Sentiment Analysis for Marketing

PHASE -5

DATE	1 November 2023
TEAM ID	Proj-212173-Team-1
PROJECT NAME	Sentiment analysis for Marketing

INDEX

1)List of figures and tables:

2)Abstract:

Sentiment analysis is a powerful tool in marketing that involves the use of natural language processing and machine learning techniques to analyze and interpret the emotions, opinions, and attitudes expressed in textual data, such as customer reviews, social media posts, and online comments.

The primary goal of sentiment analysis in marketing is to gain valuable insights into how customers perceive a brand, product, or service, and to use this information to make data-driven decisions and improve marketing strategies.

This abstract explores the utilization of BERT in marketing, where it excels in comprehending the nuances and context of customer opinions and emotions from various textual sources, including social media, product reviews, and online conversations.

BERT-based sentiment analysis, businesses can make more informed, data-driven decisions, enhance customer experiences, and stay ahead in a highly competitive market."

BERT's bidirectional context understanding enables it to capture intricate language structures, thereby offering more accurate sentiment analysis results.

Sentiment analysis also involves feature extraction to capture relevant keywords and context that contribute to the overall sentiment.

Once sentiment scores are assigned, businesses and marketers can derive valuable insights, such as identifying trends and areas for improvement in their products, services, or marketing strategies.

3)Introduction:

Sentiment analysis, in the context of marketing, is a powerful tool that enables businesses to gain valuable insights into how customers feel about their products, services, and brand as a whole. In today's highly competitive marketplace, understanding and harnessing customer sentiment is critical for making informed business decisions and crafting effective marketing strategies.

Sentiment analysis, also known as opinion mining, involves using natural language processing and machine learning techniques to analyze text data, such as social media posts, customer reviews, and surveys, in order to determine the emotional tone, attitudes, and opinions expressed within the content. By systematically classifying this textual data as positive, negative, or neutral, companies can gauge customer satisfaction, identify areas for improvement, and measure the success of their marketing efforts.

This tool is invaluable for marketing professionals because it allows them to:

1. Understand Customer Perception: Sentiment analysis helps marketers gauge how customers perceive their products, services, and brand. By analyzing social media mentions and online reviews, businesses can uncover what people like and dislike about their offerings.

2. Competitive Analysis: Marketers can compare the sentiment surrounding their brand to that of their competitors. This helps identify strengths and weaknesses in the market and informs strategies for differentiation.

3. Customer Engagement: By monitoring sentiment in real-time, marketers can engage with customers promptly. Addressing negative sentiment and acknowledging positive feedback demonstrates a commitment to customer satisfaction.

4. Product Development: Sentiment analysis can guide product or service enhancements based on customer feedback, ultimately leading to better products and greater customer satisfaction.

5. Content Creation: Understanding the sentiment of your target audience allows for more personalized and relevant content creation. It ensures that marketing messages align with customer emotions and preferences.

6. Campaign Evaluation: Marketers can measure the success of their marketing campaigns by tracking changes in sentiment before and after a campaign. This helps assess the impact of marketing efforts on customer perception.

7. Modules for sentiment analysis in marketing:

    *Social media monitoring

    *Customer review analysis

    *Product feedback analysis:

    *Market research analysis:

4)Literature Survey:

<u>Sentiment Analysis for Marketing</u>

i)A literature survey on the Naive Bayes algorithm:

The algorithm would involve reviewing research papers, articles, books, and other resources that discuss various aspects of Naive Bayes, its applications, improvements, and comparisons with other machine learning algorithms. Here's a summary of key topics and areas you might explore in such a survey

Introduction to Naive Bayes:

Start with an introduction to the Naive Bayes algorithm, explaining its probabilistic foundations and the "naive" assumption of feature independence.

Applications of Naive Bayes:

Investigate the various domains where Naive Bayes is commonly applied, such as text classification, spam detection, sentiment analysis, recommendation systems, and medical diagnosis.

Text Classification and Naive Bayes:

Explore how Naive Bayes is used in text classification tasks, such as document categorization and sentiment analysis. Review the effectiveness of Naive Bayes compared to other methods in these contexts.

Evaluation and Performance:

Analyze performance metrics and evaluation techniques for Naive Bayes models. Discuss commonly used metrics like accuracy, precision, recall, F1-score, and ROC curves.

ii)A literature survey on the k-Nearest Neighbors (KNN) algorithm:

The algorithm would involve reviewing research papers, articles, books, and other resources that discuss various aspects of KNN, its applications, enhancements, and comparisons with other machine learning algorithms. Here's a summary of key topics and areas you might explore in such a survey:

Introduction to k-Nearest Neighbors (KNN):

Start with an introduction to the KNN algorithm, explaining how it works and its basic principles in pattern recognition and classification.

KNN Algorithm Variants:

Investigate variations of the KNN algorithm, such as weighted KNN, distance-weighted KNN, and kernel-based KNN, and their respective advantages and use cases.

Parameter Selection:

Examine techniques for choosing the optimal value of "k" (the number of neighbors) and how different values of "k" affect the algorithm's bias-variance trade-off.

Outlier Detection:

Study the use of KNN in outlier detection and its effectiveness in identifying anomalies in datasets.

iii)A literature survey on logistic regression algorithm:

The algorithm would involve reviewing research papers, articles, books, and other resources that discuss various aspects of logistic regression, its applications, enhancements, and comparisons with other machine learning algorithms. Here's a summary of key topics and areas you might explore in such a survey:

Introduction to Logistic Regression:

Begin with an introduction to logistic regression, explaining its foundation as a binary classification algorithm and the logistic function used to model the probability of outcomes.

Logistic Regression Variants:

Investigate different variants of logistic regression, such as multinomial logistic regression (for multiclass classification) and ordinal logistic regression (for ordered categorical outcomes).

Applications of Logistic Regression:

Examine the various domains where logistic regression is commonly applied, such as medical diagnosis, marketing analytics, credit scoring, and social sciences.

Evaluation and Performance Metrics:

Analyze performance metrics for logistic regression models, including accuracy, precision, recall, F1-score, ROC curves, and AUC-ROC, and discuss how to interpret these metrics.

iv)A literature survey on Transformers algorithm :

The algorithm would involve reviewing research papers, articles, books, and other resources that discuss various aspects of Transformers, their applications, enhancements, and impact on natural language processing (NLP) and other machine learning tasks. Here's a summary of key topics and areas you might explore in such a survey:

Introduction to Transformers:

Begin with an introduction to Transformers, explaining their architecture and the mechanisms they use, including self-attention and positional encoding.

