Natural Language Processing

**TEXT EMOTION DETECTION**

horizontal line

TABLE OF CONTENT

[**INTRODUCTION 1**](#_arolcxe0i15c)

[**PROBLEM STATEMENT 1**](#_la5jp5tnimjw)

[**ROLE OF NLP 2**](#_ufz8w4b1jin7)

[**TECHNOLOGIES USED 2**](#_te1zmabl5ni7)

[**FLOW DAIGRAM 3**](#_3j06vi96rqa3)

[**MODEL USED 4**](#_1i71eyuoonk)

[**OUTPUT 6**](#_3si48bgcedq8)

[**CODE 7**](#_njfqbpx8y56v)

[**SCOPE OF IMPROVEMENT 8**](#_ma1ntnl9nl11)

[**CONCLUSION 8**](#_jcfb10i1bsfh)

# 

# 

# 

# TEAM MEMBERS

# ROHIT SHARMA -2020UCO1697

# VISHAL VERMA - 2020UCO1648

# ARCHIT BANSAL - 2020UCO1709

# 

# 

# INTRODUCTION

Emotion detection through text analysis is a rapidly growing area of research in the field of affective computing. With the increasing importance of human-computer interaction in various domains, the ability of computers to recognize and respond to human emotions has become more important than ever. Text-based emotion detection involves the analysis of language patterns, tone, and word choice to infer the emotional state of a writer. This technology has numerous applications, including in marketing, healthcare, customer service, and social media. In addition, text-based emotion detection can also be used for more critical applications such as identifying suicide risks. Overall, text-based emotion detection has the potential to revolutionize the way humans interact with technology, making it more intuitive and responsive to our emotional needs.

# PROBLEM STATEMENT

The goal of this project is to develop an emotion detection system using Natural Language Processing (NLP) techniques to analyze text data and accurately identify the underlying emotional states of the author. The system should be able to classify text into different emotions, such as happiness, sadness, anger, fear, surprise, or neutral. The system will be trained on a large dataset of text documents with labeled emotions, and the model will use various NLP techniques such as text preprocessing, feature extraction, and classification algorithms to predict emotions accurately. The system's performance will be evaluated using standard evaluation metrics, and the model's results will be compared to human-labeled emotional states to determine its accuracy and usefulness in real-world applications.

# ROLE OF NLP

Natural Language Processing (NLP) plays a crucial role in emotion detection systems. NLP is a subfield of artificial intelligence that focuses on enabling computers to understand and process human language. Emotion detection systems that rely on text analysis require NLP techniques to accurately identify and classify the emotional content of a message.

