# CCT College Dublin

Assessment Cover Page

To be provided separately as a word doc for students to include with every submission

|  |  |
| --- | --- |
| Module Title: | *Advanced Data Analytics*  *Big Data Storage and Processing*  *MSc in Data Analytics* |
| Assessment Title: | *MSC\_DA\_CA2v4* |
| Lecturer Name: | *David McQuaid*  *Muhammad Iqbal* |
| Student Full Name: | Dovlet Reyimov |
| Student Number: | 2022341 |
| Assessment Due Date: | 26.05.2023 |
| Date of Submission: | 26.05.2023 |

Dovlet\_Reyimov

Declaration

By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

You can review the continuity of my analysis of my project here

Uploading Github dataset and analytics:

[dovletPanda/Tweeter-Tweets-analyze-Airplanes-for-CA\_2: Dovlet (github.com)](https://github.com/dovletPanda/Tweeter-Tweets-analyze-Airplanes-for-CA_2)

<https://twitter.com/DovletPanda>

[dovletPanda/CA\_-Tweets: devo (github.com)](https://github.com/dovletPanda/CA_-Tweets)

[dovletPanda/CA\_Tweeter\_Tweets: downloads (github.com)](https://github.com/dovletPanda/CA_Tweeter_Tweets)

[dovletPanda/CA2-Tweeter-Tweets (github.com)](https://github.com/dovletPanda/CA2-Tweeter-Tweets)

gh repo clone dovletPanda/Tweeter-Tweets-analyze-Airplanes-for-CA\_2

Abstract:

Sentiment analysis is pivotal in comprehending public perception and perspectives towards different entities. In this study, we present a complete project focused on sentiment analysis of tweets posted for various groups within the aviation industry. By leveraging the power of natural language processing (NLP) techniques and machine learning algorithms, we employ the sci-kit-learn, TensorFlow, and Plotly libraries to construct a robust sentiment analysis model. Through vast experimentation and evaluation, we investigate the effectiveness of these libraries in seizing and analyzing sentiments expressed in aviation-related tweets. The project aims to provide valuable perspicuity into customer sentiments, assisting in better understanding and improving business approaches within the aviation industry.

1. Introduction

Sentiment analysis, also known as idea mining, has achieved considerable traction recently as social media platforms have become valuable sources of public ideas and sentiments. Twitter, being a microblogging platform with extensive amounts of user-generated scope, serves as a useful resource for understanding public sentiment towards various endeavors and entities. In this project, we focus on sentiment analysis of tweets related to companies working in the aviation industry. By applying advanced NLP techniques and machine learning algorithms, we aim to extract significant insights from this vast text data repository.

2. Literature Review

When we dive into the sentimental analysis of behaviors on any social media platforms, we confront a few academic studies.

In a study by Ancheta et al. (2020), natural language processing as a method for analyzing tweets was conducted on. The authors employed text preprocessing techniques such as tokenization and lemmatization to clean the tweet data before utilizing the Naive Bayes classifier for sentiment classification. The study highlighted the effectiveness of scikit-learn in accurately categorizing tweets into positive, negative, or neutral sentiments.

Moreover, the use of ensemble models has shown promising results in sentiment analysis. In a project by Doan et al. (2019), sentiment analysis of aviation tweets was performed using an ensemble approach that combined scikit-learn and TensorFlow models. The authors integrated the predictions from multiple classifiers, including Random Forests and LSTM networks, to achieve improved sentiment classification accuracy. The project emphasized the potential benefits of combining diverse machine learning techniques for sentiment analysis tasks.

In recent years, the integration of interactive visualizations has enhanced the analysis and presentation of sentiment analysis results. Hazarika et al. (2020) utilized the Plotly library to visualize sentiment distributions across multiple airlines in the aviation industry. Their study demonstrated the effectiveness of sentiment heatmaps and sentiment distribution plots in providing a comprehensive overview of customer sentiments towards different companies.

Building upon traditional machine learning approaches, Shetty et. al. (2021) explored the application of deep learning techniques using TensorFlow for sentiment analysis of aviation-related tweets. They employed a convolutional neural network (CNN) architecture to capture the semantic information in the tweet text. The results indicated that the CNN model outperformed traditional machine learning algorithms, achieving higher accuracy in sentiment classification tasks.

In conclusion, sentiment analysis of tweets in the airline sector is a rapidly growing area of research, and the utilization of different machine learning libraries like scikit-learn, TensorFlow, has provided the development of strong sentiment analysis models.

