**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* |
| **Assessment Title:** | *MSC\_DA\_CA2v4* |
| **Lecturer Name:** | *David McQuaid*  *Muhammad Iqbal* |
| **Student Full Name:** | Rita Raher |
| **Student Number:** | sbs22115 |
| **Assessment Due Date:** | 11/11/2022 |
| **Date of Submission:** | 06/11/2022 |

Icon

Description automatically generated with medium confidence

**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. |

Table of Contents

[Introduction 2](#_Toc118614658)

[Methodology 2](#_Toc118614659)

[Comparative Analysis 3](#_Toc118614660)

[Runtime 3](#_Toc118614661)

[Throughput 3](#_Toc118614662)

[Insert 3](#_Toc118614663)

[Data 4](#_Toc118614664)

[Data Storage and Processing 4](#_Toc118614665)

[Big Data 5](#_Toc118614666)

[Data Preparation 5](#_Toc118614667)

[Pre-processing 5](#_Toc118614668)

[Feature engineering 6](#_Toc118614669)

[Exploratory Data Analysis 6](#_Toc118614670)

[Topic Modelling 7](#_Toc118614671)

[Sentiment analysis over time 7](#_Toc118614672)

[Leader Sentiment Analysis 7](#_Toc118614673)

[Test for Stationarity 8](#_Toc118614674)

[Autocorrelation plots 9](#_Toc118614675)

[Forecasting Sentiment 10](#_Toc118614676)

[Evaluation 10](#_Toc118614677)

[Deployment 12](#_Toc118614678)

[Conclusion 13](#_Toc118614679)

[Bibliography 13](#_Toc118614680)

## Introduction

This first section compares two databases, MySQL and MongoDB, using the Yahoo! Cloud Serving Benchmark (YCSB) framework. It ran benchmarking tests on both to gauge which had the best performance. MongoDB was the resulting winner; it is twice as fast as MySQL. The second part of the project performed a sentiment analysis on tweets related to the UK government to understand public opinion on the current UK government party. This project used big-data architecture for storage, processing and part of the analysis. The sentiment of tweets related to the UK government was analyzed to identify trends and seasonality. A time series forecasting model using Skforecast with an XGBRegressor was able to accurately predict sentiment for the next 7, 30, and 60 days.

## Methodology

This project will follow the CRISP-DM methodology, jointly developed in 1996 by DiamlerChrysler, SPSS and NCR (Azevedo and Santos, 2008). The framework steps include Business understanding, Data Understanding, Data Preparation, Modelling, Evaluation and Deployment.

## Comparative Analysis

This section is a comparative analysis between MySQL and MongoDB. MySQL is an open-source, free-to-use relational database, which developed by Oracle in 1994. It requires a schema definition and is a good choice for a traditional database is needed. MongoDB, developed in 2007, is a schema-less NoSQL database that stores data in documents. It goal was to replace the MySQL structured database and to be more user-friendly. It was designed with big data in mind; therefore, it has a highly scalable query-ability and provides faster access to data. MongoDB is a good choice for unstructured data.

The YCSB framework was developed in 2010 to help facilitate performance comparisons between cloud databases (Cooper et al., 2010). Therefore to help with this comparison section, the YCSB library will perform benchmark tests on both MySQL and MongoDB. The results will be gathered for both databases based on operation counts of 1000, 10000, 10000, 100000, 1000000, 10000000.

### Runtime

When looking at the runtime performance, MongoDB performed the best. Figure 1 shows MongoDB is more than 2 times faster than MySQL

