

AIRLINE RECOMMENDER SYSTEM

Himanshu Joshi (hsjoshi)
Atharv Jangam (ajangam)
Trishna Patil (tripatil)

Luddy School of Informatics, Computing, and Engineering
Indiana University Bloomington

CSCI-B 565 DATA MINING - Prof. Yuzhen Ye

Table of Contents

1. <u>Abstract and Introduction</u>	3
1.1 Abstract	3
1.2 Introduction	3
1.3 Keywords	4
2. <u>Data Description</u>	4
3. <u>Methodologies Used</u>	5
3.1 Data Cleaning	6
3.1.1 Outliers	7
3.1.2 Merging Similar Columns	7
3.2 Data Visualization	7
3.2.1 Correlation between Columns	7
3.2.2 Multivariate Relations	9
3.3 Logistic Regression	17
3.4 K-Nearest Neighbors	17
3.5 Association Rules Mining (Apriori Algorithm)	18
3.6 Sentiment Analysis with Random Forest Classifier	20
4. <u>Results</u>	24
5. <u>References</u>	25
6. <u>Citations</u>	26

1. Abstract, Introduction, and Keywords

1.1 Abstract

The airline plays an important role in transportation; it can be used for combat, transporting huge cargo, people, and more in the most efficient and safest manner. Since you only have a 1 in 9,821 chance of dying from an air transport incident, flying is actually one of the safest forms of transportation. With these safeties in mind, it also becomes important to determine which airline has the best overall performance and rating. There are numerous airline service providers; passengers often get confused about which airline to choose, as some have good ratings while others don't. It becomes crucial to have an airline recommender system where travelers can decide on an airline along with the required date and time of travel.

Also, from another point of view, the airlines should upgrade themselves to attract more passengers. In this world full of reviews, it becomes important to have a good review for the prestige and trust of passengers. Otherwise, many competitors are out there, which can hamper your business.

1.2 Introduction

The airline recommender system is a recommender system where the system recommends whether to fly with the airline service according to the previous feedback of travelers and their experiences. As far as we know, no major machine learning models are in place to recommend supplementary services to airline firms.

Open Source data is available on the Systrax website, which is hosted on the airline quality website. This project aims to use that data and create a recommender system by various data mining techniques, which will be helpful for other travelers to choose their flight and what kind of service they can expect. The project also focuses on the airline perspective, where they can introspect their services and make a better experience for the passengers by visualizing the various ratings and recommendations by the passengers.

The data also contains reviews of passengers, and this project supports the sentimental analysis of that by various data mining techniques.

1.3 Keywords

Features, data frame, Linear Regression, Decision Tree, Logistic Regression, K Nearest Neighbour (KNN), Sentimental analysis, Apriori Algorithm, frequent itemset, Association Rules, Random Forest, Data Mining

2. Data Description

The dataset we used for this project was scraped from the Skytrax website, which is available freely on Kaggle. [\(Click here to access the dataset\)](#). Some part of the data was scraped by us, but the data on Kaggle was in good shape and oriented, so we moved forward with it to achieve new goals and implement the data mining techniques. We maintain that recommender systems hold the key to customer centricity because they can recognize and cater to the customer's demands at all points of contact throughout the traveler's journey.

```

Int64Index: 27284 entries, 0 to 41217
Data columns (total 14 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   airline_name                          27284 non-null  object
1   author                                27284 non-null  object
2   author_country                        27284 non-null  object
3   content                               27284 non-null  object
4   cabin_flown                           27284 non-null  object
5   overall_rating                        27284 non-null  float64
6   seat_comfort_rating                  27284 non-null  float64
7   cabin_staff_rating                   27284 non-null  float64
8   food_beverages_rating                27284 non-null  float64
9   inflight_entertainment_rating         27284 non-null  float64
10  value_money_rating                   27284 non-null  float64
11  Month                                27284 non-null  int64
12  Year                                 27284 non-null  int64
13  recommended                          27284 non-null  int64
dtypes: float64(6), int64(3), object(5)
memory usage: 3.1+ MB

```

Figure 1: Data Description

The data is in 'CSV' format consisting of 27284 rows and 14 rows. Six columns are floats, 3 are integers, and 5 are strings.

It has data from 292 airlines parametrized by airline name, seat comfort, cabin staff, food beverages, inflight entertainment, value money, and overall ratings. The maximum for each rating is 10, and the minimum is 0.

It also consists of 27284 individuals parameterized by the author(Name of the passenger), author country, content(reviews), cabin flown, year, month, and recommendation ('1' for Yes and '0' for No).

The data is available from the year 2011 to 2015.

For training and testing the models, we used 80 % of the data for training and 20% of the data for testing, and for sentiment analysis, 25% for testing and 75% for the train.