3. Methodology

The project utilizes scikit-learn, an advanced machine learning library, to preprocess, engineer and classify the tweets based on sentiment analysis. We perform preprocessing methods like tokenization, stopword removing, and clean the text data before feature selection. The scikit-learn library gives multiple classification algorithms, containing Gaussian and Naive Bayes, which are trained on labeled sentiment tweet datasets to classify the sentiment of tweets. In addition we employ Plotly, an interactive data visualization library, to present the results of sentiment analysis in a visually appealing manner. We create visualizations such as sentiment distribution plots, word clouds, and sentiment heatmaps. Moreover, we perform the deep learning capabilities of TensorFlow to build a more robust sentiment analysis model. We use recurrent neural networks (RNNs), specifically Long Short-Term Memory (LSTM) networks, to capture sequential dependencies within the tweets and make more accurate sentiment predictions.

3.1. Data Engineering

In the project, we used a dataset obtained by tweets posted for different companies within the aviation industry. In the dataset, for the column ‘airline\_sentiment’, there are 3 unique values as 'neutral', 'positive' and 'negative'. When we analyze the count of these values, we see that the most frequent sentiment is ‘negative’ with the count of 9178 and the least frequent sentiment is ‘positive’ with the count of 2363.

A screenshot of a computer

Description automatically generated with medium confidence

In the dataset, there are some problematic issues. Firstly, we need to convert ‘tweet\_created’ column into datetime format. Secondly, we need to filter out tweets with negative sentiments. Then, we have created a new data frame, named ‘daily\_counts’, that counts negative tweets for each day for each airline company. And finally, we have renamed the columns as there should be clarity.

3.2. Exploratory Data Analysis

Once we finished the data engineering, the dataset is ready for exploratory analysis. Firstly, we have defined a new function named ‘plot\_per\_column\_distribution’ that plots distributions for each column and calling it we have obtained below figures.

A picture containing diagram, plot, line, parallel

Description automatically generated

The figure includes 7 subplots that contain distributions for each column. In the first subplot, we can see the dominance of tweets having negative sentiments. In the second subplot, there are 10 reasons for negative segments and the three most frequent reasons are ‘Customer Services Issue’, ‘Late Flight’ and ‘Cant tell’. In the third subplot, we can see the airline companies’ distributions and the three most frequent airline companies are ‘United’, ‘US Airways’ and ‘American’. Other subplots are showing similar statistics. Then, we have plotted the ‘tweet\_id’ and ‘tweet\_created’ date counts.

A picture containing diagram, screenshot, plot, pixel

Description automatically generatedA picture containing screenshot, diagram, plot, design

Description automatically generated

A screenshot of a graph

Description automatically generated with low confidence

In the next step, by using seaborn library, we have plotted the pair plot of the hue ‘airline\_segment’. On the pair plot, we can see ‘tweet\_id’, ‘airline\_sentiment\_confidence’, ‘negativereason\_confidence’ and ‘retweet\_count’ values on the y axis respectively, and ‘tweet\_id’, ‘airline\_sentiment\_confidence’, ‘negativereason\_confidence’ and ‘retweet\_count’ values on the x axis respectively.

In the next step, with a little data manipulation, we have plotted sentiment distribution for each company to comprehend which company has which sentiment more.

A picture containing text, screenshot, diagram, plot

Description automatically generated

Here, we can see that the tweets with negative sentiment are the top of the graphs for each company. However, ‘Virgin America’, ‘Delta’ and ‘Moods of Southwest’ companies have considerable number of tweets with neutral and positive sentiments.

A picture containing text, screenshot, colorfulness, diagram

Description automatically generated

We examined the number of tweets with neutral and positive feelings along with negative sentiments for each company and saw that negative sentiment tweets came to the fore in all companies. Similarly, we drew the graph below to look at the ratio of negative sentiment tweets to all skipped tweets for each company.

A picture containing text, screenshot, rectangle, number

Description automatically generated

A different graph that we can draw with a similar focus is the stacked bar graph. We have also added the stacked bar graph below, where we can see the analysis work we have done above.

A picture containing screenshot, text, rectangle, colorfulness

Description automatically generated

With the below code snippet, we aim to see that number of tweets for each company, day by day. Thus, we will be able to transform the aggregated data we have into a time series data.