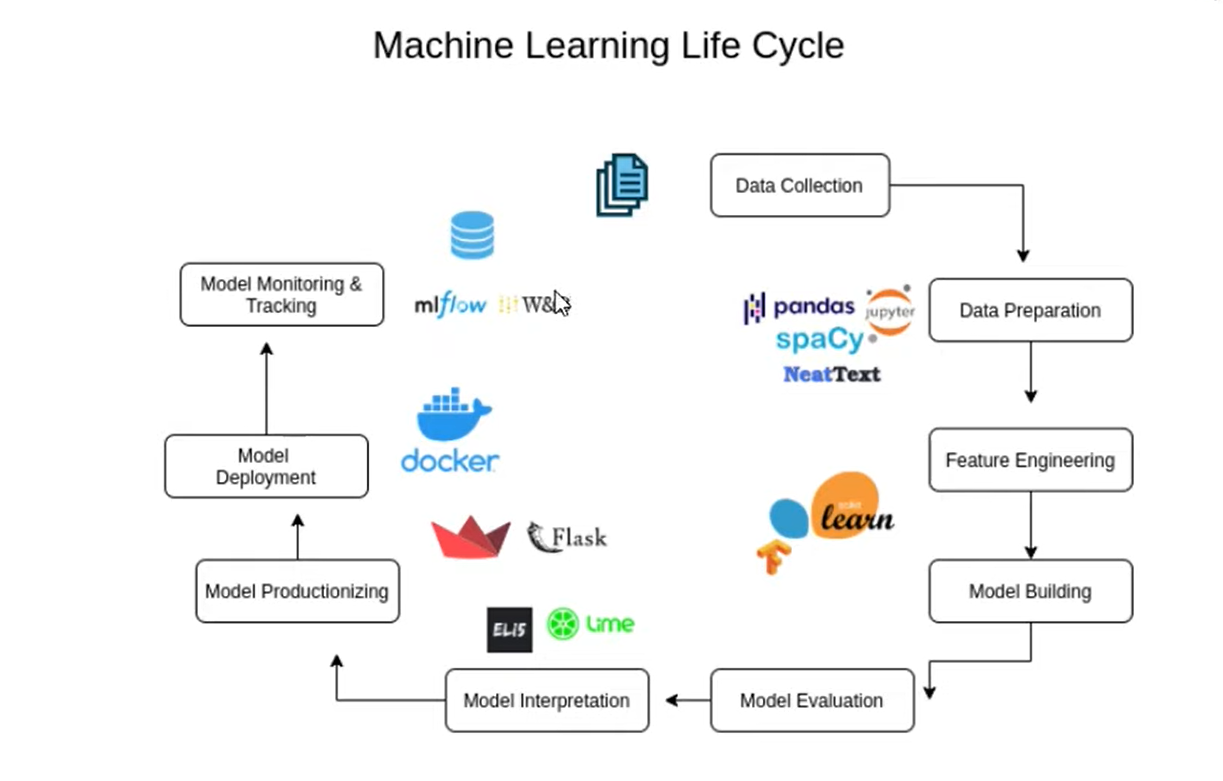
NLP techniques are used to preprocess the text data by removing stop words, stemming or lemmatizing the words, and converting the text to a numerical representation that can be used for machine learning algorithms. These techniques are critical to ensure that the model can identify relevant features and patterns in the text data that are associated with different emotions.

