

Customer Support On Twitter

Data Science With Python Lab Project Report

Bachelor
in
Computer Science

By

Team Name

S190103

S190672



Rajiv Gandhi University Of Knowledge And Technologies

S.M. Puram , Srikakulam -532410

Andhra Pradesh, India

Abstract

Customer support on Twitter refers to the practice of providing assistance and resolving customer inquiries, issues, or complaints through the social media platform Twitter. Social media platforms like Twitter can be leveraged efficiently to promote the business as well as to address customer complaints. Twitter provides a platform for customers to reach out to companies or brands directly, allowing them to ask questions, seek help, or voice their concerns publicly. Customer support on Twitter typically involves addressing a wide range of customer needs, such as product inquiries, technical support, billing or refund requests, order tracking, and general customer service inquiries.

Efficient customer service leads to enhanced customer satisfaction which in turn can generate higher revenue for the organisation. This study investigates how customers respond to companies' products on Twitter through NLP (Natural Language Processing) techniques such as Sentiment Analysis and Topic Modeling. The objective of Sentiment Analysis is to predict the behaviour of users to discover trend/pattern and to track public opinion. We will perform several forms of pre-processing and filter out the conversations on customer support in Twitter. The dataset for this project was taken from Kaggle and included around 3 million customer support tweets between customers and several different companies. We will use Tools like VADER for sentiment analysis, SciKit-Learn for vectorizing and topic modeling, DBSCAN for clustering. Throughout this project, we will explore various strategies, tools, and best practices to enhance customer support on Twitter.

Contents

Abstract	1
1 Introduction	3
1.1 Introduction To Project	3
1.2 Application	4
1.3 Motivation Towards Project	5
1.4 Problem Statement	6
2 Approach To Your Project	7
2.1 Explain About Your Project	7
2.2 Data Set	9
2.3 Prediction technique	11
2.4 Graphs	12
2.5 Visualization	15
3 Code	22
4 Conclusion and Future Work	29

Chapter 1

Introduction

1.1 Introduction To Project

In today's digital landscape, social media platforms have become a crucial channel for communication and engagement. Among these platforms, Twitter has emerged as a powerful tool for individuals, businesses, and organizations to connect, share information, and build relationships. Recognizing its potential, many companies have integrated Twitter into their customer support strategies, leveraging its real-time nature and wide reach to provide efficient and responsive assistance to their customers.

The "Customer Support on Twitter" project aims to explore and optimize the customer support experience on this popular social media platform. With the advent of technology and changing consumer expectations, businesses must adapt to meet the evolving needs of their customers. This project recognizes that providing effective customer support on Twitter is a vital component of maintaining customer satisfaction, loyalty, and overall brand reputation.

Through this project, we aim to empower businesses to optimize their customer support on Twitter, transforming it into a seamless and efficient process. By harnessing the power of technology, data-driven insights, and best practices, companies can cultivate stronger relationships with their customers, foster loyalty, and ultimately drive business growth.

Twitter's unique characteristics, such as its concise format and public nature, make it an ideal platform for customers to seek assistance, voice concerns, and provide feedback

directly to companies. Recognizing this potential, organizations have begun to leverage Twitter as a crucial customer support channel, enabling them to respond quickly and efficiently to customer inquiries and issues.

This project focuses on enhancing customer support on Twitter, aiming to provide an exceptional user experience and improve overall customer satisfaction. By streamlining and optimizing customer support interactions on Twitter, businesses can strengthen their relationships with customers, enhance brand reputation, and foster loyalty.

1.2 Application

The "customer support on Twitter" project can have several applications and benefits for businesses.

Here are some potential applications:

Real-time issue resolution: Twitter provides a platform for customers to reach out to businesses directly with their concerns or issues. By providing customer support on Twitter, businesses can address these concerns in real-time, offering prompt solutions and improving customer satisfaction.

Brand reputation management: Twitter is a popular platform for customers to share their experiences, both positive and negative. By actively engaging with customers on Twitter and providing excellent customer support, businesses can manage their brand reputation effectively and turn negative experiences into positive ones.

Proactive customer engagement: Twitter allows businesses to proactively engage with customers by monitoring relevant conversations and identifying opportunities to assist or provide information. By monitoring brand mentions and relevant hashtags, businesses can reach out to customers, offer support, and create a positive impression.

Social media marketing and promotion: Incorporating customer support on Twitter can also serve as a marketing and promotional tool. By engaging with customers publicly, businesses can showcase their excellent customer service, demonstrate their commitment to customer satisfaction, and attract potential customers who value good support.

Gathering customer feedback: Twitter provides a valuable platform for businesses to gather feedback from customers. By actively participating in conversations, responding to

feedback, and encouraging customers to share their thoughts, businesses can gain insights into customer preferences, pain points, and areas for improvement.

Crisis management: During times of crisis or emergencies, Twitter becomes a crucial communication channel. By providing customer support on Twitter, businesses can effectively communicate important updates, address concerns, and provide assistance to customers during challenging situations.

Data analysis and insights: Twitter offers various analytics tools and data tracking features that can provide valuable insights into customer behavior, sentiment, and trends. By analyzing this data, businesses can identify patterns, adjust their customer support strategies, and make informed decisions to enhance their overall customer experience..

1.3 Motivation Towards Project

Prompt Response: When a customer reaches out with a problem or question on Twitter, it's crucial to respond promptly. Aim to acknowledge their inquiry within a short period, ideally within minutes or hours, to show that their issue is being taken seriously.

