cct | College Dublin
Computing • IT • Business

BUSINESS REPORT

Plane Simple – Airline Company

Time Series and Text Sentiment Analysis

May 2023

PREPARED BY:
Leandro Marigo

CONTENTS

INTRODUCTION

Business Description

A new airline company, PlaneSimple, is looking to see if it can benefit from machine learning and has tasked me with performing several analyses on tweets about airline companies.

Part 1: Time Series - Forecasting Airline Interest
PlaneSimple is looking to model the frequency at which people Tweet about flying. They would like to model the number of tweets as a time series and make forecasts into the future.

a. Briefly explain why Time Series may be useful in this situation and describe the analysis objective.
b. Train an appropriate Time Series model on this data. Include justification for any hyper parameters chosen, and how validation was performed (e.g., train-test split).
c. Use the model to make a forecast for one week into the future. Note if data was resampled, a different forecast period may be applied, where appropriate.
d. Explain the results and conclusions, as well as discuss any limitations to the analysis.

Part 2: Text Analysis - Analysing the Tweet Content
PlaneSimple is interested not just in how many tweets there are, but what people are talking about.

a. Briefly explain why Text Analysis may be useful in this situation and describe the analysis objective.
b. Train an appropriate model to perform Sentiment Analysis on the tweets.
c. Train an appropriate model to perform Topic Modelling on the tweets.
d. Explain the results and conclusions, as well as discuss any limitations to the analysis.

Part 3: Combine both models

a. Train an appropriate time series model to forecast positive tweet sentiment.
b. Train an appropriate time series model to forecast an appropriate topic related to the subject.
c. Explain the results and conclusions, as well as discuss any limitations to my analysis.

SECTIONS

Section 1	Exploratory Data Analysis
Section 2	Time Series - Forecasting Airline Interest
Section 3	Text Analysis - Analyzing the Tweet Content
Section 4	Combine both models (Time-Series and Text Analysis)

Section 1: Exploratory Data Analysis

Dataset Attributes:

The dataset contains 15 features with a mix of numerical and categorical, and 14640 observations/rows.

It was used dataprep.eda to explore the distribution of data for all features and look for missing values and outliers.

Observations from Dataprep.EDA

Data insights

- Airline Sentiment is majoritarialy negative.
- Main reasons for negative sentiment are --> Customer Service and Late Flight.
- Most tweeted airline is United followed by US Airways.
- The variable 'tweet_created' is a string and needs to be converted to datetime type.

Missing values

- Total Missing Cells: 61962
- Total Missing Cells (%): 28.2%

Variables with most missing values:

- airline_sentiment_gold has 14600 (99.73%) missing values.
- negativereason_gold has 14608 (99.78%) missing values.
- tweet_coord has 13621 (93.04%) missing values.

Section 2: Time Series – Forecasting Airline Interest

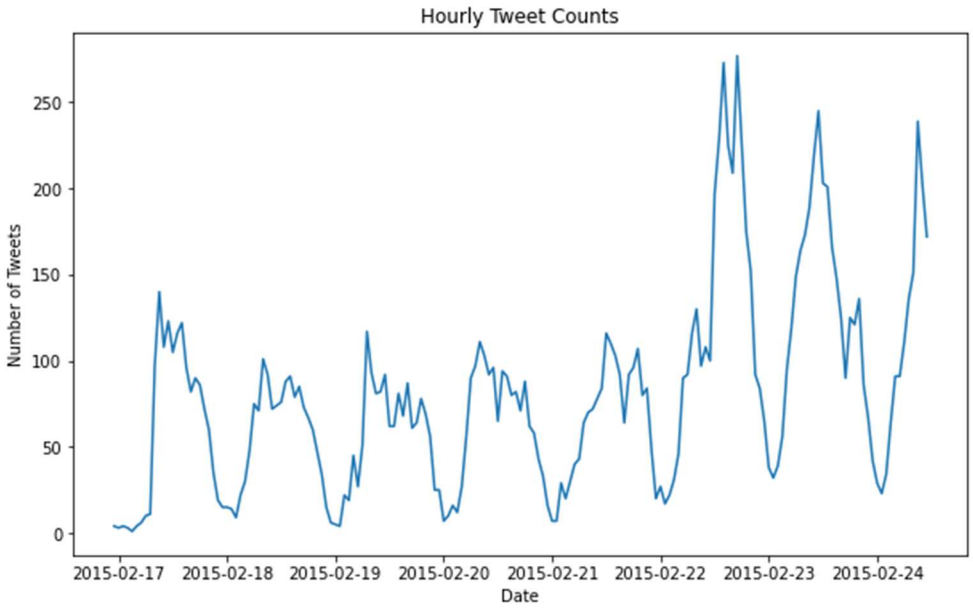
Objectives

The objective of the analysis is to develop a time series model that accurately predicts the frequency of tweets about airline companies and use this model to make forecasts into the future.

Time Series analysis is useful in this situation because it helps to model the patterns and trends in the data over time, allowing us to make forecasts into the future. By modelling the frequency at which people tweet about flying and airline companies, PlaneSimple can gain insights into trends in airline interest and potentially use this information to make better and informed business decisions around marketing, sales, and customer service.

2.1 Data Preparation for Time Series Analysis

The variable 'tweet_created' which contains the date and time of which the tweet was posted, is transformed into datetime type, resampled to hours, and plotted in a chart to visualize the number of tweets over time.



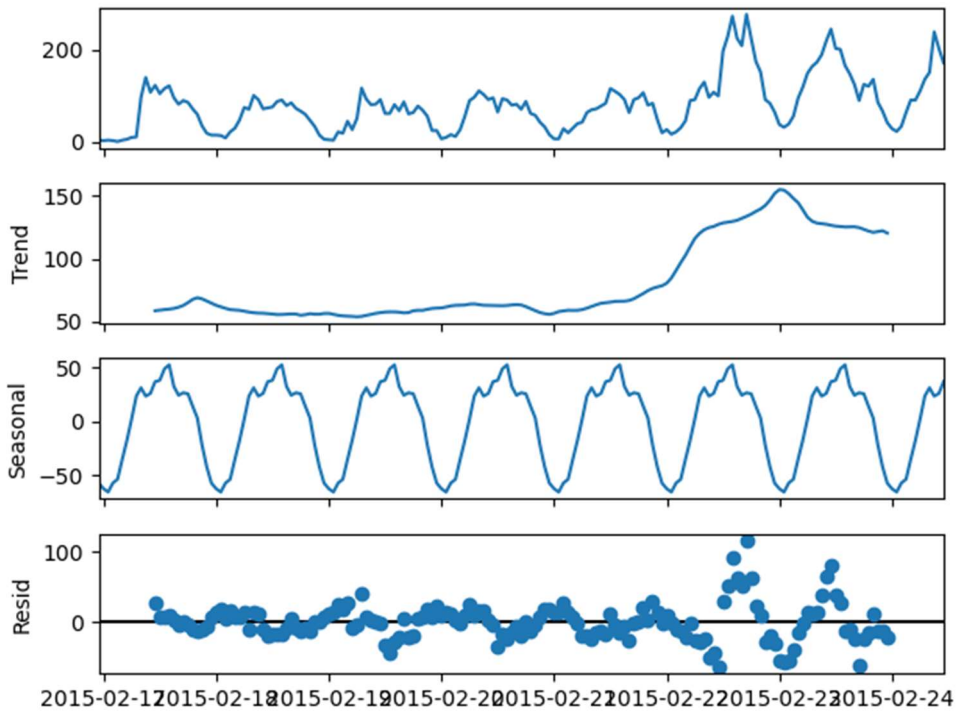
2.2 Modelling

It is important to decompose the time series into its four components, Level, Trend, Seasonality and Noise to observe the change in the patterns of the time series.

