National College of Ireland

Configuration Manual

MSc Research Project
Master of Science in Artificial Intelligence

Forename Surname
Student ID: x23311550

School of Computing
National College of Ireland

Supervisor: Dr. Devanshu Anand

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Niall Kierans
Student ID:	x23311550
Programme:	MSCAITOP
Year:	2024
Module:	MSC Artificial Intelligence
Supervisor:	Dr. Devanshu Anand
Submission Due Date:	12th August 2024
Project Title:	Configuration Manual
Word Count:	1488
Page Count:	16

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Niall Kierans
Date:	11th August 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Niall Kierans
x233115501

1 Introduction

This configuration manual explains all the steps from design right through to the evaluation results. This project covers sentiment analysis on Google reviews for airports using Natural Language Processing (NPL) and Machine Learning (ML) which covers the data input, python libraries, data transformation, modeling methods, and evaluation results.

2 Environment Setup

2.1 System specification

The implementation of this project was completed via Juypter Notebook from Anaconda, using a Dynabook laptop with 8 GB RAM and, 64-bit operating system on Windows 11 Home edition. There was no additional hardware or software required as the goal of the project to to find to best prediction models.

2.2 Technical specification

The models were written in Python program languages and the following packages where used to complete the project: Pandas

1. Numpy
2. NLTK
3. Re - Regular expression operations
4. matplotlib
5. seaborn
6. time
7. string
8. textblob
9. sklearn
10. autocorrect

11. langdetect
12. vaderSentiment
13. nrclex
14. wordcloud

3 Dataset

The Google reviews data was retrieved and downloaded via a third-party vendor called APIfy. This was an easy, quick, and inexpensive way to receive the data required for the project. The data for each airport was downloaded individually in CSV format. Each file was then unzipped and the data was combined manually in Excel, only keeping the small number of features required for the project.

4 Implementation

The implementation steps will be discussed in this section. There is only one data set, but 10 Jupyter Notebook files (.ipynb). Screenshots will form part of this section to provide more details on the process. A summary of the models applied to the Airport data set is as follows:

1. **Textblob** - Compute sentiment polarity and subjectivity.
2. **Vader** - Analyze sentiment polarity scores (positive, negative, neutral).
3. **NRCLex** - Assign emotions to text (joy, anger, sadness, etc.).
4. **Random Forest** - Ensemble learning method using multiple decision trees.
5. **SVM** - Classification technique that finds the hyperplane separating different classes.
6. **Naive Bayes** - Probabilistic classifier based on Bayes' theorem.

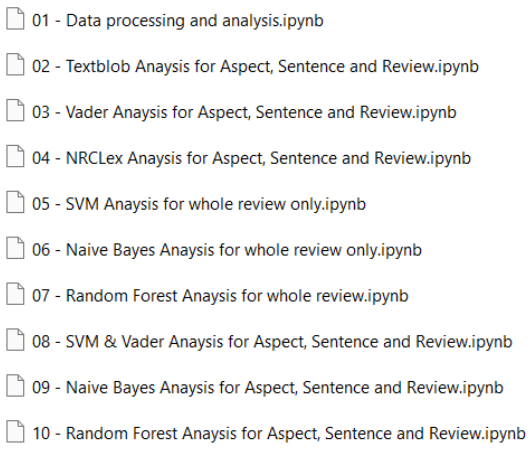

Figure 1: List of the Python Code Files

4.1 Data Preprocessing

1. The data was loaded from the same location as the Jypyter Notebook file using the following code:

```
#Data Collection
df = pd.read_excel('Data 6 airports.xlsx')
```

Figure 2: Code to load the data

2. Used the following steps to review the data before moving forward
   (a) Use the **info** function to review the features for non-null count and data type
   (b) Check the unique (nunique) values for each feature.
3. Any rows where the review comment was blank were removed.
4. New feature added to count the number of characters in a review

```
# Function to count characters in a string
def count_characters(text):
    if isinstance(text, str):
        return len(text)
    else:
        return 0  # Return 0 for NaN values

# Add a new column 'char_count' with the count of characters in 'Review Comment'
data['char_count'] = data['Review Comment'].apply(count_characters)
print(data)
```

Figure 3: Code to Count Review Characters

5. Finally review the number of Google stars ratings to get an idea of the distribution of the data.

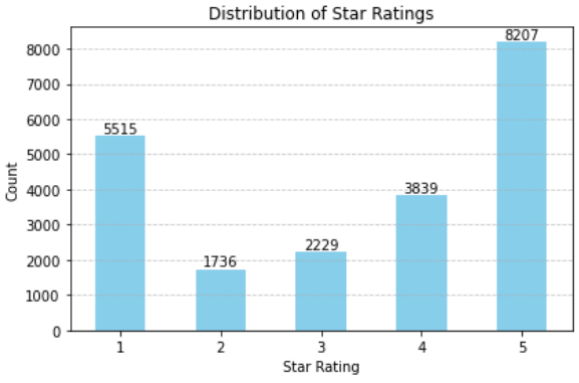

Figure 4: Distribution of Google Reviews Star Ratings

6.

4.2 Data Collection Analysis

1. Code to review the data in several different ways

   (a) Count of overall Google star ratings.

   (b) Count of Google star ratings per airport.

   (c) Google star ratings distribution by day of the week.