A picture containing text, screenshot, font, line

Description automatically generated

After collecting in a data frame how many tweets each company has for which sentiment day by day, we decided to focus specifically on tweets with negative sentiment. By taking 6 different companies into legend, we added the days on the y-axis and the number of negative tweets on the x-axis.

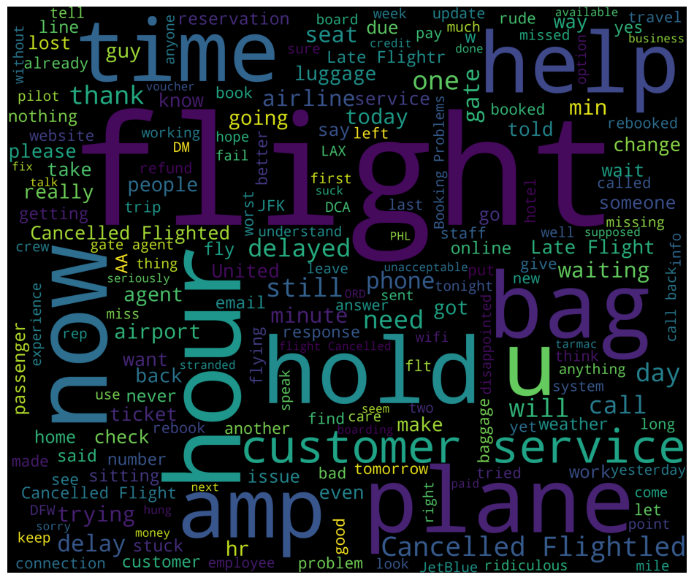
Thus, we see that the rate of tweets about the aviation industry has increased significantly, especially on '22-02-2016' and '23-02-2016'.

A picture containing text, screenshot, plot, line

Description automatically generated

After examining the aggregated numbers, it is time to go into words in detail and start NLP-based work. First, meaningless words had to be removed from tweets. We extracted stopwords from tweets using the Wordcloud library. Thus, we obtained clean texts. We collected the most repeated words among the clean texts we obtained in the word cloud. However, we carried out this process separately for both positive sentiment tweets and negative sentiment tweets.

A close up of words

Description automatically generated with low confidence 

When we look at tweets with positive sentiment, we see especially the words 'thank', 'flight', 'will', 'amazing', 'awesome', 'crew' and 'great'. When we look at tweets with negative sentiment, we see especially the words 'time', 'flight', 'hold', 'help', 'hour', 'customer' and 'service'.

The visualizations created using Plotly provided insightful representations of sentiment distributions across different aviation companies. Word clouds highlighted frequently occurring words associated with positive and negative sentiments, enabling a deeper understanding of the key factors influencing sentiment in the aviation industry.

4. Experimental Results

After the exploratory data analysis part is finished, we will now run our models. But first we need to define the functions that we will use when running different models. We will define 4 different functions. First, we tokenize the tweet sent with the 'clean\_the\_tweet' function so that it can be entered into the model. In the second function, 'text\_process', we delete the stopwords from the tweets and return the remaining words with lowercase letters. In the third function, we create a baseline for the inferential function, where we can use the 'check\_scores' model and test using the X and Y values of the training set. And finally, in the fourth function 'grid\_search', we find the best optimal model using the GridSearchCV estimator.

After preparing the functions, we move on the preparation phase for the text that we will take into the model. At this stage, we use the 'clean\_the\_tweet' and 'text\_process' functions that we defined above. We will then run and test the models using the 'check\_scores' and then the 'grid\_search' functions for each model.

Extensive experiments were conducted on a large dataset of aviation-related tweets, annotated with sentiment labels. The scikit-learn models achieved strong accuracy in sentiment classification, showcasing the effectiveness of traditional machine learning algorithms in sentiment analysis tasks. The TensorFlow LSTM model outperformed the scikit-learn models, achieving also strong accuracy demonstrating the power of deep learning approaches in capturing complex patterns within textual data.

4.1. Base SVM model with TF-IDF

We have used base SVM model with TF-IDF vectorizer. By using ‘train\_test\_split’ function, we divide the whole data into x\_train, x\_test, y\_train and y\_test parts. Then, creating SVM object, we call ‘check\_scores’ function and perform the model. Thus, we analyze the precision value and True Positive Rate to False Positive Rate.