- **Level:** It is the main value that goes on average with time.
- **Trend:** The trend is the value that causes increasing or decreasing patterns in a time series.
- **Seasonality:** This is a cyclic event that occurs in time series for a short time and causes the increasing or decreasing patterns for a short time in a time series.
- **Noise:** These are the random variations in the time series.

There are two types of components, additive and multiplicative. It's observed from the chart above that the time series can be treated as additive. "By visualization, we can say the time series is additive if the increasing or decreasing pattern of the time series is similar throughout the series." (Verma, 2021)

Decomposed values



These values are also observed in more details in the technical report (jupyter notebook).

The Augmented Dickey-Fuller test (ADF test)

The Augmented Dickey-Fuller method (ADF test) is used to test if the time series is stationary or not.

According to the research (Verma, 2021). "ADF (Augmented Dickey-Fuller) test is a statistical significance test which means the test will give results in hypothesis tests with null and alternative hypotheses. As a result, we will have a p-value from which we will need to make inferences about the time series, whether it is stationary or not."

Results of ADF test

```
ADF Statistic: -4.983081778618611
p-value: 2.3969325318762712e-05
Critical Values:
        1%: -3.468
        5%: -2.878
        10%: -2.576
```

In this case the p-value is higher than significance level of 0.05, therefore the time series is considered non-stationary.

For non-stationary ARIMA algorithm will be employed to build the machine learning model.

2.3 Searching for Hyperparameters

Autocorrelation, Partial Autocorrelation and Akaike Information Criterion (AIC score) were used to search for best hyperparameter for p, d, and q for the ARIMA model.

The parameters of the ARIMA model are defined as follows:

- p: The number of lag observations included in the model, also called the lag order.
- d: The number of times that the raw observations are differenced, also called the degree of differencing.
- q: The size of the moving average window, also called the order of moving average.

After visualizing the plots and create a list sorted from the lowest AIC score to the highest the hypermeters selected for the ARIMA model were, p=4, d=1, q=3.

2.4 Train / Test split

The data was split into train 80% and test 20%

2.5 ARIMA Model

According to Auhl (2021). "The ARIMA model (an acronym for Auto-Regressive Integrated Moving Average), essentially creates a linear equation which describes and forecasts your time series data. This equation is generated through three separate parts which can be described as:

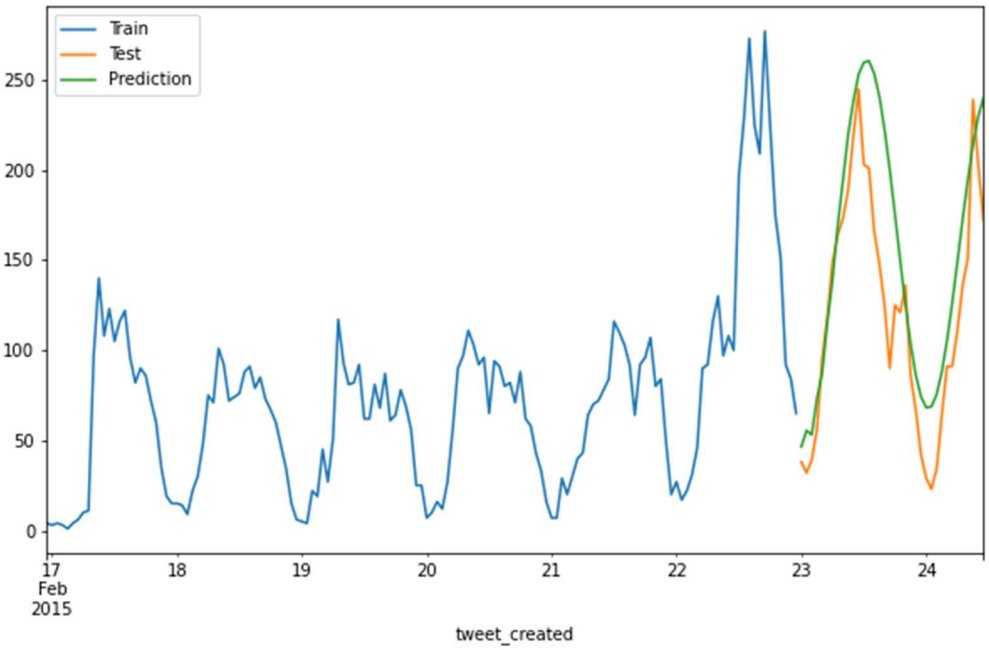
- AR — auto-regression: equation terms created based on past data points.
- I — integration or differencing: accounting for overall "trend" in the data
- MA — moving average: equation terms of error or noise based on past data points.

Together, these three parts make up the AR-I-MA model."

2.6 ARIMA Model Evaluation

Mean squared error was used to evaluate the model as a metric and a chart was plotted to visualize the prediction vs actual values and it was observed that the model performed well.

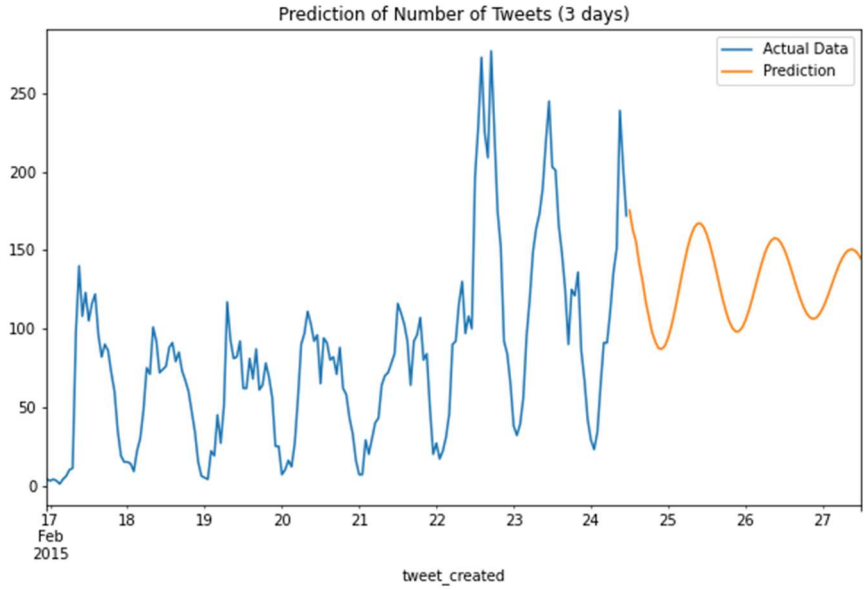
The green line in the chart represents the predicted values while the orange line are the actual values.