   (d) Google star ratings by hour of the day

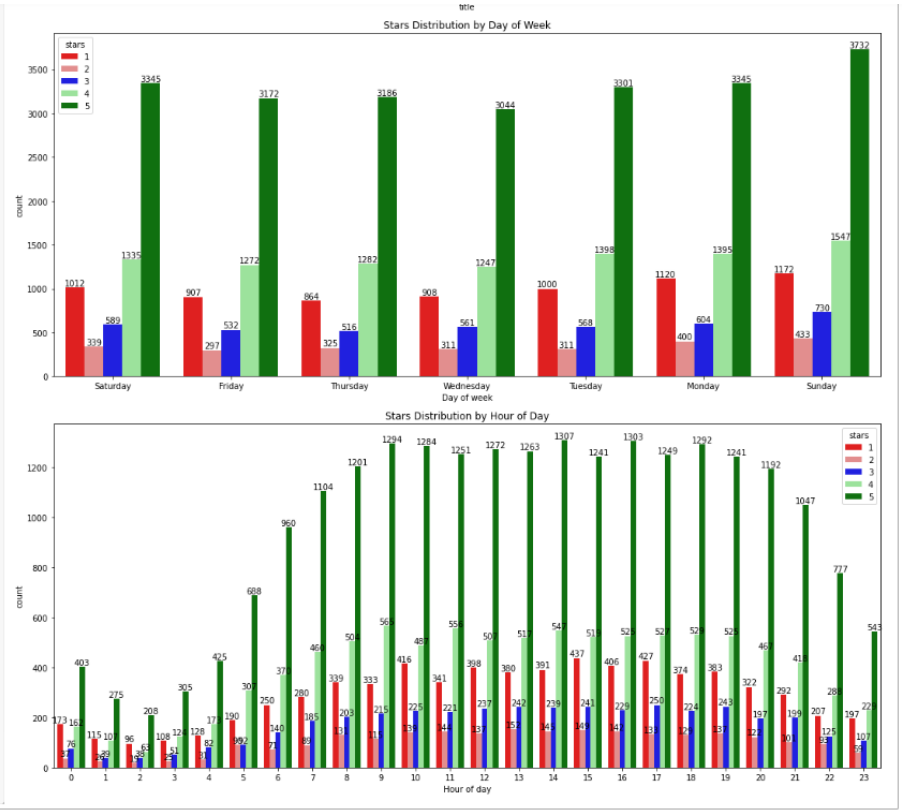

Figure 5: Examples of Analysis of Google Star Ratings

2. Retrieve the most popular aspects from the Google Reviews and generate a word cloud visual.

	Noun	Frequency
10	airport	12411
27	security	4189
24	staff	3557
55	flight	3125
1	time	3124
4	people	2414
56	hour	2021

Figure 6: Top 7 Aspects from Google Reviews

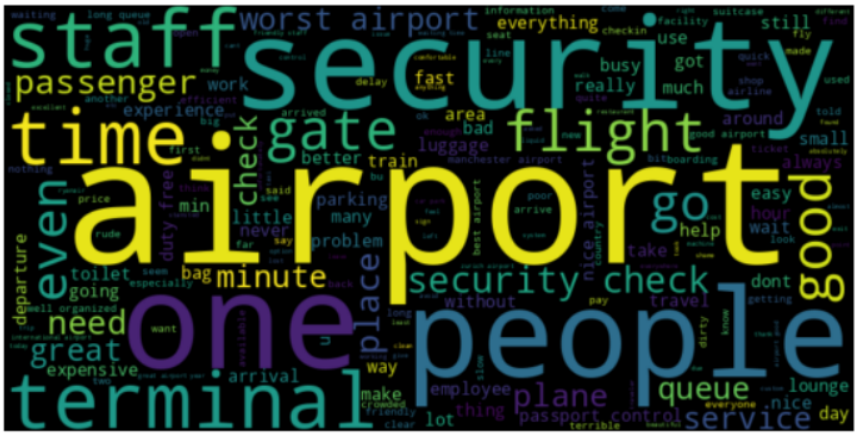

Figure 7: Word Cloud of the most common Aspects

```
Top 20 Bigrams:
security check: 924
worst airport: 749
passport control: 687
duty free: 462
nice airport: 453
```

```
Top 20 Trigrams:
worst airport ever: 187
one best airport: 123
one worst airport: 110
duty free shop: 91
airport ive ever: 81
```

```
Top 20 Fourgrams:
worst airport ive ever: 48
one best airport world: 31
worst airport ' ever: 27
one best airport europe: 26
easy find way around: 21
```

Figure 8: Top 5 Bi-grams, Tri-grams and Four-grams

3. Using Ngrams in Python to generate the top 20 Uni-grams, Bi-grams, Trigrams and Four-grams from all the Google Reviews.

4. Generate the following information and visualisations from Ngrams for the Google Reviews by Airport:

   (a) Top ten Bi-grams for each of the six airports.

   (b) Top ten Trigrams for each of the six airports.

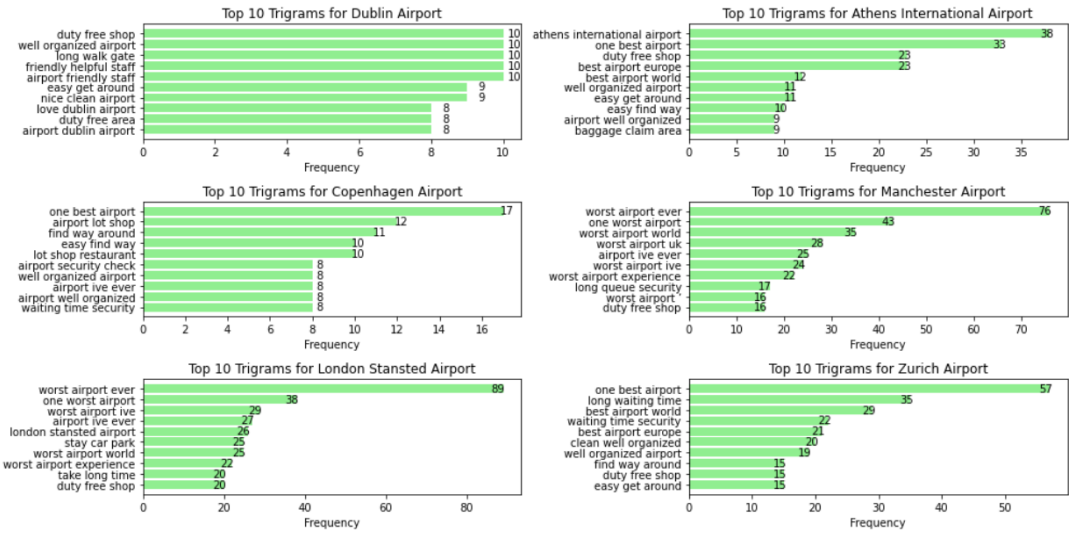

Figure 9: Top 10 Tri-grams For Each Airport

5. Generate Bi-grams and Tri-grams for the top 5 aspect words:
   (a) Airport
   (b) Security
   (c) Staff
   (d) Flight
   (e) Time

