IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ongart P.
10th December 2021

Outline

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

Executive Summary

Since we would like to know the cost of a rocket launch by determine if the first stage will be landed. So, we collect data both from SpaceX and Wikipedia to get launch historical. And with this information, use it to train several AI model as Logistic Regression, Support Vector Machine, Decision Tree, and K-Nearest Neighbors to find out which model is better one.

And with the result, all models almost similar but Decision Tree accuracy rate is higher than other with less false positive. Then we may use Decision Tree model as solution to predict opportunity of the first stage to be landed.

Introduction

- 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.
- So, we would like to know which AI model is suitable to determine the first stage landed

Section 1

Methodology

Methodology

Executive Summary

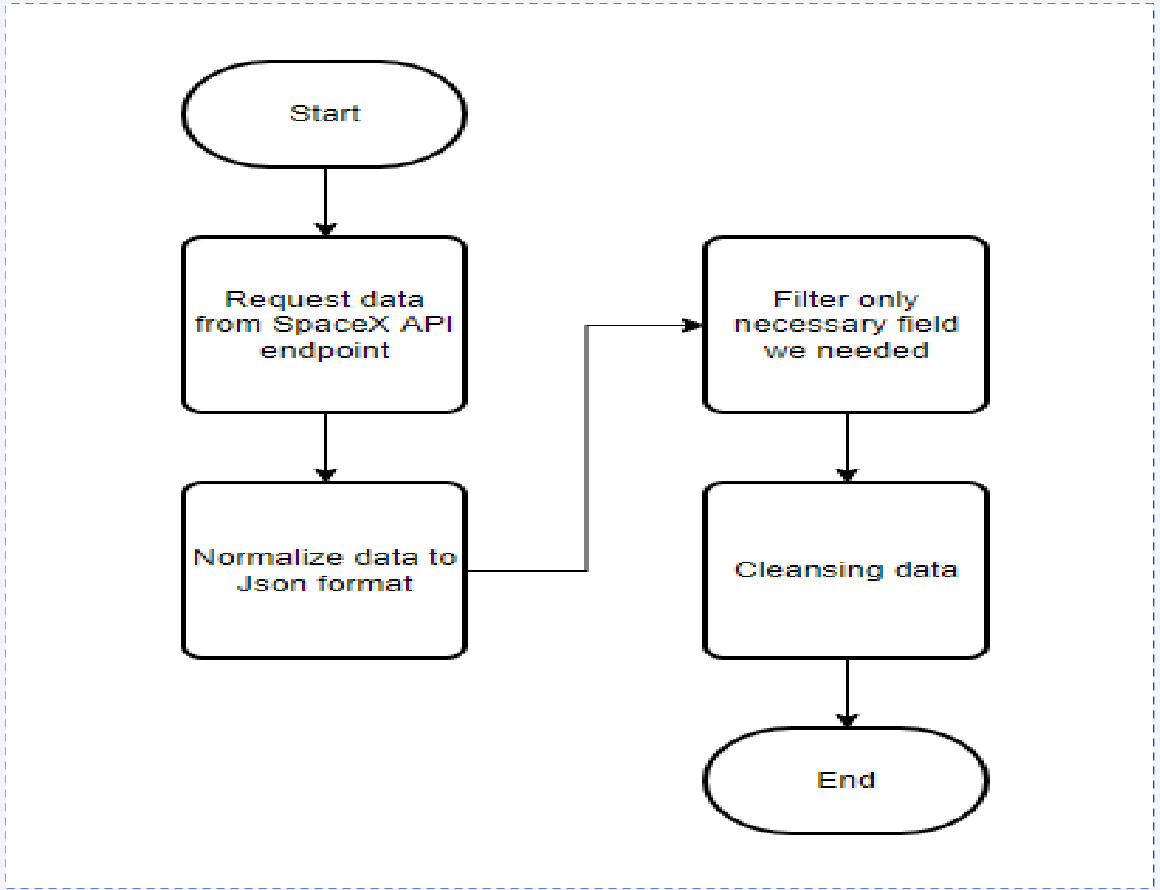
- Data collection methodology:
    - SpaceX API
    - Web scraping from Wikipedia
- Perform data wrangling
    - For data from API, we dealing with missing values of “PayloadMass” with mean value
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - With data, we standardize data by transform it and use it to evaluate each AI Model. And each AI model, we use GridSearchCV to find the best parameters

Data Collection

- We have 2 data sources to be collected. The first one is SpaceX API that we can use “requests” Python library to retrieve. And the second one is Wikipedia, we use web crawler that use “requests” and “BeautifulSoap” to extract data.
- After get data from both datasource, we need to consider data quality and do cleansing job if necessary

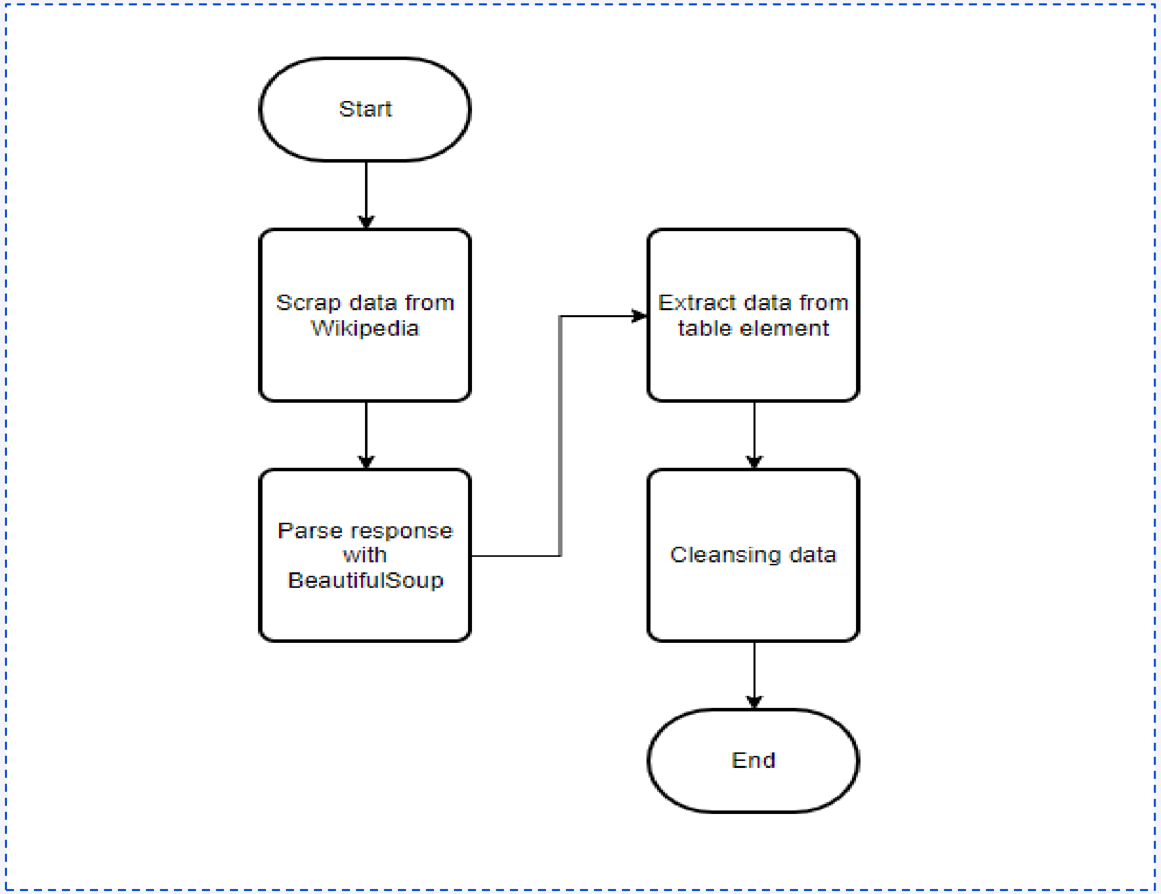
Data Collection – SpaceX API

- GitHub URL of the completed SpaceX API notebook


Start
Request data from SpaceX API endpoint
Normalize data to Json format
Filter only necessary field we needed
Cleansing data
End

