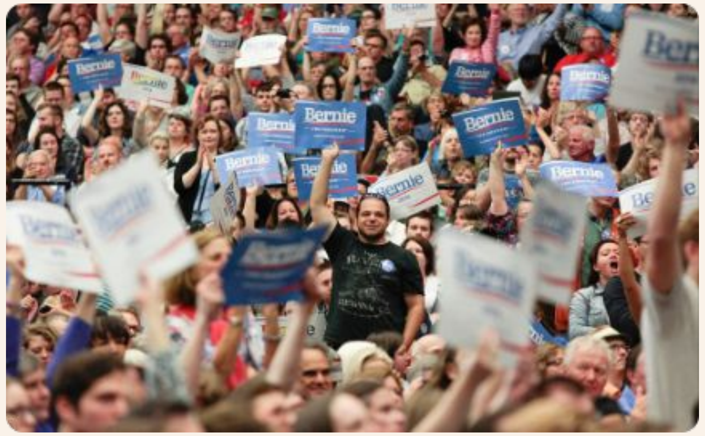
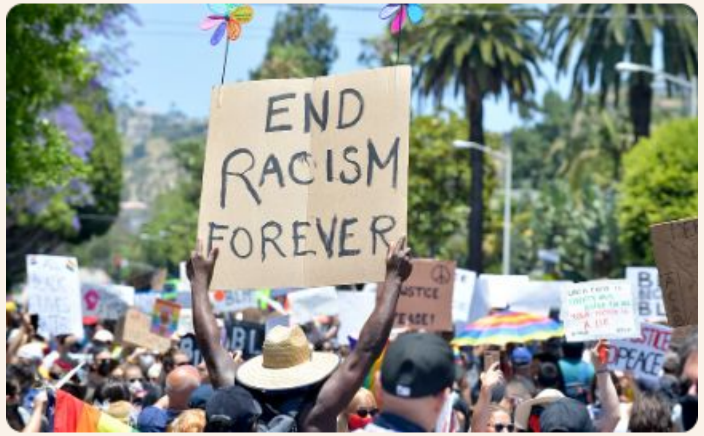
Introduction to Twitter Sentiment Analysis

Twitter sentiment analysis is the process of using natural language processing and machine learning techniques to extract and analyze the emotional tone and opinion expressed in tweets. By understanding the sentiment of tweets, businesses and organizations can gain valuable insights into customer opinions, brand perception, and emerging trends.

Why Sentiment Analysis Matters





Business

Understand how customers perceive your brand and products, and adjust your strategy accordingly.

Politics

Monitor public opinion on candidates and issues, and track sentiment during elections.

Public Action

Identify emerging trends and public sentiment around social issues and causes, and inform policy decisions.

Project Pipeline

1. **Data Collection**

   Import necessary dependencies and read the dataset to begin the analysis.

2. **Exploratory Data Analysis**

   Analyze the dataset to understand its structure, distribution, and any potential issues.

3. **Data Preprocessing**

   Clean, transform, and prepare the data for the machine learning models.

4. **Train-Test Split**

   Divide the dataset into training and testing subsets to evaluate model performance.

5. **Feature Engineering**

   Use techniques like TF-IDF to create numerical representations of the text data.