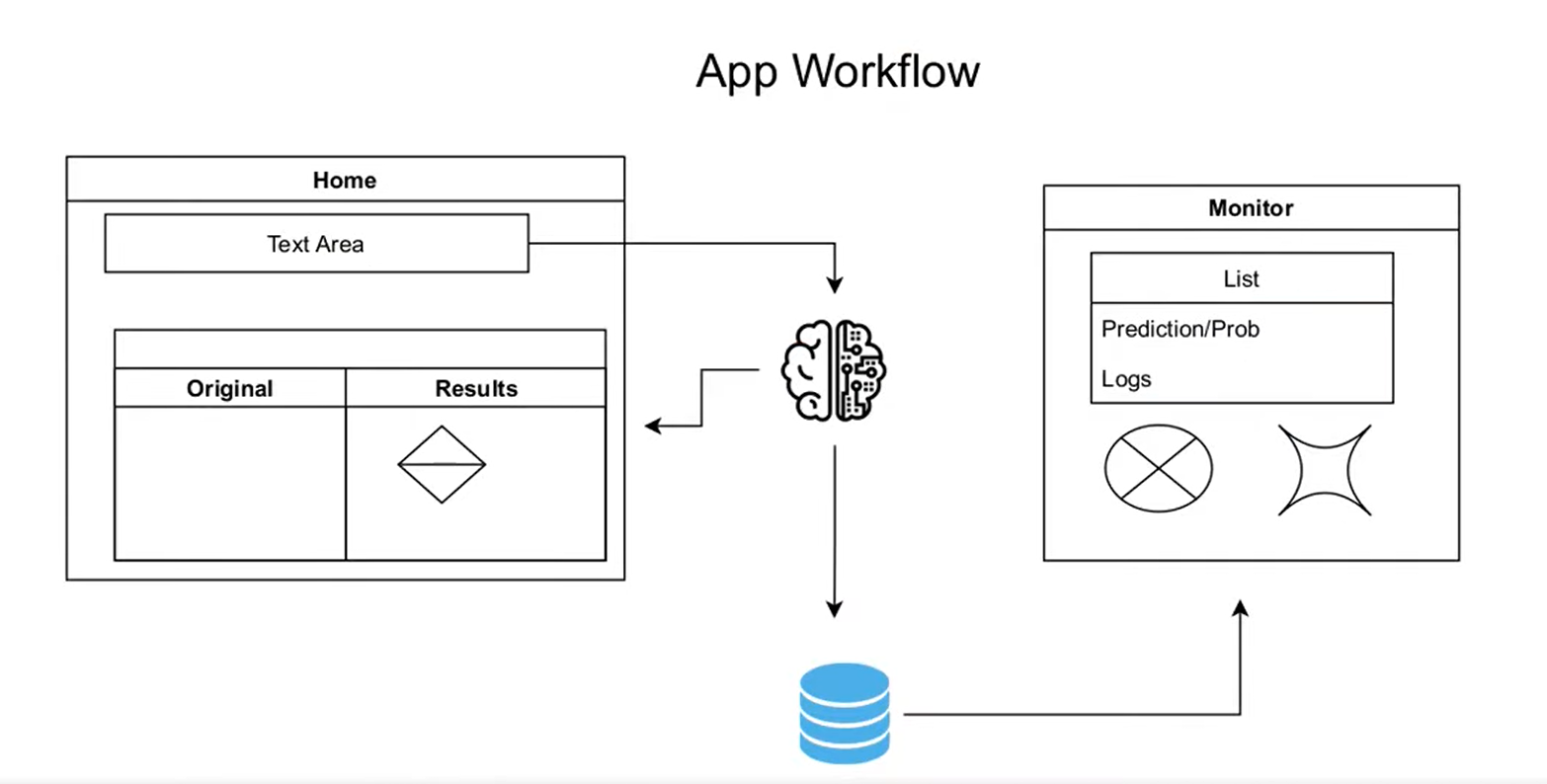
# TECHNOLOGIES USED

Model creation :

1. **Libraries**  : We will need several libraries such as pandas, NumPy, seaborn, and neattext to work with the dataset and build the model.
2. **Machine learning models used** : train several models (Random Forest, Naive Bayes, Logistic Regression and Linear SVM) on the training data. We will use the TF-IDF vectorizer to transform the text into numerical features that can be used by the models.
3. **Frontend Application** : For the frontend, we have developed an interactive user interface using the Streamlit application. Streamlit is an open-source framework that enables the development of highly customizable and user-friendly data-driven applications. The Streamlit interface has been designed to provide a seamless user experience and easy navigation, allowing users to input text data and receive real-time emotion predictions with ease. The use of Streamlit has allowed us to develop a responsive and intuitive frontend that meets the needs of our users.

# FLOW DIAGRAM





# MODEL USED

The following ML algorithms were used for prediction:

Using the count vectorizer, three models were built using different classifiers.

1. The first model used the Multinomial Naive Bayes Classifier, which achieved an accuracy of 58.46%. This algorithm uses Bayes' theorem with the "naive" assumption of independence between every pair of features to classify data.
2. The second model used Linear SVM and achieved an accuracy of 62.00%. This algorithm tries to find the hyperplane that best separates the data into different classes by maximizing the margin between the closest points from each class.
3. The third model used Logistic Regression and achieved an accuracy of 62.47%. This algorithm tries to find the best fitting logistic curve that can map input features to a binary output label, indicating which class the data point belongs to.