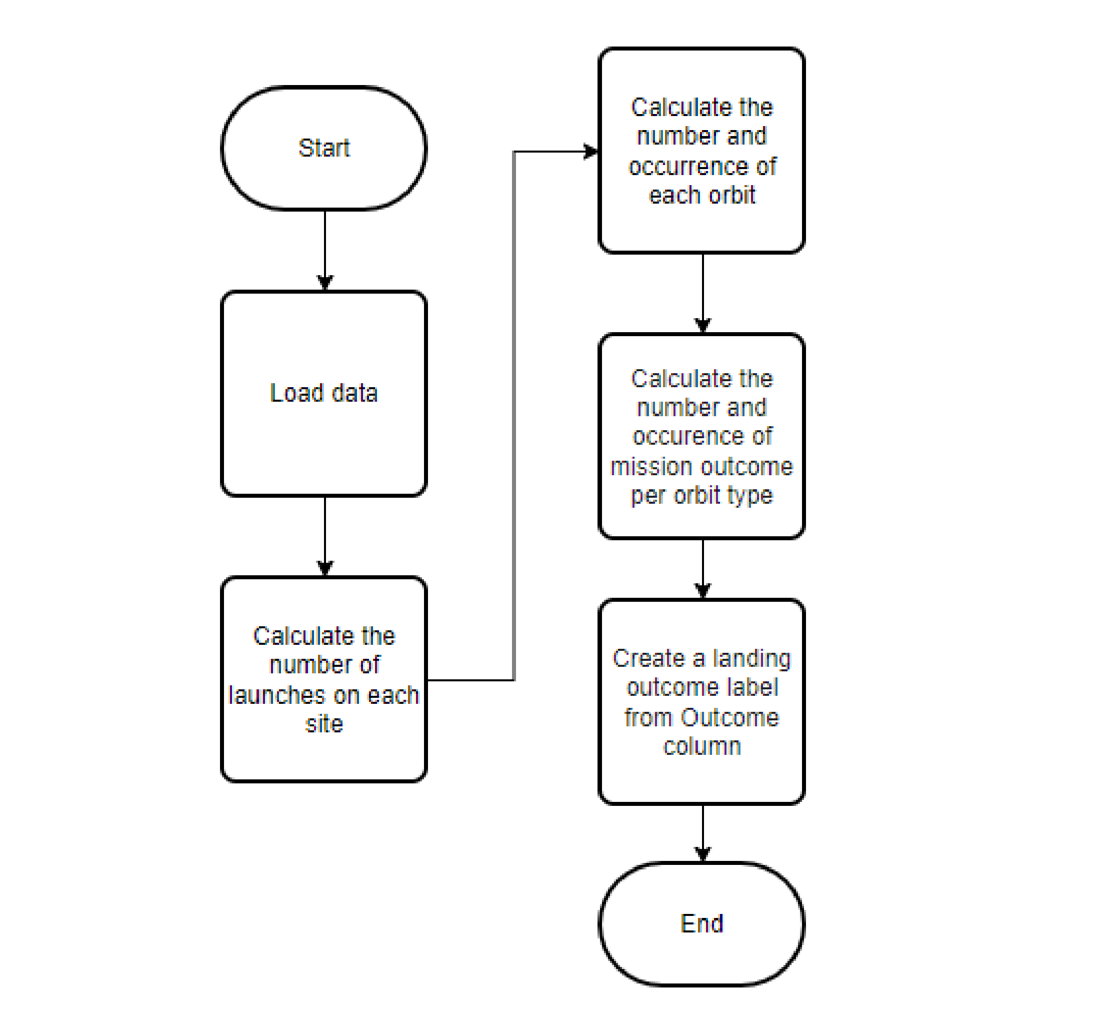
Data Collection – Scraping

- [GitHub URL of the completed SpaceX API calls notebook]


Start
Scrap data from Wikipedia
Parse response with BeautifulSoup
Extract data from table element
Cleansing data
End

Data Wrangling

- GitHub URL of your completed data wrangling related notebooks


Start
Load data
Calculate the number of launches on each site
Calculate the number and occurrence of each orbit
Calculate the number and occurence of mission outcome per orbit type
Create a landing outcome label from Outcome column
End

EDA with Data Visualization

- We use scatter chart to see relationship between considered fields such as Flight Number & Payload Mass, Flight Number & Launch Site, Payload & Launch Site , Flight Number & Orbit Type, Payload & Orbit Type. We will see behavior of data between each parameters
- And we use bar chart to see relationship between Success rate & Orbit type. This will help us to find which orbits have high success rate
- For line chart, we used it to see success rate trend for yearly basis
- GitHub URL of your completed EDA with data visualization notebook

EDA with SQL

- We use SQL to inquiry data to see information follow:
    - The names of the unique launch sites in the space mission
    - Records where launch sites begin with the string 'CCA’
    - The total payload mass carried by boosters launched by NASA (CRS)
    - Average payload mass carried by booster version F9 v1.1
    - The date when the first successful landing outcome in ground pad was achieved
    - The names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
    - The total number of successful and failure mission outcomes
    - The names of the booster_versions which have carried the maximum payload mass
    - The failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
    - Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20
- GitHub URL of your completed EDA with SQL notebook

Build an Interactive Map with Folium

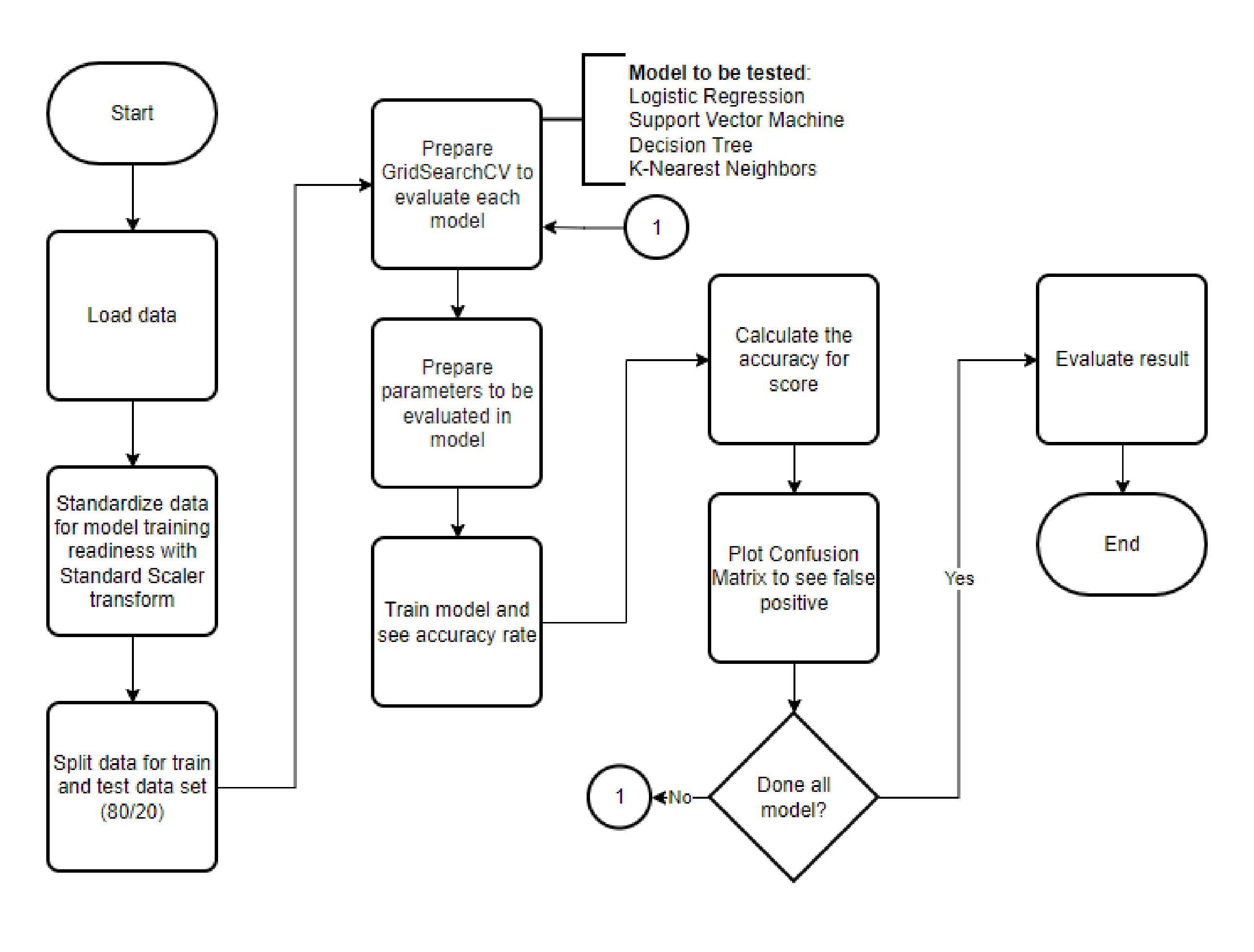
- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- We use Folium for analytic insight to see all Launch sites on the map with marker to show Launch sites, success rate. And with lines object to show distance between Launch sites and nearest beach.
- We use marker circle to show position with label, and lines to show distance between 2 locations
- GitHub URL of your completed interactive map with Folium map

Build a Dashboard with Plotly Dash

- We built a Plotly Dash application for end-users to perform interactive visual analytics on SpaceX launch data in real-time.
- We add Launch Site Drop-down for user to select site information they needed. We have Pie Chart to show success rate information. And scatter chart to see relationship between Launch Site, Payload Mass, and Booster version
- GitHub URL of your completed Plotly Dash lab

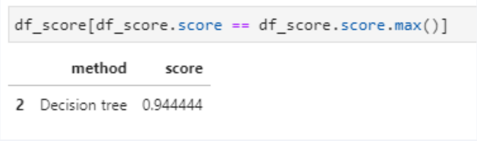
Predictive Analysis (Classification)

- For predictive analysis portion, we standardize data and then split data into training and test data. And then evaluate 4 AI modes Logistic Regression, Support Vector Machine, Decision Tree, and K-Nearest Neighbors with GridSearchCV. Then use Confusion Matrix to see result.
- GitHub URL of your completed predictive analysis lab