Active Listening: Carefully read and understand the customer's message to grasp the specific problem or concern they are facing. This step ensures that the response is tailored to address their needs accurately.

Empathy and Understanding: Show empathy and understanding towards the customer's situation. Let them know that you acknowledge their frustration or inconvenience and assure them that you're committed to finding a resolution.

Clear Communication: Provide concise and clear responses to the customer's queries or concerns. Use simple language to avoid confusion and ensure that the customer understands the steps or information you are providing.

Problem Resolution: Work towards finding a solution to the customer's problem. If possible, offer step-by-step guidance or instructions to help them resolve the issue independently. If the problem requires further assistance, offer to escalate the matter to the appropriate department or provide alternate contact channels for more in-depth support.

Transparency and Updates: Keep the customer informed throughout the process. If additional time is needed to investigate or resolve their issue, communicate this to them

and provide regular updates on the progress. Transparency builds trust and demonstrates your commitment to resolving their problem.

Personalization: Treat each customer as an individual. Use their name and refer to their specific issue to create a personalized and attentive experience. This approach shows that you value their concerns and are dedicated to resolving them.

Follow-Up: After the initial interaction, consider following up with the customer to ensure their problem has been fully resolved and to inquire about their satisfaction with the support provided. This gesture demonstrates a commitment to excellent customer service and helps build long-term relationships.

1.4 Problem Statement

Develop a predictive model to identify the best customer-friendly company which provides best service and quick response for customer queries/complaints, it should improve overall customer satisfaction and enhance a company's business through social media platform Twitter. The goal is to address customer inquiries, complaints, and feedback promptly and professionally, ensuring a positive customer experience and strengthening the brand's reputation.

Chapter 2

Approach To Your Project

2.1 Explain About Your Project

Aim:

The goal of our project "Customer Support on Twitter" is to address customer inquiries, complaints, and feedback promptly and professionally, ensuring a positive customer experience and strengthening the brand's reputation.

Overview:

In the age of social media, providing excellent customer support on platforms like Twitter has become an essential aspect of maintaining a strong brand image and fostering positive customer relationships. This project aims to develop and implement an effective customer support strategy on Twitter, enabling businesses to engage with their customers, address their concerns, and provide timely assistance. This introduction will provide an overview of the project, its objectives, and the key components involved.

Objectives:

The primary objectives of the Customer Support on Twitter project include:

- a. Enhancing customer satisfaction: By promptly responding to customer inquiries, complaints, and feedback on Twitter, the project aims to improve overall customer satis-

faction levels.

b. Building brand loyalty: Through personalized interactions and problem resolution, the project aims to foster a sense of loyalty among customers, enhancing their trust and commitment to the brand.

c. Increasing customer engagement: By actively engaging with customers on Twitter, the project aims to create a vibrant community, encouraging discussions, and boosting customer engagement levels.

d. Streamlining support processes: The project aims to implement efficient workflows, tools, and resources to ensure smooth handling of customer support requests and enhance team productivity.

Key Components:

To achieve the project objectives, the following key components will be addressed: a. Team setup and training: Assembling a dedicated customer support team and providing them with the necessary training on effective communication, problem-solving, and utilizing Twitter as a support platform.

b. Establishing response guidelines: Defining response time targets, escalation procedures, and brand tone of voice to ensure consistent and effective communication across all customer interactions.

c. Monitoring and listening: Implementing social media monitoring tools to track brand mentions, keywords, and customer sentiment to proactively identify support opportunities and emerging issues.

d. Rapid response management: Developing a system for prioritizing and categorizing customer support requests, enabling the team to respond quickly and efficiently.

e. Knowledge base development: Creating a comprehensive knowledge base with FAQs, troubleshooting guides, and relevant resources to empower both customers and support agents with accurate information.

f. Collaboration with other departments: Establishing effective collaboration channels with other departments, such as product development and marketing, to facilitate efficient issue resolution and gather customer feedback.

g. Performance measurement: Implementing metrics and analytics tools to track key

performance indicators (KPIs) such as response time, customer satisfaction ratings, and resolution rates, enabling continuous improvement.

Benefits: Implementing an effective customer support strategy on Twitter can yield several benefits, including:

- a. Enhanced customer experience: By offering quick and personalized assistance, businesses can create positive experiences that contribute to customer loyalty and advocacy.

- b. Brand visibility and reputation management: Resolving customer issues publicly on Twitter showcases a brand's commitment to customer satisfaction, improving its reputation and attracting potential customers.

- c. Valuable customer insights: By actively engaging with customers on Twitter, businesses can gain valuable insights into their preferences, pain points, and expectations, enabling them to refine products and services.

- d. Cost-effective support channel: Twitter provides a cost-effective customer support channel compared to traditional methods, as it allows support teams to handle multiple queries simultaneously.

Conclusion: The Customer Support on Twitter project aims to establish a robust customer support strategy on the platform, benefiting both businesses and customers. By leveraging the power of Twitter, businesses can enhance customer satisfaction, build brand loyalty, and streamline support processes. With the right team, tools, and guidelines in place, this project can lead to improved customer experiences and stronger customer relationships.

2.2 Data Set

source:<https://www.kaggle.com/datasets/thoughtvector/customer-support-on-twitter>

The Customer Support on Twitter dataset is a large, modern corpus of tweets and replies to aid innovation in natural language understanding and conversational models, and for study of modern customer support practices and impact.

