SENTIMENT ANALYSIS OF AIRLINE TWEETS

Project Overview

Sentiment analysis has proven to be an effective way to understand people's opinion or sentiment about a topic of interest, a product or services to mention a few. It could be used in basically in any field ranging from health care to financial industries to marketing. The analysis can be used to judge or predict people sentiment towards a topic, issue, products or services. Sentiment analysis is used here to analyze customers confidence in an airline based on their experience. There were three classes of sentiments for this project which are neutral, positive, negative. Customers opinion are collected and analyzed based on their experience with six different airlines.

Background

Sentiment analysis provides a means of automating the classification of people's sentiments towards a topic of interest when presented with a document or text. In this analysis six airlines where surveyed using customer's tweets and a prediction model was trained and tested using the airline tweet dataset. This model will be used in predicting or classification of customer's experience based on customers sentiments towards each of these airlines.

The classification could be subjective classification –if sentiments is subjective or objective sentiment or polarity classification if sentiments are made based on categories (positive, negative or neutral). In this project I did a polarity classification on customers sentiments on different airlines

Data summary

The data that was used was an airline tweet dataset, which composed of six different airlines. The distribution of the tweet sentiments is shown in the bar chart in fig 1. The data summary for the tweets is:

Number of Airlines:	6
Number of tweets(rows):	14640
Number of columns:	15
Classes:	Negative (9178), Positive (2363), Neutral (3099)
Airline:	Virgin America, United Southwest, Delta, US airways, American
Column names:	tweet_id, airline_sentiment, airline_sentiment_confidence, negativereason, negativereason_confidence, airline, airline_sentiment_gold, name, negativereason_gold, retweet_count, text, tweet_coord, tweet_created, tweet_location, user_timezone

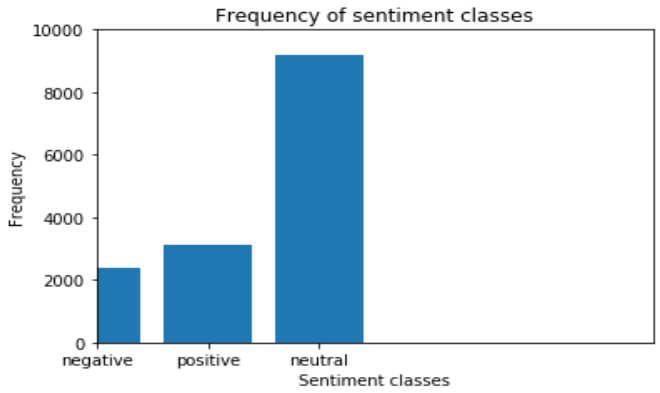

Fig1: Frequency of sentiment classes

Process

The airline tweet data set was divided into training and testing data set. The training dataset was used to build our model and train it to learn how to be able to predict an output given customer's tweeted airline sentiment data.

The pipeline model was built using the Term Frequency–Inverse Document Frequency (TFIDF) as the vectorizer and a logistic regression as its classifier. This model was chosen because takes into account the frequency of a word in a document, it also factors in how many documents are involved in which the word appears in. The traning and testing data was divided in the ratio 7 to 3 respectively. The testing dataset was used to test how accurate our model is and how well it can make prediction when presented with an input of customer's sentiments

Findings and Analysis

The testing dataset was used to test the model and it had an accuracy of approximately 72% which reflects how confident our model can predict given a certain a text input of customers sentiment from an airline. Since the logistic regression was used as our classifies, the accuracy level lies between 0 and 1.

From the accuracy results it shows that given an input(a sentiment data) the model can predict the out put with an accuracy of 72% approximately

From the graph it shows that when we have a small amount of data, it has a very low accuracy (that is prediction /accuracy level) but when we have large amount of data that our model can learn from, the prediction accuracy increases because it has more data to learn from.

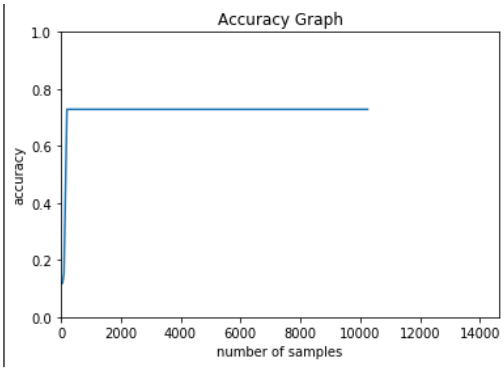

Fig 2: Accuracy graph

Goals

Overall, the goal of the project is to be able to predict customer's sentiment given a dataset of tweets based sentiments as input using the model that was built.

Conclusions

In conclusion, the model has an accuracy of 0.72 in predicting a customer's sentiment based on the given tweet dataset which is a tweet of their experience with different airlines. Therefore the model can predict the likelihood of a negative sentiment with an accuracy of 72%

CODE

```
import csv
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.feature_extraction.text import TfidfVectorizer
import matplotlib.pyplot as plt

data_set = open(r"C:\Users\Jane\Desktop\Tweets.csv",encoding="utf8")
#read the data set excluding the colums
all_tweets_classes = [i for i in csv.reader(data_set)][0:]
all_tweets = all_tweets_classes[1:][:]

#seperate training and testing dataset
partition= int((70/100)*len(all_tweets))
all_training_tweets = all_tweets[0:partition]
training_tweets =[row[0] for row in all_training_tweets]
class_training_tweets = [row[1] for row in all_training_tweets]



all_testing_tweets = all_tweets[partition-1: ]
testing_tweets = [row[0] for row in all_testing_tweets]
class_testing_tweets = [row[1] for row in all_testing_tweets]

#creating a pipeline
pipeline_data = Pipeline(steps = [("vectorizer", TfidfVectorizer(ngram_range =(1,2))), ("classifier",
LogisticRegression())])

#visualization of our class distribution
plt.axis([0, 5, 1,10000])
plt.xlabel("Sentiment classes")
plt.ylabel("Frequency")
plt.title("Frequency of sentiment classes")
data = open(r"C:\Users\Jane\Desktop\Tweets.csv",encoding="utf8")
tweet_data = [i[1] for i in csv.reader(data)][1:]
print(len(tweet_data))
tweet_data.sort()
```

```
negative =[i for i in tweet_data if i == "negative"]
print(len(negative))
positive =[i for i in tweet_data if i == "positive"]
print(len(positive))
neutral =[i for i in tweet_data if i == "neutral"]
print(len(neutral))
plt.bar(["negative","positive","neutral"],[len(positive),len(neutral),len(negative)])
plt.show()

#training and testing of our model
accuracy_list = []
for batch in (20, 50, 100, 200, 500, 1000, 2000, 3000, len(training_tweets)):
    pipeline_data.fit(training_tweets[:batch], class_training_tweets[:batch])
    print("tfidf",pipeline_data.score(testing_tweets,class_testing_tweets))
    accuracy_list.append(pipeline_data.score(testing_tweets,class_testing_tweets))

#Visualiztion of our of the accuracy of our model
plt.axis([0, len(all_tweets), 0, 1])
plt.xlabel("number of samples")
plt.ylabel("accuracy")
plt.title("Accuracy Graph")
plt.plot([20,50,100, 200, 500, 1000, 2000, 3000, len(training_tweets)], accuracy_list)
```