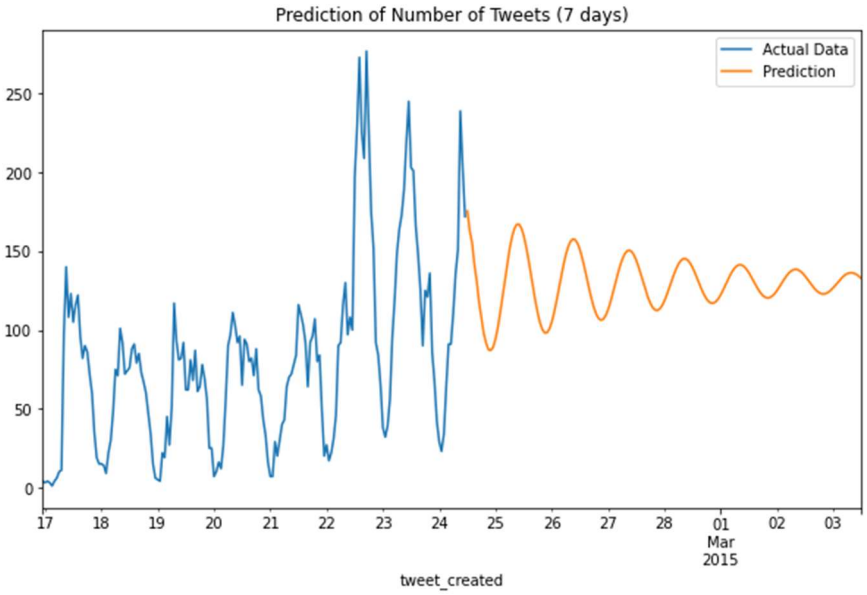


2.7 Forecast using ARIMA

Using the ARIMA trained model a forecast was created for the next 3 and 7 days and the values were plotted in a chart for evaluation.

The ARIMA model seems to be performing better when predicting on a shorter period (3 vs 7 days) and it might have happened because of the seasonality component. In this case the SARIMA model that considers the Seasonality factor will be employed to compare the results.





2.8 SARIMA Hyperparameters

The SARIMA model adds the S to the ARIMA model which stands for Seasonality and therefore can be much more powerful than the ARIMA model when it comes to predicting seasonal values. This model will add specific autoregressive, differencing and moving average terms to the seasonal part of the time series and these are called P, D and Q.

AIC score was also used to search for hyperparameter:

p=2, d=0, q=2, P=0, D=2, Q=2

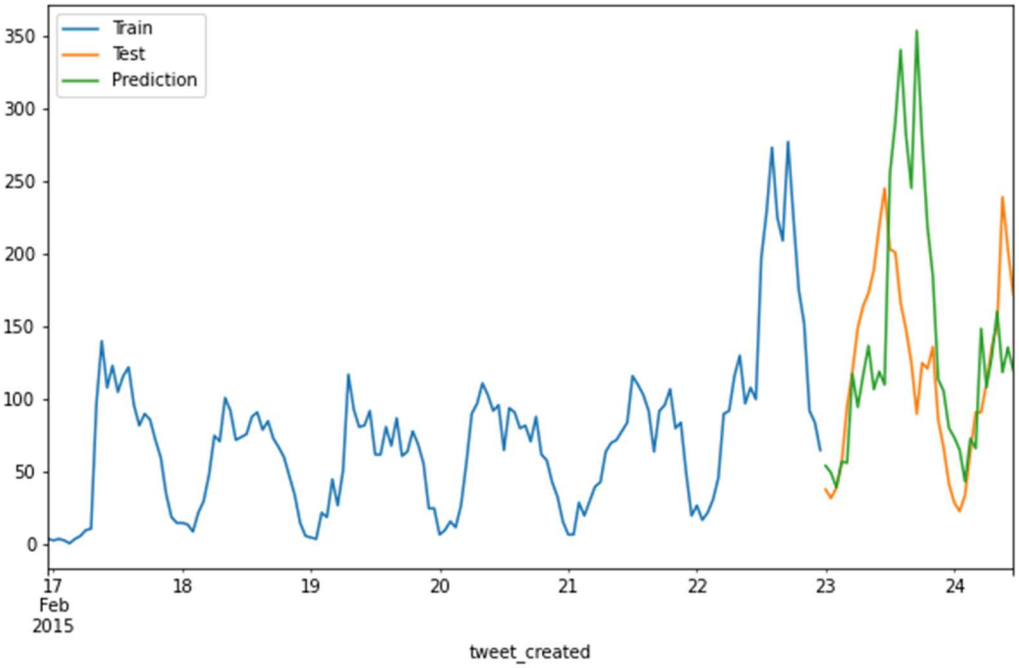
2.9 SARIMA Model

At this stage the model is trained using the same train / test split from ARIMA model.

2.9.1 SARIMA Model Evaluation

Mean squared error and visuals were used for evaluation.

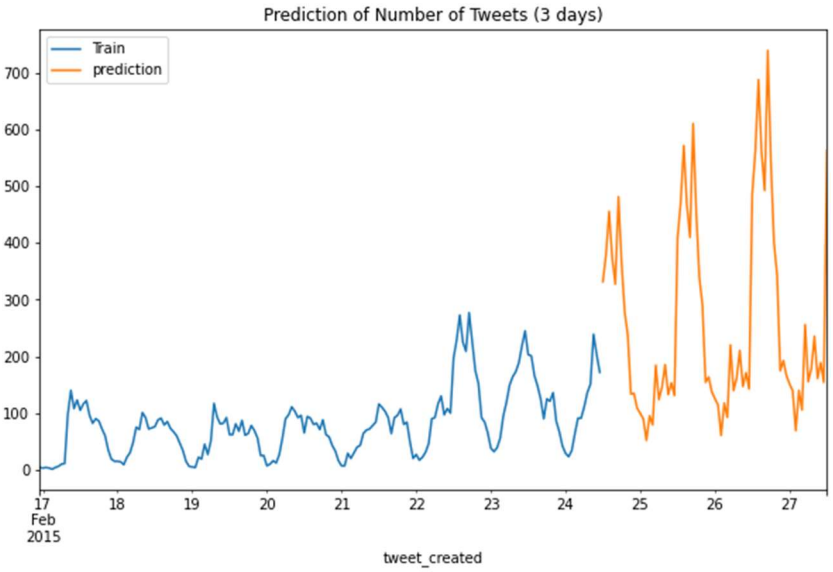
Observing the graph, it seems that the SARIMA model is not performing well capturing an upward trend that is higher than real data points.

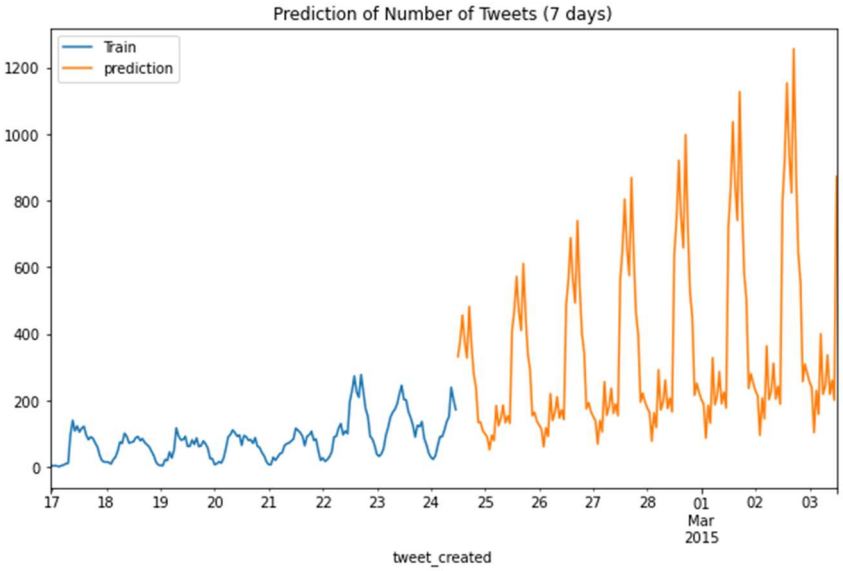


2.9.2 Forecast using SARIMA

Observing the graphs apparently there is an upward trend being captured in the forecast.