6. **Model Evaluation**

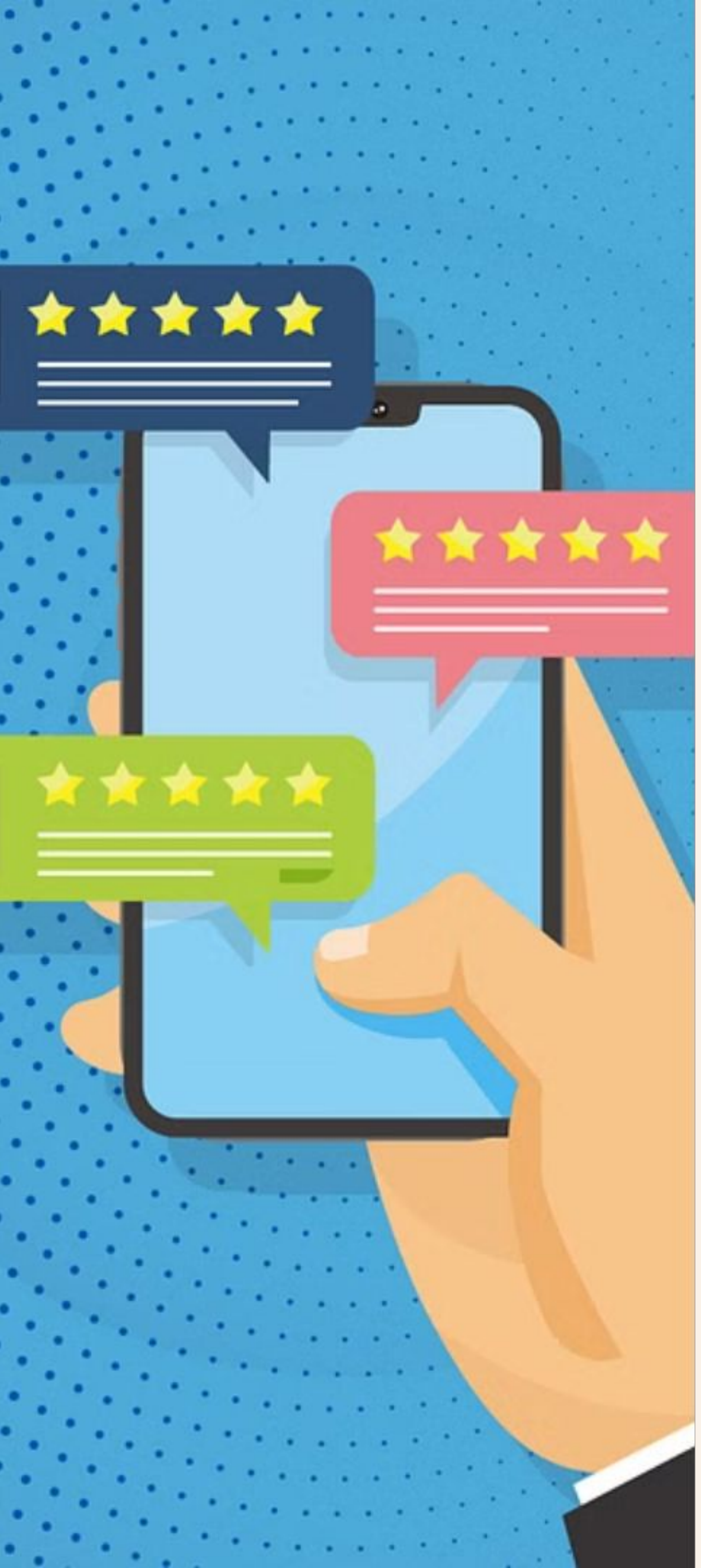
   Implement custom evaluation functions to assess the models on various metrics.

7. **Model Building**

   Train and fine-tune the machine learning models, such as Logistic Regression and Naive Bayes.

8. **Conclusion**

   Summarize the key findings and insights gained from the sentiment analysis project.



Implementing Twitter Sentiment Analysis

In this project, we aim to overcome the challenges of identifying sentiments in tweets by implementing a Twitter sentiment analysis model. We will analyze the sentiment of tweets from the Sentiment140 dataset using a machine learning pipeline. The pipeline involves the use of the following classifiers:

- Logistic Regression
- Bernoulli Naive Bayes

We will also incorporate the use of Term Frequency-Inverse Document Frequency (TF-IDF) for analysis. The performance of these classifiers will be evaluated using accuracy, ROC-AUC Curve, and F1 Scores.