3. Methodologies Used

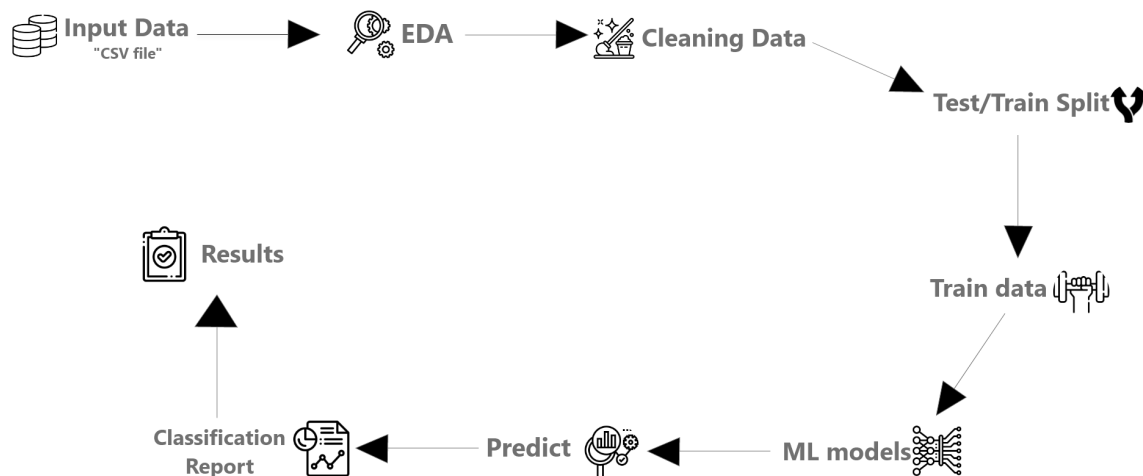
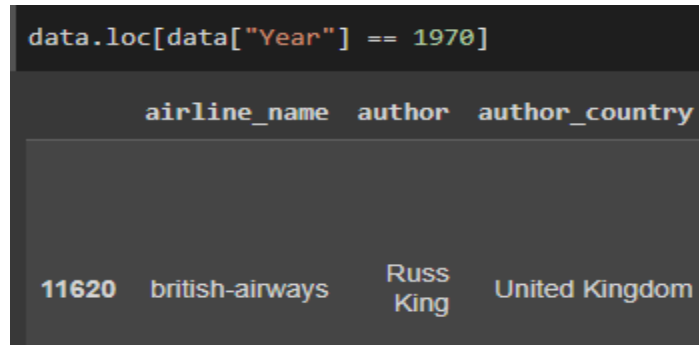


Figure 2: Methodology

3.1 Data Cleaning

3.1.1 Outliers

In the dataset, we identified a single row that was an outlier, as it had the year 1970. And during that time, the airline British Airline was not even founded, so we decided to remove the entire row.



```
data.loc[data["Year"] == 1970]
```

	airline_name	author	author_country
11620	british-airways	Russ King	United Kingdom

Figure 3: British Airline data in the year 1970

3.1.2 Merging Similar Columns

We identified year and month, which can be merged to form a new column, 'date,' and also changed the data type to DateTime.

3.1.3 Deleting the columns for sentiment analysis.

As for the sentiment analysis, we have dropped some columns we won't use.

3.2 Data Visualization

3.2.1 Correlation between Columns

Data Visualization is an important part of Data Mining. It helps us get visual insights about the dataset, such as some striking features or patterns, which can help us choose the appropriate machine learning algorithms to apply.

We plotted a few basic plots showing the recommendations by the authors, overall rating, and distribution of classes.

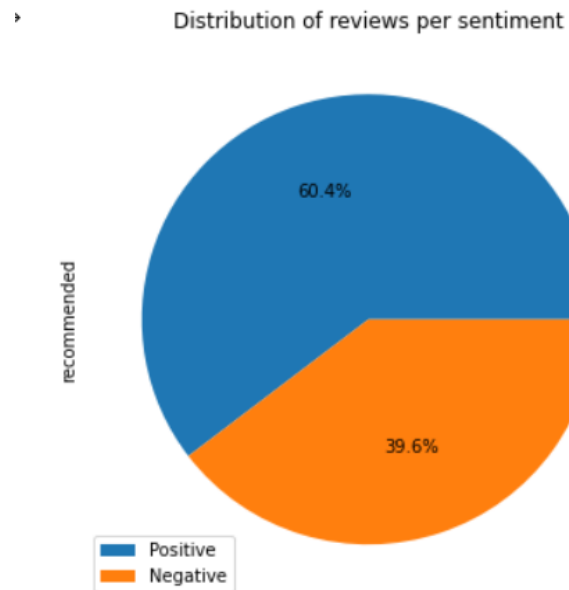


Figure 4: Recommended Pie Chart

Distribution of recommendations in the whole dataset. There are 16750 people who have recommended (1) and 10808 who have not recommended (0).

We can interpret that the no of passengers who have recommended the airline is 60.4%, while 39.6% have not recommended the airline.