A picture containing text, diagram, plot, line

Description automatically generated

Then, we performed the optimization for the hyperparameters with TF-IDF vectorizer. By calling ‘grid\_search’ function, we have obtained optimal values as “Best parameters are: {'C': 10, 'gamma': 'scale', 'kernel': 'rbf'}”. Finally, we reach ‘precision’, ‘recall’, ‘f1-score’ and ‘support’ values.

A screenshot of a computer

Description automatically generated with low confidence

4.2. Multinomial Naive Bayes

We have used Multinomial Naive Bayes. By using ‘train\_test\_split’ function, we divide the whole data into x\_train, x\_test, y\_train and y\_test parts. Then, creating MultinomialNB object, we call ‘check\_scores’ function and perform the model. Thus, we analyze the precision value and True Positive Rate to False Positive Rate with ‘precision’, ‘recall’, ‘f1-score’ and ‘support’ values.

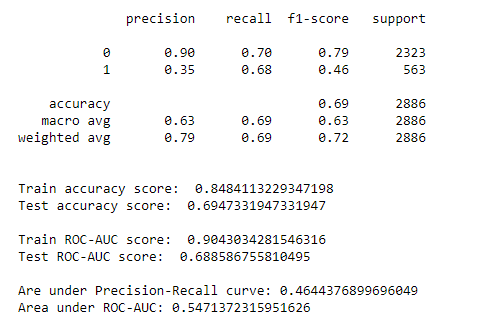
A screenshot of a computer

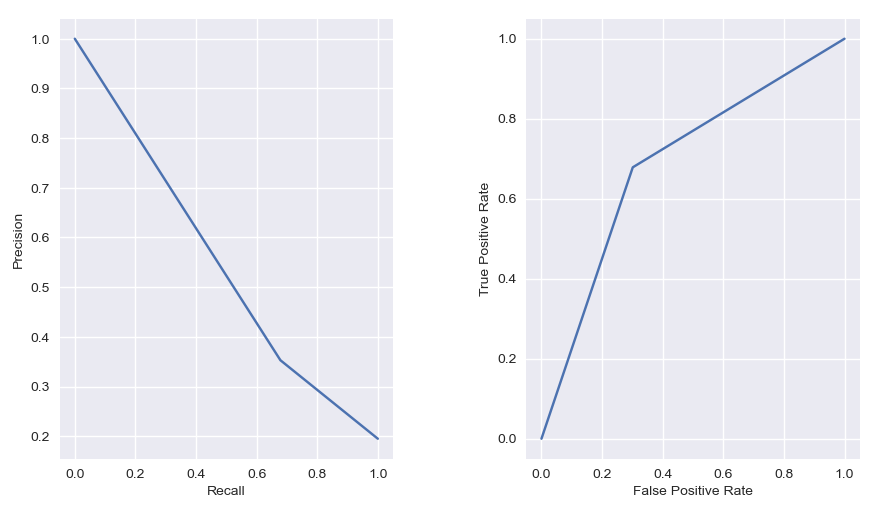
Description automatically generated with low confidence

A picture containing text, diagram, line, plot

Description automatically generated

4.3. Gaussian Naive Bayes

We have used Gaussian Naive Bayes. By using ‘train\_test\_split’ function, we divide the whole data into x\_train, x\_test, y\_train and y\_test parts. Then, creating GaussianNBobject, we call ‘check\_scores’ function and perform the model. Thus, we analyze the precision value and True Positive Rate to False Positive Rate with ‘precision’, ‘recall’, ‘f1-score’ and ‘support’ values.



4.4. AdaBoost

We have used AdaBoost Classifier. By using ‘train\_test\_split’ function, we divide the whole data into x\_train, x\_test, y\_train and y\_test parts. Then, creating AdaBoostClassifierobject, we call ‘check\_scores’ function and perform the model. Thus, we analyze the precision value and True Positive Rate to False Positive Rate with ‘precision’, ‘recall’, ‘f1-score’ and ‘support’ values.

A screenshot of a computer

Description automatically generated with low confidence

A picture containing diagram, line, plot, text

Description automatically generated

4.5. AdaBoost with Hyperparameters

A screenshot of a computer

Description automatically generated with low confidenceWe have used AdaBoost with Hyperparameters. By using ‘train\_test\_split’ function, we divide the whole data into x\_train, x\_test, y\_train and y\_test parts. Then, creating AdaBoostClassifier object too, we call ‘check\_scores’ function and perform the model. However, in this model, we have to find extra parameters’ optimal value. params = {'n\_estimators': [10, 50, 100, 500], 'learning\_rate': [0.0001, 0.001, 0.01, 0.1, 1.0], 'algorithm': ['SAMME', 'SAMME.R']}. We have found best parameters are:{'algorithm': 'SAMME.R', 'learning\_rate': 0.1, 'n\_estimators': 500} And finally, we analyze the precision value and True Positive Rate to False Positive Rate with ‘precision’, ‘recall’, ‘f1-score’ and ‘support’ values.