Context Natural language remains the densest encoding of human experience we have, and innovation in NLP has accelerated to power understanding of that data, but the datasets driving this innovation don't match the real language in use today. The Customer

Support on Twitter dataset offers a large corpus of modern English (mostly) conversations between consumers and customer support agents on Twitter, and has three important advantages over other conversational text datasets:

Focused - Consumers contact customer support to have a specific problem solved, and the manifold of problems to be discussed is relatively small, especially compared to unconstrained conversational datasets like the reddit Corpus. **Natural** - Consumers in this dataset come from a much broader segment than those in the Ubuntu Dialogue Corpus and have much more natural and recent use of typed text than the Cornell Movie Dialogs Corpus. **Succinct** - Twitter’s brevity causes more natural responses from support agents (rather than scripted), and to-the-point descriptions of problems and solutions. Also, its convenient in allowing for a relatively low message limit size for recurrent nets. **Inspiration** The size and breadth of this dataset inspires many interesting questions:

Can we predict company responses? Given the bounded set of subjects handled by each company, the answer seems like yes! Do requests get stale? How quickly do the best companies respond, compared to the worst? Can we learn high quality dense embeddings or representations of similarity for topical clustering? How does tone affect the customer support conversation? Does saying sorry help? Can we help companies identify new problems, or ones most affecting their customers? Acknowledgements Dataset built with PointScape.

Content The dataset is a CSV, where each row is a tweet. The different columns are described below. Every conversation included has at least one request from a consumer and at least one response from a company. Which user IDs are company user IDs can be calculated using the inbound field.

tweet id:

A unique, anonymized ID for the Tweet. Referenced by response tweet id and in response to tweet id.

author id:

A unique, anonymized user ID. @s in the dataset have been replaced with their associated anonymized user ID.

inbound:

Whether the tweet is "inbound" to a company doing customer support on Twitter. This feature is useful when re-organizing data for training conversational models.

created at:

Date and time when the tweet was sent.

text:

Tweet content. Sensitive information like phone numbers and email addresses are replaced with mask values like email .

response tweet id:

IDs of tweets that are responses to this tweet, comma-separated.

in response to tweet id:

ID of the tweet this tweet is in response to, if any.

2.3 Prediction technique

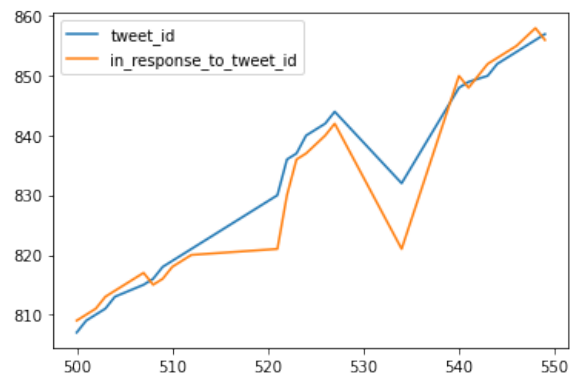
Sentiment Analysis: Sentiment analysis, also known as opinion mining, is a technique used to determine the sentiment expressed in a piece of text. In the context of customer support on Twitter, sentiment analysis can be applied to analyze the sentiment of customer interactions, such as tweets or direct messages, to understand the overall sentiment and emotional tone of the messages.

Sentiment analysis involves classifying customer tweets or messages into positive, negative, or neutral sentiments. Natural Language Processing (NLP) techniques, such as machine learning algorithms or rule-based approaches, can be employed to analyze the sentiment expressed in customer interactions. This prediction technique helps identify customers' emotional states, enabling support teams to prioritize and address negative sentiments more effectively.

Various techniques can be employed for sentiment analysis, including rule-based methods, machine learning algorithms (such as Naive Bayes, Support Vector Machines, or deep learning models like recurrent neural networks), and hybrid approaches that combine multiple techniques.

2.4 Graphs

```
1 import numpy as np
2 import pandas as pd
3 import seaborn as sns
4 import matplotlib.pyplot as plt
5 df=pd.read_csv('/home/rgukt/Downloads/twitter/twcs/twcs.csv')
6 df1=df.head(50)
7 a=df1["author_id"]
8 id_author,frequency=np.unique(a,return_counts=True)
9 x=np.asarray(id_author)
10 y=np.asarray(frequency)
11 df1=df[500:550]
12 df1=df1.dropna()
13 df1.plot()
14 plt.legend()
15 plt.show()
```

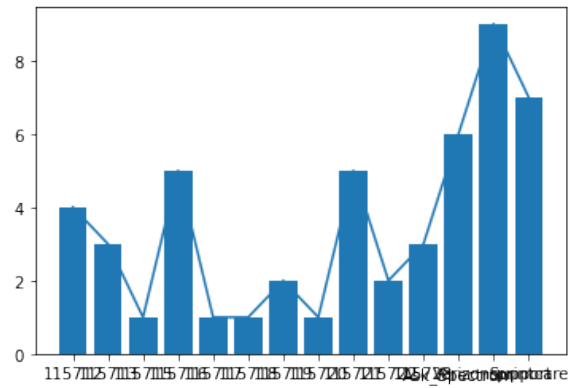


```
1 plt.plot(x,y)
2 plt.annotate('created_at',xy=(10.6,121873.0),
```

```

3 arrowprops=dict(facecolor='black',shrink=0.005),
4 xytext=(10.4,122392.0))
5 plt.bar(x,y)
6 plt.axis("tight")