Correlation between Columns

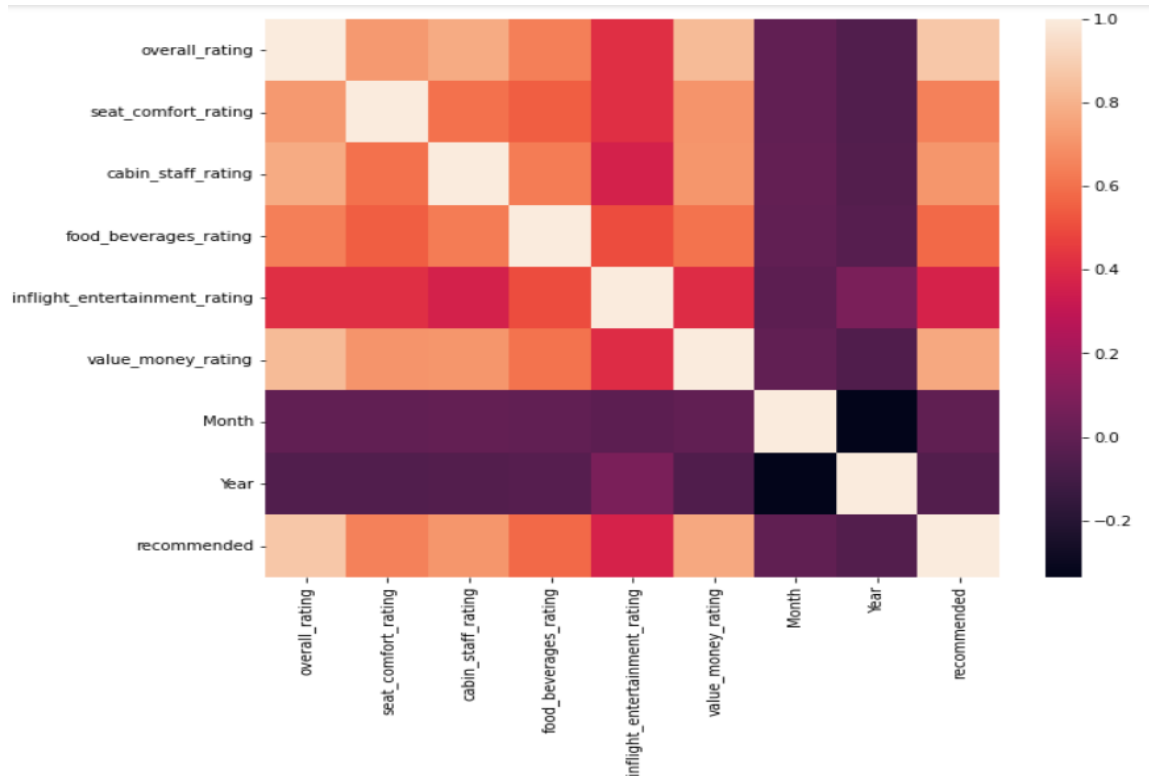


Figure: 5: HeatMap of the columns in a data frame

From the above figure, the recommended column is highly related to overall ratings while the inflight entertainment rating is least related to other columns.

Hence people are least concerned about inflight entertainment, and maybe the airline companies should focus on other parameters to increase revenue.

3.2.2 Multivariate relations

A statistical method for analyzing data, including many measurements or observations, is known as multivariate analysis (MVA).

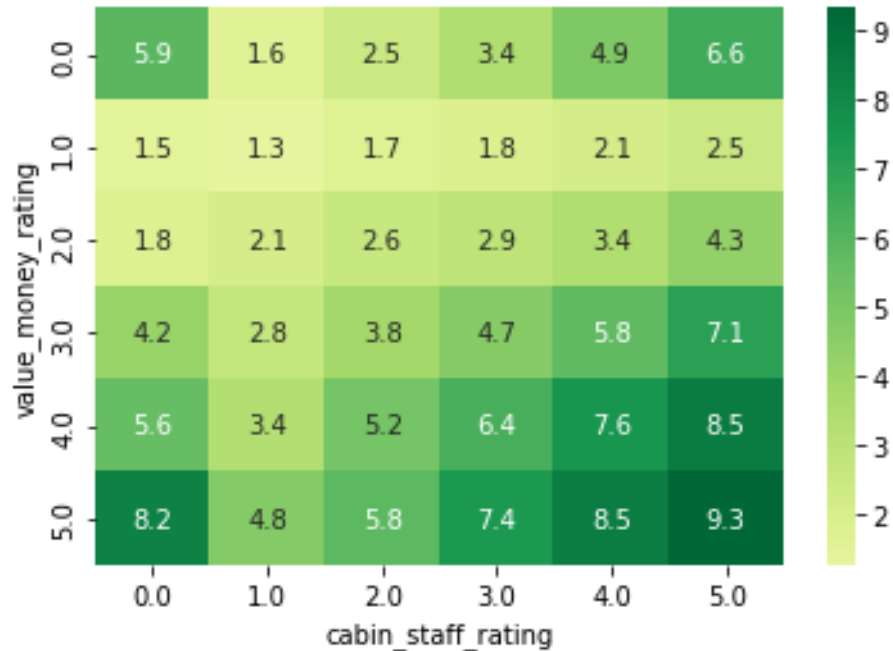


Figure 6: Multivariate Analysis of 3 Columns

In the above plot, we created a multivariate analysis of 3 columns, with `index='value_money_rating'`, `columns='cabin_staff_rating'`, `values='overall_rating'`

Model Analysis (On a few data)

Once the important columns were identified, we performed cross-validation scores before training the model. It is the method that determines whether the model can generalize over the whole dataset. We had set the recommended column as a target to identify the behaviors of other columns.

