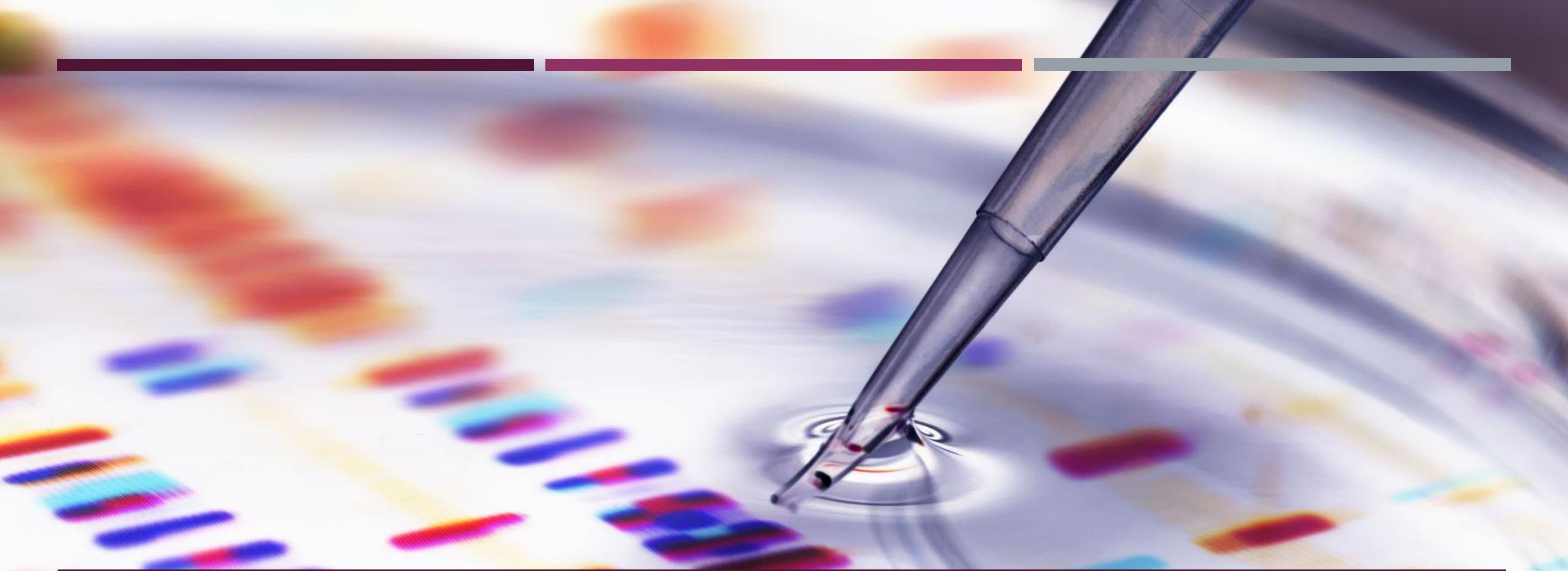
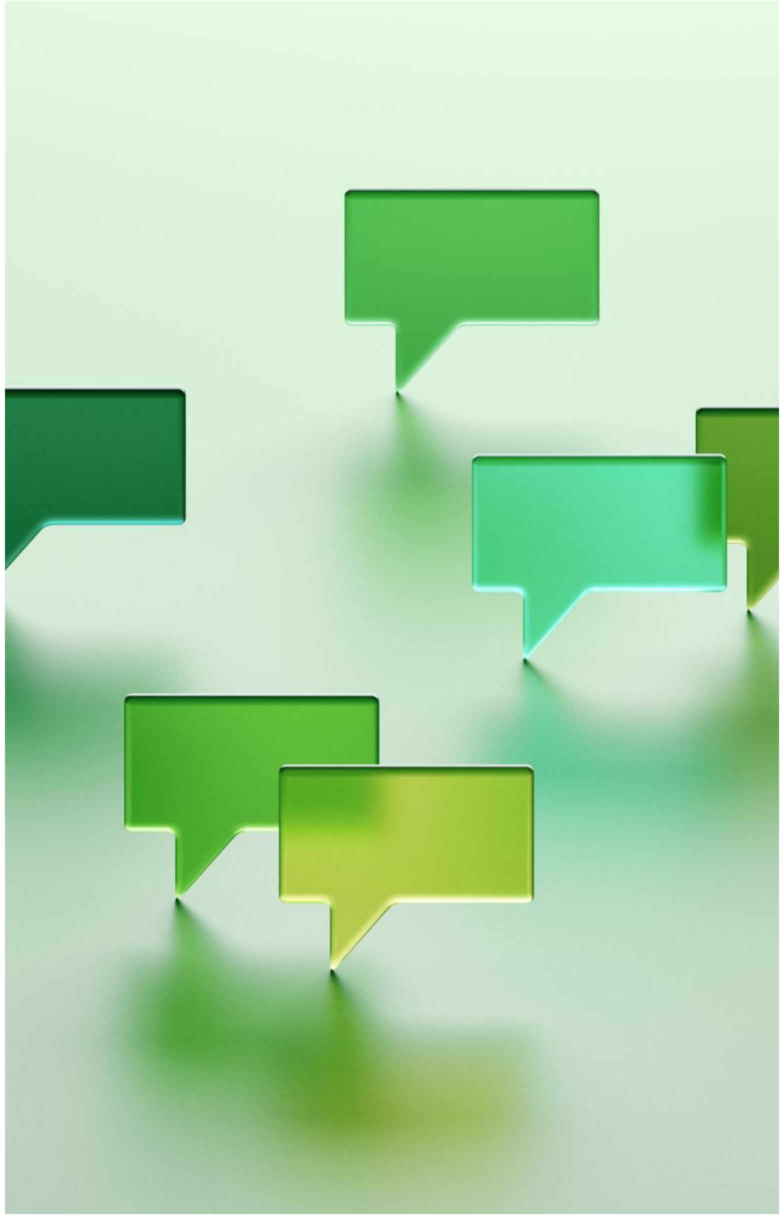