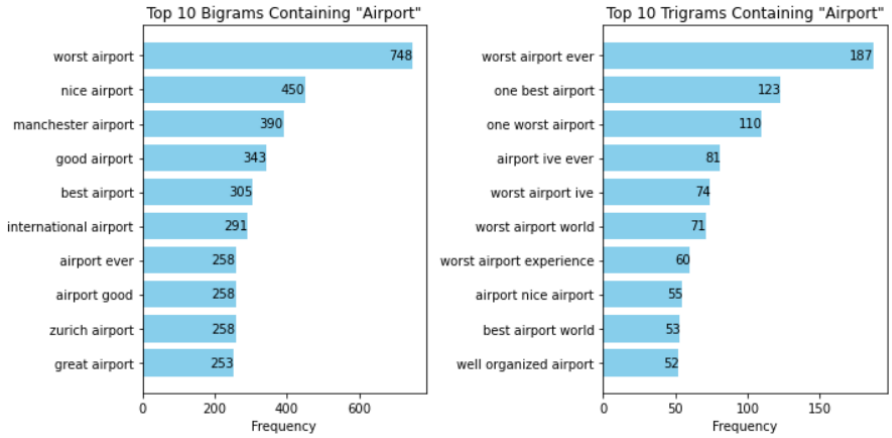

Figure 10: Top 10 Bi-gram and Tri-gram for the Aspect 'Airport')

4.3 Data Preprocessing before NLP and ML

Before any natural language processing or machine learning techniques were deployed there was a process to clean the data beforehand. The cleaning steps generate a new feature called 'cleaned reviews comment'. Here are the steps involved:

- Convert the text to lowercase
- Remove any URLs
- Remove any emails
- Remove punctuation
- Remove numbers
- Tokenise the text
- Remove stop words and lemmatise
- Rejoin tokens into strings
- Remove extra white spaces
- Remove leading and trailing white spaces

```python
# Get stop words
stop_words = set(stopwords.words('english'))
lemmatizer = WordNetLemmatizer()

# Function to clean text
def clean_text(text):
    text = str(text).lower()  # Convert to lowercase
    text = re.sub(r'http\S+', '', text)  # Remove URLs
    text = re.sub(r'\S*@\S*\s?', '', text)  # Remove emails
    text = re.sub(f'[{string.punctuation}]', '', text)  # Remove punctuation
    text = re.sub(r'\d+', '', text)  # Remove numbers
    tokens = word_tokenize(text)  # Tokenize the text
    tokens = [lemmatizer.lemmatize(word) for word in tokens if word not in stop_words]  # Remove stopwords and lemmatize
    text = ' '.join(tokens)  # Rejoin tokens into a string
    text = re.sub(' +', ' ', text)  # Remove extra whitespace
    text = text.strip()  # Remove leading/trailing whitespace
    return text
```

Figure 11: Python Code to Clean Data Before NLP and ML

4.4 Sentiment Analysis - Natural Language Processing (NLP)

Each of the NLP sentiment analyses was processed at different levels of the review left by passengers for each airport. The '**Whole Review**', '**Sentence Level**' and '**Aspect Level**' with the objective of understanding which returned the best evaluation scores. On the second line of each of the NPL sentiment files, you can enter the number of aspects you want to run. The default number is 50 however you can change the number as desired. The higher the aspect number the longer the coding will take to execute the results.

```python
# Capture the start time
start_time = time.time()

# Enter the top x amount of aspects to use in the Evaluation
num_of_aspects = 50

#Data Collection
df = pd.read_excel('Data 6 airports.xlsx')
```

Figure 12: Enter the number of aspects required

Each of the sentiments for NPL use the Google Star Ratings as a comparison for the evaluation. Google Star ratings 4 and 5 = Positive, 3 = Neutral, and 1 and 2 = Negative.

```python
# Map 5-point google star ratings to sentiments - 1 & 2 = Negative, 3 = Neutral, 4 & 5 = Positive
def map_rating_to_sentiment(rating):
    if rating in [1, 2]:
        return 'negative'
    elif rating == 3:
        return 'neutral'
    else:
        return 'positive'

# Apply mapping to actual sentiment ratings
data['actual sentiment'] = data['stars'].apply(map_rating_to_sentiment)
```

Figure 13: Google Star Rating Sentiment Ratings

4.4.1 Textblob

The positive, negative and neutral sentiments were detected using the Textblob library from Python. The sentiment will add two new features to the data set, '**whole review sentiment**' and '**whole review compound**'. If the compound score is equal to 0, then the sentiment is Neutral, if the compound score is greater than 0 then the sentiment is Positive and if the compound score is less than 0 then the sentiment is negative.

```python
# Function to calculate sentiment scores using TextBlob
def textblob_sentiment_score(text):
    if pd.isna(text):
        return 0  # If text is NaN, return neutral (compound 0)
    blob = TextBlob(text)
    sentiment_score = blob.sentiment.polarity
    return sentiment_score

# Function to determine sentiment and compound score for a whole review
def whole_review_sentiment_textblob(review):
    sentiment_score = textblob_sentiment_score(review)
    # Determine sentiment label based on sentiment score
    sentiment = 'positive' if sentiment_score > 0 else 'negative' if sentiment_score < 0 else 'neutral'
    return {'sentiment': sentiment, 'compound': sentiment_score}
```

	publishedAtDate	title	stars	text	textTranslated	Review Comment	char_count	Cleaned Review Comment	Extracted Nouns	Whole Review Sentiment	Whole Review Compound
0	2024-04-06T07:11:06.489Z	Dublin Airport	5	Nie mam problemu sprawnie zawsze.	I have no problem working efficiently all the ...	I have no problem working efficiently all the ...	51	problem working efficiently time	[problem, time]	neutral	0.000000
2	2024-04-06T04:09:21.237Z	Dublin Airport	5	Everything is well-organized.	NaN	Everything is well-organized.	29	everything wellorganized	[everything]	neutral	0.000000
3	2024-04-06T04:07:51.293Z	Dublin Airport	3	New scanning machines need people that know ho...	NaN	New scanning machines need people that know ho...	103	new scanning machine need people know use long...	[machine, people, wait, line]	positive	0.136364

Figure 14: Textblob Model

Comparing the Google star rating sentiment against the Texblob sentiment to get the evaluation scores using the confusion matrix, accuracy, precision, recall and f1 scores.

