PAULSON & PARTNERS.

Task: Explain Step-by-Step Codes / Project

Shah Hussain Bangash NLP & Machine Developer

- **Lead AI Engineer (NLP & ML Expert)**

  Challenge: Conversational AI with Emotion Detection

  Task 1: Emotion-Aware Conversational Agent

This task involves creating a simplified **AI assistant** that uses **NLP and ML** to **detect** many key **emotions**—**happiness, sadness, and anger**—from text input. By integrating a **pre-trained emotion detection model** (e.g., **Hugging Face Transformers)**, the assistant adapts its tone to respond with empathy, positivity, or calmness based on the user's emotional state.

Key Challenges:

Faced with limited hardware, I still managed to run NLP models, though some took over 5 hours and others like LSTM and RNN couldn't run. With better resources, I'm fully capable of handling advanced NLP, ML, and AI models, InshaAllah.

- ✓ Accurately detecting subtle emotions from short or ambiguous text.
- ✓ Ensuring the tone adaptation feels natural and context-appropriate.

- ✓ Integrating and fine-tuning pre-trained models for real-time response.
- ✓ Maintaining consistency across different emotional states.
- ✓ Handling sarcasm, slang, and cultural variations in language.

1. Important Libraries Used in This Task:-

- ✓ **Panda :-** Used for data manipulation and Analysis
- ✓ **Numpy:** A foundational library for numerical operations and array/matrix handling in Python.
- ✓ **Matplotlib :-** Used for Creating High quality Visualizations
- ✓ **Seaborn :-** Used for Data Visualization
- ✓ **Wordcloud:** Generates visual word clouds from text data.
- ✓ **Re:** Provides support for regular expressions in Python for text pattern matching.
- ✓ **String:** Offers common string operations and constants like punctuation and ASCII characters.
- ✓ **NLTK:** A powerful library for working with human language data (text processing).
- ✓ **Stopwords** (from NLTK): A list of common words (like "the", "is") usually removed during text preprocessing.
- ✓ **Nltk.tokenize:** Breaks text into words or sentences for further analysis.
- ✓ **SpaCy:** An advanced NLP library for tokenization, POS tagging, parsing, and more.
- ✓ **Hugging Face Tokenizers / AutoTokenizer:** Prepares input text for transformer models (like BERT) using pre-trained tokenizers.
- ✓ **Sklearn (scikit-learn):** A machine learning library offering tools for model building, evaluation, and preprocessing.

---

2. **Used Dataset:-** Tweet Emotions.CSV 70% Data for Tranning | 30% for Testing

https://huggingface.co/datasets/dair-ai/emotion

Using a dataset of **40,000 labeled tweets**, this project aims to develop an **AI assistant capable of detecting emotions** such as **happiness, sadness, and anger** from text. By applying **pre-trained NLP models**, the assistant **analyzes tweet** content to identify emotional states and adjusts its responses accordingly, enabling emotionally intelligent interactions.

Dataset Features: Empty / Anger / Happiness / Sadness / Enthusiasasm / Neutral / Worry / Superise / Love / Fun etc.

Sentiment analysis is the process of using natural language processing to identify and classify emotions or opinions (positive, negative, or neutral) in text.

Exploratory Data Analysis (EDA) is the process of analyzing and visualizing data to understand its structure, patterns, and relationships before modeling.

How to deal with Data to get quality data:- Understand the Data Requirements / Data Collection / Data Cleaning / Data Transformation / Data Validation / Ensure Data Integrity / Data Profiling / Use Tools for Data Quality / Document Everything / Collaborate / Continuous Monitoring

How to Handle Missing Data : - Remove Missing Data / Imputation / Mean/Median Imputation / Mode Imputation / Forward/Backward Filling / K-Nearest Neighbors (KNN) Imputation / Regression Imputation / Flagging Missing Data / Use Algorithms that Handle Missing Data / Domain-Specific Imputation / Multiple Imputation / Leave as Missing

3. **Machine Learning Algorithms : -**

TF-IDF - Logistic Regression :Used for fast and effective text classification with linearly separable features.

