Capstone Project – Classification

Project -Airline Passenger Referral Prediction

Team Members

Sanchit Misra

Mohit Jain ,

Tushar Hande,

Problem Statement: Data includes airline reviews from 2006 to 2019 for popular airlines around the world with multiple choice and free text questions. Data is scraped in Spring 2019. The main objective is to predict whether passengers will refer the airline to their friends.

Attribute Information:

1. Airline: Name Of The Airline.
2. Overall: Overall Point Is Given To The Trip Between 1 To 10.
3. Author: Author Of The Trip
4. Review date: Date Of The Review
5. Customer Review: Review Of The Customers In Free Text Format
6. Aircraft: Type Of The Aircraft
7. Traveler type: Type Of Traveler (E.G. Business, Leisure)
8. Cabin: Cabin At The Flight
9. Route: Route of the flight.
10. Date Flown: Flight Date
11. Seat comfort: Rated Between 1-5
12. Cabin Service: Rated Between 1-5
13. Food : Rated Between 1-5
14. Entertainment: Rated Between 1-5
15. Ground service: Rated Between 1-5
16. Value for money: Rated Between 1-5
17. Recommended: Binary, Target Variable.

Data Inspection:-

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 131895 entries, 0 to 131894
Data columns (total 17 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   airline          65947 non-null  object 
 1   overall          64017 non-null  float64
 2   author           65947 non-null  object 
 3   review_date      65947 non-null  object 
 4   customer_review  65947 non-null  object 
 5   aircraft         19718 non-null  object 
 6   traveller_type   39755 non-null  object 
 7   cabin            63303 non-null  object 
 8   route            39726 non-null  object 
 9   date_flown       39633 non-null  object 
 10  seat_comfort     60681 non-null  float64
 11  cabin_service    60715 non-null  float64
 12  food_bev         52608 non-null  float64
 13  entertainment    44193 non-null  float64
 14  ground_service   39358 non-null  float64
 15  value_for_money  63975 non-null  float64
 16  recommended      64440 non-null  object 
dtypes: float64(7), object(10)
memory usage: 17.1+ MB
```

Information of whole data set

Interpretation:-

- By examining the info() method, we can see that there are **131895** rows in total and that the maximum number of non-NaN values is only 65947, indicating that every odd row is a NaN.
- We next considered removing all odd rows from the dataset, however we realized later that the dataset's end still contained NaN rows.

Checked and Handled Null and Duplicate Values:

```
airlines_df.isnull().sum()

airline             65948
overall             67878
author              65948
review_date         65948
customer_review     65948
aircraft           112177
traveller_type      92140
cabin               68592
route               92169
date_flown          92262
seat_comfort        71214
cabin_service       71180
food_bev            79287
entertainment       87702
ground_service      92537
value_for_money     67920
recommended         67455
dtype: int64
```

```
#Null-values in percentage before removal.
missing_values_per_check(airlines_df)
```

	column_name	percent_missing
0	aircraft	85.050229
1	ground_service	70.159597
2	date_flown	69.951097
3	route	69.880587
4	traveller_type	69.858600
5	entertainment	66.493802
6	food_bev	60.113727
7	seat_comfort	53.992949
8	cabin_service	53.967171
9	cabin	52.005004
10	value_for_money	51.495508
11	overall	51.463664
12	recommended	51.142955
13	customer_review	50.000379
14	review_date	50.000379
15	author	50.000379
16	airline	50.000379

```
airlines_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 65947 entries, 0 to 65946
Data columns (total 19 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   airline          65947 non-null  object
 1   overall          64017 non-null  float64
 2   author           65947 non-null  object
 3   review_date      65947 non-null  object
 4   customer_review  65947 non-null  object
 5   aircraft         19718 non-null  object
 6   traveller_type   39755 non-null  object
 7   cabin            63303 non-null  object
 8   route            39726 non-null  object
 9   seat_comfort     60681 non-null  float64
 10  cabin_service    60715 non-null  float64
 11  food_bev         52608 non-null  float64
 12  entertainment    44193 non-null  float64
 13  ground_service   39358 non-null  float64
 14  value_for_money  63975 non-null  float64
 15  recommended      64440 non-null  object
 16  month            39633 non-null  float64
 17  year             39633 non-null  float64
 18  day              39633 non-null  float64
dtypes: float64(10), object(9)
memory usage: 9.6+ MB
```

Observation:-

- After Duplicate and Null value removal we have now **65947** valid rows in our dataset that sequenced from 0 to 65946.
- We have total 19 columns where **10** columns have **float** values and **9** have **categorical** values.

Checking unique values of Data Set:

