# ANALYSIS OF EDINBURGH AIRPORT CUSTOMERS RESPONSES

Chekalina Alisa , Cristea Carmen , Vo Nguyen Thao Nhi

Université Panthéon-Sorbonne
Supervised by Armand L'Huiller and Aryan  Razaghi

## Table des matières

# Introduction

This paper investigates the feedback from visitors to Edinburgh Airport in order to identify key aspects of their experience and uncover opportunities for service improvement. Analyzing travelers' comments about Edinburgh Airport is crucial for several reasons. In today's competitive landscape, where airports are constantly competing for passengers, feedback from travelers plays a pivotal role in maintaining and enhancing service quality. As one of the largest and busiest airports in the UK, Edinburgh Airport serves millions of passengers annually. In such an environment, understanding passengers' opinions on various aspects of the airport including customer service, cleanliness, convenience, and security—is essential.

In this project, we employed various methods to analyze the data, including graphing, analysis of categorical variables and their effect on satisfaction levels using a logit model, sentiment analysis, examination of the impact of sentiment on satisfaction levels, topic modeling, and analysis of comments with the most frequently mentioned words.

## Code components / README

Before we delve into describing our work, it's important to familiarize ourselves with the structure of our code. Our project consists of 11 scripts and one primary notebook, named "main", which contains the entire analysis.

- **Load_data.py**: This script handles the automatic loading of the database.
- **Cleaning.py**: This script replaces missing values, converts date-type variables, and removes unnecessary quotation marks from column names.
- **Cleaning_categorial.py:** This script prepares data for categorical variables analysis.
- **Variables_transformation.py**: This script modifies variables for use in the analysis and renames some columns.
- **Preprocessing.py**: This script prepares the text for analysis, including tokenization and removal of stop words.
- **Categorial_functions.py**: This script contains functions used for categorical variables analysis.
- **Function_analysis_freq.py**: This script is used for the manual analysis of comments with the most frequent words.
- **Topic_modeling.py**: This script contains functions used in topic modeling.
- **Predict_sentiments.py**: This script contains functions used for predicting the sentiments.
- **Vader.py**: This script contains functions used for analyzing the sentiments based on Vader method.
- **Graph.py**: This script contains functions used to build some useful graphs for our analysis.

## Analyzing of cleaned database

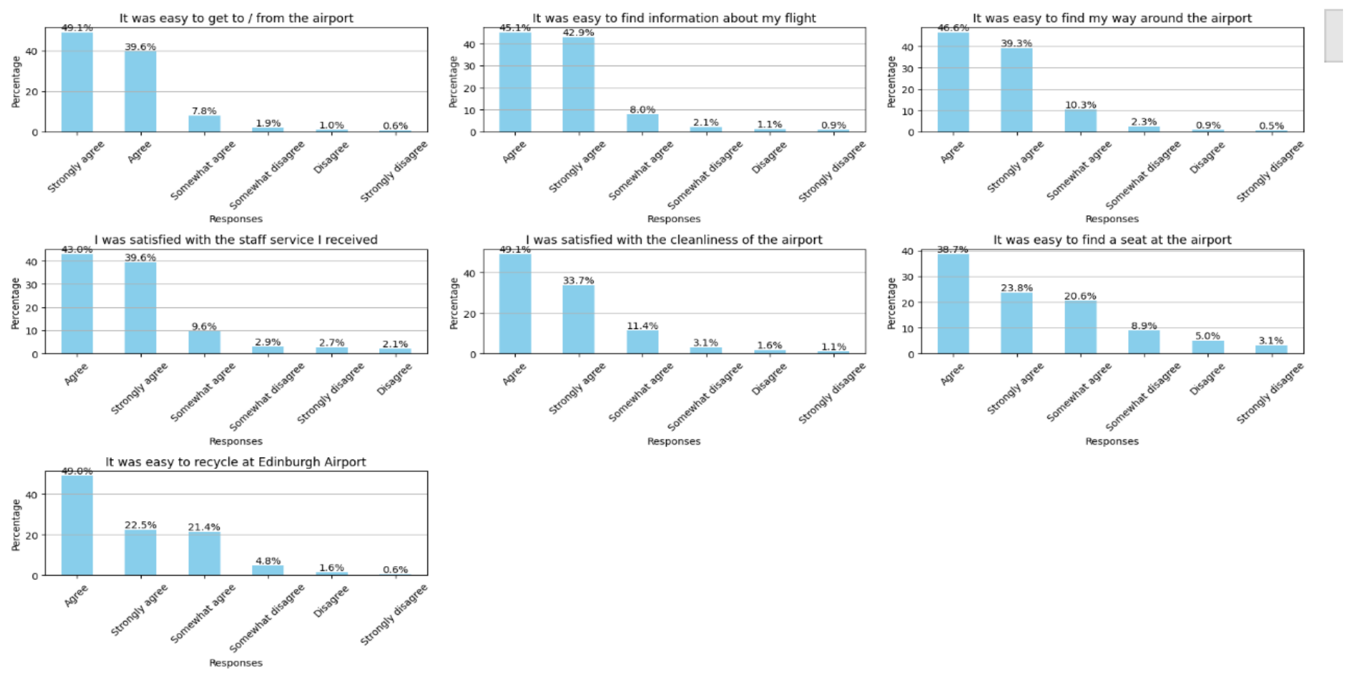
Our database, “customer_service_responses”, is essentially a compilation of responses from airport visitors to a comprehensive questionnaire. Before processing, the database consists of 24,934 responses and includes 62 columns, which correspond to approximately 62 questions answered by passengers. (In reality, there are 60 questions, as two variables, indicating the start and end of the survey, are automatically included.)

Most questions require respondents to choose from a set of given options. For example, the question “It was easy to get to/from the airport” requires respondents to select one of the following options: “Strongly disagree, disagree, somewhat disagree, somewhat agree, agree, strongly agree.” However, there are two questions that require an open-ended response:

1. *“Overall, were you satisfied with your most recent experience at Edinburgh Airport?”*
2. *“Are there any other premium services that you would like to see introduced at Edinburgh Airport?”*

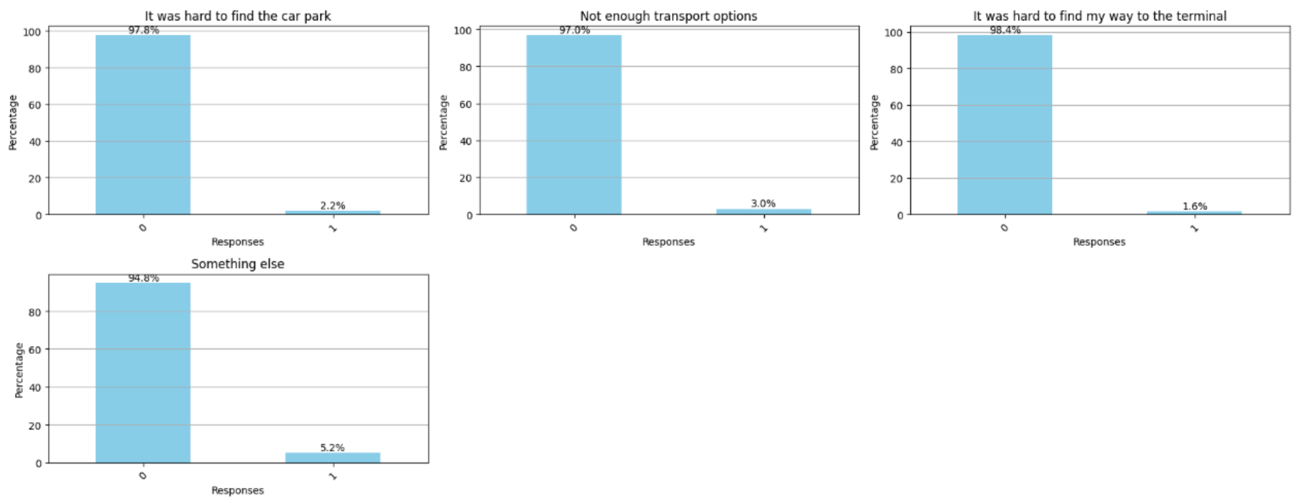
These open-ended questions are the primary focus of our textual data analysis

After cleaning our database using the cleaning function, we proceeded to generate several graphs to gain insights into the general opinions of visitors.

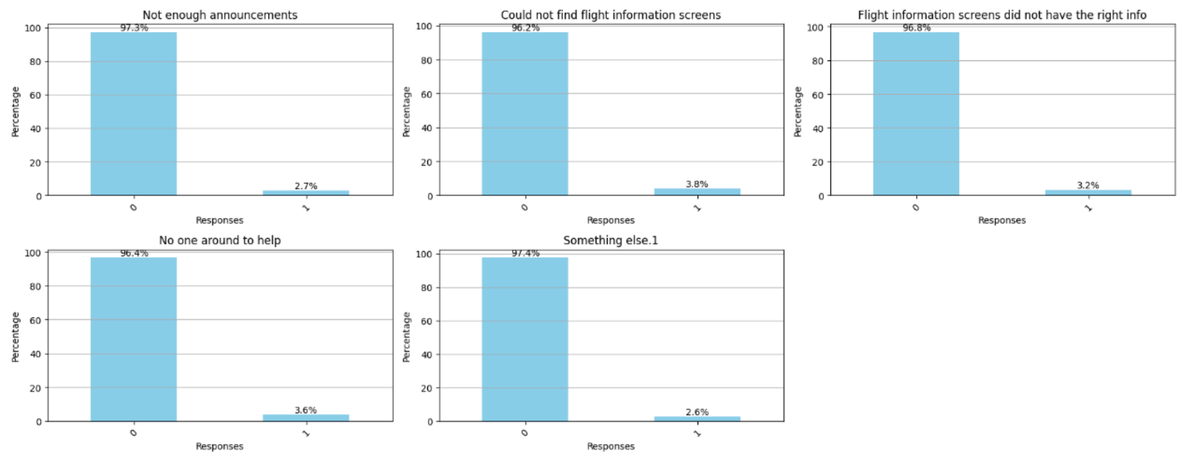


The first graph illustrates the percentage of responses to the main questions of the questionnaire. It reveals a notably high percentage of individuals expressing satisfaction with the airport's services. However, it's worth noting that a significant portion of respondents may select "Strongly Agree" simply to expedite the survey process. Therefore, our focus should primarily be on negative responses, as they often highlight areas of improvement for the airport. Among the questions compared, it's apparent that the highest level of disagreement is observed with the statement "It was easy to find a seat in the airport," indicating a potential issue with the availability of seating.

Then, we constructed graphs for the clarifying questions. For example, after the question “It was easy for me to find my way to/from the airport”, respondents who disagreed with the statement were asked to clarify what aspect of their experience they found difficult. These clarifying questions allow for a deeper understanding of the reasons for negative responses and identify areas for attention and improvement.



We can observe that the largest percentage of the chosen option is “Something else”, so there was something else that they did not like in the commitment, we will analyze this question later.



In the flight information survey, the highest number of respondents reported difficulty finding flight information screens, accounting for 3.8% of the total. While this percentage may seem relatively small, when we calculate the actual number, it equates to approximately 950 individuals. This figure underscores the significance of the issue, highlighting that a considerable number of people experienced challenges in accessing flight information screens.

It's evident from the feedback about staff (we continue without graphs) that people are expressing concerns about the rudeness of the staff (5,3%) and a perceived lack of assistance available (5,7%). These complaints suggest that there may be issues with customer service and staffing levels at the airport. Addressing these concerns could help improve the overall experience for visitors.

Regarding cleanliness, approximately 2,200 individuals, constituting 8.8% of respondents, noted dirty toilets as an issue. Additionally, 5.1% of respondents, a notable portion, cited trash as a cleanliness concern. These findings indicate areas where improvements in cleanliness and maintenance may be necessary to enhance the overall perception of the airport.

The subsequent question addressed seating availability, reflecting the prevalent dissatisfaction observed earlier. Specifically, a significant number of respondents expressed concerns about inadequate seating in various areas. Among these, the most common complaint was the lack of seats in front of the gates, with 26.2% of respondents, totaling approximately 6,550 individuals, highlighting this issue. This substantial number underscores the importance of addressing seating shortages, not only in front of exits but also in other areas such as those before security and restaurants, to enhance passenger comfort and satisfaction.

In the final question regarding trash separation, a notable number of respondents mentioned challenges related to recycling bins (11,6%) and the absence of proper trash separation facilities (8,7%). This feedback underscores the importance of implementing effective waste management practices, including providing accessible recycling bins and improving trash separation processes, to promote environmental sustainability within the airport premises.

In summary, this analysis reveals some weaknesses within the airport, notably the insufficient availability of seating for travelers. Additionally, concerns regarding cleanliness, accessibility of flight information screens, staff behavior, and trash separation were also identified. Addressing these shortcomings, particularly the seating shortage, will be pivotal in improving the overall passenger experience and satisfaction levels at the airport.

# Categorical variables

## General analysis

### 1. Data cleaning

As for the categorical variables, we first clean and prepare our database for further analysis. For that purpose, we use the following functions: *map_agree_disagree, map_likely_unlikely and cleaning.*

The function *map_agree_disagree* and *map_likely_unlikely* converts responses from agree/disagree, and respectively likely/unlikely to binary values (0/1).

We map responses “Strongly disagree”, “Disagree”, and “Somewhat disagree” to 0 and responses “Somewhat agree”, “Agree”, and “Strongly agree” to 1. Analogically, we map responses “Very unlikely” and “Unlikely” to 0 and responses “Somewhat likely”, “Likely”, and “Very likely” to 1.

*The cleaning function* drops duplicate rows, removes columns that contain non-categorical variables (such as date, text variables), keeps columns with less than 10% missing values, renames some columns for better readability, drops rows with missing values in the “travellers_number” column, maps “Yes” and “No” satisfaction responses to 1/0 in the column “satisfaction_binary”, uses both *map_agree_disagree* and *map_likely_unlikely* functions as previously described, creates dummy variables for categorical variables (such as “age”, “travellers_number”, “transport_to_from_airport”) and inserts them in the original DataFrame. Finally, it returns the clean DataFrame to further be used for our analysis. After applying this function to our DataFrame (df_non_text), we are left with 34 explanatory features.