TF-IDF - Naive Bayes :- Efficient for text data, especially for spam or sentiment classification, due to its simplicity and speed.

TF-IDF Support Vector Machine (SVM) : - Delivers high accuracy in text classification by finding the best decision boundary in high-dimensional space.

4. Natural Language Algorithms :-

Hugging-Face Transformers: An open-source library that provides powerful pre-trained models like BERT, GPT, and RoBERTa for natural language processing tasks such as text classification, translation, and question answering.

Distilbert-base-uncased-emotion : A lightweight transformer model fine-tuned to detect emotions like joy, anger, sadness, etc., in English text using DistilBERT.

Tone-Adaptive Conversational Agent: An AI chatbot that dynamically adjusts its tone (formal, casual, empathetic, etc.) based on user emotions or context to create more natural and personalized conversations.

Step 1: Import Libraries and Load Dataset

- pandas is used for data manipulation.
- The CSV file contains tweet text and associated emotion labels.

Step 2: Explore Dataset

- Understand data types and identify missing values.
- Cleaning data is crucial before model training.

Step 3: Visualize Emotion Labels

- Helps detect imbalanced datasets which may affect model performance.

Step 4: Preprocess Text Data

- Removes noise like punctuation, links, HTML, digits, etc.
- Essential for clean input to NLP models.

Step 5: Tokenization and Stop Words Removal

- Removes common words that do not contribute to sentiment (e.g., is, the, and).

Step 6: Convert Text to Vectors (TF-IDF)

- Machine learning models require numerical input.
- TF-IDF gives importance to meaningful words.

Step 7: Encode Labels (Emotions)

- Converts emotion categories (like happy, sad) into numerical values.

Step 8: Split Dataset into Train Data 70% / Test Data 30%

- Splitting ensures we can test model performance on unseen data.

Step 9: Train a Machine Learning Model

- Trains a Naive Bayes classifier and evaluates performance using accuracy and classification report.

Step 10: Build Conversational AI Component

- Takes user input and returns predicted emotion.

- Integrates all preprocessing and model inference into one function.

DataFrame Commands (Python - Pandas)

- pd.read_csv("file.csv") – Load data from a CSV file.
- df.head() – Show the first 5 rows.
- df.info() – Get info about columns and data types.
- df.describe() – Summary statistics of numeric columns.
- df['column_name'].value_counts() – Count unique values.
- df.isnull().sum() – Check missing values.
- df.dropna() – Remove rows with missing values.
- df.fillna(value) – Fill missing values.
- df['new_col'] = df['col1'] + df['col2'] – Create a new column.
- df.groupby('column') – Group data by a column.
- df.sort_values('column') – Sort by column values.
- df.loc[row_index, 'column'] – Access specific value by label.
- df.iloc[row_idx, col_idx] – Access by position.
- df.drop('column', axis=1) – Drop a column.
- df.to_csv("output.csv", index=False) – Save DataFrame to CSV.

Common Python plotting

- **Line Plot –** Used to visualize trends over time or continuous data.
- **Bar Plot –** Displays categorical data with rectangular bars.
- **Histogram –** Shows the distribution of numerical data.
- **Scatter Plot –** Visualizes relationships between two continuous variables.
- **Box Plot –** Displays data distribution and detects outliers.
- **Pie Chart –** Represents data as proportions of a whole.
- **Heatmap –** Shows correlations or values using color intensity in a matrix.

