SENTIMENT ANALYSIS USING PYTHON

**PROJECT REPORT**

***Submitted by***

*Gurditya Khurana [RA1911003010572]*

*Kasshish Raina [RA1911003010573]*

*Parikshit Singh [RA1911003010593]*

*On 08/04/2022*

TABLE OF CONTENTS

|  |  |
| --- | --- |
| CONTENTS | PAGE NO. |
| ABSTRACT | Page-1 |
| ACKNOWLEDGEMENTS | Page-2 |
| INTRODUCTION | Page-3 |
| LITERATURE SURVEY | Page-4 |
| METHODOLOGY | Page-6 |
| SOFTWARE REQUIREMENTS | Page-8 |
| WORKING MODEL DESCRIPTION & RESULT | Page-9 |
| ADVANTAGES | Page-15 |
| CONCLUSION | Page-16 |
| REFERENCES | Page-17 |

ABSTRACT

The aim of this project is to develop a operational classifier for accurate and automatic sentiment classification of a known tweet data set. In the past decade, new forms of communication, such as microblogging and text messaging have emerged and become ubiquitous. Python sentiment analysis is a methodology for analyzing a piece of text to discover the sentiment hidden within it. It accomplishes this by combining machine learning and natural language processing (NLP). Sentiment analysis allows us to examine the feelings expressed in a piece of text. While there is no limit to the range of information conveyed by tweets and texts, often these short messages are used to share opinions and sentiments that people have about what is going on in the world around them. We have worked on the following task, the task is: Given a tweet, classify whether the tweet is of positive, negative, or neutral sentiment.

ACKNOWLEDGEMENTS

I would like to express our team’s warm gratitude to our guide, Mrs. B. Priyalakshmi for her valuable guidance, consistent encouragement, personal caring, timely help and providing us with an excellent atmosphere for doing research. All through the work, in spite of her busy schedule, she has extended cheerful and cordial support to me for completing this research work.

INTRODUCTION

Twitter has emerged as a major micro-blogging website, having over millions of users and generating over millions tweets every day. Twitter has consistently attracted users to convey their opinions and perspective about any issue, brand, company etc. Due to this reason, Twitter is used as an informative source by many organizations, institutions and companies. On Twitter, users are allowed to share their opinions in the form of tweets, using only 140 characters. This leads to people compacting their statements by using abbreviations, emoticons, short forms etc. Along with this, people convey their opinions by using sarcasm and polysemy. Hence it is justified to term the Twitter language as unstructured. In order to extract sentiment from tweets, sentiment analysis is used. The results from this can be used in many areas like analyzing and monitoring changes of sentiment with an event, sentiments regarding a particular brand or release of a particular product, analyzing public view of government policies etc. A lot of research has been done on Twitter data in order to classify the tweets and analyze the results. In this paper we aim to review of some researches in this domain and study how to perform sentiment analysis on Twitter data using Python. The scope of this paper is limited to that of the machine learning models and we show the comparison of efficiencies of these models with one another.

