Opening a new Bermese Restaurant in Toronto

APPLIED DATA SCIENCE CAPSTONE

Report Prepared by : Mohamed Abuabchal

(https://github.com/abuabchal/testrepo/tree/main/Applied%20Data%20Science%20Capstone)

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
- Discussion
- Conclusion

EXECUTIVE SUMMARY

- This Capstone project focuses on identifying the most suitable location for opening an authentic Burmese restaurant in Toronto, Canada. Given the scarcity of Burmese restaurants in the area, this presents a potential business opportunity for an entrepreneur based in Canada. The entrepreneur aims to open the restaurant in neighborhoods where Asian cuisine is popular, leveraging the similarities between Burmese and other Asian foods.

Introduction

- For this Capstone project, I am creating a hypothetical scenario for a concept Burmese restaurateur who wants to explore opening an authentic Burmese restaurant in Toronto area. The idea behind this project is that there may not be enough Burmese restaurants in Toronto and it might present a great opportunity for this entrepreneur who is based in Canada. As Burmese food is very similar to other Asian cuisines, this entrepreneur is thinking of opening this restaurant in locations where Asian food is popular (aka many Asian restaurants in the neighborhood). With the purpose in mind, finding the location to open such a restaurant is one of the most important decisions for this entrepreneur and I am designing this project to help him find the most suitable location.

Methodology

- The overall methodology includes:
  1. Data collection, wrangling, and formatting, using:
     - SpaceX API
     - Web scraping
  2. Exploratory data analysis (EDA), using:
     - Pandas and NumPy
     - SQL
  3. Data visualization, using:
     - Matplotlib and Seaborn
     - Folium

METHODOLOGY

Data collection, wrangling, and formatting

First, I need to get the list of neighborhoods in Toronto, Canada. This is possible by extracting the list of neighborhoods from wikipedia page

(“ https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M ”) I did the web scraping by utilizing pandas html table scraping method as it is easier and more convenient to pull tabular data directly from a web page into dataframe.

However, it is only a list of neighborhood names and postal codes. I will need to get their coordinates to utilize Foursquare to pull the list of venues near these neighborhoods. To get the coordinates, I tried using Geocoder package but it was not working so I used the csv file provided by IBM team to match the coordinates of Toronto neighborhoods. After gathering all these coordinates, I visualized the map of Toronto using Folium package to verify whether these are correct coordinates

METHODOLOGY

Data collection, wrangling, and formatting

First, I need to get the list of neighborhoods in Toronto, Canada. This is possible by extracting the list of neighborhoods from wikipedia page

(“ https://en.wikipedia.org/wiki/List_of_postal_codes_of_Canada:_M ”) I did the web scraping by utilizing pandas html table scraping method as it is easier and more convenient to pull tabular data directly from a web page into dataframe.

However, it is only a list of neighborhood names and postal codes. I will need to get their coordinates to utilize Foursquare to pull the list of venues near these neighborhoods. To get the coordinates, I tried using Geocoder package but it was not working so I used the csv file provided by IBM team to match the coordinates of Toronto neighborhoods. After gathering all these coordinates, I visualized the map of Toronto using Folium package to verify whether these are correct coordinates

METHODOLOGY

	Postcode	Borough	Neighbourhood
0	M1B	Scarborough	Rouge, Malvern
1	M1C	Scarborough	Highland Creek, Rouge Hill, Port Union
2	M1E	Scarborough	Guildwood, Morningside, West Hill
3	M1G	Scarborough	Woburn
4	M1H	Scarborough	Cedarbrae

Out[16]:

	Postal Code	Latitude	Longitude
0	M1B	43.806686	-79.194353
1	M1C	43.784535	-79.160497
2	M1E	43.763573	-79.188711
3	M1G	43.770992	-79.216917
4	M1H	43.773136	-79.239476

METHODOLOGY

Exploratory Data Analysis (EDA)

Pandas and NumPy

Functions from the Pandas and NumPy libraries are used to derive basic information about the data collected, which includes:

- The number of launches on each launch site
- The number of occurrence of each orbit
- The number and occurrence of each mission outcome

SQL

The data is queried using SQL to answer several questions about the data such as:

- The names of the unique launch sites in the space mission
- The total payload mass carried by boosters launched by NASA (CRS)
- The average payload mass carried by booster version F9 v1.1

METHODOLOGY

Data Visualization

Matplotlib and Seaborn

Functions from the Matplotlib and Seaborn libraries are used to visualize the data through scatterplots, bar charts, and line charts.

The plots and charts are used to understand more about the relationships between several features, such as:

- The relationship between flight number and launch site
- The relationship between payload mass and launch site
- The relationship between success rate and orbit type

Folium

Functions from the Folium libraries are used to visualize the data through interactive maps.

The Folium library is used to:

- Mark all launch sites on a map
- Mark the succeeded launches and failed launches for each site on the map
- Mark the distances between a launch site to its proximities such as the nearest city, railway, or highway

METHODOLOGY

Machine Learning Prediction

Functions from the Scikit-learn library are used to create our machine learning models.

The machine learning prediction phase include the following steps:

- Standardizing the data
- Splitting the data into training and test data
- Creating machine learning models, which include:
  - Logistic regression
  - Support vector machine (SVM)
  - Decision tree
  - K nearest neighbors (KNN)
- Fit the models on the training set
- Find the best combination of hyperparameters for each model
- Evaluate the models based on their accuracy scores and confusion matrix

RESULTS

- The results are split into 5 sections:
  - SQL (EDA with SQL)
  - Matplotlib and Seaborn (EDA with Visualization)
  - Folium
  - Dash
  - Predictive Analysis
- In all of the graphs that follow, class 0 represents a failed launch outcome while class 1 represents a successful launch outcome.