Transformer Variants:

Investigate different variants of the Transformer architecture, such as BERT, GPT, RoBERTa, and XLNet, each designed for specific NLP tasks.

Pre-training and Fine-tuning:

Explore the concept of pre-training on large text corpora and fine-tuning on specific tasks, which has been a breakthrough in NLP.

Applications of Transformers:

Examine the various applications of Transformers in NLP, including text classification, machine translation, sentiment analysis, question answering, and text generation.

Multimodal Transformers:

Investigate the extension of Transformer models to handle multimodal data, combining text and images or other modalities in tasks like image captioning and visual question answering.

Efficiency and Compression:

Study techniques for making Transformers more computationally efficient, such as distillation, pruning, and quantization.

v)A literature survey on decision trees algorithm :

The algorithm could cover various aspects, including their applications, algorithms, and advancements. Some key points to explore might include:

Introduction to Decision Trees:

Begin with an overview of what decision trees are, their importance in machine learning, and their role in decision-making.

Decision Tree Algorithms:

Discuss popular decision tree algorithms like ID3, C4.5, CART, and Random Forest. Highlight their strengths and weaknesses.

Pruning and Optimization:

Discuss techniques for pruning decision trees to prevent overfitting and improve their generalization capabilities.

Handling Categorical and Numeric Data:

Explain how decision trees handle different types of data and the techniques used for splitting and node evaluation.

5) PROBLEM DEFINITION:

To perform sentiment analysis on the given Twitter U.S.Airlines dataset to understand the customer satisfaction of each Airlines.

PROBLEM ANALYSIS:

Airline service sentimental analysis is the process of using natural language processing (NLP) and machine learning to identify the sentiment of customer feedback from social media, customer reviews, surveys, and other sources about airline services.

- **Identify common customer complaints**. It is used to identify common customer complaints, such as delayed flights, lost baggage, and rude staff. This information can be used to improve customer service and operations.
- **Track customer satisfaction over time.** It helps to track customer satisfaction over time. This information can be used to identify trends and to measure the effectiveness of customer service initiatives.
- **Identify and respond to customer complaints**. By using this the company can able to identify customer complaints on social media and other online platforms. This information can be used to respond to customer complaints promptly and to resolve issues.
- **Improve marketing campaigns.** It is mainly used to understand how customers perceive their marketing campaigns. This information can be used to improve the effectiveness of future marketing campaigns.

a)DESIGN THINKING:

BRAINSTORMING:

Brainstorming is a group creativity technique by which efforts are made to find a conclusion for a specific problem by gathering a list of ideas spontaneously contributed by its members. It is a way to generate a large number of ideas in a short period of time.

Using the mural template the ideas of each members in our team had been gathered to analyze the problem.

The ideas are,

- Monitor competitor's market tactics.
- Use NLP to analyze customer review.
- Generate automatic response to the customers.
- Identify the product influencers on social media.
- Tracking customers having interest on similar products.
- Identify trends in marketing over time.
- Track sentiment of customers during new product launch.
- Analyze buyer's expectations.

- Analyze the reason for the fall of your competitor and use it as an advantage.
- Late response from provider side irritates the customer.
- Monitor defect from our product side and resolve it.
- Bad reviews from former customer may influence the new ones so, try to resolve the defect in our product or services.
- Analyze the fall and raise of a product.

b)INNOVATION AND PROBLEM SOLVING:

1.GROUPING IDEAS:

Grouping ideas is a process of organizing ideas into categories or clusters based on their similarities. This can be done manually or using a variety of tools and techniques.

The ideas gathered in brainstorming are grouped based on their similarities.

GROUP 1:

- Use NLP to analyze customer review.
- Bad reviews from former customer may influence the new ones so, try to resolve the defect in our product or services.
- Resolve negative reviews.
- Late response from provider side.

GROUP 2:

- Analyze the fall and raise of a product.
- Monitor competitor's market tactics.
- Analyze the reason for the fall of your competitor and use it as an advantage.

GROUP 3:

- Analyze buyer's expectations.
- Find out the buyers who are more interested in your product.

2.PRIORITIZING IDEAS:

Idea prioritization is the process of evaluating and ranking ideas based on their potential value and feasibility, to determine which should be pursued and which should be set aside.

After grouping, based on their potential value the ideas are prioritized.

- Bad reviews from former customer may influence the new ones so, try to resolve the defect in our product or services.
- Late response from provider side.
- Analyze the fall and raise of a product.
- Find out the customers who are interested in your product.

6)IMPORTING DATASET AND PERFORMING DATA PREPROCESSING AND ANALYSIS

a)IMPORTING THE DATASET:

The U.S Twitter Airlines dataset from kaggle has been imported to perform sentiment analysis.

Dataset Link: https://www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment

DATA PREPROCESSING:

Data preprocessing is a crucial step in sentiment analysis, as it helps clean and prepare the text data for analysis. It involves the cleaning and transformation of raw data into a format that is suitable for analysis or for training machine learning models. The goal of data preprocessing is to enhance the quality of the data, making it more reliable and easier to work with.

Here are the steps you can follow to perform data preprocessing for sentiment analysis:

b)DATA CLEANING:

This involves identifying and correcting errors or inconsistencies in the data, such as missing values, outliers, and duplicates.

- ✓ **Handling missing values:** Identifying and filling in or removing missing data points.
- ✓ **Outlier detection and treatment:** Identifying and handling data points that are significantly different from the majority of the data.
- ✓ **Noise reduction:** Reducing random variations and errors in the data.

TEXT PREPROCESSING:

Text preprocessing in sentiment analysis is a crucial step that involves cleaning and transforming textual data to prepare it for analysis.

The goal of text preprocessing is to improve the quality of the text data, reduce noise, and make it suitable for sentiment analysis tasks.

TECHNIQUES USED IN TEXT PREPROCESSING :

- **Lowercasing:** Converting all text to lowercase helps ensure that the analysis is not case-sensitive. This way, "good" and "Good" are treated as the same word.