Note: I explored different methods to remove the trend and tested different hyperparameters for SARIMA, the AIC model is taking more than 1 hour to run so I will keep the range between (0, 3), however there should be room for improvements if the range increases. I was not able to find a solution for removing the trend, the best results appeared when I searched for hyperparameters using the stationary data for the SARIMA model which seems has reduced the trend but not completely removed it. I will keep this on the radar as a next step to further optimize the model and improve predictions.





2.9.3 Results, Conclusions and Limitations

Both models ARIMA and SARIMA will be considered and presented as two different scenarios, ARIMA is predicting values within a lower range and therefore will be called the conservative scenario and SARIMA is capturing an upward trend in its prediction which is hard to confirm with the limitations of the data sample (only one week of data). To confirm if there's an overall upward trend in the number of tweets per day, more data (time) would be needed. The recommendation is to keep tracking of the data for at least a whole month or quarter and keep working and optimizing both models to be able to predict more accurately.

Section 3: Text Analysis – Analyzing the Tweet Content

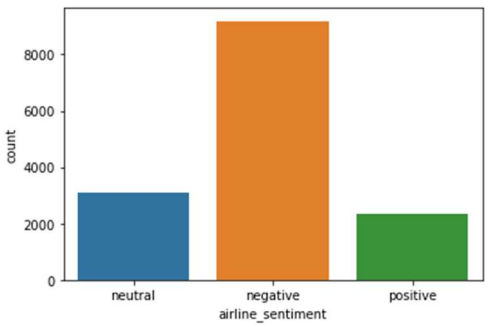
Objectives

The objective of text analysis is to gain deeper insights into the tweet content, understand customer sentiment, identify topics and themes, extract key information, identify emerging trends, and conduct comparative analysis. This information can assist PlaneSimple in understanding customer perceptions, improving their services, addressing customer concerns, and staying informed about the evolving landscape of customer discussions.

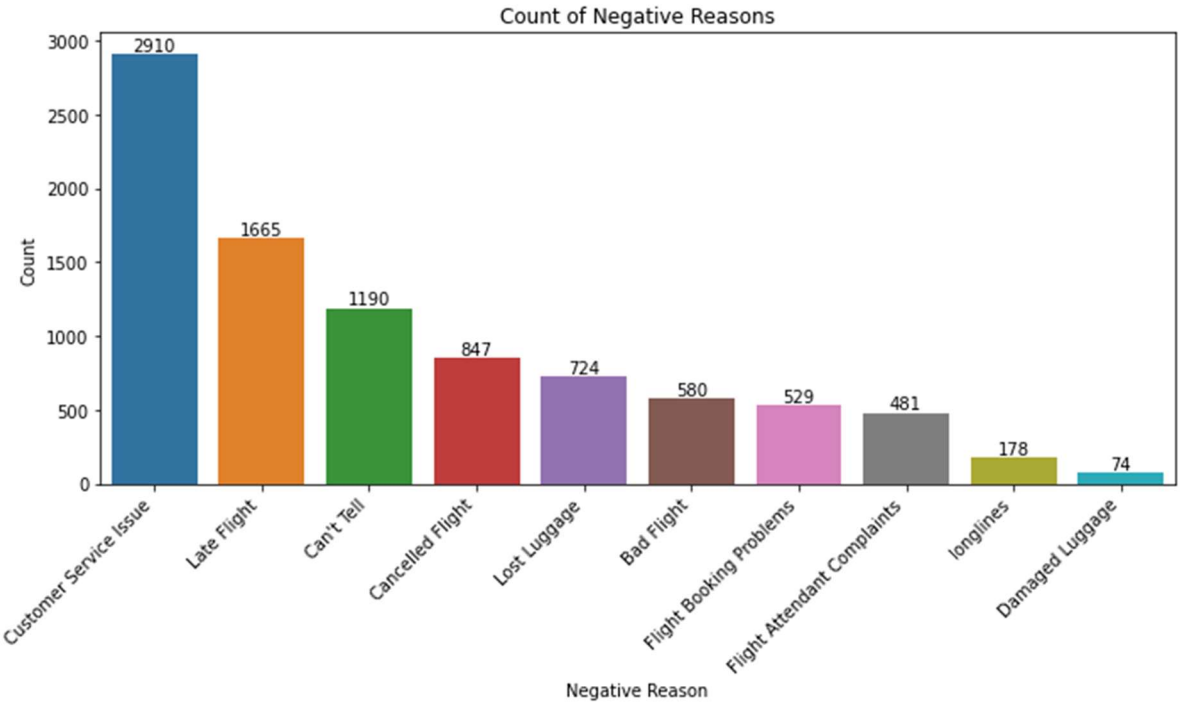
3.1 EDA and Data Prep. for Text Analysis

A supervised machine learning based approach will be employed using the pre-labeled variable "airline_sentiment".

Sentiment distribution - It's observed the variable is unbalanced containing more negative tweets than the other two labels.



It's observed from the chart below that most of the negative tweets are related to Customer Service issues and Late Flight.

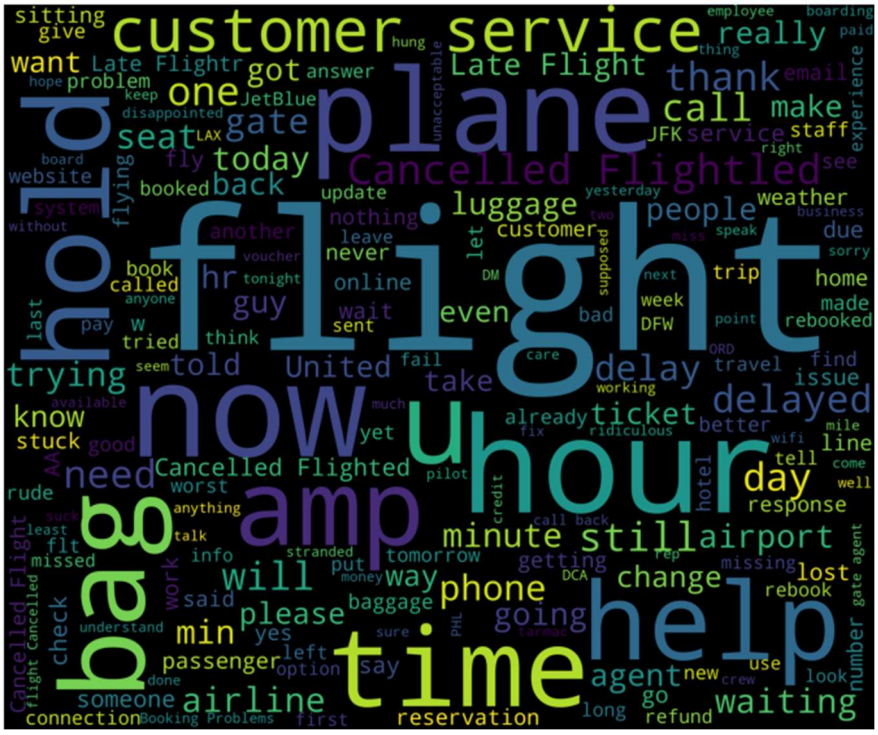


During this stage duplicates were removed, a heatmap and pairplot were plotted to identify feature correlation and exploration was performed using the Wordcloud library to visualize the occurrence of positive and negative words.