Start
Load data
Standardize data for model training readiness with Standard Scaler transform
Split data for train and test data set (80/20)
Prepare GridSearchCV to evaluate each model
Model to be tested:
Logistic Regression
Support Vector Machine
Decision Tree
K-Nearest Neighbors
1
Prepare parameters to be evaluated in model
Train model and see accuracy rate
Calculate the accuracy for score
Plot Confusion Matrix to see false positive
Done all model?
No
1
Yes
Evaluate result
End

Results

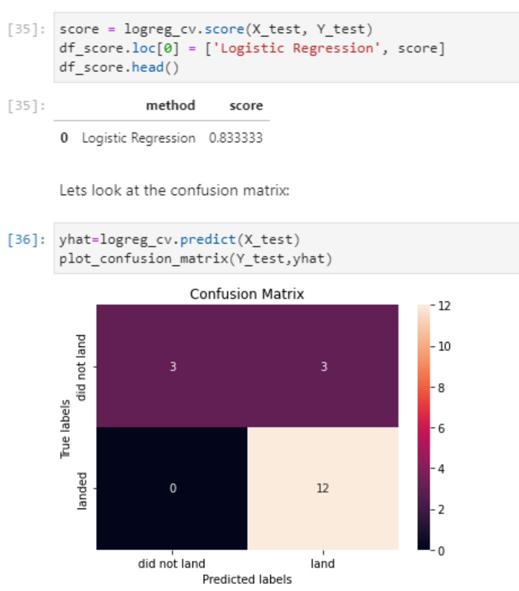
- With the result we found Decision Tree has most accuracy rate and score, but anther models have same score. Hence, we can use <u>**Decision Tree**</u> model for our prediction method



```
df_score[df_score.score == df_score.score.max()]
```

	method	score
2	Decision tree	0.944444

- Predictive analysis results



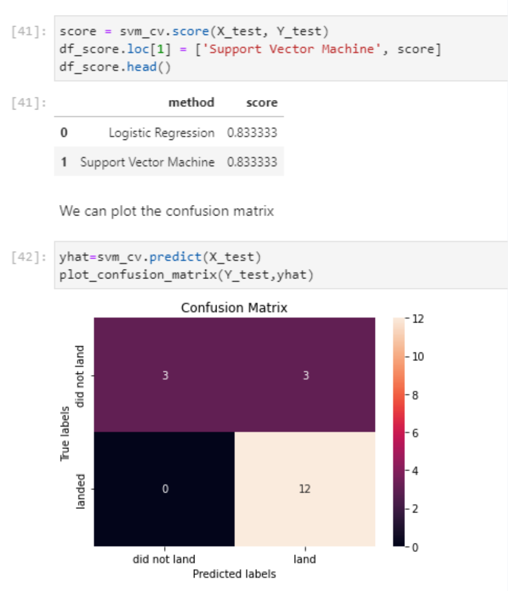
```
[35]: score = logreg_cv.score(X_test, Y_test)
      df_score.loc[0] = ['Logistic Regression', score]
      df_score.head()
```

[35]:

	method	score
0	Logistic Regression	0.833333

Lets look at the confusion matrix:

```
[36]: yhat=logreg_cv.predict(X_test)
      plot_confusion_matrix(Y_test,yhat)
```



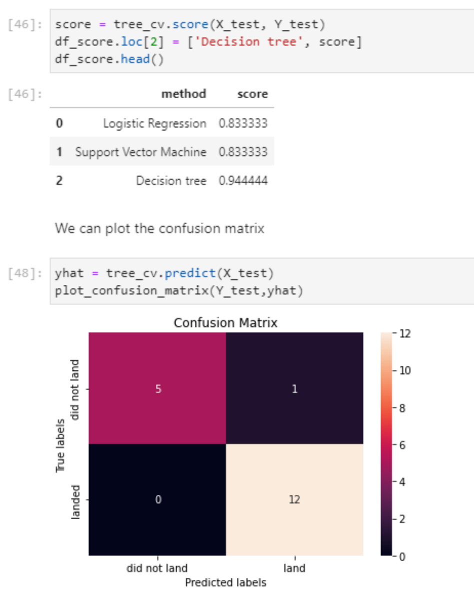
```
[41]: score = svm_cv.score(X_test, Y_test)
      df_score.loc[1] = ['Support Vector Machine', score]
      df_score.head()
```

[41]:

	method	score
0	Logistic Regression	0.833333
1	Support Vector Machine	0.833333

We can plot the confusion matrix

```
[42]: yhat=svm_cv.predict(X_test)
      plot_confusion_matrix(Y_test,yhat)
```



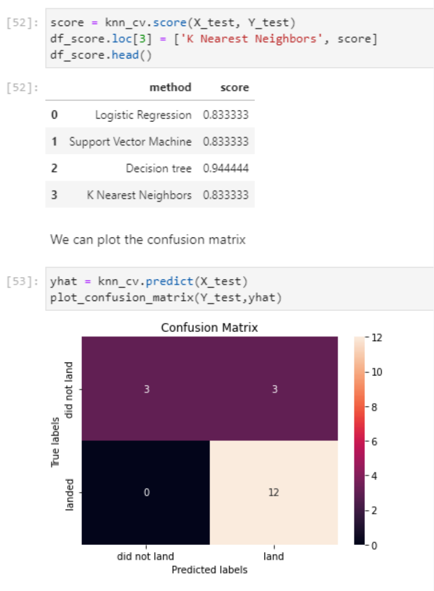
```
[46]: score = tree_cv.score(X_test, Y_test)
      df_score.loc[2] = ['Decision tree', score]
      df_score.head()
```

[46]:

	method	score
0	Logistic Regression	0.833333
1	Support Vector Machine	0.833333
2	Decision tree	0.944444

We can plot the confusion matrix

```
[48]: yhat = tree_cv.predict(X_test)
      plot_confusion_matrix(Y_test,yhat)
```



```
[52]: score = knn_cv.score(X_test, Y_test)
      df_score.loc[3] = ['K Nearest Neighbors', score]
      df_score.head()
```

[52]:

	method	score
0	Logistic Regression	0.833333
1	Support Vector Machine	0.833333
2	Decision tree	0.944444
3	K Nearest Neighbors	0.833333

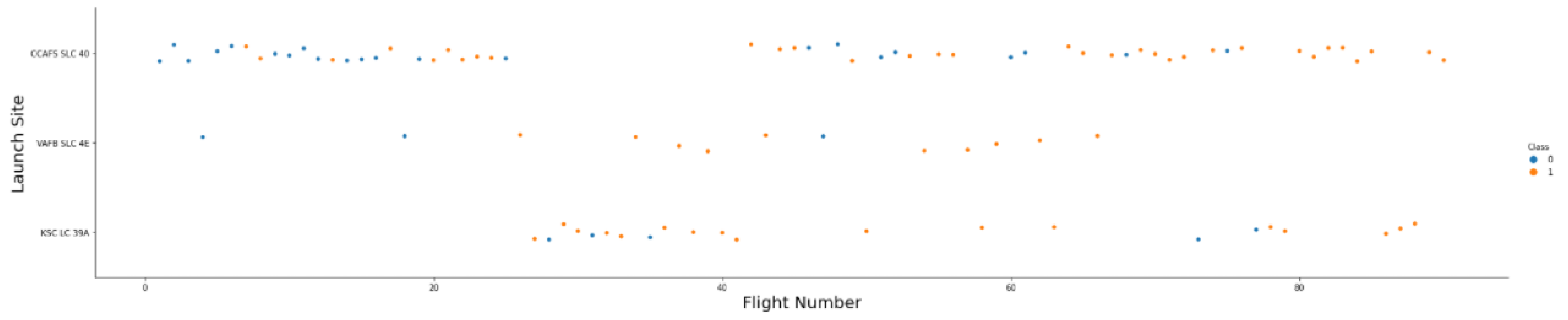
We can plot the confusion matrix

```
[53]: yhat = knn_cv.predict(X_test)
      plot_confusion_matrix(Y_test,yhat)
```

Section 2

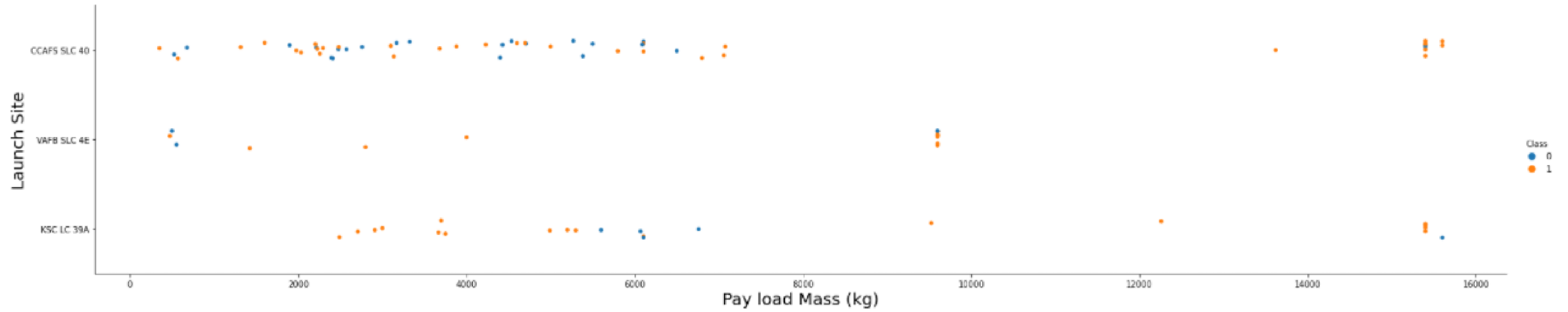
Insights drawn from EDA

Flight Number vs. Launch Site



- With scatter plot, we can see that most used Launch Site is “CCAFS SLC 40” follow by “VAFB SLC 4E” and “KSC LC 39A”