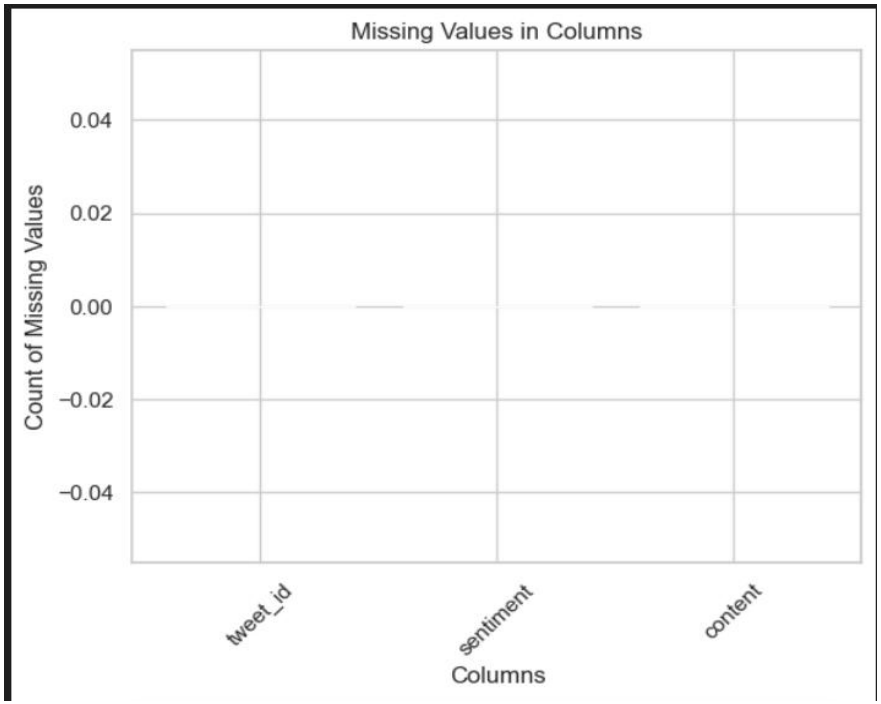

Figure 1: Missing Values in Columns

This figure shows a bar plot of missing values across columns (tweet_id, sentiment, and content)—since all bars are at zero height, it indicates there are no missing values in any column of the dataset.