```
The column overall has unique values :
1.0      17383
10.0      8530
9.0       7850
8.0       7209
2.0       5988
7.0       4590
3.0       4041
5.0       3187
6.0       2635
4.0       2604
Name: overall, dtype: int64
 --------------------------------------------
The column traveller type has unique values :
Solo Leisure      14798
Couple Leisure    10285
Family Leisure     7583
Business           7089
Name: traveller_type, dtype: int64
 --------------------------------------------
The column cabin has unique values :
Economy Class      48558
Business Class     10326
Premium Economy     2799
First Class         1620
Name: cabin, dtype: int64
 --------------------------------------------
```

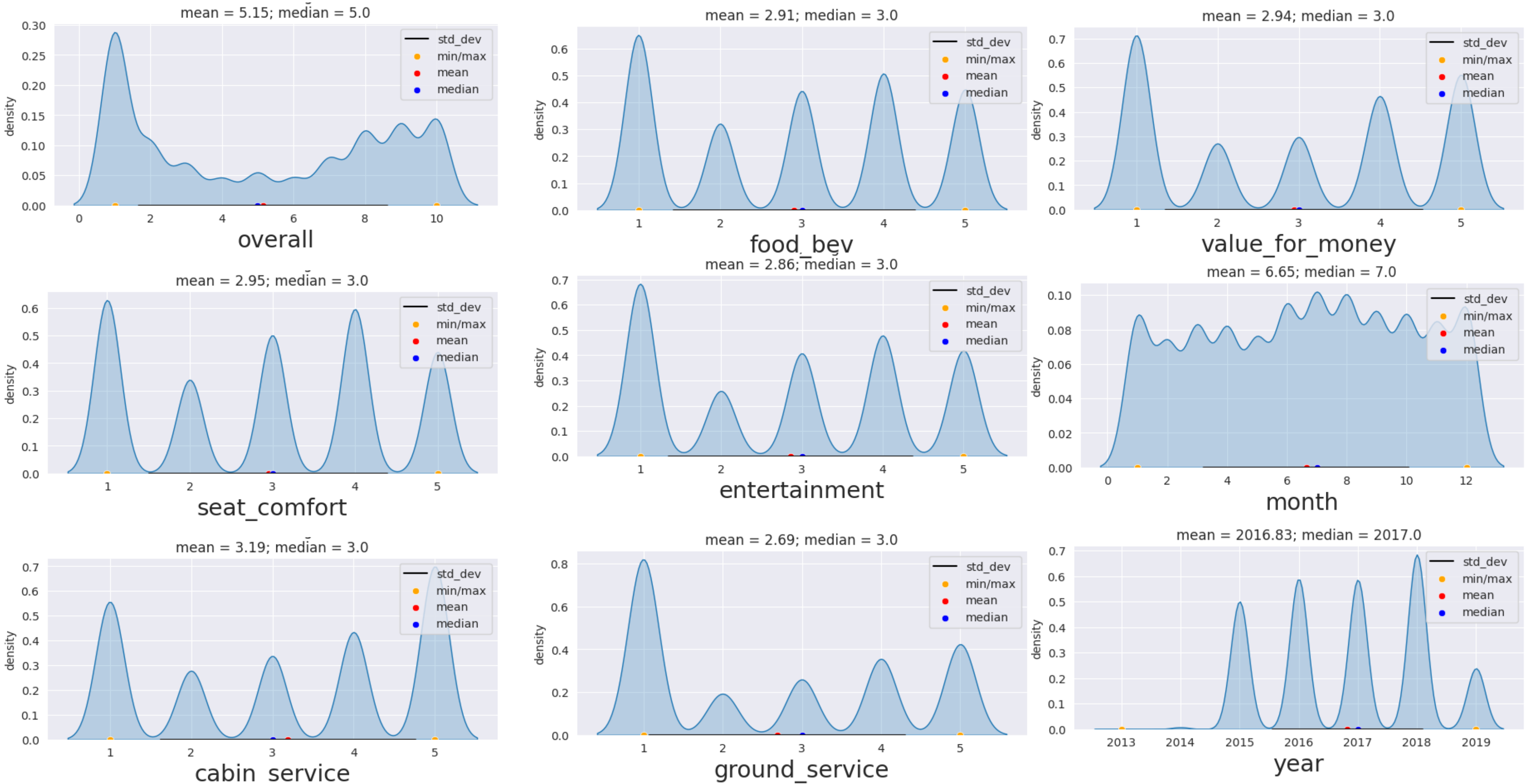
```
The column seat comfort has unique values :
1.0     15222
4.0     14433
3.0     12139
5.0     10665
2.0      8222
Name: seat_comfort, dtype: int64
 --------------------------------------------
The column cabin service has unique values :
5.0     18426
1.0     14660
4.0     11428
3.0      8887
2.0      7314
Name: cabin_service, dtype: int64
 --------------------------------------------
The column food bev has unique values :
1.0     14440
4.0     11264
5.0      9955
3.0      9824
2.0      7125
Name: food_bev, dtype: int64
 --------------------------------------------
The column entertainment has unique values :
1.0     13432
4.0      9410
5.0      8250
3.0      8017
2.0      5084
Name: entertainment, dtype: int64
```