RESULTS

SQL (EDA with SQL)

t[16]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	0	0	0	None	1.0	0	B0003	-80.577366	28.561857	0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	0	0	0	None	1.0	0	B0005	-80.577366	28.561857	0
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	0	0	0	None	1.0	0	B0007	-80.577366	28.561857	0
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	0	0	0	None	1.0	0	B1003	-120.610829	34.632093	0
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	0	0	0	None	1.0	0	B1004	-80.577366	28.561857	0

RESULTS

SQL (EDA with SQL)

```
   FlightNumber        Date BoosterVersion  PayloadMass Orbit    LaunchSite  \
0             1  2010-06-04       Falcon 9  6104.959412   LEO  CCAFS SLC 40   
1             2  2012-05-22       Falcon 9   525.000000   LEO  CCAFS SLC 40   
2             3  2013-03-01       Falcon 9   677.000000   ISS  CCAFS SLC 40   
3             4  2013-09-29       Falcon 9   500.000000    PO   VAFB SLC 4E   
4             5  2013-12-03       Falcon 9  3170.000000   GTO  CCAFS SLC 40   

       Outcome  Flights  GridFins  Reused   Legs LandingPad  Block  \
0    None None        1     False   False  False        NaN    1.0   
1    None None        1     False   False  False        NaN    1.0   
2    None None        1     False   False  False        NaN    1.0   
3  False Ocean        1     False   False  False        NaN    1.0   
4    None None        1     False   False  False        NaN    1.0   

   ReusedCount Serial   Longitude   Latitude  Class  
0            0  B0003  -80.577366  28.561857      0  
1            0  B0005  -80.577366  28.561857      0  
2            0  B0007  -80.577366  28.561857      0  
3            0  B1003 -120.610829  34.632093      0  
4            0  B1004  -80.577366  28.561857      0  
Index(['FlightNumber', 'Date', 'BoosterVersion', 'PayloadMass', 'Orbit',
       'LaunchSite', 'Outcome', 'Flights', 'GridFins', 'Reused', 'Legs',
       'LandingPad', 'Block', 'ReusedCount', 'Serial', 'Longitude', 'Latitude',
       'Class'],
      dtype='object')
```

RESULTS

SQL (EDA with SQL)

```
   FlightNumber        Date BoosterVersion  PayloadMass Orbit    LaunchSite  \
0             1  2010-06-04       Falcon 9  6104.959412   LEO  CCAFS SLC 40
1             2  2012-05-22       Falcon 9   525.000000   LEO  CCAFS SLC 40
2             3  2013-03-01       Falcon 9   677.000000   ISS  CCAFS SLC 40
3             4  2013-09-29       Falcon 9   500.000000    PO   VAFB SLC 4E
4             5  2013-12-03       Falcon 9  3170.000000   GTO  CCAFS SLC 40

       Outcome  Flights  GridFins  Reused   Legs LandingPad  Block  \
0    None None        1     False   False  False        NaN    1.0
1    None None        1     False   False  False        NaN    1.0
2    None None        1     False   False  False        NaN    1.0
3  False Ocean        1     False   False  False        NaN    1.0
4    None None        1     False   False  False        NaN    1.0

   ReusedCount Serial   Longitude   Latitude  Class
0            0  B0003  -80.577366  28.561857      0
1            0  B0005  -80.577366  28.561857      0
2            0  B0007  -80.577366  28.561857      0
3            0  B1003 -120.610829  34.632093      0
4            0  B1004  -80.577366  28.561857      0
Index(['FlightNumber', 'Date', 'BoosterVersion', 'PayloadMass', 'Orbit',
       'LaunchSite', 'Outcome', 'Flights', 'GridFins', 'Reused', 'Legs',
       'LandingPad', 'Block', 'ReusedCount', 'Serial', 'Longitude', 'Latitude',
       'Class'],
      dtype='object')
```

RESULTS

SQL (EDA with SQL)

Out[26]:

```
Borough
Central Toronto       9
Downtown Toronto     19
East Toronto          5
East York             5
Etobicoke            11
Mississauga           1
North York           24
Queen's Park          1
Scarborough          17
West Toronto          6
York                  5
Name: Neighborhood, dtype: int64
```

Out[60]:

	Neighborhood	Thai Restaurant	Cluster Labels	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
2	Brockton, Exhibition Place, Parkdale Village	0.0	2	43.636847	-79.428191	Pure Yoga Toronto	43.637330	-79.423800	Yoga Studio
38	The Danforth West, Riverdale	0.0	2	43.679557	-79.352188	Cafe Fiorentina	43.677743	-79.350115	Italian Restaurant
38	The Danforth West, Riverdale	0.0	2	43.679557	-79.352188	Athen's Pastries	43.678166	-79.348927	Greek Restaurant
38	The Danforth West, Riverdale	0.0	2	43.679557	-79.352188	Book City	43.677413	-79.352734	Bookstore
38	The Danforth West, Riverdale	0.0	2	43.679557	-79.352188	Il Fornello	43.678604	-79.346904	Italian Restaurant
...	...	...	...	...	...	...	...	...	...
23	Little Portugal, Trinity	0.0	2	43.647927	-79.419750	Pilot Coffee Roasters	43.646610	-79.419606	Coffee Shop
23	Little Portugal, Trinity	0.0	2	43.647927	-79.419750	The Goods	43.649259	-79.424022	Vegetarian / Vegan Restaurant
23	Little Portugal, Trinity	0.0	2	43.647927	-79.419750	The Tampered Press	43.650062	-79.417280	Coffee Shop
23	Little Portugal, Trinity	0.0	2	43.647927	-79.419750	Trinity Bellwoods Park	43.647072	-79.413756	Park
38	The Danforth West, Riverdale	0.0	2	43.679557	-79.352188	Starbucks	43.678879	-79.346357	Coffee Shop

566 rows × 9 columns

RESULTS

SQL (EDA with SQL)

- The total payload mass carried by boosters launched by NASA (CRS)

Total payload mass by NASA (CRS)
45596

- The average payload mass carried by booster version F9 v1.1

Average payload mass by Booster Version F9 v1.1
2928

- The date when the first successful landing outcome in ground pad was achieved

Date of first successful landing outcome in ground pad
2015-12-22

RESULTS

SQL (EDA with SQL)

- The names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

- The total number of successful and failure mission outcomes

number_of_success_outcomes	number_of_failure_outcomes
100	1

RESULTS

SQL (EDA with SQL)

- The names of the booster versions which have carried the maximum payload mass

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

RESULTS

SQL (EDA with SQL)

- The failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

DATE	booster_version	launch_site
2015-01-10	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	F9 v1.1 B1015	CCAFS LC-40