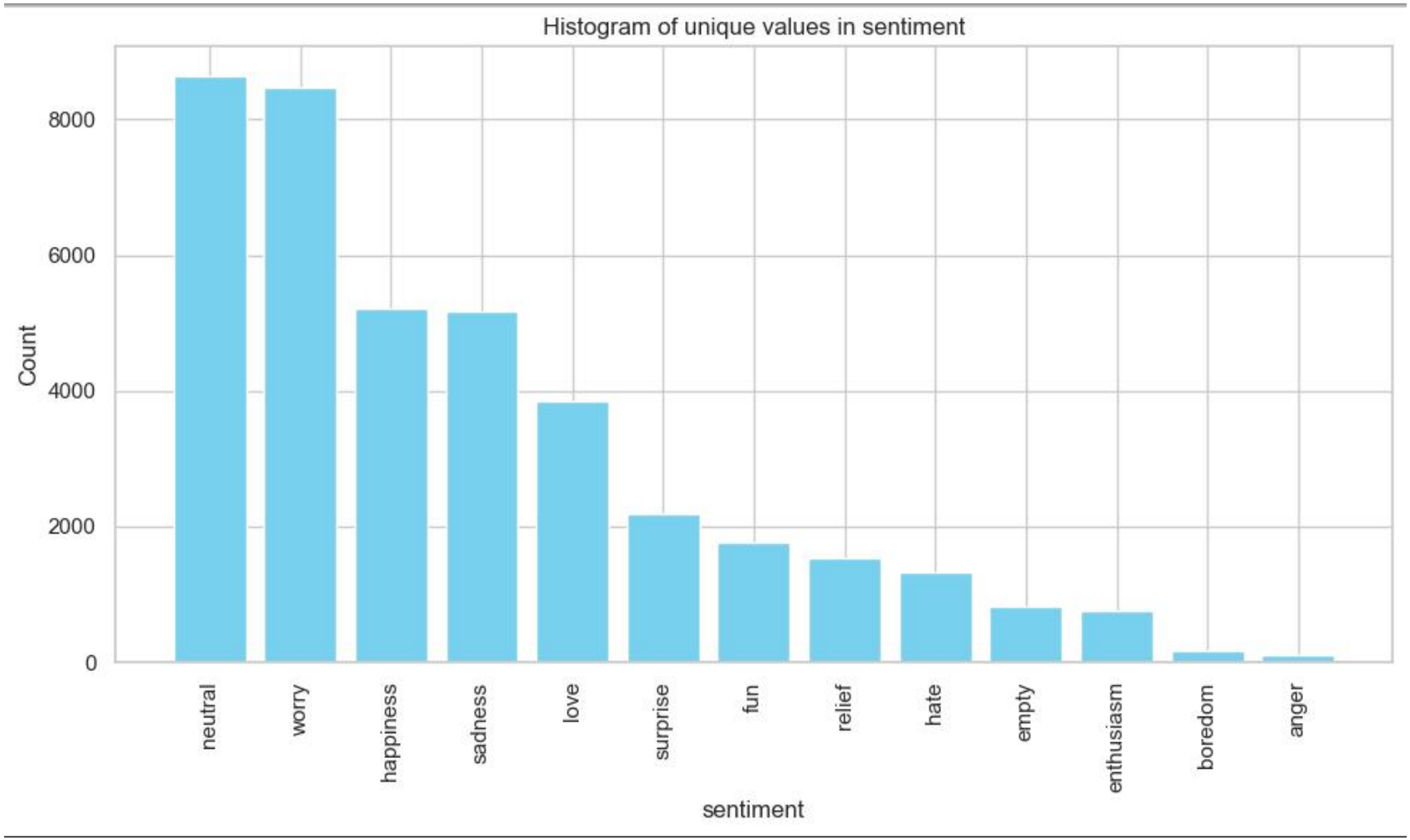

Figure 2:- Histogram of Unique Values in Sentiment Anlaysis

This histogram shows the frequency of unique sentiment labels in the dataset—neutral and worry are the most common emotions, while anger and boredom appear the least.