The features that will be used for the sentiment analysis will be "text" and "airline_sentiment."

Positive Wordcloud

Negative Wordcloud



3.2 Text Preprocessing - Sentiment Analysis

At this stage stop word and punctuation are removed and words reduced into its roots through stemming.

3.3 Feature Selection

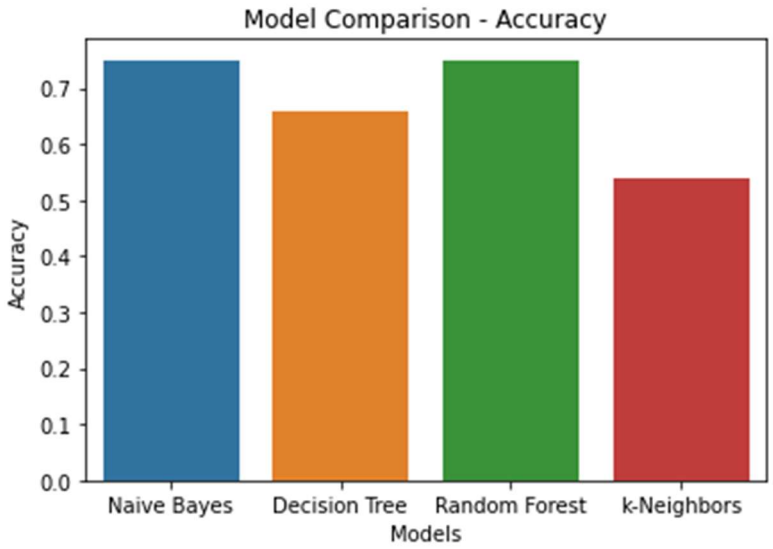
At this stage the features 'text' and 'airline_sentiment' are stored in the X and y variables and a bag of words is created using Count Vectorizer method.

3.4 Sentiment Analysis ML Models

Four models were selected to predict positive, neutral, and negative tweets based on its content.

model1 = Multinomial Naïve Bayes
model2 = Decision Tree Classifier
model3 = Random Forest Classifier
model4 = K-neighbors Classifier

Metrics for Accuracy, Precision, Recall, and f1-score were plotted in conjunction with confusion matrix to evaluate the model's performance. A comparison of accuracy was done and the models that performed best are Naïve Bayes and Radom Forest.

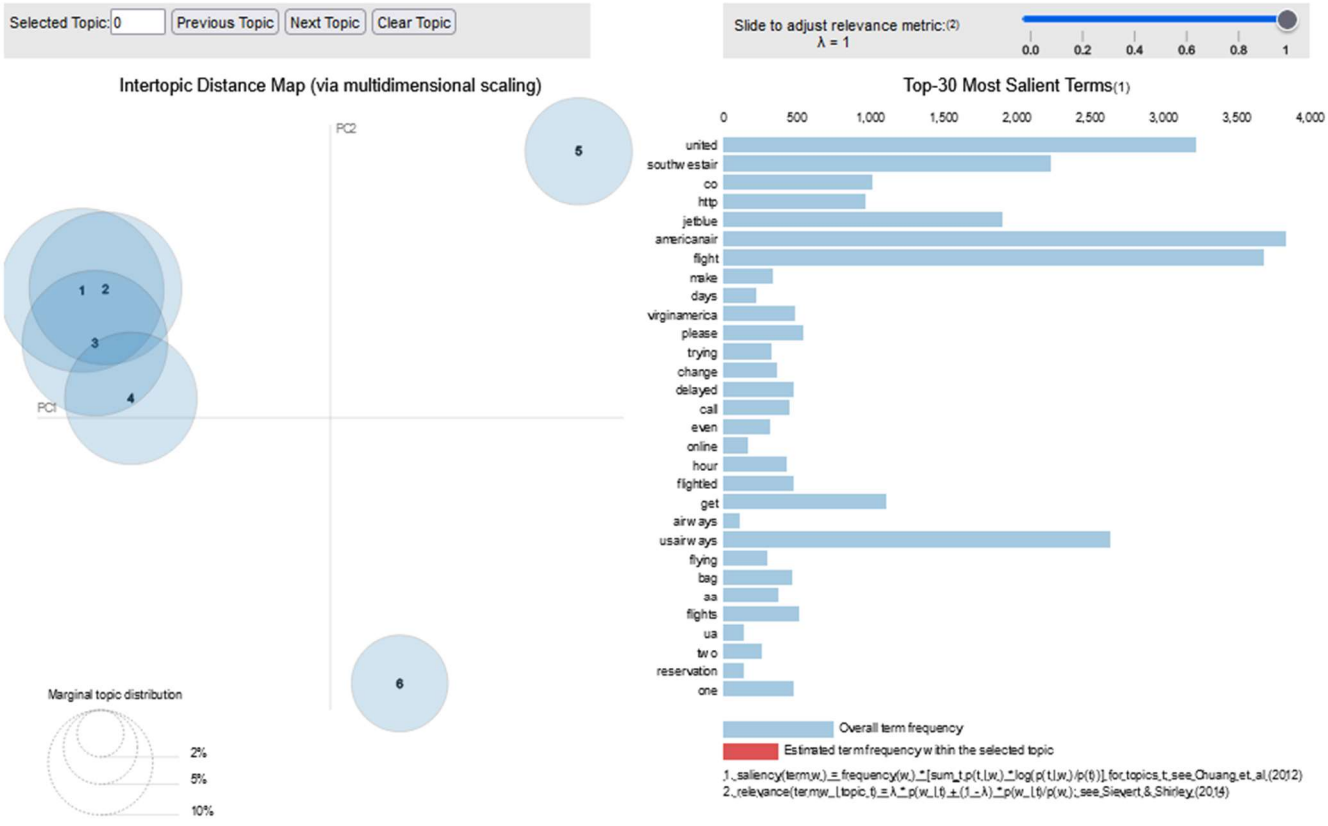


3.5 Text Preprocessing - Topic Modelling

Functions were created to support the text preprocessing before implementing the model, the functions will basically turn sentences into a list of words, run through stop words and remove it, if necessary, create a dictionary of ids for each word and translate the words into their corresponding ids. More details in the technical report (jupyter notebook).

