IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Oliver Morris
11/10/2022

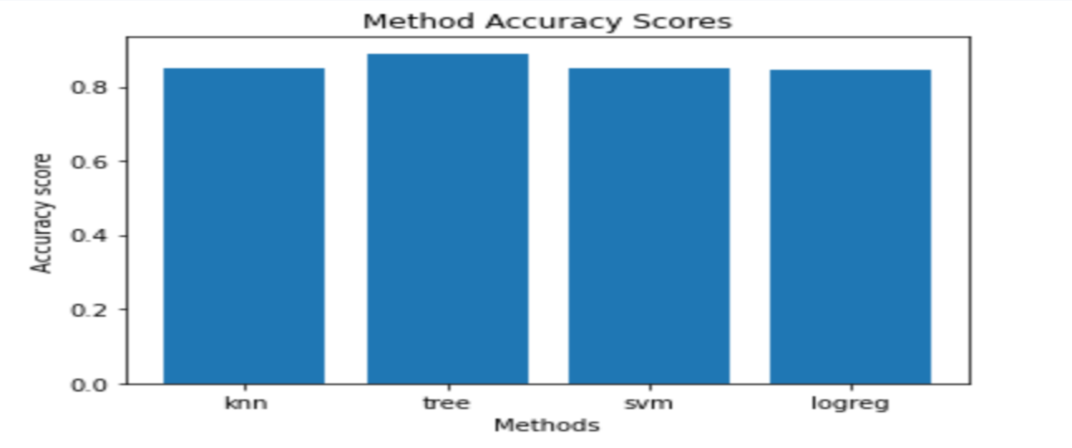
Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- Summary of methodologies:
  - Data was scraped from the SpaceX URL using Beautiful Soup
  - Data was then wrangled to fill nulls and organize data
  - EDA was undertaken to determine key relationships between data
  - Foilum was used to build a map of sites
  - A number of different prediction models were trialled to find the optimum predictor

- Accuracy score summarized below



Introduction

- In this capstone, we will predict if the Falcon 9 first stage will land successfully.
- SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.
- Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

Section 1

Methodology

Methodology

Executive Summary

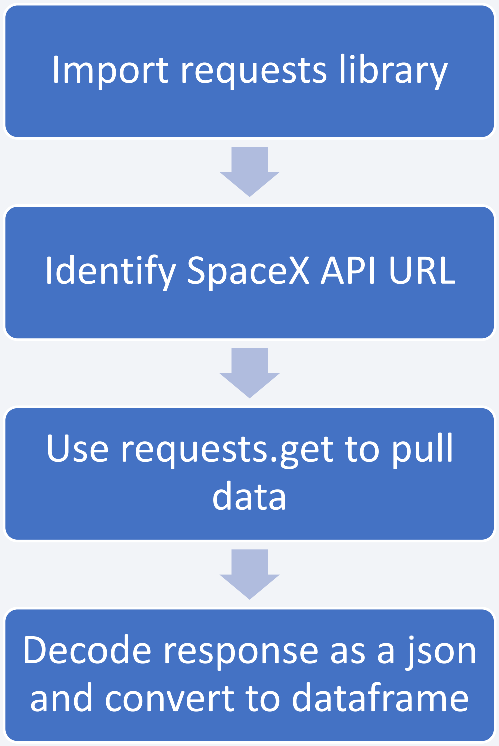
- Data collection methodology:
    - Scraped from the SpaceX URL using Beautiful Soup
- Perform data wrangling
    - Filling nulls and identifying key data points
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection – SpaceX API

- Data was collected from the SpaceX API URL as per flowchart opposite.

- GitHub URL of the completed SpaceX API calls notebook:
- https://github.com/morriol/IBM-DS-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