```
The column ground service has unique values :
1.0     15740
5.0      8135
4.0      6816
3.0      4971
2.0      3696
Name: ground_service, dtype: int64
 --------------------------------------------
The column value for money has unique values :
1.0     19862
5.0     15369
4.0     12938
3.0      8269
2.0      7537
Name: value_for_money, dtype: int64
 --------------------------------------------
The column recommended has unique values :
no      33894
yes     30546
Name: recommended, dtype: int64
 --------------------------------------------
The column year has unique values :
2018.0    10406
2016.0     8964
2017.0     8909
2015.0     7614
2019.0     3626
2014.0      112
2013.0        2
Name: year, dtype: int64
 --------------------------------------------
```

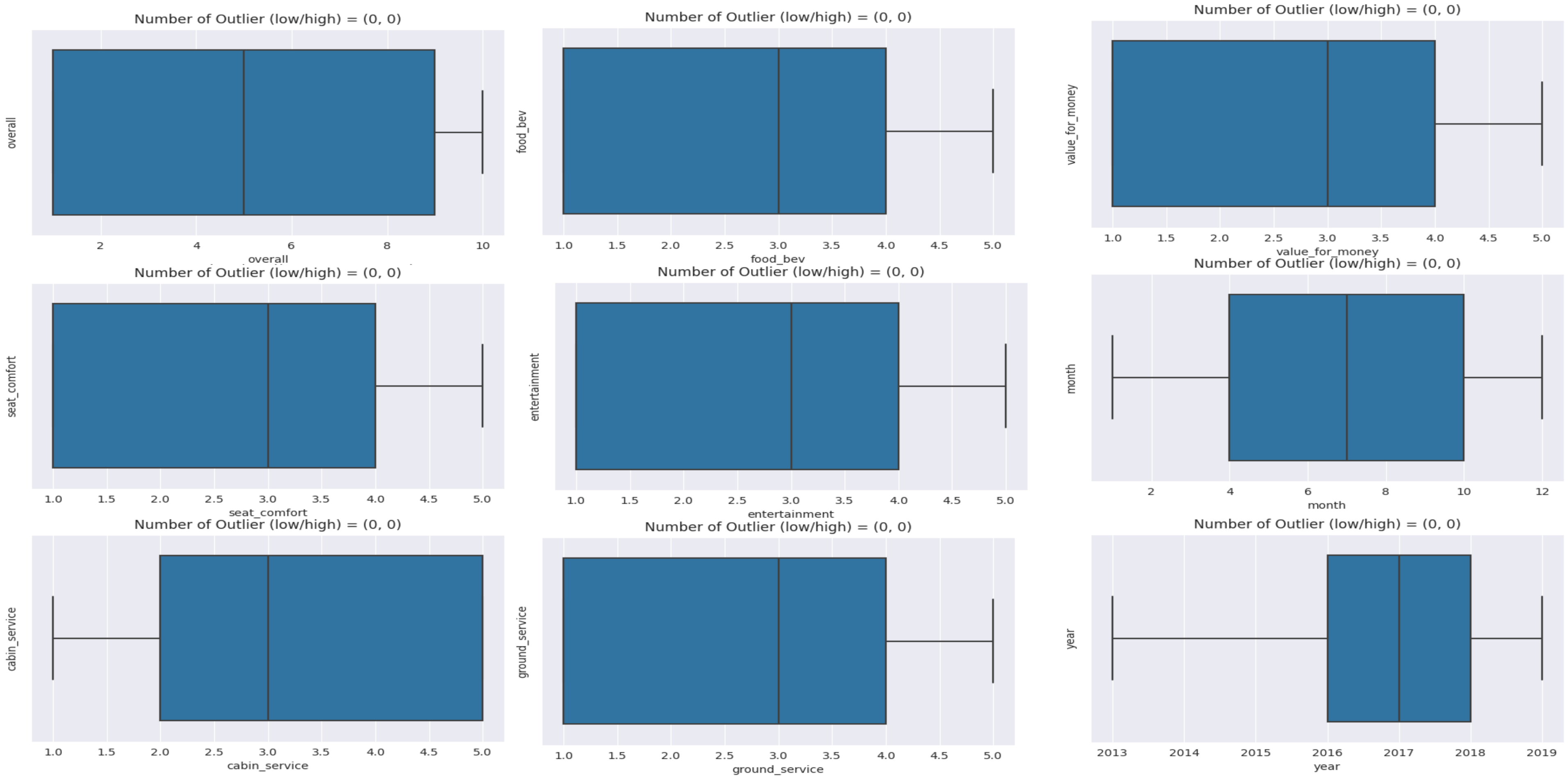
Observation:-

- The number of unique values present in dataset and the target variable has balanced data.
- The day variable has only single value in it, therefore this column may be dropped