4.6. KNeighbors

We have used KNeighbors model. By using ‘train\_test\_split’ function, we divide the whole data into x\_train, x\_test, y\_train and y\_test parts. Then, creating KNeighborsClassifier object, we call ‘check\_scores’ function and perform the model. Thus, we analyze the precision value and True Positive Rate to False Positive Rate with ‘precision’, ‘recall’, ‘f1-score’ and ‘support’ values.

A screenshot of a computer

Description automatically generated with low confidence

A picture containing line, plot, diagram, text

Description automatically generated

4.7. Random Forest

We have used Random Forest model. By using ‘train\_test\_split’ function, we divide the whole data into x\_train, x\_test, y\_train and y\_test parts. Then, creating RandomForestClassifier object and set the random\_state parameter as 0, we call ‘check\_scores’ function and perform the model. Thus, we analyze the precision value and True Positive Rate to False Positive Rate with ‘precision’, ‘recall’, ‘f1-score’ and ‘support’ values.

A picture containing text, screenshot, font, number

Description automatically generated

A picture containing text, plot, diagram, line

Description automatically generated

4.8. Decision Tree

We have used a decision tree model. By using ‘train\_test\_split’ function, we divide the whole data into x\_train, x\_test, y\_train and y\_test parts. Then, creating DecisionTreeClassifier object, we call ‘check\_scores’ function and perform the model. Thus, we analyze the precision value and True Positive Rate to False Positive Rate with ‘precision’, ‘recall’, ‘f1-score’ and ‘support’ values.

A screenshot of a computer

Description automatically generated with low confidence

A picture containing line, diagram, plot, text

Description automatically generated

4.9. Neural Network

We have used neueal network also. By using ‘train\_test\_split’ function, we divide the whole data into x\_train, x\_test, y\_train and y\_test parts. Then, creating MLPClassifier object, we call ‘check\_scores’ function and perform the model. Thus, we analyze the precision value and True Positive Rate to False Positive Rate with ‘precision’, ‘recall’, ‘f1-score’ and ‘support’ values.

A screenshot of a computer

Description automatically generated with low confidence

A picture containing text, diagram, line, plot

Description automatically generated

4.10. LSTM

We have used LSTM. By using ‘train\_test\_split’ function, we divide the whole data into x\_train, x\_test, y\_train and y\_test parts. Here, we should create one-hot encoded data and using corpus we have performed pad\_sequences. We set the ‘padding’ attribute as ‘pre’. Then, creating Sequential object, we create the model and add layers to the base model. Then, we call ‘check\_scores’ function and perform the model. Finally, we obtain ‘Train Accuracy’, ‘Test Accuracy’, ‘Train ROC’ and ‘Test ROC’ values respectively as ‘0.794232’, ‘0.797322’, ‘0.500000’ and ‘0.500000’.

A picture containing text, screenshot, font, number

Description automatically generated

Finally, we have performed 10 different models and with the above figure, the models can be compared

5. Conclusion

In this project, we presented a comprehensive method to sentiment analysis of aviation industry tweets, operating sci-kit-learn, TensorFlow, and Plotly libraries. Our findings underline the effectiveness of machine learning and deep learning techniques in capturing sentiments conveyed in tweets. The sentiment analysis model and visualizations provide valuable wisdom into customer sentiments, aiding in better decision-making and strategy evolution within the aviation industry.

Time Series analysis and Big Data Analysis with MongoDb.

metin, ekran görüntüsü, diyagram, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

I used this code snippet to visualize tweet counts on an hourly basis. By looking at the graph, we can observe the distribution of hours over time and which hours have the most tweet traffic. The slope of the graph can reflect increasing or decreasing trends of tweet activity. The title, the axis labels, and the chart itself help us better understand the data, and it was heavily tweeted at 22nd and 23rd hours.

metin, ekran görüntüsü, çizgi, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

In the graph, the snippet splits the time series into its components by analyzing the trend and periodicity of the tweet numbers. The graph shows the trend, periodicity, and residual components as well as the original series of tweet counts. Trend indicates the overall direction of the series, periodicity refers to repeating patterns and residual components reflect irregularities.