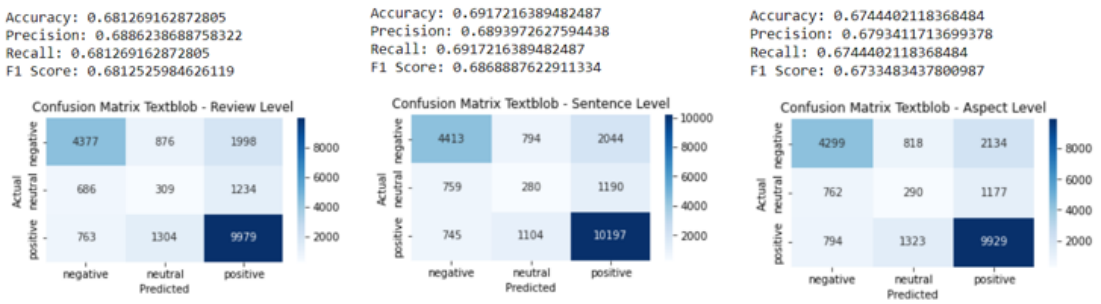

Figure 15: Evaluation Scores for Textblob

4.4.2 Vader

The positive, negative and neutral sentiments were detected using the Vader library from Python. The sentiment will add two new features to the data set, '**whole review**

sentiment' and '**whole review compound**'. If the compound score is equal to 0, then the sentiment is Neutral, if the compound score is greater than 0 then the sentiment is Positive and if the compound score is less than 0 then the sentiment is negative.

```python
# Initialize VADER sentiment analyzer
analyzer = SentimentIntensityAnalyzer()

# Whole Level Sentiment Analysis (Complete reivew)
def vader_sentiment_score(text):
    if pd.isna(text):
        return {'compound': 0}
    sentiment_dict = analyzer.polarity_scores(text)
    return sentiment_dict

def whole_review_sentiment_vader(review):
    sentiment_dict = vader_sentiment_score(review)
    compound = sentiment_dict['compound']
    return {'sentiment': 'positive' if compound > 0 else 'negative' if compound < 0 else 'neutral', 'compound': compound}
```

	publishedAtDate	title	stars	text	textTranslated	Review Comment	char_count	Cleaned Review Comment	Extracted Nouns	Whole Review Sentiment	Whole Review Compound
0	2024-04-06T07:11:06.489Z	Dublin Airport	5	Nie mam problemu sprawnie zawsze.	I have no problem working efficiently all the ...	I have no problem working efficiently all the ...	51	problem working efficiently time	[problem, time]	positive	0.6070
2	2024-04-06T04:09:21.237Z	Dublin Airport	5	Everything is well-organized.	NaN	Everything is well-organized.	29	everything wellorganized	[everything]	neutral	0.0000
3	2024-04-06T04:07:51.293Z	Dublin Airport	3	New scanning machines need people that know ho...	NaN	New scanning machines need people that know ho...	103	new scanning machine need people know use long...	[machine, people, wait, line]	negative	-0.4215

Figure 16: Vader Model

Comparing the Google star rating sentiment against the Vader sentiment to get the evaluation scores using the confusion matrix, accuracy, precision, recall and f1 scores.

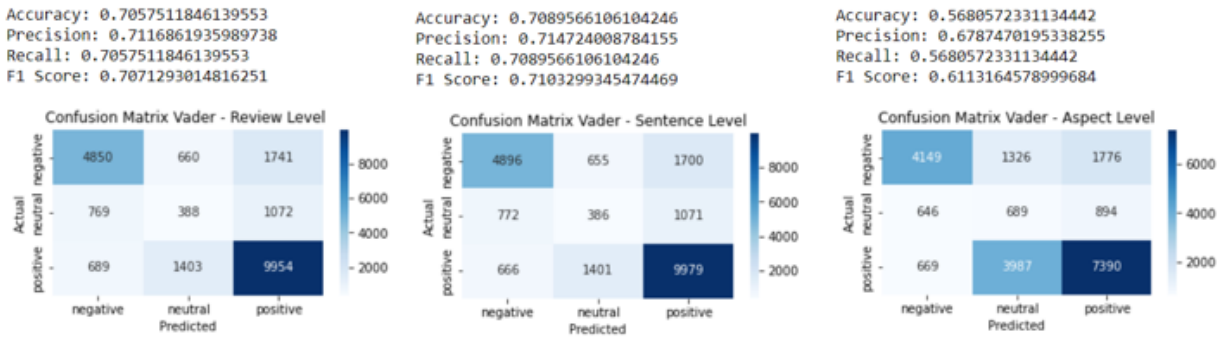

Figure 17: Evaluation Scores for Vader

4.4.3 NRCLex

The positive, negative and neutral sentiments were detected using the NRCLex library from Python. The sentiment will add two new features to the data set, '**whole review sentiment**' and '**whole review compound**'. If the compound score is equal to 0, then the sentiment is Neutral, if the compound score is greater than 0 then the sentiment is Positive and if the compound score is less than 0 then the sentiment is negative.

```python
def nrc_sentiment_score(text):
    if pd.isna(text):
        return {'compound': 0}
    nrc_obj = NRCLex(text)
    return nrc_obj.affect_frequencies

def whole_review_sentiment_nrc(review):
    sentiment_dict = nrc_sentiment_score(review)
    # Calculate a compound score based on the affect frequencies
    compound = sentiment_dict['positive'] - sentiment_dict['negative']
    return {
        'sentiment': 'positive' if compound > 0 else 'negative' if compound < 0 else 'neutral',
        'compound': compound
    }
```