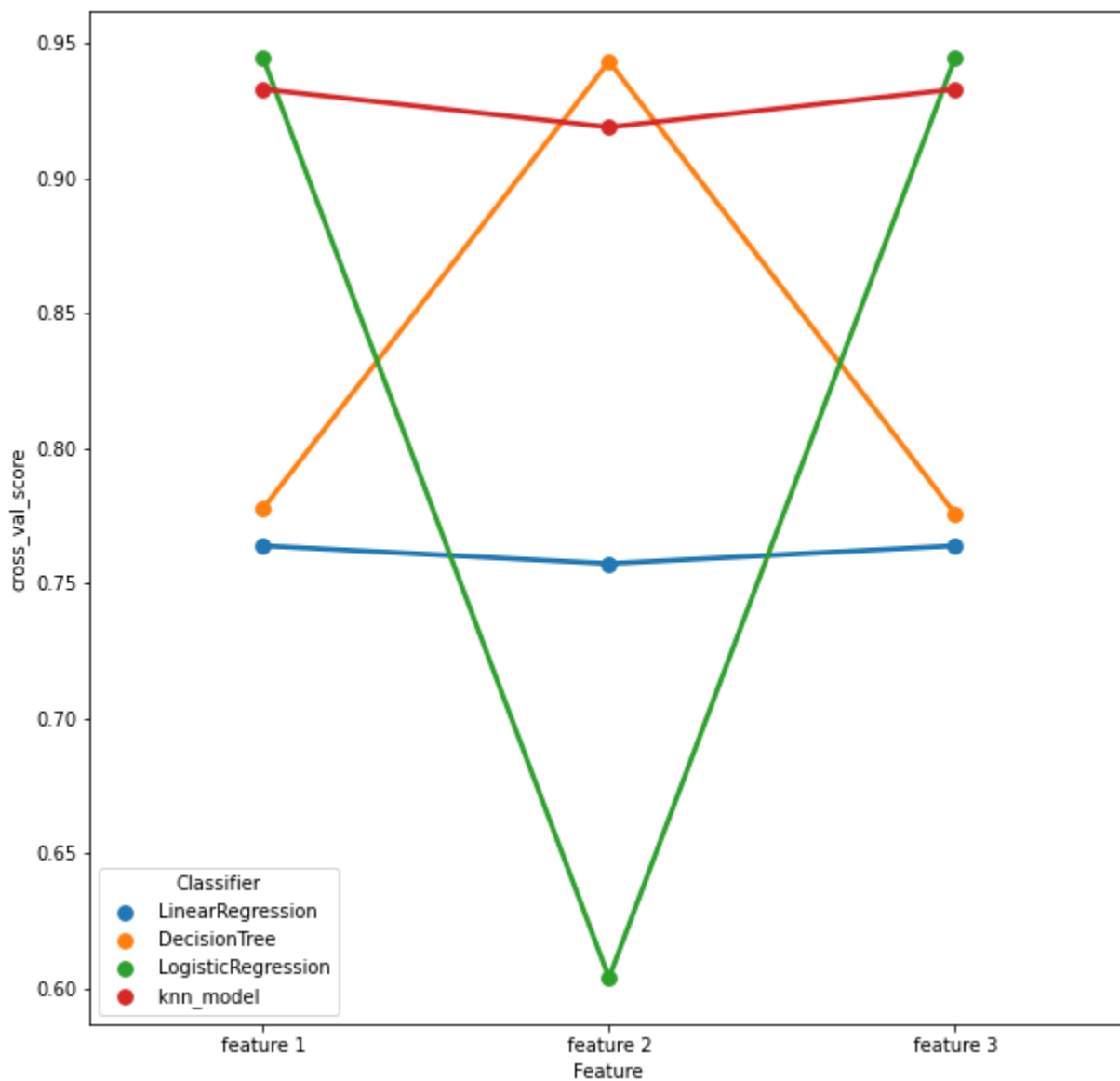


Figure 7: Cross-validation scores and features for different models

We identified 3 features:

Feature 1: It will be 10 columns of data without 'recommended' and 'author' columns from the data frame. The textual columns will be Encoded and standardized using StandardScaler.

Feature 2: It will be just one column, 'overall_rating,' normalized using the L2 norm.