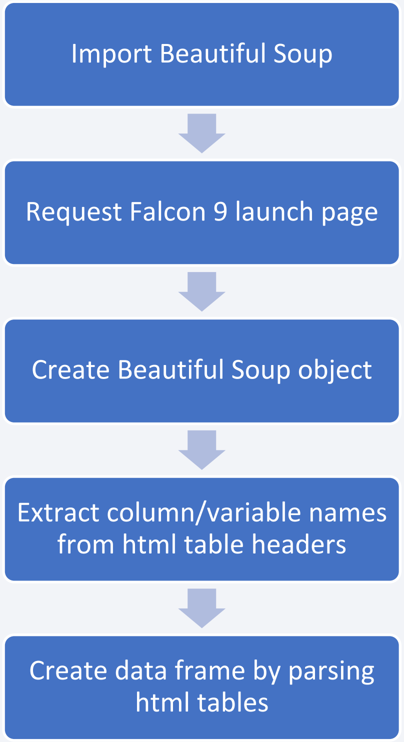
Flowchart of SpaceX API calls here



Data Collection - Scraping

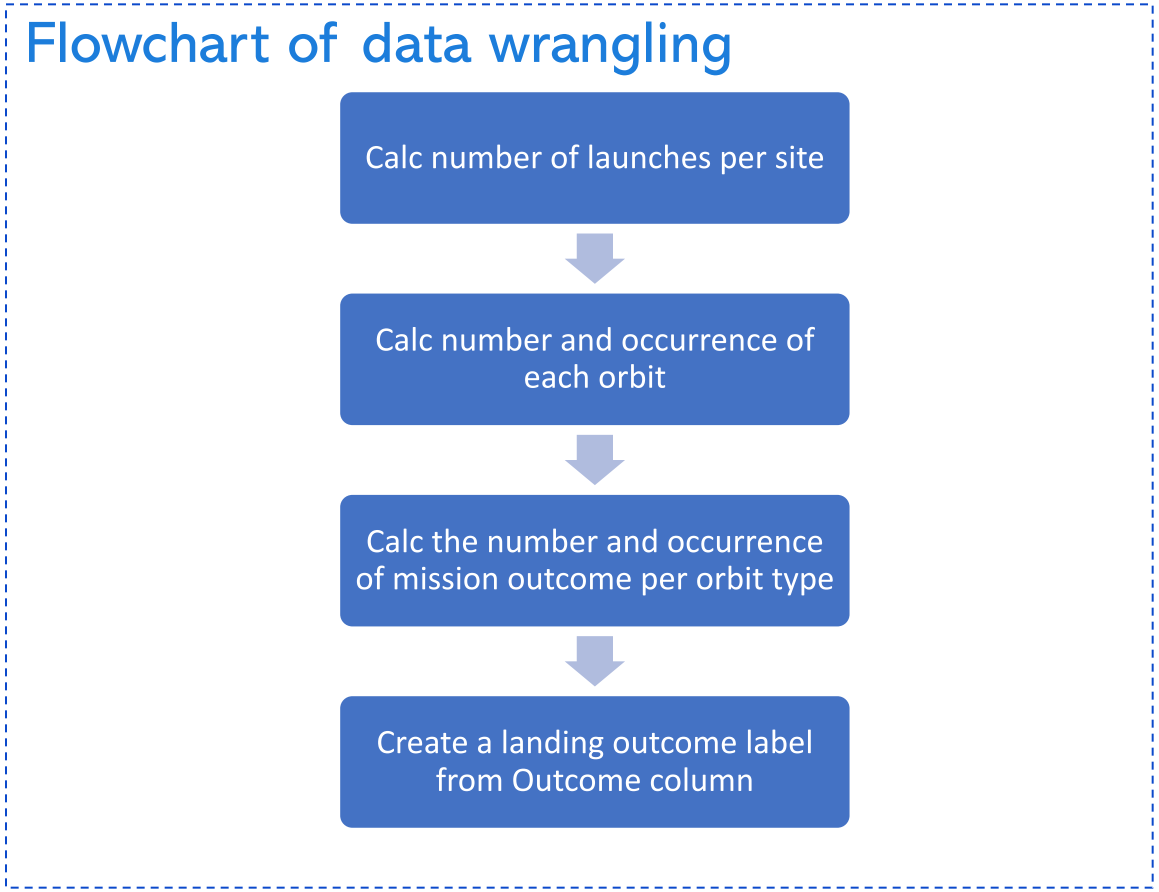
- Data was collected from the SpaceX API URL as per flowchart opposite.
- URL to git hub repo
- https://github.com/morriol/IBM-DS-Capstone/blob/main/jupyter-labs-webscraping.ipynb

Flowchart of web scraping



Data Collection - Wrangling

- Data was wrangled as per flow chart opposite.
- URL to git hub repo
- https://github.com/morriol/IBM-DS-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

- Summary of charts plotted and why used:
  - Scatter plot of flight number and payload mass with launch outcome to identify clusters of mass and flight number associated with successful launches
  - Scatter plot of flight number and site with launch outcome to identify clusters of sites and flight number associated with successful launches
  - Scatter plot of mass and site with launch outcome to identify clusters of sites and mass associated with successful launches
  - Bar chart to identify relationship between orbit and success rate
  - Line chart to examine success rate over time
- GitHub URL:
  - https://github.com/morriol/IBM-DS-Capstone/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- SQL queries performed:
  - Display unique launch sites
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1
  - List the date when the first successful landing outcome in ground pad was achieved
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - List the total number of successful and failure mission outcomes
  - List the names of the boosterversions which have carried the maximum payload mass
  - List the records which will display the month names, failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015.
  - Rank the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

- GitHub URL of completed EDA:
- https://github.com/morriol/IBM-DS-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Predictive Analysis (Classification)

- Model development process:
  - Created models using logistic regression, SVM, KNN and Classification Tree to predict classification (success/failure) of future launches
  - Parameters were optimized using GridSearchCV
  - Plotted a confusion matrix to identify types of errors being made by the model

- GitHub URL of completed predictive analysis lab
- https://github.com/morriol/IBM-DS-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

