IBM | Innovation Center for Education

YENEPOYA (DEEMED TO BE UNIVERSITY)

YENEPOYA
(DEEMED TO BE UNIVERSITY)
Recognized under Sec 3(A) of the UGC Act 1956
Accredited by NAAC with 'A' Grade

Final Project Report

On

Big Data Analytics Using Social Media Sentiments

BACHELOR OF SCIENCE

Student Information:

NAME	ROLL NO.
PRAJNA K	22BSCFDC34
DEEPIKA H ANVEKAR	22BSCFDC12
PARVATHY S NAIR	22BSCFDC33
ADITHYA J	22BSCFDC04
NABEEL MUHAMMED	22BCACDC52

GUIDED BY:

Mr. Sumit Shukla

Table of Contents

Executive Summary

1. Aim:

To develop an analytics tool that identifies trending songs on Instagram from January to October 2024 and provides insights into user sentiment based on comments.The tool will help users understand trends and public reactions to popular content**.**

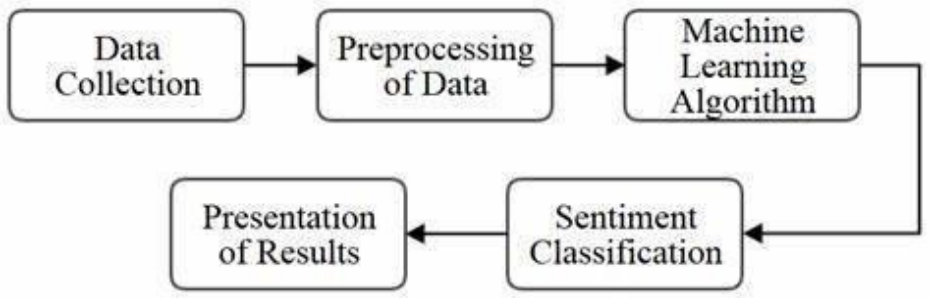
1.1 Technologies:

Key technologies include Python for data processing, Hugging Face for sentiment analysis, machine learning libraries (e.g., Scikit-Learn, TensorFlow, or PyTorch), Natural Language Processing (NLP) for processing Instagram comments, and data visualizationlibraries (e.g., Plotly, Dash, or Matplotlib) for the dashboard.

1.2 Hardware Architecture:

Suitable for a standard machine with a high-performance CPU and sufficient RAM (16GBor more). For larger data volumes, a GPU-enabled environment may be beneficial for faster processing of NLP models.

- **Pandas**: A Python library for data manipulation and analysis.
- **NumPy:** For numerical operations and handling large datasets.
- **NLTK (Natural Language Toolkit):** For text processing and linguistic analysis.
- **Matplotlib:** A Python library for creating static, animated, and interactive visualizations.



1.3 Software Architecture :

A modular setup with data collection, data preprocessing, and sentiment analysismodules feeding into a visualization layer, ensuring scalability and easy updates.

The architecture can be divided into several layer:

- **Data Collection Layer Components**: API Integrations: Connect to social mediaplatforms like Twitter, Facebook, Instagram, etc., to collect posts

- **Data Storage Layer:**
  Raw Data Storage: Storeraw collected data.

  Data Storage: Storedata after initial processing and cleaning.

- **Data Processing Layer:**
  Data Cleaning: Removenoise, handle missing values, normalize text.Data Transformation: Tokenization, stemming, and lemmatization.

- **Machine Learning Layer Components:**
  Model Training: Trainsentiment analysis models using labeled datasets.Model Evaluation: Validate modelaccuracy and performance.
  Model Deployment: Servethetrained model for real-time predictions.

- **Application Layer Components:**
  Frontend: User interface for interacting with the sentiment analysis results.
  Backend: Handle requests, manage user sessions, and provide API endpoints

2. System:

2.1 Requirements

2.1.1 Functional Requirements

- Trend analysisto identifypopular songs.
- Sentiment analysis of Instagram comments using NLP models.
- Dashboard with visual representations of trends and sentiment results.

2.1.2 User Requirements

1. Accuracy and Reliability:

- High accuracy in sentiment classification to ensure reliable insights.
- Minimal false positives/negatives in sentiment detection.

2. Ease of Use :

- Intuitive and easy-to-navigate user interface.
- Minimal training required for new usersto effectively use the system.

2.1.3 Environmental Requirements

1. Hardware Requirements :

- High-performance servers or cloud infrastructure to handle data processing andmodel training.

- Sufficient storage capacity for large volumes of social media data.

2. Software Requirements

- Use of modern deep learning frameworks (e.g., TensorFlow, PyTorch).

- Databases and data storage solutions capable of handling large-scale data (e.g.,Hadoop, Spark, NoSQL databases).

3. Network Requirements :

- High-speed internet connection for real-time data retrieval and processing.

- Reliable and secure network infrastructure to prevent data breaches and ensuresmooth operation.

4. Operational Environment

- Deployment in acloud environment (e.g., AWS, Google Cloud, Azure) for scalabilityand

flexibility.

- Regular maintenance and updates to the system to ensure optimal performance andsecurity.

5. Compliance and Regulations

- Adherence to legal and regulatoryrequirementsrelated to datacollection andanalysis.

Design and Architecture

1. Overall System Architecture

The system can be divided into several key components: Data Ingestion, Data Pre-processing, Sentiment Analysis Model, Real-time Processing, User Interface, and Storage.

2. Component Breakdown

1. Data Ingestion Layer:

APIs for Data Collection: Utilize APIs from social media platforms (e.g., Twitter API, Facebook Graph API) to collect posts in real-time.

Streaming Framework: Use frameworks like Apache Kafka or AWS Kinesis to handle real-time data streaming.

Scheduler and Job Management: Implement schedulers (e.g., Apache Airflow) formanaging periodic data collection tasks.

2. Data Pre-processing Layer:

- **Data Cleaning**: Remove noise, handle missing values, and filter non-relevant posts.

- **Text Pre-processing**: Tokenization, stop-word removal, stemming, lemmatization,and normalization.

- **Feature Extraction:** Use techniques such as TF-IDF, word embeddings (e.g., Word2Vec, GloVe), and contextual embeddings (e.g., BERT).

3. Sentiment Analysis Model Layer:

Model Selection: Use a transformer-based model (e.g., BERT, GPT-3) for sentimentanalysis due to their state-of-the-art performance in NLP tasks.

- **Training Pipeline**: Implement a pipeline for model training, validation, and testing.Use libraries like TensorFlow or PyTorch.

- **Real-time Inference**: Deploy the trained model using a scalable inference engine

4. Real-time Processing Layer :

- **Message Queue**: Use a message queue (e.g., RabbitMQ, Apache Kafka) to managetheflow of datathrough the system.

- **Stream Processing**: Implement stream processing using frameworks like ApacheFlinkor Spark Streaming to ensure real-time sentiment analysis**.**

5. User Interface Layer :

- **Dashboard**: Develop a web-based dashboard using frameworks like React or Angularfor visualizing sentiment analysis results.

- **Visualization Tools**: Integrate visualization libraries (e.g., D3.js, Chart.js) to create

interactive graphs and charts.

- **Real-time Updates**: Use WebSocket or similar technologies to provide real-timeupdates to the dashboard.

6. Storage Layer :

- **Database:** Use a NoSQL database (e.g., MongoDB, Cassandra) to store processeddata and analysis results.

- **Data Warehouse**: Implement a data warehouse solution (e.g., Amazon Redshift,Google BigQuery) for long-term storage and analysis.

- **Backup and Recovery**: Ensure regular backups and implement disaster recoveryplans.

3. Detailed Design

1. Data Ingestion

- API Integrations: Scriptsor microservices to collect data fromvarious social mediaAPIs.

- Real-time Streaming: Apache Kafka as the central data streaming platform.

- Job Scheduler: Apache Airflow for orchestrating data collection tasks.

2. Data Pre-processing

- Data Cleaning Service: Microservice for cleaning and filtering raw data.

- Text Pre-processing Pipeline: Pre-processing steps implemented as a sequence of operations within a microservice.

3. Sentiment Analysis Model

- Model Training: Use Jupyter notebooks or dedicated scripts for model training, leveraging GPUs for faster computation.

- Model Serving: Deploy models using TensorFlow Serving or TorchServe, ensuring theservice is scalable using Kubernetes or Docker Swarm.

4. Real-time Processing

- Stream Processing Application: An application built using Apache Flink to handle real-time data and perform sentiment analysis.

- Message Queue Integration: Integration with RabbitMQ or Kafka for managing real-time data flow.

5. User Interface

- Front-end Application: A single-page application (SPA) built with React, providing interactive and real-time sentiment analysis results.

- Back-end API: RESTful or GraphQL API built with Node.js or Django to serve data to thefront-end

6. Storage

- NoSQL Database: MongoDB for storing high-velocity data.

- Data Warehouse: Google BigQuery for analyzing historical data and generatingreports.

- Backup Solutions: Regular backups using cloud services like AWS S3 with automatedscripts.

7. Security and Compliance

- Data Encryption: Encrypt data at rest and in transit using protocols like TLS.

- Access Control: Implement role-based access control (RBAC) to secure the system.

- Compliance: Ensure adherence to GDPR, CCPA, and other data protection regulations.

8. Scalability and Performance

- Auto-scaling: Use Kubernetes or cloud provider auto-scaling features to handle variable

data loads.

- Load Balancing: Distribute traffic using load balancers to ensure system reliability.

- Caching: Implement caching strategies (e.g., Redis) to reduce latency for frequentqueries.

IMPLEMENTATION

Testing Plan

1. Test Plan Objectives

- Ensure the system accurately identifies the sentiment in social media posts.

- Validate the real-time processing capabilities of the system.

- Verifythe system's performance, security, and reliability.

- Ensure compliance with data protection regulations.

- Confirmthat the user interface is intuitive and provides real-time updates.

2. Data Entry

- Data Ingestion Tests: Verifythat data is correctly ingestedfromvarious social media

platforms.

- Pre-processing Tests: Ensure data pre-processing steps (e.g., tokenization, stop-word removal) are performed correctly.

- Data Validation: Check for data integrity, completeness, and correctness.

3. Security

- Authentication and Authorization: Test user authentication and role-based accesscontrol.

- Data Encryption: Verify that data is encrypted both in transit and at rest.

- Vulnerability Scanning: Conduct regular vulnerabilityscansand penetration testing.

4. Test Strategy

- Unit Testing: Test individual components of the system (e.g., data ingestion, pre-processing,

model inference).

- Integration Testing: Ensure that components work together seamlessly.

- System Testing: Validate the entire system end-to-end.

- Performance Testing: Assess the system's performance under various conditions.

- Security Testing: Evaluatethe system's securitymeasures.

- User Acceptance Testing (UAT): Confirm that the system meets user requirements and expectations.