```



```

1 plt.xticks(rotation=90)
2 plt.title("frequency of tweets from different author_id's")
3 plt.xlabel('author id')
4 plt.ylabel('frequency')
5 plt.scatter(x,y)

```

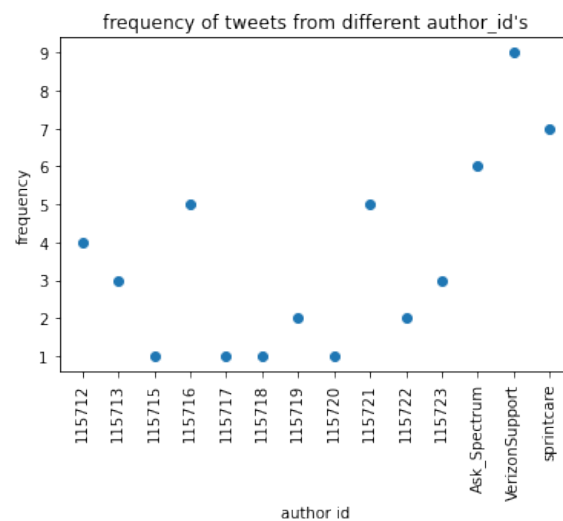


Figure 2.1: frequency of tweets from different author id

```

1 colors=sns.color_palette("husl",10)
2 pd.Series(df['author_id'])

```

```

3 [500:550]).value_counts().plot(kind="bar",color=colors,figsize=
4 (8,6),fontsize=10,rot=0,title="Total No.Of Tweets for Companies")
5 plt.xlabel('Tweets',fontsize=10)
6 plt.ylabel('No. of Tweets',fontsize=10)
7 plt.show()

```

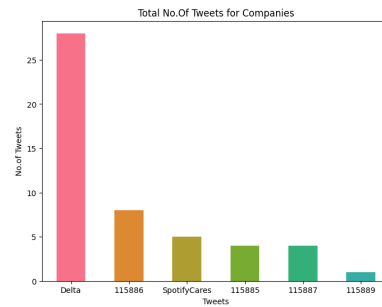
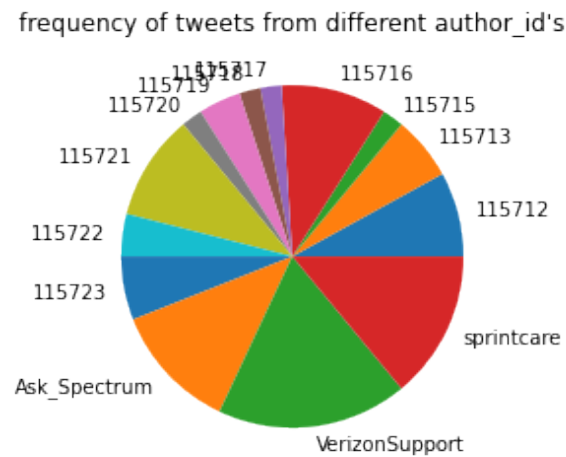


Figure 2.2: Total No.Of Tweets for Companies

```

1 plt.pie(y,labels=x)
2 plt.title("frequency of tweets from different author_id's")
3 plt.show()

```



```

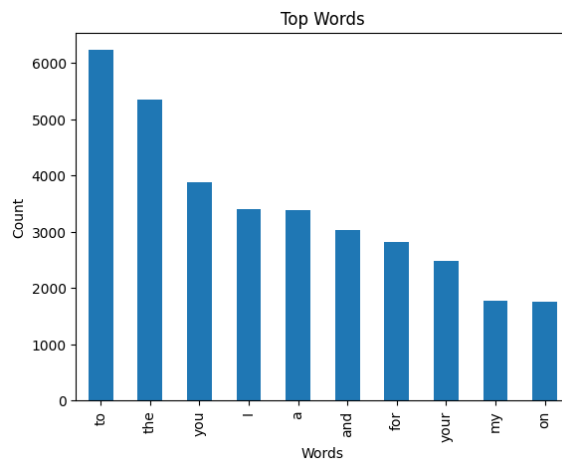
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 # Tokenize the text data
5 df['tokens'] = df['text'].apply(lambda x: x.split())

```

```

6
7 # Flatten the token list
8 tokens_flat = [token for sublist in df['tokens'] for token in sublist]
9
10 # Count the occurrences of each word
11 word_counts = pd.Series(tokens_flat).value_counts()
12
13 # Plot a bar plot of top words
14 top_words = word_counts.head(10)
15 top_words.plot(kind='bar')
16 plt.xlabel('Words')
17 plt.ylabel('Count')
18 plt.title('Top Words')
19 plt.show()

```



2.5 Visualization

```

1 plt.plot(x,y)
2 plt.xlabel('author id')
3 plt.ylabel('frequency')
4 plt.xticks(rotation=90)
5 plt.title("frequency of tweets from different author_id")

