```
import pandas as np
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

DATASET UPLOAD

```
# Step 1: Import necessary libraries
from google.colab import files
import pandas as pd

# Step 2: Upload the CSV file
uploaded = files.upload()
```

Choose Files sentimentdataset (2).csv

- **sentimentdataset (2).csv**(text/csv) - 170776 bytes, last modified: 5/16/2025 - 100% done

```
Saving sentimentdataset (2).csv to sentimentdataset (2).csv
```

DATA EXPLORATION

```
# Read the uploaded CSV file into a pandas DataFrame
# Replace 'your_file.csv' with the actual name of the uploaded file
df = pd.read_csv(next(iter(uploaded)))

# Display the DataFrame
print(df.head())
```

```
   Unnamed: 0.1  Unnamed: 0  \
0             0           0
1             1           1
2             2           2
3             3           3
4             4           4

                                            Text    Sentiment  \
0   Enjoying a beautiful day at the park!        ...   Positive
1   Traffic was terrible this morning.           ...   Negative
2   Just finished an amazing workout! 💪          ...   Positive
3   Excited about the upcoming weekend getaway!  ...   Positive
4   Trying out a new recipe for dinner tonight.  ...   Neutral

              Timestamp            User     Platform  \
0  2023-01-15 12:30:00   User123          Twitter
1  2023-01-15 08:45:00   CommuterX        Twitter
2  2023-01-15 15:45:00   FitnessFan      Instagram
3  2023-01-15 18:20:00   AdventureX       Facebook
4  2023-01-15 19:55:00   ChefCook        Instagram

                                   Hashtags  Retweets  Likes       Country  \
0   #Nature #Park                                15.0   30.0     USA
1   #Traffic #Morning                             5.0   10.0     Canada
2   #Fitness #Workout                            20.0   40.0   USA
3   #Travel #Adventure                            8.0   15.0     UK
4   #Cooking #Food                               12.0   25.0    Australia

   Year  Month  Day  Hour
0  2023      1   15    12
1  2023      1   15     8
2  2023      1   15    15
3  2023      1   15    18
4  2023      1   15    19
```

DESCRIBE

```
df.describe()
```

	Unnamed: 0.1	Unnamed: 0	Retweets	Likes	Year	Month	Day	Hour
count	732.000000	732.000000	732.000000	732.000000	732.000000	732.000000	732.000000	732.000000
mean	366.464481	369.740437	21.508197	42.901639	2020.471311	6.122951	15.497268	15.521858
std	211.513936	212.428936	7.061286	14.089848	2.802285	3.411763	8.474553	4.113414
min	0.000000	0.000000	5.000000	10.000000	2010.000000	1.000000	1.000000	0.000000
25%	183.750000	185.750000	17.750000	34.750000	2019.000000	3.000000	9.000000	13.000000
50%	366.500000	370.500000	22.000000	43.000000	2021.000000	6.000000	15.000000	16.000000
75%	549.250000	553.250000	25.000000	50.000000	2023.000000	9.000000	22.000000	19.000000
max	732.000000	736.000000	40.000000	80.000000	2023.000000	12.000000	31.000000	23.000000

CHECK NULL

```
df.isnull().sum()
```

	0
Unnamed: 0.1	0
Unnamed: 0	0
Text	0
Sentiment	0
Timestamp	0
User	0
Platform	0
Hashtags	0
Retweets	0
Likes	0
Country	0
Year	0
Month	0
Day	0
Hour	0

DROP