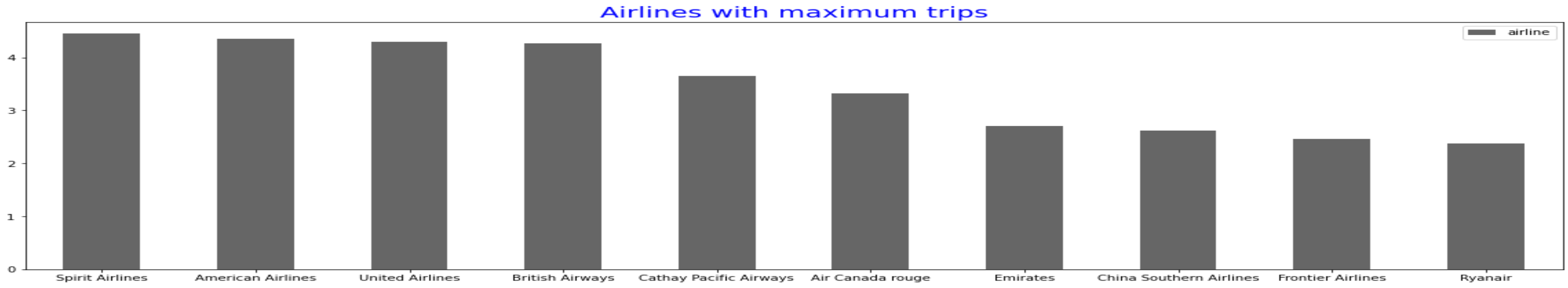
Exploratory Data Analysis(EDA):- Univariate distribution

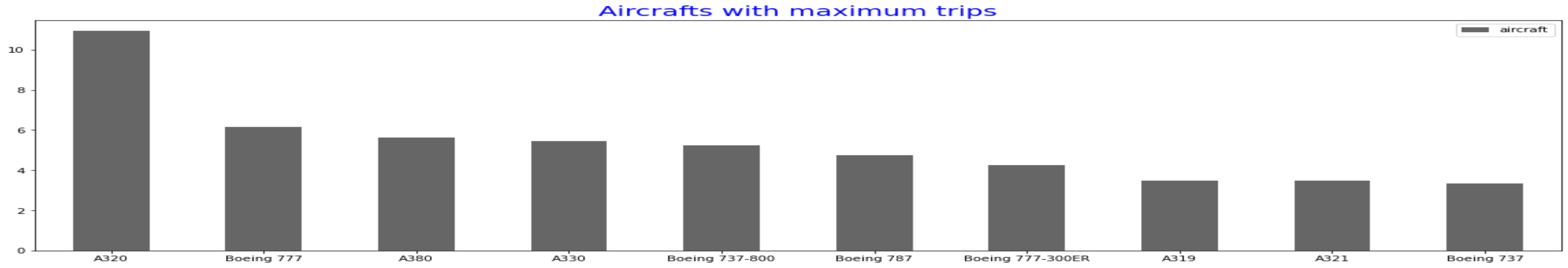


Outliers detection:



Analysed Maximum trip by Airline and top Aircrafts with maximum trips:

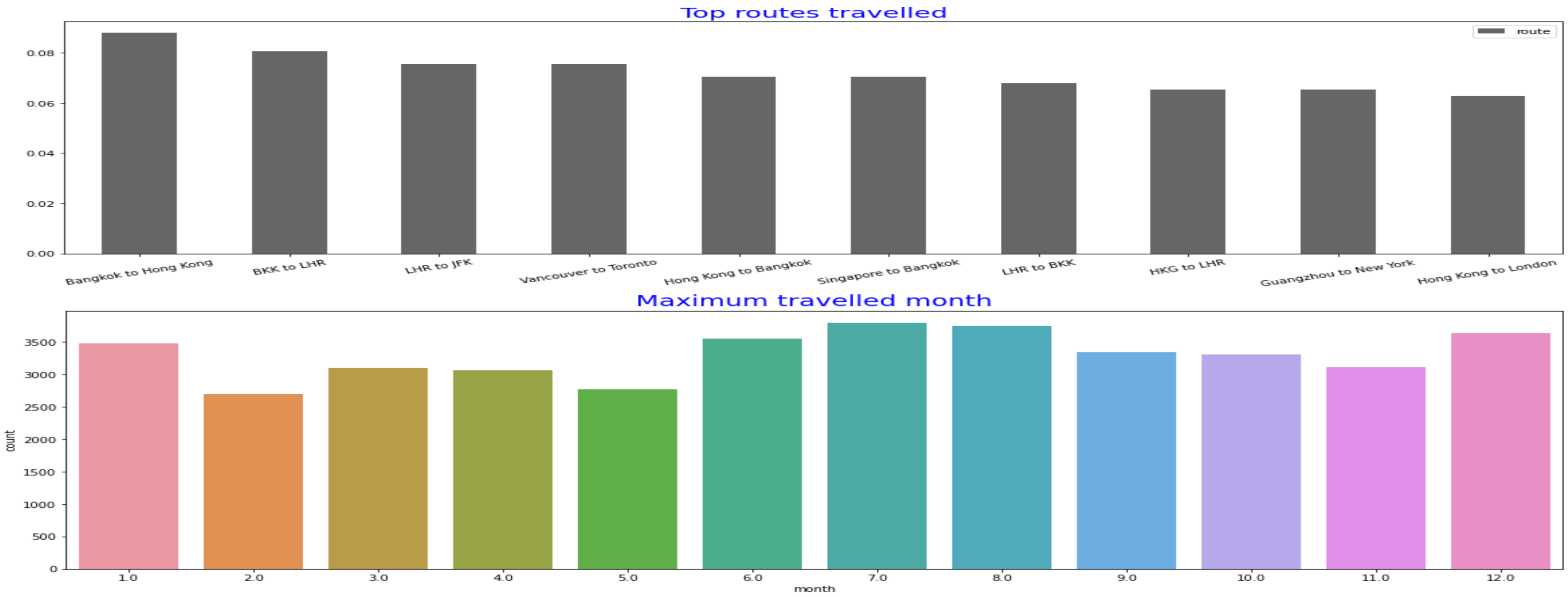




Observation:-

- According to the above analysis, Spirit Airlines holds the top spot for most journeys taken followed by American and United airlines.
- Airbus A320 Aircraft holds the top spot for most journeys taken followed by Boeing 777 and Airbus A380 aircraft.

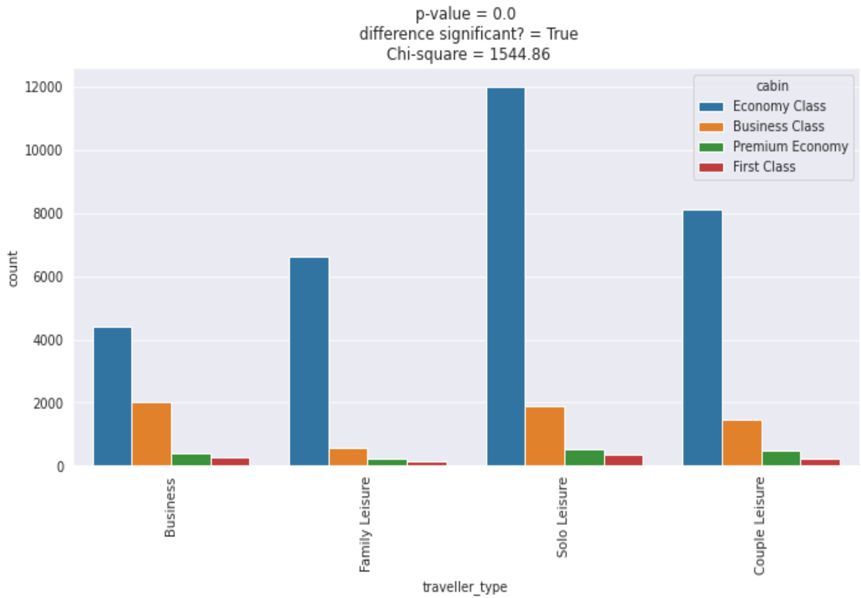
Top routes travelled across the world and Maximum Travelled Month:



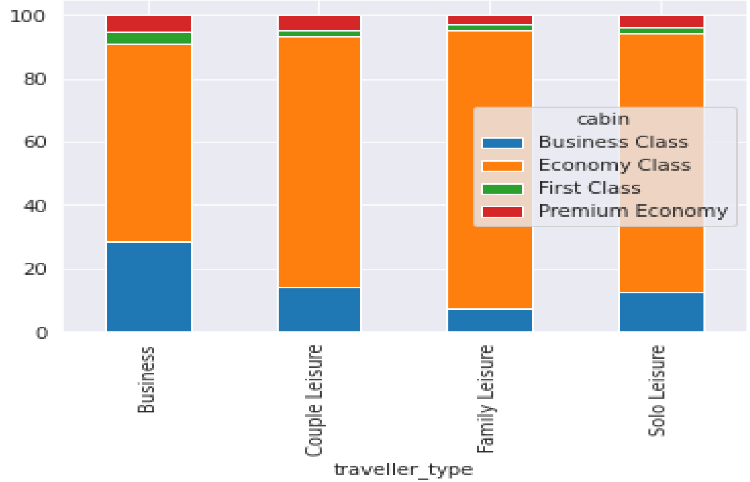
Observation:-

- According to the above analysis, Bangkok to Hong Kong journey tops the position followed by Bangkok to London and London to New York.
- The month of July is said to be the one with the highest travel. The second-most popular month for travel is December.