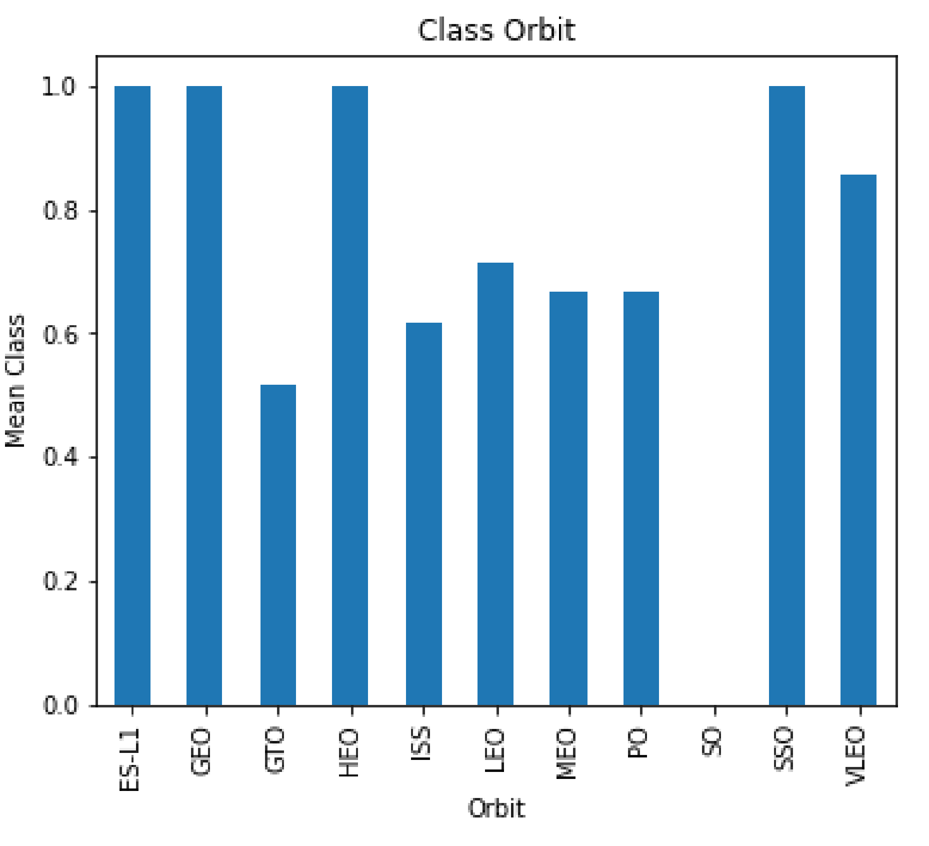
Payload vs. Launch Site



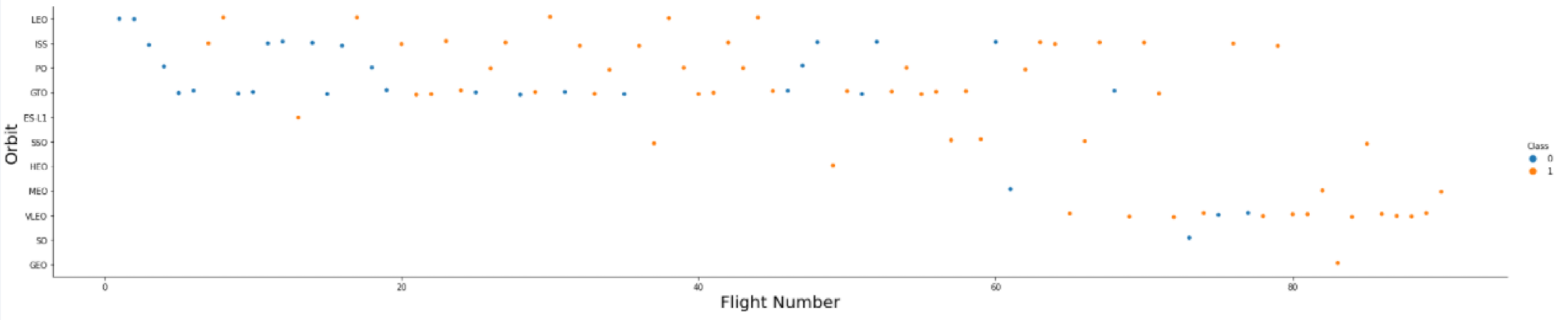
- If you observe Payload Vs. Launch Site scatter point chart you will find for the “VAFB-SLC 4E” launch site, there are no rockets launched for heavy payload mass(greater than 10000)

Success Rate vs. Orbit Type

- Follow the chart we will see that ES-L1, GEO, HEO, and SSO orbit has got most success rate

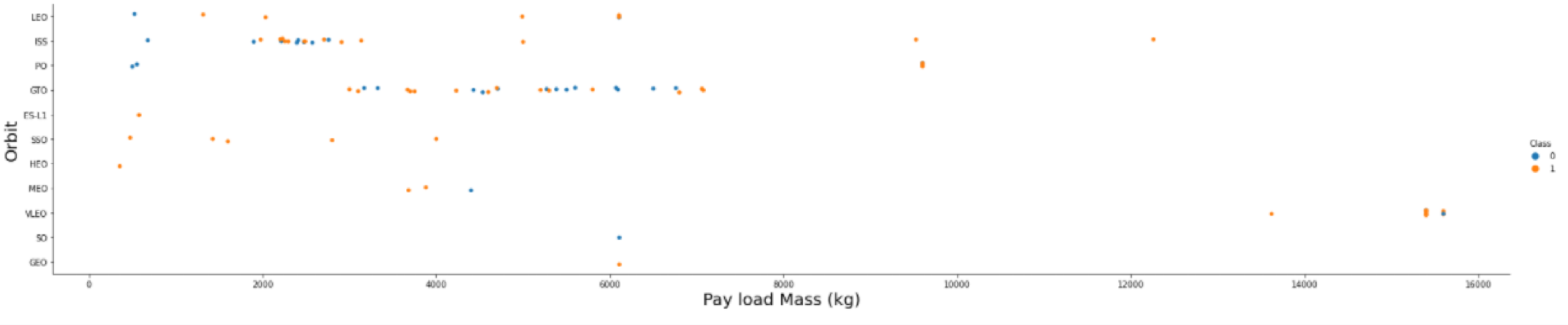


Flight Number vs. Orbit Type



- You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit

Payload vs. Orbit Type

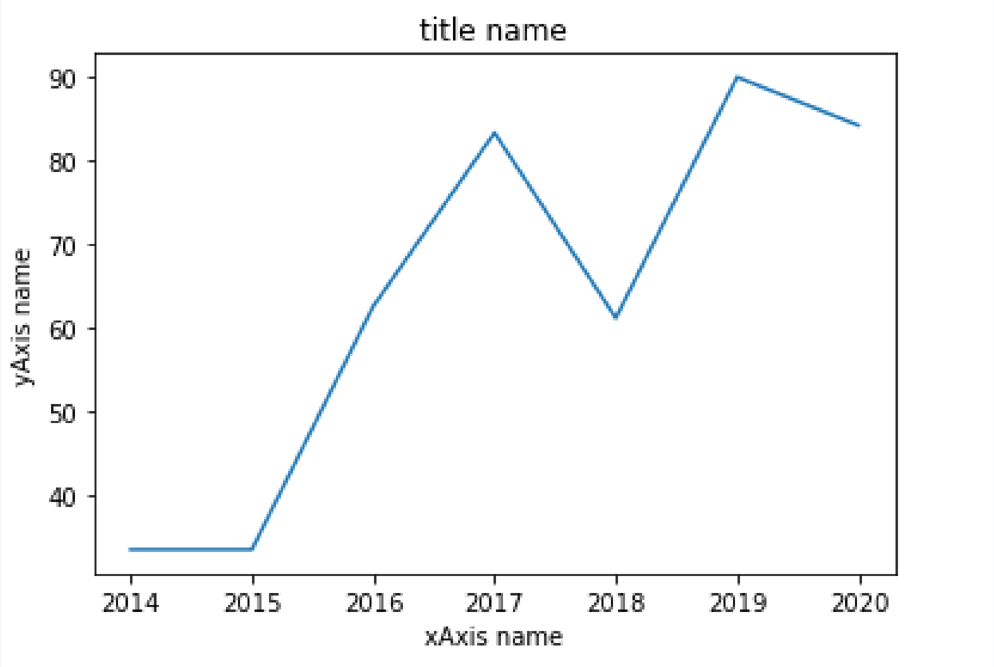


- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.