### 2. Correlation between features

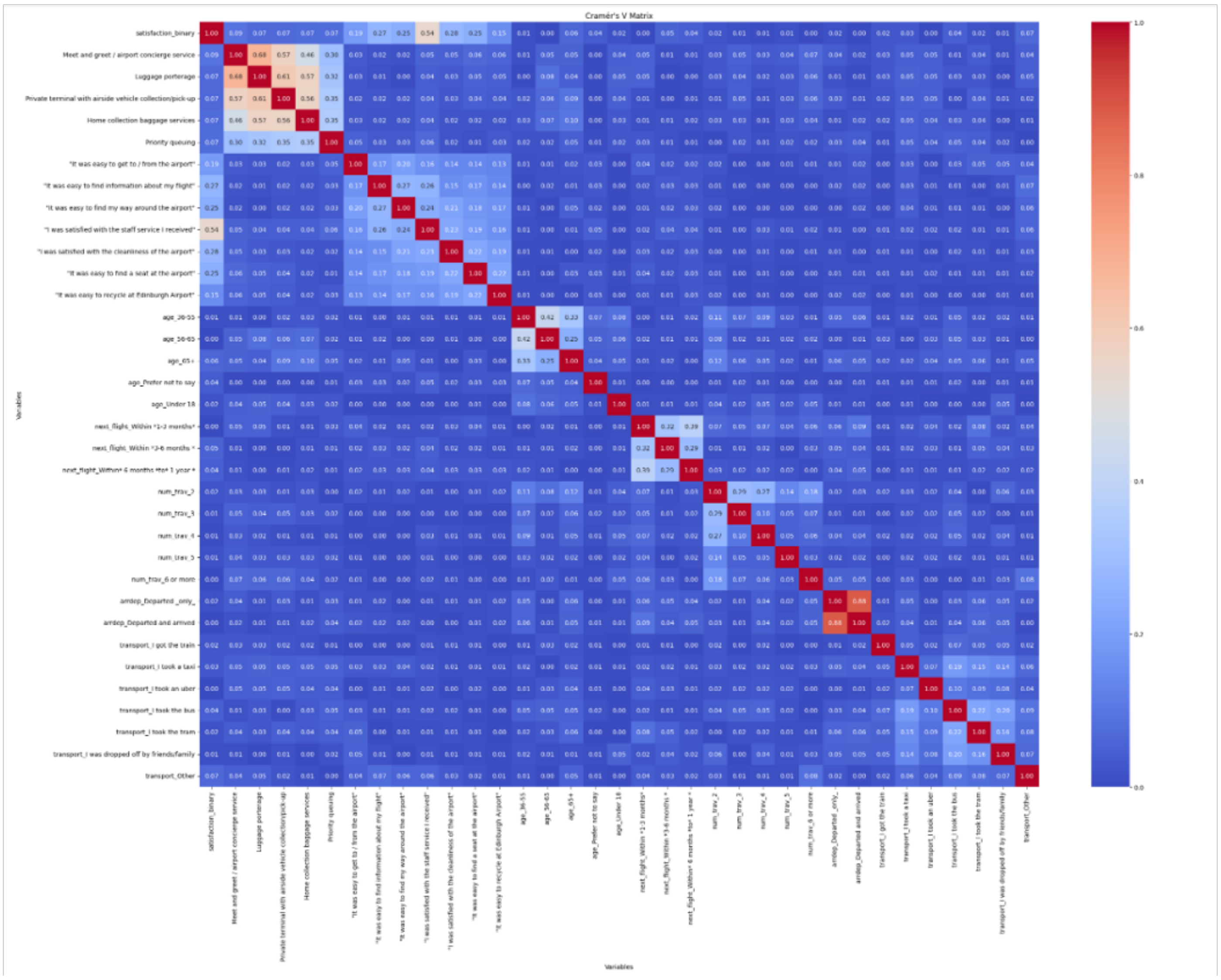
To assess the correlation between each pair of categorical variables, we perform a correlation analysis using Cramer’s V statistic.

The function *cramers_v* is used to calculate Cramér's V statistic in order to measure the strength of association between two categorical variables. As an output, it returns a value between 0 and 1 indicating the strength of the association.

The function *calculate_cramers_v_matrix* computes a matrix of Cramer’s V statistic for all pairs of categorical variables in the DataFrame. It returns a DataFrame where each cell contains the Cramer’s V statistic for the corresponding pair of variables.

Finally, we use the function *plot_cramers_v_heatmap* to provide a visual representation of the correlation between variables using a heatmap.

When applying these functions to our DataFrame (df_non_text), we see the heatmap associated to our features, with the color intensity indicating the strength of the association between variables.



Even from a glance at this heatmap, we can notice that we have some clusters of variables that are highly correlated. We have plenty of examples: “Luggage porterage” and “Meet and greet / airport concierge service”, “Home collection baggage services” and “Meet and greet / airport concierge service”, “Private terminal with airside vehicle collection/pick-up” and “Meet and greet / airport concierge service” etc.

As we prepare to train a logistic regression model on our dataset, it's crucial to address any multicollinearity among the variables. Highly correlated predictors can lead to several issues, including challenges in disentangling the individual effects of each predictor on the outcome variable. Additionally, multicollinearity can exacerbate overfitting and introduce bias into our model's inferences. By identifying and mitigating multicollinearity, we can enhance the reliability and interpretability of our logistic regression analysis.

For that matter, we iteratively determine the most correlated pair of variables, and we eliminate one of them – the one that is the least correlated with the target variable “satisfaction_binary”. We perform the same step until there are no more pairs of variables with a correlation higher than 0.3 (we use 0.3 as a threshold).

By doing this, we determined that the following variables have to be excluded in order to properly perform the logistic regression: “Luggage porterage”, “age_36-55”, “next_flight_Within *1-3 months”, ”arrdep_Departed and arrived”, “Private terminal with airside vehicle collection/pick-up”, “Home collection baggage services”, “age_56-65”, “num_trav_3”, “num_trav_4”. We exclude them from our DataFrame (df_non_text) using the function *drop_columns.* When we plot again the heatmap, we see that no pair of correlated variables (higher than 0.3) is left. We have 25 features left.

## Modelisation

### 1. Oversampling the data

After analyzing our data, we saw that the proportion of satisfied people is much higher than that of unsatisfied people (87% versus 13%). This is problematic for multiple reasons: the logistic regression is highly sensitive to class imbalance, and it tends to become biased towards the majority class, which can lead to inaccurate predictions for the minority class (minority class instances classified as majority class).

In order to solve this problem, we balance classes using SMOTE (Synthetic Minority

Over-sampling Technique). The function *oversample_data* takes as input a DataFrame containing both the features and the target variable. The target_column parameter specifies the name of the target variable. The function splits the data into training and testing sets using the specified test_size ratio, in our case 0.3. Then, it applies SMOTE to the training data to create synthetic samples for the minority class, ensuring a balanced class distribution. The function returns a tuple containing the following:

- X_train: Features used for training
- X_test: Features used for testing
- y_train: Target variable used for training
- y_test: Target variable used for testing
- os_data_X: Oversampled features
- os_data_y: Oversampled target variable

After having applied this function to our DataFrame, we noticed that the classes of the target variable become balanced: we have 50% of satisfied visitors and 50% of unsatisfied visitors.

### 2. Feature selection

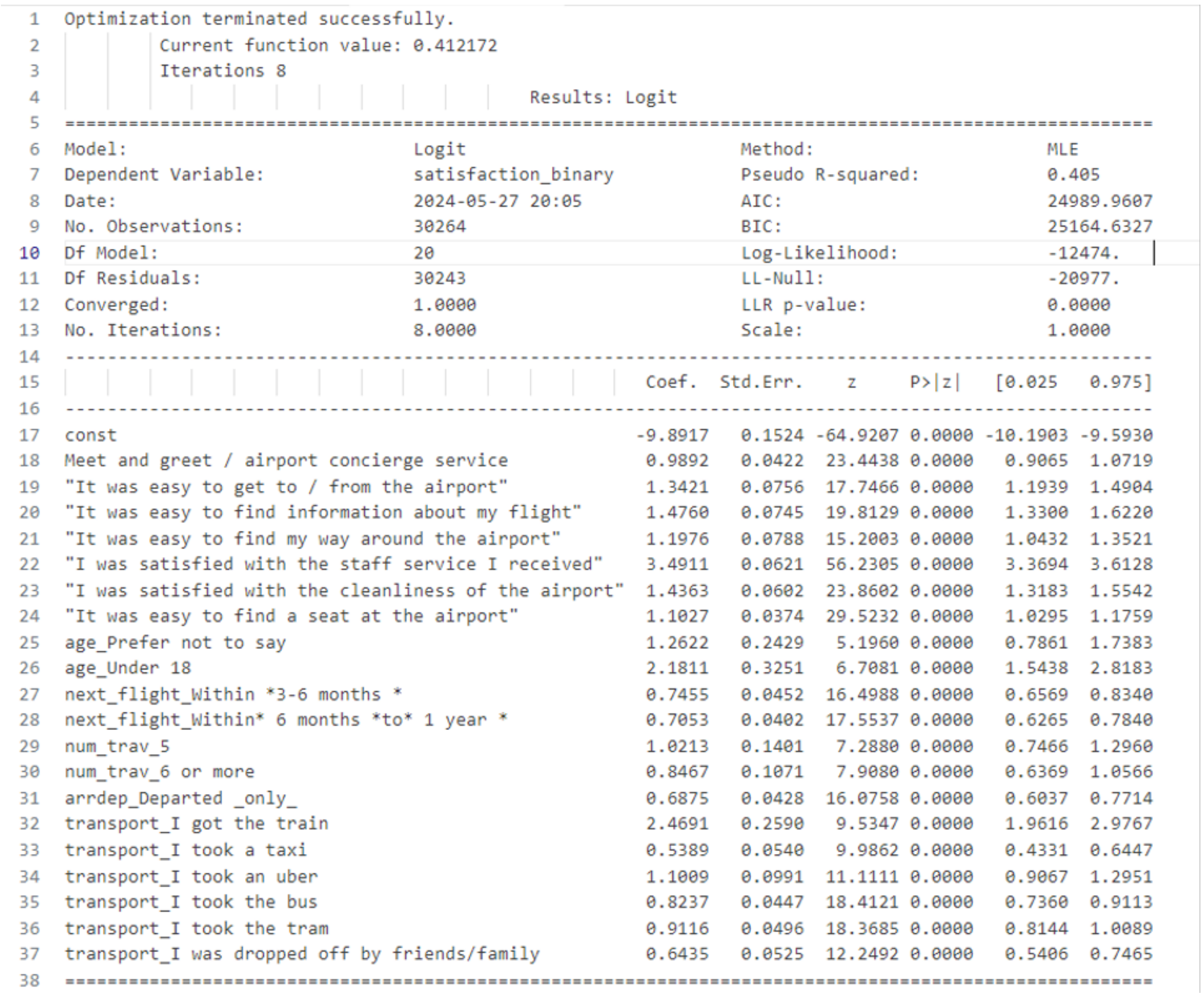
This function aims to select the optimal set of features using Recursive Feature Elimination with Cross-Validation (RFECV). It takes an estimator (Logistic Regression in our case), the feature data, and the target variable as inputs. By fitting the model to the data, it determines and returns the most relevant features, helping to improve model performance and reduce complexity. After applying this function to our dataset, only 20 features remain relevant for our analysis:

'Meet and greet / airport concierge service', 'It was easy to get to / from the airport', 'It was easy to find information about my flight', 'It was easy to find my way around the airport', 'I was satisfied with the staff service I received', 'I was satisfied with the cleanliness of the airport', 'It was easy to find a seat at the airport', 'age_Prefer not to say', 'age_Under 18', 'next_flight_Within *3-6 months *', 'next_flight_Within* 6 months *to* 1 year *', 'num_trav_5', 'num_trav_6 or more', 'arrdep_Departed _only_', 'transport_I got the train', 'transport_I took a taxi', 'transport_I took an uber', 'transport_I took the bus', 'transport_I took the tram', 'transport_I was dropped off by friends/family'. These are the features to use to train our logistic regression model.

### 3. Logit regression

The purpose of the *logit_regression* function is to fit a logistic regression model using a specified set of features from a dataset and to provide a summary of the model. In our case, the specified features are the 20 variables derived from the feature selection process. This is useful for understanding the relationship between the features (predictors) and the target variable, “satisfaction_binary”. To achieve it, we use the library “statsmodels”. After fitting the model, the function prints a detailed summary of the logistic regression, including coefficients associated to the explanatory variables, p-values, confidence intervals, and model fit statistics (AIC, BIC), as you can see in the image below.

Finally, the function returns the fitted model result object so it can be used for further analysis and prediction.



