IBM PHASE-2

PROJECT SUBMISSION

Project title: Sentiment analysis for marketing – Guideline

Phase 2 : Innovation

INTRODUCTION:

- Sentiment analysis is a valuable tool in marketing that allows businesses to gain insights into consumer opinions, attitudes, and emotions towards their products or services. The insights obtained from sentiment analysis can be used to make data-driven decisions, tailor marketing strategies, and enhance customer experiences. In this phase, we will discuss how advanced techniques like ensemble methods, deep learning architectures, and fine-tuning pre-trained sentiment analysis models (BERT, RoBERTa) can be applied to improve the accurarobustness of sentiment prediction systems.

INNOVATIVE TECHNIQUES:

1. Ensemble Methods :

Ensemble methods involve combining multiple models to improve predictive performance. In the context of sentiment analysis, you could create an ensemble of various sentiment analysis models, each with its strengths and weaknesses. By aggregating their predictions, you can often achieve a more accurate and robust sentiment analysis system.

2. Deep Learning Architectures:

Deep learning architectures, such as recurrent neural networks (RNNs), long short-term memory (LSTM) networks, and convolutional neural networks (CNNs), have shown promising results in sentiment analysis. These models can capture complex patterns and relationships in textual data, making them suitable for sentiment prediction tasks.

3. Fine-Tuning Pre-trained Models (e.g., BERT, RoBERTa) :

Pre-trained models like **BERT** (Bidirectional Encoder Representations from Transformers) and **RoBERTa** (A Robustly Optimized BERT Pretraining Approach) are state-of-the-art models in natural language processing. Fine-tuning these models specifically for sentiment analysis tasks can significantly improve the accuracy of sentiment predictions. Fine-tuning involves training the pre-trained models on domain-specific data related to sentiment analysis, adapting them to the specific nuances and vocabulary of the marketing domain.

- **Data Preparation :** Prepare a dataset of labeled sentiment data related to your marketing domain. Ensure that the dataset is appropriately preprocessed and labeled with sentiment labels (e.g., positive, negative, neutral).

- **Fine-Tuning Process :** Take the pre-trained **BERT** or **RoBERTa** model and train it on your labeled dataset using specialized fine-tuning techniques. Fine-tuning involves adjusting the model's parameters to better predict sentiment in your specific context.

- **Validation and Evaluation :** Validate the fine-tuned model using a separate validation dataset and evaluate its performance using metrics such as accuracy, precision, recall, and F1 score.

- **Hyperparameter Tuning :** Experiment with different hyperparameters, optimization algorithms, and learning rates to further optimize the model's performance.

- **Model Deployment :** Once satisfied with the performance, deploy the fine-tuned model in your sentiment analysis system to predict sentiment accurately for marketing-related text.

Import Necessary Libraries:

Import necessary libraries such as numpy,pandas,seaborn,matplotlib,pyplot..etc

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.ensemble import VotingClassifier
import matplotlib.pyplot as plt
import seaborn as sns
```

Load the Dataset:

```
# "Tweets.csv" is in the current directory
df = pd.read_csv("Tweets.csv")
```

	tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason	negativereason_confidence	airline	airline_sentime
0	570306133677760513	neutral	1.0000	NaN	NaN	Virgin America	
1	570301130888122368	positive	0.3486	NaN	0.0000	Virgin America	
2	570301083672813571	neutral	0.6837	NaN	NaN	Virgin America	
3	570301031407624196	negative	1.0000	Bad Flight	0.7033	Virgin America	
4	570300817074462722	negative	1.0000	Can't Tell	1.0000	Virgin America	

Data Cleaning:

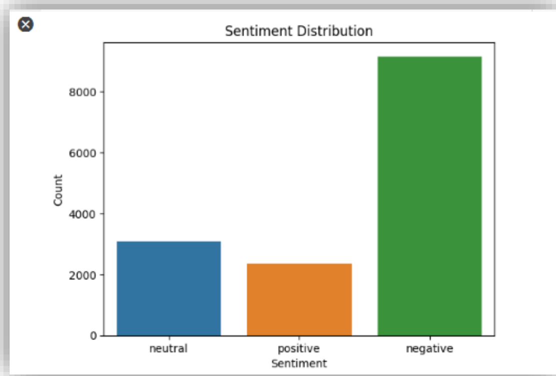
```
[6] # Drop unnecessary columns
    df = df.drop(['tweet_id', 'airline_sentiment_gold', 'negativereason_gold'], axis=1)