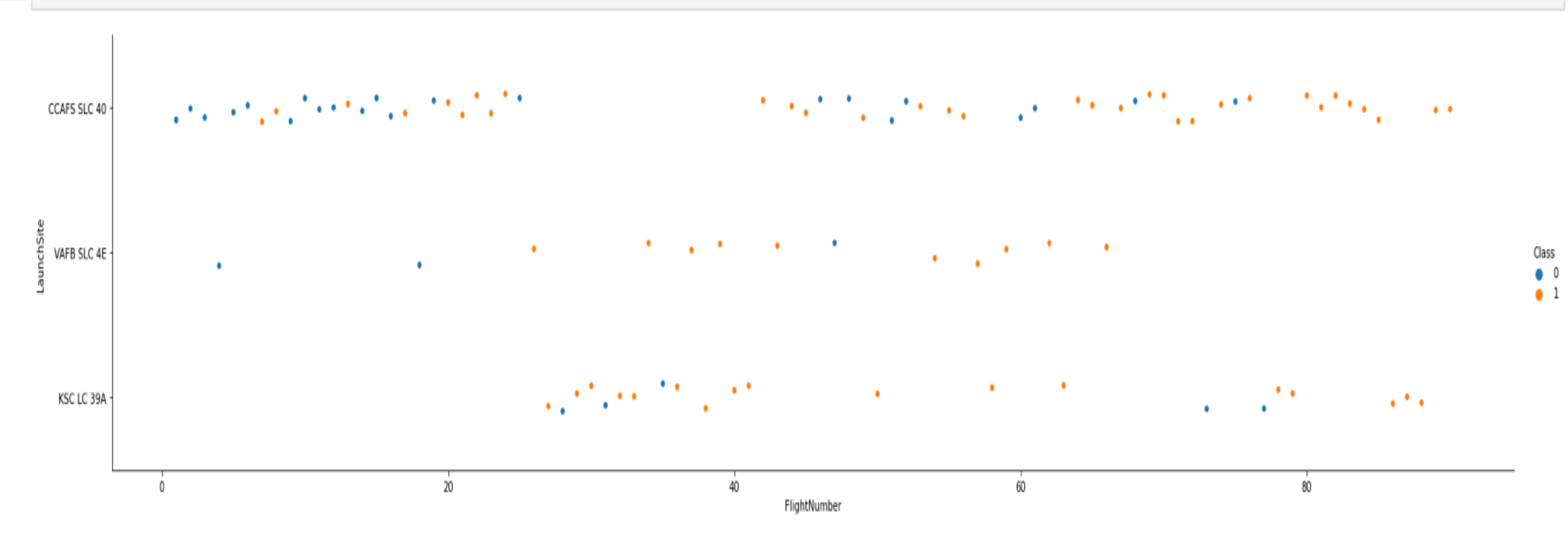
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