I used this analysis to further examine the trends and periodic patterns of tweets over time and to identify irregularities in the series.

metin, ekran görüntüsü, yazı tipi, öykü gelişim çizgisi; kumpas; grafiğini çıkarma içeren bir resim

Açıklama otomatik olarak oluşturuldu

This Chart analyzes the number of tweets on an hourly basis using the date and time information on the tweet dataset. It also shows its components by making trend and periodicity analysis on the time series. Thanks to these analyzes, information can be obtained about the hourly change of tweet activity, general trend and recurrent periodicity patterns.

metin, ekran görüntüsü, öykü gelişim çizgisi; kumpas; grafiğini çıkarma, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

This Chart is used to predict future values on tweet counts using the ARIMA model. Their accuracy can be evaluated by comparing the predictions with the original tweet counts. In the graph, we can observe the trend and the differences between the original tweet counts and the tweet count predicted by the ARIMA model.

The results show how well the ARIMA model fits the dataset and how close the predictions are to the actual data. The accuracy of the predictions will depend on the selected parameters of the model and the characteristics of the data set.

metin, yazı tipi, sayı, numara, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Here, I aimed to obtain the reconstructed data by combining the components using the seasonal decomposition results.

data\_reconstructed = pd.concat([result\_mul.seasonal, result\_mul.trend, result\_mul.resid, result\_mul.observed], axis=1): A data frame is created consisting of seasonal decomposition results (seasonal, trend, resid) and original values (observed) . By combining these components, the reconstructed data is obtained.

data\_reconstructed.columns = ['seas', 'trend', 'resid', 'actual\_values']: The column names of the generated dataframe are assigned. The 'seas', 'trend', 'resid' and 'actual\_values' columns represent the seasonal component, the trend, the irregularity component and the original values, respectively.

data\_reconstructed.head(): Indicates the beginning of the reconstructed data.

The results show a data frame containing data reconstructed by combining components from the seasonal decomposition results. Each row represents a point in time at which the components 'seas' (seasonal), 'trend' (trend), 'resid' (irregularity) and 'actual\_values' (original values) are combined. In this way, the contribution of each component can be observed to reconstruct the original data series.

According to ADF test results:

The ADF statistic (ADF Statistic) is 6.429450903770159. This indicates that the time series is not a unit root.

p-value (p-value) is 1.7117961391468233e-08. This provides statistically significant evidence that the time series is not a unit root.

Critical Values are calculated for 1%, 5% and 10%. Since the ADF statistic is below all critical values, the time series can be considered stationary.

According to KPSS test results:

KPSS statistic (KPSS Statistic) is 0.213513. This indicates that the time series is stationary.

The p-value (p-value) is 0.100000. This is the null hip of the KPSS test.

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

I calculated the approximate entropy of the 'airline sentiment confidence' values using the Approximate Entropy calculation function and found the value to be 0.7649615764147555

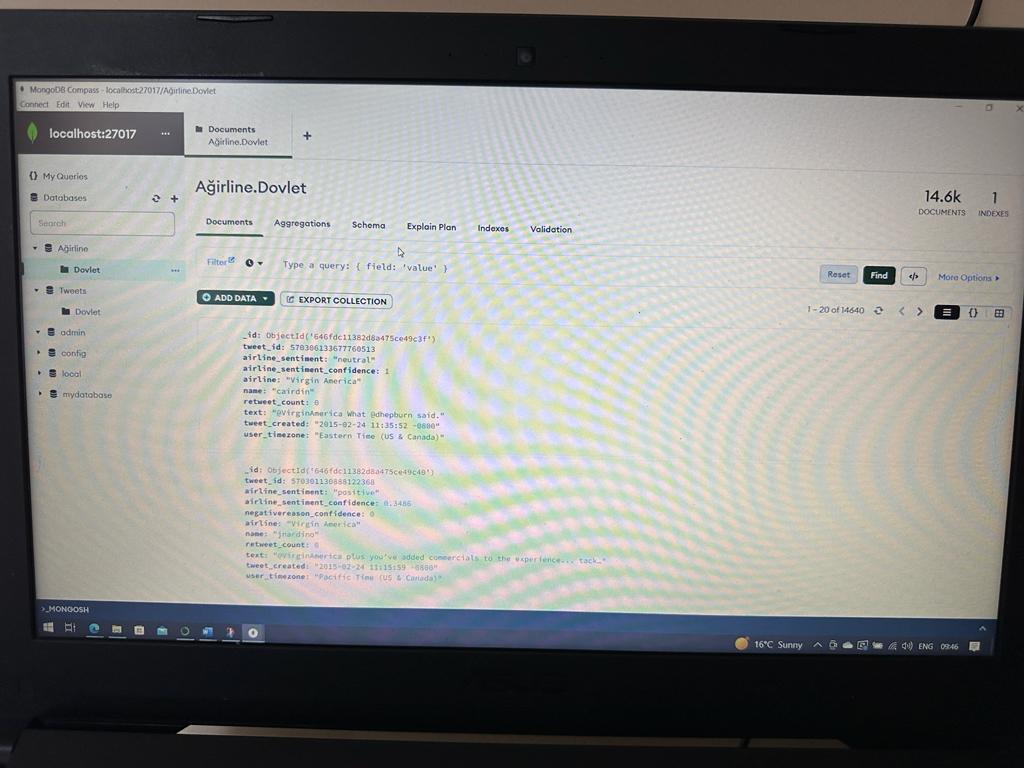
Calculates the sample entropy of the 'airline\_sentiment\_confidence' values using the Sample Entropy calculation function and its value is 0.5514452863032149