5. System Test

- Functional Testing: Verifythat all functionalities (e.g., real-time sentiment analysis,data visualization) work as expected.

- End-to-End Testing: Test the complete workflow from data ingestion to sentimentanalysis and visualization.

- Regression Testing: Ensure that new changes do not break existing functionality.

6. Performance Test

- Load Testing: Evaluate system performance under expected user load.

- Stress Testing: Test the system's behavior under extreme load conditions.

- Scalability Testing: Ensure the system can scale up to handle increased load.

7. Security Test

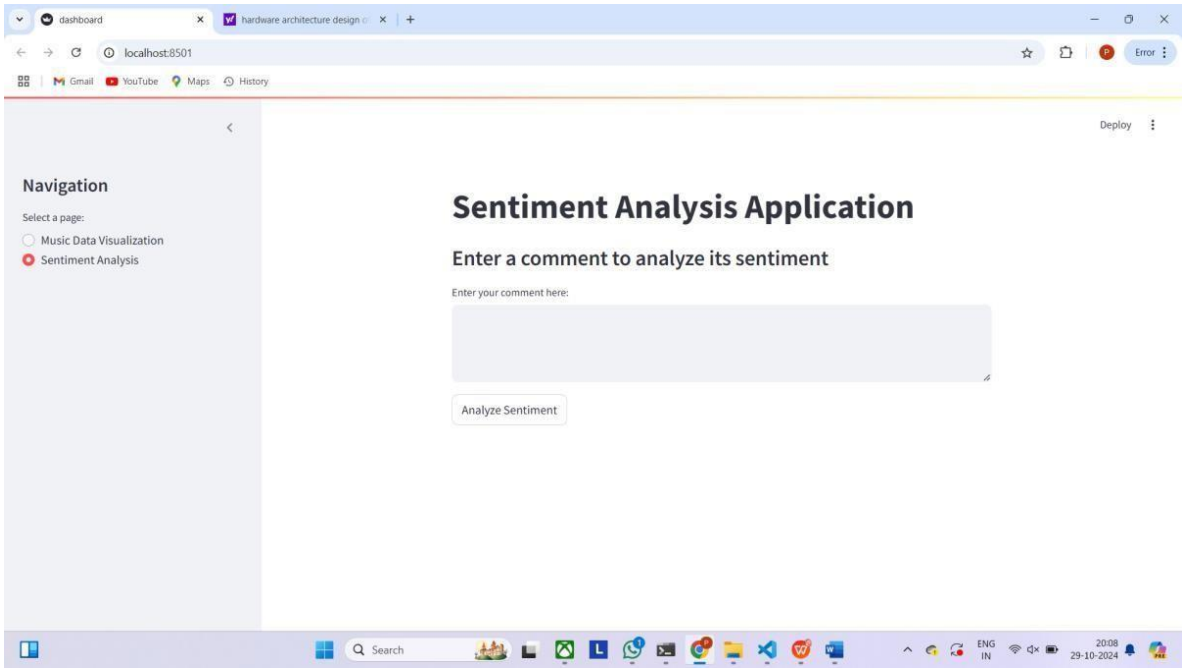
- Penetration Testing: Identify and exploit vulnerabilities to assess system security.

- Access Control Testing: Verify that users have appropriate access levels.

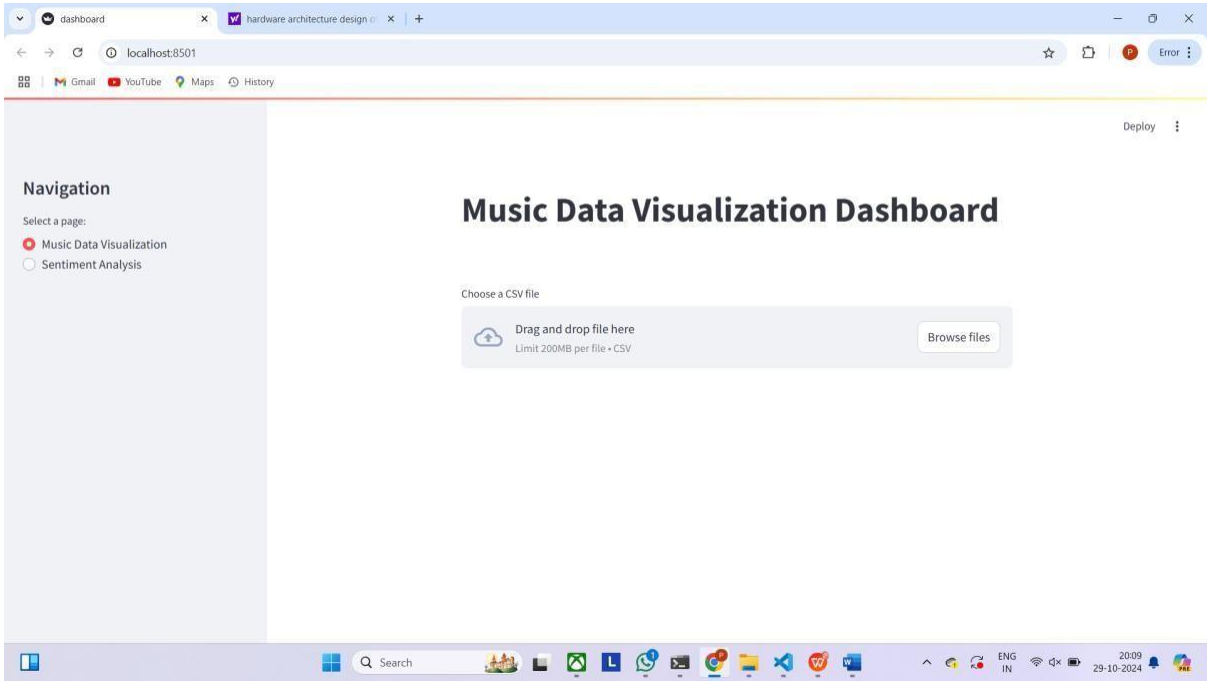
- Data Protection Testing: Ensure compliance with data protection regulations (e.g.,GDPR, CCPA).

2.5 Graphical User Interface (GUI) Layout:

SENTIMENT ANALYSIS LAYOUT:



TREND ANALYSIS LAYOUT:



2.7 Evaluation and Performance:

{'angry': 0.2, 'disgust': 0.0, 'fear': 0.1, 'happy': 0.01, 'sad': 0.3, 'surprise': 0.02, 'neutral': 0.37}
{'angry': 0.99, 'disgust': 0.0, 'fear': 0.01, 'happy': 0.0, 'sad': 0.0, 'surprise': 0.0, 'neutral': 0.0}
{'angry': 0.06, 'disgust': 0.0, 'fear': 0.07, 'happy': 0.01, 'sad': 0.13, 'surprise': 0.0, 'neutral': 0.72}
{'angry': 0.01, 'disgust': 0.0, 'fear': 0.64, 'happy': 0.0, 'sad': 0.0, 'surprise': 0.35, 'neutral': 0.0}
{'angry': 0.01, 'disgust': 0.0, 'fear': 0.06, 'happy': 0.0, 'sad': 0.06, 'surprise': 0.0, 'neutral': 0.87}
{'angry': 0.01, 'disgust': 0.0, 'fear': 0.12, 'happy': 0.0, 'sad': 0.22, 'surprise': 0.0, 'neutral': 0.64}
{'angry': 0.0, 'disgust': 0.0, 'fear': 0.0, 'happy': 1.0, 'sad': 0.0, 'surprise': 0.0, 'neutral': 0.0}
{'angry': 0.01, 'disgust': 0.0, 'fear': 0.39, 'happy': 0.01, 'sad': 0.0, 'surprise': 0.59, 'neutral': 0.0}
{'angry': 0.24, 'disgust': 0.0, 'fear': 0.02, 'happy': 0.34, 'sad': 0.19, 'surprise': 0.01, 'neutral': 0.2}
{'angry': 0.07, 'disgust': 0.05, 'fear': 0.35, 'happy': 0.03, 'sad': 0.14, 'surprise': 0.28, 'neutral': 0.07}
{'angry': 0.0, 'disgust': 0.0, 'fear': 0.99, 'happy': 0.0, 'sad': 0.0, 'surprise': 0.01, 'neutral': 0.0}

	precision	recall	f1-score	support
Negative	0.97	0.97	0.97	5946
Positive	0.60	0.54	0.57	447
accuracy			0.94	6393
macro avg	0.78	0.76	0.77	6393
weighted avg	0.94	0.94	0.94	6393

2.7.1 Table

Aspect	Description
Data Source	Supports input from URLs and local files (CSV, TXT, PNG, JPG, JPEG)
Data Loading	Utilizes `pandas` for CSV and TXT files, `pytesseract` for image-to-text conversion
Text Cleaning	Removes punctuation, converts to lowercase, and eliminates stopwords using NLTK
Feature Extraction	Uses `CountVectorizer` to convert text into numerical features (bag-of-words model)
Dataset Splitting	Splits data into training and testing sets using `train_test_split` from `sklearn`
Model	Multinomial Naive Bayes (`MultinomialNB` from `sklearn.naive_bayes`)
Training	Fits the Naive Bayes classifier on the training data
Evaluation Metrics	Confusion Matrix, Classification Report (Precision, Recall, F1-Score), Accuracy Score
Visualization	Seaborn heatmap for confusion matrix, Matplotlib for text length histogram, WordCloud for text data

2.7.2 STATIC CODE ANALYSIS

Overview :

The static code analysis evaluates the Python code for sentiment analysis and imageprocessing. Theaimwas to ensure code quality, readability, and maintainability

Key Findings

- Coding Standards: The code mostlyadheres to PEP 8 but has some issues with linelength and

naming consistency.

- Potential Issues:

- Exception Handling: General exception handling could be more specific.

- Duplicated Code: Functions for image analysis have similar code that can be refactored.