Bivariate Analysis:



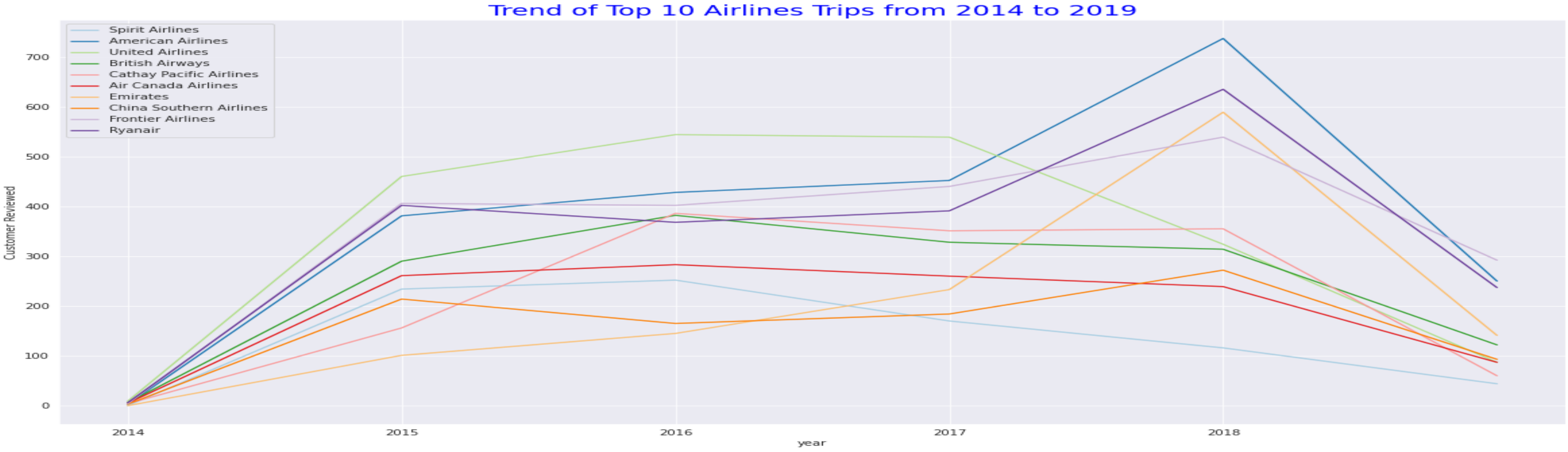
cabin	Business Class	Economy Class	First Class	Premium Economy
traveller_type				
Business	28.581510	62.187721	3.796754	5.434016
Couple Leisure	14.195369	78.984238	2.004281	4.816112
Family Leisure	7.596940	87.589027	1.899235	2.914798
Solo Leisure	12.914131	81.169709	2.251521	3.664638



Observation:-

- All types of traveller strongly prefer the economy class.
- Some of the Business class and Couple Leisure people choose business class for travelling.
- First class is least preferred among all traveller type categories.

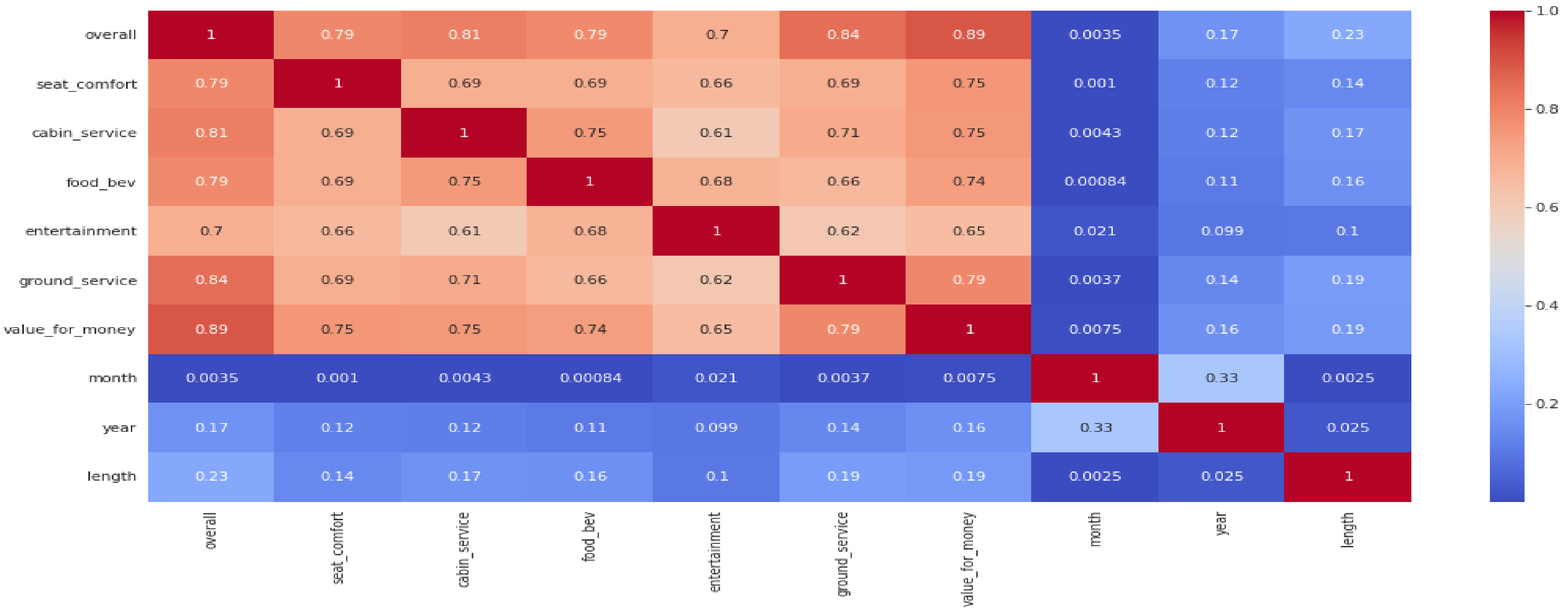
Trend of Top 10 Airlines Trips from 2014 to 2019:



Observation:-

- In its initial days American airlines was at low, but in 2018 it reaches to its all time peak and become top preferred airlines for their operations.
- United airlines of USA retained their top position from 2014 to 2017, then its number of reviews degraded further.
- Emirates airlines was least preferred in its initial days, but it increased its customer base in 2018 and retained their place in top 3 airlines.
- We have also analyzed above that spirit airlines is least preferred by travellers, However we can also see that in the above trend graph.

Identifying Multicollinearity:



From the Heatmap we can see that:

- **All the Rating features are very strongly correlated to each other and to reduce it we can remove all rating features except overall column and proceed for further analysis .**

Data Pre-processing:

1. As to Work on sentiment analysis of the Customer review as it will be referred to someone or not. we have selected only three feature i.e. Recommended, Overall and Customer review.
2. After removal of all the Null values still around 1487 Null values present in the target feature i.e. Recommended column.
3. We handled Null Values of Recommended feature with the help of feature engineering on Overall Ratting Column.