- **Tokenization:** Tokenization is the process of splitting text into individual words or tokens. It breaks down sentences or paragraphs into a list of words or sub-phrases, making it easier to analyze.

- **Removing Punctuation**: Removing punctuation marks like commas, periods, and exclamation points can help reduce noise and improve the accuracy of sentiment analysis.

- **Removing Stop Words:** Stop words are common words like "the," "and," "is," etc., that often do not carry significant sentiment information. Removing them can reduce the dimensionality of the data and improve processing speed.

- **Stemming and Lemmatization**: Stemming and lemmatization are techniques for reducing words to their root forms. For example, "running," "ran," and "runner" might be reduced to "run." This helps to group similar words together and reduce dimensionality.

- **Handling Emoticons and Emoji:** Sentiment analysis should take into account emoticons and emoji as they convey sentiment. You may choose to map them to sentiment labels.

- **Removing HTML Tags:** If dealing with text from web sources, it's common to encounter HTML tags. Removing these tags is essential to ensure that the analysis focuses on the text content.

- **Handling URLs and User Mentions:** URLs and user mentions (e.g., @username) are often irrelevant for sentiment analysis and can be removed or replaced.

- **Spell Checking and Correction:** Correcting spelling errors can improve the accuracy of sentiment analysis by ensuring that words are correctly recognized.

7) DATA VISUALIZATION

Data visualization is the graphical representation of data to help people understand and interpret the information contained within it. It involves creating visual representations, such as charts, graphs, maps, and dashboards, to present data in a way that is visually appealing, informative, and accessible. The primary goal of data visualization is to make complex data more understandable, revealing patterns, trends, and insights that may not be apparent from raw data alone.

Here are key aspects of data visualization:

1. Data Presentation: Data visualization transforms data into visual elements like lines, bars, points, shapes, colors, and text. These elements convey information more effectively than rows and columns of numbers.

2. Understanding Complex Data: Visualization simplifies complex data, enabling users to grasp information quickly and make data-driven decisions.

3. Revealing Patterns and Trends: Visualization can highlight patterns, trends, outliers, and correlations in the data, making it easier to draw conclusions and insights.

4. Communication: Data visualization is a powerful tool for communicating data and findings to a broad audience. It helps convey information in a way that is accessible to both technical and non-technical stakeholders.

5. Exploration and Discovery: Visualization can facilitate data exploration by allowing users to interact with data, zoom in on specific details, or filter data to discover hidden insights.

6. Decision-Making: Data visualizations are valuable for decision-making processes, as they provide a clear and concise representation of data that can support informed choices.

TYPES OF DATA VISUALIZATIONS USED ARE:

1)BAR CHART:

Bar charts can display the distribution of sentiment categories (e.g., positive, negative, neutral) in a dataset. Each sentiment category is represented by a bar, and the height of the bar corresponds to the number of occurrences, providing a clear view of sentiment distribution.

- To display sentiment distribution by negative reason (Count the number of the negative reasons).
- To visualize number of tweets for each airlines.
- To graphically represent the sentiment distribution (positive, negative and neutral) for different airlines.

Sub plots are used to represents the sentiment distribution (positive, negative and neutral) of all airlines.

2)PIE CHART:

Pie charts are **used to visualize sentiment distribution (positive, negative, neutral).** Each sentiment category is represented as a slice of the pie, with the size of the slice proportional to the percentage of each sentiment in the dataset.

3)HEAT MAPS:

Heat maps can be **used to visualize the sentiment of text data over different airlines** with their airline sentiment confidence score.

4)SCATTER PLOT:

Scatter plots can be used to display the relationship between airline sentiment confidence and the negative reason confidence.

5)VIOLIN PLOT:

Violin plots can display the distribution of negative reason confident scores of different airlines**.** They show the shape of the distribution, including the median and quartiles, and can help identify differences in sentiment among groups.

6)BOX PLOT:

Box plots are used to display the distribution of airline sentiment confidence and negative reason confidence.

7)WORD CLOUDS:

Word clouds visually represent the most frequently occurring words in a dataset, with word size indicating frequency. It is used to visualize the negative reviews from customer.

8)Model development and evaluation:

Bert:

BERT, which stands for Bidirectional Encoder Representations from Transformers, is a natural language processing (NLP) model developed by Google in 2018. It's designed to understand the context of words in a sentence by considering the surrounding words, both before and after a given word.

BERT is a pre-trained model. In the context of natural language processing (NLP), "pre-trained" means that the model is initially trained on a large corpus of text data before fine-tuning it for specific NLP tasks.

Key characteristics of BERT:

1. Transformer Architecture: BERT is based on the Transformer architecture, a deep learning model that has proven highly effective for a wide range of NLP tasks. Transformers use self-attention mechanisms to capture relationships between words in a sentence.

2.Pre-training and Fine-tuning: BERT goes through a two-step process. In the pre-training phase, the model is trained on a large corpus of text data. During this phase, BERT learns to predict missing words in sentences, considering the surrounding context. This pre-training results in a general understanding of language. In the fine-tuning phase, the pre-trained model is adapted to specific NLP tasks like text classification, named entity recognition, and question-answering.

3.Contextual Word Embeddings: BERT produces contextual word embeddings, which means the meaning of a word can vary depending on the words around it. This is in contrast to traditional word embeddings like Word2Vec or GloVe, which produce fixed, context-independent representations of words.

4.Deep and Bidirectional: BERT is a deep model with multiple layers, allowing it to capture complex language patterns. It's also bidirectional, so it looks at both preceding and following words when processing a word in a sentence.

5.State-of-the-Art Performance: BERT achieved state-of-the-art results on a wide range of NLP benchmarks when it was introduced. It significantly improved the accuracy of NLP tasks, including sentiment analysis, language translation, and text summarization.

6.Multi-lingual Capabilities: BERT has been pre-trained in multiple languages, enabling it to understand and generate text in a variety of languages, making it a versatile model for multilingual NLP applications.

Training and Testing using BERT:

Training and testing a sentiment analysis model for marketing using the BERT algorithm involves several steps. BERT (Bidirectional Encoder Representations from Transformers) is a powerful pre-trained language model that can be fine-tuned for various natural language processing tasks, including sentiment analysis.

