also be beneficial. Extending the classification problem could also prove to be beneficial. For example, the inclusion and differentiation of sentiment categories such as neutral, mixed, and even emotion based categories (e.g., happy, sad, anger) could prove to be beneficial. This sentiment-rich content derived from social media platforms can potentially provide a coherent understanding of the public sentiment on a given topic. This creates numerous avenues for advanced analytical extensions for the knowledge graph function within the Streamlit application. Incorporating dynamic network analyses or community detection could be useful in future work to analyse graph-based sentiment propagation models to social media behaviour in a networked and relational manner. The general

framework created from this study may be modified for some business uses in areas like brand tracking, analyzing consumer sentiment, predicting sentiment in politics, or for crisis communications. The use of Twitter streaming APIs/ real-time data coupled with a deployment to the cloud would generate a scalable, automated system.

References

Ahmed, W., Semary, N. A., Amin, K. M. and Hammad, M. (2023). Sentiment analysis on twitter using machine-learning techniques and tf–idf feature extraction: A comparative study, *International Journal of Computers and Information* **10**(3): 52–57.

Albladi, A., Islam, M. and Seals, C. (2025). Sentiment analysis of twitter data using nlp models: A comprehensive review, *IEEE Access* **13**: 30444–30468.

Bello, A., Ng, S.-C. and Leung, M.-F. (2023). A bert framework to sentiment analysis of tweets, *Sensors* **23**(1): 506.

Enagandula, P. (2024). Sentiment analysis using bert pre-trained model, SSRN Working Paper.

Gautam, G. and Yadav, D. (2014). Sentiment analysis of twitter data using machine-learning approaches and semantic analysis, *Proceedings of the 7th International Conference on Contemporary Computing (IC3 2014)*, Noida, India, pp. 437–442.

He, L. (2024). Enhanced twitter sentiment analysis with dual joint classifier integrating roberta and bert architectures, *Frontiers in Physics* **12**: 1477714.

Jonnala, N. S., Ram Teja, A. V. S., Rajeswari, S. R. et al. (2025). Leveraging hybrid model for accurate sentiment analysis of twitter data, *Scientific Reports* **15**: 24438.

Maurya, C. G. and Jha, S. K. (2024). Sentiment analysis: A hybrid approach on twitter data, *Procedia Computer Science* **235**: 990–999.

Sahoo, A., Chanda, R., Das, N. and Sadhukhan, B. (2023). Comparative analysis of bert models for sentiment analysis on twitter data, *2023 9th International Conference on Smart Computing and Communications (ICSCC)*, Kochi, Kerala, India, pp. 658–663.

Sharma, U., Pandey, P. and Kumar, S. (2025). Context-based sentiment analysis using a bigru–distilbert fusion model for covid-19 tweets, *Scientific Reports* **15**: 39475.

Singh, S., Kumar, K. and Kumar, B. (2022). Sentiment analysis of twitter data using tf-idf and machine-learning techniques, *2022 International Conference on Machine-Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*, pp. 1–7.

Srivastava, A., Singh, V. and Drall, G. S. (2019). Sentiment analysis of twitter data, *International Journal of Healthcare Information Systems and Informatics* **14**(2): 1–16.

Talukder, M. A., Uddin, M. A., Roy, S. et al. (2025). A hybrid deep learning model for sentiment analysis of covid-19 tweets with class balancing, *Scientific Reports* **15**: 27788.