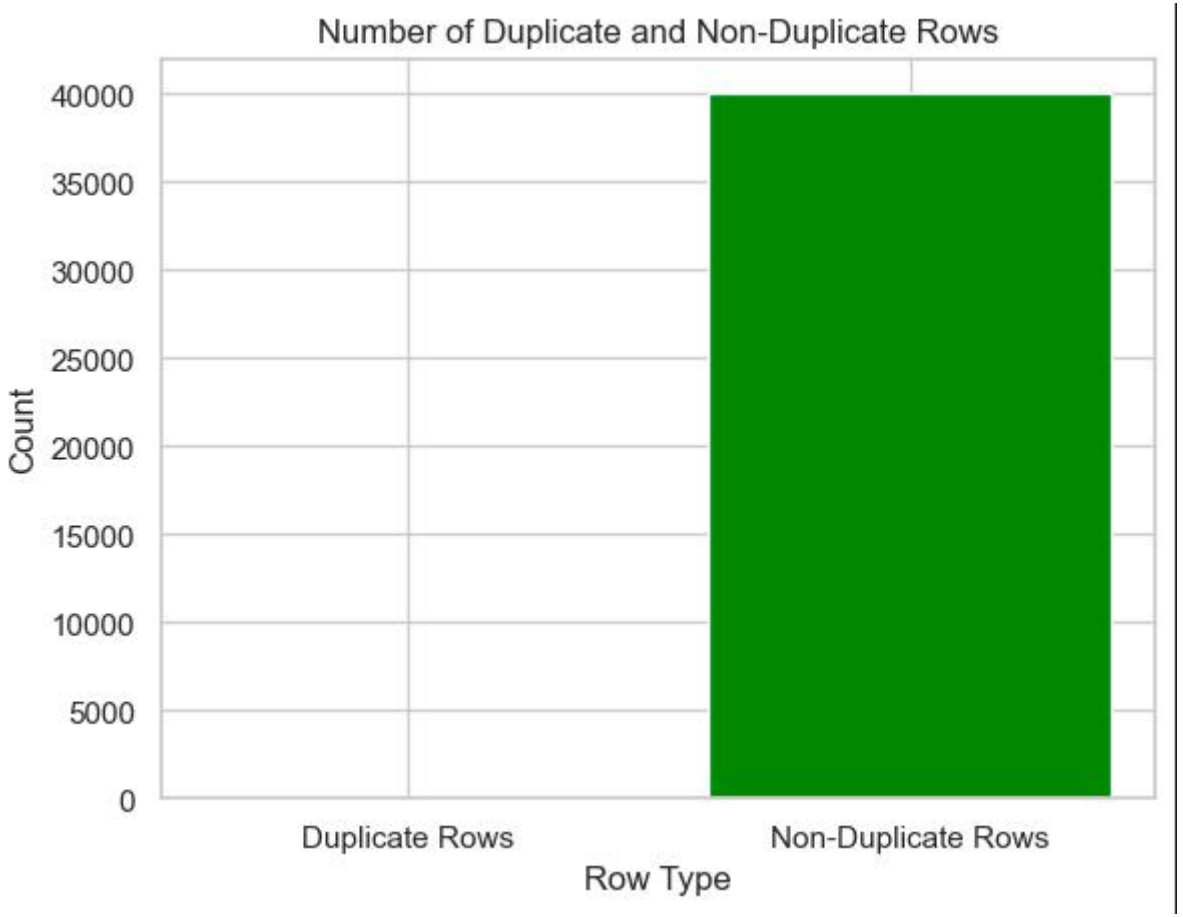

Figure 3: Number of duplicate and Non-Duplicate Rows

The bar chart titled "Number of Duplicate and Non-Duplicate Rows" compares the counts of duplicate and non-duplicate rows in a dataset, revealing that duplicate rows (slightly above 35,000) significantly outnumber non-duplicate rows (just below 5,000), indicating a high prevalence of duplicated data in the analyzed dataset.