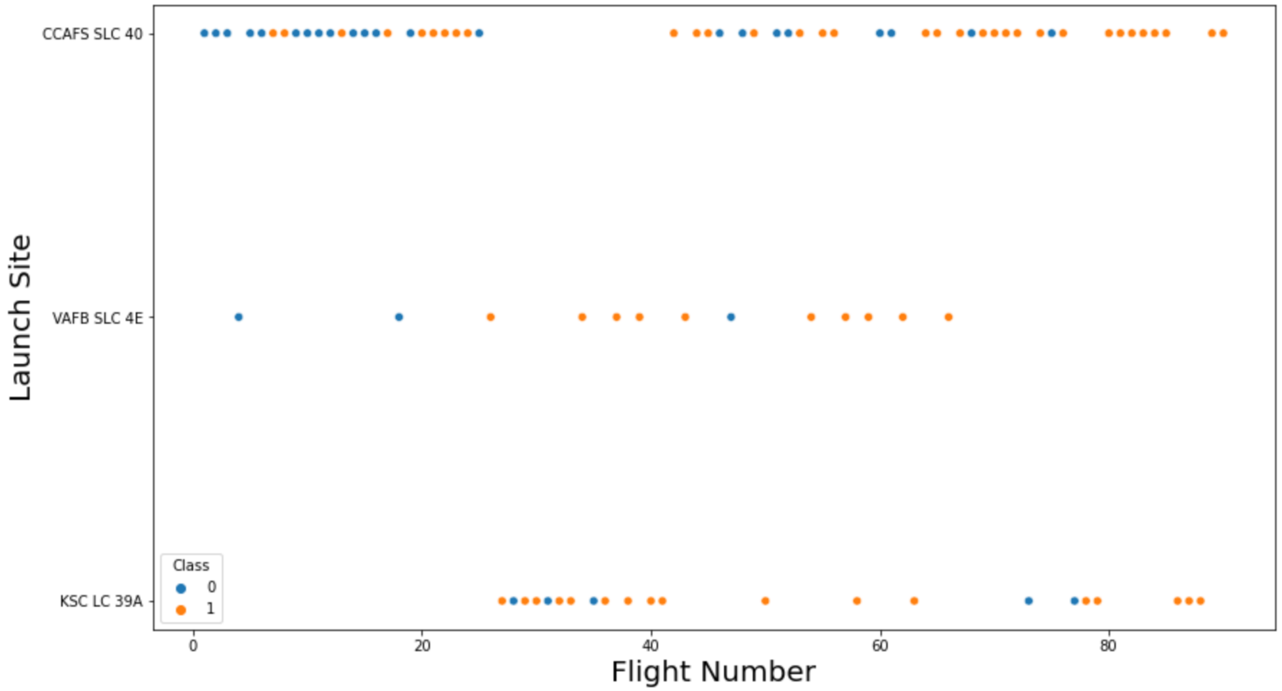
- The count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order

landing__outcome	landing_count
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

RESULTS

Matplotlib and Seaborn (EDA with Visualization)

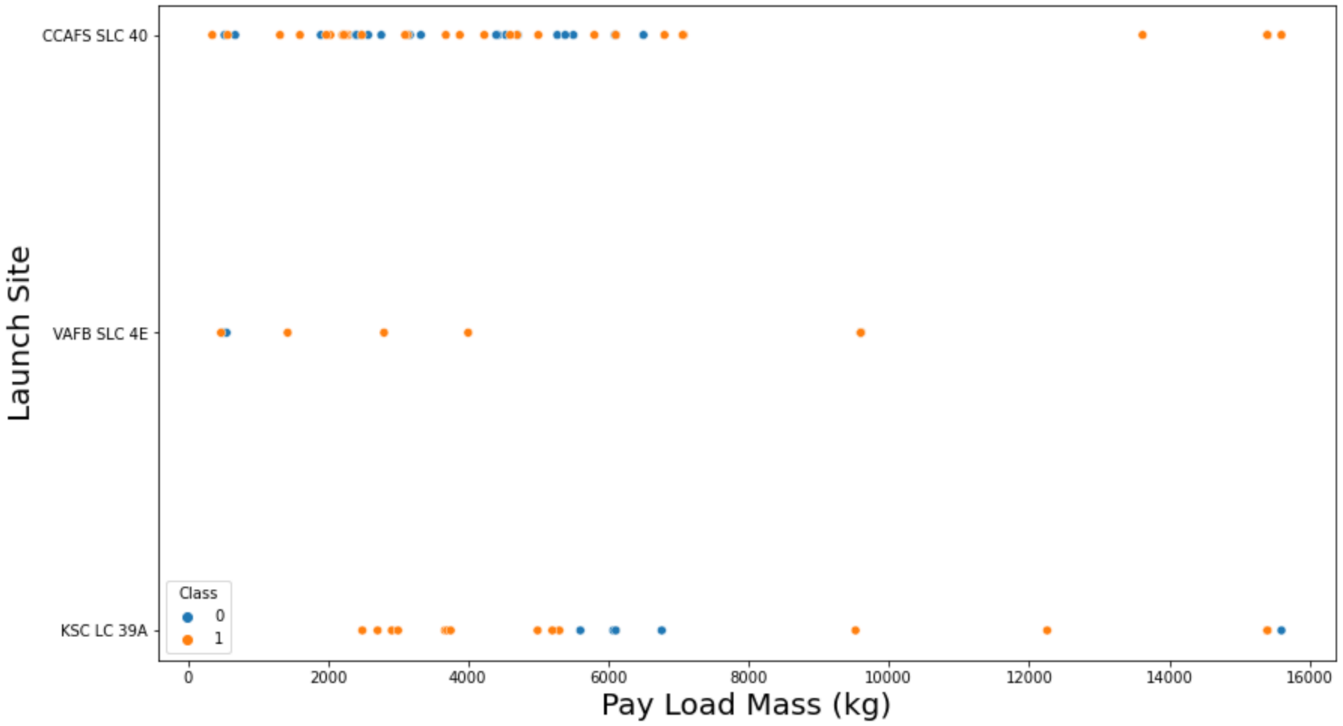
The relationship between flight number and launch site



RESULTS

Matplotlib and Seaborn (EDA with Visualization)

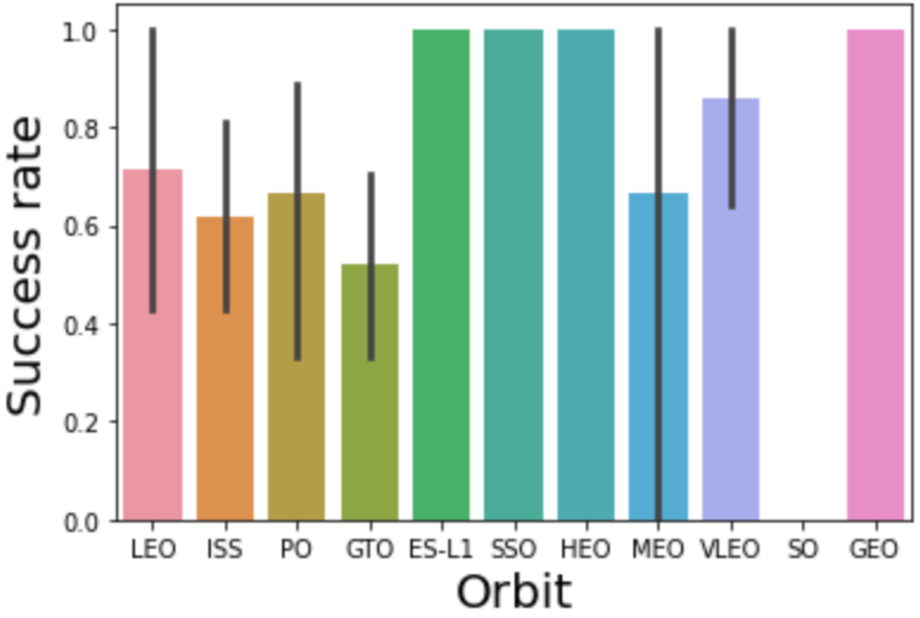
The relationship between payload mass and launch site



