**PROJECT: Sentiment analysis for marketing**

**Phase 5:-**

**Design Thinking:**

**1. Data Collection:**

*- Objective:* The first step is to gather data containing customer reviews and sentiments about competitor products. This dataset can be collected from various sources, such as online review platforms, social media, or customer feedback forms.

*- User-Centered Approach:* In a Design Thinking context, it's essential to keep the end-users in mind. Consider what specific questions or challenges your users have that this data can address.

**2. Data Preprocessing:**

*- Cleaning:* Raw textual data often contains noise, such as irrelevant characters, punctuation, and special symbols. Data cleaning involves removing or replacing these elements to ensure consistency and accuracy.

*- Tokenization:* Break the text into individual words or tokens.

*- Stop Word Removal:* Eliminate common words (e.g., "the," "and") that don't carry significant meaning.

*- Stemming/Lemmatization:* Reduce words to their root form to consolidate variations of the same word.

**3. Sentiment Analysis Techniques:**

*- Bag of Words (BoW):* This technique represents text as a collection of words, ignoring grammar and word order. It's a basic but effective way to analyze sentiment based on word frequencies.

*- Word Embeddings:* Methods like Word2Vec, GloVe, or FastText represent words as dense vectors in a continuous space. This captures semantic relationships between words and can enhance sentiment analysis.

*- Transformer Models (e.g., BERT):* These advanced models can capture contextual information and are highly effective for sentiment analysis tasks, achieving state-of-the-art results.

**4. Feature Extraction:**

- After applying sentiment analysis techniques, you'll have features that represent sentiment scores or classifications (e.g., positive, negative, neutral) for each review.

- Additional features could include metadata like the date of the review, product attributes mentioned, and the user's profile information.

**5. Visualization:**

- Visualizations are powerful for conveying insights. Some common visualizations in sentiment analysis include:

*- Sentiment Distribution:* Histograms or pie charts showing the distribution of sentiments (e.g., percentage of positive, negative, and neutral reviews).

- *Time Series Analysis:* Line charts depicting sentiment trends over time.

*- Word Clouds*: Displaying frequently mentioned words in positive and negative reviews.

*- Heatmaps:* Showing correlations between sentiment and other variables like product attributes.

**6. Insights Generation:**

- With the visualized data, you can extract meaningful insights. Here's how Design Thinking principles can help:

*- Empathy:* Put yourself in the shoes of the customers who left the reviews. What are their pain points, desires, and motivations?

*- Define:* Clearly define the key insights and challenges derived from the sentiment analysis.

*- Ideate*: Brainstorm potential solutions or strategies to address the identified challenges.

*- Prototype:* Develop and test prototypes of these solutions.

*- Test:* Gather feedback from stakeholders and users to refine and validate the solutions.

**7. Business Impact:**

- The ultimate goal is to translate these insights into actionable strategies that can impact the business positively. This might involve refining products, improving customer service, or developing targeted marketing campaigns based on customer sentiments.

**Here's how sentiment analysis can be applied to marketing projects overall:**

**Customer Feedback Analysis:**

Sentiment analysis can be used to analyze customer reviews, comments, and feedback on social media, review platforms, or surveys to understand how customers feel about your products, services, or brand. Positive sentiment can be leveraged for testimonials and marketing materials, while negative sentiment can guide improvements.

**Social Media Monitoring:**

Monitoring social media conversations and sentiment around your brand, products, or industry can provide insights into what's being said and help you adjust your marketing strategies accordingly.

**Brand Reputation Management:**

Sentiment analysis can be used to track changes in your brand's reputation over time. If you notice a negative trend, you can take action to address any issues and improve your brand's image.

**Competitor Analysis:**

Analyzing sentiment related to your competitors can help you identify opportunities and areas where you can differentiate your brand or services.

**Content Creation and Personalization:**

Sentiment analysis can help in tailoring marketing content and messages to match the emotional tone of your audience, making your campaigns more relatable and engaging.