```



```
6 plt.show()
```

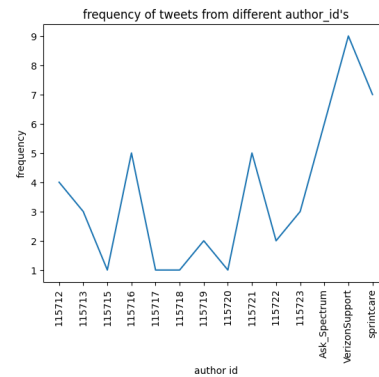
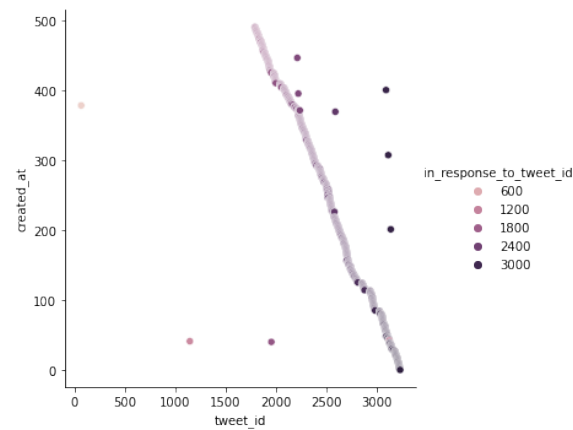


Figure 2.3: frequency of tweets from different author ids

```
1 sns.relplot(x="tweet_id",y="created_at",data=k, hue='in_response_to_tweet_id',
2             )
plt.show()
```

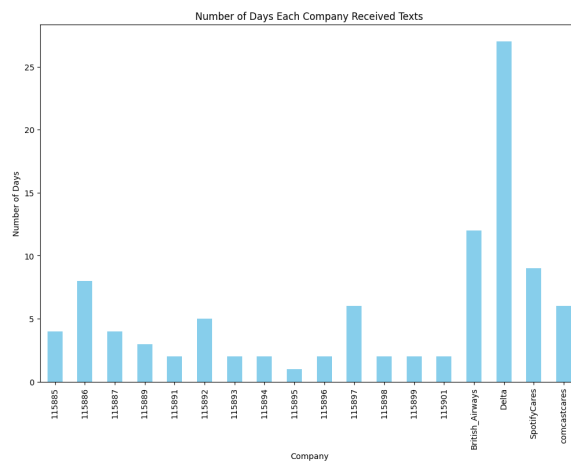


```
1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 data = pd.DataFrame(d)
5 data=data[500:600]
6 # Convert 'created_at' column to datetime format
7 data['created_at'] = pd.to_datetime(data['created_at'])
8
9 # Group by 'author_id' and calculate the number of unique days
```

```

10 company_days = data.groupby('author_id')['created_at'].nunique()
11
12 # Plot the graph
13 plt.figure(figsize=(12, 8))
14 company_days.plot(kind='bar', color='skyblue')
15 plt.title('Number of Days Each Company Received Texts')
16 plt.xlabel('Company')
17 plt.ylabel('Number of Days')
18 plt.xticks(rotation=90)
19 plt.show()

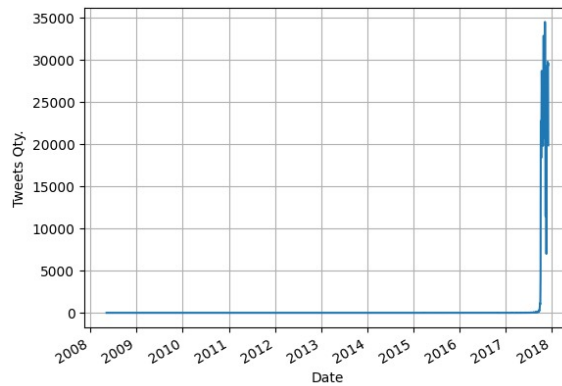
```



```

1 count_by_date = inbound_tweets.groupby(['timestamp']) \
2     .count()[['tweet_id']] \
3     .sort_values(['timestamp'], ascending = True) \
4     .rename({'tweet_id': 'qty'}, axis='columns')
5 display(count_by_date.head())
6 plt.figure()
7 plt.plot(count_by_date.index.values, count_by_date.qty)
8 plt.xlabel('Date')
9 plt.ylabel('Tweets Qty.')
10 plt.grid(True)
11 plt.gcf().autofmt_xdate()
12 plt.show()

```



```

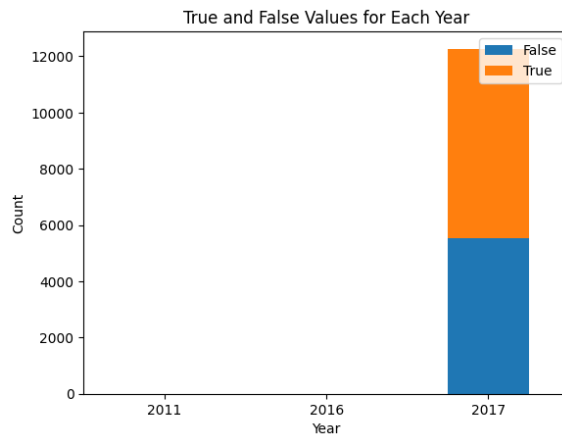
1 .import pandas as pd
2 import matplotlib.pyplot as plt
3 df = pd.DataFrame(d)
4 # Convert 'created_at' column to datetime format
5 df['created_at'] = pd.to_datetime(df['created_at'])
6
7 # Extract the year from 'created_at'
8 df['year'] = df['created_at'].dt.year
9
10 # Group by 'year' and 'inbound', and count the occurrences
11 year_counts = df.groupby(['year', 'inbound']).size().unstack()
12
13 # Plot the graph
14 plt.figure(figsize=(12, 8))
15 year_counts.plot(kind='bar', stacked=True)
16 plt.title('True and False Values for Each Year')
17 plt.xlabel('Year')
18 plt.ylabel('Count')
19 plt.xticks(rotation=0)
20 plt.legend(['False', 'True'], loc='upper right')
21 plt.show()