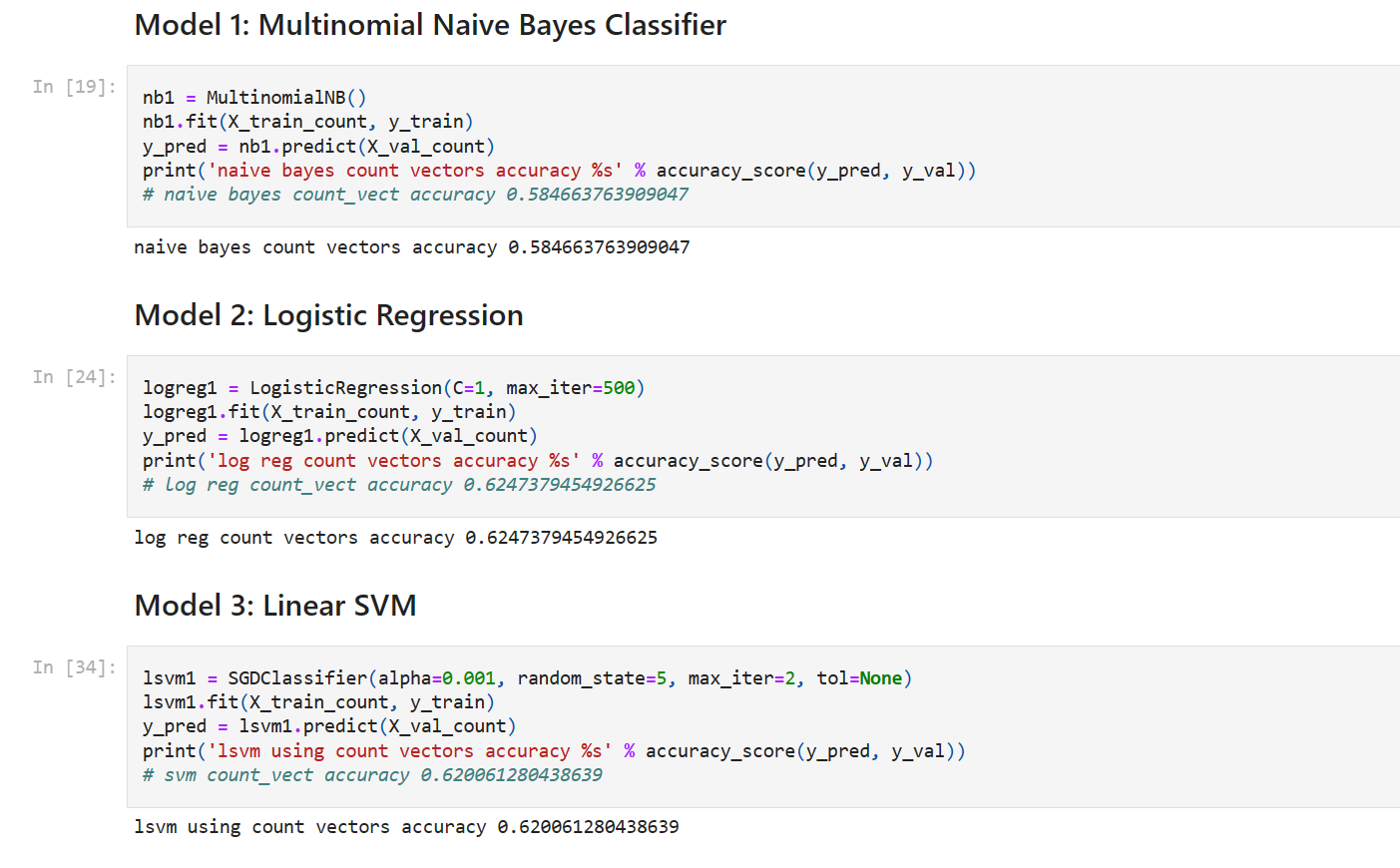
Using the TF-IDF vectorizer, three models were also built using different classifiers.