```
#Check null values of target variable where overall ratings given
(airlines_df['recommended'].isnull() & airlines_df['overall'].notna()).sum()
```

```
1487
```

```
#Fill recommended column null values based on overall ratings given
df2.loc[df2["overall"] <= 5.0, "recommended"] = 'no'
df2.loc[df2["overall"] > 5.0, "recommended"] = 'yes'
```

```
#Duplicate values in target variable
len(airlines_df.loc[airlines_df['recommended'] != airlines_df['recommended'] ])
```

```
0
```

Preparing dataset(Text Cleaning) of Customer review feature:

Text cleaning is the process of preparing raw text for NLP (Natural Language Processing) so that machines can understand human language. Following approach is used here to clean the text of customer reviews:

- **Use pos_tag with nltk**:- POS Tagging in NLTK is a process to mark up the words in text format for a particular part of a speech based on its definition and context.
- Remove all character which are excluded from "**a-z and A-Z**".
- Convert words into **Lowercase** and **split** them through space.
- Remove **stopwords** using **nltk** library.
- Lemmatization of reviews and get the meaningful words using **WordNetLemmatizer**.
- **Join** back the words that were split before.
- Initiate **tokenization** process.

```
len(cust_df['customer_review'][0])
```

```
1143
```

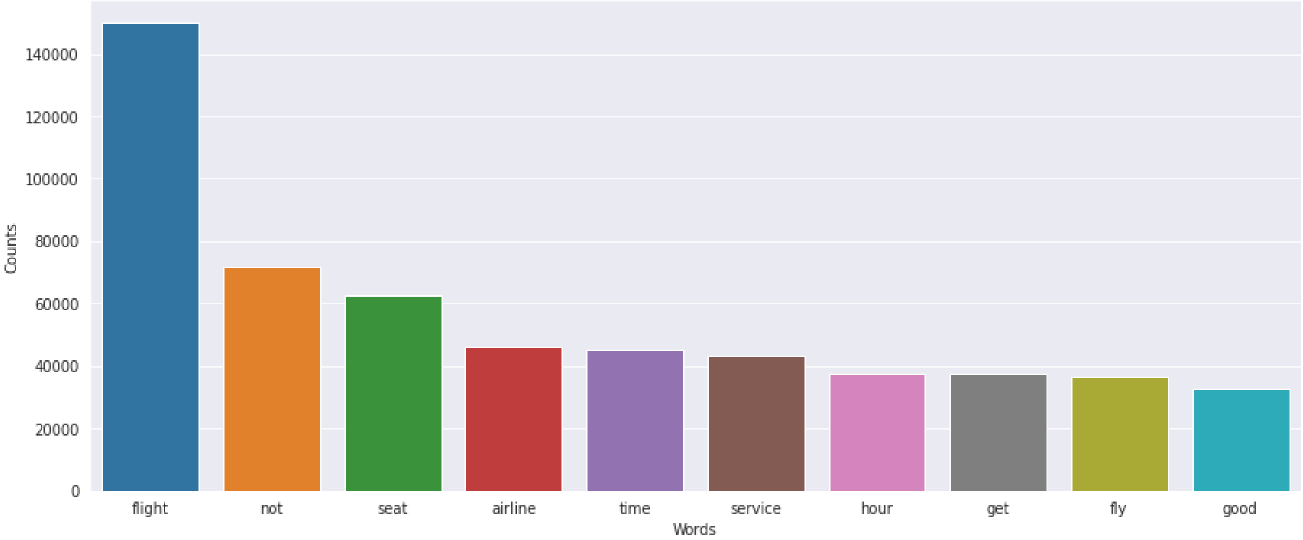
```
len(cust_df['tokenized_review'][0])
```

```
740
```

Observation:-

- We have checked length of 1st index of customer review before and after the text cleaning method by using NLTK (Natural Language Toolkit) and re (Regular Expression)

Identified Top 10 Words In Whole Customer Review Feature:



	Words	Counts
0	flight	149817
1	not	71986
2	seat	62693
3	airline	46015
4	time	45169
5	service	43398
6	hour	37602
7	get	37601
8	fly	36750
9	good	32761

Observation:-

- We Found that “Flight” Word has a count of 149817 in the whole customer review feature followed by “not”, ”Seat”, “airline”, etc.

Conversion of text data to numerical data for Model Understanding:

Count Vectorizer and TF-IDF is the mostly used method to convert text data into numerical data. For our project we have selected TF-IDF.

- **TF-IDF(Term Frequency and inverse Document frequency)**:-

1. It is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling.
2. The TF-IDF value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general. TF-IDF is one of the most popular term-weighting schemes today. A survey conducted in 2015 showed that 83% of text-based recommender systems in digital libraries use TF-IDF.

Model Building:

A machine learning model is a program that can find patterns or make decisions from a previously unseen data set. The process of running a machine learning algorithm on a dataset (called training data) and optimizing the algorithm to find certain patterns or outputs is called model training. The resulting function with rules and data structures is called the trained machine learning model.

We have used below some of the model for our project to achieve high accuracy result of Airline Passenger Referral Prediction

- **Logistic Regression.**
- **Naïve Bayes.**
- **Passive Aggressive Classifier.**
- **Random Forest Classifier.**
- **Gradient Boosting Classifier.**

Comparing evaluation metrics of the models being used:

```
#  Get performance on different models
final_result_df
```

	Model_Name	Train_Accuracy	Test_Accuracy	Precision_Train	Precision_Test	Recall_Train	Recall_test	ROC_AUC_Train	ROC_AUC_Test	AUC	Model_training_time
0	LogisticRegression	92.45	91.26	92.60	91.10	91.44	89.97	92.40	91.18	0.911773	31.547762
1	PassiveAggressiveClassifier	93.77	89.68	96.29	91.58	90.40	85.64	93.62	89.41	0.894093	26.339245
2	GradientBoostingClassifier	88.57	87.52	89.22	87.48	86.44	85.35	88.47	87.38	0.873795	1188.170723
3	GaussianNB	83.09	80.53	79.62	76.81	86.65	83.19	83.25	80.71	0.807097	4.745999
4	MultinomialNB	86.05	85.89	83.92	83.30	87.47	87.08	86.12	85.97	0.859732	1.059351
5	RandomForestClassifier	100.00	89.40	99.99	88.89	100.00	88.19	100.00	89.32	0.893173	242.026412