Feature 3: It will be a combination of the above two features, which consist of 11 columns.

In these 3 features, we tested our data and got the plot above. Interpretations from the above plot are

1. For any model, feature 1 and feature 3 were almost similar. This is because feature 1 had 10 columns, and feature 3 had 11 columns. One additional column was the overall rating, and hence we could interpret that the overall recommendation is not only based on the overall rating but also on other individual columns.
2. Earlier, we had believed that overall rating is responsible for a recommendation, but that claim holds false.
3. We believe that Logistic Regression is a good choice for our model, and it should give us better accuracy when we train the data.
4. The Decision Tree classifier worked well on feature 2, because feature 2 had just one column with values ranging from 0.0 to 10.0, so it worked well for that.
5. But on the other hand, the decision tree classifier performed poorly when the column size was more. Hence it is difficult to find any patterns for the dataset, and that's why it performed badly.
6. The Linear Regression is a poor choice as the model performed badly for all 3 features.
7. The K Nearest Neighbour model performed average on all 3 features, so it can be another choice after Logistic Regression.
8. When there was just one feature, all models behaved hyperactively except for KNN, which handled the situation quite well.
9. With these cross-Val scores, we could use Logistic Regression for feature 1 and feature 3.
10. Similarly, we can also use the decision tree for feature 2.
11. But the best model would be KNN as it has a slightly lower accuracy for feature 1 and 3 and better for feature 2.

LogisticRegression					
Model Score: 0.9455744914788345					
Classification Report:					
	precision	recall	f1-score	support	
0	0.93	0.94	0.93	2140	
1	0.96	0.95	0.96	3317	
accuracy			0.95	5457	
macro avg	0.94	0.94	0.94	5457	
weighted avg	0.95	0.95	0.95	5457	

knn_model					
Model Score: 0.9351291918636614					
Classification Report:					
	precision	recall	f1-score	support	
0	0.92	0.91	0.92	2140	
1	0.94	0.95	0.95	3317	
accuracy			0.94	5457	
macro avg	0.93	0.93	0.93	5457	
weighted avg	0.94	0.94	0.94	5457	

Figure 8: Model Summary for feature 1

We tested the model by training and fitting the data. The test size was 20% and the training was 80%. The target was the same as defined above (recommended) and feature 1 was also the same as defined above.

We got an accuracy of 94% for Logistic Regression and 93% for KNN. If we refer to figure 7, we can see that logistic regression had slightly higher accuracy than KNN for feature 1. Hence the point stands true.

We also tested other models like Linear Regression and Decision Tree Classifier, and the scores were above 90 but less than logistic and KNN, which was a bit surprising.

The figure also denotes the f-1 scores of both models, which are above 92% and higher than both models. The F1 Score is used to measure the performance of binary classification.

The recall tells the model performance in terms of measuring the count of true positives over true positives and false negatives combined. It was also better for the above two models than the rest of the models.

The precision score, which tells us the true positive over true positive and false positive combined is also good for both models.

```

LogisticRegression
Model Score: 0.6078431372549019
Classification Report:
              precision    recall  f1-score   support

         0         0.00         0.00         0.00         2140
         1         0.61         1.00         0.76         3317

 accuracy          0.61         0.61         0.61         5457
 macro avg         0.30         0.50         0.38         5457
 weighted avg         0.37         0.61         0.46         5457

-----

knn_model
/usr/local/lib/python3.8/dist-packages/sklearn/metrics/_classification.py:136: UserWarning:
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.8/dist-packages/sklearn/metrics/_classification.py:136: UserWarning:
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.8/dist-packages/sklearn/metrics/_classification.py:136: UserWarning:
  _warn_prf(average, modifier, msg_start, len(result))
Model Score: 0.9422759758108851
Classification Report:
              precision    recall  f1-score   support

         0         0.91         0.94         0.93         2140
         1         0.96         0.94         0.95         3317

 accuracy          0.94         0.94         0.94         5457
 macro avg         0.94         0.94         0.94         5457
 weighted avg         0.94         0.94         0.94         5457

```

Figure 9: Model Summary for feature 2

As we can see in the above figure, Logistic Regression performed badly for feature 2. As we saw in Figure 7, this was as expected because we just have one feature that is 'overall_rating'.

While we have a good model score for KNN which is around 94%. KNN also performed well for feature 1 and also for feature 2. Hence KNN has a good average score.

We have a Decision tree, too, which has an accuracy of around 94%, which is the same as KNN. The model summary suggested KNN and Decision Tree best for feature two since it is a simple classification.