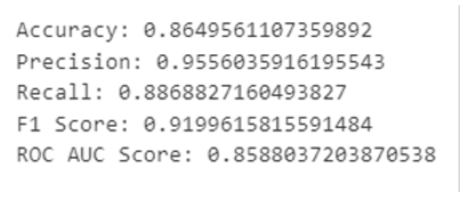
```
Optimization terminated successfully.
         Current function value: 0.412172
         Iterations 8
                                       Results: Logit
=====================================================================================================
Model:                           Logit                          Method:                 MLE
Dependent Variable:              satisfaction_binary            Pseudo R-squared:       0.405
Date:                            2024-05-27 20:05               AIC:                    24989.9607
No. Observations:                30264                          BIC:                    25164.6327
Df Model:                        20                             Log-Likelihood:         -12474.
Df Residuals:                    30243                          LL-Null:                -20977.
Converged:                       1.0000                         LLR p-value:            0.0000
No. Iterations:                  8.0000                         Scale:                  1.0000
-----------------------------------------------------------------------------------------------------
                                                                  Coef.  Std.Err.    z     P>|z|   [0.025   0.975]
-----------------------------------------------------------------------------------------------------
const                                                           -9.8917   0.1524 -64.9207 0.0000 -10.1903 -9.5930
Meet and greet / airport concierge service                       0.9892   0.0422  23.4438 0.0000   0.9065  1.0719
"It was easy to get to / from the airport"                       1.3421   0.0756  17.7466 0.0000   1.1939  1.4904
"It was easy to find information about my flight"                1.4760   0.0745  19.8129 0.0000   1.3300  1.6220
"It was easy to find my way around the airport"                  1.1976   0.0788  15.2003 0.0000   1.0432  1.3521
"I was satisfied with the staff service I received"              3.4911   0.0621  56.2305 0.0000   3.3694  3.6128
"I was satisfied with the cleanliness of the airport"            1.4363   0.0602  23.8602 0.0000   1.3183  1.5542
"It was easy to find a seat at the airport"                      1.1027   0.0374  29.5232 0.0000   1.0295  1.1759
age_Prefer not to say                                            1.2622   0.2429   5.1960 0.0000   0.7861  1.7383
age_Under 18                                                     2.1811   0.3251   6.7081 0.0000   1.5438  2.8183
next_flight_Within *3-6 months *                                 0.7455   0.0452  16.4988 0.0000   0.6569  0.8340
next_flight_Within* 6 months *to* 1 year *                       0.7053   0.0402  17.5537 0.0000   0.6265  0.7840
num_trav_5                                                       1.0213   0.1401   7.2880 0.0000   0.7466  1.2960
num_trav_6 or more                                               0.8467   0.1071   7.9080 0.0000   0.6369  1.0566
arrdep_Departed _only_                                           0.6875   0.0428  16.0758 0.0000   0.6037  0.7714
transport_I got the train                                        2.4691   0.2590   9.5347 0.0000   1.9616  2.9767
transport_I took a taxi                                          0.5389   0.0540   9.9862 0.0000   0.4331  0.6447
transport_I took an uber                                         1.1009   0.0991  11.1111 0.0000   0.9067  1.2951
transport_I took the bus                                         0.8237   0.0447  18.4121 0.0000   0.7360  0.9113
transport_I took the tram                                        0.9116   0.0496  18.3685 0.0000   0.8144  1.0089
transport_I was dropped off by friends/family                    0.6435   0.0525  12.2492 0.0000   0.5406  0.7465
=====================================================================================================
```

## Main conclusions:

- We can notice from this result table that the variable with the greatest impact on overall satisfaction level is “I was satisfied with the staff service I received”. Even though the coefficients are not directly interpretable, we can deduce that being satisfied with staff service will strongly increase the probability of being satisfied with the experience at the airport. In consequence, it is recommended to prioritize staff training to improve visitor satisfaction.
- The next strongest coefficient is associated with the variable ”transport_I got the train”. Knowing that the reference modality is “I drove”, we can deduce that taking the train rather than driving increases the probability of being satisfied with the experience at the Edinburgh airport. It could be related to the parking conditions, a problem that will further be examined through text analysis.
- We also notice that all means of transport: taxi, uber, bus, tram, friends drop off are all associated with an increase in the probability of satisfaction.
- As for other factors that highly impact the satisfaction of visitors, we observe that the ease of finding information about the flight, as well as the cleanliness of the airport seem to positively impact the probability of satisfaction.
- The impact of the variables “next_flight_Within * 3-6 months” and “next_flight_Within  6 months to 1 year*” could be affected by endogeneity problems. We could assume that the outcome variable, the satisfaction, could impact the likelihood of planning future flights from the airport.

- The logistic regression model explains approximately 40.5% of the variability in the outcome variable. It indicates that the model captures a significant portion of the variation in the data but there is still room for improvement. Some other factors, not included in the model, may also influence satisfaction.

The function *evaluate_model* is aimed at assessing the performance of a fitted logistic regression model on a test dataset using several key evaluation metrics. We use the fitted logistic regression model to predict probabilities for the test data and to convert them into binary class prediction. As usual, we apply a threshold of 0.5 as follows: if a probability of 0.5 or higher is predicted then the variable y_pred_binary will be equal to1. Otherwise, y_pred_binary will be equal to 0. To assess the performance of the logistic regression, we compute 5 metrics: accuracy, precision, recall, F1 score, and ROC AUC score, calculated using "sklearn_metrics". These metrics are aimed at comparing y_test and y_pred_binary and providing a quantitative measure of the performance of the model. Finally, this function prints and returns the metrics for further analysis.



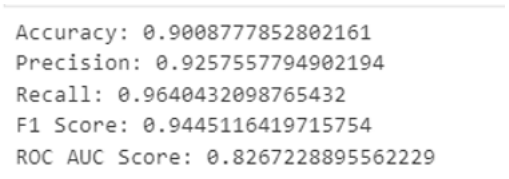
```
Accuracy: 0.8649561107359892
Precision: 0.9556035916195543
Recall: 0.8868827160493827
F1 Score: 0.9199615815591484
ROC AUC Score: 0.8588037203870538
```

- Accuracy (0.86): the model correctly predicts the satisfaction level of approximately 86% of the cases in the dataset
- Precision (0.96): it suggests that when the model predicts a customer to be satisfied, it is correct approximately 96% of the time
- Recall (0.89): it indicates that the model correctly identifies approximately 89% of all satisfied customers
- F1 Score (0.92): the F1 score is the harmonic mean of precision and recall, it provides a balance between precision and recall, with higher values indicating better overall performance. An F1 score of 0.92 suggests that the model achieves a good balance between precision and recall.
- ROC AUC (0.86): the ROC AUC (Receiver Operating Characteristic Area Under the Curve) is a measure of the model's ability to distinguish between the positive and negative classes. In our case, the model performs relatively well in ranking the predictions and distinguishing between satisfied and unsatisfied customers.

## 4. Random Forest

The purpose of the function *random_forest* is to train a Random Forest classifier on a dataset, in our case df_non_text. It also evaluates its performance using several metrics and returns key outputs for further analysis.

We first split the DataFrame into both features and outcome variable. Then, we further split it into training and test sets (30% for testing) and train the model on the training set using a classifier with 100 trees. The trained model is used to further predict data labels for the test set. To assess the performance of the model on the test set, the function prints the 5 chosen metrics: accuracy, precision, recall, F1 score, and ROC AUC score and returns a tuple containing main outputs.



```
Accuracy: 0.9008777852802161
Precision: 0.9257557794902194
Recall: 0.9640432098765432
F1 Score: 0.9445116419715754
ROC AUC Score: 0.8267228895562229
```

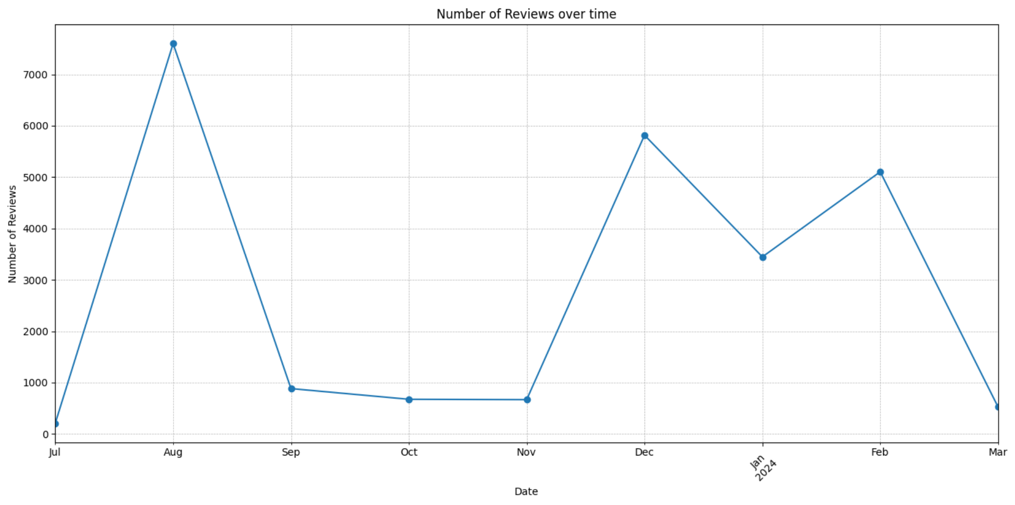
From this output table, we observe that all metrics have improved with a Random Forest classifier, compared to a Logistic Regression model. However, while the satisfaction predictions are more accurate, the interpretability of this type of model is quite poor and it does not bring any specific insight into possible improvement lines.

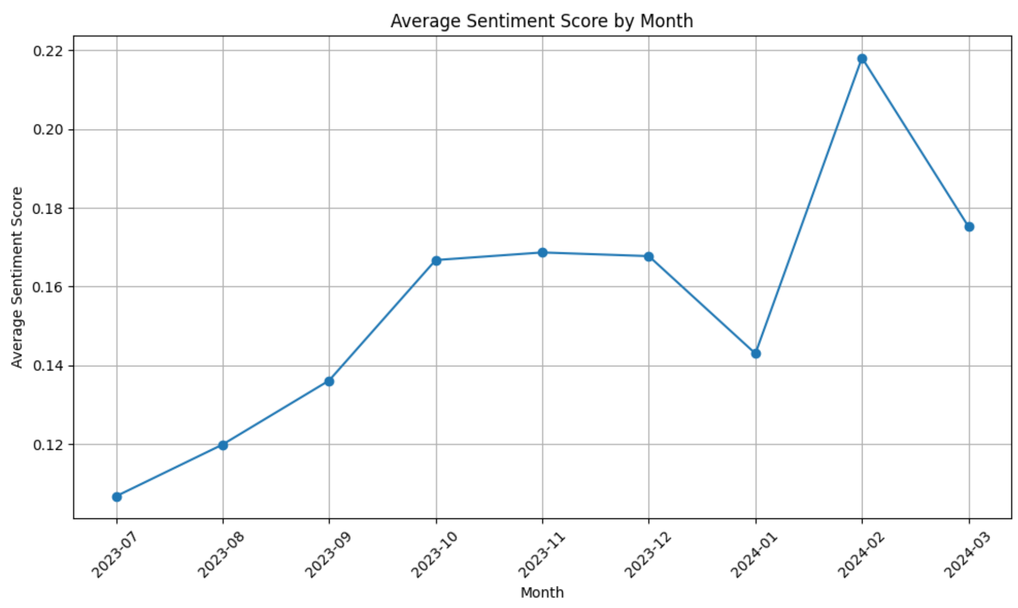
## Text variables analysis

To analyze the textual data, we utilized a function designed to transform the text into a suitable format for analysis. This transformation process ensures that the text data is appropriately processed and ready for further analysis.

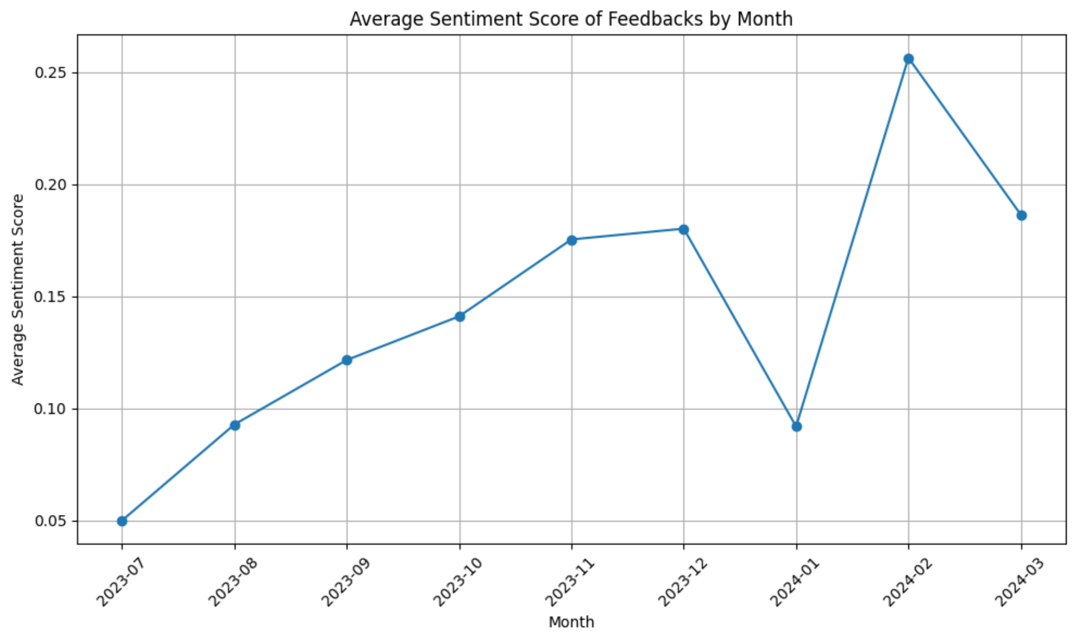
### Sentiment analysis

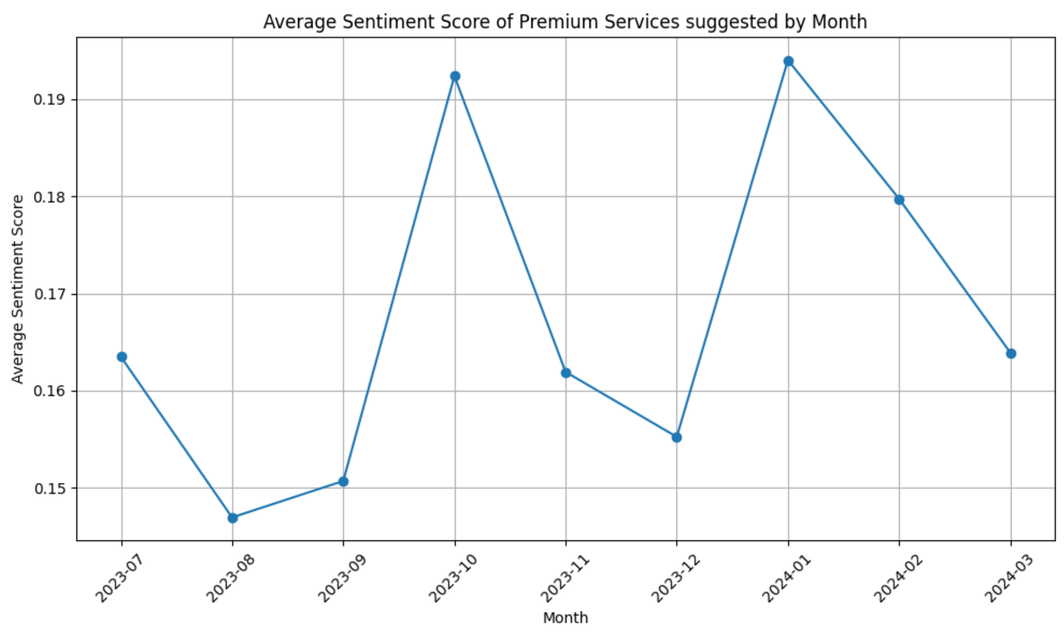
We will first analyze the sentiments by calculating polarity score of opinion by applying Vader method.  The polarity score typically ranges from -1 to 1, where -1 indicates a highly negative sentiment, 0 indicates neutral sentiment, and 1 indicates a highly positive sentiment. To understand how the average polarity score varied, we will compare between these two graphs below.