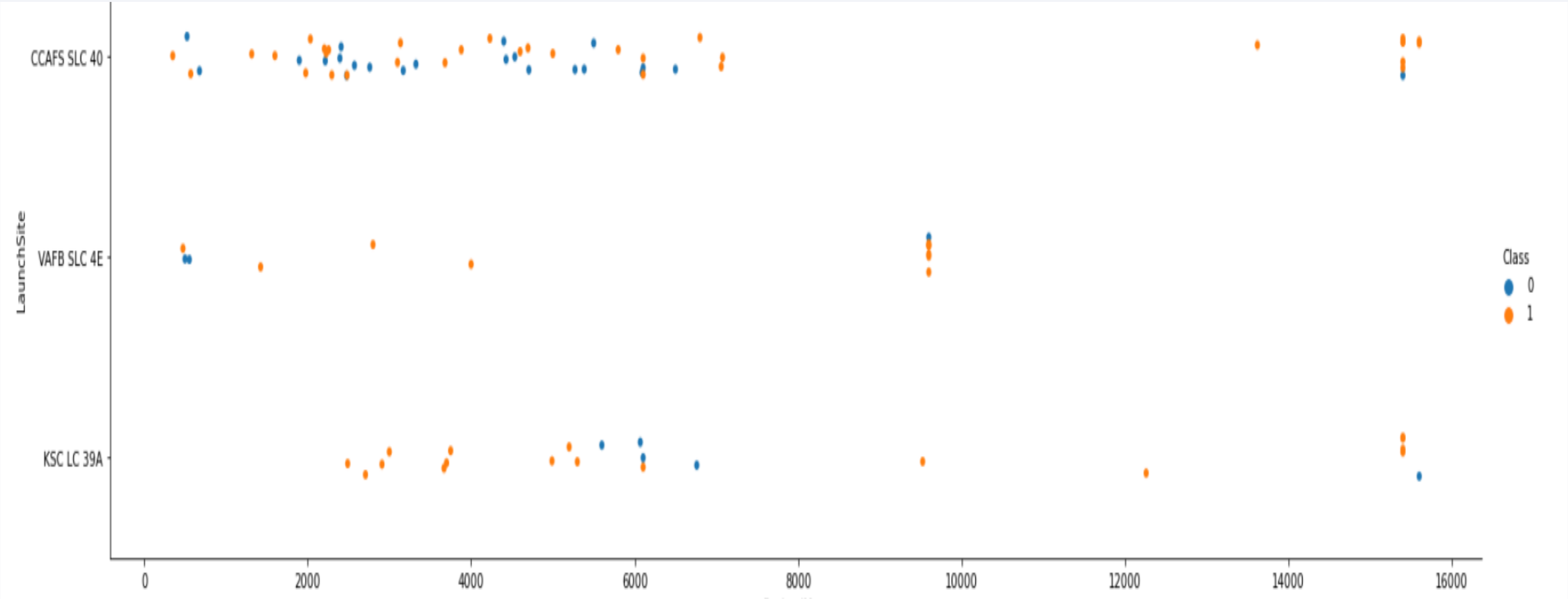
Section 2

Insights drawn from EDA

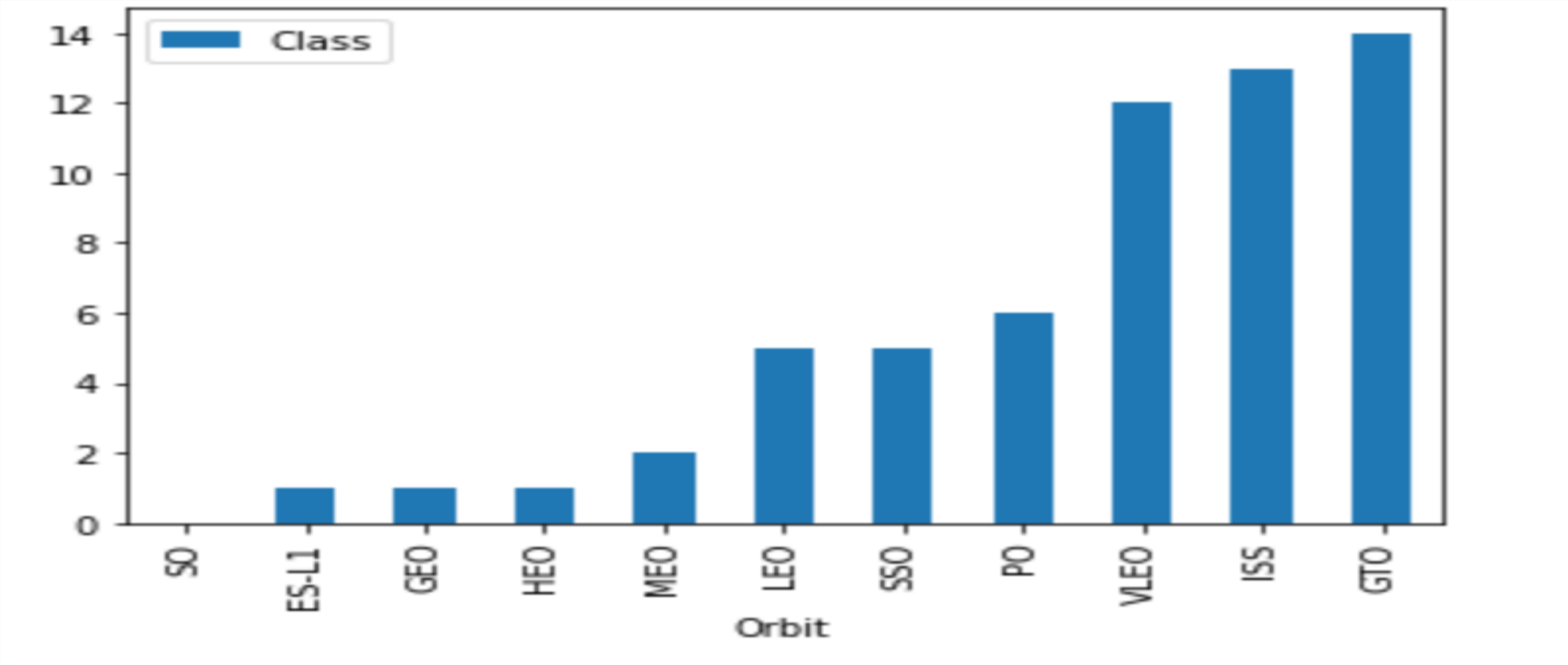
Flight Number vs. Launch Site



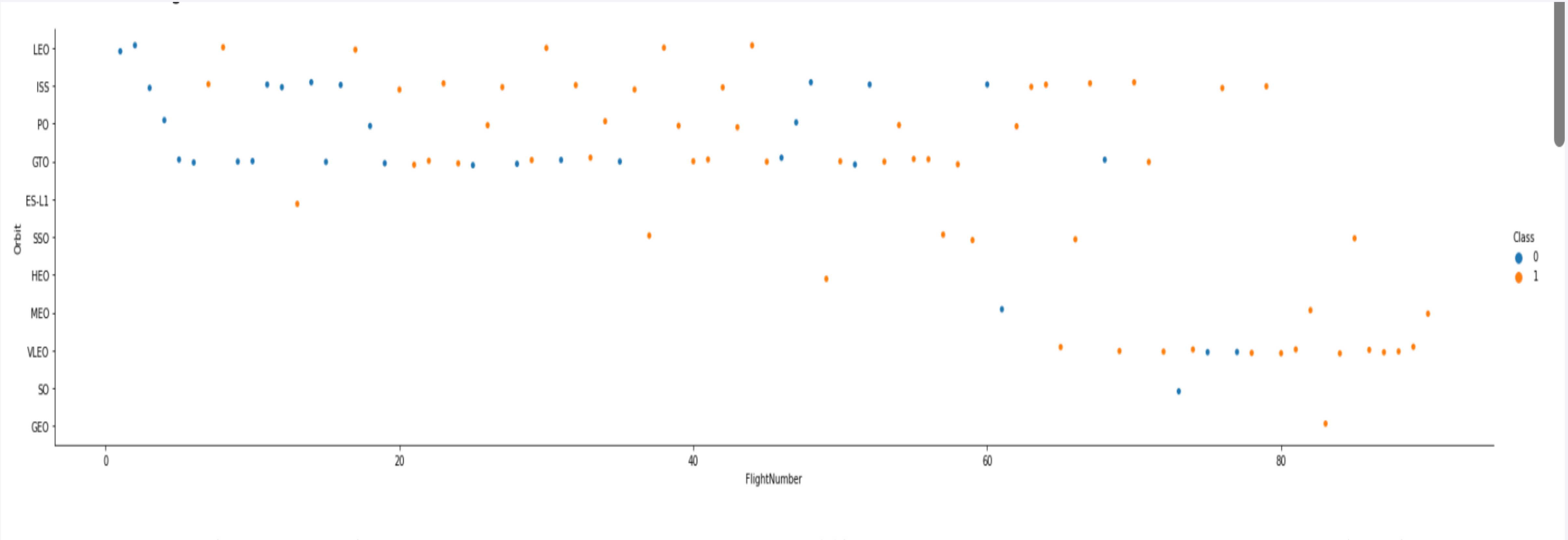
Payload vs. Launch Site



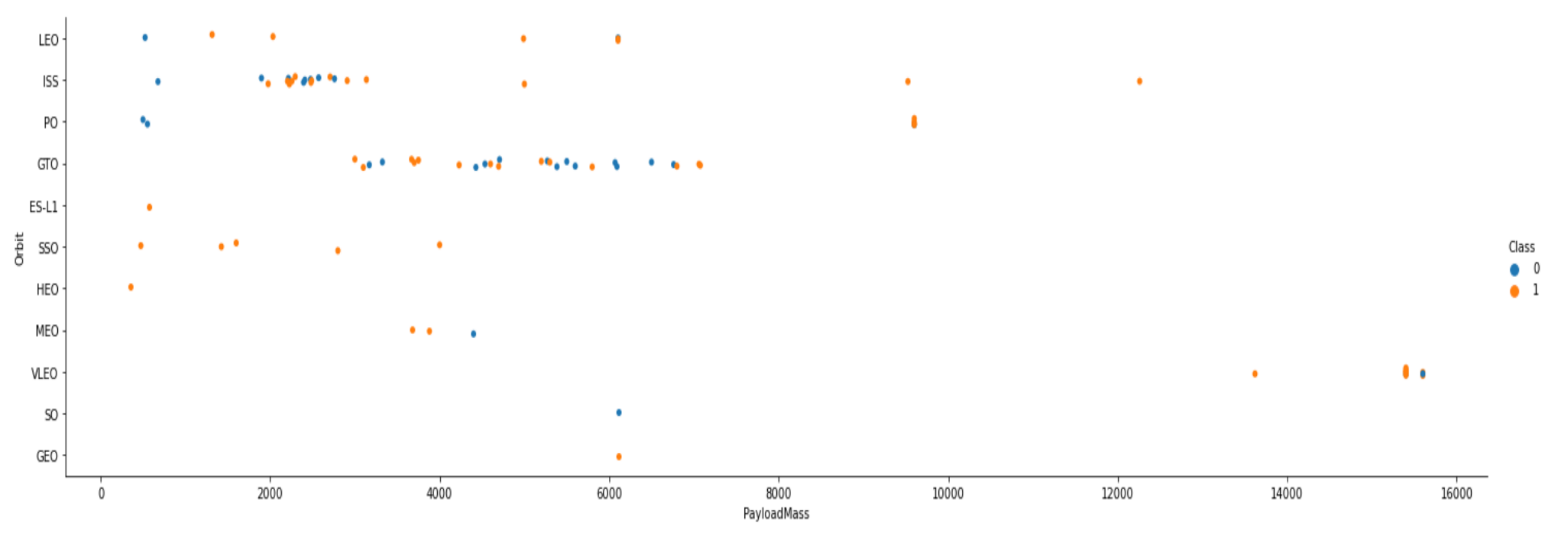
Success Rate vs. Orbit Type



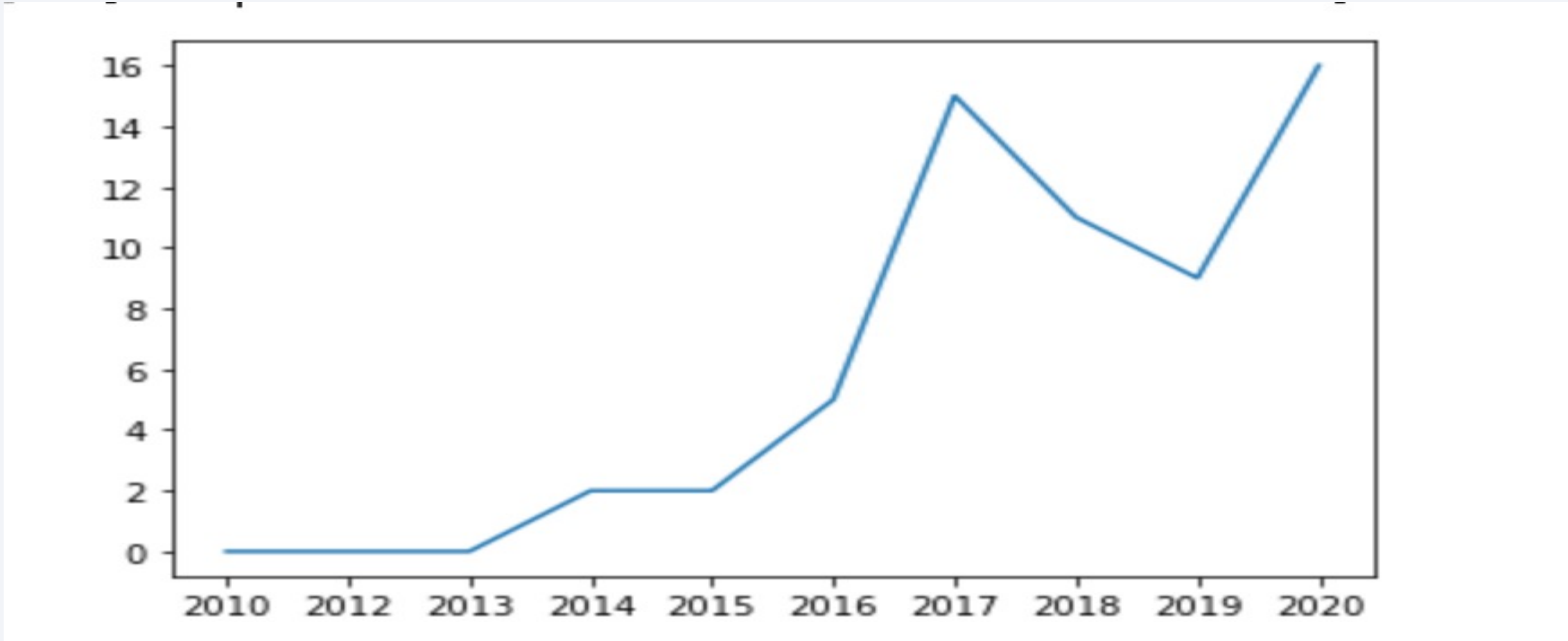
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

Display the names of the unique launch sites in the space mission

[15]:
```
%sql Select LAUNCH_SITE From SPACEXTBL Group By LAUNCH_SITE
```

```
 * sqlite:///my_data1.db
Done.
```

[15]:

Launch_Site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Launch Site Names Begin with 'CCA'

[21]:
```
%sql Select * From Spacextbl where Launch_Site Like "CCA%" Limit 5
```

* sqlite:///my_data1.db
Done.

[21]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

Display the total payload mass carried by boosters launched by NASA (CRS)