**Campaign Evaluation:**

After launching marketing campaigns, you can use sentiment analysis to assess their impact on the target audience. This can help you determine which aspects of your campaign were well-received and which need improvement.

**Product Development:**

Analyzing sentiment around products or features can provide valuable insights for product development. Understanding what customers like or dislike can guide your product roadmap.

**Crisis Management:**

Detecting negative sentiment early can help you respond swiftly to PR crises, minimizing potential damage to your brand's reputation.

**Influencer and Affiliate Marketing:**

Sentiment analysis can help you identify influencers whose values and sentiments align with your brand, ensuring a more authentic partnership.

**Market Research:**

Sentiment analysis can be part of market research to understand consumer preferences, pain points, and emerging trends.

A computer screen shot of a computer

Description automatically generated

# Design Framework:fj

A diagram of a model

Description automatically generated

**tweet\_id**

1. **Data Collection:**

The first step in our process was data collection. We used a dataset of tweets, which is a common source of data for sentiment analysis due to the short, concise nature of tweets.

**Dataset Link:** https://[www.kaggle.com/datasets/crowdflower/twitter-](http://www.kaggle.com/datasets/crowdflower/twitter-) airline-sentiment

A screenshot of a computer

Description automatically generated

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **tweet\_id** | **airline\_sen timent** | **airline\_sen timent\_co nfidence** | **negativere ason** | **negativere ason\_con fidence** | **airline** | **airline\_sen timent\_gol d** | **name** |
| **570306133** | neutral | 1,0 |  |  | Virgin |  | cairdin |
| **677760513** |  |  | America |  |
| **570301130** | positive | 0,3486 |  | 0,0 | Virgin |  | jnardino |
| **888122368** |  |  |  | America |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **tweet\_id** | **airline\_sen timent** | **airline\_sen timent\_co nfidence** | **negativere ason** | **negativere ason\_con fidence** | **airline** | **airline\_sen timent\_gol d** | **name** |
| **570301083** | neutral | 0,6837 |  |  | Virgin |  | yvonnalynn |
| **672813571** |  |  | America |  |
| **570301031** | negative | 1,0 | Bad Flight | 0,7033 | Virgin |  | jnardino |
| **407624196** |  |  |  |  | America |  |
| **570300817** | negative | 1,0 | Can't Tell | 1,0 | Virgin |  | jnardino |
| **074462722** |  |  |  |  | America |  |
| **570300767** | negative | 1,0 | Can't Tell | 0,6842 | Virgin |  | jnardino |
| **074181121** |  |  |  |  | America |  |
| **570300616** | positive | 0,6745 |  | 0,0 | Virgin |  | cjmcginnis |
| **901320704** |  |  |  | America |  |
| **570300248** | neutral | 0,634 |  |  | Virgin |  | pilot |
| **553349120** |  |  | America |  |
| **570299953** | positive | 0,6559 |  |  | Virgin |  | dhepburn |
| **286942721** |  |  | America |  |
| **570295459** | positive | 1,0 |  |  | Virgin |  | YupitsTate |
| **631263746** |  |  | America |  |
| **570294189** | neutral | 0,6769 |  | 0,0 | Virgin |  | idk\_but\_yo |
| **143031808** |  |  |  | America | utube |
| **570289724** | positive | 1,0 |  |  | Virgin |  | HyperCami |
| **453216256** |  |  | America | Lax |
| **570289584** | positive | 1,0 |  |  | Virgin |  | HyperCami |
| **061480960** |  |  | America | Lax |
| **570287408** | positive | 0,6451 |  |  | Virgin |  | mollanders |
| **438120448** |  |  | America | on |
| **570285904** | positive | 1,0 |  |  | Virgin |  | sjespers |
| **809598977** |  |  | America |  |
| **570282469** | negative | 0,6842 | Late Flight | 0,3684 | Virgin |  | smartwater |
| **121007616** |  |  |  |  | America | melon |
| **570277724** | positive | 1,0 |  |  | Virgin |  | ItzBrianHu |
| **385734656** |  |  | America | nty |
| **570276917** | negative | 1,0 | Bad Flight | 1,0 | Virgin |  | heatherovi |
| **301137409** |  |  |  |  | America | eda |
| **570270684** | positive | 1,0 |  |  | Virgin |  | thebrandira |
| **619923457** |  |  | America | y |