```

```

1 neg_tweets = data.groupby(['author_id', 'inbound']).count().iloc[:, 0]
2 total_tweets = data.groupby(['author_id'])['inbound'].count()
3

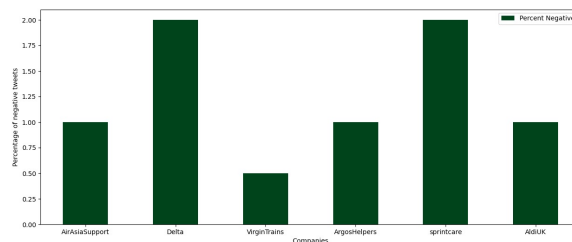
```



```

4 my_dict = {'AirAsiaSupport': neg_tweets[0] / total_tweets[0],
5 'Delta': neg_tweets[3] / total_tweets[1], 'VirginTrains':
6 neg_tweets[6] / total_tweets[2], 'ArgosHelpers':
7 neg_tweets[9] / total_tweets[3], 'sprintcare': neg_tweets[12] /
8 total_tweets[4], 'AldiUK': neg_tweets[15] / total_tweets[5]}
9 perc = pd.DataFrame.from_dict(my_dict, orient = 'index')
10 perc.columns = ['Percent Negative']
11 print(perc)
12 ax = perc.plot(kind = 'bar', rot=0, colormap = 'Greens_r', figsize = (15,6))
13 ax.set_xlabel('Companies')
14 ax.set_ylabel('Percentage of negative tweets')
15 plt.show()

```



```

1 from wordcloud import WordCloud
2
3 # Concatenate all clean_text
4 text = ' '.join(df['clean_text'])
5
6 # Generate a word cloud

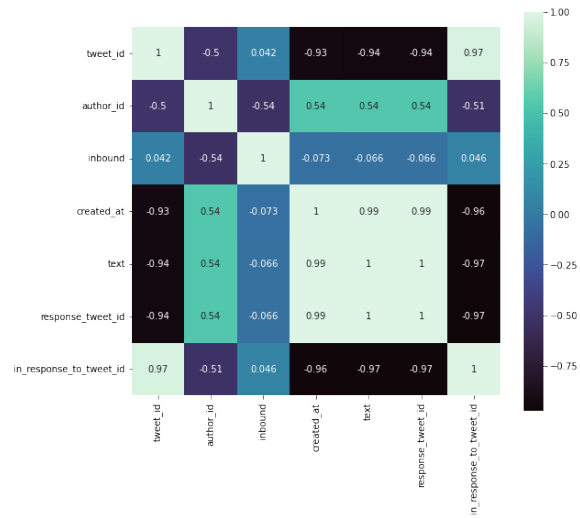
```



```

1 plt.figure(figsize=(9,8))
2 sns.heatmap(k.corr(),square=True,annot=True,cmap='mako',center=0)

```



Chapter 3

Code

DATASET

```
1 import pandas as pd
2 df= pd.read_csv('twcs.csv')
3 df
```

	tweet_id	author_id	inbound	created_at	text	response_tweet_id	in_response_to_tweet_id
0	1	sprintcare	False	Tue Oct 31 22:10:47 +0000 2017	@115712 I understand. I would like to assist y...	2	3.0
1	2	115712	True	Tue Oct 31 22:11:45 +0000 2017	@sprintcare and how do you propose we do that	NaN	1.0
2	3	115712	True	Tue Oct 31 22:08:27 +0000 2017	@sprintcare I have sent several private messag...	1	4.0
3	4	sprintcare	False	Tue Oct 31 21:54:49 +0000 2017	@115712 Please send us a Private Message so th...	3	5.0
4	5	115712	True	Tue Oct 31 21:49:35 +0000 2017	@sprintcare I did.	4	6.0
...
12270	16323	119590	True	Wed Nov 01 00:11:52 +0000 2017	@115765 @ATVIAssist please read it's not worki...	16320	NaN
12271	16326	ATVIAssist	False	Wed Nov 01 03:20:00 +0000 2017	@119591 Good evening, I'm not seeing much conn...	16327,16328	16329.0
12272	16327	119591	True	Wed Nov 01 07:29:34 +0000 2017	@ATVIAssist I have a PS4 and zombies plays fin...	NaN	16326.0
12273	16328	119591	True	Wed Nov 01 07:28:52 +0000 2017	@ATVIAssist Wired and I've got fiber Internet ...	NaN	16326.0
12274	16329	119591	True	Wed Nov 01 00:02:46 +0000 2017	@ATVIAssist Black ops 3 on Xbox one for me run...	NaN	NaN