3.6 LDA Results Visualization

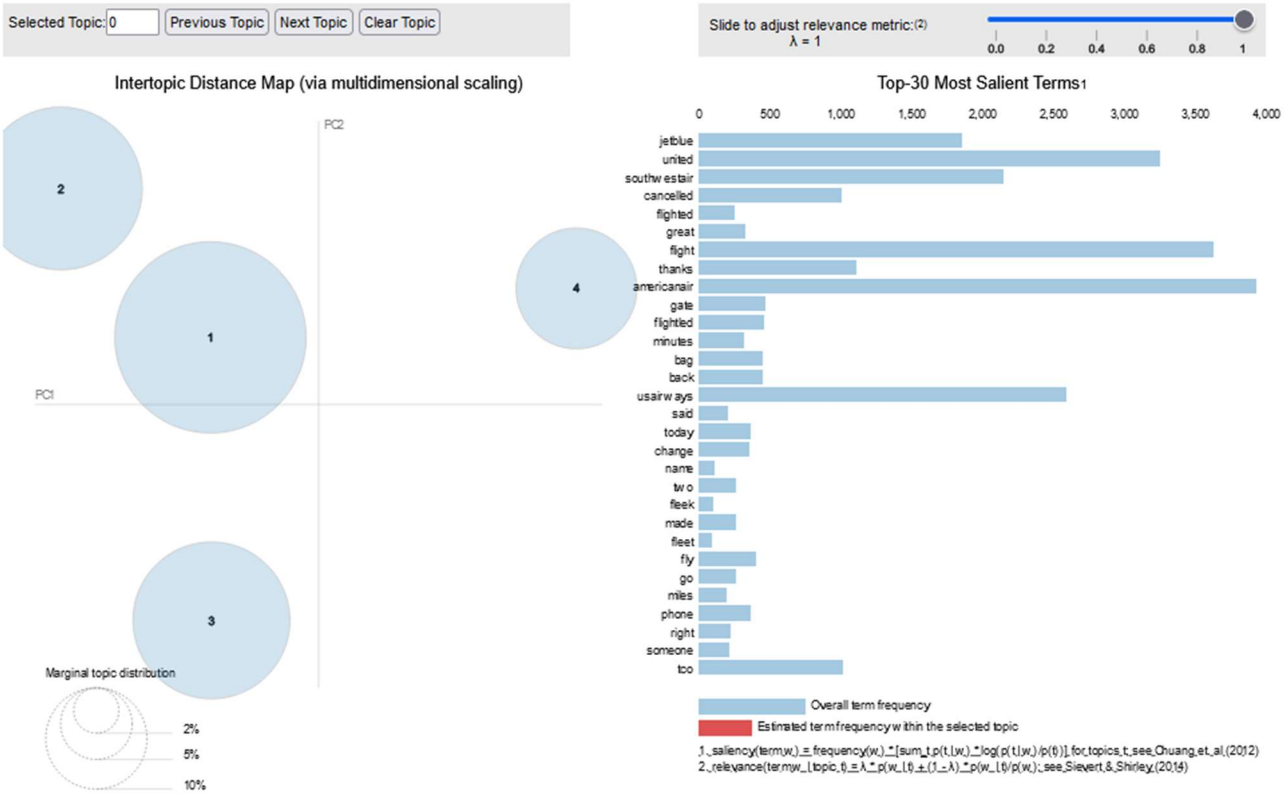
Latent Dirichlet Allocation (LDA) method is used to create topics based on the words from tweets. These topics are clusters of words that the model identifies as related. As there are six airline companies mentioned in the tweets, six topics were selected to continue the exploration. In the "Intertopic Distance Map" we can see that some of the topics overlap so the number of topics will be reduced after optimizing the model. This visual is powerful when it comes to visualize the topics and occurrences of words contained in each.



3.7 Optimizing LDA Model

After analyzing the results from the LDA visualization more words to be removed were identified and added to the stop_words function. Tittles were converted to lowercase and the number of topics was reduced to 4. After optimization a new general WordCloud and LDA was visualized.

After optimization the model was able to better distinguish the topics as seen in the chart.



3.8 Results, Conclusions and Limitations

Four different machine learning models were employed for sentiment analysis and two of them (Naive Bayes and Random Forest) performed relatively well with good accuracy, however with room for improvements. As a next step I will look further into pre-processing the text by removing more of the stop words and punctuations and even employ LDA (Latent Dirichlet Allocation) before passing the data through the machine learning models for sentiment analysis to improve accuracy.

LDA (Latent Dirichlet Allocation) was employed and able to distinguish 4 different topics across the tweets, (which can be visualized in the "Intertopic Distance Map"). However, the data is very limited on this end as there is not much variation across the topics which basically approaches the customers opinions about flights and airline companies, making it more difficult to distinguish across different topics.

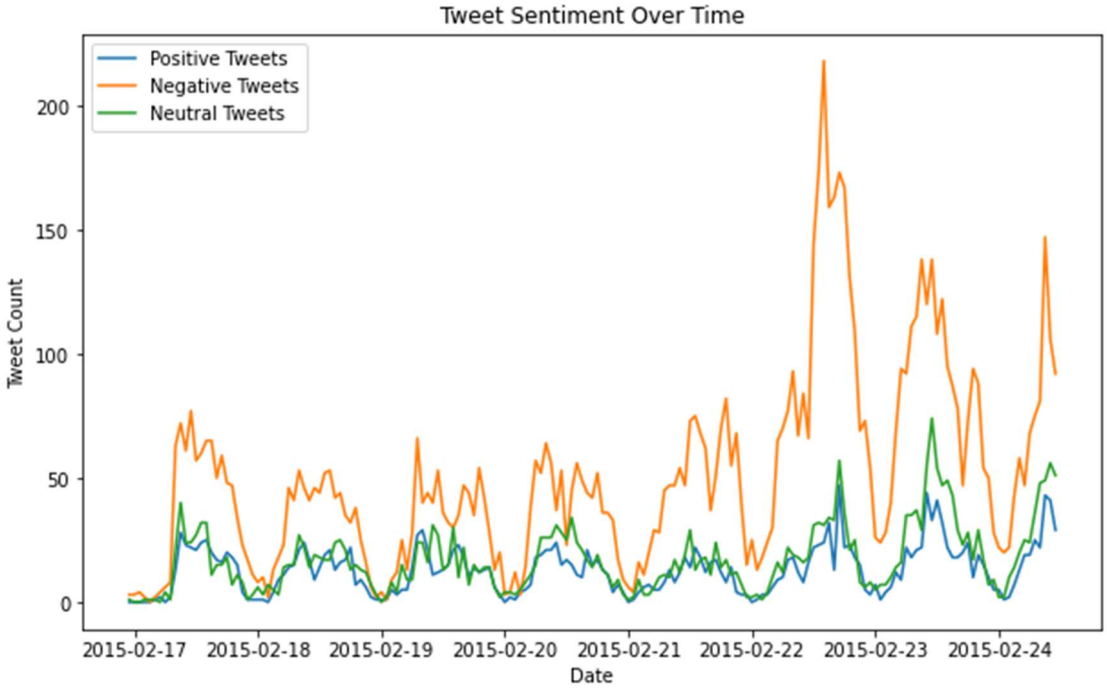
Section 4: Combine both models (Time-Series and Text Analysis)