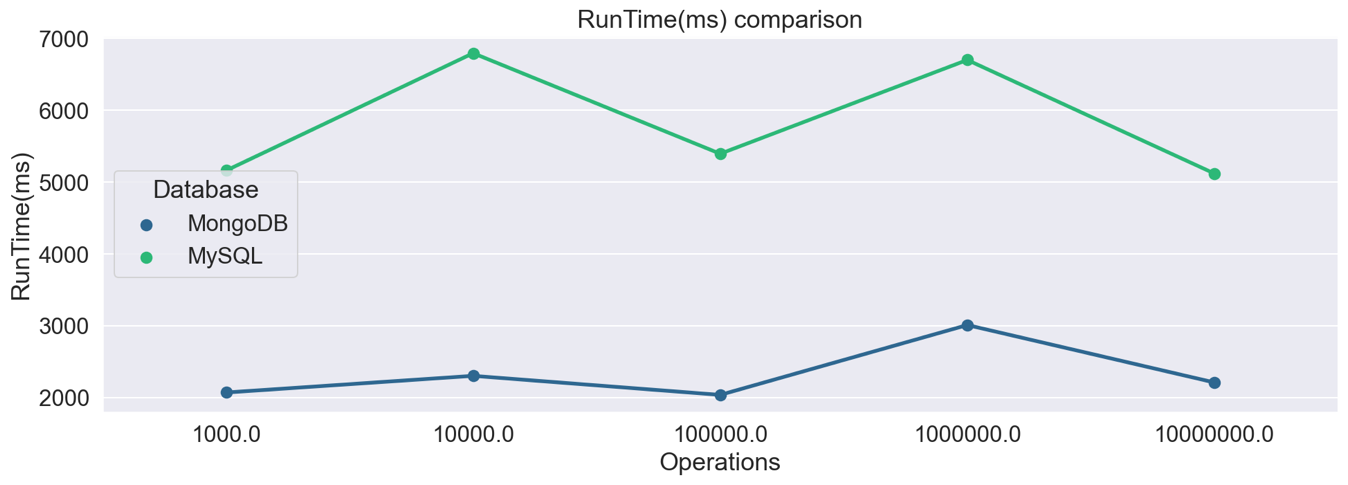


Figure 1: RunTime(ms) comparison

### Throughput

Throughput measures how many units of information a system can process within a certain amount of time. When analysing both databases’ throughputs (ops/sec), MongoDB could process more than double the information that MySQL could.

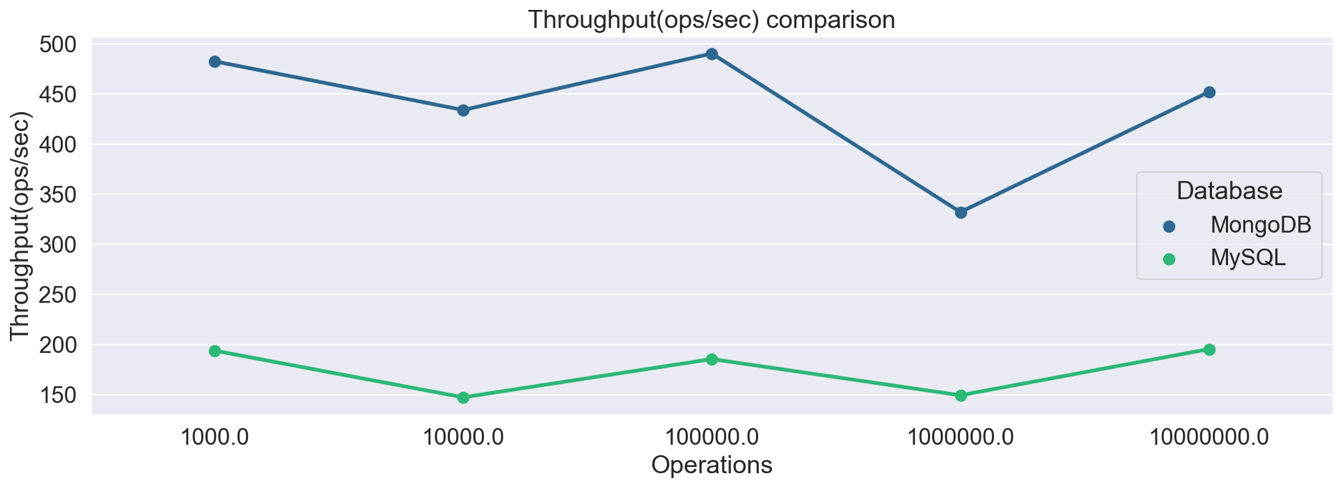


Figure 2: Throughput

### Insert

The benchmark test for insert looks at how long it takes to insert into both databases. Examining the databases side by side for the 95th and 99th percentage latency for insert on 10,000 operations, it is clear that MongoDB performs best.