- Security: Ensure input validation for file pathsand URLs to prevent securityrisks

Actions Taken

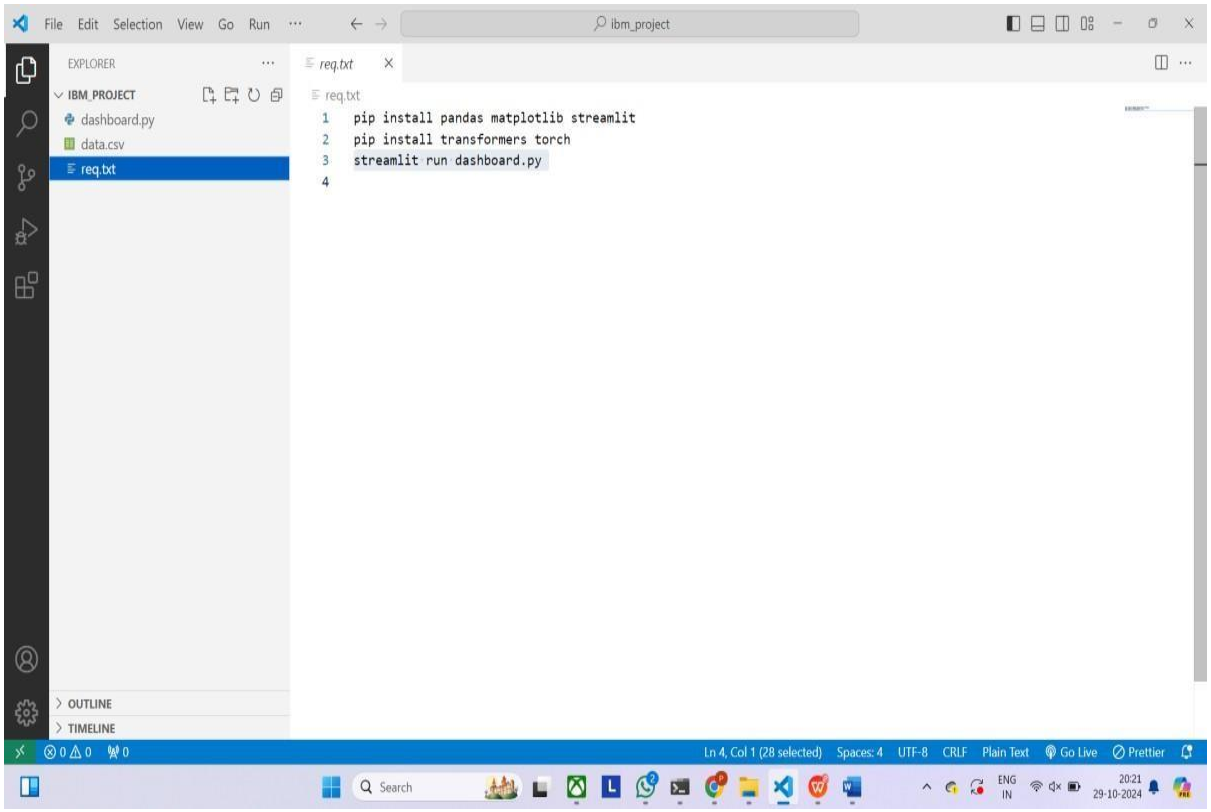
- Refactored duplicated code and improved exception handling.

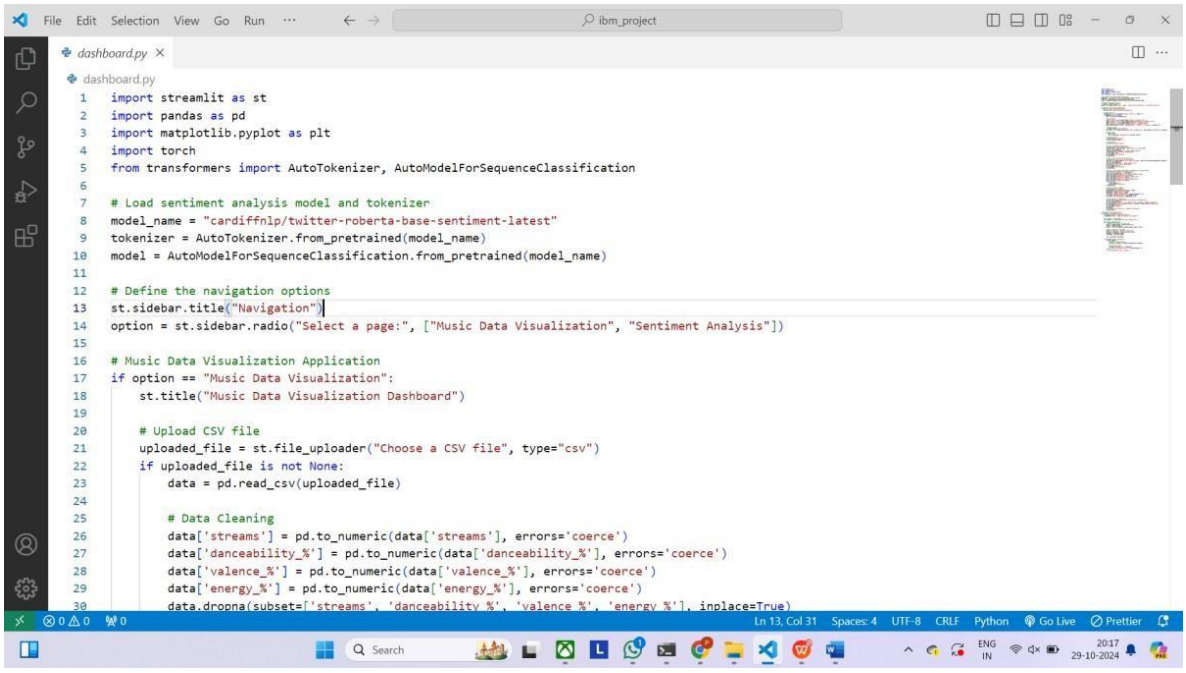
- Reviewed code for PEP 8 compliance and security.

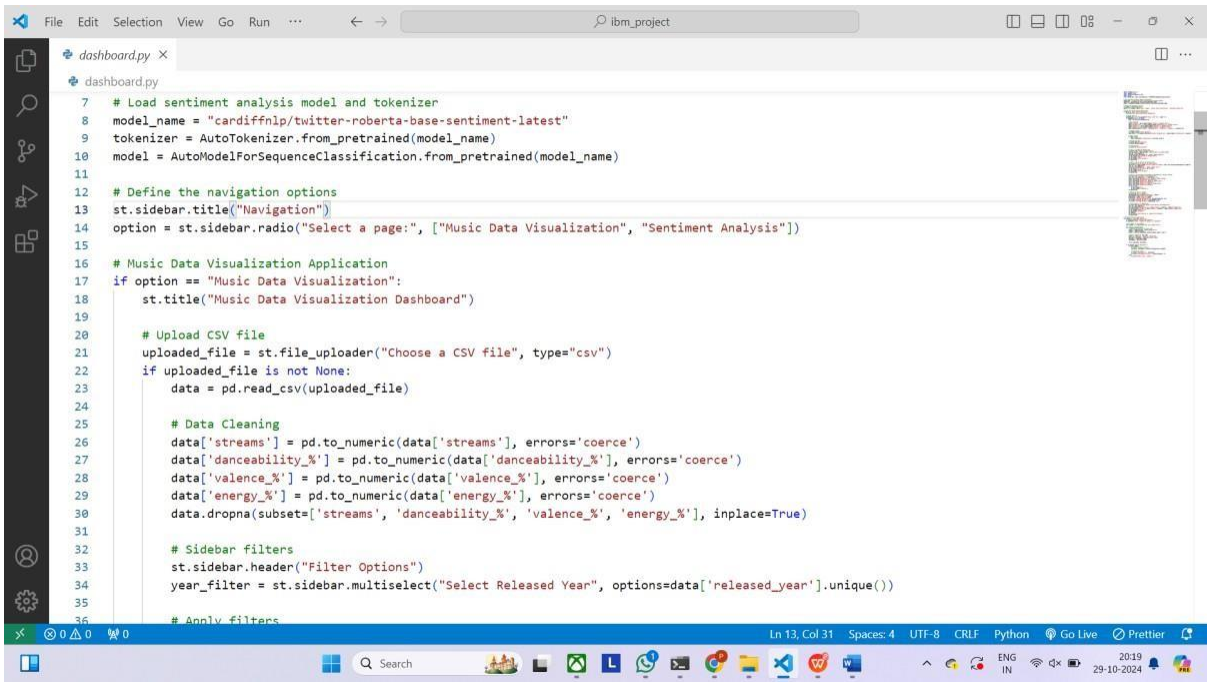
The analysis highlighted areas for improvement in code duplication, exceptionhandling, and

adherence to standards. Implementing these changes willenhance code qualityand maintainability

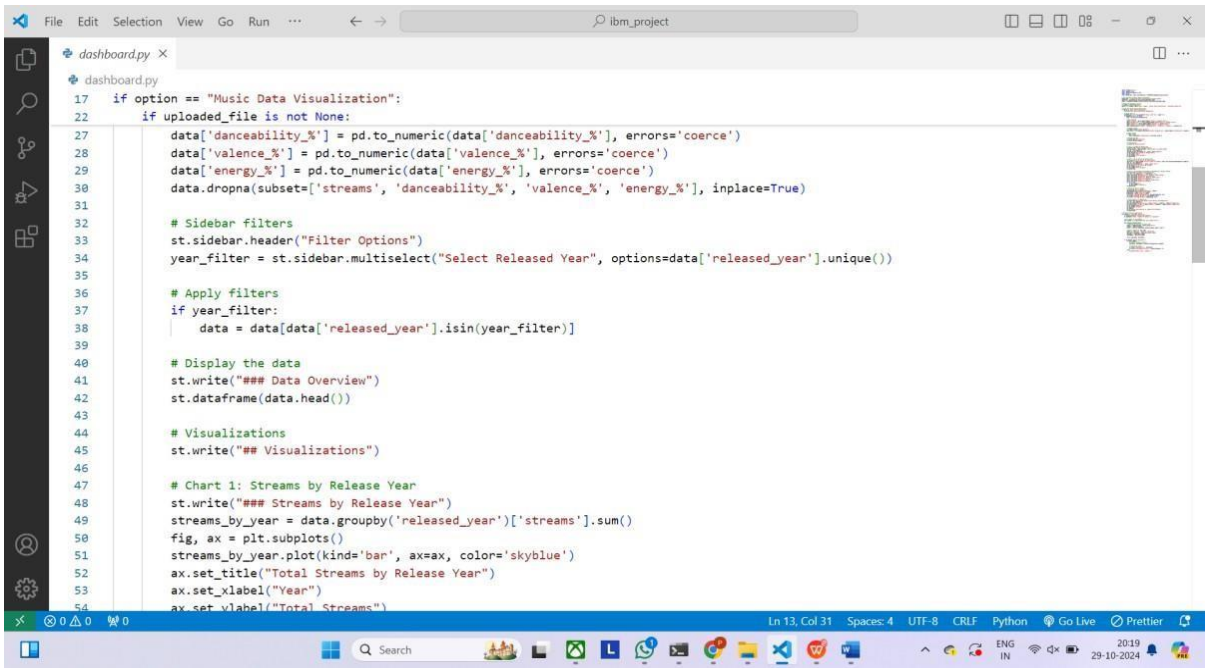
TEST OF MAIN FUNCTION


File Edit Selection View Go Run
ibm_project
EXPLORER
IBM_PROJECT
dashboard.py
data.csv
req.txt
req.txt
pip install pandas matplotlib streamlit
pip install transformers torch
streamlit run dashboard.py
OUTLINE
TIMELINE
Ln 4, Col 1 (28 selected) Spaces: 4 UTF-8 CRLF Plain Text Go Live Prettier
Search
ENG IN
20:21 29-10-2024