# Data Preprocessing:

After collecting the data, we performed several preprocessing steps to clean and prepare the data for analysis. These steps include

* **Lowercasing**: We converted all the text to lowercase to ensure that the same words in different cases are not considered as different words.
* **Removing Punctuation and Special Characters**: We removed all punctuation and special characters from the text as they do not contribute to sentiment.
* **Removing Stop Words**: We removed common words that do not carry much information (like "is", "the", "and", etc.). These words are called stop words.
* **Tokenization**: We broke down the text into individual words or tokens.
* **Lemmatization**: We reduced the words to their base or root form (e.g., "running" to "run"). This helps in reducing the dimensionality of the data and grouping similar sentiments together.

In[n]:

import pandas as pd import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.feature\_extraction.text import CountVectorizer from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import confusion\_matrix, classification\_report

import matplotlib.pyplot as plt import seaborn as sns

from nltk.corpus import stopwords

from nltk.stem import WordNetLemmatizer import re

import nltk nltk.download('stopwords') nltk.download('wordnet')

# Load the dataset

df = pd.read\_csv('/kaggle/input/twitter-airline-sentiment/ Tweets.csv')

# Display the first 5 rows of the dataframe df.head()

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **tweet\_i d** | **airline\_s entimen t** | **airline\_s entimen t\_confid ence** | **negativ ereason** | **negativ ereason**  **\_confid ence** | **airline** | **airline\_s entimen t\_gold** | **name** | **negativ ereason**  **\_gold** | **retweet**  **\_count** |
| **5703061** | neutral | 1,0 |  |  | Virgin |  | cairdin |  | 0 |
| **3367776** |  |  | America |  |  |
| **0513** |  |  |  |  |  |
| **5703011** | positive | 0,3486 |  | 0,0 | Virgin |  | jnardino |  | 0 |
| **3088812** |  |  |  | America |  |  |
| **2368** |  |  |  |  |  |  |
| **5703010** | neutral | 0,6837 |  |  | Virgin |  | yvonnaly |  | 0 |
| **8367281** |  |  | America | nn |  |
| **3571** |  |  |  |  |  |
| **5703010** | negative | 1,0 | Bad | 0,7033 | Virgin |  | jnardino |  | 0 |
| **3140762** |  |  | Flight |  | America |  |  |
| **4196** |  |  |  |  |  |  |  |

In[n]:

# Drop unnecessary columns

df = df[['airline\_sentiment', 'text']]

# Display the first 5 rows of the dataframe after dropping unnecessary columns

df.head()

Out[2]:

|  |  |  |
| --- | --- | --- |
|  | **airline\_sentiment** | **text** |
| **0** | neutral | @VirginAmerica What @dhepburn said. |
| **1** | positive | @VirginAmerica plus you've added commercials to the experience... tacky. |
| **2** | neutral | @VirginAmerica I didn't today... Must mean I need to take another trip! |
| **3** | negative | @VirginAmerica it's really aggressive to blast obnoxious "entertainment" in your guests' faces &amp; they have little recourse |

In[3]:

def preprocess\_text(text):

# Remove punctuations and numbers

text = re.sub('[^a-zA-Z]', ' ', text)

# Single character removal

text = re.sub(r'\s+[a-zA-Z]\s+', ' ', text)

# Removing multiple spaces

text = re.sub(r'\s+', ' ', text)

# Converting to Lowercase

text = text.lower()

# Lemmatization

#text = text.split()