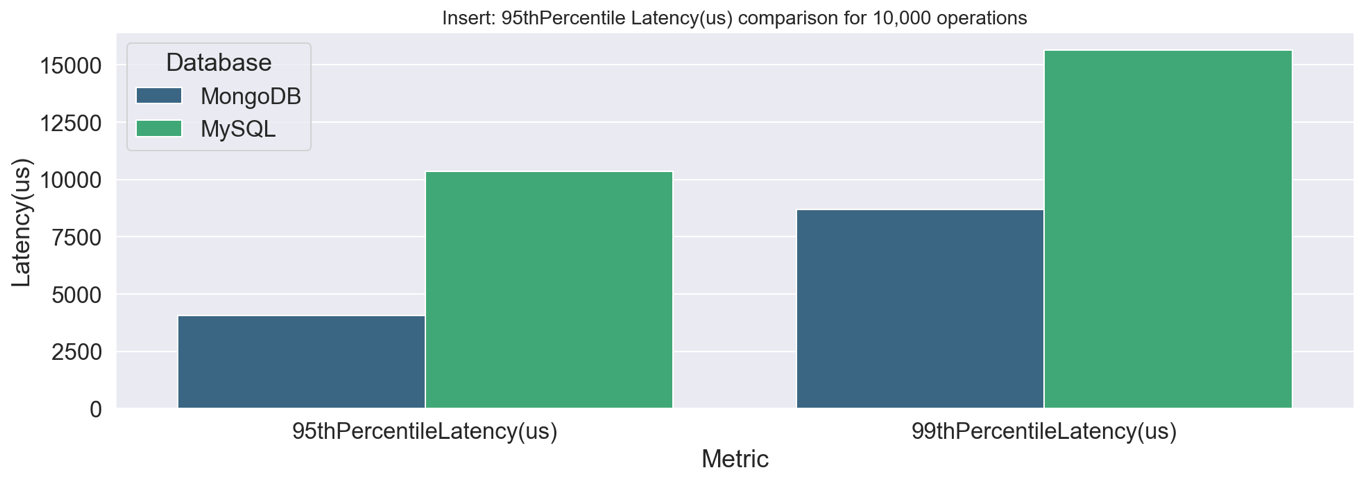


Figure 3: Insert comparison 95th and 99th Percentile

**Conclusion**

The overall winner when it comes to speed is MongoDB. MongoDB would be an excellent choice if the infrastructure needs to support big data, real-time analytics and a large number of queries on the data.

## Data

The Twitter data for this project was extracted using SNScrape. SNScrape is a scraping tool for social networking services (SNS). It allows developers to extract tweets without requesting developer access from Twitter which can take some time. For this project, Twitter developer access was explored, however, it only returned seven days of tweets, and this project required a year’s worth of tweet data for sentiment analysis. SNScrape also supports the scraping of other social media sites such as Facebook, Instagram, Reddit and more. It allows the scraping of date, id, username, tweet content, like count, retweet count, reply count and more.

Using SNScrape services, tweets found with the keyword ‘ukgov’ were extracted between 23 Oct 2021 – 23 Oct 2022. 50 tweets per day were extracted and added to the dataframe. The tweet dataframe with the following info: Datetime, Tweet id, Text, Username, reweetCount, likecount, and hashtags were created.