```
df.dropna()                    # Drop rows with any NaN
df.dropna(axis=1)             # Drop columns with any NaN
```

	Unnamed: 0.1	Unnamed: 0	Text	Sentiment	Timestamp	User	Platform	Hashtags	Retweet
0	0	0	Enjoying a beautiful day at the park! ...	Positive	2023-01-15 12:30:00	User123	Twitter	#Nature #Park	15
1	1	1	Traffic was terrible this morning. ...	Negative	2023-01-15 08:45:00	CommuterX	Twitter	#Traffic #Morning	5
2	2	2	Just finished an amazing workout! 💪 ...	Positive	2023-01-15 15:45:00	FitnessFan	Instagram	#Fitness #Workout	20
3	3	3	Excited about the upcoming weekend getaway! ...	Positive	2023-01-15 18:20:00	AdventureX	Facebook	#Travel #Adventure	8
4	4	4	Trying out a new recipe for dinner tonight. ...	Neutral	2023-01-15 19:55:00	ChefCook	Instagram	#Cooking #Food	12
...	...	...	...	...	...	...	...	...	
727	728	732	Collaborating on a science project that receiv...	Happy	2017-08-18 18:20:00	ScienceProjectSuccessHighSchool	Facebook	#ScienceFairWinner #HighSchoolScience	20
728	729	733	Attending a surprise birthday party organized ...	Happy	2018-06-22 14:15:00	BirthdayPartyJoyHighSchool	Instagram	#SurpriseCelebration #HighSchoolFriendship	25
729	730	734	Successfully fundraising for a school charity ...	Happy	2019-04-05 17:30:00	CharityFundraisingTriumphHighSchool	Twitter	#CommunityGiving #HighSchoolPhilanthropy	22
730	731	735	Participating in a multicultural festival, cel...	Happy	2020-02-29 20:45:00	MulticulturalFestivalJoyHighSchool	Facebook	#CulturalCelebration #HighSchoolUnity	21
731	732	736	Organizing a virtual talent show during challe...	Happy	2020-11-15 15:15:00	VirtualTalentShowSuccessHighSchool	Instagram	#VirtualEntertainment #HighSchoolPositivity	24

732 rows × 15 columns

DUPLICATE

```
# Count duplicate rows
duplicate_rows = df.duplicated().sum()
print(f"Number of duplicate rows: {duplicate_rows}")
```

```
Number of duplicate rows: 0
```

ENCODING

```
import pandas as pd

# Try reading with UTF-8 encoding
df = pd.read_csv('sentimentdataset (2).csv', encoding='utf-8')


# Alternative common encoding
df = pd.read_csv('sentimentdataset (2).csv', encoding='ISO-8859-1')  # aka 'latin1'
```

ONE-HOT ENCODING

```
# One-hot encode all categorical columns automatically
df_encoded = pd.get_dummies(df)
```

```
df_encoded = pd.get_dummies(df, columns=['Sentiment'])


encoded = pd.get_dummies(df['Sentiment'])
df = pd.concat([df, encoded], axis=1)


print(df.head)
```

```
728  2018-06-22 14:15:00            BirthdayPartyJoyHighSchool    Instagram
729  2019-04-05 17:30:00   CharityFundraisingTriumphHighSchool      Twitter
730  2020-02-29 20:45:00    MulticulturalFestivalJoyHighSchool     Facebook
731  2020-11-15 15:15:00    VirtualTalentShowSuccessHighSchool    Instagram

                                        Hashtags  Retweets  Likes  ...  \
0        #Nature #Park                                15.0   30.0  ...
1        #Traffic #Morning                             5.0   10.0  ...
2        #Fitness #Workout                            20.0   40.0  ...
3        #Travel #Adventure                            8.0   15.0  ...
4        #Cooking #Food                               12.0   25.0  ...
..                                            ...       ...    ...  ...
727         #ScienceFairWinner #HighSchoolScience    20.0   39.0  ...
728    #SurpriseCelebration #HighSchoolFriendship    25.0   48.0  ...
729      #CommunityGiving #HighSchoolPhilanthropy    22.0   42.0  ...
730         #CulturalCelebration #HighSchoolUnity    21.0   43.0  ...
731   #VirtualEntertainment #HighSchoolPositivity    24.0   47.0  ...