#lemmatizer = WordNetLemmatizer()

#text = [lemmatizer.lemmatize(word) for word in text if not word in set(stopwords.words('english'))]

#text = ' '.join(text)

return text

# Apply the preprocessing to the 'text' column df['text'] = df['text'].apply(preprocess\_text)

# Display the first 5 rows of the dataframe after preprocessing df.head()

Out[3]:

|  |  |  |
| --- | --- | --- |
|  | **airline\_sentiment** | **text** |
| **0** | neutral | @VirginAmerica What @dhepburn said. |
| **1** | positive | @VirginAmerica plus you've added commercials to the experience... tacky. |
| **2** | neutral | @VirginAmerica I didn't today... Must mean I need to take another trip! |
| **3** | negative | @VirginAmerica it's really aggressive to blast obnoxious "entertainment" in your guests' faces &amp; they have little recourse |

# Feature Extraction:

After preprocessing, we converted the text data into numerical features that can be used by a machine learning algorithm. We used the TF-IDF (Term Frequency-Inverse Document Frequency) method for this. TF-IDF gives a weight to each word signifying its importance in the document and across a corpus of documents.

# Model Training:

We used a Random Forest Classifier for sentiment analysis. Random Forest is a versatile and widely used algorithm that works well for many tasks. It creates a set of decision trees from a randomly selected subset of the training set, which then aggregates votes from different decision trees to decide the final class of the test object.

# Model Evaluation:

After training the model, we evaluated its performance using a confusion matrix and calculated metrics such as accuracy, precision, recall, and F1- score. These metrics give us a quantitative measure of the model's performance.

# Insights & Interpretation:

Finally, we interpreted the results of the sentiment analysis. This involves understanding the performance of the model, identifying any areas of improvement, and drawing insights from the model's predictions.

In[1]:

# Splitting the data into training and testing sets from sklearn.model\_selection import train\_test\_split X\_train, X\_test, y\_train, y\_test = train\_test\_split(df['text'], df['airline\_sentiment'], test\_size=0.2, random\_state=42)

# Feature Extraction

from sklearn.feature\_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(max\_features=2500, min\_df=7, max\_df=0.8)

X\_train = vectorizer.fit\_transform(X\_train).toarray() X\_test = vectorizer.transform(X\_test).toarray()

# Model Training

from sklearn.ensemble import RandomForestClassifier classifier = RandomForestClassifier(n\_estimators=1000, random\_state=0)

classifier.fit(X\_train, y\_train)

A close up of a logo

Description automatically generated

Out[1]:

In[2]:

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score

def evaluate\_model(y\_test, y\_pred):

print('Classification Report:')

print(classification\_report(y\_test, y\_pred))

print('Confusion Matrix:')

print(confusion\_matrix(y\_test, y\_pred))

print('Accuracy Score:')

print(accuracy\_score(y\_test, y\_pred))

y\_pred = classifier.predict(X\_test) evaluate\_model(y\_test, y\_pred)

out[2]:

Classification Report:

|  |  |  |  |
| --- | --- | --- | --- |
| precision | recall | f1-score | support |
| negative 0.79 | 0.95 | 0.86 | 1889 |
| neutral 0.65 | 0.41 | 0.50 | 580 |
| positive 0.80 | 0.50 | 0.62 | 459 |
| accuracy |  | 0.77 | 2928 |
| macro avg 0.75 | 0.62 | 0.66 | 2928 |
| weighted avg 0.76 | 0.77 | 0.75 | 2928 |
| Confusion Matrix: |  |  |  |

|  |  |  |
| --- | --- | --- |
| [[1799 | 65 | 25] |
| [ 312 | 235 | 33] |
| [ 169 | 60 | 230]] |