. Here's a step-by-step guide on how to do this:

1. Data Collection:

Gather labeled data for sentiment analysis. This data should consist of text samples (e.g., customer reviews, social media comments) along with their corresponding sentiment labels (e.g., positive, negative, neutral).

2. Data Preprocessing:

Clean and preprocess the data, which may include tasks such as lowercasing, removing punctuation, tokenization, and handling special characters.

3. Tokenization:

Use the BERT tokenizer to convert text data into subword tokens. BERT uses WordPiece tokenization, which splits words into smaller units.

4. Pre-trained BERT Model:

Download a pre-trained BERT model. You can use pre-trained BERT models from Hugging Face's Transformers library. These models are available in various sizes (e.g., BERT-Base, BERT-Large).

5. Fine-Tuning:

Fine-tune the pre-trained BERT model on your sentiment analysis dataset. This involves training the model on your labeled data to adapt it to the specific task of sentiment classification.We'll need to create an appropriate architecture (often adding a classification layer) and define loss functions.

We can use popular deep learning frameworks like PyTorch or TensorFlow to implement this fine-tuning. Transfer learning techniques are usually applied, where you load the pre-trained weights and fine-tune the model on your data for a few epochs.

6. Training Parameters:

Set training parameters, such as batch size, learning rate, and the number of epochs. Experiment with hyperparameters to optimize model performance.

7. Evaluation:

After training, evaluate the model's performance on the testing dataset. Common evaluation metrics for sentiment analysis include accuracy, precision, recall, F1 score, and confusion matrices.

8. Hyperparameter Tuning:

If the model's performance is not satisfactory, you may need to perform hyperparameter tuning, such as adjusting the learning rate or batch size, or exploring different model architectures.

9. Model Deployment:

Once you are satisfied with the model's performance, you can deploy it for marketing sentiment analysis. This can be in the form of an API, web application, or integrated into your marketing analytics tools.

10. Continuous Monitoring:

Continuously monitor the model's performance in a real-world marketing context. Re-train and fine-tune the model as necessary to keep it up to date and accurate.

11. Sentiment Visualization and Reporting:

Visualize and report sentiment analysis results to extract valuable insights for marketing strategies.

Remember that BERT is a resource-intensive model, and fine-tuning may require substantial computational resources. Pre-trained models, such as DistilBERT or RoBERTa, can be viable alternatives if computational limitations are a concern.

Confusion matrix:

A confusion matrix is a useful tool for evaluating the performance of a classification model, such as a sentiment analysis model used in marketing. It provides a clear summary of the model's predictions and how they compare to the actual labels in the dataset. A typical confusion matrix for sentiment analysis has four components: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

Components of a confusion matrix for sentiment analysis in a marketing :

True Positives (TP): These are the cases where the model correctly predicted a positive sentiment. In marketing, this means the model correctly identified instances where customers or users expressed positive opinions or sentiments about a product, service, or brand.

True Negatives (TN): These are the cases where the model correctly predicted a negative sentiment. In a marketing context, it means the model correctly identified instances where customers expressed negative opinions or sentiments.

False Positives (FP): These are the cases where the model incorrectly predicted a positive sentiment when the actual sentiment was negative. In marketing, this might represent situations where the model mistakenly identifies negative comments as positive. This could lead to missed opportunities to address customer concerns.

False Negatives (FN): These are the cases where the model incorrectly predicted a negative sentiment when the actual sentiment was positive. In marketing, this could indicate that the model fails to recognize positive feedback, potentially missing opportunities to promote or amplify positive customer experiences.

Accuracy:

This is the overall correctness of the model's predictions and is calculated as (TP + TN) / (TP + TN + FP + FN). It represents the percentage of correctly classified instances.

Precision:

Precision measures the proportion of positive predictions that were correct. It is calculated as TP / (TP + FP). In marketing, it shows how many of the identified positive sentiments were actually correct.

Recall (Sensitivity):

Recall measures the proportion of actual positive cases that the model correctly predicted as positive. It is calculated as TP / (TP + FN). In marketing, it shows how effectively the model captures positive sentiments.

F1-Score:

The F1-score is the harmonic mean of precision and recall and is often used to balance these two metrics. It is calculated as 2 * (Precision * Recall) / (Precision + Recall).

Specificity:

Specificity measures the proportion of actual negative cases that the model correctly predicted as negative. It is calculated as TN / (TN + FP). In marketing, it shows how effectively the model captures negative sentiments.

By examining the confusion matrix and associated metrics, you can get a clear picture of how well your sentiment analysis model is performing in the marketing context. This information can help you refine your model, improve marketing strategies, and better respond to customer sentiments.