RESULTS

Matplotlib and Seaborn (EDA with Visualization)

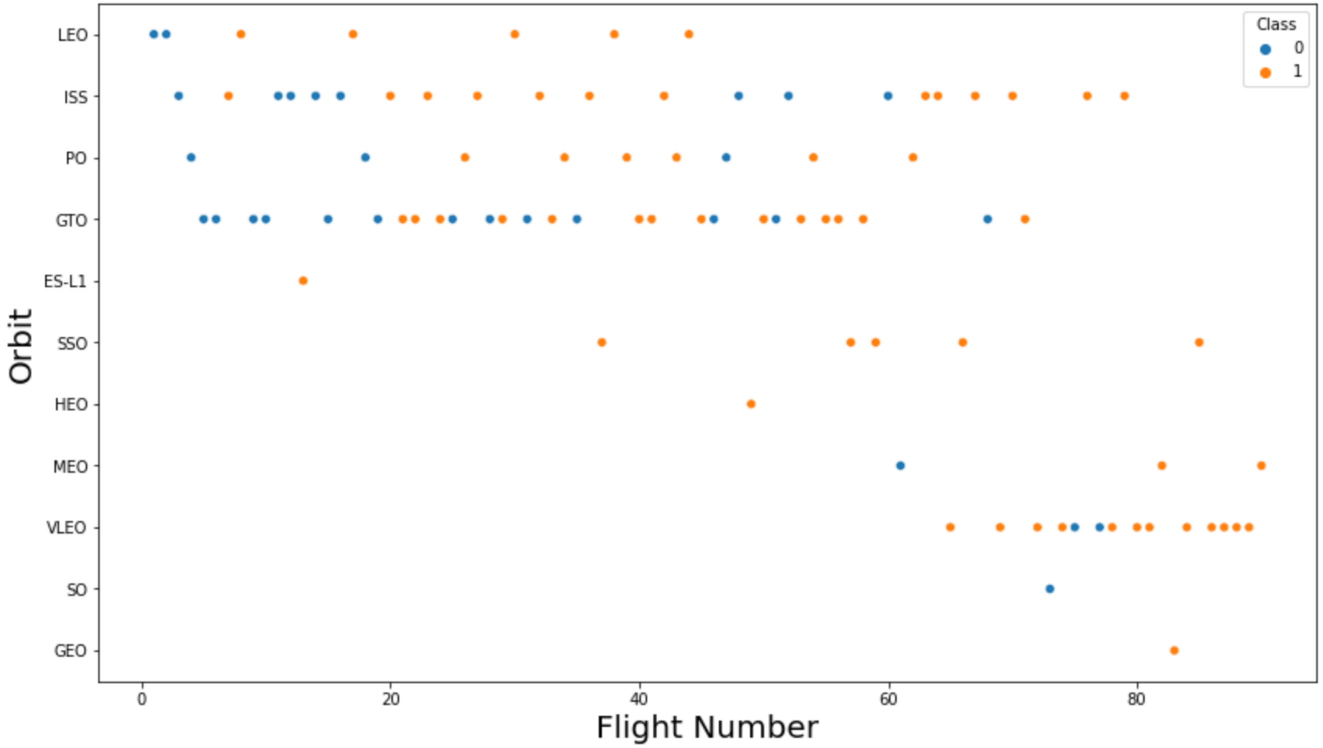
- The relationship between success rate and orbit type



RESULTS

Matplotlib and Seaborn (EDA with Visualization)

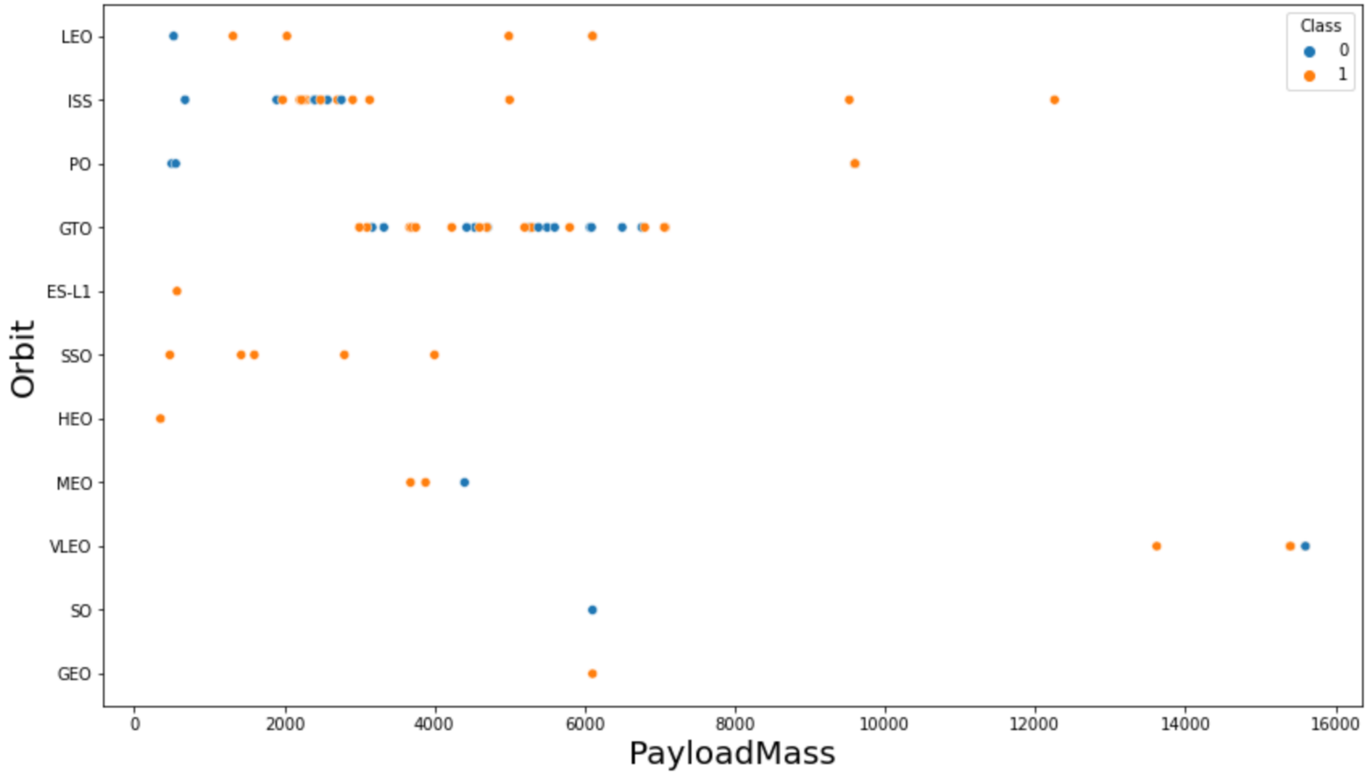
- The relationship between flight number and orbit type