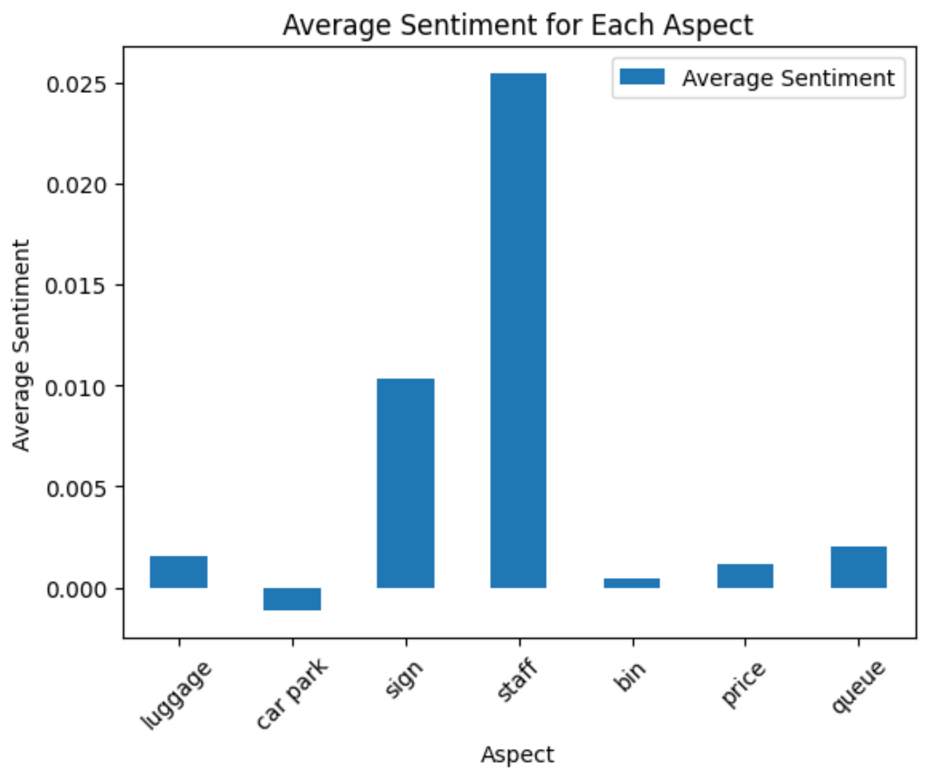
As can be seen, the sentiments seemed to be positive overall. The lowest level was witnessed in July 2023. This was the beginning of the survey, which means that there were fewer opinions during this month than in other months and could explain why the polarity score was the lowest. The positivity of sentiments increased gradually until December 2023, before dropping in January 2024. From July 2023 to August 2023, there was an increase in the number of opinions received and in the positivity of sentiments. In addition, from August 2023 to September 2023, there was a sharp decrease in the number of opinions, whereas the positivity level continued to grow. From September 2023 to October 2023, the number of opinions remained unchanged, while the positivity level kept increasing. However, in the next month, the opinions' number remained stable and so did the positivity level. From November 2023 to December 2023, there was a decrease in the number of feedbacks, while the sentiments' positivity was unchanged. This means that most opinions were positive, despite the low total number of feedbacks. From December 2024 to January 2024, a significant decrease in the number of feedbacks went hand in hand with an unsignificant decrease in the sentiments' positivity. This indicates that though the general trend of opinions was positive, the negativity level also marked an increase. From January 2024 to February 2024, there was an increase in both total number of feedbacks and the level of positivity. March 2024 is the last month of the survey. Since we only observed the first 4 days of this month, there were not many feedbacks and suggestions recorded, so the positivity level also dropped.





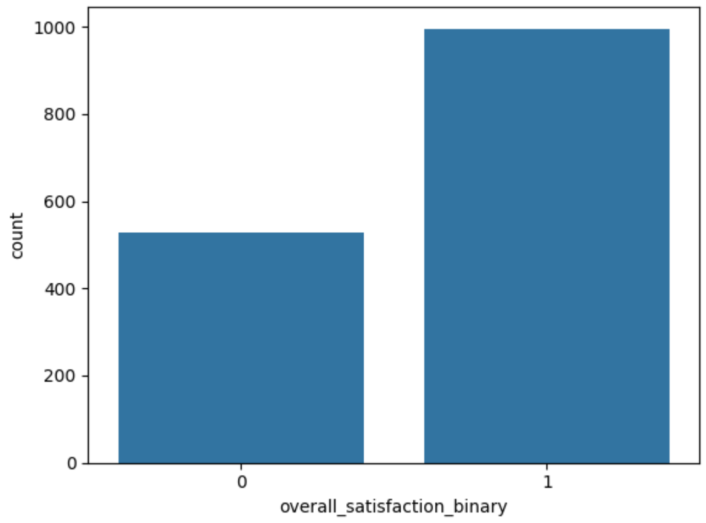
It is noticeable that the suggestions on premium services marked numerous fluctuations in terms of polarity score while the overall opinions increased gradually during 2023 and fluctuated during the first quarter of 2024. Meanwhile, the general trend was positive, though it tended to decrease during peak season.

We then try to calculate the average sentiment's trend on some aspects concerned, including luggage, car park, sign, staff, bin, price and queue. Most of the aspects received positive opinions, except for car park. Staff and sign were the two most welcomed aspects, marking the satisfaction from staff's help and the convenience of information displayed at the airport. The other aspects were positive overall but not significant.



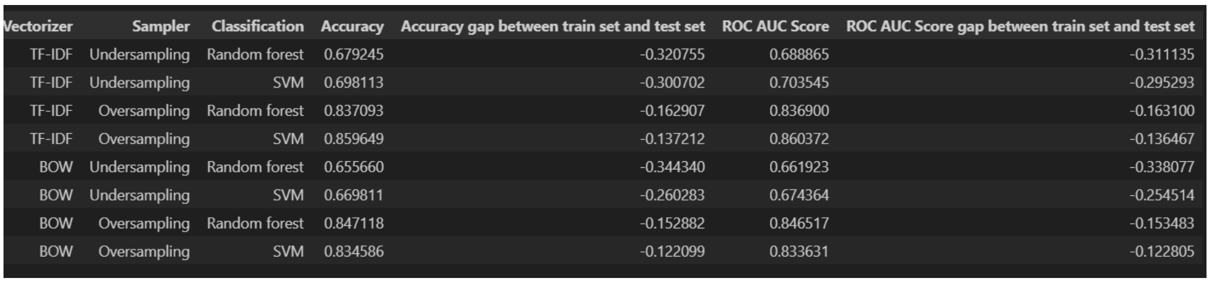
Moreover, it is worth noting that the proportion between positive feedbacks and negatives feedbacks was imbalanced, which can directly affect the overall polarity score. Therefore, to improve the service quality of the airport, although this implies a positive signal, we should focus more on negative opinions and what they are about.

We are now interested in building a model to predict the sentiments based on the two available text variables.



We first noticed that the dataset was imbalanced. This imbalance can cause biased predictions as the model would tend to predict the major class more frequently. Therefore, we will resample our dataset to solve this problem. There are then 2 ways: over-sampling, or under-sampling.

In addition, to convert text data into numerical vectors that machine learning algorithms can process, we would use either TF-IDF or Bag-of-words. When it comes to the classification, we decided to choose Random forest and SVM since these 2 methods have proved their efficiency in analyzing sentiments (Ref: Al Amrani, Yassine, Mohamed Lazaar, and Kamal Eddine El Kadiri. "Random forest and support vector machine based hybrid approach to sentiment analysis." *Procedia Computer Science* 127 (2018): 511-520.)



| Vectorizer | Sampler | Classification | Accuracy | Accuracy gap between train set and test set | ROC AUC Score | ROC AUC Score gap between train set and test set |
|---|---|---|---|---|---|---|
| TF-IDF | Undersampling | Random forest | 0.679245 | -0.320755 | 0.688865 | -0.311135 |
| TF-IDF | Undersampling | SVM | 0.698113 | -0.300702 | 0.703545 | -0.295293 |
| TF-IDF | Oversampling | Random forest | 0.837093 | -0.162907 | 0.836900 | -0.163100 |
| TF-IDF | Oversampling | SVM | 0.859649 | -0.137212 | 0.860372 | -0.136467 |
| BOW | Undersampling | Random forest | 0.655660 | -0.344340 | 0.661923 | -0.338077 |
| BOW | Undersampling | SVM | 0.669811 | -0.260283 | 0.674364 | -0.254514 |
| BOW | Oversampling | Random forest | 0.847118 | -0.152882 | 0.846517 | -0.153483 |
| BOW | Oversampling | SVM | 0.834586 | -0.122099 | 0.833631 | -0.122805 |

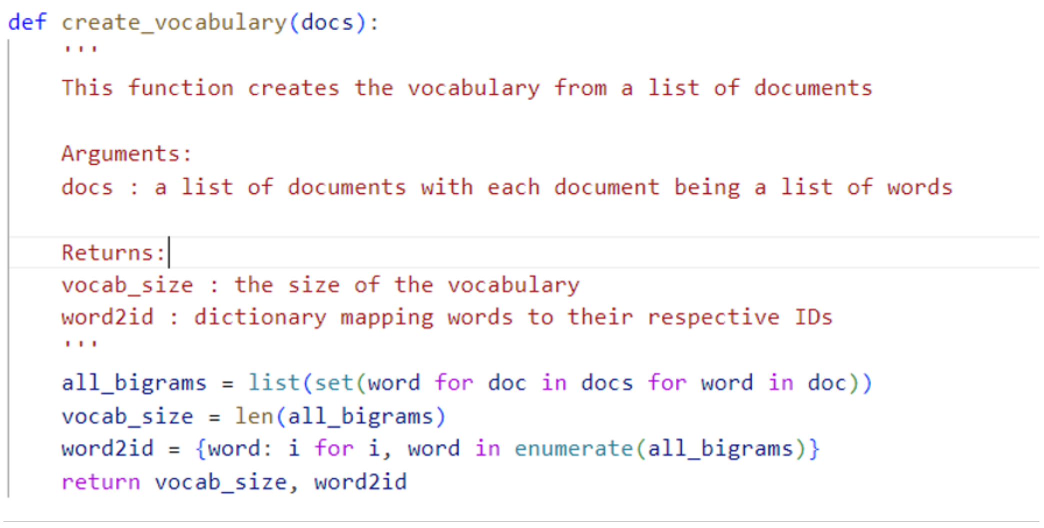
Judging based on Accuracy score and ROC AUC score, we can conclude that the best combination to predict the sentiments in our case is TF-IDF + Oversampling + SVM with an accuracy score of roughly 0.86 and an AUC ROC score of around 0.86.

## Topic modeling

### Detailed description of Topic Modeling from scratch

Remark: Before constructing the topic modeling functions, we performed preprocessing and tokenization using bigrams.

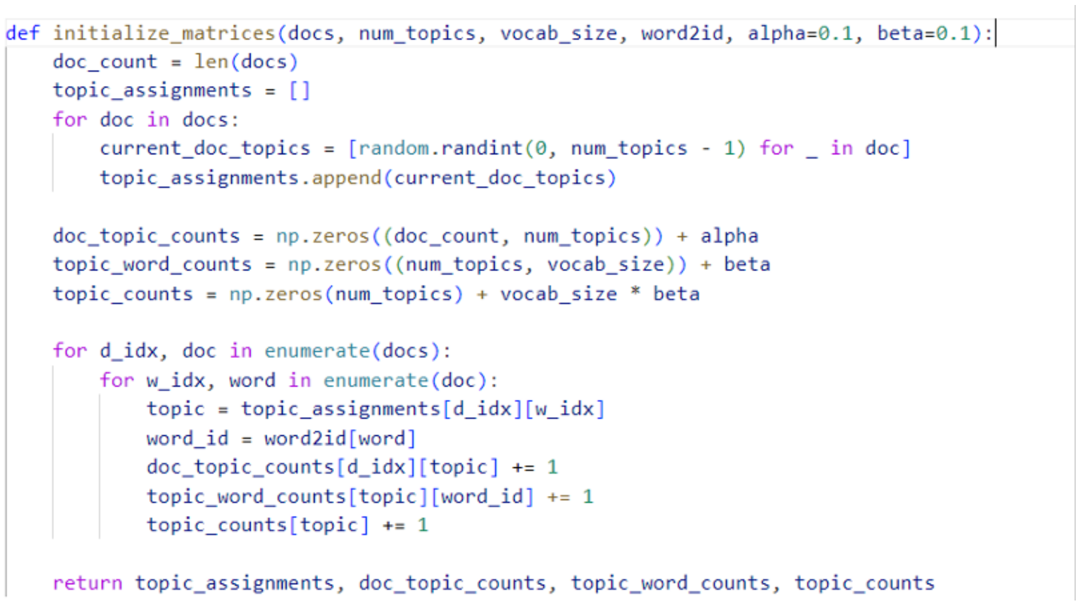
1. **The function create_vocabulary**



```python
def create_vocabulary(docs):
    '''
    This function creates the vocabulary from a list of documents

    Arguments:
    docs : a list of documents with each document being a list of words

    Returns:
    vocab_size : the size of the vocabulary
    word2id : dictionary mapping words to their respective IDs
    '''
    all_bigrams = list(set(word for doc in docs for word in doc))
    vocab_size = len(all_bigrams)
    word2id = {word: i for i, word in enumerate(all_bigrams)}
    return vocab_size, word2id
```

This function generates vocabulary from a given list of documents. Each document is represented as a list of bigrams in our case. The function extracts all unique bigrams from the documents, assigns a unique numerical ID to each bigram, and returns the size of the vocabulary along with a dictionary that maps each bigram to its corresponding ID. This is a fundamental step for the implementation of Topic Modeling.