12275 rows x 7 columns

Figure 3.1: Dataset of Customer Support of Twitter

Data Processing

```
1 import numpy as np
2 import re
3 import nltk
4 import matplotlib.pyplot as plt
5 from nltk.corpus import stopwords
6 from nltk.stem import PorterStemmer
```

```

7  nltk.download('punkt')
8  nltk.download('stopwords')
9
10 # Load the dataset
11 df = pd.read_csv('/content/sample_data/twcs.csv')
12
13 # Drop irrelevant columns
14 df = df.drop(['tweet_id', 'author_id', 'created_at', 'response_tweet_id', '
    in_response_to_tweet_id'], axis=1)
15
16 # Handle missing values
17 df = df.dropna()
18
19 # Text preprocessing
20 def clean_text(text):
21     # Remove special characters and links
22     text = re.sub(r'\W+', ' ', text)
23     text = re.sub(r'http\S+', '', text)
24
25     # Convert to lowercase
26     text = text.lower()
27
28     # Remove stopwords
29     stop_words = set(stopwords.words('english'))
30     tokens = nltk.word_tokenize(text)
31     filtered_tokens = [word for word in tokens if word not in stop_words]
32
33     # Stemming
34     stemmer = PorterStemmer()
35     stemmed_tokens = [stemmer.stem(word) for word in filtered_tokens]
36
37     return ' '.join(stemmed_tokens)
38

```



```

39 df['clean_text'] = df['text'].apply(clean_text)
40
41 # Split the dataset into train and test sets
42 from sklearn.model_selection import train_test_split
43 from sklearn.feature_extraction.text import TfidfVectorizer
44 from sklearn.svm import LinearSVC
45 from sklearn.ensemble import RandomForestClassifier
46 from sklearn.metrics import accuracy_score
47 X = df['clean_text']
48 y = df['inbound']
49 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
50                                                    random_state=42)
51 # Feature extraction using TF-IDF vectorization
52 vectorizer = TfidfVectorizer()
53 X_train_vec = vectorizer.fit_transform(X_train)
54 X_test_vec = vectorizer.transform(X_test)
55 df.head()

```

	inbound	text	clean_text
0	False	@115712 I understand. I would like to assist y...	115712 understand would like assist would need...
1	True	@sprintcare and how do you propose we do that	sprintcar propos
2	True	@sprintcare I have sent several private messag...	sprintcar sent sever privat messag one respond...
3	False	@115712 Please send us a Private Message so th...	115712 pleas send us privat messag assist clic...
4	True	@sprintcare I did.	sprintcar

Support Vector Machines (SVM)

```

1 # Train a LinearSVC classifier
2 classifier = LinearSVC()
3 classifier.fit(X_train_vec, y_train)
4 # Make predictions on the test set
5 y_pred = classifier.predict(X_test_vec)
6 # Evaluate the accuracy of the model

```

```
7 accuracy = accuracy_score(y_test , y_pred)
8 print("Accuracy:", accuracy)
```

```
1 Accuracy: 0.974745417515275
```

Logistic Regression

```
1 # Train a Logistic Regression classifier
2 classifier = LogisticRegression()
3 classifier.fit(X_train_vec , y_train)
4 # Make predictions on the test set
5 y_pred = classifier.predict(X_test_vec)
6 # Evaluate the accuracy of the model
7 accuracy = accuracy_score(y_test , y_pred)
8 print("Accuracy:", accuracy)
```

```
1 Accuracy: 0.9441955193482688
```

Random Forest

```
1 # Train a Random Forest classifier
2 classifier = RandomForestClassifier()
3 classifier.fit(X_train_vec , y_train)
4 # Make predictions on the test set
5 y_pred = classifier.predict(X_test_vec)
6 # Evaluate the accuracy of the model
7 accuracy = accuracy_score(y_test , y_pred)
8 print("Accuracy:", accuracy)
```

```
1 Accuracy: 0.9279022403258655
```

KNN

```
1 from sklearn.neighbors import KNeighborsClassifier
2 # Train a k-Nearest Neighbors classifier
3 k = 5 # Specify the desired number of neighbors
4 classifier = KNeighborsClassifier(n_neighbors=k)
```

```

5 classifier.fit(X_train_vec , y_train)
6 # Make predictions on the test set
7 y_pred = classifier.predict(X_test_vec)
8 # Evaluate the accuracy of the model
9 accuracy = accuracy_score(y_test , y_pred)
10 print("Accuracy:" , accuracy)