RESULTS

Matplotlib and Seaborn (EDA with Visualization)

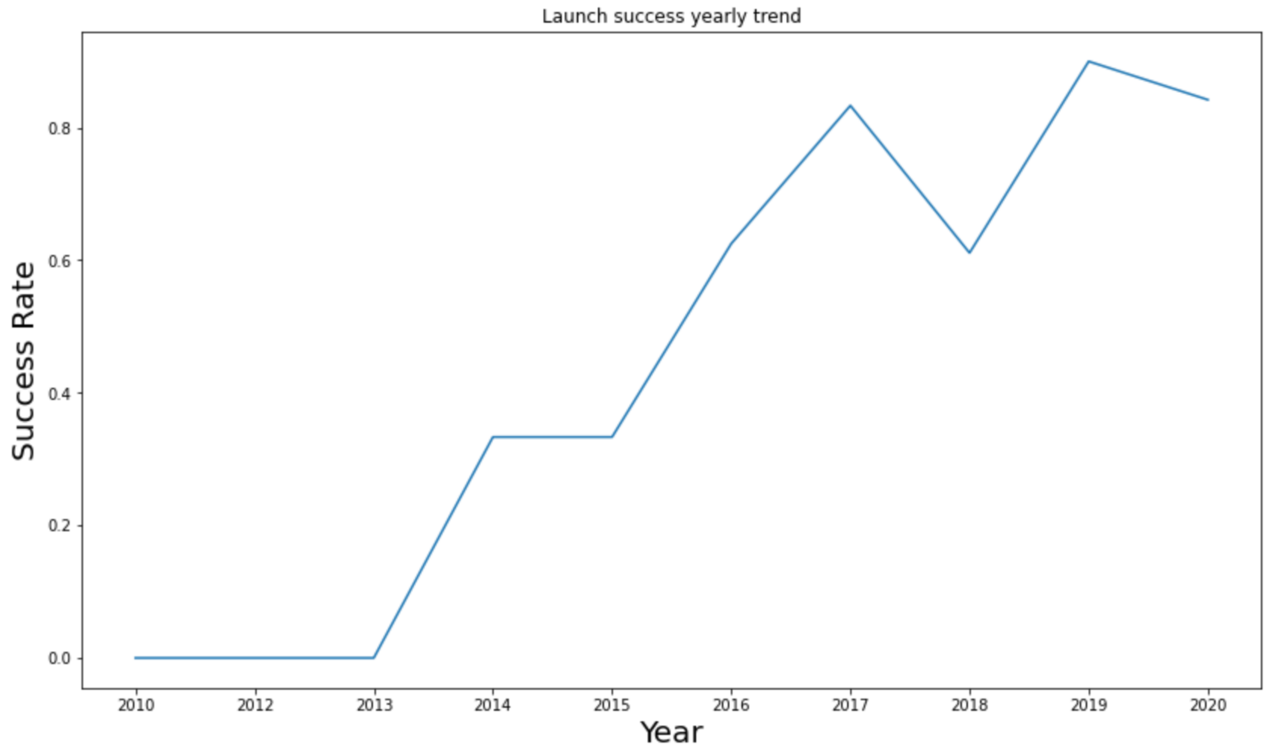
- The relationship between payload mass and orbit type



RESULTS

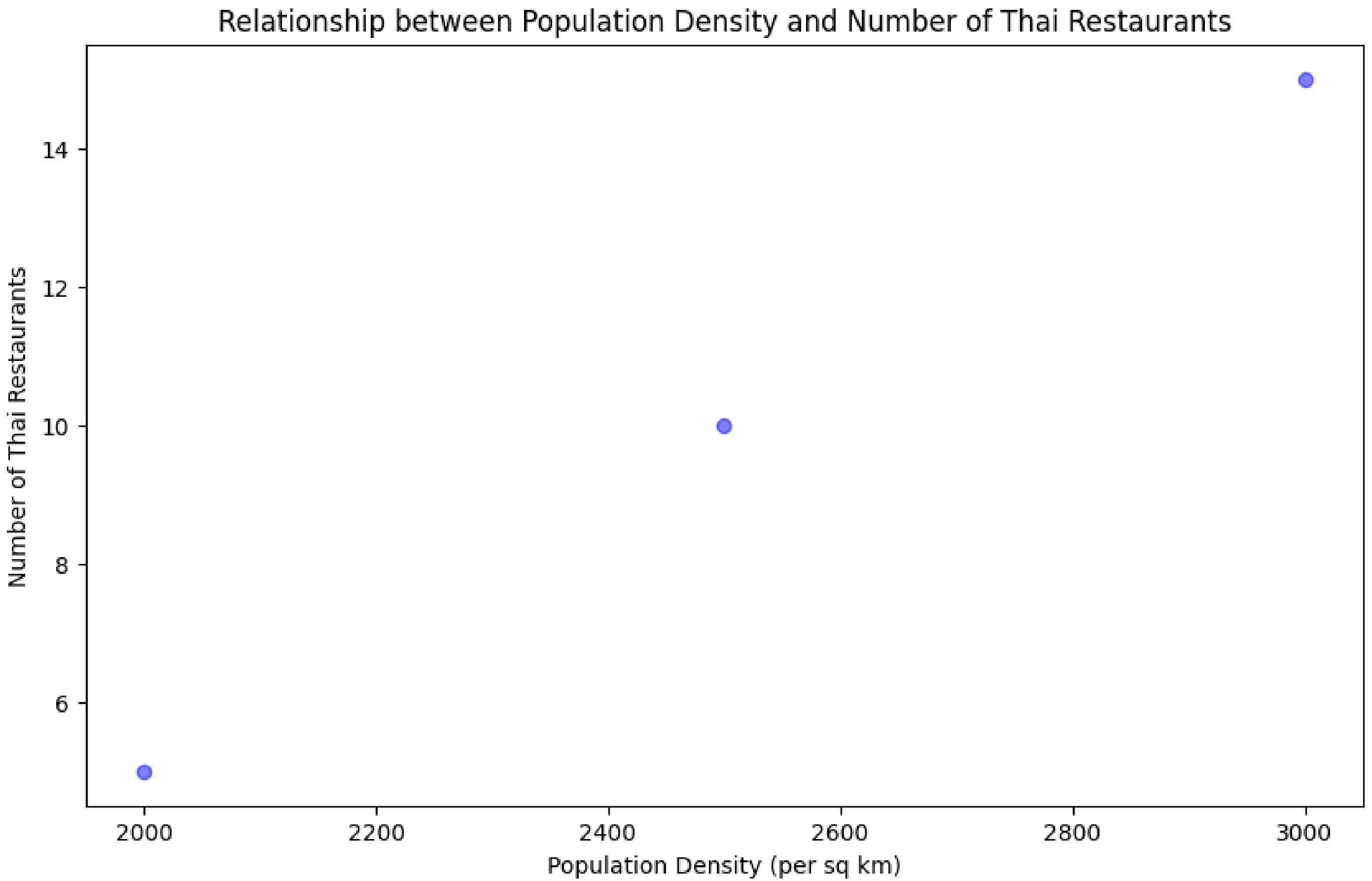
Matplotlib and Seaborn (EDA with Visualization)