key Libraries for Sentiment Analysis

When it comes to sentiment analysis, having the right libraries at your disposal is crucial. Here are some key libraries that can enhance your sentiment analysis workflow:

1. **Text Processing**

   gensim, nltk, spacy

2. **Data Manipulation**

   pandas, numpy

3. **Visualization**

   seaborn, matplotlib, wordcloud

4. **Machine Learning**

   sklearn

Data Collection and Preprocessing

Data Collection 1

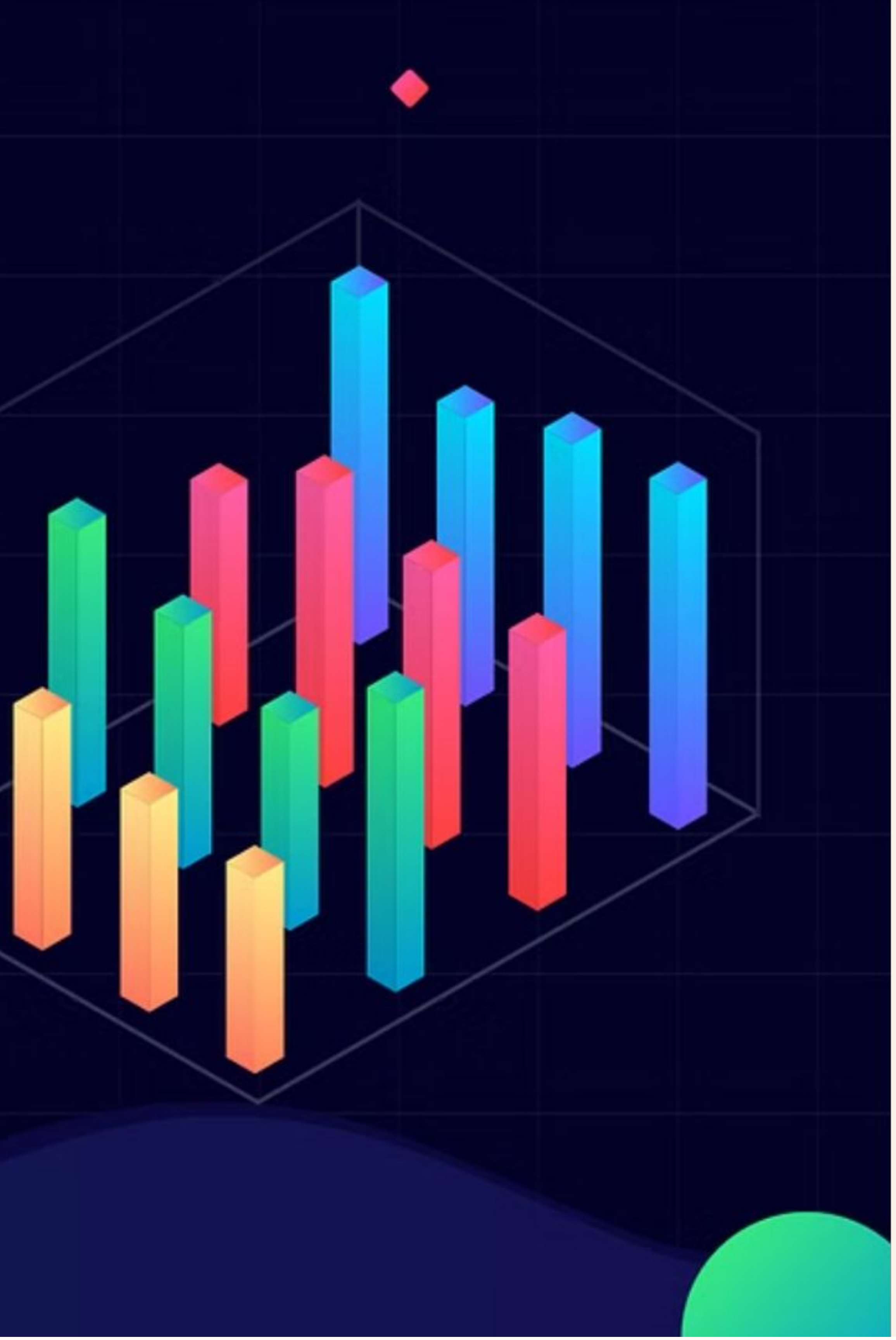
Gather data from kaggle dataset and understanding it.