## Data Storage and Processing

The tweets dataframe was exported to a csv, the csv was then imported into a virtual machine to perform big data operations. A MySQL database, benchtest, was created to hold the tweets and a new table was created:

CREATE TABLE `uk\_tweets\_likes2` (

`id` int DEFAULT NULL,

`Datetime` text,

`Tweet ID` text,

`Text` text,

`Username` text,

`retweetCount` text,

`likeCount` text,

`hashtags` text

);

The tweets were loaded into the MySQL table, uk\_tweets\_likes2:

LOAD DATA local INFILE '/home/hduser/Downloads/ukgov\_new.csv' INTO TABLE uk\_tweets FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\r\n' IGNORE 1 LINES;

Using PySpark in a Jupyter notebook, a connection was made to the database using mysql-connector-python, a MSQL driver. Data was pulled from the MySQL database into PySpark for some part of the analysis.

Diagram

Description automatically generated

Figure 4: Project Architecture

MongoDB and MySQL are two of the most in-demand databases for web applications. The decision to go with MySQL was down to the fact that it is still one of the most in demand used Databases in the world. According to a 2022 survey by StackOverflow, 46% of professional developers use MySQL, whereas only 28% use MongoDB. (Stack Overflow, 2022).

## Big Data

## Data Preparation

### Pre-processing

The NLTK library was used to remove stopwords from the tweets. Stopwords are words which do not express any meaning and hence are just noise in the dataset. Preprocessing is conducted on the tweets to remove stopwords, convert them to lowercase, remove numbers, remove Twitter handles and remove punctuations. After preprocessing, the tweets are ready for sentiment analysis.Diagram

Description automatically generated

Figure 5: Preprocessing of tweets before sentiment analysis

### Feature engineering

From the datatime column, additional columns are feature engineered to create columns for ‘Date’, ‘Month’, ‘Year’, ‘Week’, ‘day\_of\_year’, ‘day\_of\_week’, ‘quarter’, ‘season’. Creating these additional columns will help to compare sentiment over these various time ranges. The frequency of the observations in the dataset is daily, ranging from Oct 2021 – Oct 2022. Additional features are created for word count and character count of the tweets.

Two well-known libraries can be utilised to create a sentiment score: Textblob and Vader. TextBlob is a library for processing textual data. It has a sentiment analyser which returns two properties when analysing text: Polarity and Subjectivity. Polarity returns a score between -1.0 and 1.0. 1.0 being positive and -1.0 being negative. Subjectivity is a score between 0.0 and 1.0, representing the amount of opinion vs factual information in a given text. The higher the subjectivity, the more the text contains personal opinions.

Vader, Valence Aware Dictionary and Sentiment Reasoner, was developed specifically for social media and, therefore may be more tuned to the sentiments expressed on Twitter (Hutto and Gilbert, 2014). Vader returns a dictionary of 4 scores: Positive, Neutral, Negative and Compound, all of which range between -1 and 1. Vader takes into account the word-order sensitive relationships which are normally lost in the cases of a bag-of-words approach, an approach which the TextBlob follows (Hutto and Gilbert, 2014). The compound score from Vader will be utilised for tracking the overall sentiment of a tweet.

## Exploratory Data Analysis

The dataset contains 18,481 tweets which an average word count of 30, with an average of one reweet and five likes. The tweets were extracted from Twitter based on the keyword ‘ukgov’. This keyword targeted tweets discussing the UK Government. While exploring the dataset, it was discovered that there were an unseemly high number of neutral sentiment tweets, almost 4,000, which affected the median value in all the preceding data analysis, see Figure 6. The analysis looks to positive and negative sentiment, and therefore it is decided to remove any tweets with a compound value of 0, see Figure 7. The data seems to be a more balance amount of negative and positive sentiment.

Chart, histogram

Description automatically generated

Figure 6: Before removing tweets with a compound of 0

Chart, histogram

Description automatically generated

Figure 7: After removing

### Topic Modelling

Using Genism, an Natural Language Processing Library, a Latent Dirichlet Allocation (LDA) model was used to explore tweets. The tweets were cleaned further, removing nouns and adjectives. Three main topics were found: Brexit, Football and Scotland.