- The launch success yearly trend



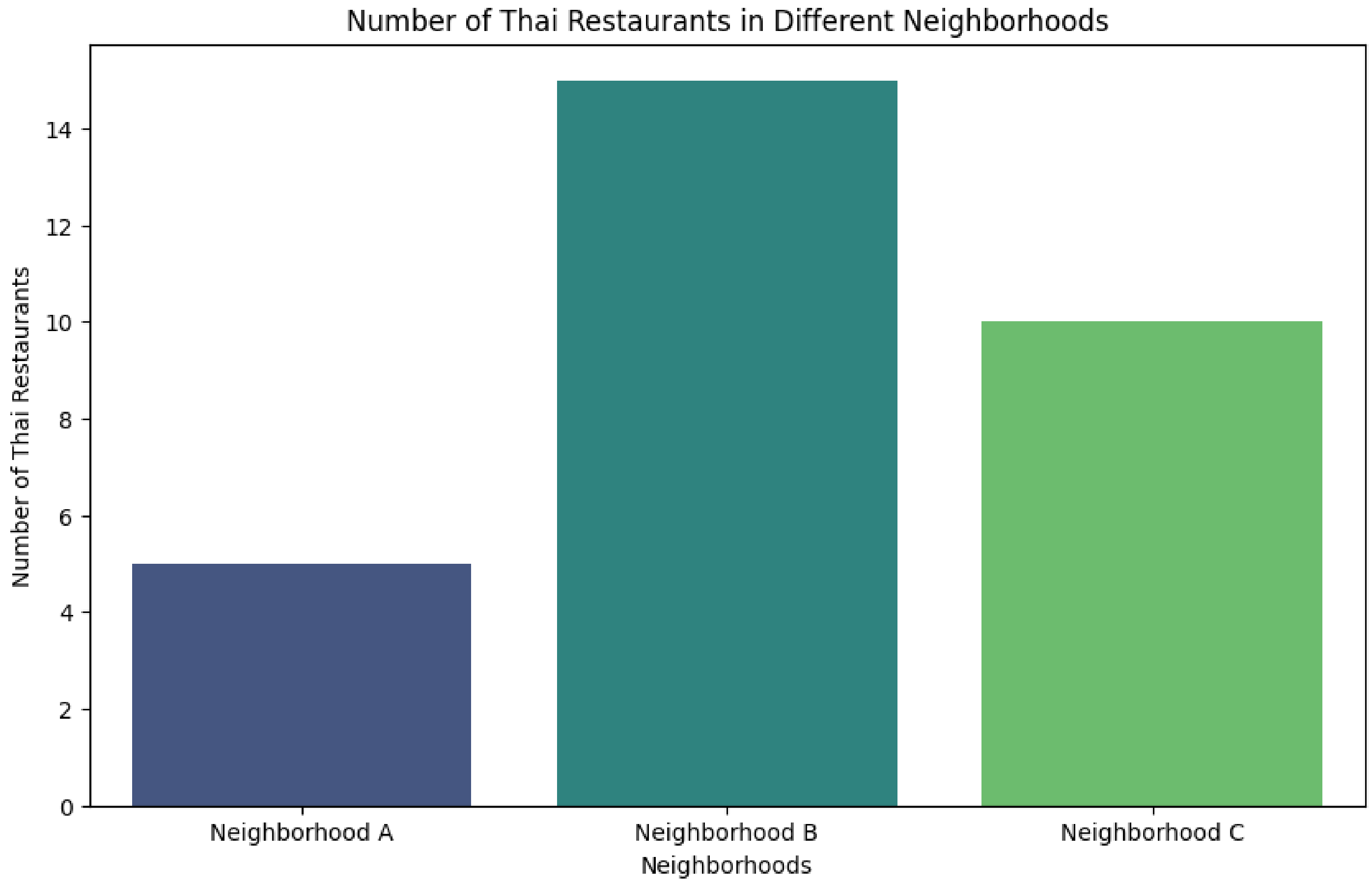
RESULTS

Matplotlib and Seaborn (EDA with Visualization)