2. **The function *initialize_matrices***

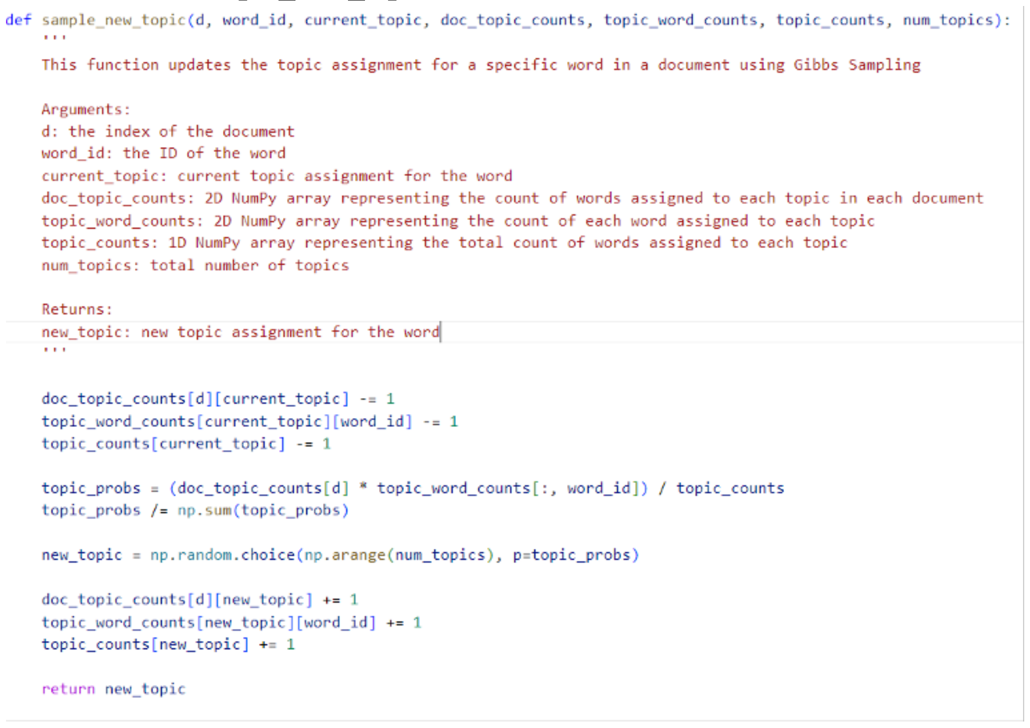


```python
def initialize_matrices(docs, num_topics, vocab_size, word2id, alpha=0.1, beta=0.1):
    doc_count = len(docs)
    topic_assignments = []
    for doc in docs:
        current_doc_topics = [random.randint(0, num_topics - 1) for _ in doc]
        topic_assignments.append(current_doc_topics)

    doc_topic_counts = np.zeros((doc_count, num_topics)) + alpha
    topic_word_counts = np.zeros((num_topics, vocab_size)) + beta
    topic_counts = np.zeros(num_topics) + vocab_size * beta

    for d_idx, doc in enumerate(docs):
        for w_idx, word in enumerate(doc):
            topic = topic_assignments[d_idx][w_idx]
            word_id = word2id[word]
            doc_topic_counts[d_idx][topic] += 1
            topic_word_counts[topic][word_id] += 1
            topic_counts[topic] += 1

    return topic_assignments, doc_topic_counts, topic_word_counts, topic_counts
```

The goal of this function is to set up the initial conditions for topic modeling using Gibbs Sampling. It assigns random topics to bigrams in documents and creates matrices to keep track of how many bigrams are assigned to each topic and how topics are distributed across documents.

To summarize, we randomly assign topics to each bigram in every document. Then, we count how many bigrams are assigned to each topic in each document (“doc_topic_counts”) and how many times each bigram is assigned to each topic (“topic_word_counts”). Lastly, we keep track of the total count of bigrams assigned to each topic (“topic_counts”).

### 3. The function *sample_new_topic*



```python
def sample_new_topic(d, word_id, current_topic, doc_topic_counts, topic_word_counts, topic_counts, num_topics):
    '''
    This function updates the topic assignment for a specific word in a document using Gibbs Sampling

    Arguments:
    d: the index of the document
    word_id: the ID of the word
    current_topic: current topic assignment for the word
    doc_topic_counts: 2D NumPy array representing the count of words assigned to each topic in each document
    topic_word_counts: 2D NumPy array representing the count of each word assigned to each topic
    topic_counts: 1D NumPy array representing the total count of words assigned to each topic
    num_topics: total number of topics

    Returns:
    new_topic: new topic assignment for the word
    '''

    doc_topic_counts[d][current_topic] -= 1
    topic_word_counts[current_topic][word_id] -= 1
    topic_counts[current_topic] -= 1

    topic_probs = (doc_topic_counts[d] * topic_word_counts[:, word_id]) / topic_counts
    topic_probs /= np.sum(topic_probs)

    new_topic = np.random.choice(np.arange(num_topics), p=topic_probs)

    doc_topic_counts[d][new_topic] += 1
    topic_word_counts[new_topic][word_id] += 1
    topic_counts[new_topic] += 1

    return new_topic
```

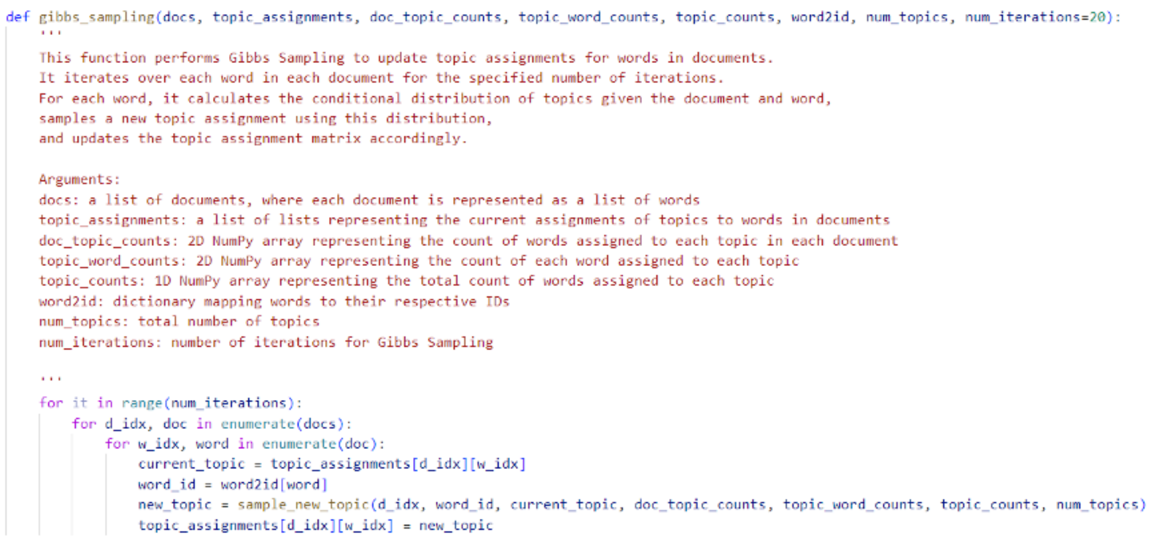
purpose of this function is to update the topic assignment for a specific bigram in a document using Gibbs Sampling, which is a technique commonly used in topic modeling.

This function adjusts the topic assignment for a particular bigram within a document based on the current state of the topic assignments and the underlying topic-bigram distributions. It employs Gibbs Sampling, a probabilistic method, to iteratively update the topic assignments.

To achieve this, the function first adjusts the count matrices to remove the contribution of the current topic assignment for the bigram. Then, it calculates the conditional probability distribution of topics given the bigram's context in the document. Using these probabilities, it samples a new topic assignment for the bigram.

After updating the count matrices with the new topic assignment, the function returns the newly assigned topic for the bigram.  To conclude, topics are reassigned

### 4. The function *gibbs_sampling*



```python
def gibbs_sampling(docs, topic_assignments, doc_topic_counts, topic_word_counts, topic_counts, word2id, num_topics, num_iterations=20):
    '''
    This function performs Gibbs Sampling to update topic assignments for words in documents.
    It iterates over each word in each document for the specified number of iterations.
    For each word, it calculates the conditional distribution of topics given the document and word,
    samples a new topic assignment using this distribution,
    and updates the topic assignment matrix accordingly.

    Arguments:
    docs: a list of documents, where each document is represented as a list of words
    topic_assignments: a list of lists representing the current assignments of topics to words in documents
    doc_topic_counts: 2D NumPy array representing the count of words assigned to each topic in each document
    topic_word_counts: 2D NumPy array representing the count of each word assigned to each topic
    topic_counts: 1D NumPy array representing the total count of words assigned to each topic
    word2id: dictionary mapping words to their respective IDs
    num_topics: total number of topics
    num_iterations: number of iterations for Gibbs Sampling

    '''
    for it in range(num_iterations):
        for d_idx, doc in enumerate(docs):
            for w_idx, word in enumerate(doc):
                current_topic = topic_assignments[d_idx][w_idx]
                word_id = word2id[word]
                new_topic = sample_new_topic(d_idx, word_id, current_topic, doc_topic_counts, topic_word_counts, topic_counts, num_topics)
                topic_assignments[d_idx][w_idx] = new_topic
```

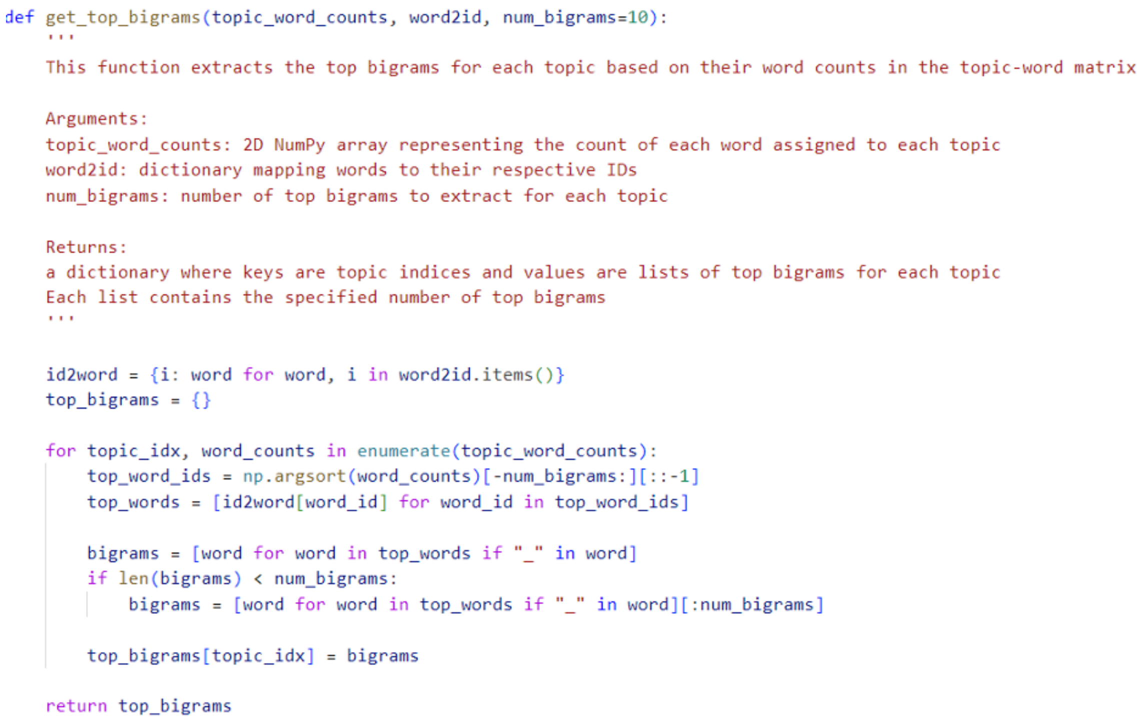
This This function implements the Gibbs Sampling algorithm for updating topic assignments.

For each iteration, the function iterates over each bigram in each document. It considers every bigram in every document in the corpus. For each bigram, it calculates the conditional distribution of topics given the document and the bigram. This distribution represents the likelihood of each topic being assigned to the bigram, based on the current state of the model. Using this calculated distribution, it samples a new topic assignment for the bigram. So, it randomly selects a topic based on the probabilities determined by the conditional distribution. Once a new topic is sampled, it updates the topic assignment matrix (“topic_assignments”) accordingly by assigning the sampled topic to the bigram at its corresponding position in the matrix.

The same process is repeated for the specified number of iterations, thereby refining its topic assignments based on the observed data.

Overall, this function facilitates the core process of Gibbs Sampling, enabling the topic model to gradually converge towards a stable representation of topics in the corpus.