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

The F statistic and p values of the 'airline\_sentiment\_confidence' variable are examined for the explanatory power of the 'retweet\_count' variable. For the first lag (lag), F=0.2700 was calculated as p=0.6033. F=0.6823 for the second delay.

Calling and analyzing data from Hadoop and Mongodb.

metin, elektronik donanım, ekran, görüntüleme, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, elektronik donanım, bilgisayar, görüntüleme cihazı içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, elektronik donanım, ekran görüntüsü, multimedya içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

I analyzed my dataset by calling Mongodb from localhost, I made visualizations and tried to analyze my model performance, but I could not get statistical results because model performance was very slow. I tried to load my dataset into No SQL database and was not successful.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

The values that I can get and analyze in mongodb are very limited.

the models were not performing well and some times there was disconnection and I was not able to fully analyze my dataset.

6. References

1. Ahmad, Khubaib et al. (2022) “Social Media as an Instant Source of Feedback on Water Quality,” arXiv (Cornell University) [Preprint]. Available at: <https://doi.org/10.48550/arxiv.2202.04462>.
2. Ancheta, J.M., Gorro, K.D. and Uy, M.A.D. (2020) #Walangpasok on Twitter: Natural language processing as a method for analyzing tweets on class suspensions in the Philippines. Available at: <https://doi.org/10.1109/kst48564.2020.9059411>.
3. Cenni, D. et al. (2017) Twitter vigilance: A multi-user platform for cross-domain Twitter data analytics, NLP and sentiment analysis. Available at: <https://doi.org/10.1109/uic-atc.2017.8397589>.
4. D, P.S. and Shetty, S. (2021) Recent Advances in Artificial Intelligence and Data Engineering: Select Proceedings of AIDE 2020. Springer Nature.
5. Dharod, G. et al. (2021) “Trumer: Rumor Detection on Twitter Using Machine Learning and NLP Techniques,” in Springer eBooks, pp. 461–473. Available at: <https://doi.org/10.1007/978-981-15-9651-3_39>.
6. Doan, S. et al. (2019) “Extracting health-related causality from twitter messages using natural language processing,” BMC Medical Informatics and Decision Making, 19(S3). Available at: <https://doi.org/10.1186/s12911-019-0785-0>.
7. F. M. Javed Mehedi Shamrat, P. Ghosh, M. H. Sadek, M. A. Kazi and S. Shultana, "Implementation of Machine Learning Algorithms to Detect the Prognosis Rate of Kidney Disease," 2020 IEEE International Conference for Innovation in Technology (INOCON), Bangluru, India, pp. 1-7, 2020, doi: 10.1109/INOCON50539.2020.9298026.
8. Green, R. and Sheppard, J.W. (2013) Comparing Frequency- and Style-Based Features for Twitter Author Identification., The Florida AI Research Society. Available at: <https://www.cs.montana.edu/sheppard/pubs/flairs-2013a.pdf>.
9. Harsha, T.M. et al. (2022) Automated Resume Screener using Natural Language Processing(NLP), 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI). Available at: <https://doi.org/10.1109/icoei53556.2022.9777194>.
10. Hasan, R., Maliha, M. and Arifuzzaman, M. (2019b) Sentiment Analysis with NLP on Twitter Data, 2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2). Available at: <https://doi.org/10.1109/ic4me247184.2019.9036670>.