9)CODE SAMPLE:

```
#Load the dataset
df=pd.read_csv('Tweets.csv')
#df.head() returns first five rows
df.head()
```

```
#df.fillna() is used to fill the missing values
df['airline_sentiment_confidence'].fillna(df['airline_sentiment_confidence'].mean(), inplace=True)
df['negativereason_confidence'].fillna(df['negativereason_confidence'].median(), inplace=True)
df['negativereason'].fillna(df['negativereason'].mode(),inplace=True)
df['user_timezone'].fillna(method='ffill', inplace=True)
col=["negativereason_gold","airline_sentiment_gold","tweet_coord","tweet_location"]
df.drop(col,axis=1,inplace=True)
df['negativereason'].fillna('No text', inplace=True)
#Recheck whether the dataframe has null values or not
df.isnull().sum()
```

```
#Text Preprocessing
#Lowercasing the text
df['new_text'] = df['text'].astype(str).str.lower()
df['new_text']
```

```
def clean_txt(text):

    text=re.sub(r'@[a-zA-Z0-9]+','',text)#removes username
    text=re.sub(r'#\w+','',text)#removes hashtag
    text=re.sub(r'https?:/\/\s+','',text)#removes URL
    text=re.sub(r'RT[\s]+','',text)#removes retweet
    return text
df['new_text']=df['new_text'].astype(str).apply(clean_txt)
df['new_text']
```

```
#Removing Punctuation
def remove_punctuation(text):
    return ''.join([char for char in text if char not in string.punctuation])

df['new_text'] = df['new_text'].apply(remove_punctuation)
df['new_text']
```

```
#Tokenization

nltk.download('punkt')
from nltk.tokenize import word_tokenize
def tokenize_text(text):

    tokens = word_tokenize(text)
    return tokens

df['new_text'] = df['new_text'].astype(str).apply(word_tokenize)

df['new_text']
```

```
#Removing emojis
demoji.download_codes()


def remove_emojis(text):
    return demoji.replace(text, '')
df['new_text'] = df['new_text'].apply(remove_emojis)
df['new_text']
```

```python
from torch.utils.data import Dataset, DataLoader

# Define a custom dataset, more info on how to build custom dataset can be
# found at https://pytorch.org/tutorials/beginner/data_loading_tutorial.html
class CustomDataset(Dataset):

    def __init__(
        self,
        tweets,
        labels,
        tokenizer,
        max_length
    ):
        self.tweets = tweets
        self.labels = labels
        self.tokenizer = tokenizer
        self.max_length = max_length

    def __len__(self):
        return len(self.tweets)

    def __getitem__(self, idx):
        tweet = self.tweets[idx]
        label = self.labels[idx]
```

```python
        tokenize = self.tokenizer.encode_plus(
            tweet,
            add_special_tokens=True,
            max_length=self.max_length,
            return_token_type_ids=False,
            padding='max_length',
            return_attention_mask=True,
            return_tensors='pt'
        )
        return {
            'tweet': tweet,
            'input_ids': tokenize['input_ids'].flatten(),
            'attention_mask': tokenize['attention_mask'].flatten(),
            'targets': torch.tensor(label, dtype=torch.long)}
```

```python
MAX_LENGTH = 64
TEST_SIZE = 0.1
VALID_SIZE = 0.5
BATCH_SIZE = 16
NUM_WORKERS = 2

train_sampler, test_sampler = train_test_split(df, test_size=TEST_SIZE, random_state=RANDOM_STATE)
valid_sampler, test_sampler = train_test_split(test_sampler, test_size=VALID_SIZE, random_state=RANDOM_STATE)

train_set = CustomDataset(
    train_sampler['text'].to_numpy(),
    train_sampler['labels'].to_numpy(),
    tokenizer,
    MAX_LENGTH
)
test_set = CustomDataset(
    test_sampler['text'].to_numpy(),
    test_sampler['labels'].to_numpy(),
    tokenizer,
    MAX_LENGTH
)
valid_set = CustomDataset(
    valid_sampler['text'].to_numpy(),
    valid_sampler['labels'].to_numpy(),
    tokenizer,
    MAX_LENGTH
)
```

```python
train_loader = DataLoader(train_set, batch_size=BATCH_SIZE, num_workers=NUM_WORKERS)
test_loader = DataLoader(test_set, batch_size=BATCH_SIZE, num_workers=NUM_WORKERS)
valid_loader = DataLoader(valid_set, batch_size=BATCH_SIZE, num_workers=NUM_WORKERS)
```

```python
from torch import nn
class AirlineSentimentClassifier(nn.Module):

    def __init__(self, num_labels):
        super (AirlineSentimentClassifier, self).__init__()
        self.bert = BertModel.from_pretrained(MODEL)
        self.dropout = nn.Dropout(p=0.2)
        self.classifier = nn.Linear(self.bert.config.hidden_size, num_labels)

    def forward(self, input_ids, attention_mask):
        outputs = self.bert(
            input_ids=input_ids,
            attention_mask=attention_mask
        )
        pooled_output = outputs[1]
        pooled_output = self.dropout(pooled_output)
        out = self.classifier(pooled_output)
        return out
```

```python
n_epochs = 10
learning_rate =2e-5

# Loss function
criterion = nn.CrossEntropyLoss()

# Optimizer
optimizer = AdamW(model.parameters(), lr=learning_rate, correct_bias=False)

# Define scheduler
training_steps = len(train_loader)*n_epochs
scheduler = get_linear_schedule_with_warmup(
    optimizer,
    num_warmup_steps=0,
    num_training_steps=training_steps
)
```

```python
# Track changes in validation loss
valid_loss_min = np.Inf

for epoch in range(1, n_epochs+1):

    # Setting training and validation loss
    train_loss = []
    validation_loss = []
    tr_predictions = 0
    acc = 0
    val_predictions = 0

    ####################
    #  Train the model #
    ####################
    model = model.train()
    for data in train_loader:

        # Moving tensors to GPU on CUDA enabled devices
        if device:
            input_ids, attention_mask, targets = data["input_ids"].cuda(), data["attention_mask"].cuda(), data["targets
        # Clear the gradients of variables
        optimizer.zero_grad()
```

```
    #### Forward pass
    # Pass input through the model
    output = model(
        input_ids=input_ids,
        attention_mask=attention_mask
    )
    # Compute batch loss
    loss = criterion(output, targets)
    # Convert output probabilities to class probabilities
    _, pred = torch.max(output, 1)
    # Track correct predictions
    tr_predictions += torch.sum(pred == targets)
```

```
    #### Backward Pass
    # Compute gradients wrt to model parameters
    loss.backward()
    # To avoid exploding gradients, we clip the gradients of the model
    nn.utils.clip_grad_norm_(model.parameters(), max_norm=1.0)
    # Perform parameter update
    optimizer.step()
    # Update learning rate
    scheduler.step()
    # Update loss per mini batches
    train_loss.append(loss.item())
```

```
######################
# Validate the model #
######################
model.eval()
with torch.no_grad():
    for data in valid_loader:
```

```python
    # Compute accuracy
    train_accuracy = tr_predictions.double()/len(train_sampler)
    val_accuracy = val_predictions.double()/len(valid_sampler)

    # Print loss statistics
    print('Epoch: {}/{} \n\tTraining Loss: {:.6f} \n\tValidation Loss: {:.6f} \n\tTrain Accuracy: {:.6f} \n\tVal Accura

    # Save model if validation loss is decreased
    if val_accuracy > acc:
        print('Saving model...')
        torch.save(model.state_dict(), 'bert_base_fine_tuned.pt')
        acc = val_accuracy
```

```python
# Track test loss
test_loss = 0.0
class_predictions = list(0. for i in range(3))
class_total = list(0. for i in range(3))
predictions = []
labels = []

model.eval()
with torch.no_grad():
    for data in test_loader:

        # Moving tensors to GPU on CUDA enabled devices
        if device:
            input_ids, attention_mask, targets = data["input_ids"].cuda(), data["attention_mask"].cuda(), data["targets

        #### Forward pass
        # Pass input through the model
        output = model(
            input_ids=input_ids,
            attention_mask=attention_mask
        )
        # Compute batch loss
        loss = criterion(output, targets)
        # Update loss
        test_loss += loss.item()
        # convert output probabilities to predicted class
        _, pred = torch.max(output, 1)

        predictions.extend(pred)
        labels.extend(targets)

predictions = torch.stack(predictions) if not device else torch.stack(predictions).cpu()
labels = torch.stack(labels) if not device else torch.stack(labels).cpu()
```

```python
cm = confusion_matrix(labels, predictions)
heatmap = sns.heatmap(cm, annot=True, fmt='d', cmap='Greens')
heatmap.yaxis.set_ticklabels(heatmap.yaxis.get_ticklabels(), rotation=0, ha='right')
heatmap.xaxis.set_ticklabels(heatmap.xaxis.get_ticklabels(), rotation=30, ha='right')
plt.xlabel('True sentiment')
plt.ylabel('Predicted sentiment');
```