LogisticRegression				
Model Score: 0.9455744914788345				
Classification Report:				
	precision	recall	f1-score	support
0	0.93	0.94	0.93	2140
1	0.96	0.95	0.96	3317
accuracy			0.95	5457
macro avg	0.94	0.94	0.94	5457
weighted avg	0.95	0.95	0.95	5457

knn_model				
Model Score: 0.9351291918636614				
Classification Report:				
	precision	recall	f1-score	support
0	0.92	0.91	0.92	2140
1	0.94	0.95	0.95	3317
accuracy			0.94	5457
macro avg	0.93	0.93	0.93	5457
weighted avg	0.94	0.94	0.94	5457

Figure 10: Model Summary for feature 3

The results above are similar to feature 1 results. The score for Logistics is slightly higher than KNN, as we saw in Figure 7.

For feature 3, we combined feature 1 and feature 2 together and so the results are almost similar.

Also, we could infer that overall_rating is independent of recommendation (which is our target), which holds true.

3.3 Logistic Regression

We performed Logistic Regression on our dataset. The Columns we identified were airline_name, author_country, cabin_flow, overall_rating, seat_comfort_rating, cabin_staff_rating, food_beverages_rating, inflight_entertainment_rating, value_money_rating, date. For column date, author_country, cabin_flow, and airline_name were encoded to Label Encoder and then used standard scalar.

	precision	recall	f1-score	support
0	0.93	0.90	0.92	2140
1	0.94	0.96	0.95	3317
accuracy			0.94	5457
macro avg	0.93	0.93	0.93	5457
weighted avg	0.94	0.94	0.94	5457

Figure 11: Logistic Regression classification report

The logistic regression performed well on the new data which was as expected.

3.4 K-Nearest Neighbors

We have also used the K-Nearest Neighbors algorithm on this model which is a non-parametric supervised learning method that uses proximity to make classifications or predictions about the grouping of an individual data point. We have taken a number of neighbors as 5, and after training our model, we are getting a very good accuracy score of 0.912. The figure below shows the KNN classification report -

	precision	recall	f1-score	support
0	0.85	0.92	0.88	1986
1	0.95	0.91	0.93	3471
accuracy			0.91	5457
macro avg	0.90	0.91	0.91	5457
weighted avg	0.91	0.91	0.91	5457

Figure 12: K-Nearest Neighbor classification report

3.5 Association Rules Mining (Apriori Algorithm)

We tried to implement Apriori Algorithm on our dataset. We tried to search if there was any specific pattern in the dataset for the ratings or recommendations or the airline names, and we could infer a few association rules from it.

For apriori, we considered three columns: `airline_name`, `recommender`, and `overall_rating`.

We converted the recommended column to a string of Yeses and Nos. (1 for Yes and 0 for No) because it will be easier for us to differentiate between the overall ratings (which are integers too) from the recommended ones.

For Apriori, the settings we used were `min_support=0.0005`, `min_confidence=0.0005`, `min_length=4`, and `min_lift=2`.

We tried to find any associate rules between the `airline_name` and recommendation or both with the overall rating or all 3 towards some column(s). Some of the rules were hard to interpret, but we found them at our level best.

There were around 50 associated rules which were found on the data frame of 3 columns. The figure below denotes the first 11 and last nine rules.

	rule	Support	Confidence	Lift
0	(1.0, No)	0.142653	0.996416	2.515519
1	(1.0, air-canada)	0.005058	0.035330	2.008157
2	(air-canada-rouge, 1.0)	0.015064	0.105223	4.083630
3	(1.0, american-airlines)	0.009273	0.064772	3.052122
4	(frontier-airlines, 1.0)	0.002749	0.019201	2.632498
5	(1.0, royal-air-maroc)	0.000880	0.006144	2.748155
6	(spirit-airlines, 1.0)	0.003482	0.024322	5.065382
7	(sunwing-airlines, 1.0)	0.005938	0.041475	3.017475
8	(1.0, united-airlines)	0.013965	0.097542	3.314128
9	(aeromexico, 10.0)	0.000880	0.005390	2.042294
10	(air-astana, 10.0)	0.001136	0.006962	2.922051
41	(1.0, allegiant-air, No)	0.002126	0.014849	3.215269
42	(1.0, american-airlines, No)	0.009273	0.064772	4.440147
43	(austrian-airlines, 1.0, No)	0.000733	0.001851	2.524567
44	(british-airways, 1.0, No)	0.003739	0.009438	2.524567
45	(cathay-pacific-airways, 1.0, No)	0.001503	0.003794	2.524567
46	(1.0, china-eastern-airlines, No)	0.000953	0.002406	2.524567
47	(1.0, china-southern-airlines, No)	0.001100	0.007680	2.205756
48	(delta-air-lines, 1.0, No)	0.001686	0.011777	2.362538
49	(1.0, emirates, No)	0.002786	0.019457	2.033916

There were around 50 associated rules which were found on the data frame of 3 columns. The figure above denotes the first 11 and last 9 rules.