     Vibrancy    Whimsy           Whispers of the Past    Winter Magic   \
0        False            False                   False           False
1        False            False                   False           False
2        False            False                   False           False
3        False            False                   False           False
4        False            False                   False           False
..         ...              ...                     ...             ...
727      False            False                   False           False
728      False            False                   False           False
729      False            False                   False           False
730      False            False                   False           False
731      False            False                   False           False

      Wonder    Wonder        Wonder          Wonderment       Yearning   \
0       False         False           False            False       False
1       False         False           False            False       False
2       False         False           False            False       False
3       False         False           False            False       False
4       False         False           False            False       False
..        ...           ...             ...              ...         ...
727     False         False           False            False       False
728     False         False           False            False       False
729     False         False           False            False       False
730     False         False           False            False       False
731     False         False           False            False       False

      Zest
0     False
1     False
2     False
3     False
4     False
..      ...
727   False
728   False
729   False
730   False
731   False

[732 rows x 294 columns]>
```

TRAIN TEST

```
from sklearn.model_selection import train_test_split

# X = features (everything except the target)
# y = target column (e.g., 'Sentiment')
X = df.drop('Sentiment', axis=1)
y = df['Sentiment']

# Split: 80% training, 20% testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print("Training set:", X_train.shape)
print("Testing set:", X_test.shape)
```

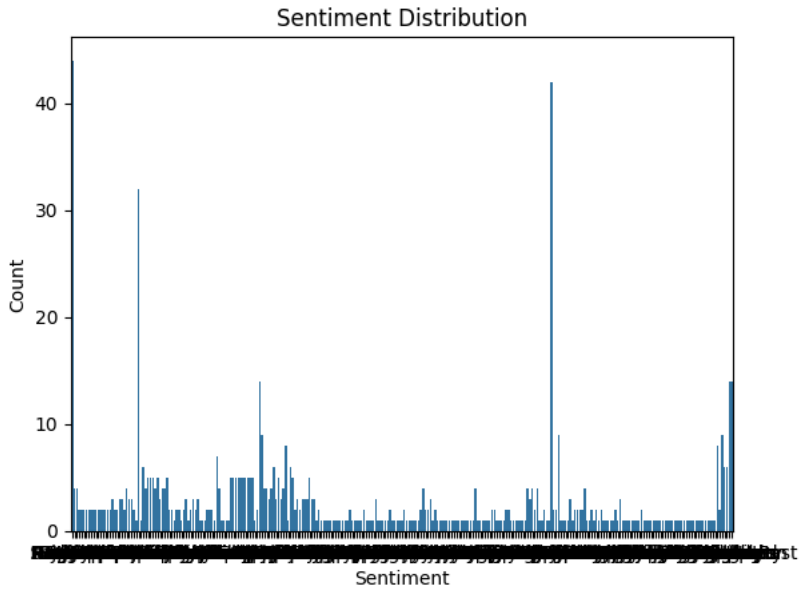
```
Training set: (585, 293)
Testing set: (147, 293)
```

VISUALIZATION

BAR PLOT

```
import matplotlib.pyplot as plt
import seaborn as sns
```

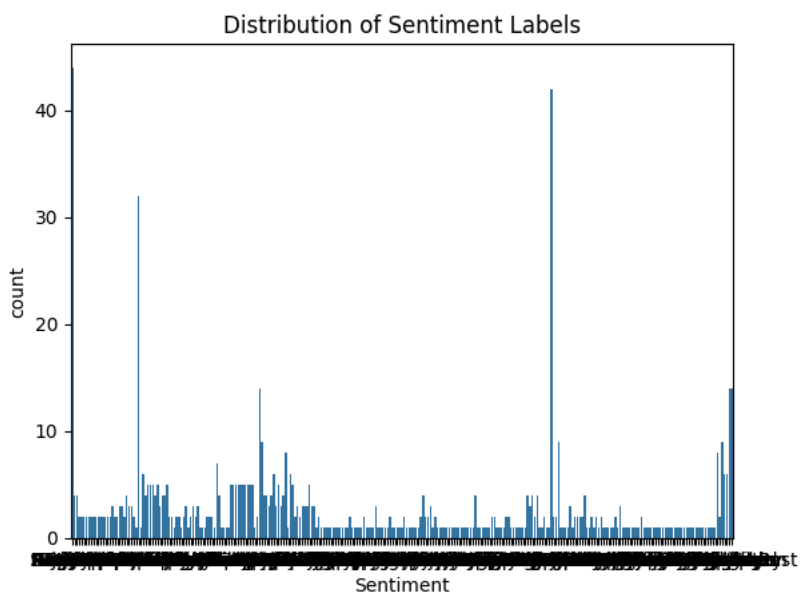
```
sns.countplot(x='Sentiment', data=df)
plt.title("Sentiment Distribution")
plt.xlabel("Sentiment")
plt.ylabel("Count")
plt.show()
```



COUNTPLOT

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.countplot(x='Sentiment', data=df)
plt.title('Distribution of Sentiment Labels')
plt.show()
```