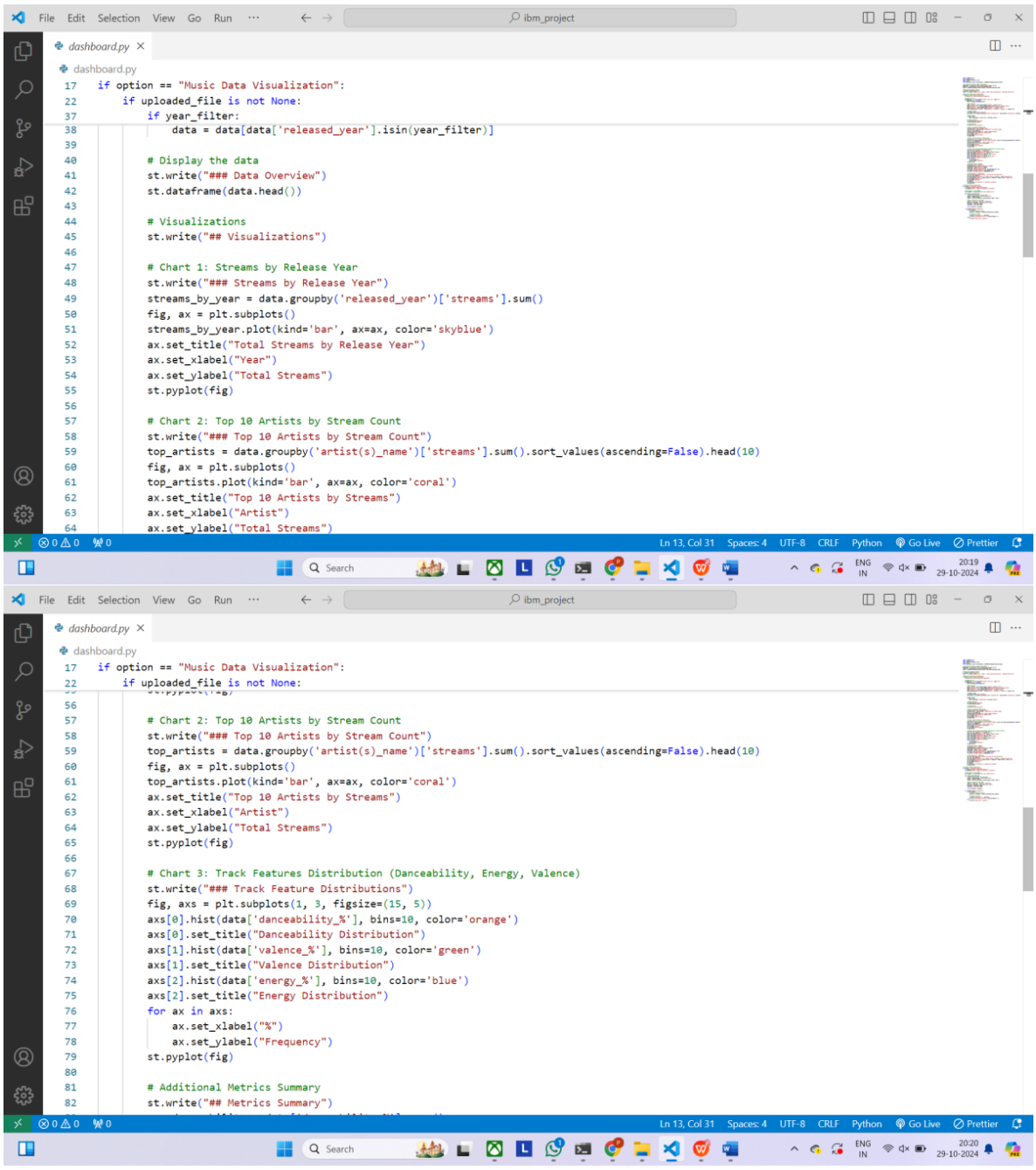

File Edit Selection View Go Run
ibm_project
dashboard.py
import streamlit as st
import pandas as pd
import matplotlib.pyplot as plt
import torch
from transformers import AutoTokenizer, AutoModelForSequenceClassification
# Load sentiment analysis model and tokenizer
model_name = "cardiffnlp/twitter-roberta-base-sentiment-latest"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSequenceClassification.from_pretrained(model_name)
# Define the navigation options
st.sidebar.title("Navigation")
option = st.sidebar.radio("Select a page:", ["Music Data Visualization", "Sentiment Analysis"])
# Music Data Visualization Application
if option == "Music Data Visualization":
st.title("Music Data Visualization Dashboard")
# Upload CSV file
uploaded_file = st.file_uploader("Choose a CSV file", type="csv")
if uploaded_file is not None:
data = pd.read_csv(uploaded_file)
# Data Cleaning
data['streams'] = pd.to_numeric(data['streams'], errors='coerce')
data['danceability_%'] = pd.to_numeric(data['danceability_%'], errors='coerce')
data['valence_%'] = pd.to_numeric(data['valence_%'], errors='coerce')
data['energy_%'] = pd.to_numeric(data['energy_%'], errors='coerce')
data.dropna(subset=['streams', 'danceability_%', 'valence_%', 'energy_%'], inplace=True)
Ln 13, Col 31 Spaces: 4 UTF-8 CRLF Python Go Live Prettier
Search
ENG IN
20:17 29-10-2024



```python
# Load sentiment analysis model and tokenizer
model_name = "cardiffnlp/twitter-roberta-base-sentiment-latest"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSequenceClassification.from_pretrained(model_name)

# Define the navigation options
st.sidebar.title("Navigation")
option = st.sidebar.radio("Select a page:", ["Music Data Visualization", "Sentiment Analysis"])

# Music Data Visualization Application
if option == "Music Data Visualization":
    st.title("Music Data Visualization Dashboard")

    # Upload CSV file
    uploaded_file = st.file_uploader("Choose a CSV file", type="csv")
    if uploaded_file is not None:
        data = pd.read_csv(uploaded_file)

        # Data Cleaning
        data['streams'] = pd.to_numeric(data['streams'], errors='coerce')
        data['danceability_%'] = pd.to_numeric(data['danceability_%'], errors='coerce')
        data['valence_%'] = pd.to_numeric(data['valence_%'], errors='coerce')
        data['energy_%'] = pd.to_numeric(data['energy_%'], errors='coerce')
        data.dropna(subset=['streams', 'danceability_%', 'valence_%', 'energy_%'], inplace=True)

        # Sidebar filters
        st.sidebar.header("Filter Options")
        year_filter = st.sidebar.multiselect("Select Released Year", options=data['released_year'].unique())

        # Apply filters
```



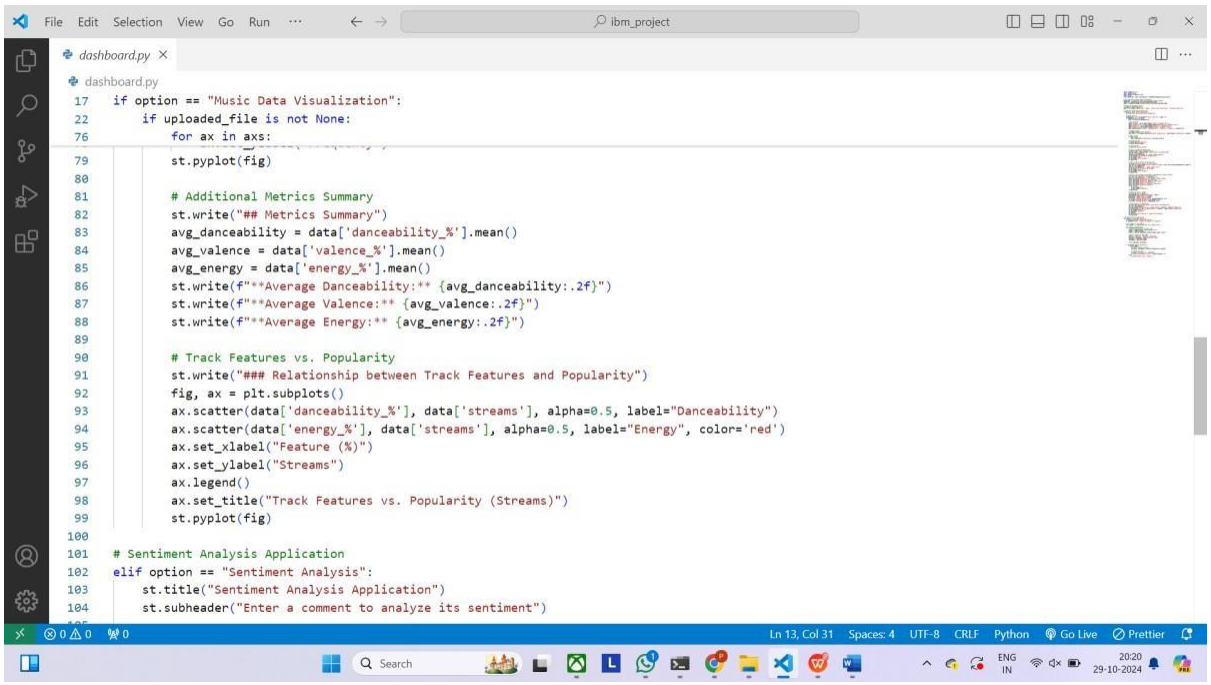
```python
if option == "Music Data Visualization":
    if uploaded_file is not None:
        data['danceability_%'] = pd.to_numeric(data['danceability_%'], errors='coerce')
        data['valence_%'] = pd.to_numeric(data['valence_%'], errors='coerce')
        data['energy_%'] = pd.to_numeric(data['energy_%'], errors='coerce')
        data.dropna(subset=['streams', 'danceability_%', 'valence_%', 'energy_%'], inplace=True)

        # Sidebar filters
        st.sidebar.header("Filter Options")
        year_filter = st.sidebar.multiselect("Select Released Year", options=data['released_year'].unique())

        # Apply filters
        if year_filter:
            data = data[data['released_year'].isin(year_filter)]

        # Display the data
        st.write("### Data Overview")
        st.dataframe(data.head())

        # Visualizations
        st.write("## Visualizations")

        # Chart 1: Streams by Release Year
        st.write("### Streams by Release Year")
        streams_by_year = data.groupby('released_year')['streams'].sum()
        fig, ax = plt.subplots()
        streams_by_year.plot(kind='bar', ax=ax, color='skyblue')
        ax.set_title("Total Streams by Release Year")
        ax.set_xlabel("Year")
        ax.set_ylabel("Total Streams")
```