LITERATURE SURVEY

There are two techniques widely used to detect the sentiments from text. They are Symbolic techniques and Machine Learning techniques. Sentiment analysis using Symbolic Techniques. A symbolic technique uses the availability of lexical resources. Turney suggested an approach for sentiment analysis called „bag of words‟. In the mentioned approach, individual words are neglected and only collections of words are considered. He gathered word having adjectives or adverb for the polarity of review from a search engine Altavista. A lexical database called WordNet was used by Kamps et al which determines an emotional matter in a word. WordNet carries synonyms and distance metric to find the orientation of adjectives. To overcome obstacles in lexical substitution task, Baroni et al developed a system supported by word space model formalism thereby representing local words. Emoti Net conceptually represented the text that stored the structure of real events in a domain. This was introduced by Balahur et al . B. Sentiment analysis using Machine Learning Techniques Under this technique, there are two sets, namely a training set and a test set. Generally the dataset which is collected from different sources and whose behavior and output values are known to us falls into the category of training data sets. In contrast with this, the datasets whose values or behavior are unknown to us are called as test data sets. Here different classifiers are trained with training data and then unknown data or we can say a test data is given to this model to get desired results. Machine Learning consists of various different classifiers such as Ensemble classifier, k-means, Artificial Neural Network etc. These are used to classify reviews . Y.Mejova et al in his research work proposed that we can use presence of each character, frequency of occurrences of each character, word which is considered as negation etc. as features for creating feature vector. He also shows that we can effectively use unigram and bigram approaches to make feature vector in Sentiment analysis. Domingos et al suggested that Naive Bayes works well for dependent features for certain problem. Zhen Niu et al found a new model. This model is based on Bayesian algorithm. In this model, some efficient approaches are used for selecting feature, computation of weight and classification. Barbosa et al designed a 2 step analysis method which is an automatic sentiment analysis for classifying tweets. In the first step, tweets are classified into subjective and objective tweets. After that, in a second step, subjective tweets are classified as positive and negative tweets. Celikyilmaz et al developed one method as pronunciation based word clustering. This method normalizes noisy tweets. There are some words which have the same pronunciation but having different meanings. So, for eliminating this conflict, there is method mentioned above. In this mentioned method, words having same pronunciation are clustered and assigned common tokens. Wu et al in his paper recommended model, namely, the influence probability to analysis the sentiment tweets. In this, if @username is found in the tweet, it takes influencing action and helps to influencing probability. By collecting automatic tweets, Pak et al developed a method for sentiment analysis by creating twitter corpus. In his proposed work he shows that, while creating feature vector, we can use emoticons as a feature. He used a Naïve Bayesian classifier to do the sentiment analysis. Some researches made to identify the public opinion about movies, news etc. from twitter tweets. V.M. Kiran et al had taken the information from other publicly available databases like IMDB and Blippr.

METHODOLOGY

In this project, we will be building our interactive Web-app data dashboard using streamlit library in Python, where we will be doing sentiment analysis of Twitter US Airline Data. Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science.

*We will be focusing on three objectives:*

1. Build interactive data dashboards with Streamlit and Python.
2. Use pandas for data manipulation in data science workflows.
3. Create interactive plots with Plotly Python.

A picture containing text, sign, outdoor, street

Description automatically generated

Fig1:- Flowchart for sentiment analysis

So, our project will be divided into following steps:

**STEP 1: Install streamlit library**

**STEP 2: Import packages and Libraries**

**STEP 3: Load the Twitter US Airline Sentiment Data**

**STEP 4: Display Tweets in the Sidebar**

**STEP 5: Plot Interactive Bar Plots and Pie Charts**

**STEP 6: Plotting Location Data on an Interactive Map**

**STEP 7: Plot Number of Tweets by Sentiment for Each Airline**

**STEP 8: Plotting Sentiment**

**STEP 9: Word Cloud for Positive, Neutral, and Negative Tweets**

**STEP 10: Plotting star rating for each airline**

SOFTWARE REQUIREMENTS

1. Python 3.8 :- It helps us to run python in our system with the help of different IDEs.
2. Spyder IDE :- **Spyder** is a free and open source scientific environment written in **Python**, for **Python**, and designed by and for scientists, engineers and data analysts.

WORKING MODEL DESCRIPTION & RESULT

Libraries Required:

**Streamlit** is an open-source app framework for Machine Learning and Data Science teams. Create beautiful data apps in hours, not weeks. All in pure Python.

**Pandas** is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language

**NumPy** offers comprehensive mathematical functions, random number generators, linear algebra routines, Fourier transforms, and more.

**Plotly Express** is the easy-to-use, high-level interface to Plotly, which operates on a variety of types of data and produces easy-to-style figures. It provides functions to visualize a variety of types of data. Most functions such as px. bar or px.