TWITTER SENTIMENTS ANALYSIS

BY GROUP 2.



PROJECT DESCRIPTION; PROBLEM STATEMENT

Background

- In today's digital world, customers and the public frequently express their opinions, feedback, and emotions on platforms like Twitter. These short, informal, and real-time messages often contain valuable sentiment that businesses and organizations can use to better understand public perception, customer satisfaction, or reaction to products and events.

Problem Definition

- The objective of this project is to develop a machine learning model that can accurately classify the sentiment of tweets related to a specific product or brand. Each tweet will be categorized into one of three classes: Positive, neutral and negative. The model will be trained on labeled tweet data and should be able to generalize to unseen tweets to support tasks such as:

  1. Real-time brand monitoring
  2. Customer feedback analysis
  3. Public relations response prioritization

PROJECT DESCRIPTION; OBJECTIVES

Main objective :

- To build and evaluate a sentiment classifier using NLP techniques that can:
  1. Preprocess raw tweet data (cleaning, tokenization, vectorization)
  2. Train and compare multiple classification models
  3. Interpret model predictions using tools like SHAP and LIME
  4. Predict sentiment for new/unseen tweets with reasonable accuracy.
- Other objectives include;
  1. Deploying the sentiment analysis app.
  2. Plotting the sentiments per product.
  3. Creating a report on the sentiments.

PROJECT DESCRIPTION; DATA DESCRIPTION

- For this project, we are using the Twitter Sentiment Dataset from Kaggle.
- The dataset provides labeled data for training and evaluating machine learning models in sentiment classification of tweets.
- It includes thousands to millions of real tweets, each labeled with the sentiment expressed by the user — whether positive, neutral, or negative.
- It is commonly used to build models that can detect the public's mood toward products, brands, people, or events.
- The data was sourced from this source https://www.kaggle.com/datasets/saurabhshahane/twitter-sentiment-dataset.

DATA PREPARATION

- Before analyzing the data we:

1. Cleaned the dataset by dropping missing rows and converting all text to lowercase.
2. Removed noise such as URLs, mentions, hashtags, digits, and punctuation using regular expressions.
3. Tokenized the cleaned text into individual words using NLTK's word_tokenize.
4. Removed common English stopwords to reduce non-informative words.
5. Applied stemming to reduce words to their root form (e.g., "playing" → "play").
6. Performed lemmatization to convert words to their dictionary base forms (e.g., "better" → "good").
7. Vectorized the lemmatized text using CountVectorizer to create a numerical representation.
8. Generated a sparse matrix of token counts and extracted the vocabulary.
9. Added a text_length column to track the number of words in each sample.
10. Final dataset includes cleaned and structured text features ready for model training

MODELLING

Objective

- To train and evaluate multiple classification models for accurate sentiment prediction of tweets (Positive, Neutral, Negative).