- However, for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here

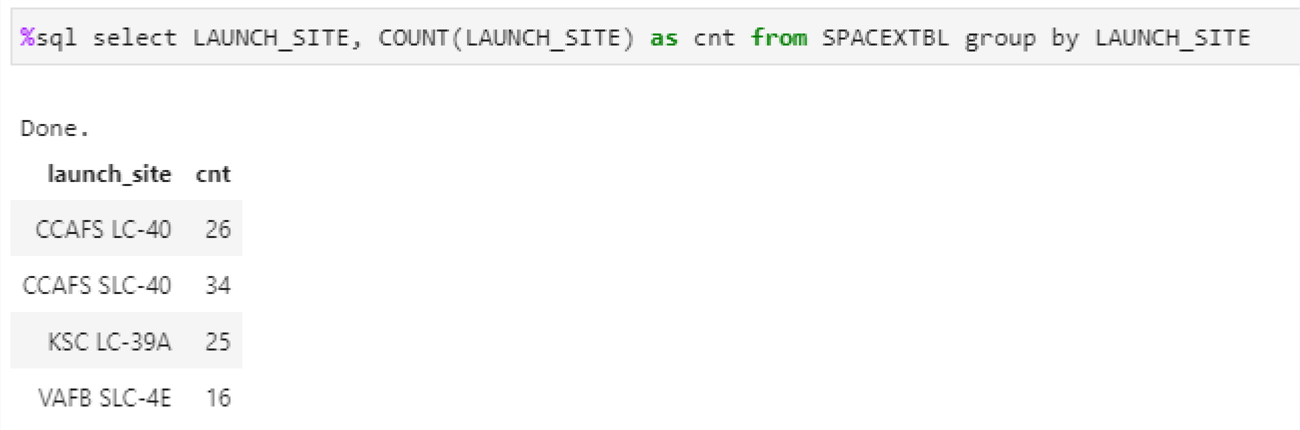
Launch Success Yearly Trend

- You can observe that the success rate since 2013 kept increasing till 2020



All Launch Site Names

- From the data, we have got 4 Launch Sites



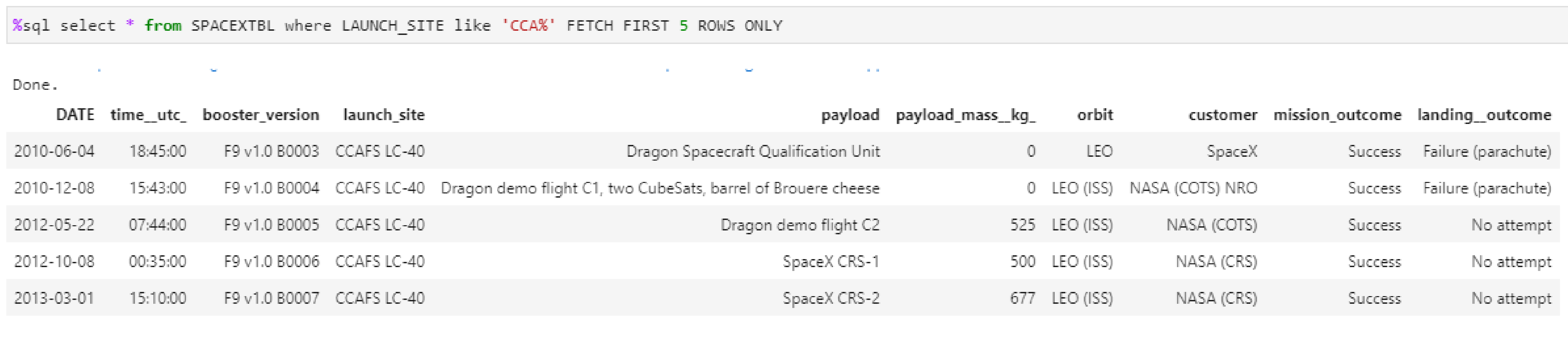
```
%sql select LAUNCH_SITE, COUNT(LAUNCH_SITE) as cnt from SPACEXTBL group by LAUNCH_SITE
```

Done.

launch_site	cnt
CCAFS LC-40	26
CCAFS SLC-40	34
KSC LC-39A	25
VAFB SLC-4E	16

Launch Site Names Begin with 'CCA'

- The top 5 record of launch sites begin with `CCA`



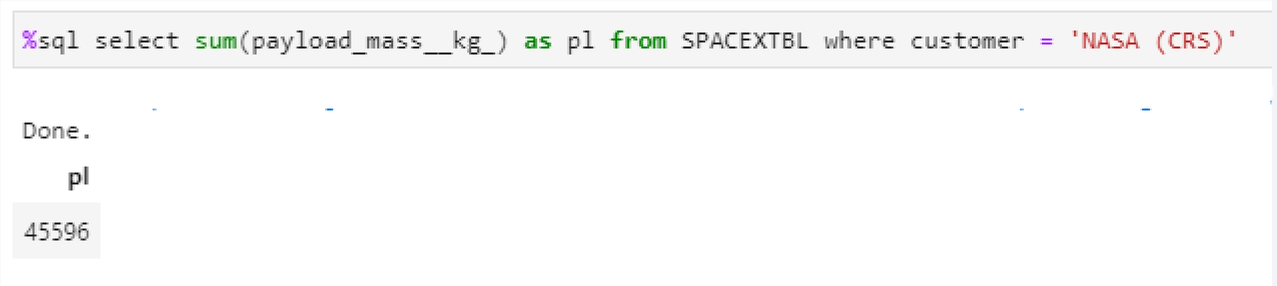
```
%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' FETCH FIRST 5 ROWS ONLY
```

```
Done.
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	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)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- The total payload carried by boosters from NASA is 45,596 kg.

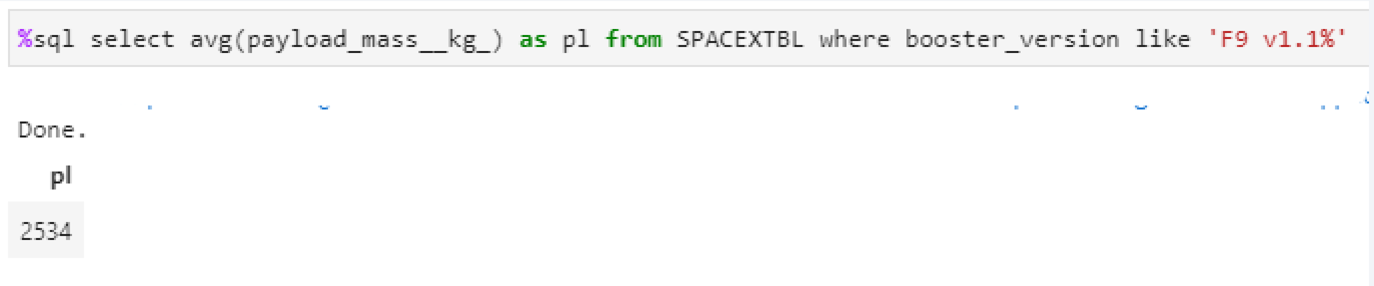


```
%sql select sum(payload_mass__kg_) as pl from SPACEXTBL where customer = 'NASA (CRS)'
```

```
Done.
   pl
45596
```

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1 is 2,534 kg.



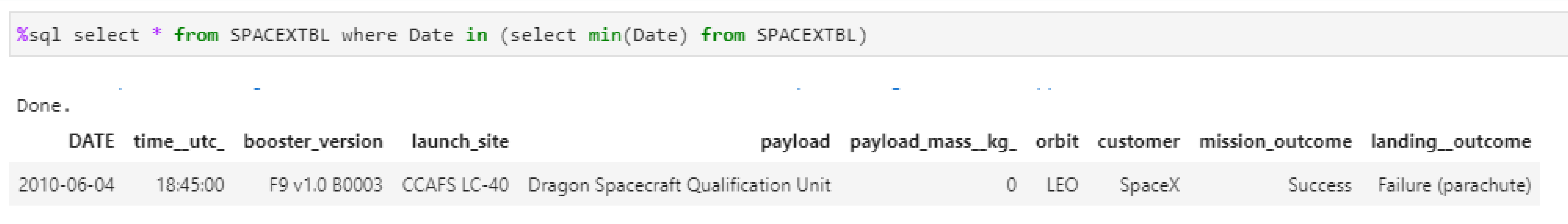
```
%sql select avg(payload_mass__kg_) as pl from SPACEXTBL where booster_version like 'F9 v1.1%'
```