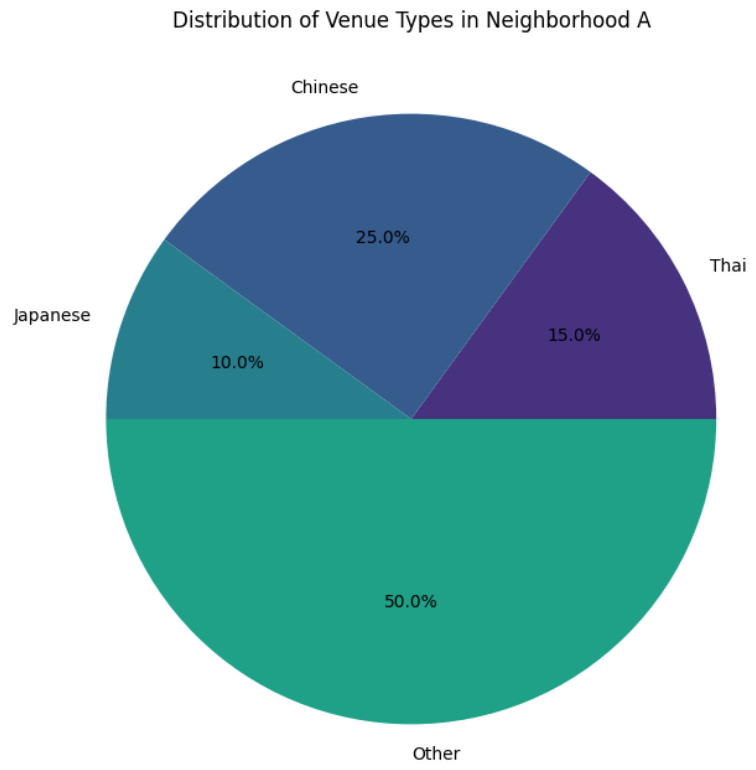
RESULTS

Matplotlib and Seaborn (EDA with Visualization)



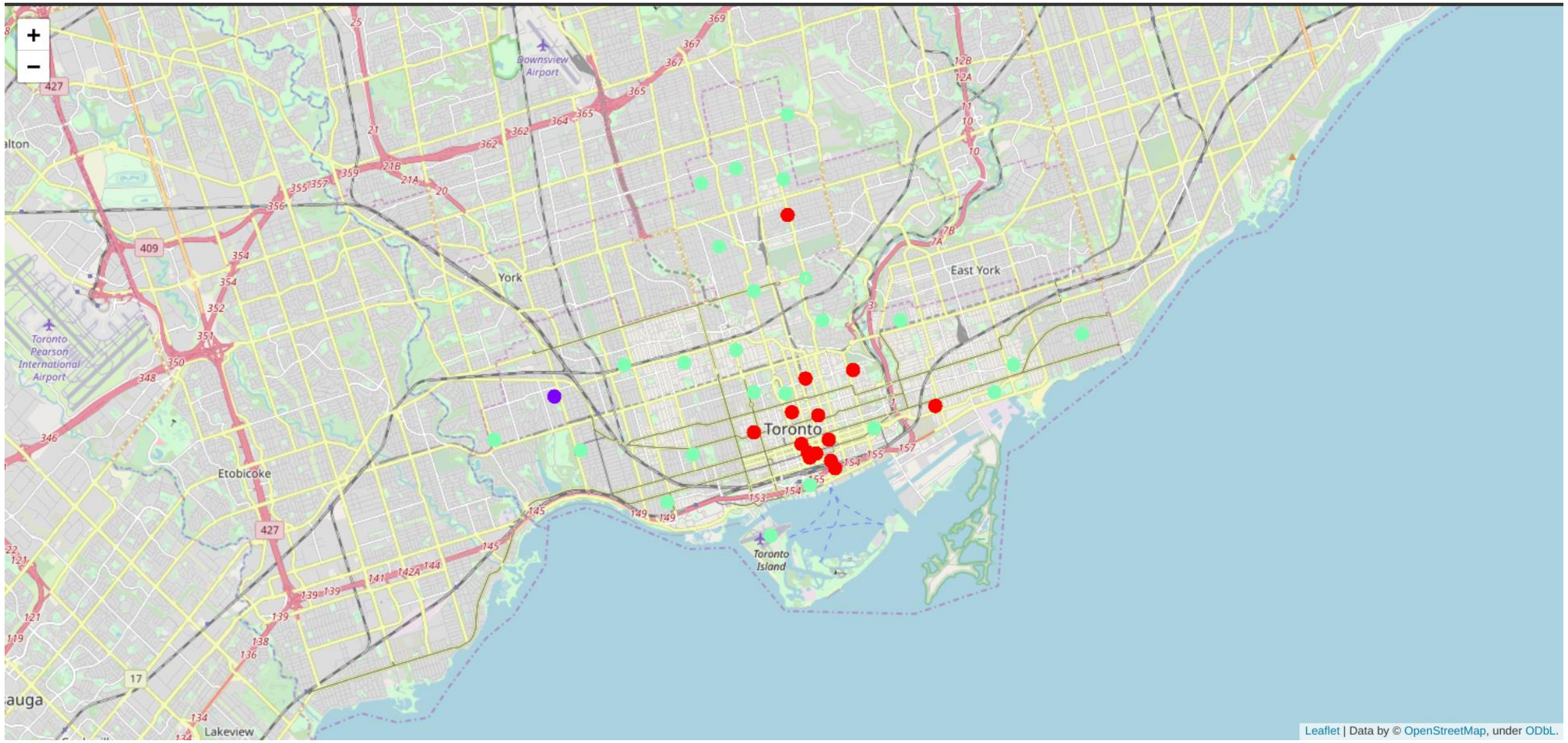
RESULTS

Matplotlib and Seaborn (EDA with Visualization)



RESULTS

Folium



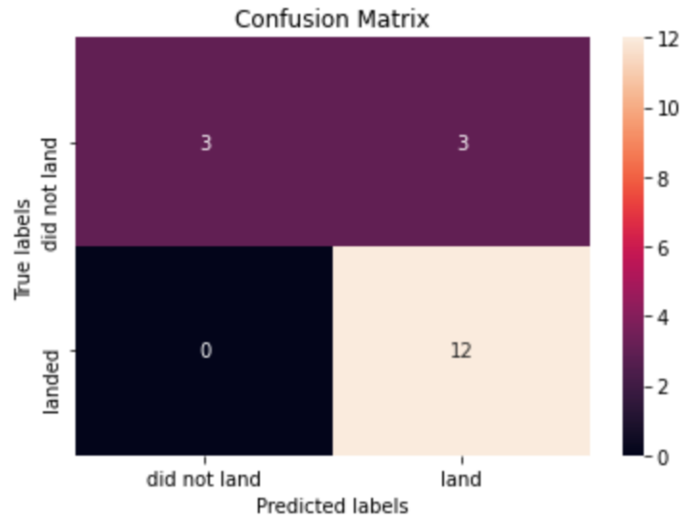
The succeeded launches and failed launches for each site on map

- If we zoom in on one of the launch site, we can see green and red tags. Each green tag represents a successful launch while each red tag represents a failed launch