Code:

import streamlit as st

import pandas as pd

import numpy as np

import plotly.express as px

st.title("Data Analysis of American Airways 📈")

st.sidebar.title("Data Analysis of American Airways 📈")

st.markdown("Analysis of Customer's feedbacks 🏴")

st.sidebar.markdown("Analysis with Python 🏴")

DATA\_URL = ("Twitter\_dataset.csv")

@st.cache(persist=True)

def load\_data():

data = pd.read\_csv(DATA\_URL)

data['tweet\_created'] = pd.to\_datetime(data['tweet\_created'])

return data

data = load\_data()

st.sidebar.subheader("Show random tweet")

random\_tweet = st.sidebar.radio('Sentiment 😊😐😤', ('positive', 'neutral', 'negative'))

st.sidebar.markdown(data.query('airline\_sentiment== random\_tweet')[["text"]].sample(n=1).iat[0, 0])

st.sidebar.markdown("### Number of tweets by Sentiment")

select = st.sidebar.selectbox('Visualization type', ['Histogram', 'Pie Chart'], key='1')

sentiment\_count = data['airline\_sentiment'].value\_counts()

sentiment\_count = pd.DataFrame({'Sentiment': sentiment\_count.index, 'Tweets': sentiment\_count.values})

if not st.sidebar.checkbox("Hide", True):

st.markdown("### Number of tweets by Sentiment")

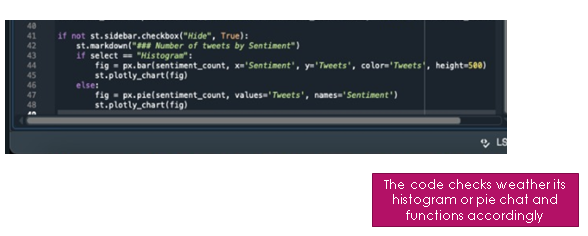
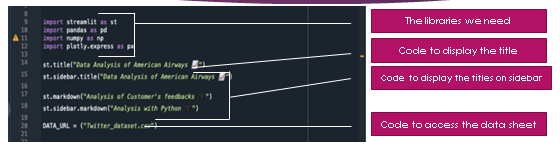
if select == "Histogram":

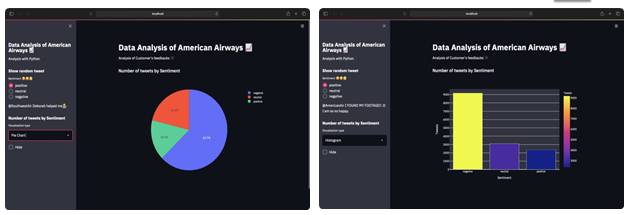
fig = px.bar(sentiment\_count, x='Sentiment', y='Tweets', color='Tweets', height=500)

st.plotly\_chart(fig)

else:

fig = px.pie(sentiment\_count, values='Tweets', names='Sentiment')

st.plotly\_chart(fig)



st.sidebar.subheader("When and where are the users tweeting from 🌎")

hour = st.sidebar.slider("Hour of day", 0, 23)

modified\_data = data[data['tweet\_created'].dt.hour == hour]

if not st.sidebar.checkbox("Close", True, key='<uniquevalueofsomesort>'):

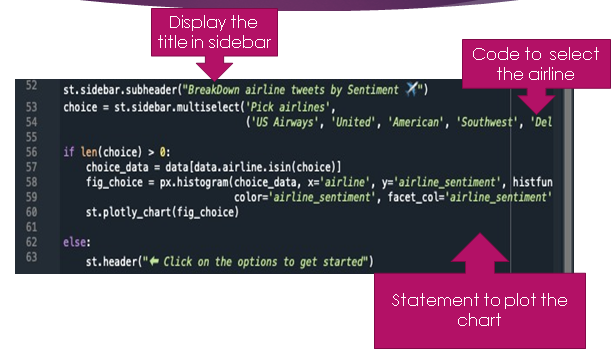
st.markdown("### Tweets location based on the time of day")