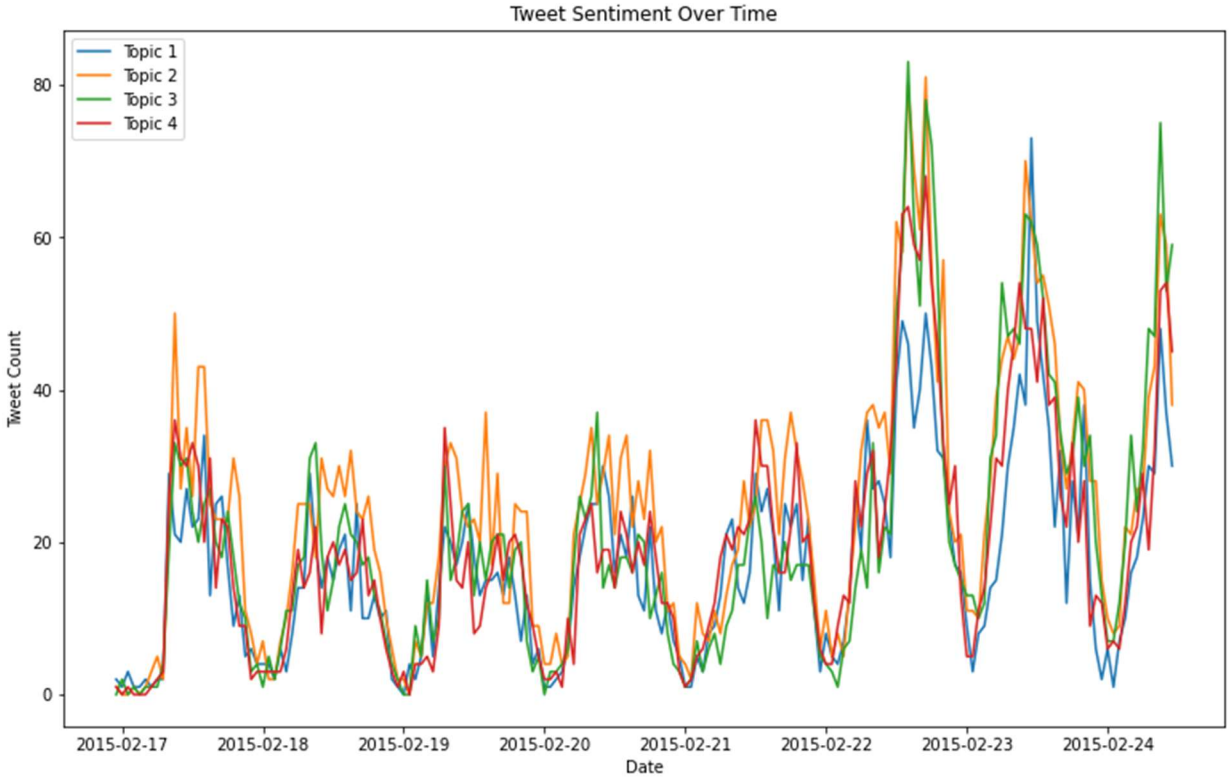
Objectives

The goal is to train appropriate time series models to forecast positive tweet sentiment and an appropriate topic related to the subject.

4.1 Exploration

At this stage charts were plotted to visualize the count of tweets per sentiment and topic.

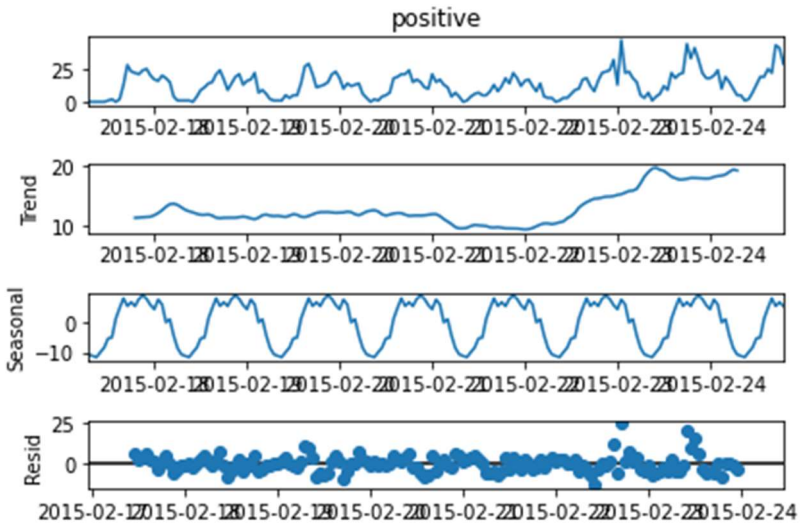

Tweet Sentiment Over Time
Positive Tweets
Negative Tweets
Neutral Tweets
Tweet Count
0
50
100
150
200
2015-02-17
2015-02-18
2015-02-19
2015-02-20
2015-02-21
2015-02-22
2015-02-23
2015-02-24
Date


Tweet Sentiment Over Time
Topic 1
Topic 2
Topic 3
Topic 4
Tweet Count
0
20
40
60
80
2015-02-17
2015-02-18
2015-02-19
2015-02-20
2015-02-21
2015-02-22
2015-02-23
2015-02-24
Date

4.2 Positive tweets prediction into the future - ARIMA Model

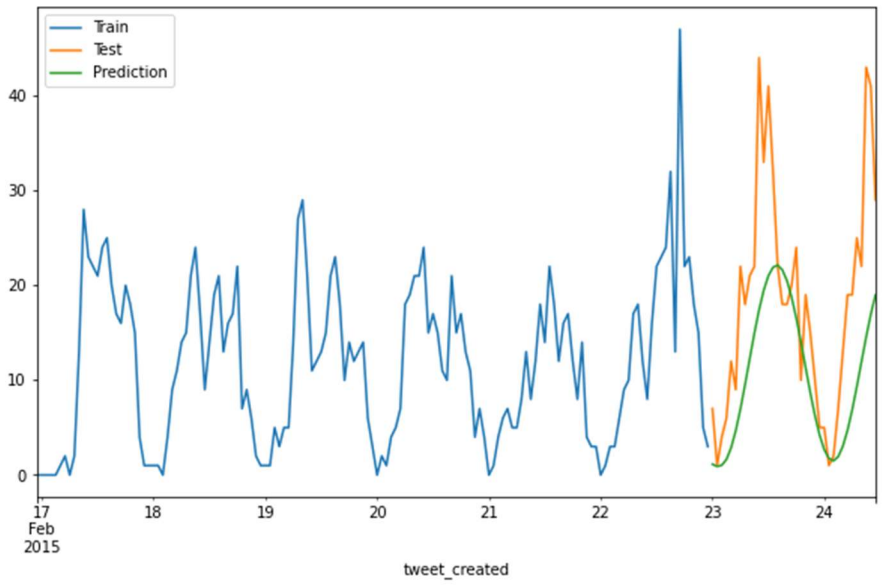
Only positive tweets were stored in a variable and decomposed into Level, Trend, Seasonal and Residual. AIC score was used for hyperparameters search.