```
The train confusion matrix of LogisticRegression is :
 [[25792  1835]
 [ 2149 22965]]
 --------------------------------------------------
The train confusion matrix of PassiveAggressiveClassifier is :
 [[26753   874]
 [ 2412 22702]]
 --------------------------------------------------
The train confusion matrix of GradientBoostingClassifier is :
 [[25003  2624]
 [ 3405 21709]]
 --------------------------------------------------
The train confusion matrix of GaussianNB is :
 [[22058  5569]
 [ 3352 21762]]
 --------------------------------------------------
The train confusion matrix of MultinomialNB is :
 [[23418  4209]
 [ 3147 21967]]
 --------------------------------------------------
The train confusion matrix of RandomForestClassifier is :
 [[27625     2]
 [    0 25114]]
 --------------------------------------------------
```

```
The test confusion matrix of LogisticRegression is :
 [[25792  1835]
 [ 2149 22965]]
 --------------------------------------------------
The test confusion matrix of PassiveAggressiveClassifier is :
 [[26753   874]
 [ 2412 22702]]
 --------------------------------------------------
The test confusion matrix of GradientBoostingClassifier is :
 [[25003  2624]
 [ 3405 21709]]
 --------------------------------------------------
The test confusion matrix of GaussianNB is :
 [[22058  5569]
 [ 3352 21762]]
 --------------------------------------------------
The test confusion matrix of MultinomialNB is :
 [[23418  4209]
 [ 3147 21967]]
 --------------------------------------------------
The test confusion matrix of RandomForestClassifier is :
 [[27625     2]
 [    0 25114]]
 --------------------------------------------------
```

Model Performance Checked:

- We Implemented trained model on some of the random examples (manually feeded) to check performance and accuracy of the model.
- We used Sklearn Pipeline library to check model performance.

```
#Implement pipeline with vectorizer and selected model
model = Pipeline([('vectorizer',tfidf),('classifier',gb_model)])
model.fit(X_train, y_train)
```

```
Pipeline(steps=[('vectorizer', TfidfVectorizer(max_features=5700)),
                ('classifier',
                 GradientBoostingClassifier(max_depth=6, min_samples_leaf=30,
                                            min_samples_split=10,
                                            n_estimators=50))])
```

	Reviews	Recommended
0	it was an average flight but facilites can be ...	no
1	I had a great experience	yes
2	hospitality are good	yes
3	seats were good	yes
4	he is not satisfied	no
5	he is angry with staff behaviour	no
6	There was clean food available	yes
7	The flight was cancelled twice, flew with anot...	no
8	bigger boarding wa time orderly seat seemed ne...	yes
9	Experience was not good	no

Observation:-

- We found that model performed quite well on some example other than dataset provided.

Conclusion:

- The highest peak of the month feature is 7. According to legend, July is the month with the most travel. December is the second-most popular month.
- Most trips are taken by Spirit Airlines, which has the highest frequency in the dataset.
- The most trips taken were made by Airbus A320 aircraft, which had the highest frequency in the dataset, followed by Boeing 777 and Airbus A380 aircraft.
- Top spot pertains to the Bangkok to Hong Kong trip that occurred most frequently in the dataset, followed by Bangkok to London and London to New York trips.
- In the column for traveler type, it is noticeable to us that Solo Leisure travelers represent the majority of the population. In the cabin column, the majority of passengers prefer the Economy class.
- Following the use of bivariate analysis, we discovered that all travelers highly favor the economy class. Some Couple Leisure and Business class travelers choose to fly in business class. Among all traveler types, first class is the least popular.
- Based on customer satisfaction, Cathay Pacific Airways of Hong Kong is the most preferred airline. As indicated by stats, we were able to make several intriguing deductions, such as the Airbus A380 operated by Emirates Airlines being the most well-liked aircraft. However, based on ratings for all airlines, Emirates Airlines is not the most well-liked airline.

Conclusion to be Continue

- American Airlines was at a downtrend in its early years, but in 2018 it reached its highest peak ever and became the most popular airline for its operations. From 2014 until 2017, United Airlines of the USA kept the top spot, but after that, the performance of its operations continued to decline. In its early years, Emirates Airlines was the least popular, but in 2018 it grew its client base and kept its position in the top 3 airlines. We have also analyzed above that spirit airlines is least preferred by travelers, However we can also see that in the above trend graph.
- Due to the linear and balanced dataset, logistic regression outperformed the other algorithms well. Gradient boosting approach came in second.
- For this sort of dataset and its specified problem statement, accuracy and f1 score are the optimal evaluation matrix that is taken into consideration.

Problems faced During Project:

- Handling the null values and duplicates: It is evident from the analysis above that our dataset has a significant number of null values.
- We can observe that a lot of rating variables have strongly correlated with the overall rating column.
- The text in the customer review field was unformatted and included both alphanumeric and special characters.
- We were unable to train the model with more data due to the computational complexity

THANK YOU