```
Done.
```

pl
2534

First Successful Ground Landing Date

- The dates of the first successful landing outcome on ground pad is 4th June 2010



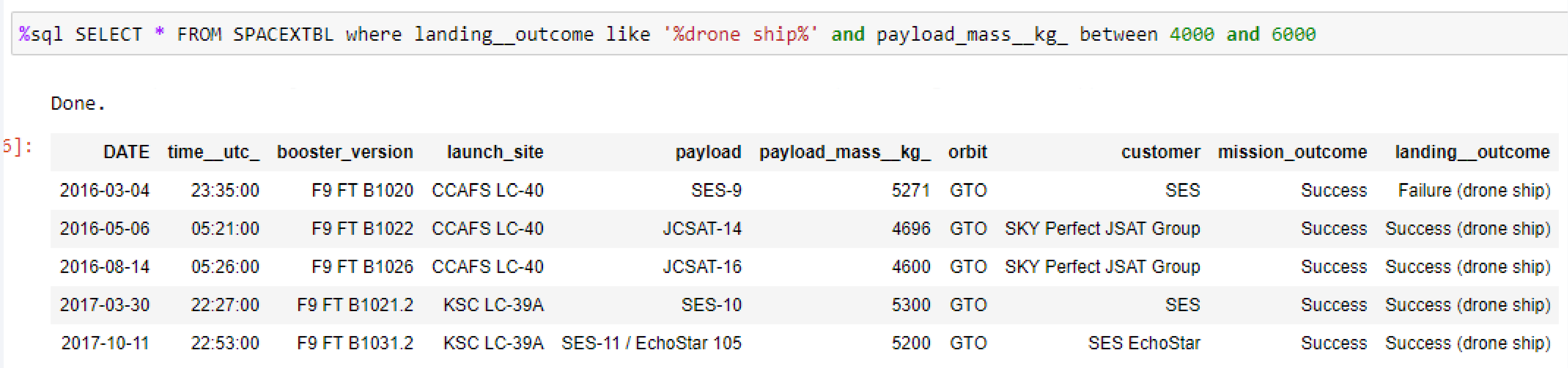
```
%sql select * from SPACEXTBL where Date in (select min(Date) from SPACEXTBL)
```

```
Done.
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)

Successful Drone Ship Landing with Payload between 4000 and 6000

- The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000



```
%sql SELECT * FROM SPACEXTBL where landing__outcome like '%drone ship%' and payload_mass__kg_ between 4000 and 6000
```

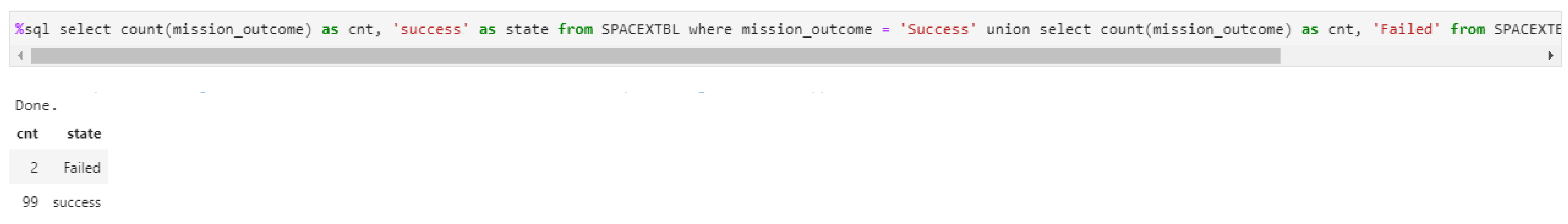
```
Done.
```

5]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2016-03-04	23:35:00	F9 FT B1020	CCAFS LC-40	SES-9	5271	GTO	SES	Success	Failure (drone ship)
2016-05-06	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-08-14	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-10-11	22:53:00	F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105	5200	GTO	SES EchoStar	Success	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- The total number of successful is 99 and failure mission outcomes is 2

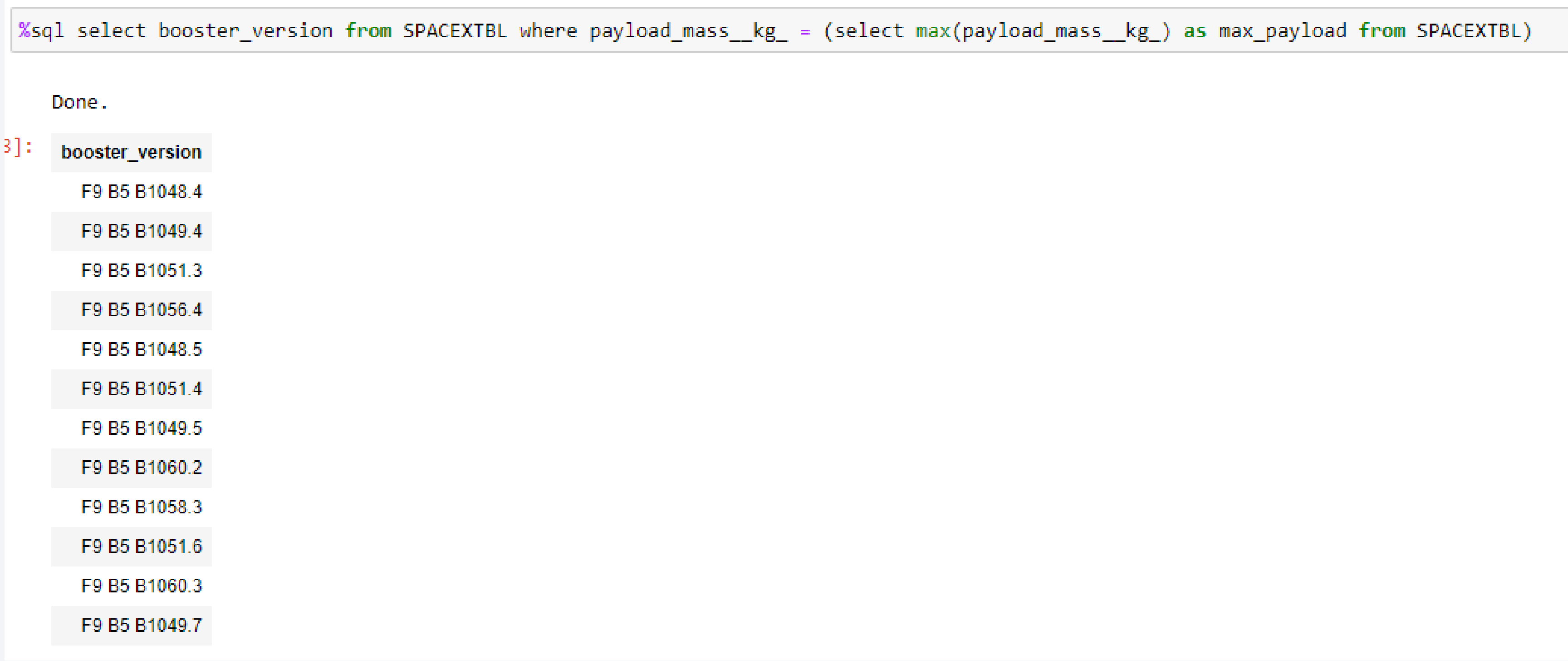


```
%sql select count(mission_outcome) as cnt, 'success' as state from SPACEXTBL where mission_outcome = 'Success' union select count(mission_outcome) as cnt, 'Failed' from SPACEXTE
```

```
Done.
cnt   state
  2   Failed
 99   success
```

Boosters Carried Maximum Payload

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



```
%sql select booster_version from SPACEXTBL where payload_mass__kg_ = (select max(payload_mass__kg_) as max_payload from SPACEXTBL)
```

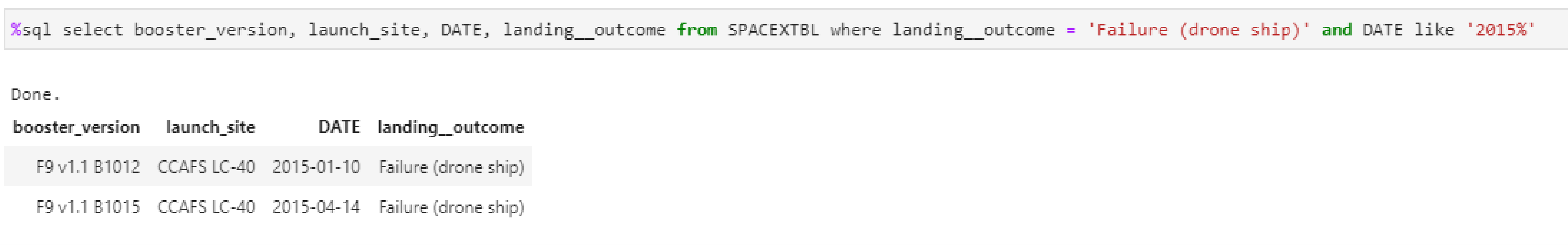
```
Done.
```

3]:

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

2015 Launch Records

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



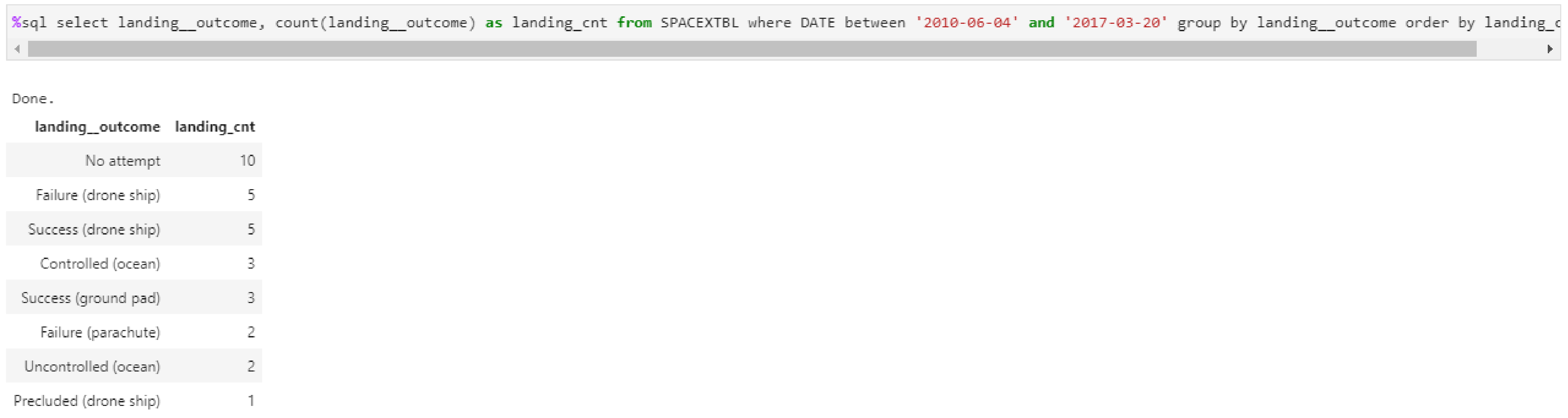
```
%sql select booster_version, launch_site, DATE, landing__outcome from SPACEXTBL where landing__outcome = 'Failure (drone ship)' and DATE like '2015%'
```

```
Done.
```

booster_version	launch_site	DATE	landing__outcome
F9 v1.1 B1012	CCAFS LC-40	2015-01-10	Failure (drone ship)
F9 v1.1 B1015	CCAFS LC-40	2015-04-14	Failure (drone ship)

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

- Rank the count of landing outcomes between the date 2010-06-04 and 2017-03-20.



```
%sql select landing__outcome, count(landing__outcome) as landing_cnt from SPACEXTBL where DATE between '2010-06-04' and '2017-03-20' group by landing__outcome order by landing_c
```

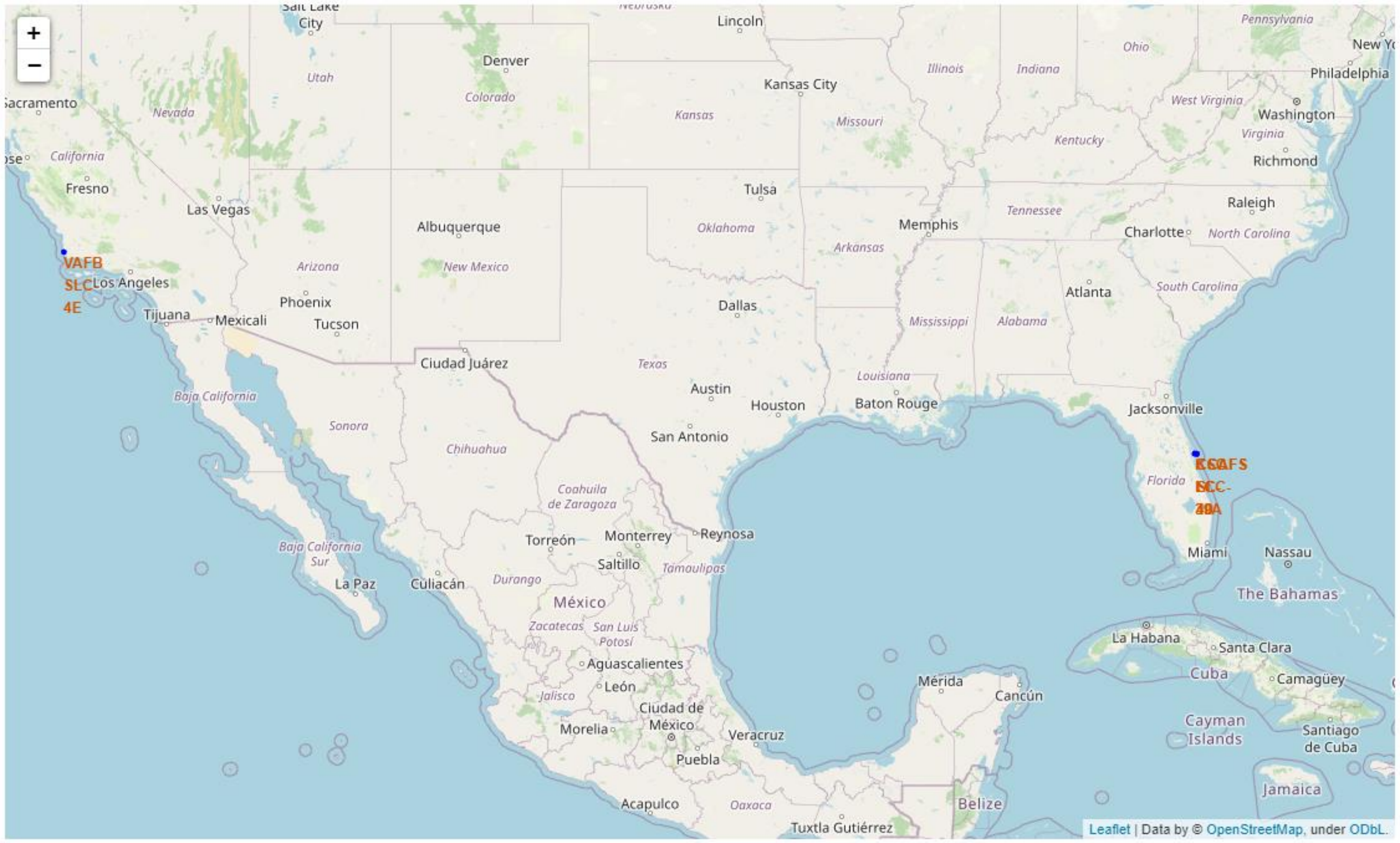
```
Done.
```

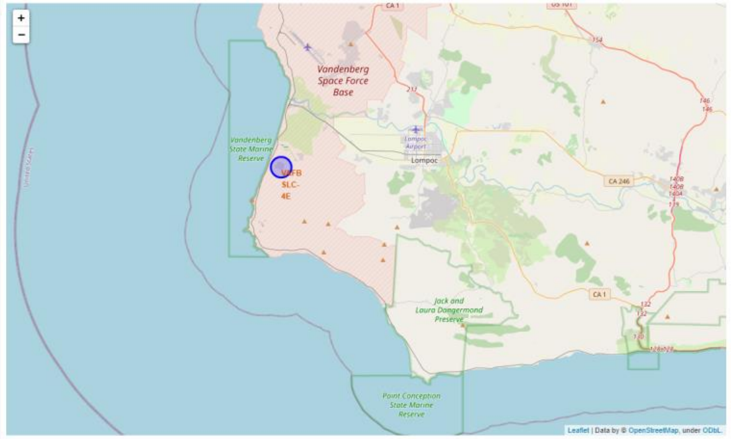
landing__outcome	landing_cnt
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 4