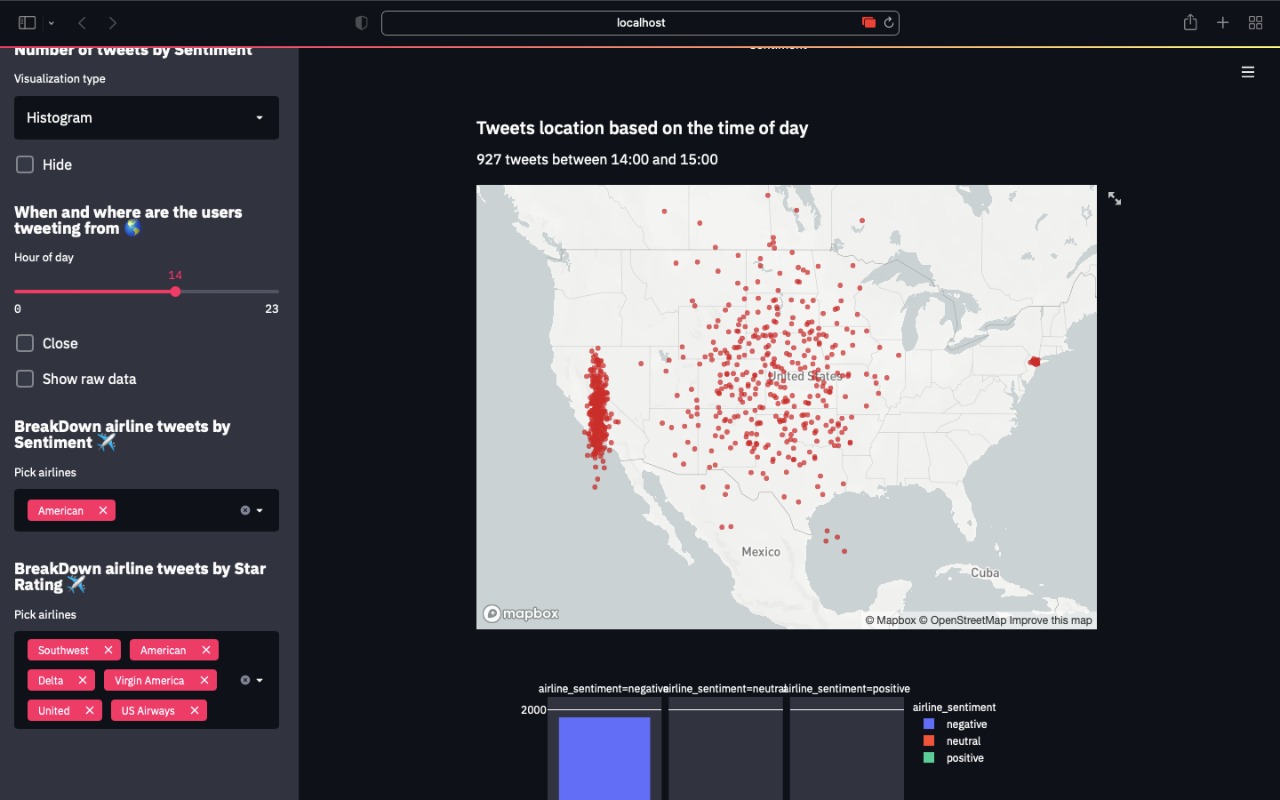
st.markdown("%i tweets between %i:00 and %i:00" % (len(modified\_data), hour, (hour + 1) % 24))

st.map(modified\_data)

if st.sidebar.checkbox("Show raw data", False):

st.write(modified\_data)





st.sidebar.subheader("BreakDown airline tweets by Sentiment ✈️")

choice = st.sidebar.multiselect('Pick airlines', ('US Airways', 'United', 'American', 'Southwest', 'Delta', 'Virgin America'))

if len(choice) > 0:

choice\_data = data[data.airline.isin(choice)]