10)OUTPUT:

Figure 1

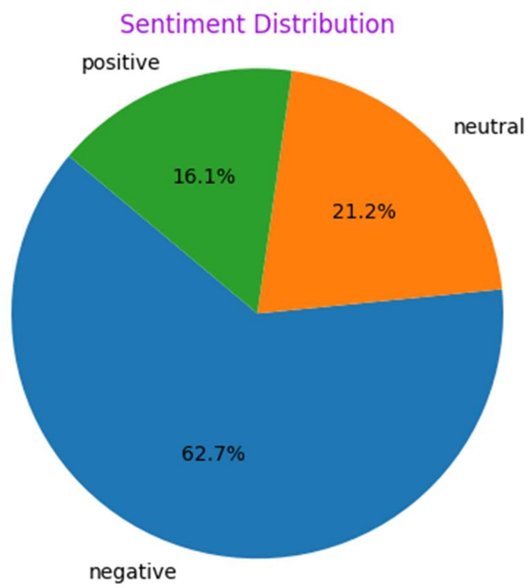


Figure 2

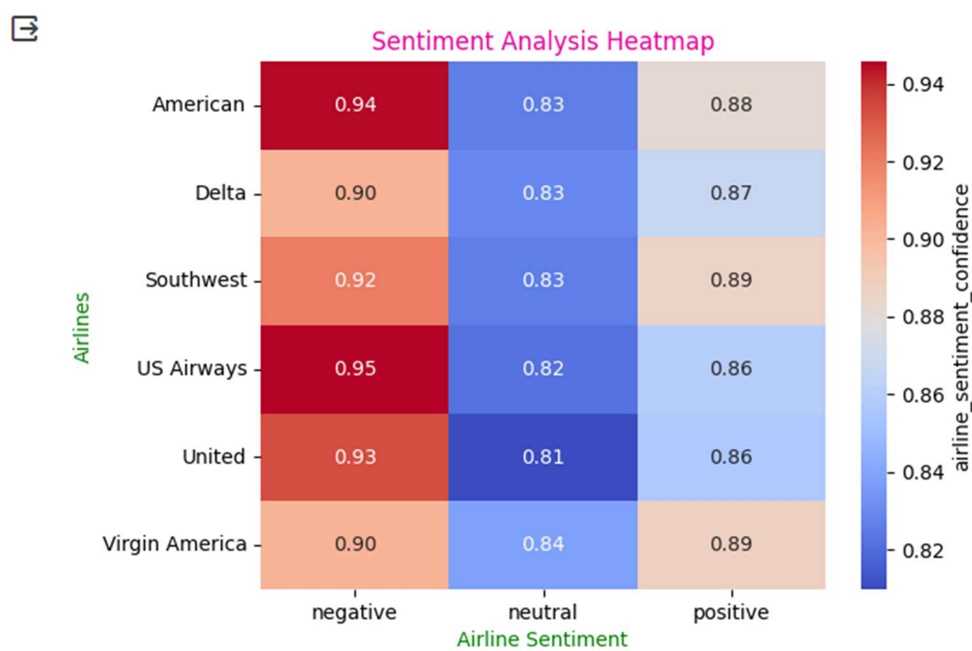


Figure 3