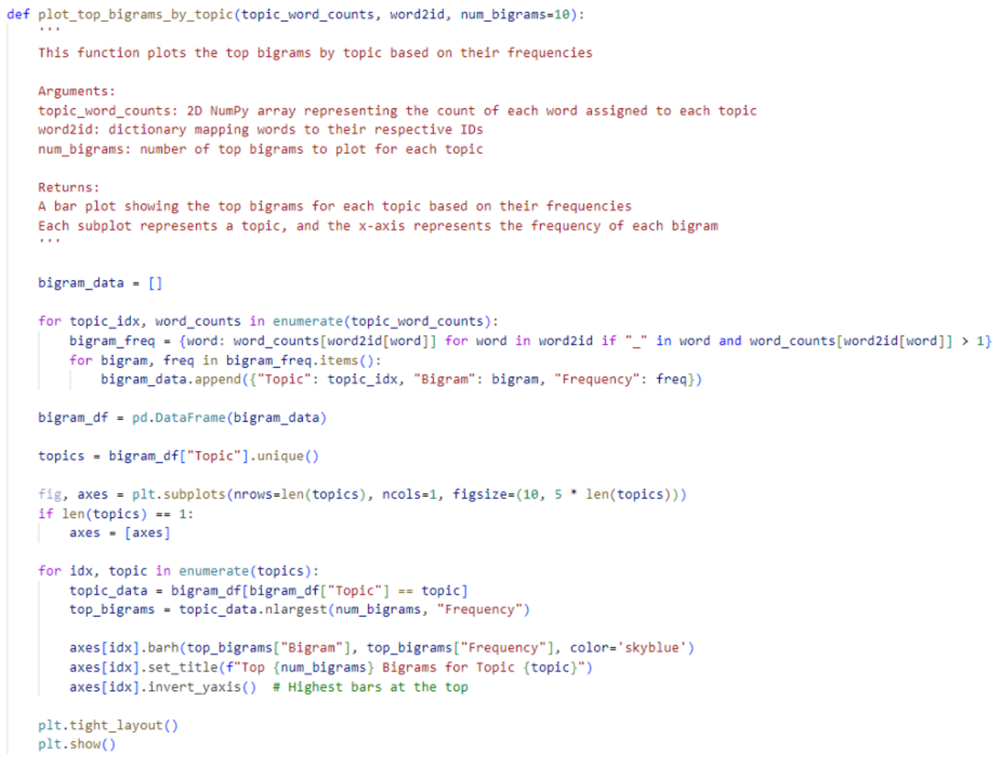
### 5. The function *get_top_bigrams*



```python
def get_top_bigrams(topic_word_counts, word2id, num_bigrams=10):
    '''
    This function extracts the top bigrams for each topic based on their word counts in the topic-word matrix

    Arguments:
    topic_word_counts: 2D NumPy array representing the count of each word assigned to each topic
    word2id: dictionary mapping words to their respective IDs
    num_bigrams: number of top bigrams to extract for each topic

    Returns:
    a dictionary where keys are topic indices and values are lists of top bigrams for each topic
    Each list contains the specified number of top bigrams
    '''

    id2word = {i: word for word, i in word2id.items()}
    top_bigrams = {}

    for topic_idx, word_counts in enumerate(topic_word_counts):
        top_word_ids = np.argsort(word_counts)[-num_bigrams:][::-1]
        top_words = [id2word[word_id] for word_id in top_word_ids]

        bigrams = [word for word in top_words if "_" in word]
        if len(bigrams) < num_bigrams:
            bigrams = [word for word in top_words if "_" in word][:num_bigrams]

        top_bigrams[topic_idx] = bigrams

    return top_bigrams
```

This function displays the top bigrams for each topic based on their word counts in the topic-word matrix. It takes as input the “topic_word_counts” matrix, which represents the count of each word assigned to each topic, and the “word2id” dictionary, mapping words to their IDs. The function returns a dictionary where the keys are topic indices, and the values are lists of top bigrams for each topic. Each list contains the specified number of top bigrams, as determined by the “num_bigrams” parameter.

## *6.* The function *plot_top_bigrams_by_topic*

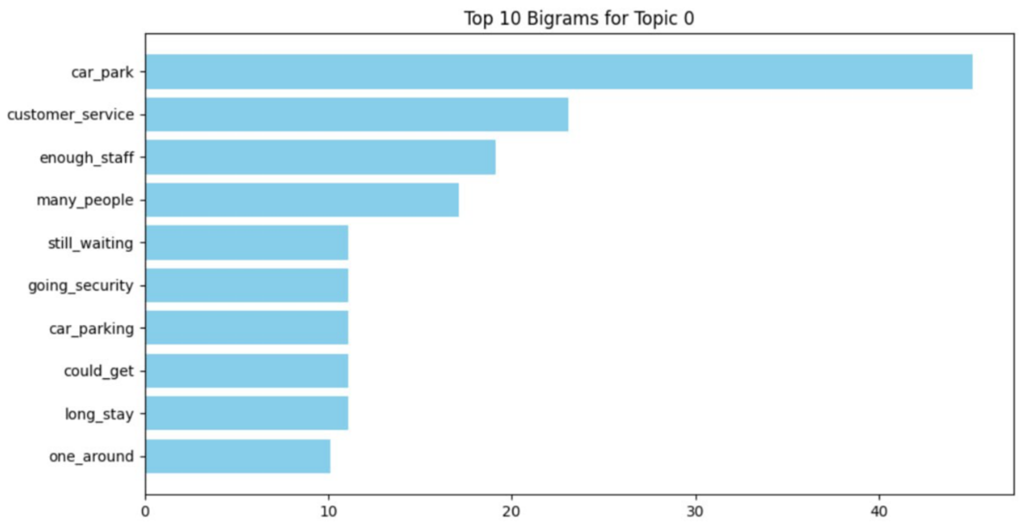


```python
def plot_top_bigrams_by_topic(topic_word_counts, word2id, num_bigrams=10):
    '''
    This function plots the top bigrams by topic based on their frequencies

    Arguments:
    topic_word_counts: 2D NumPy array representing the count of each word assigned to each topic
    word2id: dictionary mapping words to their respective IDs
    num_bigrams: number of top bigrams to plot for each topic

    Returns:
    A bar plot showing the top bigrams for each topic based on their frequencies
    Each subplot represents a topic, and the x-axis represents the frequency of each bigram
    '''

    bigram_data = []

    for topic_idx, word_counts in enumerate(topic_word_counts):
        bigram_freq = {word: word_counts[word2id[word]] for word in word2id if "_" in word and word_counts[word2id[word]] > 1}
        for bigram, freq in bigram_freq.items():
            bigram_data.append({"Topic": topic_idx, "Bigram": bigram, "Frequency": freq})

    bigram_df = pd.DataFrame(bigram_data)

    topics = bigram_df["Topic"].unique()

    fig, axes = plt.subplots(nrows=len(topics), ncols=1, figsize=(10, 5 * len(topics)))
    if len(topics) == 1:
        axes = [axes]

    for idx, topic in enumerate(topics):
        topic_data = bigram_df[bigram_df["Topic"] == topic]
        top_bigrams = topic_data.nlargest(num_bigrams, "Frequency")

        axes[idx].barh(top_bigrams["Bigram"], top_bigrams["Frequency"], color='skyblue')
        axes[idx].set_title(f"Top {num_bigrams} Bigrams for Topic {topic}")
        axes[idx].invert_yaxis()  # Highest bars at the top

    plt.tight_layout()
    plt.show()
```

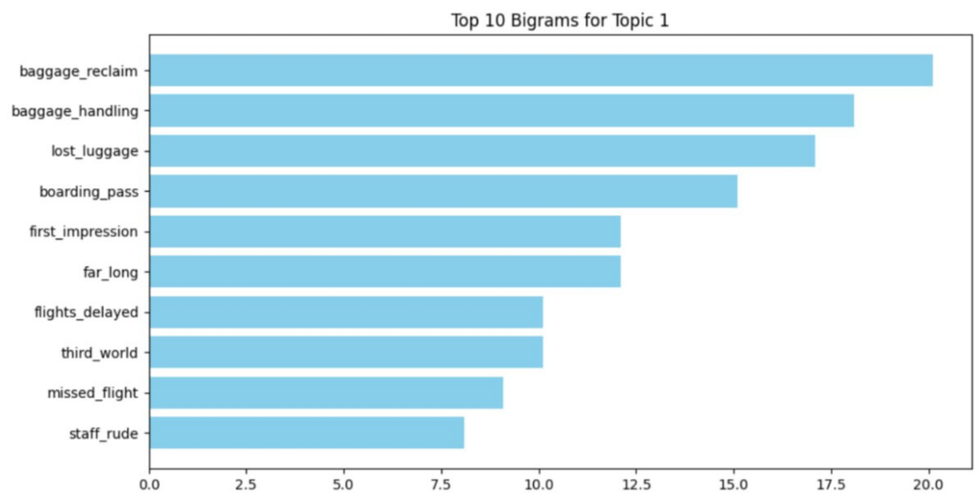
This function creates bar plots showing the top bigrams for each topic based on their frequencies. It first extracts the frequency of each bigram for each topic from the “topic_word_counts” array. Then, it creates a DataFrame “bigram_df” containing the topic index, bigram, and the associated frequency. The function finally plots a horizontal bar chart for each topic showing the top 10 bigrams based on their frequencies. As an output, we get a series of bar plots, each representing a topic. The x-axis shows the frequency of each bigram, and the y-axis shows the bigrams.

As a result of our topic modeling, we identified six main themes that airport visitors frequently discuss:

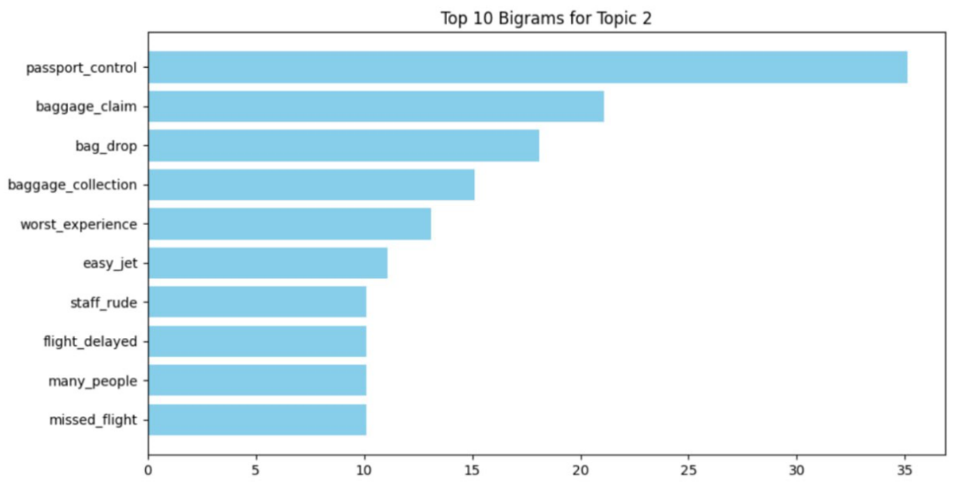
***Passport Control and Security***: This topic highlights two critical aspects of the airport experience—passport control and security. The frequent phrases "far long" and "people waiting" suggest significant issues with passenger wait times. Additionally, the mention of "special assistance" implies a potential inadequacy in the support provided during these processes.



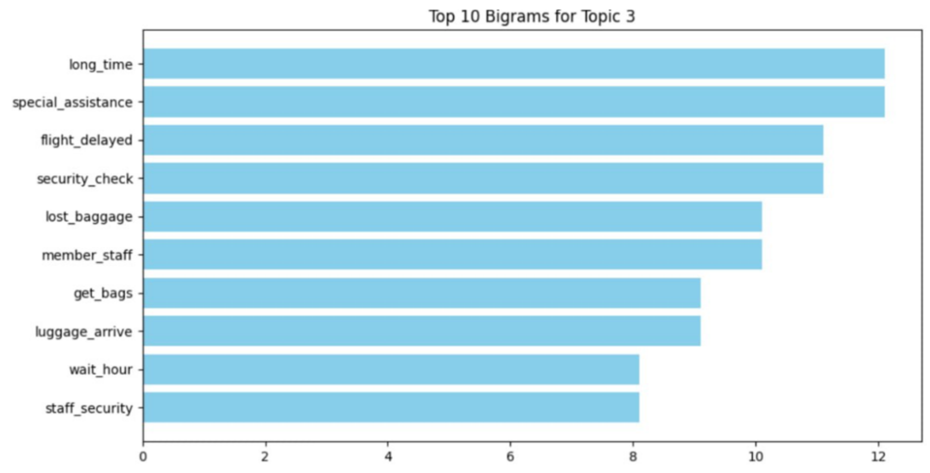
***Luggage Delays:*** This topic centers around the frustrations related to luggage, particularly the extended wait times after passport control. The expressions "long time" and "check staff" indicate a common concern about the efficiency and availability of luggage handling personnel.



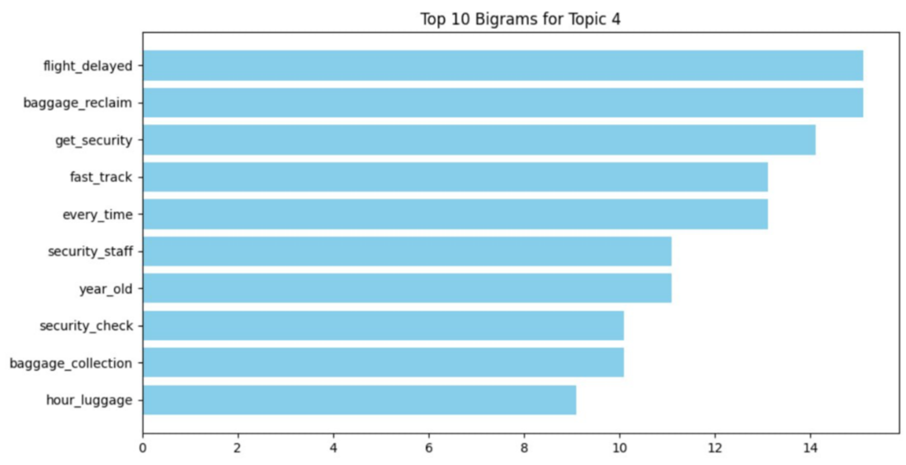
***Staff Rudeness at Passport Control***: This topic suggests a recurring issue with the demeanor of the staff at passport control. Phrases like "extremely rude," "border control," and "passport control" point to numerous complaints about the unprofessional behavior of the personnel in this area.



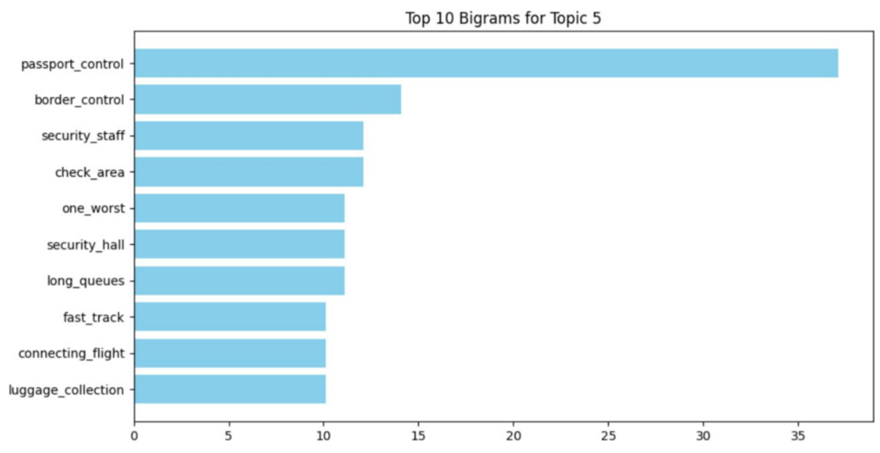
***Flight Delays:*** This theme revolves around the issue of delayed flights, though it is challenging to interpret with precision. It suggests a general dissatisfaction with the timeliness of flight schedules.