Data Cleaning 2

Removes lower cases ,mentions, punctuations ,**stopwords** using (nltk,spacy,gensium) and Urls, emails,etc

Text Preprocessing 3

Tokenize, stem, lemmatize, and convert text to lowercase for analysis.



Data Visualization for Sentiment Analysis

Visualizing data plays a crucial role in understanding sentiment analysis results. By using powerful visualization libraries like matplotlib and seaborn, we can create insightful charts, graphs, and plots to depict sentiment trends, word clouds, and more.

Machine Learning Approaches

1

Feature Engineering

Extract relevant features from text.This is achieved using the `TfidfVectorizer` class from scikit-learn, which converts a collection of raw documents into a matrix of TF-IDF features.

2

Model Training

Train supervised models like logistic regression, Bernoulli Naive Bayes on labeled sentiment data.

3

Model Evaluation

Assess model performance using metrics like accuracy score, ROC-AUC curve and confusion matrix with plot.

Evaluating Sentiment Analysis Models: Performance Metrics

Metrics used to evaluate sentiment analysis models:

- Accuracy Score: Measures overall correct predictions.
- ROC-AUC Curve: Graphical representation of model's performance.
- F1 Score: Combines precision and recall.
- Confusion Matrix: Summarizes model's performance.

Data pipeline Architecture

1. **AWS Instance Creation**

   Provision an Amazon EC2 instance to serve as the infrastructure for our data pipeline.

2. **Zookeeper and Kafka Setup**

   Install and configure Apache Zookeeper for distributed coordination. Install and configure Apache Kafka for real-time data streaming.

3. **Bucket Creation**

   Create an Amazon S3 bucket to store and manage data.

4. **Python Data Loading**

   Develop a Python script to load data into the S3 bucket.

5. **Athena Data Querying**

   Set up an Amazon Athena database to query data stored in the S3 bucket using SQL-like queries. Define tables and partitions to optimize query performance.

6. **Power Bi Connection through ODBC**

   Install and configure the Athena ODBC driver on the EC2 instance. Set up an ODBC data source to connect Power BI to Amazon Athena.