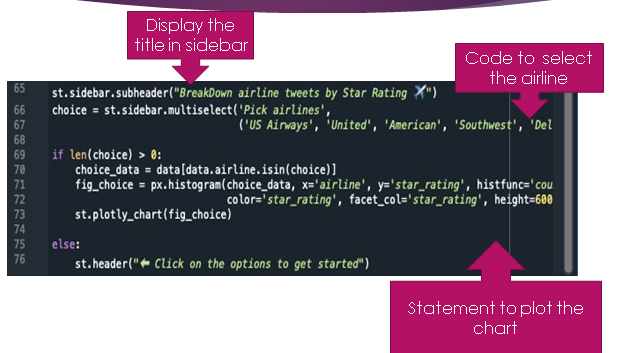
fig\_choice = px.histogram(choice\_data, x='airline', y='airline\_sentiment', histfunc='count',

color='airline\_sentiment', facet\_col='airline\_sentiment', height=600, width=600)

st.plotly\_chart(fig\_choice)

else:

st.header("⬅️ Click on the options to get started")





st.sidebar.subheader("BreakDown airline tweets by Star Rating ✈️")

choice = st.sidebar.multiselect('Pick airlines',

('US Airways', 'United', 'American', 'Southwest', 'Delta', 'Virgin America'),key='5')

if len(choice) > 0:

choice\_data = data[data.airline.isin(choice)]

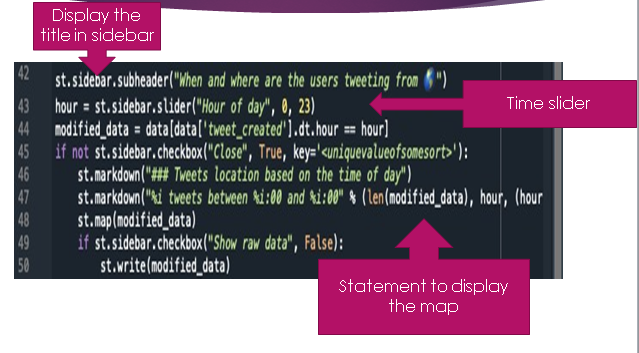
fig\_choice = px.histogram(choice\_data, x='airline', y='star\_rating', histfunc='count',

color='star\_rating', facet\_col='star\_rating', height=600, width=600)

st.plotly\_chart(fig\_choice)

else:

st.header("⬅️ Click on the options to get started")





ADVANTAGES

* Efficient prediction model to look into the sentiments of the people.
* The model does classification of tweets on the basis on sentiments.
* Geo-tagging [Latitude & Longitude] of tweets feature available.
* Breakdown of airline by sentiment feature available.
* Star Rating of airline on basis of sentiment feature is also there.

CONCLUSION

The microblogging sites like Twitter offers us a unprecedented opportunity to create and employ technologies that search for sentiments. The work done in this paper specifies an approach for sentiment analysis on Twitter data. To uncover the sentiment, we extracted the opinion words in the tweets. The overall tweet sentiment was calculated using a linear equation which incorporated emotion intensifiers too. This work is preliminary prototype. The initial results show that we are headed in the right direction.

REFERENCES

[1] [www.kaggle.com](http://www.kaggle.com)

[2]<https://monkeylearn.com/blog/sentiment-analysis-of-twitter/>

[3]<https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/>

[4]<https://www.analyticsvidhya.com/blog/2021/06/twitter-sentiment-analysis-a-nlp-use-case-for-beginners/>

[5]<https://towardsdatascience.com/twitter-sentiment-analysis-in-python-1bafebe0b566?gi=ccaabaf3996f>

[6]<https://developer.twitter.com/en/docs/tutorials/how-to-analyze-the-sentiment-of-your-own-tweets>

[7]<https://www.kdnuggets.com/2015/11/tutorial-twitter-sentiment-analysis.html>

[8]<https://ipullrank.com/step-step-twitter-sentiment-analysis-visualizing-united-airlines-pr-crisis>