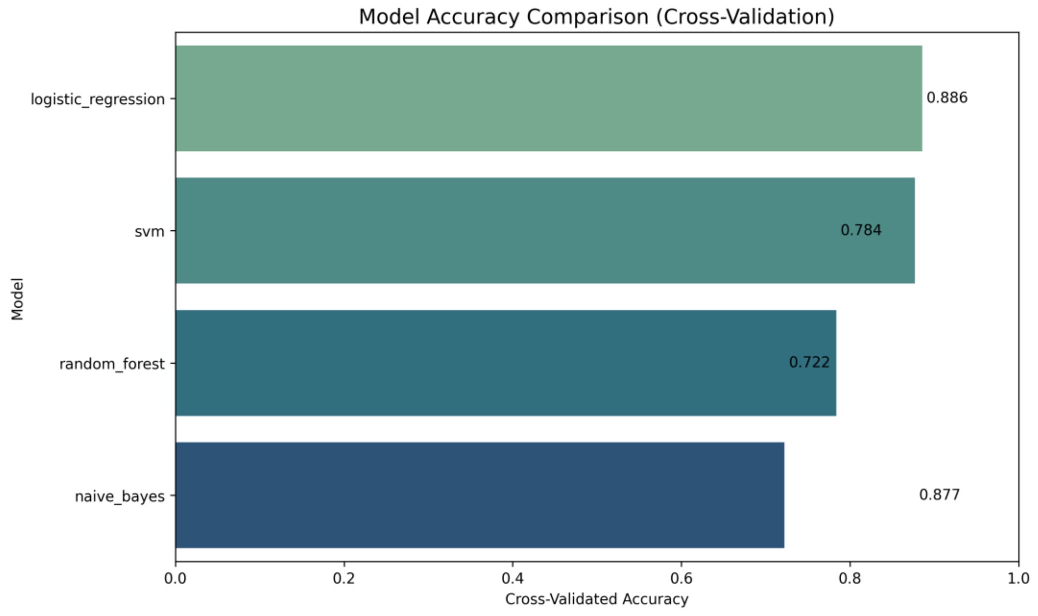
Models Evaluated

- We trained and fine-tuned the following models using GridSearchCV with 5-fold cross-validation:
  1. Logistic Regression
  2. Multinomial Naive Bayes
  3. Linear Support Vector Classifier (LinearSVC)
  4. Random Forest Classifier

Evaluation Metric: Accuracy

Vectorization: TF-IDF on cleaned tweet text

MODELLING



<u>Hyperparameter Tuning with GridSearchCV</u>

- We performed grid search to find optimal hyperparameters for each model.
- The best model was as follows;
    1. Model: Logistic Regression
    2. Best C Value: 1
    3. Cross-Val Accuracy: 88%
- Generally, a higher C means less regularization. Therefore, a value of C=1 indicates balanced performance and generalization.

MODEL INTERPRETABILITY AND EVALUATION

- For interpreting the model, we used the following tools;
  1. **Eli5 -** It highlights positive and negative contributing words in green and red, respectively.
  2. **SHAP (SHapley Additive Explanations)** – It quantifies the impact of each word on the prediction and helps visualize which words pushed the model toward positive, neutral, or negative.
  3. **LIME (Local Interpretable Model-Agnostic Explanations)** - It generates a local approximation of the model around one tweet and identifies top influential features (words) with weights.
- From the model interpretations we observed:
  1. Negative sentiment was driven by words like: "hate", "worst", "disappointed"
  2. Positive sentiment linked to: "love", "great", "amazing"
  3. Some neutral misclassifications occurred in tweets with mixed expressions (e.g., “not bad”) helped validate that the model is learning meaningful linguistic patterns

DEPLOYMENT

Objective

- To deploy the trained sentiment analysis model as an interactive web application that allows users to:
  1. Enter a tweet
  2. Receive a sentiment prediction.

Deployment

- To do this, we used Streamlit, an open-source Python library that allows you to build data science apps.
- App Features:
  1. User Input Interface - Users can type or paste a tweet.
  2. Real-Time Prediction - Predicts whether the sentiment is Positive, Neutral, or Negative.

THANK YOU
ANY
QUESTIONS?

GROUP 2.