Applications and Use Cases

Brand Monitoring

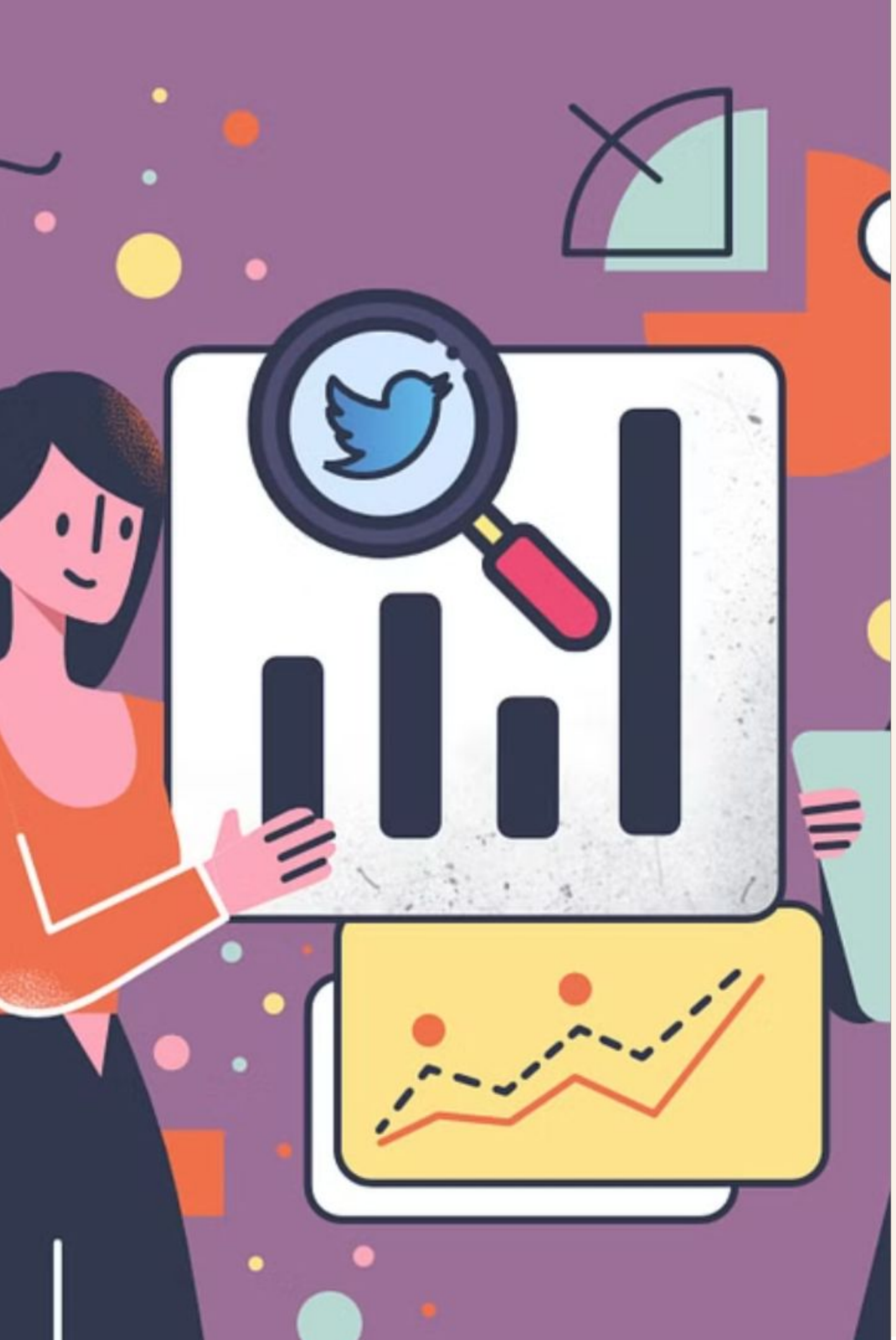
Understand customer perceptions and reactions to products or services.

Customer Service

Identify and respond to customer complaints or praise in real-time.

Marketing Campaigns

Gauge the effectiveness of marketing efforts and adjust strategies accordingly.



Conclusion: Unlocking Insights with Sentiment Analysis

Sentiment analysis provides valuable insights to businesses. By harnessing its power, companies can make data-driven decisions, enhance customer experiences, and drive business growth.

Execution time: When comparing the running time of models, Bernoulli Naive Bayes performs faster with a good accuracy score.

Accuracy: When it comes to model accuracy, logistic regression perform better of the other model, with an accuracy of 74%.

F1-score: The F1 Scores for class 0 and class 1 are:

- For class 0 (negative tweets): Accuracy:

  BNB (=0.73) = < LR (=0.74)

- For class 1 (positive tweets): Accuracy:

  BNB (=0.74) < = LR (=0.75)

AUC score: BNB (=0.63) < = LR (=0.74)

We therefore conclude that logistic regression, Bernoulli Naive Bayes, are the best models for the above dataset.

In our problem statement, logistic regression follows Occam's razor principle, which defines that for a particular problem statement, if the data has no assumptions, then the simplest model works best. Since our dataset has no assumptions and logistic regression is a simple model, this concept holds true for the mentioned dataset, although it took longer to run than the fastest model.