```

```

1 Accuracy: 0.9010183299389002

```

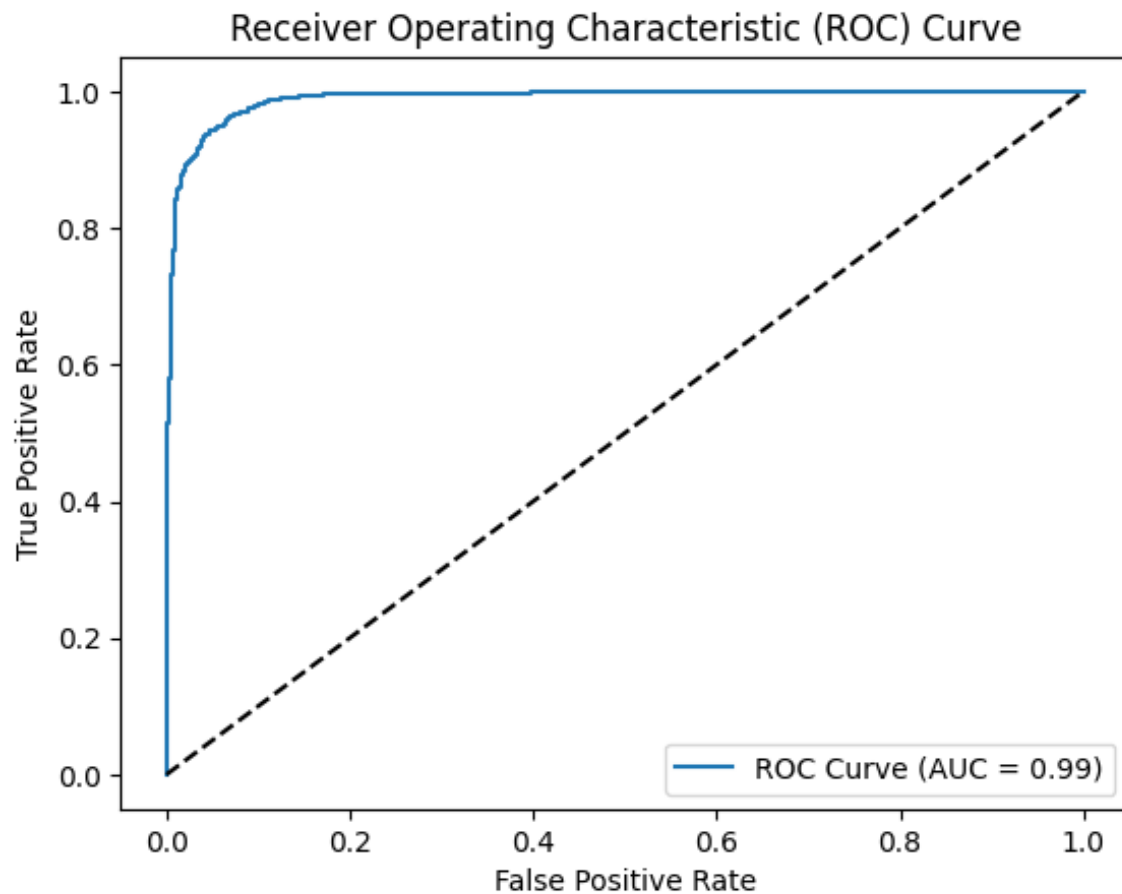
ROC Curve

```

1 from sklearn.linear_model import LogisticRegression
2 from sklearn.metrics import roc_curve , roc_auc_score
3 import matplotlib.pyplot as plt
4
5 # Train a Logistic Regression classifier
6 classifier = LogisticRegression()
7 classifier.fit(X_train_vec , y_train)
8
9 # Make predictions on the test set
10 y_scores = classifier.predict_proba(X_test_vec)[: , 1]
11
12 # Calculate the false positive rate, true positive rate, and thresholds
13 fpr , tpr , thresholds = roc_curve(y_test , y_scores)
14
15 # Calculate the AUC (Area Under the ROC Curve)
16 auc = roc_auc_score(y_test , y_scores)
17
18 # Plot the ROC curve
19 plt.plot(fpr , tpr , label='ROC Curve (AUC = %0.2f)' % auc)
20 plt.plot([0 , 1] , [0 , 1] , 'k—') # Diagonal line representing random guessing
21 plt.xlabel('False Positive Rate')
22 plt.ylabel('True Positive Rate')
23 plt.title('Receiver Operating Characteristic (ROC) Curve')
24 plt.legend(loc='lower right')

```

```
25 plt.show()
```



Recall,F1 Score,

```
1 from sklearn.model_selection import cross_validate
2 from sklearn.linear_model import LogisticRegression
3 from sklearn.metrics import accuracy_score, recall_score, f1_score
4
5 # Feature extraction using TF-IDF vectorization
6 vectorizer = TfidfVectorizer()
7 X_vec = vectorizer.fit_transform(X)
8
9 # Train a Logistic Regression classifier
10 classifier = LogisticRegression()
11
12 # Perform cross-validation and calculate metrics
```

```

13 scoring = ['accuracy', 'recall', 'f1']
14 cv_results = cross_validate(classifier, X_vec, y, scoring=scoring, cv=5)
15
16 # Get the average scores across all folds
17 average_accuracy = cv_results['test_accuracy'].mean()
18 average_recall = cv_results['test_recall'].mean()
19 average_f1 = cv_results['test_f1'].mean()
20
21 # Print the average scores
22 print("Average Accuracy:", average_accuracy)
23 print("Average Recall:", average_recall)
24 print("Average F1 Score:", average_f1)

```

```

1  Average Accuracy: 0.930346232179226
2  Average Recall: 0.9746923164519817
3  Average F1 Score: 0.9391264350852293

```

Chapter 4

Conclusion and Future Work

Implementing a robust customer support strategy on Twitter is essential for businesses to enhance customer satisfaction, build brand loyalty, and foster positive customer relationships. The Customer Support on Twitter project focuses on developing and implementing effective strategies to address customer inquiries, complaints, and feedback on the platform. By assembling a dedicated support team, establishing response guidelines, leveraging social media monitoring tools, and developing a comprehensive knowledge base, businesses can provide timely and personalized assistance to customers on Twitter.

Through sentiment analysis, intent classification, and other prediction techniques, businesses can gain valuable insights into customer sentiment, intent, and satisfaction levels. These techniques help optimize response times, prioritize support requests, and improve overall customer experience. Additionally, the project involves collaborating with other departments, such as product development and marketing, to facilitate efficient issue resolution and gather customer feedback for continuous improvement.

By measuring key performance indicators, such as response time, customer satisfaction ratings, and resolution rates, businesses can track the effectiveness of their customer support efforts and make data-driven improvements. This enables them to provide exceptional support, address customer concerns promptly, and foster positive brand perceptions on Twitter.

Ultimately, a well-executed Customer Support on Twitter project enables businesses to meet customer preferences, enhance brand visibility, and gain valuable customer insights.