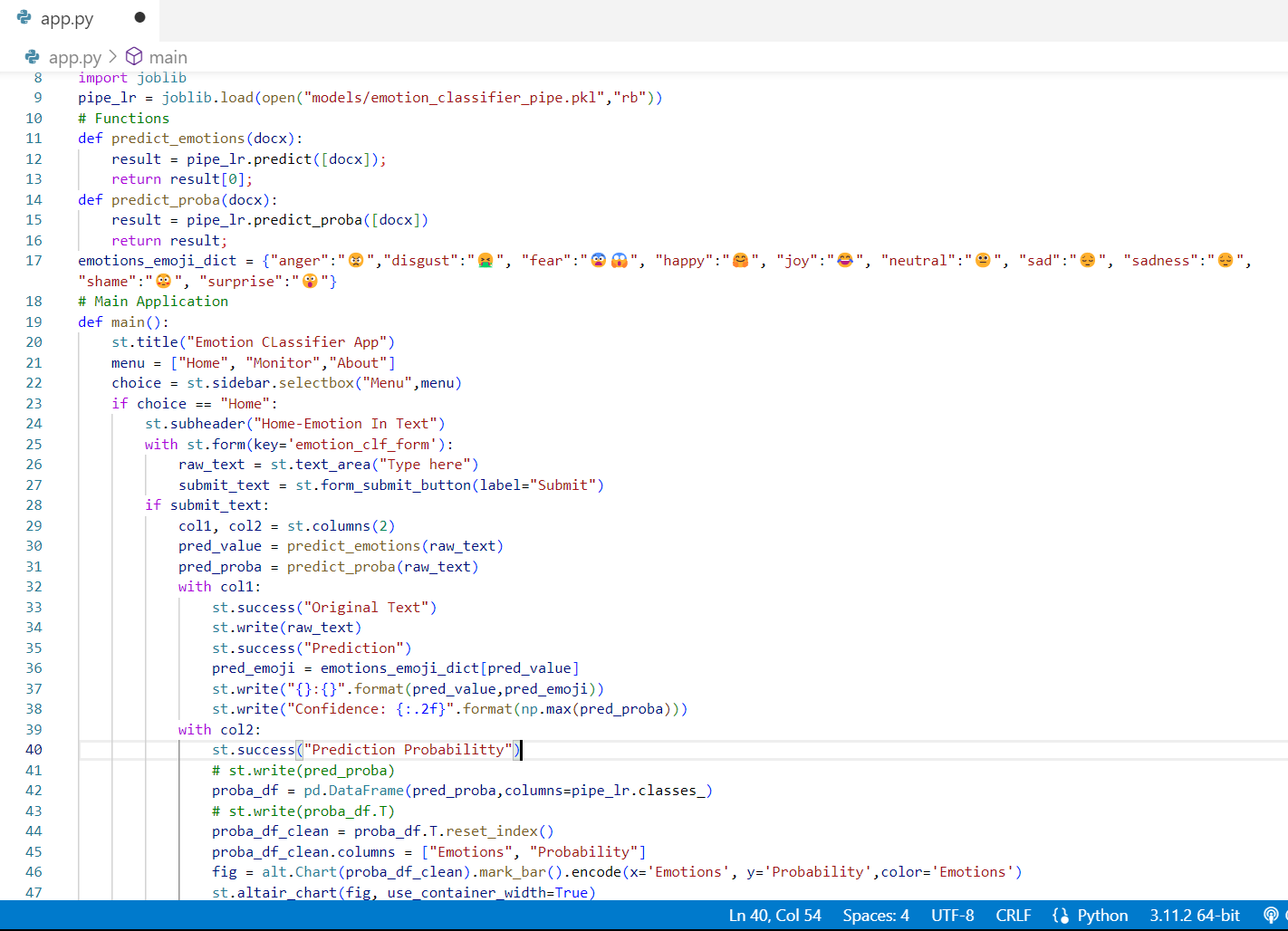
1. The first model used Multinomial Naive Bayes Classifier, which achieved an accuracy of 38.37%.
2. The second model used Linear SVM and achieved an accuracy of 38.49%. Both these models were built with the same classifiers as the count vectorizer models, but using a different feature extraction method.
3. The third model used Logistic Regression and achieved an accuracy of 40.13%. This algorithm is similar to the Logistic Regression used in the count vectorizer models but used the TF-IDF feature extraction method instead.

# OUTPUT

# 

# CODE





# SCOPE OF IMPROVEMENT

The scope of improvement for this system is to make it suitable for code-mixed text such as Hinglish (Hindi-English). This can be achieved by incorporating techniques such as language identification and incorporating language-specific features in the model. Additionally, using language-specific pre-processing techniques and data augmentation methods can also improve the performance of the system on code-mixed text.

# CONCLUSION

In conclusion, the logistic regression model outperformed the SVM and Naive Bayes models when trained on the dataset using the count vectorizer. Therefore, the logistic regression model was selected to make predictions in the application. It is important to note that while the accuracy of the model is a crucial factor in model selection, other metrics such as precision, recall, and F1 score are also considered when evaluating the performance of the model. The Performance we attained using the