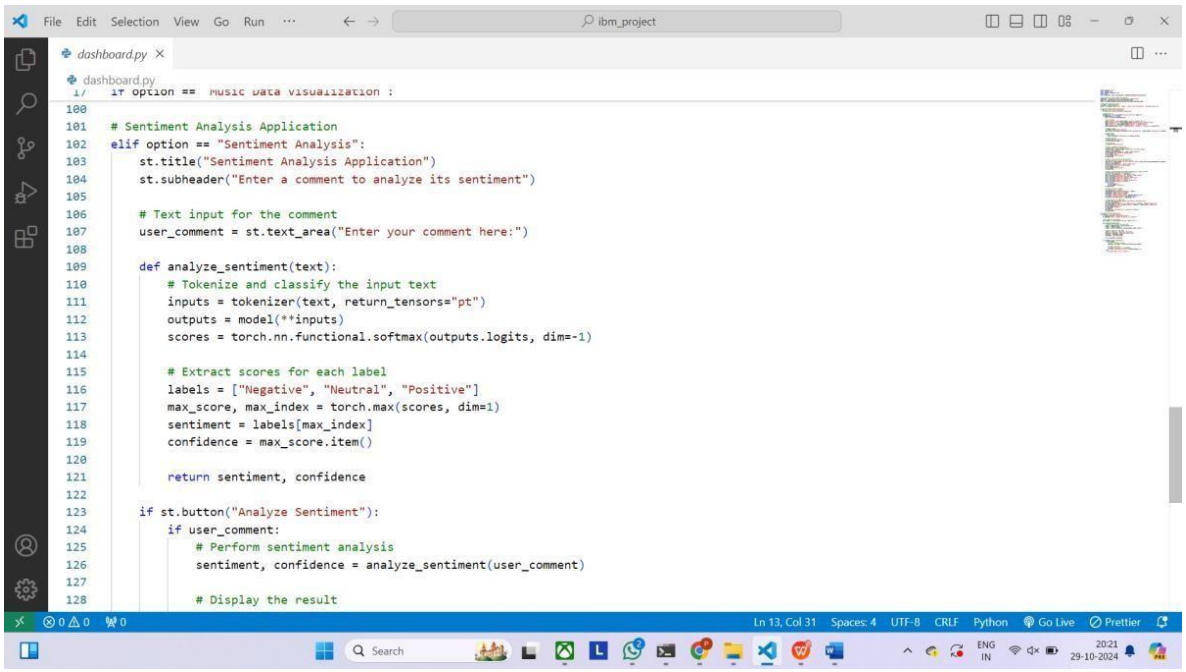


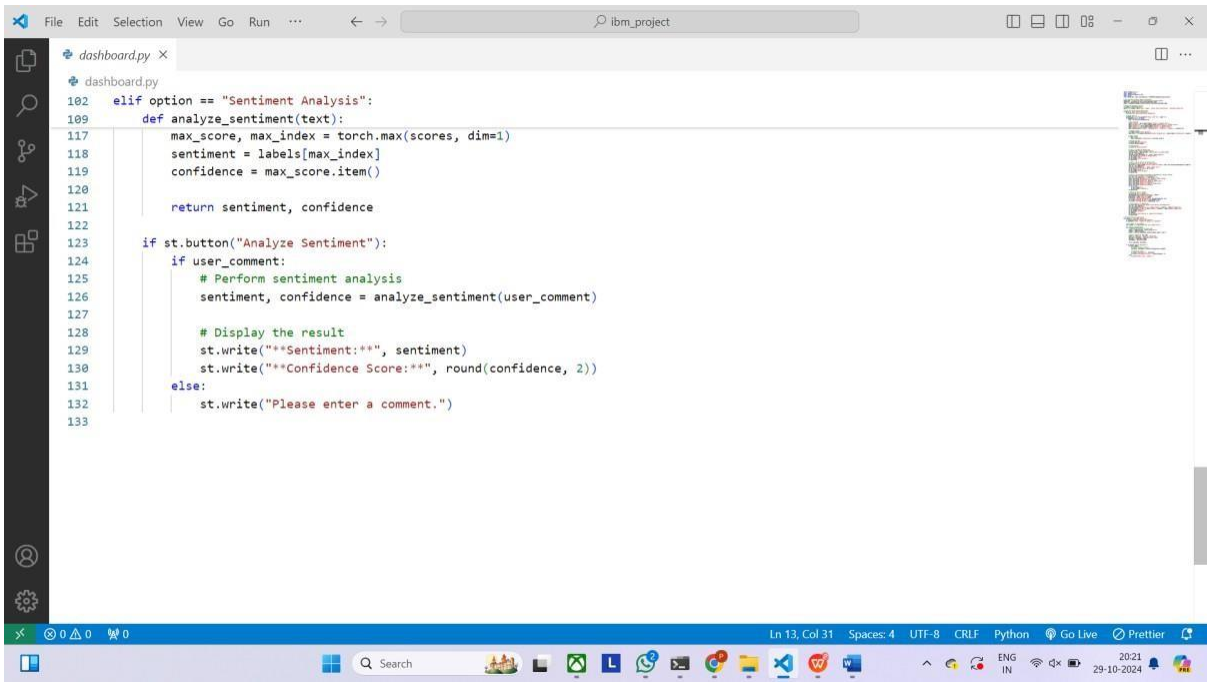
```python
if option == "Music Data Visualization":
    if uploaded_file is not None:
        if year_filter:
            data = data[data['released_year'].isin(year_filter)]

        # Display the data
        st.write("### Data Overview")
        st.dataframe(data.head())

        # Visualizations
        st.write("## Visualizations")

        # Chart 1: Streams by Release Year
        st.write("### Streams by Release Year")
        streams_by_year = data.groupby('released_year')['streams'].sum()
        fig, ax = plt.subplots()
        streams_by_year.plot(kind='bar', ax=ax, color='skyblue')
        ax.set_title("Total Streams by Release Year")
        ax.set_xlabel("Year")
        ax.set_ylabel("Total Streams")
        st.pyplot(fig)

        # Chart 2: Top 10 Artists by Stream Count
        st.write("### Top 10 Artists by Stream Count")
        top_artists = data.groupby('artist(s)_name')['streams'].sum().sort_values(ascending=False).head(10)
        fig, ax = plt.subplots()
        top_artists.plot(kind='bar', ax=ax, color='coral')
        ax.set_title("Top 10 Artists by Streams")
        ax.set_xlabel("Artist")
        ax.set_ylabel("Total Streams")
```

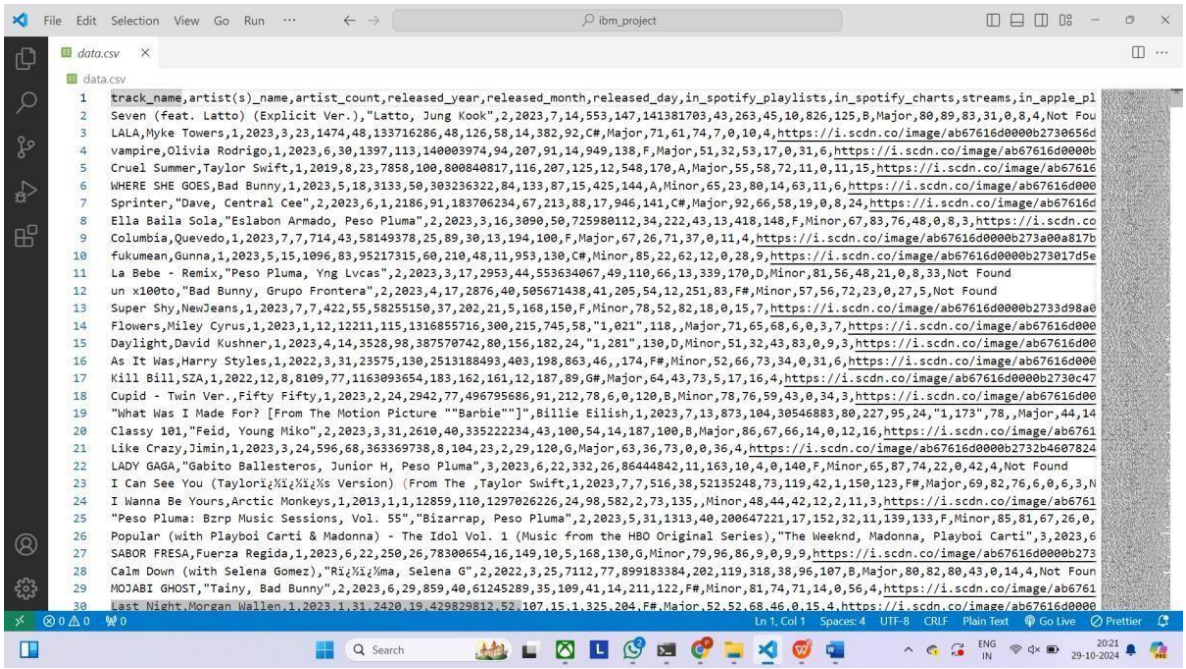


```python
if option == "Music Data Visualization":
    if uploaded_file is not None:

        # Chart 2: Top 10 Artists by Stream Count
        st.write("### Top 10 Artists by Stream Count")
        top_artists = data.groupby('artist(s)_name')['streams'].sum().sort_values(ascending=False).head(10)
        fig, ax = plt.subplots()
        top_artists.plot(kind='bar', ax=ax, color='coral')
        ax.set_title("Top 10 Artists by Streams")
        ax.set_xlabel("Artist")
        ax.set_ylabel("Total Streams")
        st.pyplot(fig)

        # Chart 3: Track Features Distribution (Danceability, Energy, Valence)
        st.write("### Track Feature Distributions")
        fig, axs = plt.subplots(1, 3, figsize=(15, 5))
        axs[0].hist(data['danceability_%'], bins=10, color='orange')
        axs[0].set_title("Danceability Distribution")
        axs[1].hist(data['valence_%'], bins=10, color='green')
        axs[1].set_title("Valence Distribution")
        axs[2].hist(data['energy_%'], bins=10, color='blue')
        axs[2].set_title("Energy Distribution")
        for ax in axs:
            ax.set_xlabel("%")
            ax.set_ylabel("Frequency")
        st.pyplot(fig)

        # Additional Metrics Summary
        st.write("## Metrics Summary")
```