[27]:
```
%sql SELECT sum(Payload_Mass__KG_) FROM SPACEXTBL WHERE Customer = "NASA (CRS)" GROUP BY Customer
```

```
 * sqlite:///my_data1.db
Done.
```

[27]:

sum(Payload_Mass__KG_)
45596

Average Payload Mass by F9 v1.1

Display average payload mass carried by booster version F9 v1.1

```
[28]: %sql SELECT avg(Payload_Mass__KG_) FROM SPACEXTBL WHERE booster_version = "F9 v1.1"
```

```
 * sqlite:///my_data1.db
Done.
```

[28]:

avg(Payload_Mass__KG_)
2928.4

First Successful Ground Landing Date

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

[58]:
```
%sql SELECT min(Date) FROM SPACEXTBL WHERE "LANDING _OUTCOME" LIKE "Success%"
```

```
 * sqlite:///my_data1.db
Done.
```

[58]:

min(Date)
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

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

[65]:
```
%sql SELECT BOOSTER_VERSION,
Payload_mass__kg_ FROM SPACEXTBL WHERE "LANDING _OUTCOME" = "Success (drone ship)" AND
Payload_mass__kg_ < 6000 AND Payload_mass__kg_ >4000
```

```
 * sqlite:///my_data1.db
Done.
```

[65]:

Booster_Version	PAYLOAD_MASS__KG_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

[69]:
```
%sql SELECT count(*) FROM SPACEXTBL WHERE mission_outcome LIKE "Success%"
```

```
 * sqlite:///my_data1.db
Done.
```

[69]:

count(*)
100

[68]:
```
%sql SELECT count(*) FROM SPACEXTBL WHERE mission_outcome NOT LIKE "Success%"
```

```
 * sqlite:///my_data1.db
Done.
```

[68]:

count(*)
1

[70]:
```
%sql SELECT count(*) FROM SPACEXTBL
```

```
 * sqlite:///my_data1.db
Done.
```

[70]:

count(*)
101

Boosters Carried Maximum Payload

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

[74]:
```
%sql SELECT booster_version,
Payload_Mass__KG_ FROM SPACEXTBL WHERE Payload_Mass__KG_ = (SELECT max(Payload_Mass__KG_) FROM SPACEXTBL)
```

```
 * sqlite:///my_data1.db
Done.
```