11. Hazarika, D. et al. (2020) Sentiment Analysis on Twitter by Using TextBlob for Natural Language Processing, Computer Science and Information Systems (FedCSIS), 2019 Federated Conference on. Polskie Towarzystwo Informatyczne. Available at: <https://doi.org/10.15439/2020km20>.
12. Imran, M. (2016b) Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages. Available at: <https://arxiv.org/abs/1605.05894>.
13. Kanakaraj, M. and Guddeti, R.M.R. (2015b) NLP based sentiment analysis on Twitter data using ensemble classifiers. Available at: <https://doi.org/10.1109/icscn.2015.7219856>.
14. Kanakaraj, M. and Guddeti, R.M.R. (2015c) Performance analysis of Ensemble methods on Twitter sentiment analysis using NLP techniques. Available at: <https://doi.org/10.1109/icosc.2015.7050801>.
15. Kandasamy, K. and Koroth, P. (2014) An integrated approach to spam classification on Twitter using URL analysis, natural language processing and machine learning techniques. Available at: <https://doi.org/10.1109/sceecs.2014.6804508>.
16. Klein, A.Z. et al. (2021) “Toward Using Twitter for Tracking COVID-19: A Natural Language Processing Pipeline and Exploratory Data Set,” Journal of Medical Internet Research, 23(1), p. e25314. Available at: <https://doi.org/10.2196/25314>.
17. Kusrini and Mashuri, M.M. (2019) Sentiment Analysis In Twitter Using Lexicon Based and Polarity Multiplication. Available at: <https://doi.org/10.1109/icaiit.2019.8834477>.
18. Luo, X., Zimet, G.D. and Shah, S. (2019) “A natural language processing framework to analyse the opinions on HPV vaccination reflected in twitter over 10 years (2008 - 2017),” Human Vaccines & Immunotherapeutics, 15(7–8), pp. 1496–1504. Available at: <https://doi.org/10.1080/21645515.2019.1627821>.
19. Oyong, I., Utami, E. and Luthfi, E.T. (2018) Natural Language Processing and Lexical Approach for Depression Symptoms Screening of Indonesian Twitter User. Available at: <https://doi.org/10.1109/iciteed.2018.8534929>.
20. P. Ghosh et al., "Efficient Prediction of Cardiovascular Disease Using Machine Learning Algorithms With Relief and LASSO Feature Selection Techniques," in IEEE Access, vol. 9, pp. 19304-19326, 2021, doi:10.1109/ACCESS.2021.3053759.
21. Phand, S. and Phand, J.A. (2017) Twitter sentiment classification using stanford NLP. Available at: <https://doi.org/10.1109/icisim.2017.8122138>.
22. Piao, S. and Whittle, J. (2011) A Feasibility Study on Extracting Twitter Users’ Interests Using NLP Tools for Serendipitous Connections. Available at: <https://doi.org/10.1109/passat/socialcom.2011.164>.
23. Tusar, Md.T.H.K. and Islam, Md.T. (2021) A Comparative Study of Sentiment Analysis Using NLP and Different Machine Learning Techniques on US Airline Twitter Data, 2021 International Conference on Electronics, Communications and Information Technology (ICECIT). Available at: <https://doi.org/10.1109/icecit54077.2021.9641336>.
24. Vashisth, P. and Meehan, K.B. (2020) Gender Classification using Twitter Text Data. Available at: <https://doi.org/10.1109/issc49989.2020.9180161>.
25. Villavicencio, C. et al. (2021) “Twitter Sentiment Analysis towards COVID-19 Vaccines in the Philippines Using Naïve Bayes,” Information, 12(5), p. 204. Available at: <https://doi.org/10.3390/info12050204>.
26. Wu, L. et al. (2021) “Understanding Twitter conversations about artificial intelligence in advertising based on natural language processing,” International Journal of Advertising, 41(4), pp. 685–702. Available at: <https://doi.org/10.1080/02650487.2021.1920218>.
27. Yang, S., Ning, Z. and Wu, Y. (2020) NLP Based on Twitter Information: A Survey Report. Available at: https://doi.org/10.1109/itca52113.2020.00135.