	publishedAtDate	title	stars	text	textTranslated	Review Comment	char_count	Cleaned Review Comment	Extracted Nouns	Whole Review Sentiment (NRC)	Whole Review Compound (NRC)
0	2024-04-06T07:11:06.489Z	Dublin Airport	5	Nie mam problemu sprawnie zawsze.	I have no problem working efficiently all the ...	I have no problem working efficiently all the ...	51	problem working efficiently time	[problem, time]	neutral	0.000000
2	2024-04-06T04:09:21.237Z	Dublin Airport	5	Everything is well-organized.	NaN	Everything is well-organized.	29	everything wellorganized	[everything]	neutral	0.000000
3	2024-04-06T04:07:51.293Z	Dublin Airport	3	New scanning machines need people that know ho...	NaN	New scanning machines need people that know ho...	103	new scanning machine need people know use long...	[machine, people, wait, line]	negative	-0.333333
4	2024-04-05T22:53:44.080Z	Dublin Airport	5	Great relatively small airport,very efficient....	NaN	Great relatively small airport,very efficient....	100	great relatively small airportvery efficient n...	[airportvery, customer, service, people]	positive	0.166667

Figure 18: NRCLex Model

Comparing the Google star rating sentiment against the NRCLex sentiment to get the evaluation scores using the confusion matrix, accuracy, precision, recall and f1 scores.

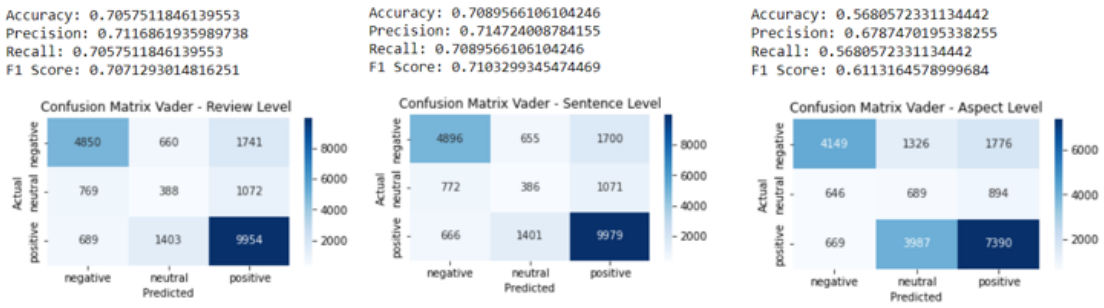

Figure 19: Evaluation Scores for NRCLex

On all the Python files there is code to monitor the computation speed in seconds. This was used mainly for the Aspect Level sentiment analysis as the higher the number of aspects required the longer the computation.

4.5 Sentiment Analysis - Machine Learning (ML)

In this section, the different levels of sentiment were not used and all outcomes are based on the complete review left by the passenger. The ML models used in this section are

```
# Capture the start time
start_time = time.time()
```

```
# Capture the end time
end_time = time.time()

# Calculate the total runtime
total_runtime = end_time - start_time

print(f"Total runtime: {total_runtime} seconds")
```

```
Total runtime: 95.29323768615723 seconds
```

Figure 20: Python Code to Provide Computation Time

Support Vector Machine (SVM), Random Forest and Naive Bayes.

4.5.1 Model Setup

All three ML methods use the below code to set up the data for the training and test.

- **CountVectorizer** - Turn the words into numbers
- **Vectorizer = CountVectorizer()**
- **X = vectorizer.fit_transform(data['Cleaned Review Comment'])** - This section contains all the passenger's reviews. It reads all the reviews and processes them into a matrix.
- **y = data['actual sentiment']** - This part is how the passenger feels - Positive, Negative or Neutral. This sentiment is determined based on the Google review star ratings (4/5 = Positive, 3 = Neutral, 1/2 = Negative)

```
# SVM for whole
# Convert text to features using CountVectorizer
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(data['Cleaned Review Comment'])
y = data['actual sentiment']
```

Figure 21: Python Code to Prepare Data for ML Methods

4.5.2 Model Train and Test

When applying all three models for the sentiment analysis 80 percent was used for training and 20 percent for test. Random state seed is set at 42.

4.5.3 Evaluation Results

The results for all three ML methods where evaluated against a confusion matrix, accuracy, precision, recall and f1 scores.

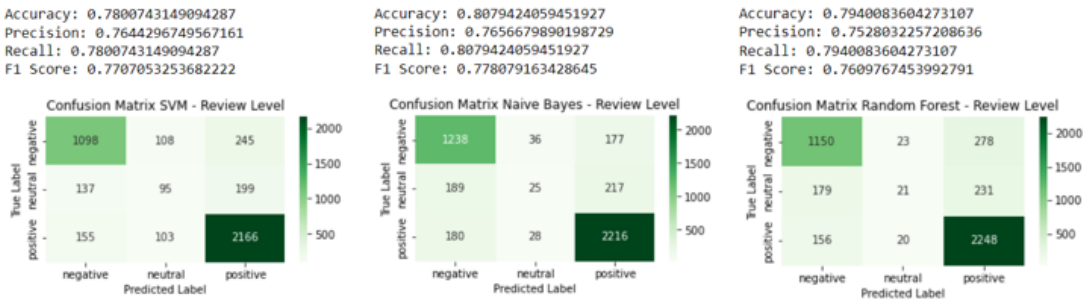

Figure 22: Enter Caption

4.6 Blending NLP and ML for Sentiment Analysis

In this section Vader will blend with the three machine learning methods with the aim to improve the evaluation scores. Vader was used as it achieved the highest evaluation scores from the NLP sentiment. Similar to the machine learning section only the full review level was used, leaving out the sentence and aspect level.