### Sentiment analysis over time

Boxplots are used to analyse the compound over various periods of time: Year, Season, Quarter and Month. Compared to 2021, there has been a slight decrease in sentiment. Figure 8 shows Q4 2021 with the highest sentiment which has not regained ground in the quarters following. October 2021 has the most positive sentiment which a dramatic drop in the following three months. Months with negative sentiment are May, August and September 2022. In May 2022, the Conservative Party suffered electoral losses at a local level, due to the cost-of-living crisis and the sentiment reflects that with a dramatic drop.

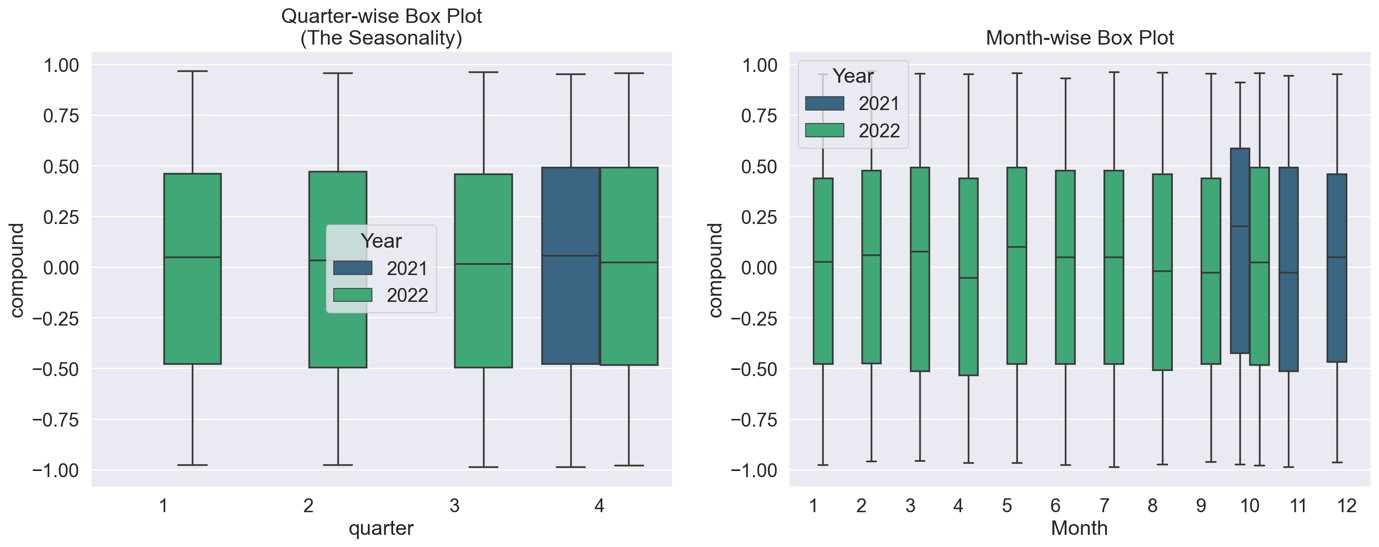


Figure 8: Quarterly and Monthly Box plot

### Leader Sentiment Analysis

Observing tweets related to the current and previous Prime Ministers/Leaders of the Conservative Party (also known as the Tory Party). Boris Johnson has a very volatile sentiment score. He held a very positive sentiment score in October 2021, which dropped dramatically in November and December 2021. Sentiment rose again in March and May 2022 and seems to be increasing again around September and October 2022, around the time of political upheaval in his party concerning the budget brought in by his successor Liz Truss and speculation around him making a comeback.

When analysing Liz Truss’s sentiment score over the past year, the median value for each month hits mostly negative and neutral sentiment, which shows she is not that popular. In the last two months of her reign, that median value for September and October was 0, which correlates with the public opinion on her budget.

Rishi Sunak’s sentiment score is at an all-time high for October 2022, when he took over as Prime Minister from Liz Truss. Analysing the previous year, his sentiment score has been turbulent. However, the median value of sentiments in the tweets related to him for February, April, May, July and Oct of 2022 managed to achieve a sentiment score greater than 0.25. He resigned from the cabinet as Finance Minister in July 2022, which saw his sentiment soar.