**Crowded and Unfriendly Passport Control:** Similar to the third topic, this one emphasizes the rudeness of the staff and the overcrowded conditions at passport control. The repeated concerns suggest that this is a significant pain point for travelers.



**Luggage Issues:** This topic again focuses on luggage problems, highlighting issues such as "lost baggage," "baggage collection," and the ineffectiveness of the "Swissport staff." The recurring mention of "staff unhelpful" suggests a common experience of insufficient assistance when dealing with lost luggage.

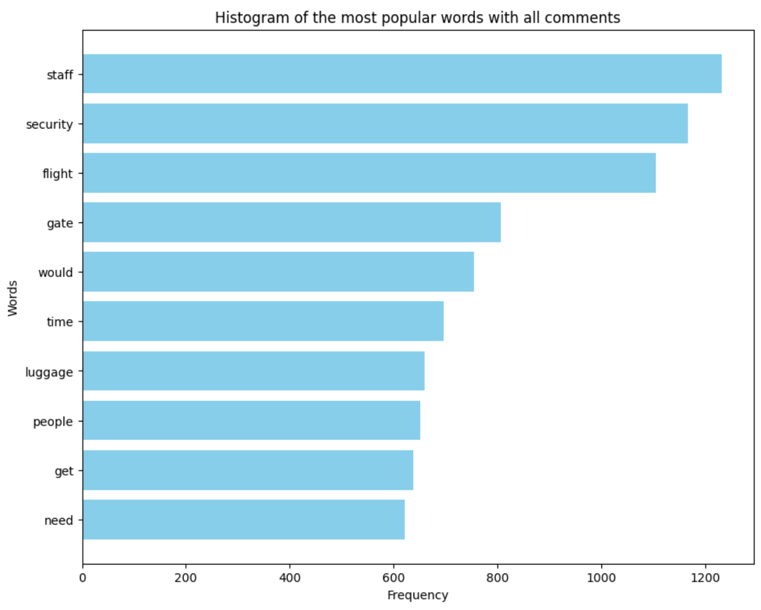


Finally, we identified a variety of topics that airport visitors care about and gained some insight into their opinions, though it is still not enough to fully grasp the essence of their frustrations. Notably, these results are consistent with those obtained from analyzing the topics using the LDA algorithm (watch notebook “main”), indicating that our algorithm performs well.

## Analysis of comments with the most popular words

### Unigrams

Finally, we decided to analyze the comments by focusing on the most frequently occurring words to better understand what travelers are saying. This approach is similar to topic modeling, but allows to delve into the essence of passenger dissatisfaction.



The first task was to identify the most frequent words. The histogram provides a general overview of the comments, showing that people often talk about "staff," "security," and "flight." Additionally, we noticed that the word "would" is used frequently.

### "Would"

```
Table of the most popular words and bigrams with would:

    Word  Frequency           Bigram  Frequency
  flight        178         like see         27
   staff        170      food option         22
    gate        151     water bottle         18
security        136   flight delayed         18
    like        110         duty free        16
   could        110 passport control         13
  people        103     staff member         13
    good        101     smoking area         12
    area         99         car park         12
    time         98   water fountain         12
     bag         96       fast track         11
     get         92     member staff         11
     one         90        could find        11
    nice         85   security staff         11
 service         79      hour flight         11
```

We decided to examine comments containing the word "would" in more detail. This word is often used with "staff," "security," and "service," suggesting that people are making recommendations for improvements in these areas, indicating some level of dissatisfaction. Also, we can notice that people propose to do smoking area, because at the airport it is impossible to smoke after the security check.

```
Table of the most popular words and bigrams with food option:

       Word  Frequency                Bigram  Frequency
     gluten         13           gluten free         11
       free         13          water bottle          3
   security         13            could find          3
      great         10            see gluten          3
    healthy         10      vegetarian vegan          2
 restaurant         10         enough gluten          2
      vegan          9          vegan please          2
       good          9           burger king          2
       time          8      price restaurant          2
       also          7              low cost          2
        one          7        flight delayed          2
       find          7 dietary requirement          2
       area          7         security area          2
  available          7          seating area          2
      could          6             area also          2
```

Looking at bigrams with "would," we found that "food options" is a common topic. Travelers are suggesting the addition of more restaurants or specific food products, such as vegetarian, gluten-free, or healthier options.

Additionally, "water bottle" is another frequently mentioned term. From the comments, it is clear that people are requesting the installation of stations where they can refill their water bottles.

"*A water bottle fill-up station would be great. Or if there is on better signage to the water station*."

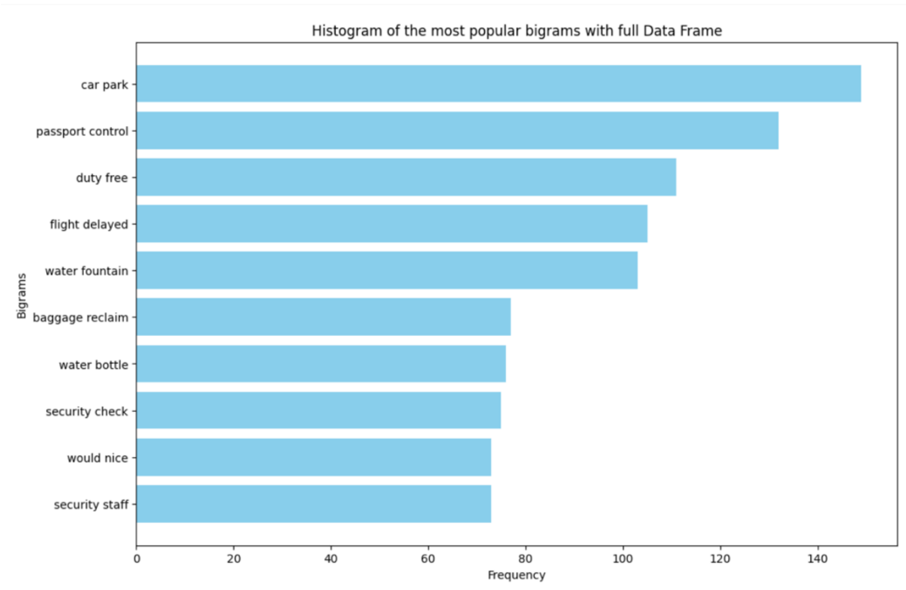
*" Additional water bottle filling station would be nice."*

**"Staff"**

The word "staff" frequently appears in the comments, prompting us to analyze feedback related to this topic. The sentiments are evenly divided, however, there is a notable quantity of negative remarks. Specifically, 174 individuals mentioned "rude" in conjunction with "staff." Referring to Topic 2, it becomes clear that the rudeness is particularly associated with the staff at passport control and security.

**Bigrams analysis**

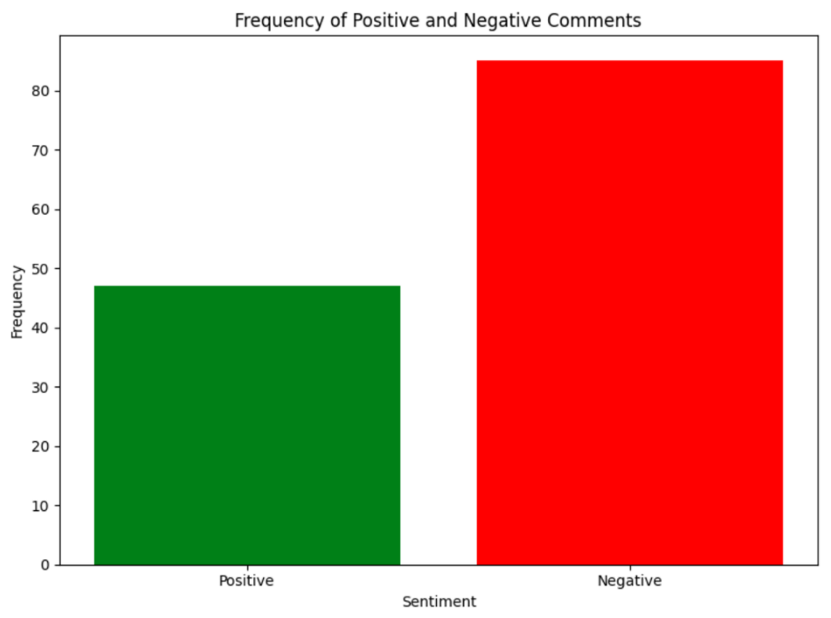
However, we believe that analyzing bigrams (two-word combinations) gives a better understanding of the text content.



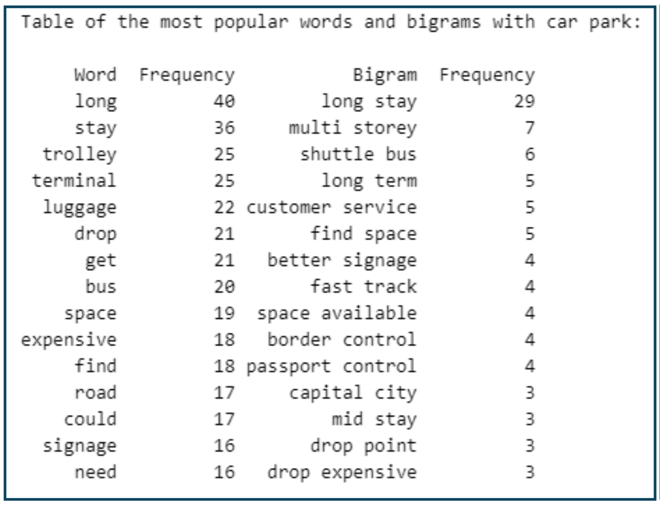
**“Car park”**

Firstly, we focused on the most common bigram: "car park." Approximately 150 people mentioned it, so we took a closer look at their comments.

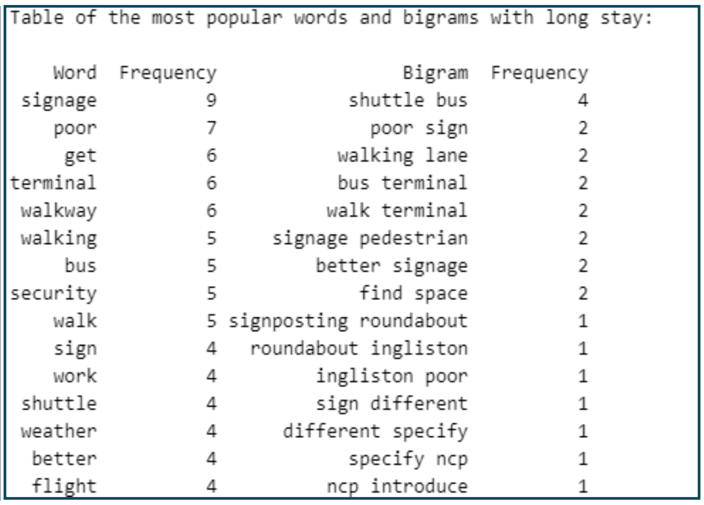
First, we examined people's sentiments on the subject. The analysis revealed that the comments were predominantly negative, indicating potential issues with the parking facilities.



Next, we analyzed the comments containing "car park" and looked at the frequency of words and bigrams, excluding the words "car," "park," and "parking" from our analysis as they were the most frequent.



```
Table of the most popular words and bigrams with car park:

     Word  Frequency             Bigram  Frequency
     long         40          long stay         29
     stay         36       multi storey          7
  trolley         25        shuttle bus          6
 terminal         25          long term          5
  luggage         22   customer service          5
     drop         21         find space          5
      get         21     better signage          4
      bus         20         fast track          4
    space         19    space available          4
expensive         18     border control          4
     find         18   passport control          4
     road         17       capital city          3
    could         17           mid stay          3
  signage         16         drop point          3
     need         16     drop expensive          3
```



```
Table of the most popular words and bigrams with long stay:

    Word  Frequency                 Bigram  Frequency
 signage          9            shuttle bus          4
    poor          7              poor sign          2
     get          6           walking lane          2
terminal          6           bus terminal          2
 walkway          6          walk terminal          2
 walking          5     signage pedestrian          2
     bus          5         better signage          2
security          5             find space          2
    walk          5  signposting roundabout          1
    sign          4    roundabout ingliston          1
    work          4         ingliston poor          1
 shuttle          4         sign different          1
 weather          4      different specify          1
  better          4            specify ncp          1
  flight          4          ncp introduce          1
```

**Car park**

**Long stay**

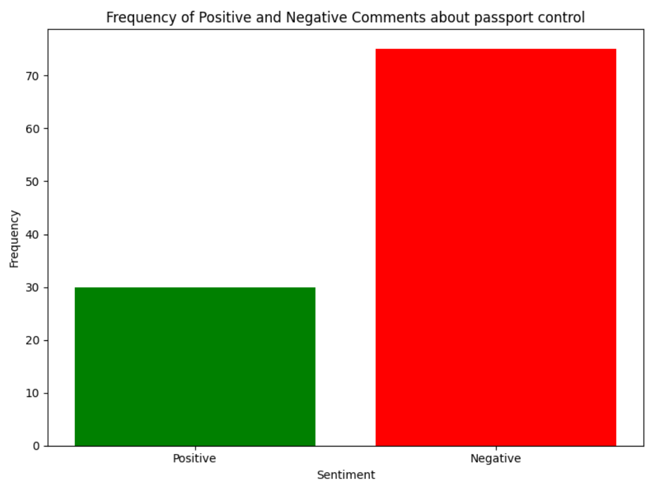
We found that "long stay" was often mentioned, with 30 people discussing this topic. Many comments about long stay parking highlighted poor signage, using the word "sign" frequently. The bigram "shuttle bus" was also commonly mentioned.