Accuracy Score: 0.773224043715847

In[3]:

import matplotlib.pyplot as plt

import seaborn as sns

def plot\_confusion\_matrix(y\_test, y\_pred):

cm = confusion\_matrix(y\_test, y\_pred)

df\_cm = pd.DataFrame(cm, index = [i for i in ['negative', 'neutral', 'positive']],

columns = [i for i in ['negative', 'neutral', 'positive']])

plt.figure(figsize = (10,7))

sns.heatmap(df\_cm, annot=True, fmt='d', cmap='Blues')

plt.title('Confusion Matrix')

plt.xlabel('Predicted')

plt.ylabel('True')

plt.show()

plot\_confusion\_matrix(y\_test, y\_pred) Out[4]:

A screenshot of a graph

Description automatically generated

In[5]:

import seaborn as sns

import matplotlib.pyplot as plt

# Creating column 'tweet\_length' df['tweet\_length'] = df['text'].apply(len)

# distribution of sentiments plt.figure(figsize=(8,6)) sns.countplot(x='airline\_sentiment', data=df) plt.title('Distribution of Sentiments') plt.show()

# Histogram of tweet lengths plt.figure(figsize=(8,6)) sns.histplot(df['tweet\_length'], bins=30) plt.title('Distribution of Tweet Lengths') plt.show()

# Boxplot of tweet lengths plt.figure(figsize=(8,6))

sns.boxplot(x='airline\_sentiment', y='tweet\_length', data=df)

plt.title('Distribution of Tweet Lengths by Sentiment') plt.show()

out[5]:

A bar graph with different colored squares

Description automatically generated

A graph of a distribution of tweets

Description automatically generated

A diagram of a distribution of tweet lengths

Description automatically generated

# Critical Analysis

The following conclusions may be drawn from the visuals and model evaluation:

# Sentiment Distribution:

The dataset's bar plot of sentiment distribution reveals that the bulk of tweets are unfavorable in nature, with neutral and supportive tweets coming in second and third. Due to the dataset's imbalance, the model may be more likely to correctly predict negative feelings than neutral or positive feelings.

**Model Execution:**

The Random Forest classifier's total accuracy was around 76%. The neutral and positive classes' accuracy, recall, and F1-score, however, are lower than those of the negative class. This implies that the model performs better at detecting negative than neutral or positive attitudes, which may be related to the dataset's imbalance.

**Confusion Matrix:**

The confusion matrix reveals that for the neutral and positive classes, the model has a disproportionately large number of false positives and false negatives. This further demonstrates the model's bias towards predicting negative feelings since it frequently misclassifies neutral and positive tweets as negative.

**Data Distribution:**

Looking at the histogram, it's obvious, as mentioned before, that there is a significant imbalance in the data in favor of negative sentiment. This is likely because people with negative sentiments are more motivated to tweet. By examining the length distribution in the box plot and the bar chart, we can conclude that the majority of tweets are between 60 to 100 characters long. Negative tweets are usually longer, also falling within the 60 to 100 character range, which further confirms the data imbalance.

In conclusion, the model fails to predict neutral and positive attitudes even if it does a fair job of predicting negative sentiments. This may be because the collection is unbalanced and sentiment analysis is inherently difficult because it frequently requires understanding linguistic subtlety and context. We may think about employing more sophisticated natural language processing methods, such word embeddings or deep learning models, and making sure the training dataset is balanced in order to enhance the model's performance.

**CONCLUSION:-**

In conclusion, the sentiment analysis project using AI has demonstrated its immense potential in extracting valuable insights from text data, offering a range of practical applications and benefits. This project has showcased the power of artificial intelligence and natural language processing in understanding and quantifying human emotions and opinion.The sentiment analysis project using AI has proven to be a powerful tool for understanding and leveraging the sentiment hidden within text data. Its applications extend across a wide range of industries and use cases, offering significant potential for enhancing decision-making, customer satisfaction, and overall business performance. As AI technology continues to advance, sentiment analysis will undoubtedly play an increasingly crucial role in shaping how businesses and organizations interact with their customers and adapt to a rapidly evolving digital landscape.