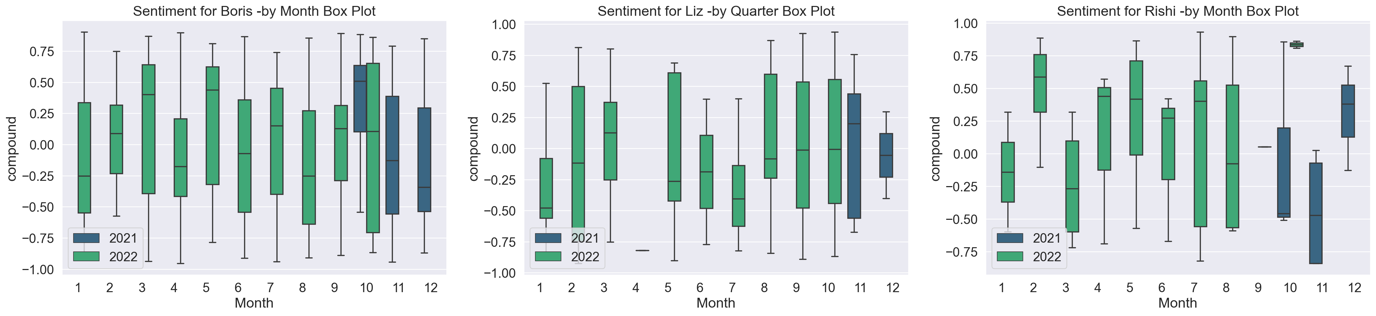


Figure 9: Comparing Party Leader sentiment over time

A linear regression line is drawn to investigate whether sentiment is increasing or decreasing over time, and it shows that only Rishi Sunak’s sentiment is rising, see Figure 10. Looking at the evidence above, it is clear that neither Truss or Sunak hold the popularity of Boris Johnson.

Chart, scatter chart

Description automatically generated

Figure 10: Leader’s sentiment with linear regression line

### Test for Stationarity

The data doesn’t appear to have any trends or seasonality and seems stationary. Timeseries data with trends generally see an increase or decrease. For seasonality, the data shows a recurring pattern at a fixed frequency based on time. This doesn’t seem to be the case with this dataset, see Figure 11.

Timeline

Description automatically generated

Figure 11: Daily Average Sentiment

**Dickey and Fuller Test (Formal Test for Stationarity)**

A standard statistical test, the Augmented Dickey-Fuller test, is performed on the dataset to verify if the time series data is stationary (Mushtaq, 2011). This statistical test will determine whether a Time Series is stationary or not. The Augmented Dickey-Fuller statistics is -88.920752. Therefore the null hypothesis is rejected. We conclude that we have a stationary dataset.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | t-statistics | p-value | Level of significance | Critical value |
| Augmented Dickey-Fuller | -88.920752 | 0.000000 | 1%  5%  10% | -3.431  -2.862  -2.567 |

Figure 12: Augmented Dickey-Fuller Results

### Autocorrelation plots

Using Pearson's correlation helps measure the strength of the linear relationship between two variables (Benesty et al., 2008). The first lag shows a 0.97 correlation which shows a high positive correlation. An excellent way to plot correlation for Time series is using Autocorrelation plots.

Autocorrelation tries to answer whether observations are random and whether a time series can be modelled with Moving Average model. If autocorrelation returns values between 1 or -1 then a strong positive or negative correlation exists.

Chart, histogram

Description automatically generated

Figure 13: Autocorrelation and Partial Autocorrelation

It begins at 1, meaning it is 100% correlated and then drops slowly as it moves through the lags. The presence of this type of correlation is an indication that autoregressive models can work well. Anything within the blue-shaded region, the error bands, represents no significant correlation with the most recent compound value. From the above plot, it is clear that there is no seasonality present (Brownlee, 2017).