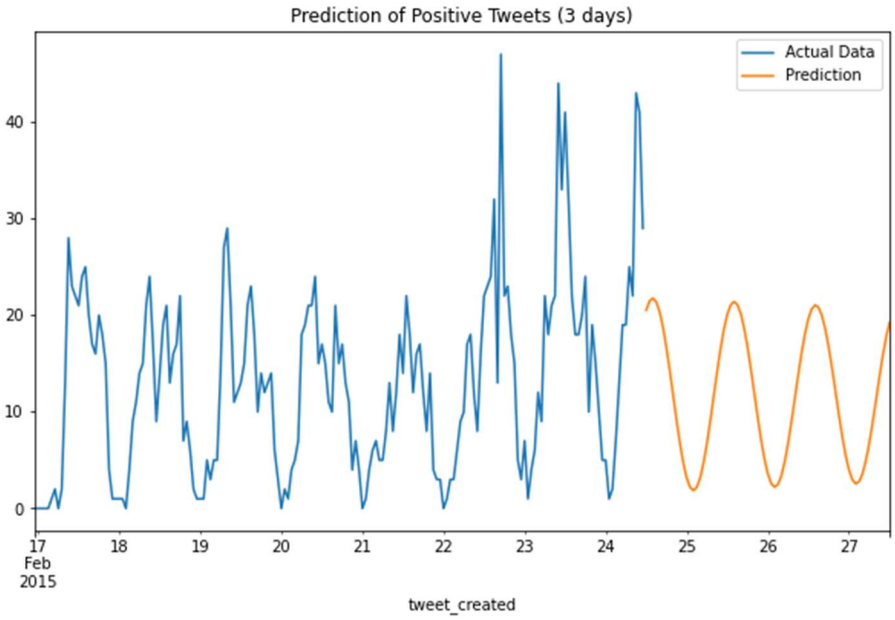
Decomposed values



A list of lowest AIC scores was plotted and based on trial and error testing the best hyperparameter found for p, d, and q are p=3, d=0, q=4.

Train / Test split was performed as 80% / 20%

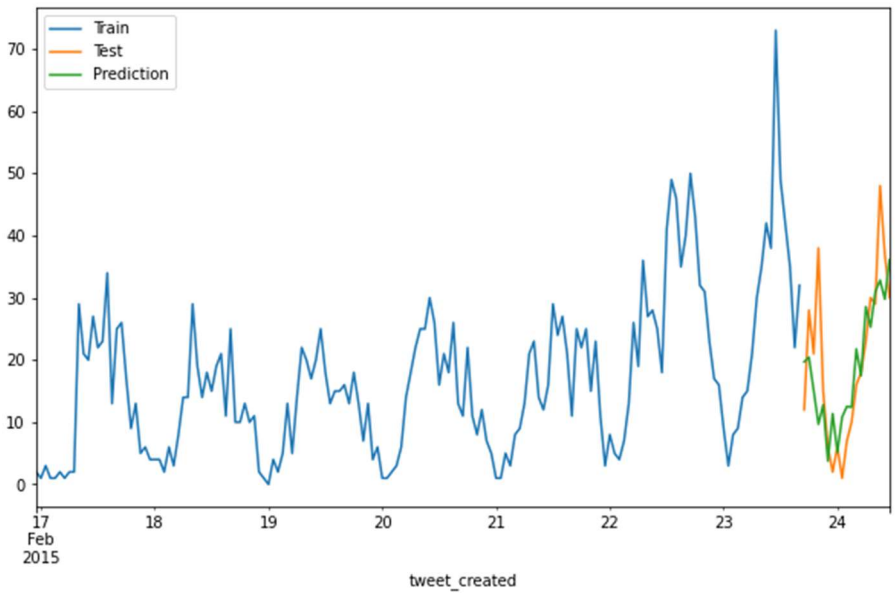


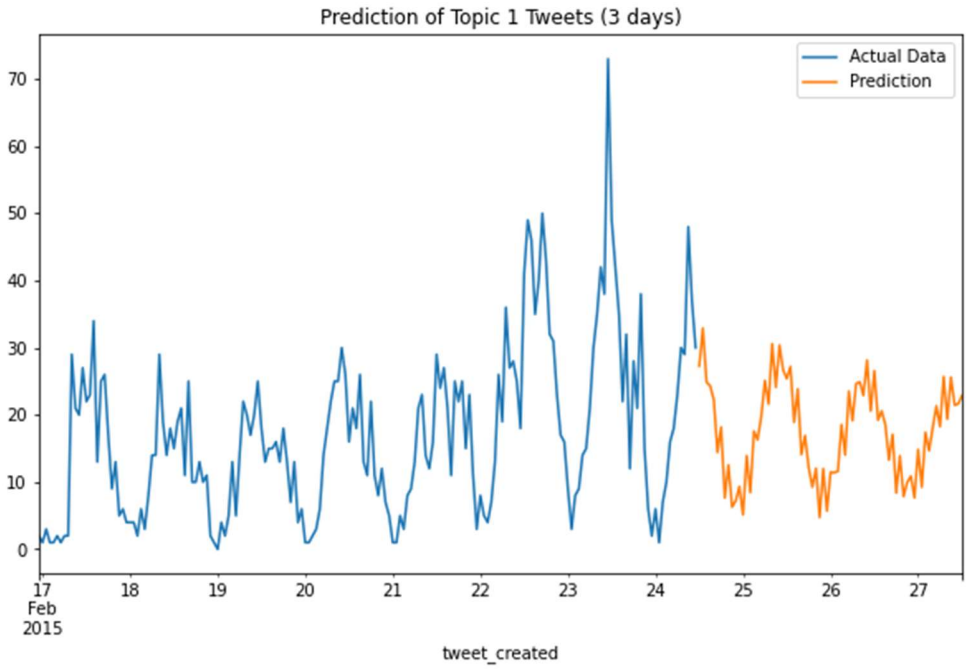


4.3 Topic 1 prediction into the future - ARIMA Model

Same process of decomposition and hyperparameters search was performed and the best parameters found were p=5, d=0, q=6.

Train / Test split was increased for training to 90% as the model was failing to predict the peak that occurs in the last days of the sample and including more data in the train model improved performance.





4.4 Results, Conclusions and Limitations

The machine learning algorithm ARIMA was employed to predict the number of positive tweets and the number of tweets contained in the Topic 1 defined during the Topic Modelling stage.

For the positive tweet prediction, the model failed to predict the spike in the number of tweets that happens by the end of the week, however, with the limitation of the data, the spike cannot be confirmed as a trend or seasonality and therefore we can say the model is performing considerably well within its capacity and information available.

For topic prediction a more robust model was employed searching for a larger range of hyperparameters and increasing the train split to 90% which showed improvements on the model's performance.

Again, in order to improve the models, it is recommended that future tweets are tracked and recorded in a data set with at least a Month or a Quarter of tweets so the models can rely on more data to learn and make better predictions, which will ultimately help the business make better and informed decisions.

BIBLIOGRAPHY

Auhl, M. (2021). What is an ARIMA Model? [online] Medium. Available at: https://towardsdatascience.com/what-is-an-arima-model-9e200f06f9eb.

Brownlee, J. (2018). How to Grid Search SARIMA Hyperparameters for Time Series Forecasting. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/how-to-grid-search-sarima-model-hyperparameters-for-time-series-forecasting-in-python/.

Nachiketa Hebbar (2020). Twitter Sentiment Analysis Using Python for Complete Beginners. [online] Medium. Available at: https://medium.com/swlh/tweet-sentiment-analysis-using-python-for-complete-beginners-4aeb4456040.

Verma, Y. (2021). Why Decompose a Time Series, and How? [online] Analytics India Magazine. Available at: https://analyticsindiamag.com/why-decompose-a-time-series-and-how/ [Accessed 14 May 2023].

Verma, Y. (2021). Complete Guide To Dickey-Fuller Test In Time-Series Analysis. [online] Analytics India Magazine. Available at: https://analyticsindiamag.com/complete-guide-to-dickey-fuller-test-in-time-series-analysis/.