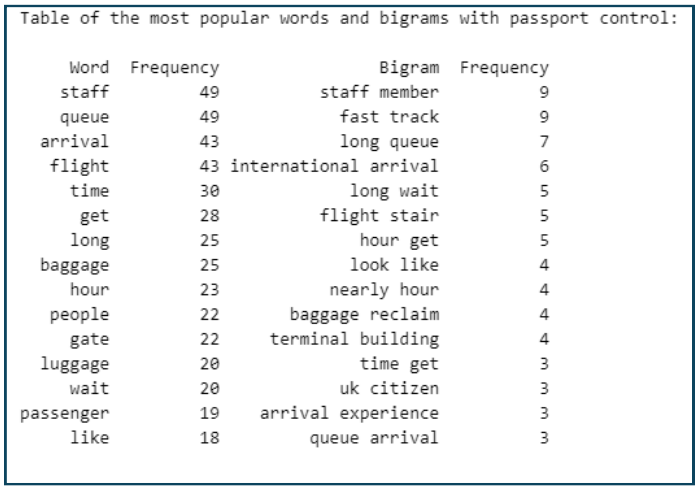
*“The signage to get to the long stay car park is really bad”*

*“Why isn’t there a shuttle bus between the long stay car park and the terminal? It was lashing with rain, howling wind, poor signage, uncovered walkways, 4 inches deep puddles, poor lighting…. Enough.”*

So, many respondents noted the poor signage for long stay parking, stating that it was difficult to find directions to it. Additionally, people complained about the challenging route between the long stay parking lot and the airport. They requested the organization of a shuttle bus service between the parking lot and the airport to alleviate these issues.

**“passport control”**





```
Table of the most popular words and bigrams with passport control:

     Word  Frequency              Bigram  Frequency
    staff         49        staff member          9
    queue         49          fast track          9
  arrival         43          long queue          7
   flight         43 international arrival        6
     time         30           long wait          5
      get         28         flight stair          5
     long         25            hour get          5
  baggage         25           look like          4
     hour         23         nearly hour          4
   people         22     baggage reclaim          4
     gate         22    terminal building         4
  luggage         20            time get          3
     wait         20          uk citizen          3
passenger         19   arrival experience         3
     like         18       queue arrival          3
```

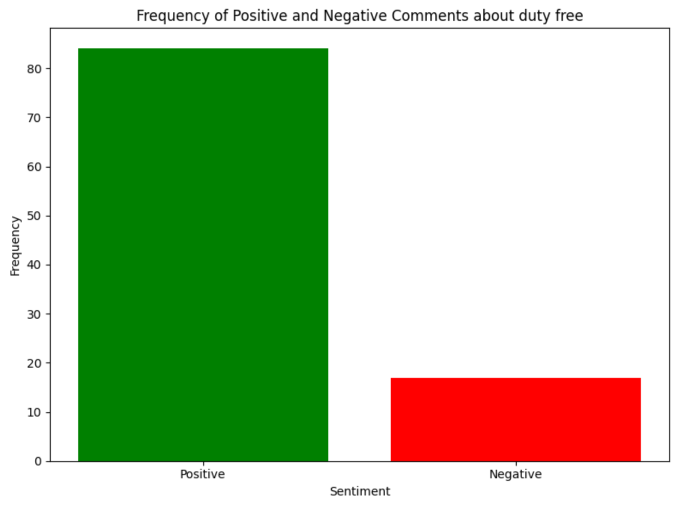
Upon reviewing feedback about passport control, it is evident that negative comments dominate. Many people express dissatisfaction with the long queues, frequently using terms like "queue"(49), "long"(25) and "wait"(20). (That refers us to the topic 1) Additionally, there are numerous complaints about the staff, with mentions of workers shouting and calls for more staff at the border. Furthermore, there is significant discontent regarding the fast track service, as it is not accessible to certain groups, such as the disabled and children. ((This we could see in the topic 3 and 5)

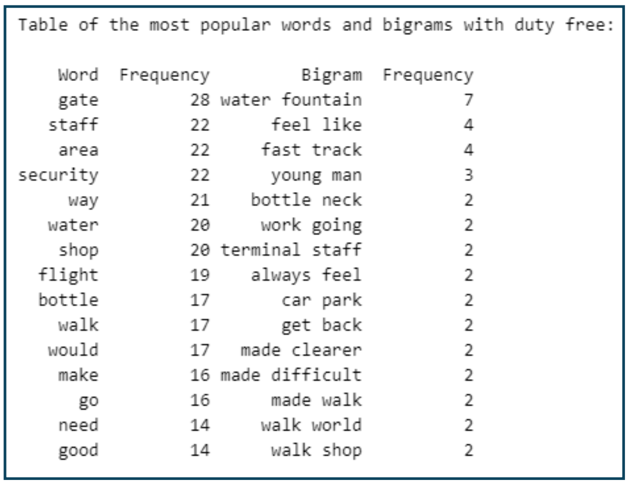
“I paid for Fast Track passport control. Chaos in arrivals, your staff member shouting orders and forcing people into main queue with insufficient signage for fast track lane. Dreadful experience”

“I think it is unacceptable for staff to shout at individuals telling them that they cannot stand once they are though in the passport control automatic area. Especially when you are waiting for children”

“Proper fast track for disabled rather than still having to queue at passport control/security and hope that someone will notice you and slot you in.”

**“duty free”**





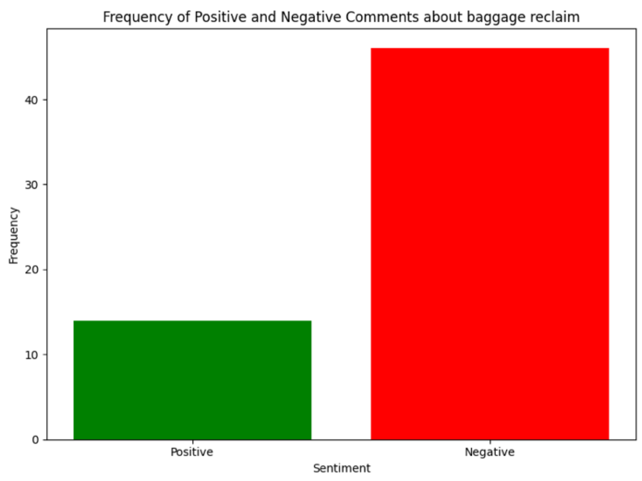
```
Table of the most popular words and bigrams with duty free:

    Word  Frequency           Bigram  Frequency
    gate         28   water fountain          7
   staff         22        feel like          4
    area         22       fast track          4
security         22        young man          3
     way         21      bottle neck          2
   water         20       work going          2
    shop         20   terminal staff          2
  flight         19      always feel          2
  bottle         17         car park          2
    walk         17         get back          2
   would         17      made clearer         2
    make         16   made difficult          2
      go         16        made walk          2
    need         14       walk world          2
    good         14        walk shop          2
```

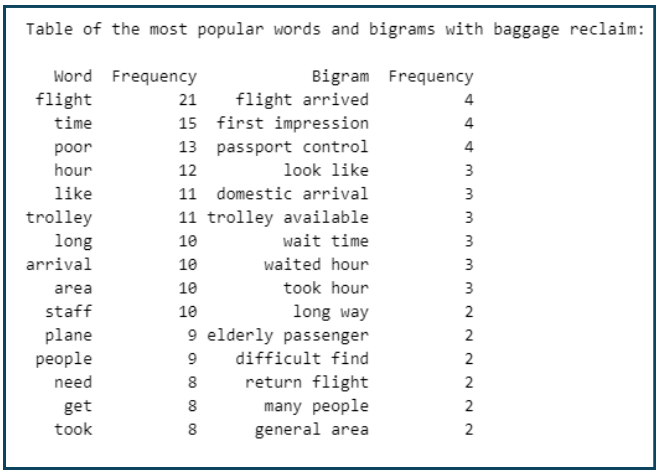
The overall comments are quite positive, indicating that people are generally satisfied with the service. However, there are a few points of concern. Many individuals express frustration over having to pass through duty-free shops to reach the gates, which some find unacceptable. Another related issue is the scarcity of water fountains, this concern was mentioned earlier.

*"Yes I object to having to walk through a duty free shop to get to the departure gate. Why?"*

*"Duty Free is a bottleneck - allow passengers to go straight to gates"*

**"baggage reclaim"**





```
Table of the most popular words and bigrams with baggage reclaim:

    Word  Frequency            Bigram  Frequency
  flight         21    flight arrived          4
    time         15  first impression          4
    poor         13  passport control          4
    hour         12         look like          3
    like         11  domestic arrival          3
 trolley         11 trolley available          3
    long         10         wait time          3
 arrival         10       waited hour          3
    area         10         took hour          3
   staff         10          long way          2
   plane          9 elderly passenger          2
  people          9    difficult find          2
    need          8     return flight          2
     get          8       many people          2
    took          8      general area          2
```

Feedback on baggage reclaim predominantly expresses negative sentiments. Words like "time," "poor," "hour," and "long" highlight the lengthy waits for baggage. (We saw the similar views in the topic 2 and 6) Additionally, many comments mention "trolley," revealing complaints about a shortage of trolleys.

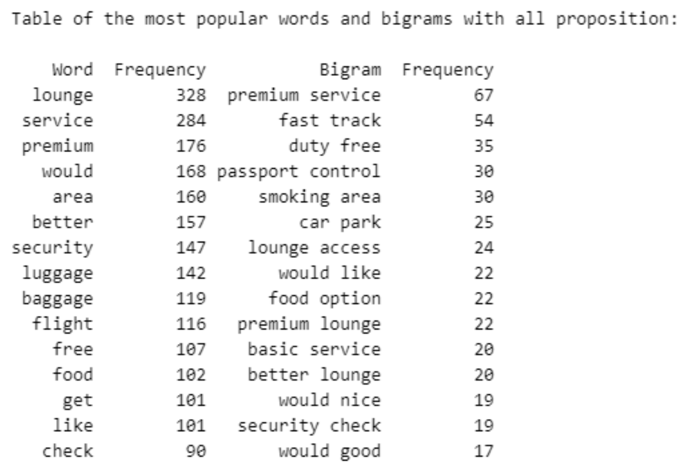
*"No trolleys at baggage reclaim and too long await on the luggage"*

*"Twice in the past 3 months on arriving back to Edinburgh there were no luggage trolleys available for customers exiting baggage reclaim"*

To improve the situation, it would be beneficial to implement a more efficient baggage handling system to minimize wait times. Increasing the availability of trolleys, particularly during high-traffic periods, would address the shortage issue. Providing real-time updates on baggage status and estimated wait times through digital screens could help manage passenger expectations and reduce frustration.

## Analysis of column with propositions about priority services

We now shift our focus to analyzing the second text column, which inquires: *"Are there any other premium services that you would like to see introduced at Edinburgh?"*



```
Table of the most popular words and bigrams with all proposition:

     Word  Frequency           Bigram  Frequency
   lounge        328  premium service         67
  service        284       fast track         54
  premium        176        duty free         35
    would        168 passport control         30
     area        160     smoking area         30
   better        157         car park         25
 security        147    lounge access         24
  luggage        142       would like         22
  baggage        119      food option         22
   flight        116   premium lounge         22
     free        107    basic service         20
     food        102    better lounge         20
      get        101       would nice         19
     like        101   security check         19
    check         90       would good         17
```

**“lounge”**

A closer examination of word frequency reveals that "lounge" is a prevalent topic. Many suggestions revolve around enhancing existing lounges, increasing their availability, or creating exclusive lounges for specific passenger groups.

*“More access to lounges” ; “I think maybe private Lounge that has shower” ; “A vip lounge membership”*

**“security”, “passport control”, “luggage”**

Furthermore, there are numerous mentions of "security," "passport control," and "luggage." Suggestions include the introduction of private security checks, dedicated passport control services, and potentially concierge services to streamline these processes.

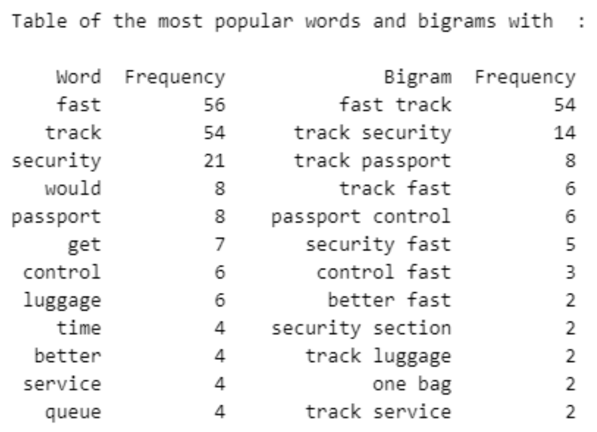
**“duty free”**

The duty-free experience is also frequently discussed, with a notable suggestion being the implementation of a preorder option. This would enable passengers to place orders in advance and conveniently collect their items at the airport.

*“Pre order servers, such as food and duty free” ; “Buy from duty free and collect when return”*

**“fast track”**

Lastly, the concept of "fast track" is often highlighted. Passengers propose the establishment of dedicated fast track lanes for security, as well as for arrivals, encompassing passport control and baggage claim.



```
Table of the most popular words and bigrams with  :

    Word  Frequency            Bigram  Frequency
    fast         56        fast track         54
   track         54    track security         14
security         21    track passport          8
   would          8        track fast          6
passport          8  passport control          6
     get          7     security fast          5
 control          6      control fast          3
 luggage          6       better fast          2
    time          4  security section          2
  better          4     track luggage          2
 service          4           one bag          2
   queue          4     track service          2
```

To conclude this analysis, we have identified several key areas that require attention to enhance the overall passenger experience. There are notable concerns related to staff interactions, facilities, and services provided at the airport. Passengers have highlighted issues with staff behavior, long wait times, and insufficient amenities. Additionally, the need for better signage, more efficient processes, and enhanced accessibility features are recurring themes. Addressing these areas of improvement will not only alleviate passenger frustrations but also contribute to a more efficient and pleasant airport experience. By focusing on these key points, airport management can significantly elevate the standard of service and satisfaction among travelers.