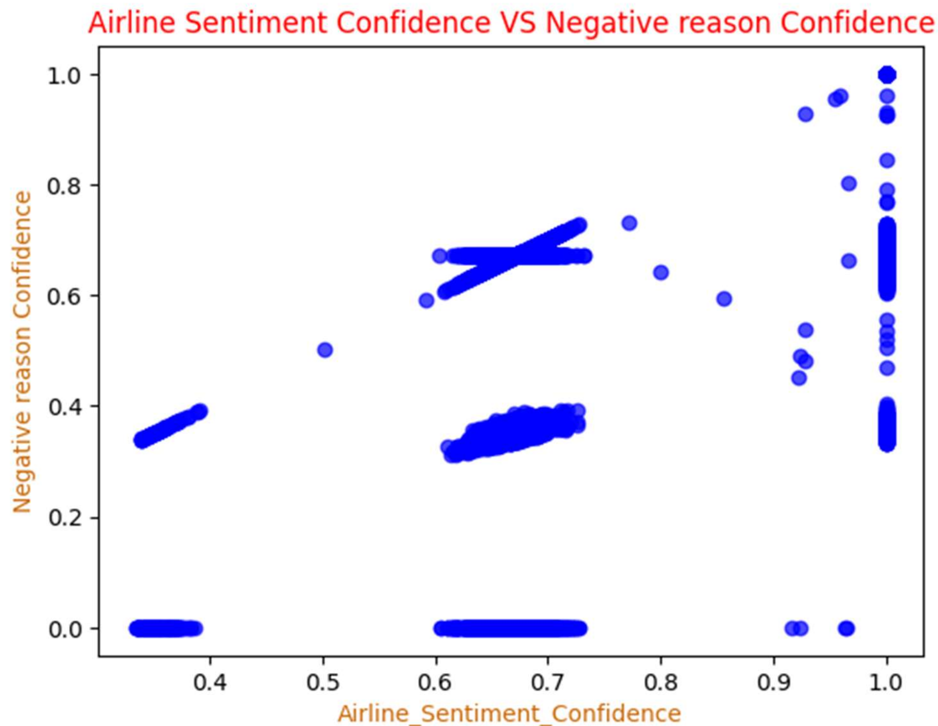


Figure 4

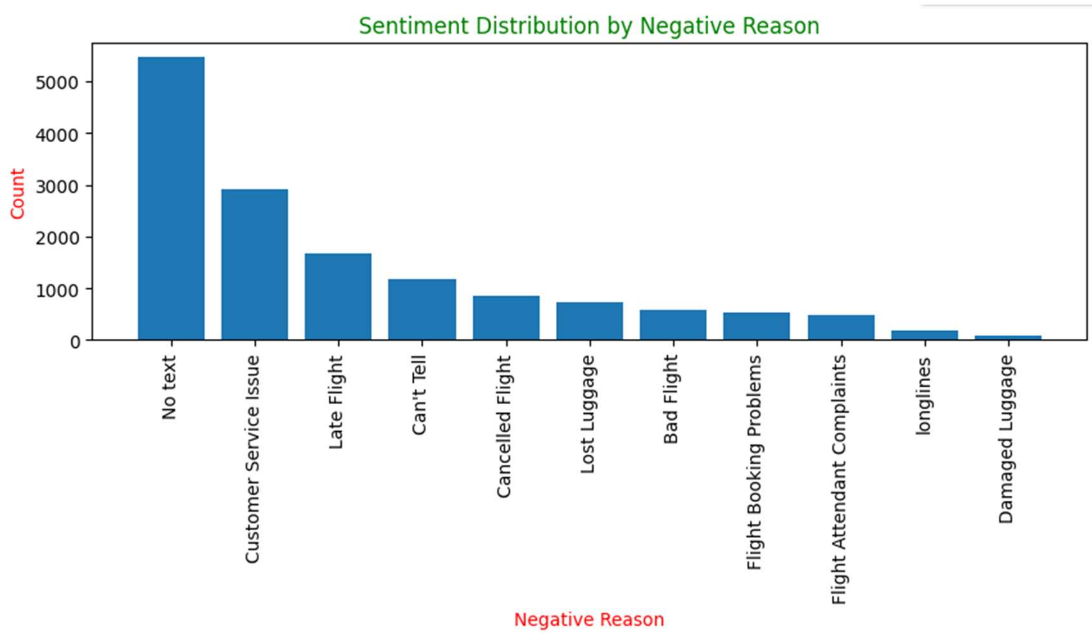
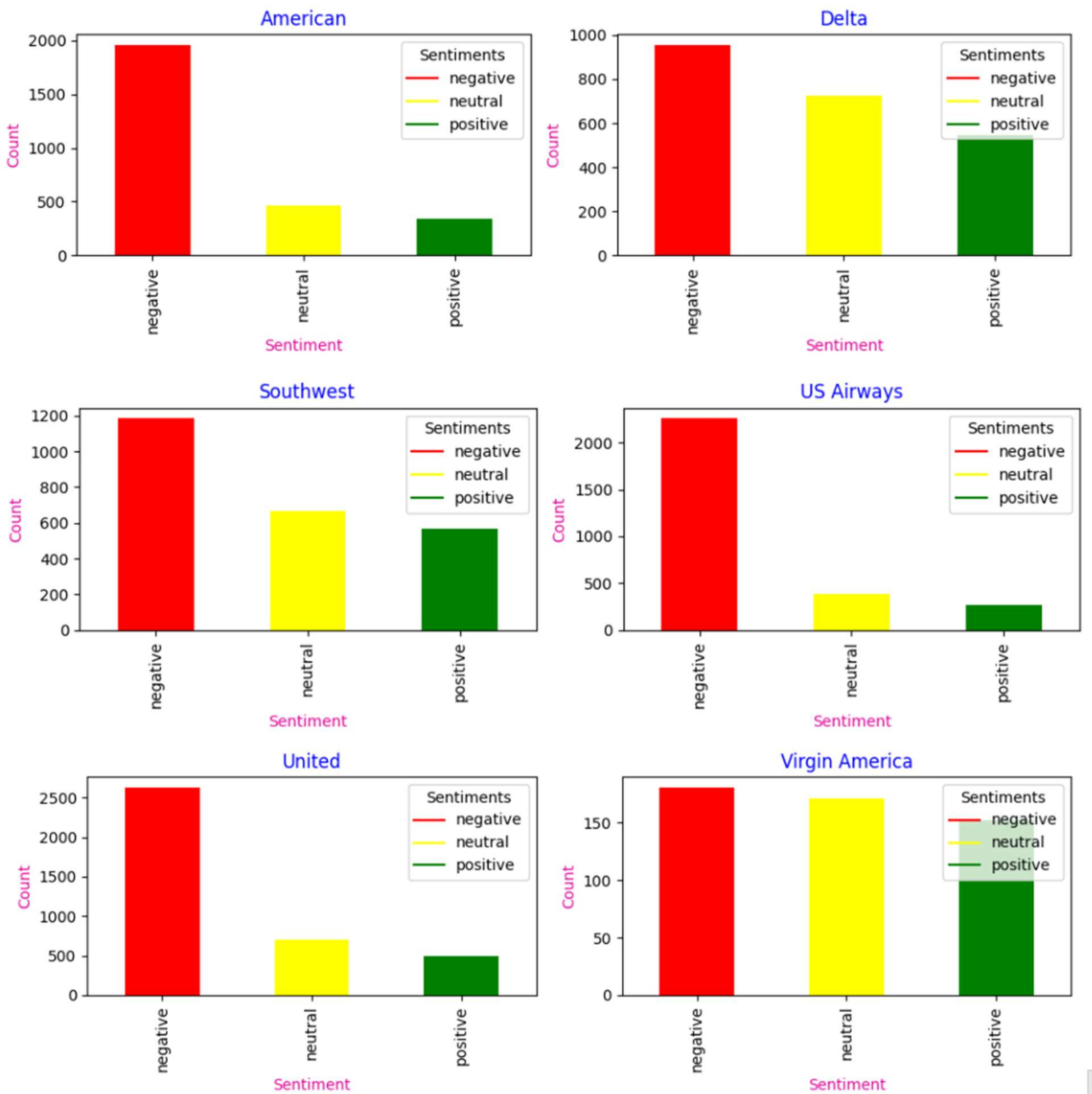