4.6.1 Model Setup

The same coding set-up as the machine learning methods in the previous section with the only difference being at different **Y** value. **y = data['Whole Review Sentiment']** = Positive, Negative and Neutral determination based on compound score from Vader (Instead of Google Review Star Ratings). Score less than 0 = Negative, Score greater than 0 = Positive and Score equal to 0 = Neutral.

	publishedAtDate	title	stars	text	textTranslated	Review Comment	char_count	Cleaned Review Comment	Extracted Nouns	Whole Review Sentiment	Whole Review Compound
0	2024-04-06T07:11:06.489Z	Dublin Airport	5	Nie mam problemu sprawnie zawsze.	I have no problem working efficiently all the ...	I have no problem working efficiently all the ...	51	problem working efficiently time	[problem, time]	positive	0.6070
2	2024-04-06T04:09:21.237Z	Dublin Airport	5	Everything is well-organized.	NaN	Everything is well-organized.	29	everything wellorganized	[everything]	neutral	0.0000
3	2024-04-06T04:07:51.293Z	Dublin Airport	3	New scanning machines need people that	NaN	New scanning machines need people that	103	new scanning machine need people know use	[machine, people, wait, line]	negative	-0.4215

Figure 23: Vader Features for use with ML Methods

4.6.2 Model Train and Test

When applying all three models for the sentiment analysis 80 percent was used for training and 20 percent for test. Random state seed is set at 42.

4.6.3 Evaluation Results

The results for all three ML methods where evaluated against a confusion matrix, accuracy, precision, recall and f1 scores.