[74]:

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

[92]:
```
%sql SELECT substr(Date, 4, 2), "Landing _Outcome", booster_version,
launch_site FROM SPACEXTBL WHERE substr(Date, 7, 4)="2015" ORDER BY "Landing _Outcome"
```

```
 * sqlite:///my_data1.db
Done.
```

[92]:

substr(Date, 4, 2)	Landing _Outcome	Booster_Version	Launch_Site
02	Controlled (ocean)	F9 v1.1 B1013	CCAFS LC-40
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
03	No attempt	F9 v1.1 B1014	CCAFS LC-40
04	No attempt	F9 v1.1 B1016	CCAFS LC-40
06	Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40
12	Success (ground pad)	F9 FT B1019	CCAFS LC-40

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

[99]:
```
%sql SELECT Date, "Landing _Outcome", booster_version,
launch_site FROM SPACEXTBL WHERE Date BETWEEN "04-06-2010" AND "20-03-2017"
AND "Landing _Outcome" LIKE "Success" ORDER BY substr(Date,7,4) DESC,
substr(Date, 4, 2) DESC, substr(Date, 1, 2) DESC
```

```
 * sqlite:///my_data1.db
Done.
```

[99]:

Date	Landing _Outcome	Booster_Version	Launch_Site
06-12-2020	Success	F9 B5 B1058.4	KSC LC-39A
16-11-2020	Success	F9 B5B1061.1	KSC LC-39A
05-11-2020	Success	F9 B5B1062.1	CCAFS SLC-40
18-10-2020	Success	F9 B5 B1051.6	KSC LC-39A
06-10-2020	Success	F9 B5 B1058.3	KSC LC-39A
18-08-2020	Success	F9 B5 B1049.6	CCAFS SLC-40
07-08-2020	Success	F9 B5 B1051.5	KSC LC-39A
13-06-2020	Success	F9 B5 B1059.3	CCAFS SLC-40
04-06-2020	Success	F9 B5 B1049.5	CCAFS SLC-40
07-03-2020	Success	F9 B5 B1059.2	CCAFS SLC-40
07-01-2020	Success	F9 B5 B1049.4	CCAFS SLC-40
17-12-2019	Success	F9 B5 B1056.3	CCAFS SLC-40
05-12-2019	Success	F9 B5B1059.1	CCAFS SLC-40
11-11-2019	Success	F9 B5 B1048.4	CCAFS SLC-40
12-06-2019	Success	F9 B5 B1051.2	VAFB SLC-4E
11-01-2019	Success	F9 B5 B1049.2	VAFB SLC-4E
15-11-2018	Success	F9 B5 B1047.2	KSC LC-39A
08-10-2018	Success	F9 B5 B1048.2	VAFB SLC-4E
10-09-2018	Success	F9 B5B1049.1	CCAFS SLC-40
07-08-2018	Success	F9 B5 B1046.2	CCAFS SLC-40