EVALUATION

```
from sklearn.model_selection import train_test_split

# X = features (everything except the target and text/identifier columns)
```

```
# y = target column (e.g., 'Sentiment')

# Identify and drop columns that are non-numeric and not intended as features
# Replace 'Text_Column_Name' and 'ID_Column_Name' with the actual names of
# columns containing text or irrelevant identifiers in your DataFrame.
# Based on the earlier output, 'Unnamed: 0.1' and 'Unnamed: 0' might be such columns.
columns_to_drop = ['Sentiment'] # Sentiment is the target
# Add other columns you don't want as features, e.g., the original text column, any index columns
if 'Unnamed: 0.1' in df.columns:
    columns_to_drop.append('Unnamed: 0.1')
if 'Unnamed: 0' in df.columns:
    columns_to_drop.append('Unnamed: 0')
# Add the actual name of your text column if it exists, e.g., 'Tweet_Text'
# if 'Tweet_Text' in df.columns:
#     columns_to_drop.append('Tweet_Text')


X = df.drop(columns=columns_to_drop, axis=1)
y = df['Sentiment']

# Before splitting, ensure X contains only numerical data.
# You can inspect X.dtypes to see the data types of each column.
# If there are still 'object' columns (which typically contain strings),
# you need to handle them (e.g., drop, encode, or apply text processing).
# For demonstration, let's drop any remaining non-numeric columns in X.
X = X.select_dtypes(include=['number'])


# Split: 80% training, 20% testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print("Training set:", X_train.shape)
print("Testing set:", X_test.shape)
```

```
Training set: (585, 6)
Testing set: (147, 6)
```