RESULTS

Predictive Analysis

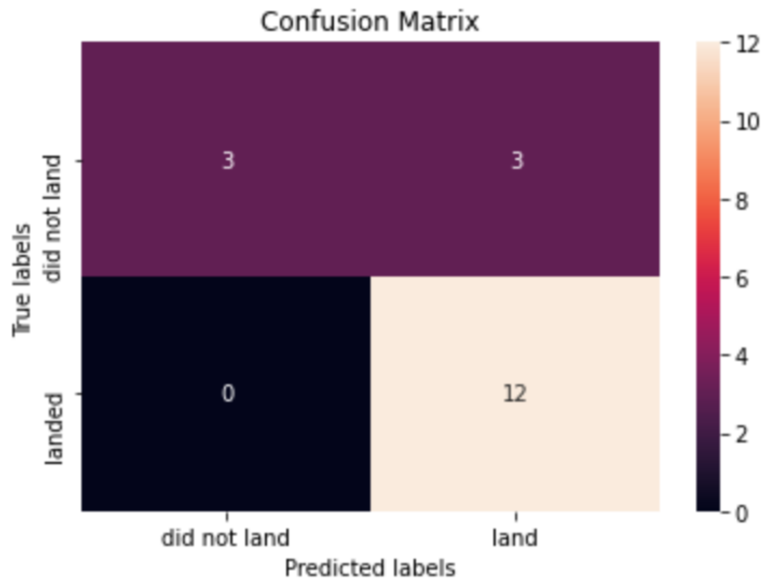
- Logistic regression
  - GridSearchCV best score: 0.8464285714285713
  - Accuracy score on test set: 0.8333333333333334
  - Confusion matrix:



RESULTS

Predictive Analysis

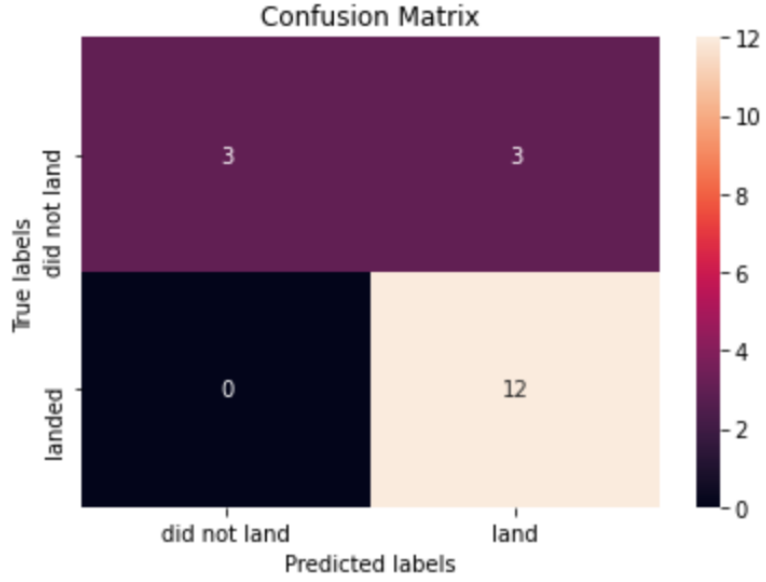
- Support vector machine (SVM)
  - GridSearchCV best score: 0.8482142857142856
  - Accuracy score on test set: 0.8333333333333334
  - Confusion matrix:



RESULTS

Predictive Analysis

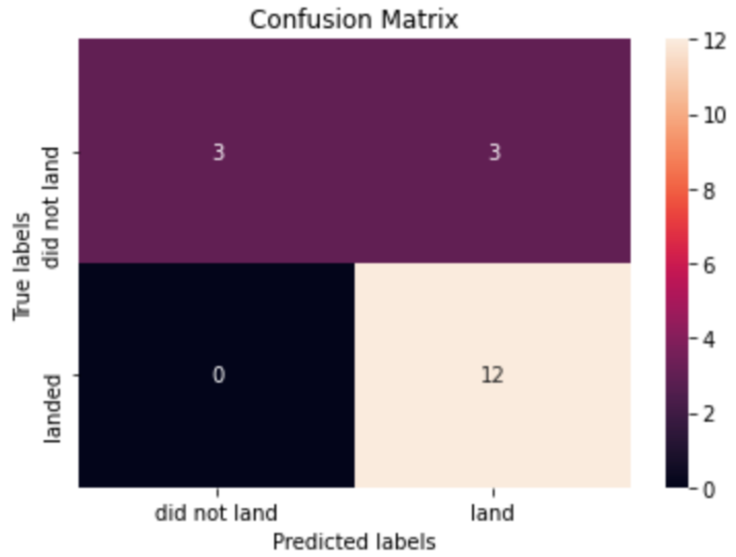
- Decision tree
  - GridSearchCV best score: 0.8892857142857142
  - Accuracy score on test set: 0.8333333333333334
  - Confusion matrix:



RESULTS

Predictive Analysis

- K nearest neighbors (KNN)
  - GridSearchCV best score: 0.8482142857142858
  - Accuracy score on test set: 0.8333333333333334
  - Confusion matrix:



RESULTS

Predictive Analysis

- Putting the results of all 4 models side by side, we can see that they all share the same accuracy score and confusion matrix when tested on the test set.
- Therefore, their GridSearchCV best scores are used to rank them instead. Based on the GridSearchCV best scores, the models are ranked in the following order with the first being the best and the last one being the worst:
  1. Decision tree (GridSearchCV best score: 0.8892857142857142)
  2. K nearest neighbors, KNN (GridSearchCV best score: 0.8482142857142858)
  3. Support vector machine, SVM (GridSearchCV best score: 0.8482142857142856)
  4. Logistic regression (GridSearchCV best score: 0.8464285714285713)

DISCUSSION

- From the data visualization section, we can see that some features may have correlation with the mission outcome in several ways. For example, with heavy payloads the successful landing or positive landing rate are more for orbit types Polar, LEO and ISS. However, for GTO, we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.
- Therefore, each feature may have a certain impact on the final mission outcome. The exact ways of how each of these features impact the mission outcome are difficult to decipher. However, we can use some machine learning algorithms to learn the pattern of the past data and predict whether a mission will be successful or not based on the given features.

Conclusion

In this project, we have gone through the process of identifying the business problem, specifying the data required, extracting and preparing the data, performing the machine learning by utilizing k-means clustering and providing recommendation to the stakeholder

The predictive model produced by decision tree algorithm performed the best among the 4 machine learning algorithms employed.