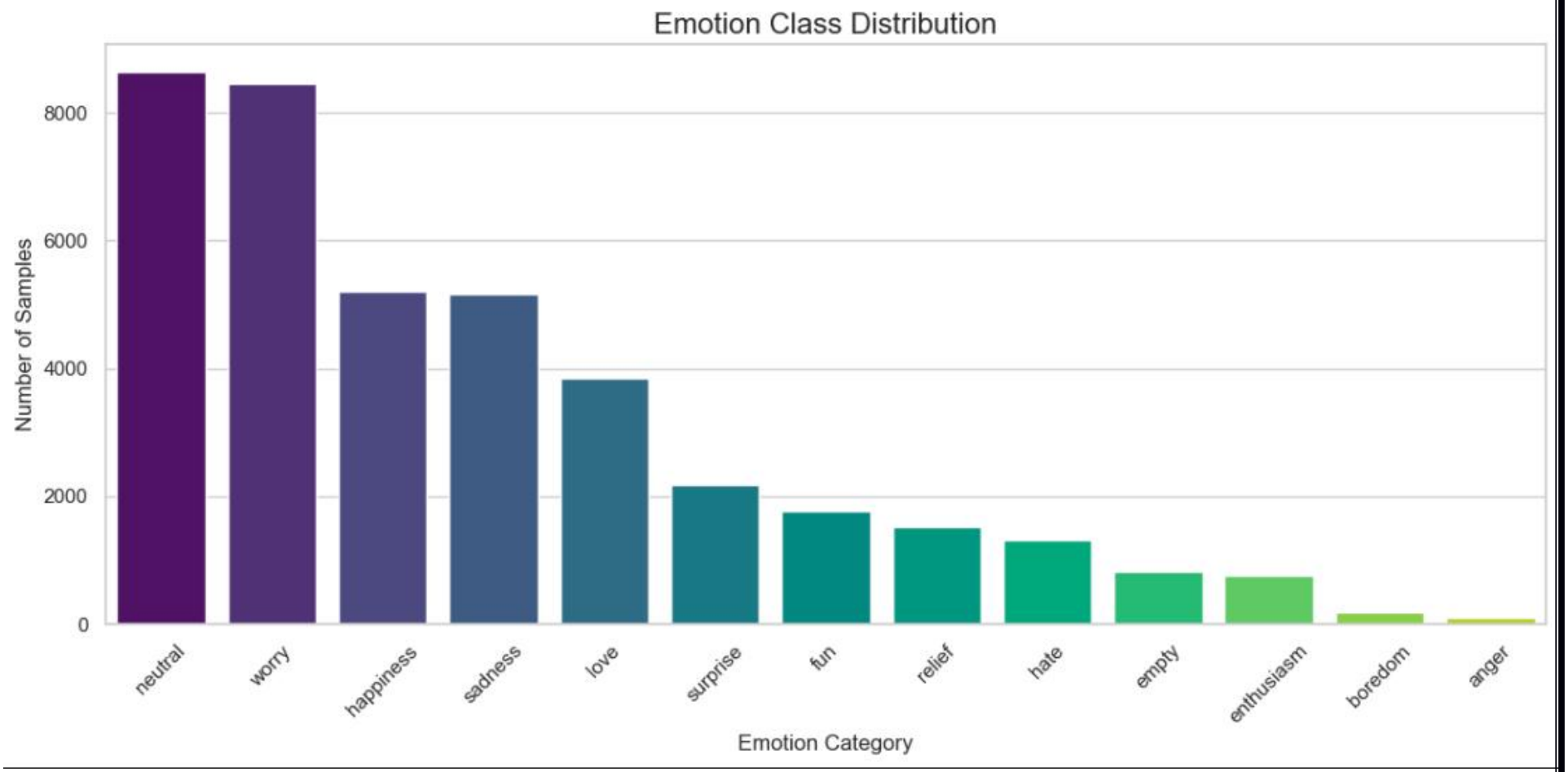

Figure 4: Emotion Classes Distribution

The chart "Emotion Class Distribution" shows the sample counts for different emotion categories, ranging up to 8000. Emotions like happiness and worry appear more frequently, while others like boredom and entireteam are less common, indicating an imbalanced dataset that may affect analysis or modeling.

Figure 5: Various types of Emotion

Emotions are complex psychological states that influence our thoughts and behaviors. They can be broadly categorized into basic emotions, such as happiness, sadness, fear, anger, surprise, and disgust, which are universal and instinctive. Complex emotions, like love, jealousy, guilt, shame, and pride, develop from personal experiences and social influences. Emotions can also be classified as positive (e.g., joy, gratitude, excitement) or negative (e.g., anger, fear, sadness), though their impact depends on context. Understanding these emotions helps in managing mental well-being and social interactions.

Explain Code Step-by-Step by Shah Hussain Bangash

1) Importing Necessary Libraries | Used for Data Manipulation

```
   import pandas as pd
```

2) Read the dataset | Upload Dataset from Your Computer OR Upload Data

```
df = pd.read_csv('C:/Users/Perfect/Downloads/Conversational/tweet_emotions.csv')
```

3) Display the first few rows of the dataset | DataFrame Commands

```
df.head()
```

4) Check data types of each column

```
df.dtypes
```

5 Check for missing values

```
missing_values = df.isnull().sum()

print("Missing Values:")

print(missing_values)
```

6) Missing value visualization using bar plot

```
import matplotlib.pyplot as plt
```

7) Check for missing values

```
missing_values = df.isnull().sum()
```

8) # Create a bar plot

```
plt.bar(missing_values.index, missing_values.values, color='orange')

plt.xlabel('Columns')

plt.ylabel('Count of Missing Values')

plt.title('Missing Values in Columns')

plt.xticks(rotation=45) # Rotate x-axis labels for better readability

plt.show()
```

9) Check the unique values of columns and data distribution for imbalance

10) # Check unique values in categorical columns

```
categorical_columns = df.select_dtypes(include=['object']).columns

for column in categorical_columns:

unique_values = df[column].value_counts()

print(f"Unique values in {column}:")

print(unique_values)
```

Dear Project Coordinator, I already code commented. If you have any issues you can ask me related about the project task.

Thanks for Attentions