DEPLOYMENT

```
# Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report


# Read CSV (replace with actual filename or use uploaded.keys())
df = pd.read_csv('sentimentdataset (2).csv', encoding='utf-8')  # Try ISO-8859-1 if utf-8 fails

# Basic preprocessing
df.dropna(inplace=True)  # Remove rows with missing values
df.drop_duplicates(inplace=True)  # Remove duplicates

# Label Encoding (if target is categorical text)
le = LabelEncoder()
df['Sentiment'] = le.fit_transform(df['Sentiment'])  # Adjust if target column has a different name

# Identify columns to drop, including the text column and potentially index columns
columns_to_drop = ['Sentiment'] # Sentiment is the target
# Add other columns you don't want as features, e.g., the original text column, any index columns
if 'Unnamed: 0.1' in df.columns:
    columns_to_drop.append('Unnamed: 0.1')
if 'Unnamed: 0' in df.columns:
    columns_to_drop.append('Unnamed: 0')
# **IMPORTANT**: Replace 'Your_Text_Column_Name' with the actual name of your text column
# You can inspect df.columns to find the correct name.
text_column_name = 'Your_Text_Column_Name' # <-- **Change this to your actual text column name**
if text_column_name in df.columns:
    columns_to_drop.append(text_column_name)


# Split into features and target, dropping the identified columns from features (X)
X = df.drop(columns=columns_to_drop, axis=1)  # Features
y = df['Sentiment']              # Target

# Ensure X contains only numerical data by selecting only numeric columns after dropping
X = X.select_dtypes(include=[np.number]) # Use np.number for a broader selection of numeric types
```

```
# Train/test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Check the data types in X_train before fitting
print("Data types in X_train:")
print(X_train.dtypes)


# Model training
# Check if X_train is empty after dropping non-numeric columns
if X_train.empty:
    print("X_train is empty after dropping non-numeric columns. Cannot train the model.")
else:
    model = LogisticRegression(max_iter=1000)
    model.fit(X_train, y_train)

    # Predictions
    y_pred = model.predict(X_test)

    # Evaluation
    print("Accuracy:", accuracy_score(y_test, y_pred))
    print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))
    print("\nClassification Report:\n", classification_report(y_test, y_pred))

# Visualization (optional)
sns.countplot(x='Sentiment', data=df)
plt.title("Sentiment Distribution")
plt.show()
```

```
Data types in X_train:
Retweets    float64
Likes       float64
Year          int64
Month         int64
Day           int64
Hour          int64
dtype: object
/usr/local/lib/python3.11/dist-packages/sklearn/linear_model/_logistic.py:465: ConvergenceWarning: lbfgs failed to converge (stat
STOP: TOTAL NO. OF ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Recall is ill-defined ar
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Recall is ill-defined ar
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Precision is ill-defined
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1565: UndefinedMetricWarning: Recall is ill-defined ar
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
Accuracy: 0.08843537414965986

Confusion Matrix:
 [[0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]

Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         2
           3       0.00      0.00      0.00         1
           5       0.00      0.00      0.00         1
           8       0.00      0.00      0.00         0
           9       0.00      0.00      0.00         1
          12       0.00      0.00      0.00         1
          15       0.00      0.00      0.00         1
          17       0.00      0.00      0.00         1
          22       0.00      0.00      0.00         3
          24       0.00      0.00      0.00         1
          26       0.00      0.00      0.00         1
          28       0.00      0.00      0.00         1
          29       0.00      0.00      0.00         2
          30       0.00      0.00      0.00         1
          31       0.00      0.00      0.00         1
          32       0.00      0.00      0.00         1
          33       0.00      0.00      0.00         1
          35       0.00      0.00      0.00         1
          36       0.00      0.00      0.00         0
          38       0.00      0.00      0.00         1
          40       0.00      0.00      0.00         1
          42       0.00      0.00      0.00         1
          45       0.00      0.00      0.00         1
          47       0.00      0.00      0.00         0
          48       0.00      0.00      0.00         0
          53       0.00      0.00      0.00         3
          54       0.00      0.00      0.00         1
          55       0.00      0.00      0.00         1
          56       0.00      0.00      0.00         3
          57       0.00      0.00      0.00         1
          58       0.00      0.00      0.00         1
          61       0.00      0.00      0.00         1
          64       0.00      0.00      0.00         2
          66       0.00      0.00      0.00         1
          68       0.00      0.00      0.00         2
          71       0.00      0.00      0.00         1
          72       0.00      0.00      0.00         0
          78       0.00      0.00      0.00         0
          79       0.00      0.00      0.00         2
          82       0.00      0.00      0.00         1
          83       0.00      0.00      0.00         2
          84       0.00      0.00      0.00         0
          89       0.00      0.00      0.00         3
          90       0.00      0.00      0.00         1
          91       0.00      0.00      0.00         1
          92       0.00      0.00      0.00         0
          93       0.00      0.00      0.00         1
          94       0.00      0.00      0.00         0
```

95	0.00	0.00	0.00	1
100	0.00	0.00	0.00	2
101	0.00	0.00	0.00	1
103	0.00	0.00	0.00	2
104	0.00	0.00	0.00	1
108	0.00	0.00	0.00	1
110	0.00	0.00	0.00	3
111	0.00	0.00	0.00	3
112	0.00	0.00	0.00	1
115	0.00	0.00	0.00	1
116	0.00	0.00	0.00	1
121	0.00	0.00	0.00	1
122	0.00	0.00	0.00	3
124	0.00	0.00	0.00	2
127	0.00	0.00	0.00	1
128	0.00	0.00	0.00	0
132	0.00	0.00	0.00	0
135	0.00	0.00	0.00	1
136	0.00	0.00	0.00	0
137	0.00	0.00	0.00	0
139	0.00	0.00	0.00	6