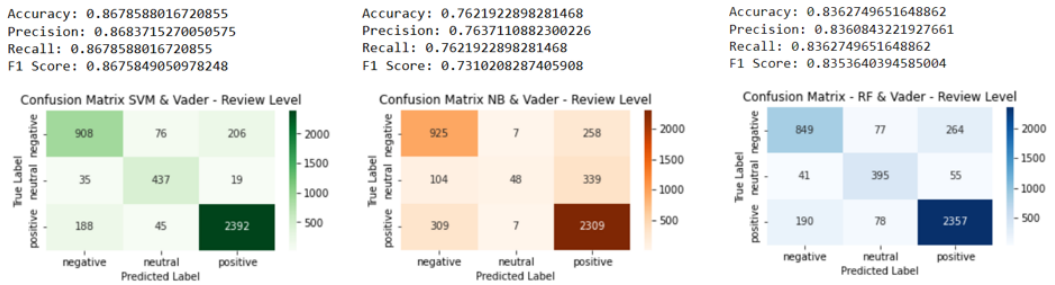

Figure 24: Evaluation Results from Blended Approach

5 Appendix

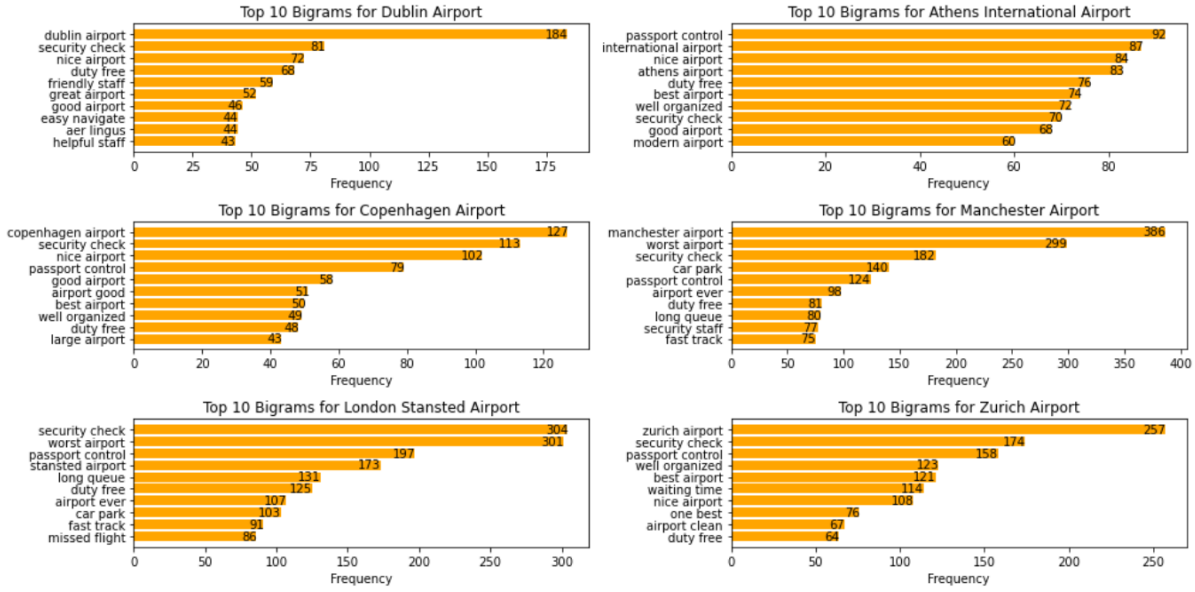

Figure 25: Top 10 Bi-grams for All Airports

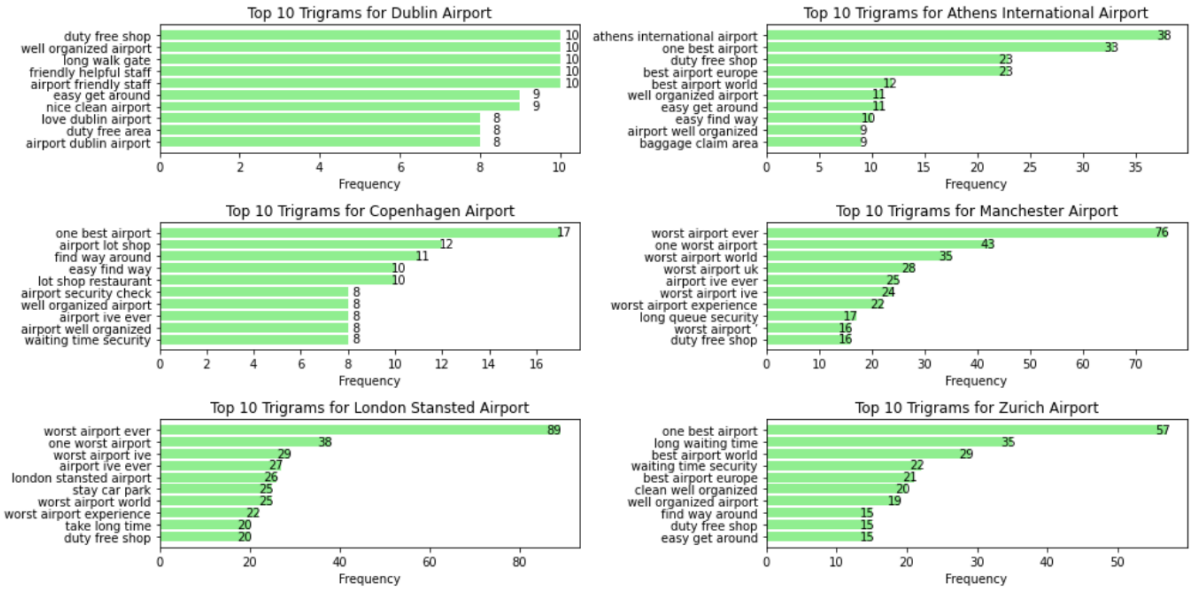

Figure 26: Top 10 Tri-grams for All Airports

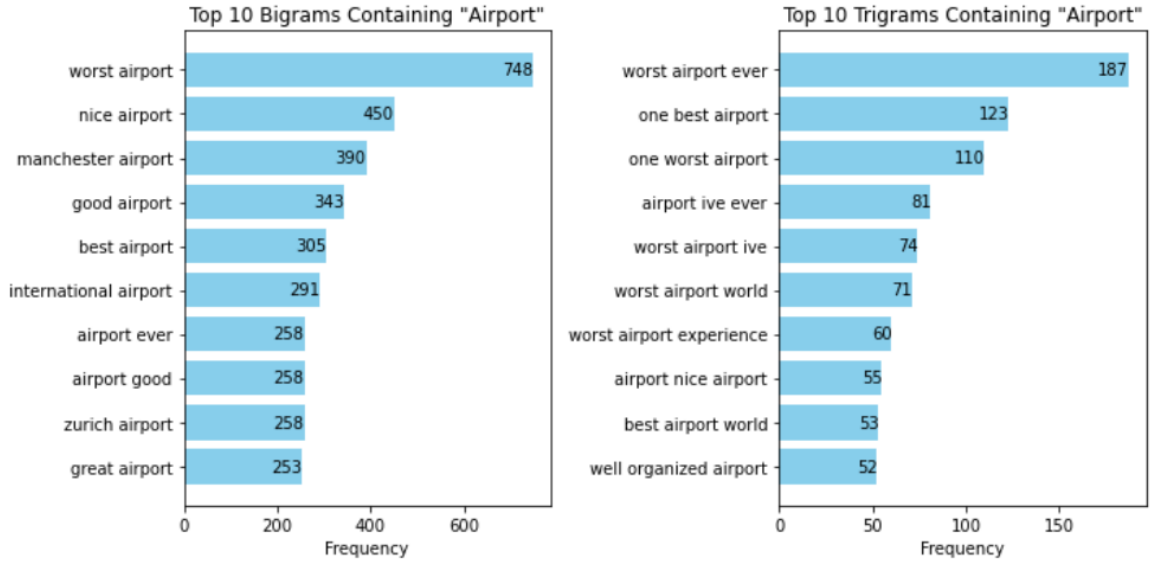

Figure 27: Top 10 Bi-grams and Tri-grams for 'Airport'

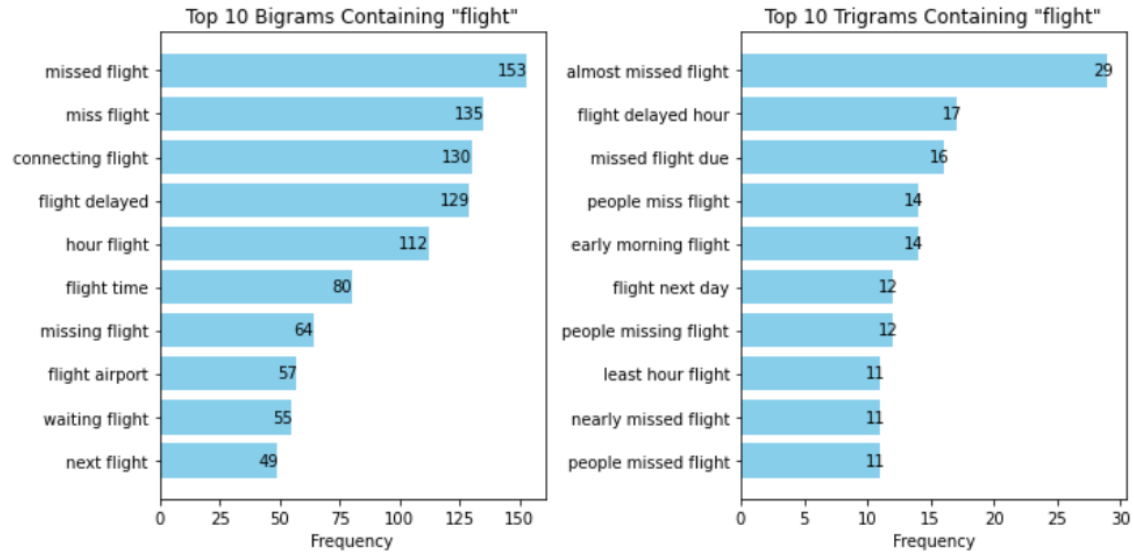

Figure 28: Top 10 Bi-grams and Tri-grams for 'Flight'