The partial autocorrelation plot shows there is a strong correlation at lag 1. The rest of the bars are not statistically significant as they are in the error bands and going towards zero.

**Decompose**

Attempting to break up the time series into its components of trend, season and residual. Observing data over 30 days, there are no notable cyclic trends identified using the seasonal decomposition additive model, see Figure 14.

Timeline

Description automatically generated

Figure 14: Trend, Seasonal, Residuals

## Forecasting Sentiment

Time series forecasting makes use of recursive autoregressive forecasters. Recursive models use previous predictions to create new values for lags in autoregressive models. A time series forecasting library called Skforecast allows using regressors from scikit-learn. In forecasting, future values are predicted based on previously observed patterns (autoregressive), or external factors are incorporated. Previously seen time data are called lags, and when the model trains, it predicts a future interval; this is called a step.

Multiple models are created using different regressors and hyper tuning. The three regressors investigated included: Random Forest Regressor, XGBoost regressor and LGBMRegressor. Using the information gathering in the analysis, which found that lag 1 has the most statistical signification. A recursive autoregressive forecaster with the random forest regressor is created with a lag of 1, a dataset of average daily sentiment about the UK government for the last year. Another Skforecast model is built that integrates with the scikit-learn regressor, XGBoost, for improved performance. XGBoost uses as an optimised, scalable end-to-end tree-boosting system designed to be highly efficient, versatile and lightweight(Chen and Guestrin, 2016). It adopts the gradient-boosting framework, which adopts a unique tree building and helps to speed up the performance. It has been adopted by many data scientists in recent years when it comes to doing a time series forecast. The model takes in a dataset of average daily sentiment about the UK government for the last year. The dataset is split into train and test, with the test data going back 30 days. This model was used to predict 7, 30 and 90 days are predicted.

Two different methods for Hyperparameter tuning were conducted on the model using the XGBRegressor. XGBoost includes many hyperparameters which need to be configured manually; they include max\_depth, the maximum tree depth, learning\_rate, the learning rate and more. Since there are many parameters to tune, the best way to identify the best combination of lags and hyperparameters for Skforecast is by using a Grid Search strategy and a random search Strategy. A third model using an LGBM Regressor is created for comparison using hyperparameter tuning to help decide on the optimal number of lags and n\_estimators for the model.

## Evaluation

Many metrics can be employed to analyse the time series forecaster accuracy. This report will look at two categories of metrics: Scale dependant metrics and Percentage error metrics.

Scale dependant metrics include Mean Absolute Error(MAE) and Root Mean Squared Error (RMSE). Mean Absolute Error(MAE) calculates the mean of the absolute differences between the predicted and the actual values. MAE is a popular metric, hailed by Hyndman, as one of the most straightforward metrics to explain (Hyndman and Koehler, 2006). Root Mean Squared Error (RMSE) is the square root of the average of the set of squared differences between the actual and predicted values. However, a disadvantage to RMSE is it can be sensitive to outliers (Hyndman and Koehler, 2006)

Percentage Error metrics include Mean Absolute Percentage Error(MAPE) and **symmetric mean absolute percentage error (SMAPE).** Mean Absolute Percentage Error(MAPE) is a standard metric used in Forecasting. MAPE has a disadvantage where it gives a heavier error on positive than negative errors (Makridakis, 1993). MAPE is not recommended for negative data or values close to zero. Therefore might not suit this project. **SMAPE, is an alternative to MAPE and solves some of its shortcomings. It can be used on negative values and on data close to zero, which makes it an ideal option for this project.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | ****Scale-Dependent Errors Metric**** | | Percentage-error metrics | | |
| Metrics | Lags | MAE | RMSE | MAPE (%) | sMAPE  original (%) | sMAPE  adjusted (%) |
| RandomForest  Regressor | 1 | 0.1134 | 0.1366 | 348.52% | 155.04% | 77.52% |
| XGBRegressor | 1 | 0.0843 | 0.1070 | 227.38% | 133.29% | 66.64% |
| XGBRegressor with Grid search | 3 | 0.0823 | 0.1021 | 128.58% | 160.76% | 80.38% |
| XGBRegressor with Random Search tuning | 12 | 0.0787 | 0.0962 | 166.49% | 160.11% | 80.05% |
| LGBMRegressor | 24 | 0.0813 | 0.1019 | 114.71% | 187.39% | 93.69% |