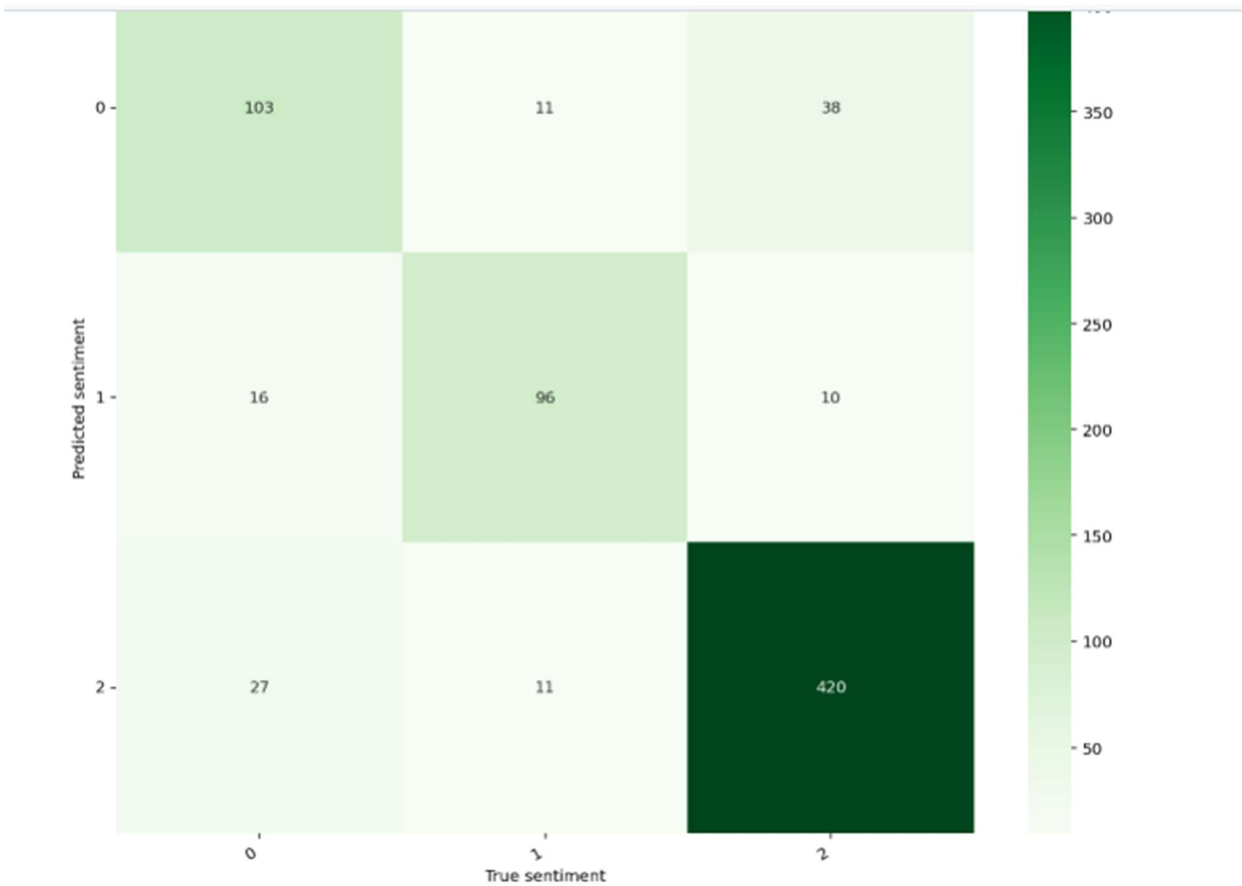
Figure 5


American
Delta
Southwest
US Airways
United
Virgin America
Sentiments
negative
neutral
positive
Count
Sentiment

	precision	recall	f1-score	support
neutral	0.68	0.71	0.69	146
positive	0.79	0.81	0.80	118
negative	0.92	0.90	0.91	468
accuracy			0.85	732
macro avg	0.79	0.81	0.80	732
weighted avg	0.85	0.85	0.85	732

Figure 6

CONFUSION MATRIX:



11)CONCLUSION:

Improved Customer Understanding:

Sentiment analysis helps marketers gain a deeper understanding of customer sentiment, allowing them to tailor their marketing strategies effectively.

Real-time Feedback:

The real-time nature of sentiment analysis enables rapid responses to customer feedback, enhancing customer satisfaction and loyalty.

Competitive Advantage:

Marketers can stay ahead of competitors by monitoring sentiment trends and making data-driven decisions.

Content Optimization:

Sentiment analysis can guide content creation by identifying what resonates positively with the audience and what doesn't.

Risk Mitigation:

Identifying negative sentiment early allows for proactive reputation management and issue resolution.

12)FUTURE ENHANCEMENT:

Multimodal Analysis:

Integrating sentiment analysis with image, audio, and video data to provide a more comprehensive understanding of customer sentiment.

Contextual Analysis:

Improving sentiment analysis models to understand context, sarcasm, and irony in text, leading to more accurate results.

Industry-Specific Models:

Developing sentiment analysis models tailored to specific industries to account for domain-specific language and nuances.

Emotion Detection:

Moving beyond positive/negative sentiment to identify specific emotions (e.g., joy, anger, sadness) expressed in customer feedback.

Personalization:

Customizing marketing strategies based on individual sentiment analysis, providing personalized experiences for customers.

Ethical Considerations:

Addressing ethical concerns in sentiment analysis, such as bias and privacy issues, to ensure fair and responsible use of the technology.

REFERENCE:

Naive Bayes algorithm

https://github.com/Anusaya2k3/-Sentimental-analysis-for-marketing.git

k-Nearest Neighbors (KNN) algorithm

https://github.com/SahanaKandukuri/Sentiment-Analysis-for-Marketing.git

Logistic regression algorithm

https://github.com/Devil1405/Sentiment-Analysis-For-Marketing.git

Transformers algorithm

https://github.com/Pachaiammal-PV/Sentiment-Analysis-for-Marketing.git

Decision tree algorithm

https://github.com/SahanaKandukuri/Sentiment-Analysis-for-Marketing.git