    # Drop rows with missing sentiment values
    df = df.dropna(subset=['airline_sentiment'])

    # Remove duplicate tweets
    df = df.drop_duplicates()

    # Convert the 'tweet_created' column to datetime
    df['tweet_created'] = pd.to_datetime(df['tweet_created'])

    # Explore the structure of the DataFrame
    print(df.info())
    print(df.head())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 14601 entries, 0 to 14639
Data columns (total 12 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   airline_sentiment             14601 non-null  object
 1   airline_sentiment_confidence  14601 non-null  float64
 2   negativereason                9157 non-null   object
 3   negativereason_confidence     10501 non-null  float64
 4   airline                       14601 non-null  object
 5   name                          14601 non-null  object
 6   retweet_count                 14601 non-null  int64
 7   text                          14601 non-null  object
 8   tweet_coord                   1015 non-null   object
 9   tweet_created                 14601 non-null  datetime64[ns, pytz.FixedOffset(-480)]
 10  tweet_location                9879 non-null   object
 11  user_timezone                 9789 non-null   object
dtypes: datetime64[ns, pytz.FixedOffset(-480)](1), float64(2), int64(1), object(8)
memory usage: 1.4+ MB
None
  airline_sentiment  airline_sentiment_confidence negativereason  \
0           neutral                        1.0000            NaN
1          positive                        0.3486            NaN
2           neutral                        0.6837            NaN
3          negative                        1.0000     Bad Flight
4          negative                        1.0000     Can't Tell
```

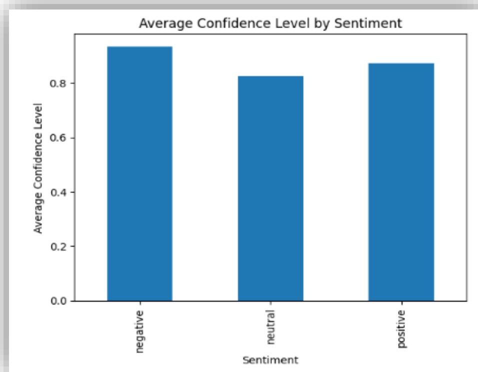
Data Analysis:

```
# Sentiment distribution
sentiment_counts = df['airline_sentiment'].value_counts()
sns.countplot(data=df, x='airline_sentiment')
plt.xlabel('Sentiment')
plt.ylabel('Count')
plt.title('Sentiment Distribution')
plt.show()

# Sentiment distribution by airline
plt.figure(figsize=(12, 6))
sns.countplot(data=df, x='airline', hue='airline_sentiment')
plt.xlabel('Airline')
plt.ylabel('Count')
plt.title('Sentiment Distribution by Airline')
plt.legend(title='Sentiment', loc='upper right')
plt.show()

# Average confidence level for each sentiment
avg_confidence = df.groupby('airline_sentiment')['airline_sentiment_confidence'].mean()
avg_confidence.plot(kind='bar')
plt.xlabel('Sentiment')
plt.ylabel('Average Confidence Level')
plt.title('Average Confidence Level by Sentiment')
plt.show()
```

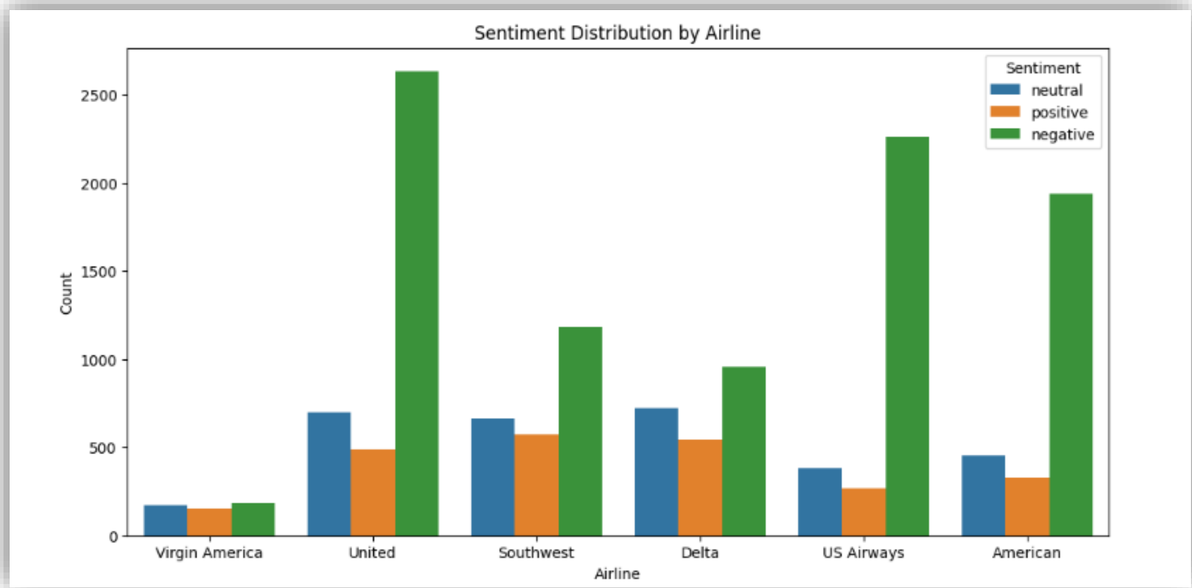
Sentiment distribution



Sentiment distribution by airline



Average confidence level for each sentiment



CONCLUSION:

These steps will help you import the dataset, clean the data, and perform basic data analysis to understand the sentiment distribution and other insights related to the "Twitter US Airline Sentiment" dataset.