Figure 15: Model Evaluation

From the above results table, it can be concluded that the first XGBRegressor was the most accurate based on the sMAPE (original and adjusted) metric being the lowest at 133.29% and 66.64% and the plot below. The autoregressive forecaster with XGBRegressor with 1 lag did a better job at predicting the true values and matching the trend lines, see Figure 16.

|  |  |
| --- | --- |
| Chart, line chart  Description automatically generated | Chart, line chart  Description automatically generated |
|  |  |
|  |  |

Figure 16: Prediction results using different parameters and regressors

The models that were tuned using the Grid Search and Random search strategies performed. This is probably done to the fact there is negative and positive values in the dataset. Scaling the data would probably see an improved performance.

Chart, line chart

Description automatically generated

Backtesting on the data was also performed to assess the accuracy of the forecasting on the existing data.

## Deployment

A dashboard of the machine learning models is deployed using Streamlit and GitHub. Streamlit connects to the GitHub repository and displays the dashboard if it detects one. Requirements for the projects are stored in a requirements.txt file to let Streamlit know what to install. Models are saved from the Jupyter notebook and deployed into the dashboard. The data visualisation for the dashboard uses Plotly. The dashboard is interactive and allows the user to select the date range for the data visualisations.

|  |  |
| --- | --- |
| Dashboard | <https://ritra-msc-sentiment-analysis-dashboardhome-itivf0.streamlit.app/> |
| GitHub | <https://github.com/RitRa/Msc_sentiment_analysis> |

Chart, line chart

Description automatically generated

Figure 17: Streamlit Dashboard of time series predictions

## Conclusion

The architecture for this project included MySQL, PySpark, and Forecasting of time series data. Skforecast library was used for the forecasting; it utilises regressors from scikit-learn learn. A forecasting model using the regressor XGBoost was found to have the best performance with an MAE of 0.8043 and sMAPE (adjusted) OF 66.64%. The model was saved and deployed on the live site powered by Streamlit and GitHub, which shows the forecasted sentiment for 1week, 1 month and 3 months going forward.

## 

## Bibliography

Benesty, J., Jingdong Chen, Yiteng Huang, 2008. On the Importance of the Pearson Correlation Coefficient in Noise Reduction. IEEE Trans. Audio Speech Lang. Process. 16, 757–765. https://doi.org/10.1109/TASL.2008.919072

Brownlee, J., 2017. A Gentle Introduction to Autocorrelation and Partial Autocorrelation. Machine Learning Mastery. URL https://machinelearningmastery.com/gentle-introduction-autocorrelation-partial-autocorrelation/ (accessed 11.6.22).

Chen, T., Guestrin, C., 2016. XGBoost: A Scalable Tree Boosting System, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 785–794. https://doi.org/10.1145/2939672.2939785

Hutto, C.J., Gilbert, E., 2014. VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text 10.

Hyndman, R.J., Koehler, A.B., 2006. Another look at measures of forecast accuracy. International Journal of Forecasting 22, 679–688. https://doi.org/10.1016/j.ijforecast.2006.03.001

Makridakis, S., 1993. Accuracy measures: theoretical and practical concerns. International Journal of Forecasting 9, 527–529. https://doi.org/10.1016/0169-2070(93)90079-3

Mushtaq, R., 2011. TESTING TIME SERIES DATA FOR STATIONARITY 19. http://dx.doi.org/10.2139/ssrn.1911068

Stack Overflow, 2022. Developer Survey [WWW Document]. Stack Overflow. URL https://survey.stackoverflow.co/2022/?utm\_source=social-share&utm\_medium=social&utm\_campaign=dev-survey-2022 (accessed 10.30.22).