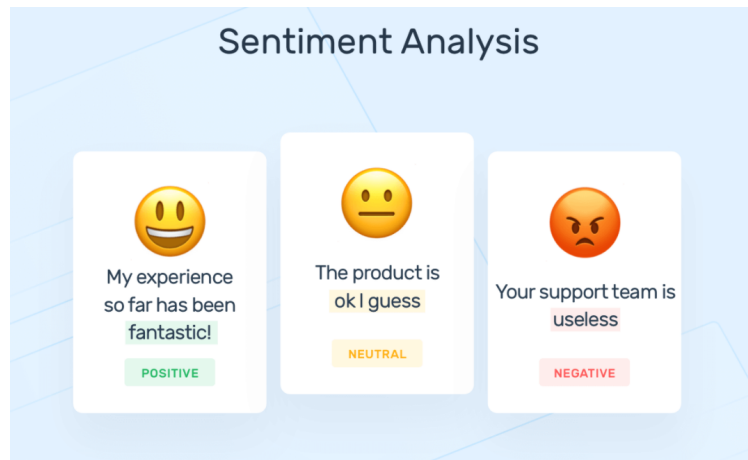
Points that can be inferred from the rules are

- The rule (1.0, No) had maximum support and confidence. Note that 1.0 is the overall rating, and 'No' is the recommendation. Hence we could infer

that the customers who had overall rated 1.0 had also not recommended the airline. Hence the claim that we were assuming to be true above, about less rating, and no recommendation, is believed to be true.

- The air-Canada airline received a very bad overall ratification which is 1.0. 0.005058 supports this rule.
- Eventually, airlines like air-Canada-rouge, American-airline, frontier-airline, royal-air-Maroc, spirit-airline, Sunwing, and united-airline have a rating of 1.0 supported by maximum transactions.
- Airlines like Aeromexico and air-Astana have a good overall rating supported by a higher confidence value.
- The maximum rating for airline name, which was supported by most other rules for the airline Aeroflot-Russian-airlines which had a maximum overall rating of 7.0, BMI-British-midland-international had an overall rating of 9.0 supported by most transactions, followed by garuda-Indonesia, Korean-air, and royal-Brunei-airlines.
- We can also infer from the Apriori that the overall rating of 4.0 and below had people not recommending the airline. Hence we can say that if the overall rating is 5.0 and above are most likely to be recommended, which we also saw in Visualizing the overall ratings and recommendations.
- There are also many transactions where the airline is not recommended by many people. Which are air-Canada-rouge, spirit-airlines, and AirAsia.
- We can also say the most unrecommended airline is aerolineas-argentinas which has a rule of (aerolineas-argentinas, 1.0, No) the most.
- There was a maximum lift of 5 for the transaction (spirit-airlines, 1.0)

3.6 Sentiment Analysis with Random Forest Classifier



	airline_name	author	content	recommended	char_count	word_count	avg_word_length
15444	emirates	A Joven	I flew SYD-BKK in Sep 2014. This was my first ...	1	651	129	5.046512
734	air-arabia	D Graubard	A320 flight Sharjah-Yanbu (Saudi Arabia) on 16...	1	477	103	4.631068
28394	qatar-airways	R Johnston	I had the pleasure of flying with Qatar from N...	1	415	96	4.322917

Random Forest on Sentiment Analysis

Random forest is a type of machine learning algorithm that is often used for sentiment analysis because it is a powerful and versatile method that can handle a large amount of data. It works by creating multiple decision trees, each of which is trained on a random subset of the data, and then combines the results of all of the trees to make a final prediction. This allows the algorithm to capture a wider range of patterns in the

data and can help improve the accuracy of the sentiment analysis. Additionally, random forest algorithms are relatively simple to implement and can be easily adjusted to work with different data types.

We used different models like SVM(Support vector machine) and Adaboost but found that random forest works the best on our dataset.

We have used different `n_estimators` and `max_depth` to find the best model.

```
MAX DEPTH: 20 / N_EST: 100 -- Accuracy: 0.84 / Precision: 0.826 / Recall: 0.925
MAX DEPTH: None / N_EST: 100 -- Accuracy: 0.851 / Precision: 0.844 / Recall: 0.919
MAX DEPTH: None / N_EST: 5 -- Accuracy: 0.799 / Precision: 0.81 / Recall: 0.862
```

From this, we can see that the second model works best with `Max_depth` as `None` and `N_estimators = 100`.

Results of Random Forest:

	precision	recall	f1-score	support
0	0.86	0.76	0.81	2780
1	0.85	0.92	0.88	4041
accuracy			0.85	6821
macro avg	0.85	0.84	0.84	6821
weighted avg	0.85	0.85	0.85	6821

Fig - Classification report for Random Forest

The classification report above gives us an accuracy of 85% on the sentiment analysis of the content or reviews. Hence, we can say that we have predicted 85% of the right sentiments of the customers from their feedback.