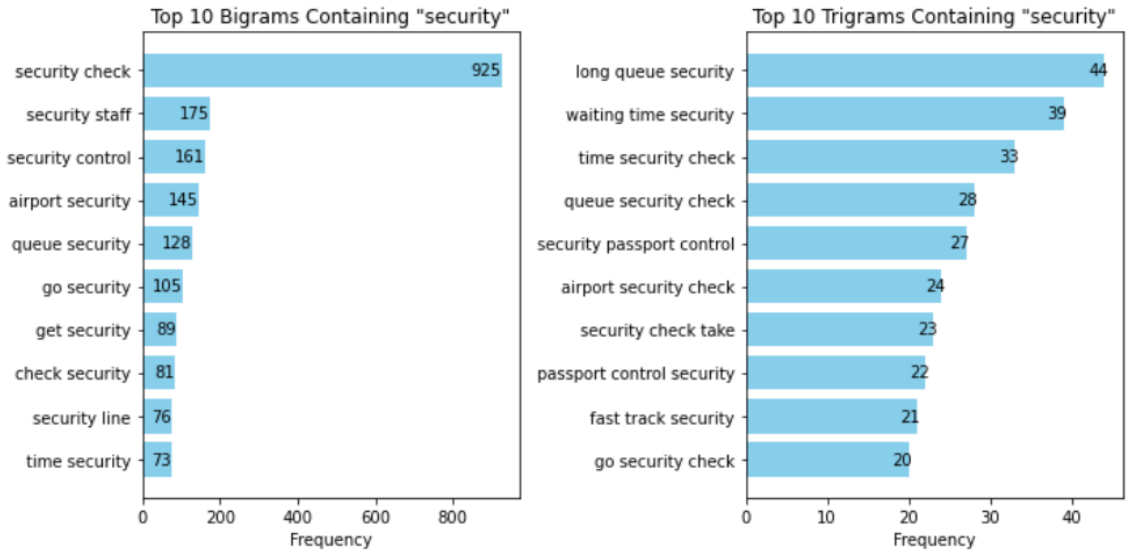

Figure 29: Top 10 Bi-grams and Tri-grams for 'Security'

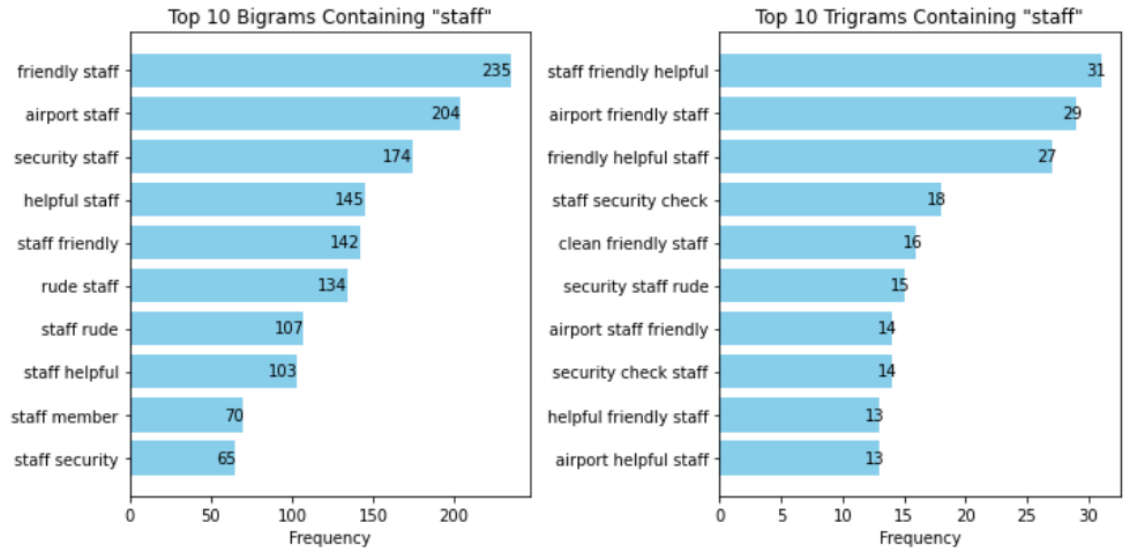

Figure 30: Top 10 Bi-grams and Tri-grams for 'Staff'

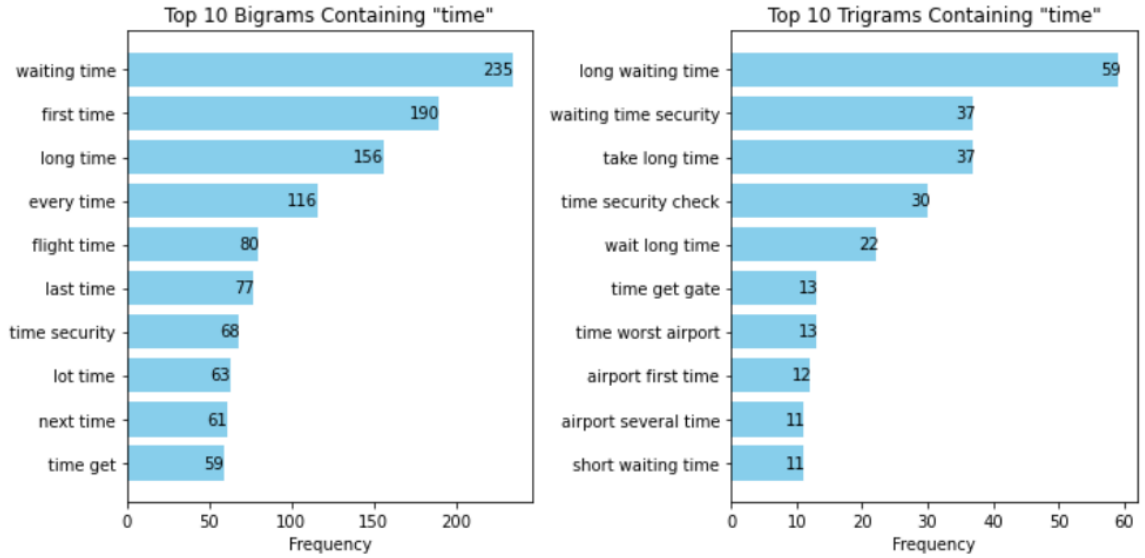

Figure 31: Top 10 Bi-grams and Tri-grams for 'Time'