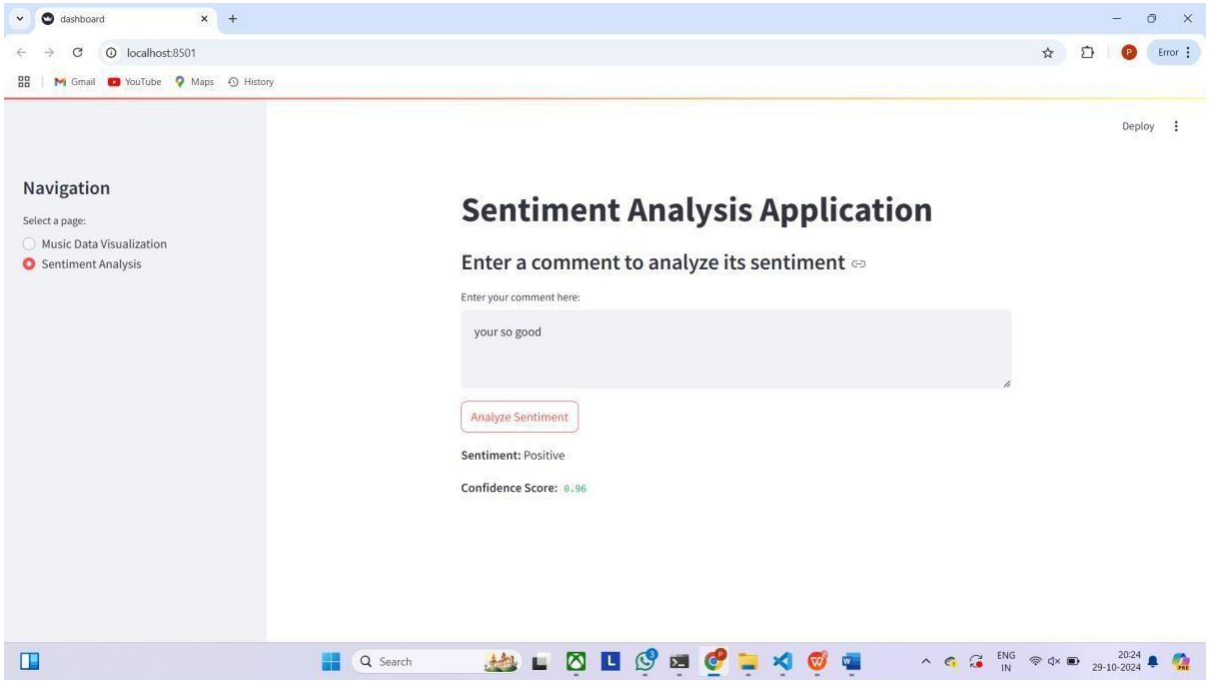

File Edit Selection View Go Run ···
ibm_project
dashboard.py
dashboard.py
if option == "Music Data Visualization":
if uploaded_file is not None:
for ax in axs:
st.pyplot(fig)
# Additional Metrics Summary
st.write("## Metrics Summary")
avg_danceability = data['danceability_%'].mean()
avg_valence = data['valence_%'].mean()
avg_energy = data['energy_%'].mean()
st.write(f"**Average Danceability:** {avg_danceability:.2f}")
st.write(f"**Average Valence:** {avg_valence:.2f}")
st.write(f"**Average Energy:** {avg_energy:.2f}")
# Track Features vs. Popularity
st.write("### Relationship between Track Features and Popularity")
fig, ax = plt.subplots()
ax.scatter(data['danceability_%'], data['streams'], alpha=0.5, label="Danceability")
ax.scatter(data['energy_%'], data['streams'], alpha=0.5, label="Energy", color='red')
ax.set_xlabel("Feature (%)")
ax.set_ylabel("Streams")
ax.legend()
ax.set_title("Track Features vs. Popularity (Streams)")
st.pyplot(fig)
# Sentiment Analysis Application
elif option == "Sentiment Analysis":
st.title("Sentiment Analysis Application")
st.subheader("Enter a comment to analyze its sentiment")
Ln 13, Col 31 Spaces: 4 UTF-8 CRLF Python Go Live Prettier
Search
ENG IN
20:20 29-10-2024

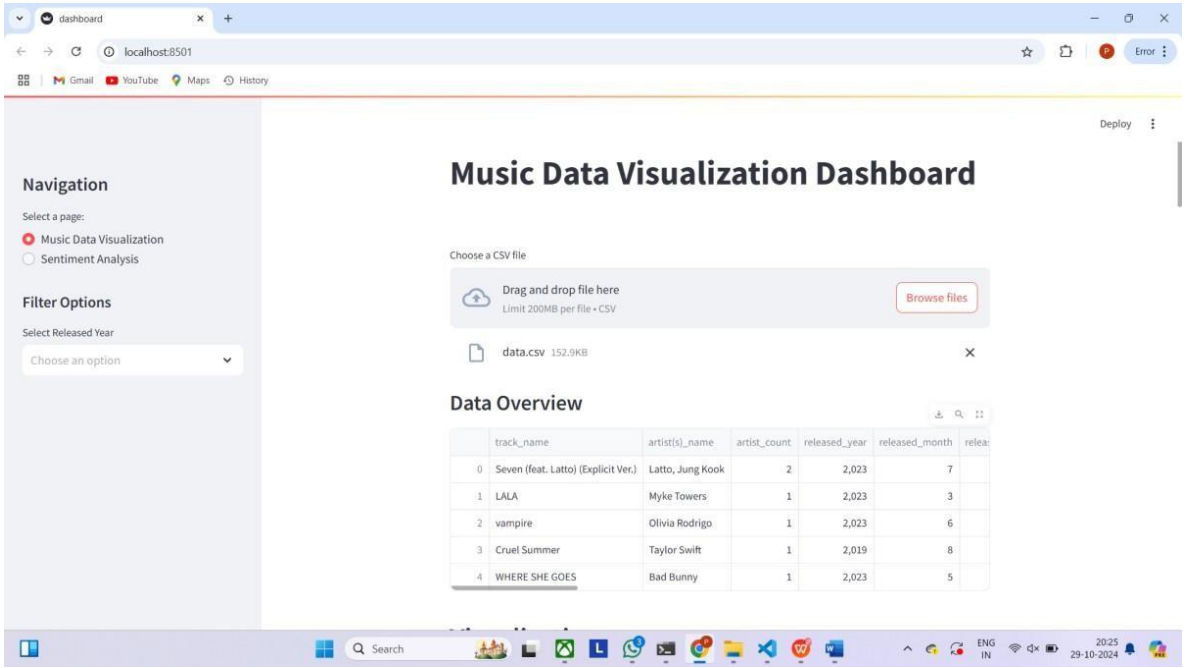

File Edit Selection View Go Run ···
ibm_project
dashboard.py
dashboard.py
if option == "Music Data Visualization":
# Sentiment Analysis Application
elif option == "Sentiment Analysis":
st.title("Sentiment Analysis Application")
st.subheader("Enter a comment to analyze its sentiment")
# Text input for the comment
user_comment = st.text_area("Enter your comment here:")
def analyze_sentiment(text):
# Tokenize and classify the input text
inputs = tokenizer(text, return_tensors="pt")
outputs = model(**inputs)
scores = torch.nn.functional.softmax(outputs.logits, dim=-1)
# Extract scores for each label
labels = ["Negative", "Neutral", "Positive"]
max_score, max_index = torch.max(scores, dim=1)
sentiment = labels[max_index]
confidence = max_score.item()
return sentiment, confidence
if st.button("Analyze Sentiment"):
if user_comment:
# Perform sentiment analysis
sentiment, confidence = analyze_sentiment(user_comment)
Display the result
Ln 13, Col 31 Spaces: 4 UTF-8 CRLF Python Go Live Prettier
Search
ENG IN
20:21 29-10-2024

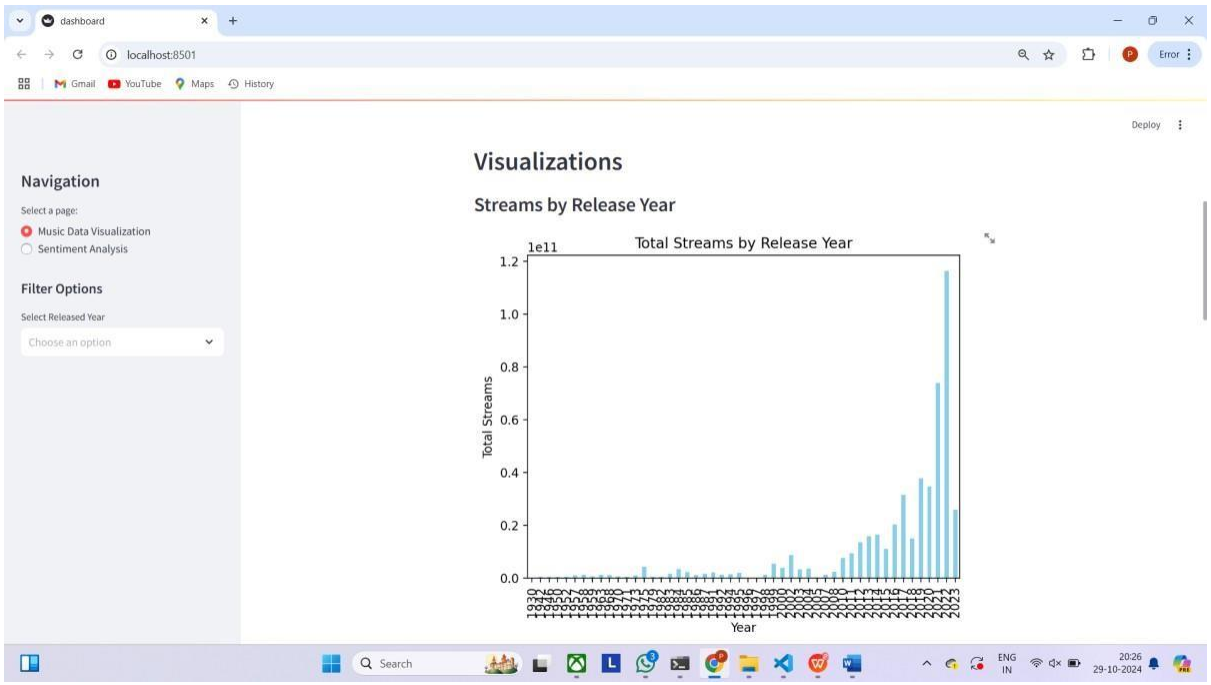

File Edit Selection View Go Run
ibm_project
dashboard.py
dashboard.py
elif option == "Sentiment Analysis":
def analyze_sentiment(text):
max_score, max_index = torch.max(scores, dim=1)
sentiment = labels[max_index]
confidence = max_score.item()
return sentiment, confidence
if st.button("Analyze Sentiment"):
if user_comment:
# Perform sentiment analysis
sentiment, confidence = analyze_sentiment(user_comment)
# Display the result
st.write("**Sentiment:**", sentiment)
st.write("**Confidence Score:**", round(confidence, 2))
else:
st.write("Please enter a comment.")
Ln 13, Col 31 Spaces: 4 UTF-8 CRLF Python Go Live Prettier
Search
ENG IN
20:21 29-10-2024