Confusion Matrix for the model:

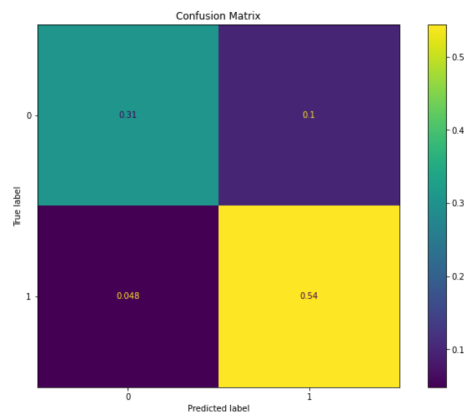
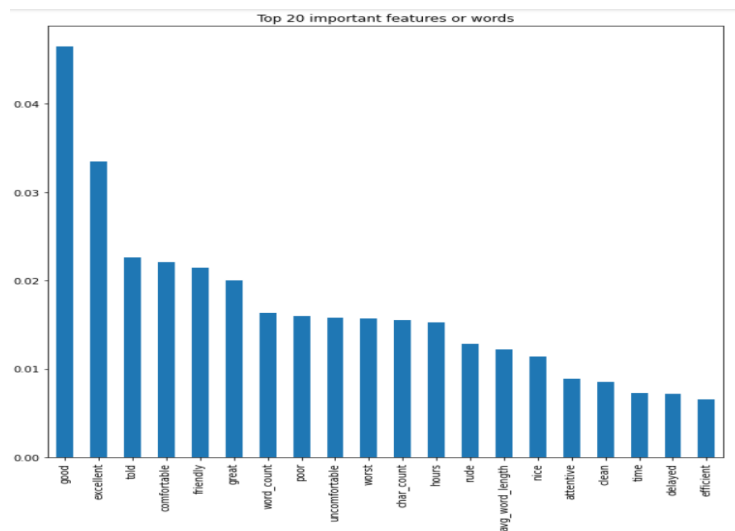


Fig: Confusion matrix for Random Forest Classifier

Plot to show the top 20 important words for the analysis:



The figure above shows us the importance of the words and how they might affect the model's decision-making. We can see here that good has the highest importance. Which was way different than what the word cloud showed us. Hence, we can say that the analysis we did is good.

4. Results

We have created a highly trained Airline Recommendation model to predict whether any customer would recommend a particular airline based on various different airline attributes.

This would be helpful for other customers while booking an airline of their choice based on other customer ratings, as well as help airlines improve based on customer feedback.

Based on the observations from the above-mentioned models, we concluded that:

- The best airlines recommended by most people and having a good overall rating are Aeroflot-Russian-airlines, BMI-British-midland-international, garuda-Indonesia, Korean-air, and royal-Brunei-airlines.
- The airlines having least ratings are air-Canada-rouge, American-airline, frontier-airline, royal-air-Maroc, spirit-airline, Sunwing, and united-airline.
- We saw that inflight entertainment had contributed very less to overall ratings and recommendations, so airlines should focus more on other features like seat comfort, cabin staff, and food beverages.
- Passengers with an overall rating of 4 and below did not recommend the airline.
- Based on the sentiment analysis, we can conclude that most people were happy with their airlines and would recommend them.
- If the customer has traveled in the economy and business class, there is a high chance the customer would recommend the airline. Instead, if customers have traveled in first class and premium economy, they won't recommend it.

5. References

1. <https://thinkingneuron.com/how-to-visualize-the-relationship-between-two-categorical-variables-in-python/>
2. <https://towardsdatascience.com/exploratory-data-analysis-eda-python-87178e35b14>
3. <https://www.shanelynn.ie/bar-plots-in-python-using-pandas-dataframes/>
4. <https://www.dataquest.io/blog/how-to-plot-a-bar-graph-matplotlib/>
5. <https://chart-studio.plotly.com/~i/26/yes-or-no/#/>
6. <https://www.geeksforgeeks.org/linear-regression-python-implementation/>
7. <https://realpython.com/linear-regression-in-python/>
8. <https://scikit-learn.org/stable/modules/tree.html>
9. <https://www.datacamp.com/tutorial/decision-tree-classification-python>
10. <https://intellipaat.com/blog/data-science-apriori-algorithm/>
11. <https://www.geeksforgeeks.org/implementing-apriori-algorithm-in-python/>
12. <https://analyticsindiamag.com/beginners-guide-to-understanding-apriori-algorithm-with-implementation-in-python/>
13. <https://libraries.io/pypi/apyori>
14. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
15. <https://www.sciencedirect.com/science/article/pii/S1877050918301625>

6. Citations

1. An Approach to Recommendation Systems Using Scalable Association Mining Algorithms on Big Data Processing Platforms: A Case Study in Airline Industry - <https://ieeexplore.ieee.org/document/9548413>
2. How recommender systems can transform airline construction and retailing - https://www.researchgate.net/publication/350217331_How_recommender_systems_can_transform_airline_offer_construction_and_retailing
3. A Comparative study on Airline Recommendation System Using Sentimental Analysis on Customer Tweets - https://www.researchgate.net/publication/323552728_A_Comparative_study_on_Airline_Recommendation_System_Using_Sentimental_Analysis_on_Customer_Tweets