Section 3

Launch Sites Proximities Analysis

Folium Map of launch sites

Section 4

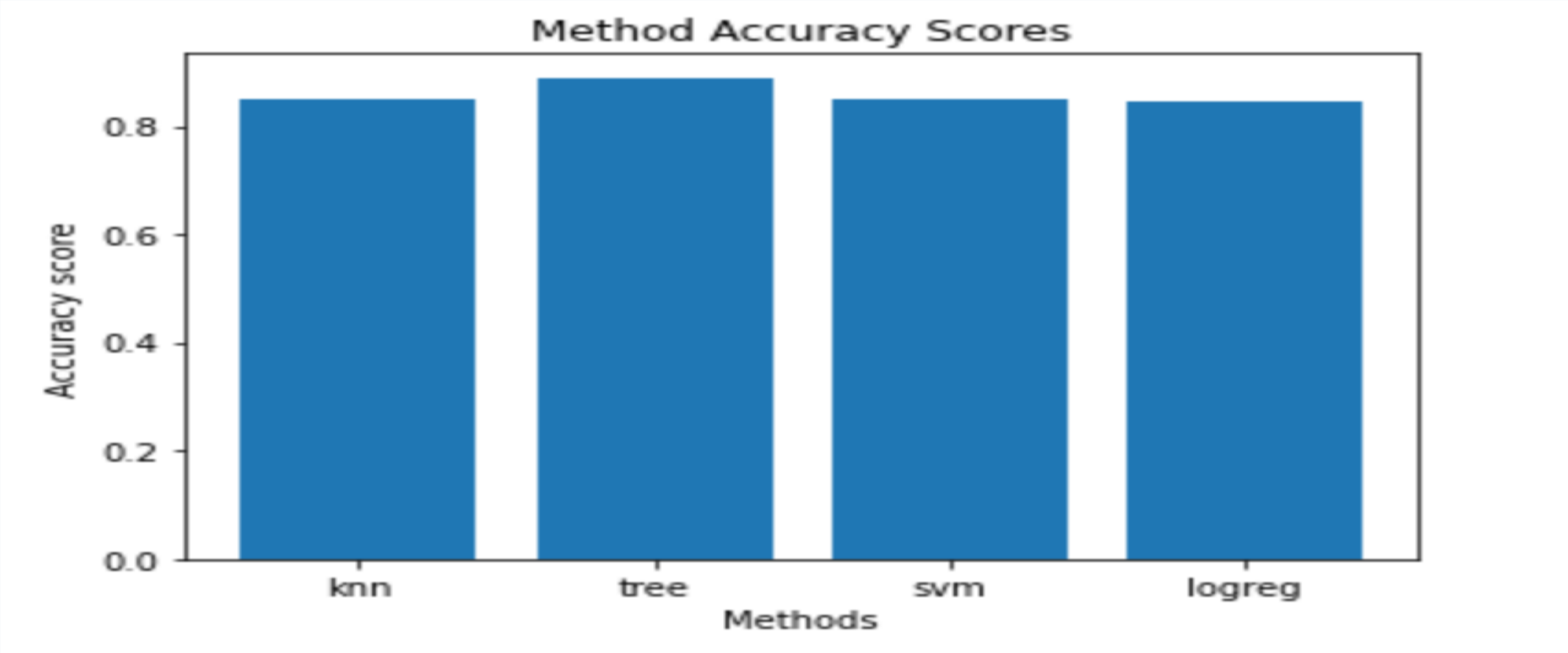
Build a Dashboard with Plotly Dash

Unable to complete Dashboard Section as no access to IBM cloud

Section 5

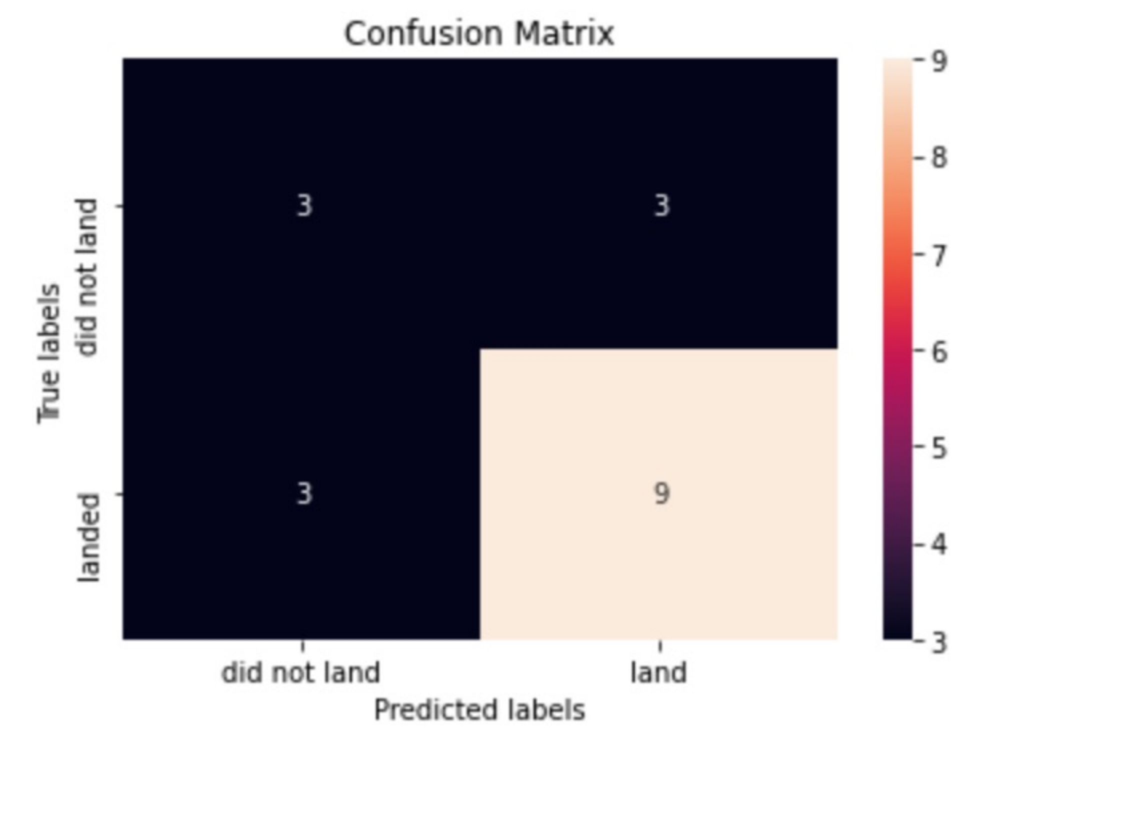
Predictive Analysis (Classification)

Classification Accuracy



Classification tree has the best accuracy score

Confusion Matrix



- This is the confusion matrix for the classification tree model.
- It shows the model correctly predicted 9/12 (true positive) landings and 3/6 (true negative) failures.

Conclusions

- Point 1: Using data wrangling and feature engineering we we able to build models with a high degree of accuracy – all with a score +80%
- Point 2: Whilst classification tree had the best accuracy other models log_reg, svm and knn all had better precision and recall
- Point 3: Therefore knn or svm may be better choices as models to use for prediction

Thank you!