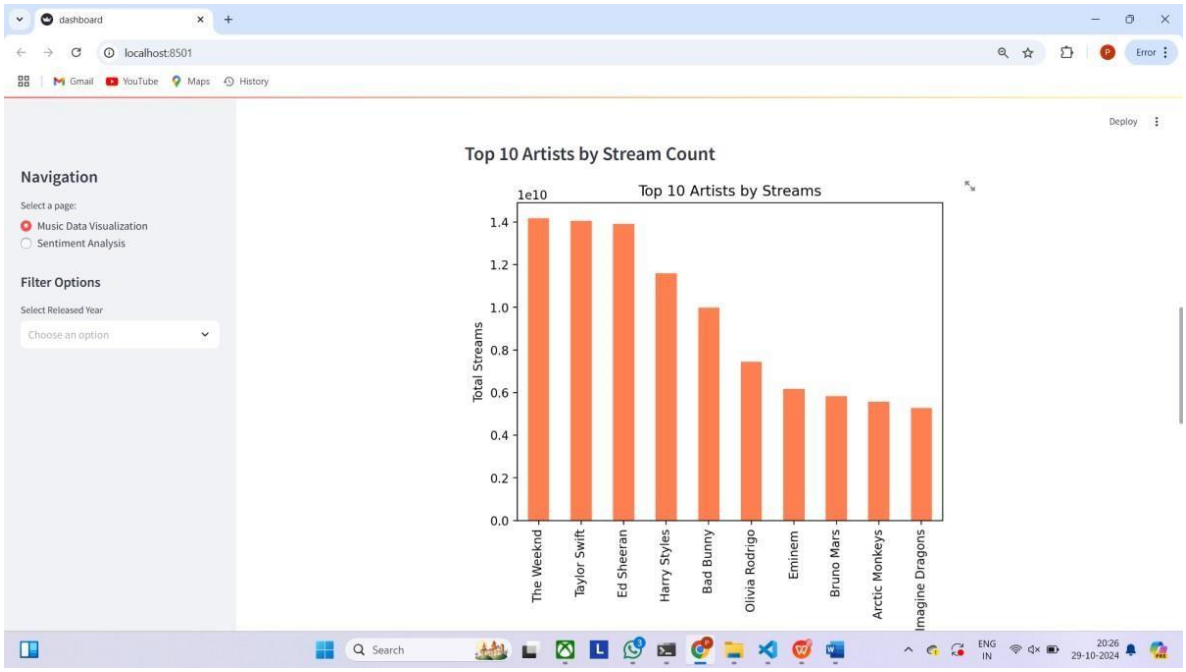

File Edit Selection View Go Run
ibm_project
data.csv
track_name,artist(s)_name,artist_count,released_year,released_month,released_day,in_spotify_playlists,in_spotify_charts,streams,in_apple_pl
Seven (feat. Latto) (Explicit Ver.),"Latto, Jung Kook",2,2023,7,14,553,147,141381703,43,263,45,10,826,125,B,Major,80,89,83,31,0,8,4,Not Fou
LALA,Myke Towers,1,2023,3,23,1474,48,133716286,48,126,58,14,382,92,C#,Major,71,61,74,7,0,10,4,https://i.scdn.co/image/ab67616d0000b2730656d
vampire,Olivia Rodrigo,1,2023,6,30,1397,113,140003974,94,207,91,14,949,138,F,Major,51,32,53,17,0,31,6,https://i.scdn.co/image/ab67616d0000b
Cruel Summer,Taylor Swift,1,2019,8,23,7858,100,800840817,116,207,125,12,548,170,A,Major,55,58,72,11,0,11,15,https://i.scdn.co/image/ab67616
WHERE SHE GOES,Bad Bunny,1,2023,5,18,3133,50,303236322,84,133,87,15,425,144,A,Minor,65,23,80,14,63,11,6,https://i.scdn.co/image/ab67616d000
Sprinter,"Dave, Central Cee",2,2023,6,1,2186,91,183706234,67,213,88,17,946,141,C#,Major,92,66,58,19,0,8,24,https://i.scdn.co/image/ab67616d
Ella Baila Sola,"Eslabon Armado, Peso Pluma",2,2023,3,16,3090,50,725980112,34,222,43,13,418,148,F,Minor,67,83,76,48,0,8,3,https://i.scdn.co
Columbia,Quevedo,1,2023,7,7,714,43,58149378,25,89,30,13,194,100,F,Major,67,26,71,37,0,11,4,https://i.scdn.co/image/ab67616d0000b273a00a817b
fukumean,Gunna,1,2023,5,15,1096,83,95217315,60,210,48,11,953,130,C#,Minor,85,22,62,12,0,28,9,https://i.scdn.co/image/ab67616d0000b273017d5e
La Bebe - Remix,"Peso Pluma, Yng Lvcas",2,2023,3,17,2953,44,553634067,49,110,66,13,339,170,D,Minor,81,56,48,21,0,8,33,Not Found
un x100to,"Bad Bunny, Grupo Frontera",2,2023,4,17,2876,40,505671438,41,205,54,12,251,83,F#,Minor,57,56,72,23,0,27,5,Not Found
Super Shy,NewJeans,1,2023,7,7,422,55,58255150,37,202,21,5,168,150,F,Minor,78,52,82,18,0,15,7,https://i.scdn.co/image/ab67616d0000b2733d98a0
Flowers,Miley Cyrus,1,2023,1,12,12211,115,1316855716,300,215,745,58,"1,021",118,,Major,71,65,68,6,0,3,7,https://i.scdn.co/image/ab67616d000
Daylight,David Kushner,1,2023,4,14,3528,98,387570742,80,156,182,24,"1,281",130,D,Minor,51,32,43,83,0,9,3,https://i.scdn.co/image/ab67616d00
As It Was,Harry Styles,1,2022,3,31,23575,130,2513188493,403,198,863,46,,174,F#,Minor,52,66,73,34,0,31,6,https://i.scdn.co/image/ab67616d000
Kill Bill,SZA,1,2022,12,8,8109,77,1163093654,183,162,161,12,187,89,G#,Major,64,43,73,5,17,16,4,https://i.scdn.co/image/ab67616d0000b2730c47
Cupid - Twin Ver.,Fifty Fifty,1,2023,2,24,2942,77,496795686,91,212,78,6,0,120,B,Minor,78,76,59,43,0,34,3,https://i.scdn.co/image/ab67616d00
"What Was I Made For? [From The Motion Picture ""Barbie""]",Billie Eilish,1,2023,7,13,873,104,30546883,80,227,95,24,"1,173",78,,Major,44,14
Classy 101,"Feid, Young Miko",2,2023,3,31,2610,40,335222234,43,100,54,14,187,100,B,Major,86,67,66,14,0,12,16,https://i.scdn.co/image/ab6761
Like Crazy,Jimin,1,2023,3,24,596,68,363369738,8,104,23,2,29,120,G,Major,63,36,73,0,0,36,4,https://i.scdn.co/image/ab67616d0000b2732b4607824
LADY GAGA,"Gabito Ballesteros, Junior H, Peso Pluma",3,2023,6,22,332,26,86444842,11,163,10,4,0,140,F,Minor,65,87,74,22,0,42,4,Not Found
I Can See You (Taylor¿¿¿¿s Version) (From The ,Taylor Swift,1,2023,7,7,516,38,52135248,73,119,42,1,150,123,F#,Major,69,82,76,6,0,6,3,N
I Wanna Be Yours,Arctic Monkeys,1,2013,1,1,12859,110,1297026226,24,98,582,2,73,135,,Minor,48,44,42,12,2,11,3,https://i.scdn.co/image/ab6761
"Peso Pluma: Bzrp Music Sessions, Vol. 55","Bizarrap, Peso Pluma",2,2023,5,31,1313,40,200647221,17,152,32,11,139,133,F,Minor,85,81,67,26,0,
Popular (with Playboi Carti & Madonna) - The Idol Vol. 1 (Music from the HBO Original Series),"The Weeknd, Madonna, Playboi Carti",3,2023,6
SABOR FRESA,Fuerza Regida,1,2023,6,22,250,26,78300654,16,149,10,5,168,130,G,Minor,79,96,86,9,0,9,9,https://i.scdn.co/image/ab67616d0000b273
Calm Down (with Selena Gomez),"Rï¿¼ï¿¼ma, Selena G",2,2022,3,25,7112,77,899183384,202,119,318,38,96,107,B,Major,80,82,80,43,0,14,4,Not Foun
MOJABI GHOST,"Tainy, Bad Bunny",2,2023,6,29,859,40,61245289,35,109,41,14,211,122,F#,Minor,81,74,71,14,0,56,4,https://i.scdn.co/image/ab6761
Last Night,Morgan Wallen,1,2023,1,31,2420,19,429829812,52,107,15,1,325,204,F#,Major,52,52,68,46,0,15,4,https://i.scdn.co/image/ab67616d000
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Plain Text Go Live Prettier
Search
ENG IN
20:21 29-10-2024

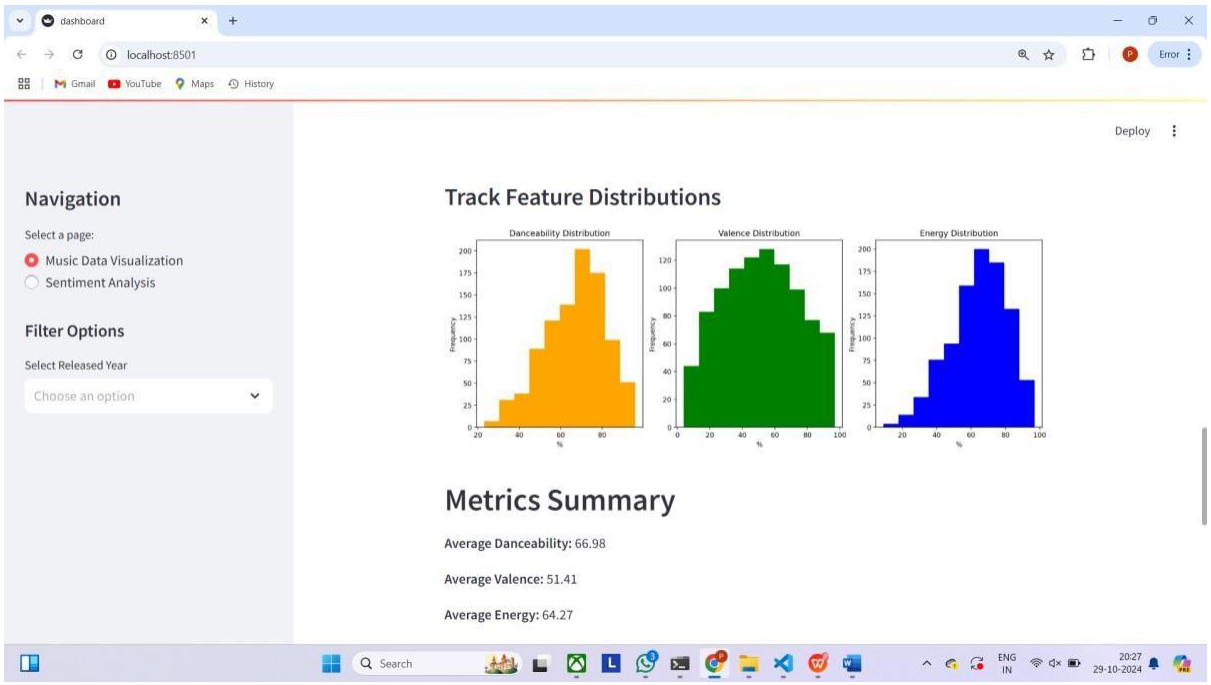
Output:

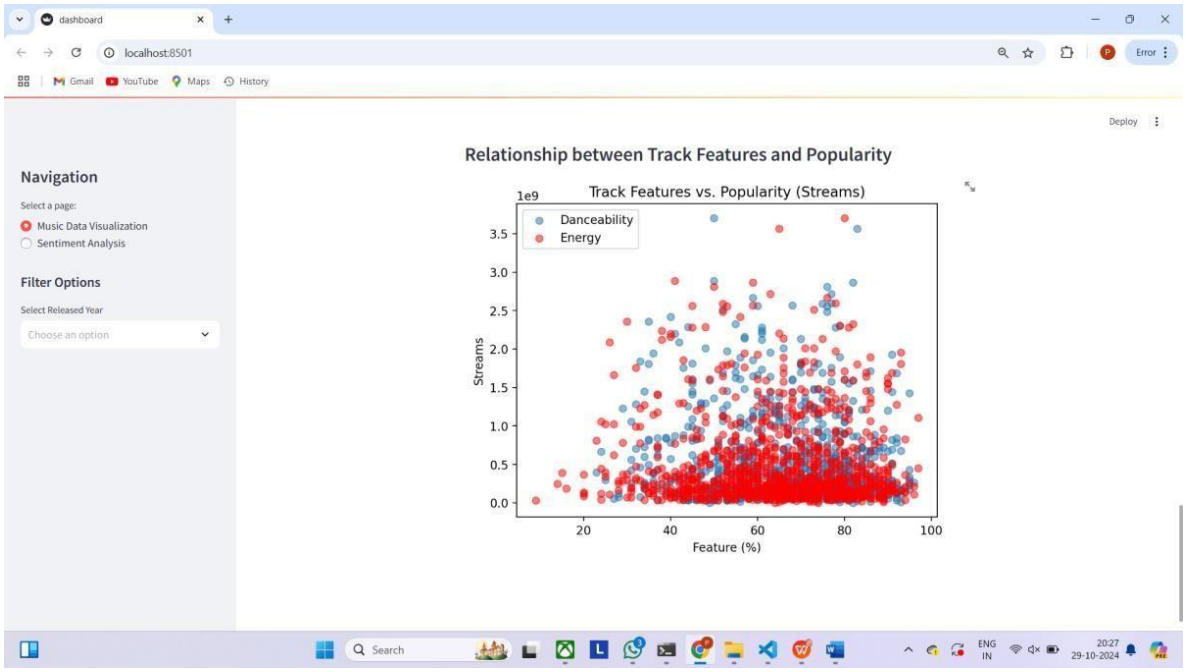

dashboard
localhost:8501
Gmail
YouTube
Maps
History
Error
Deploy
Navigation
Select a page:
Music Data Visualization
Sentiment Analysis
Sentiment Analysis Application
Enter a comment to analyze its sentiment
Enter your comment here:
your so good
Analyze Sentiment
Sentiment: Positive
Confidence Score: 0.96
Search
ENG
IN
20:24
29-10-2024

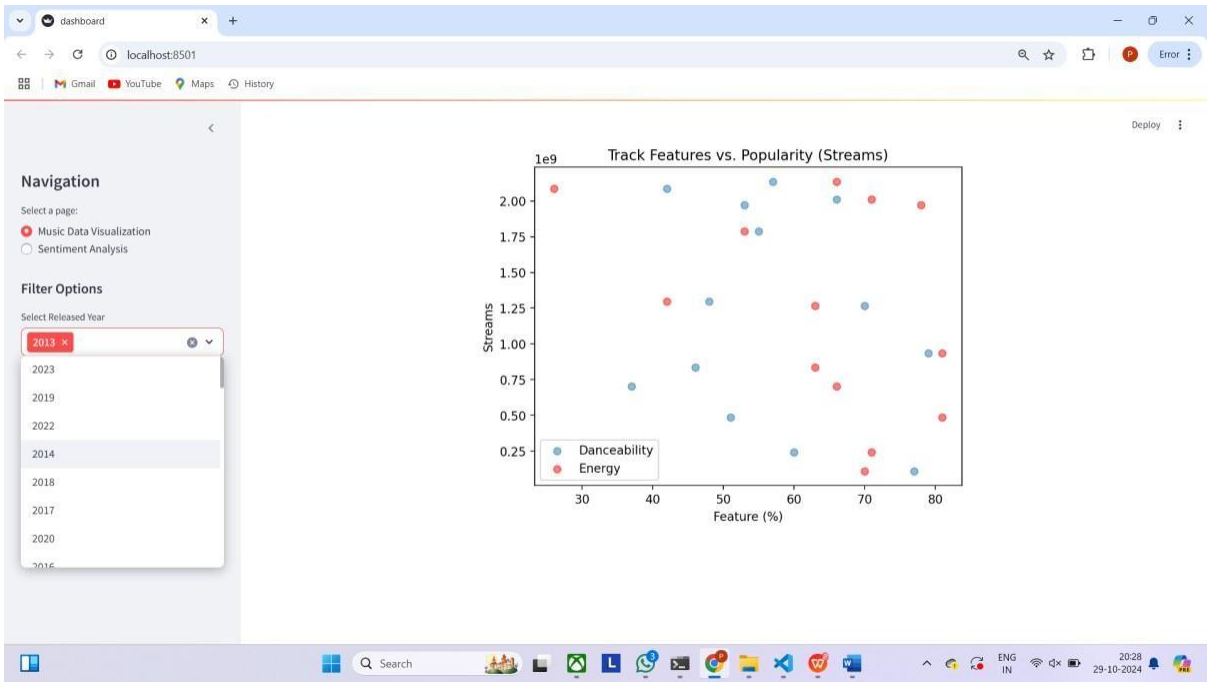

dashboard
localhost:8501
Gmail
YouTube
Maps
History
Error
Deploy
Navigation
Select a page:
Music Data Visualization
Sentiment Analysis
Filter Options
Select Released Year
Choose an option
Music Data Visualization Dashboard
Choose a CSV file
Drag and drop file here
Limit 200MB per file • CSV
Browse files
data.csv 152.9KB
Data Overview
track_name
artist(s)_name
artist_count
released_year
released_month
0 Seven (feat. Latto) (Explicit Ver.) Latto, Jung Kook 2 2,023 7
1 LALA Myke Towers 1 2,023 3
2 vampire Olivia Rodrigo 1 2,023 6
3 Cruel Summer Taylor Swift 1 2,019 8
4 WHERE SHE GOES Bad Bunny 1 2,023 5
Search
ENG
IN
20:25
29-10-2024


dashboard
localhost:8501
Gmail
YouTube
Maps
History
Error
Deploy
Navigation
Select a page:
Music Data Visualization
Sentiment Analysis
Filter Options
Select Released Year
Choose an option
Visualizations
Streams by Release Year
Total Streams by Release Year
1e11
Total Streams
Year
Search
ENG
IN
20:26
29-10-2024


dashboard
localhost:8501
Gmail
YouTube
Maps
History
Error
Deploy
Navigation
Select a page:
Music Data Visualization
Sentiment Analysis
Filter Options
Select Released Year
Choose an option
Top 10 Artists by Stream Count
Top 10 Artists by Streams
1e10
Total Streams
The Weeknd
Taylor Swift
Ed Sheeran
Harry Styles
Bad Bunny
Olivia Rodrigo
Eminem
Bruno Mars
Arctic Monkeys
Imagine Dragons
Search
ENG
IN
20:26
29-10-2024


dashboard
localhost:8501
Gmail
YouTube
Maps
History
Deploy
Navigation
Select a page:
Music Data Visualization
Sentiment Analysis
Filter Options
Select Released Year
Choose an option
Track Feature Distributions
Danceability Distribution
Valence Distribution
Energy Distribution
Frequency
%
Metrics Summary
Average Danceability: 66.98
Average Valence: 51.41
Average Energy: 64.27
Search
ENG
IN
20:27
29-10-2024


dashboard
localhost:8501
Gmail
YouTube
Maps
History
Deploy
Navigation
Select a page:
Music Data Visualization
Sentiment Analysis
Filter Options
Select Released Year
Choose an option
Relationship between Track Features and Popularity
Track Features vs. Popularity (Streams)
1e9
Danceability
Energy
Streams
Feature (%)
Search
ENG
IN
20:27
29-10-2024



Conclusion:

The conclusion of the social media analytics project on Instagram highlights the successful implementation of trend and sentiment analysis to capture the evolving music trends and audience sentiments from January to October 2024. By identifying trending songs and gauging public sentiment, the project provides insights into user engagement and preferences, offering value for influencers, marketers, and brands targeting music-basedcontent on Instagram. The dashboard serves as a powerfultool to visualize trends and sentiment shifts, helping stakeholders make data-driven decisions to enhance content relevance and audience engagement. This project underscores the potential of social media analytics in forecasting and adapting to user interests.

Reference:

1. Neri, F., Aliprandi, C., & Cuadros, M. (2012). Sentiment analysis on social media. Retrieved from https://www.researchgate.net/publication/230758119_Sentiment_Analysis_on_Social_Media
2. Zulfadzli, & Khalid, H. (2019). Sentiment analysis in social media. *Procedia Computer Science, 161*, 707-714. Retrieved from https://www.sciencedirect.com/science/article/pii/S187705091931885X
3. Rupavate, S. M., Bhagat, S. B., Dhameliya, P. J., Darji, H. K., & Chhaya, V. M. (2021). Sentiment analysis of social media data for emotion detection. *Journal of Pharmaceutical Research International, 33*(47A), 220-228. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8603338/#:~:text=The%20classification%20of%20the%20block,in%20the%20market%20or%20not.
4. Brand24. (2021). Social media sentiment analysis: Definition, tools, and examples. Retrieved from https://brand24.com/blog/social-media-sentiment-analysis/#:~:text=Social%20media%20sentiment%20analysis%20is%20a%20process%20of%20using%20natural,positive%2C%20neutral%2C%20or%20negative.
5. Comparative study of Sentiment Analysis on trending issues on Social Media (feb 2018) byhttps://www.researchgate.net/publication/324602957_Comparative_study_of_Sentiment_Analysis_on_trending_issues_on_Social_Media

6. Sentiment Analysis for Social Media (November 2013) by R. A. S. C. Jayasanka, M. D. T. Madushani, E. R. Marcus, I. A. A. U. Abeyratnhttps://www.researchgate.net/publication/268817500_Sentiment_Analysis_for_Social_Media