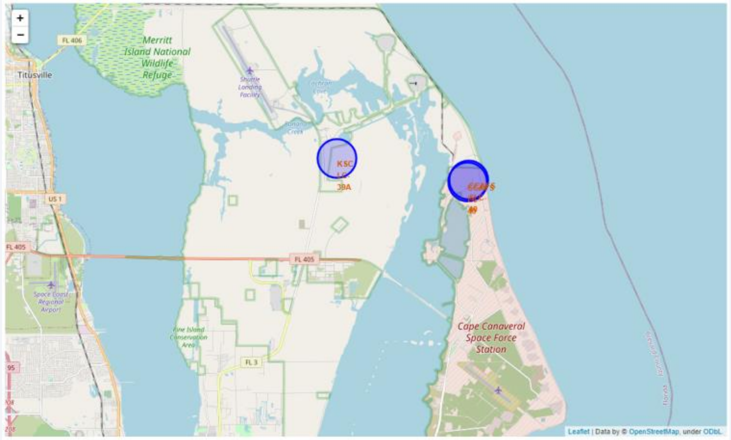
Launch Sites Proximities Analysis

Launch Sites Location





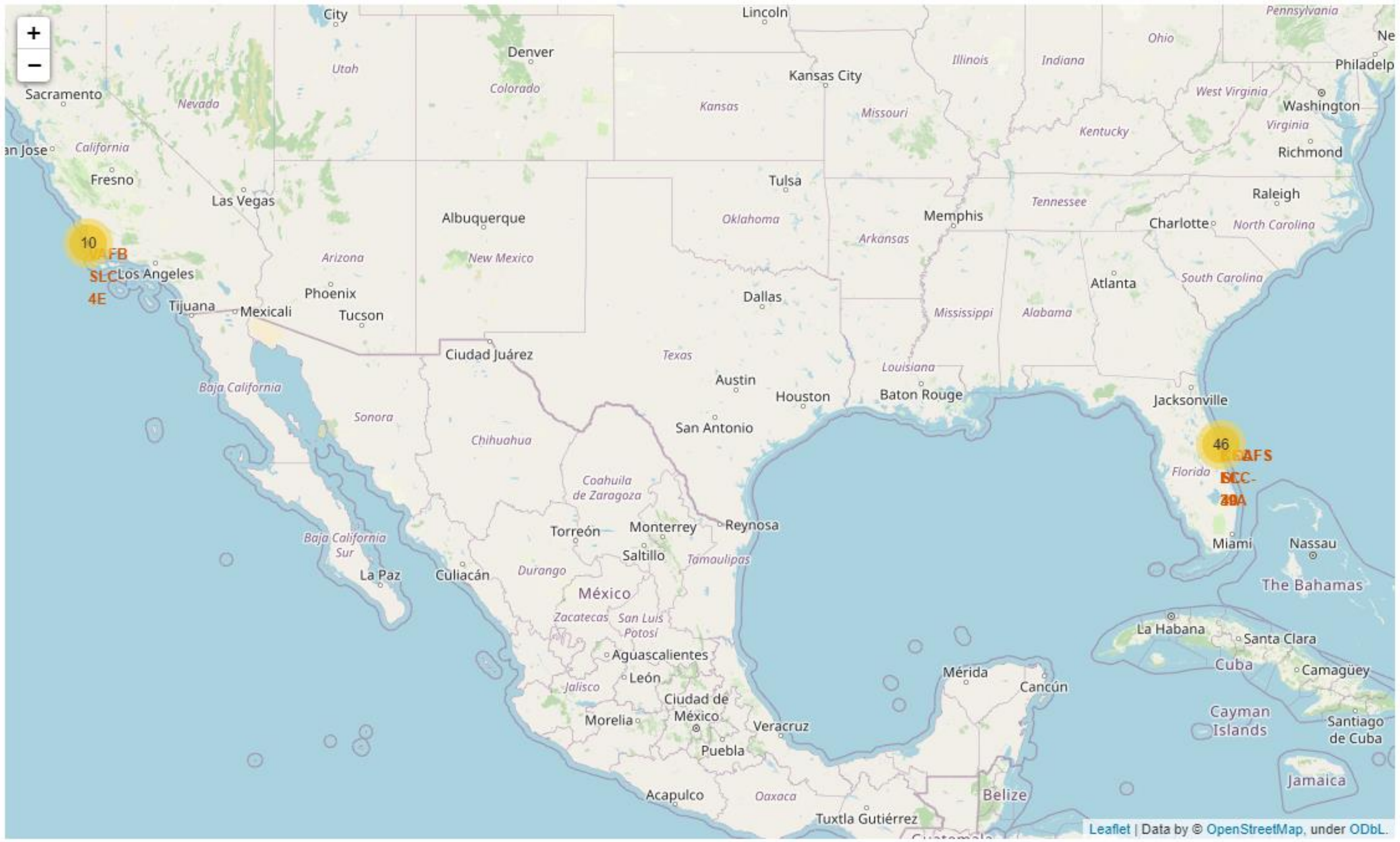
Vandenberg Space Force Base
VAFB
SLC-4E
Lompoc
Leaflet | Data by © OpenStreetMap, under ODbL.

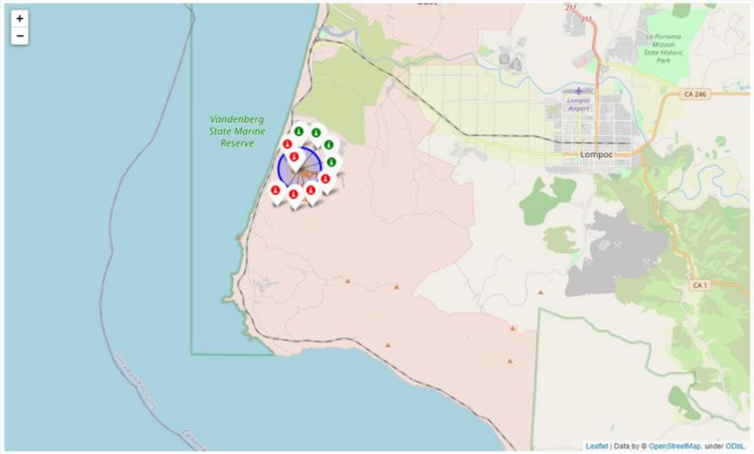


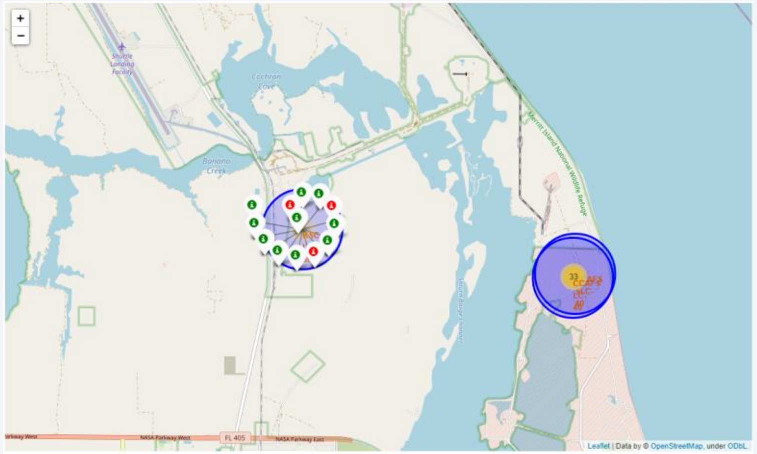
Merritt Island National Wildlife Refuge
Titusville
KSC
LC-39A
CCAFS
Cape Canaveral Space Force Station
Leaflet | Data by © OpenStreetMap, under ODbL.

- With elements on the map, you will see Launch Site location on both east and west coast

Launch Outcomes

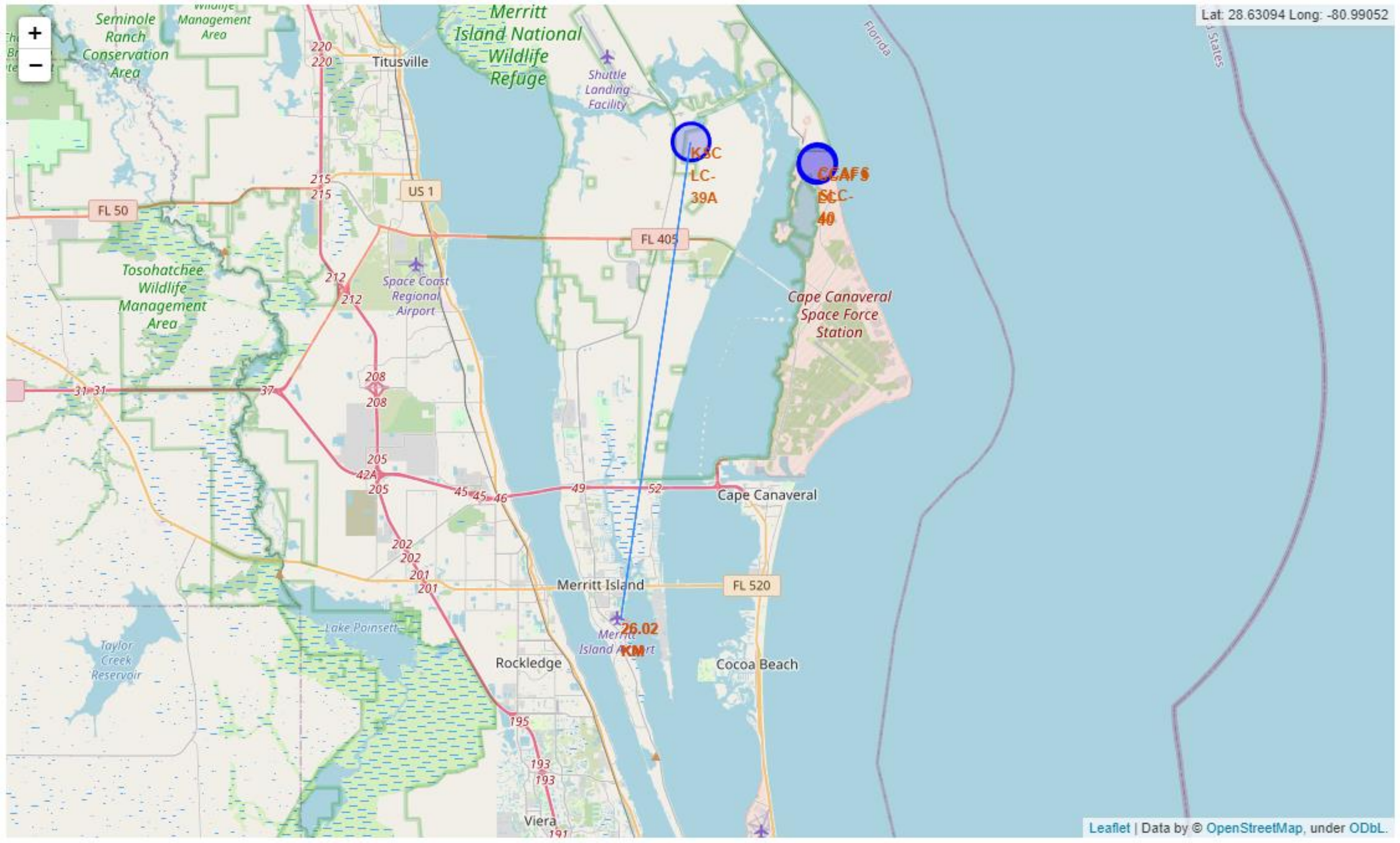






- From the maps, you will see success/fail outcomes from each Launch Site

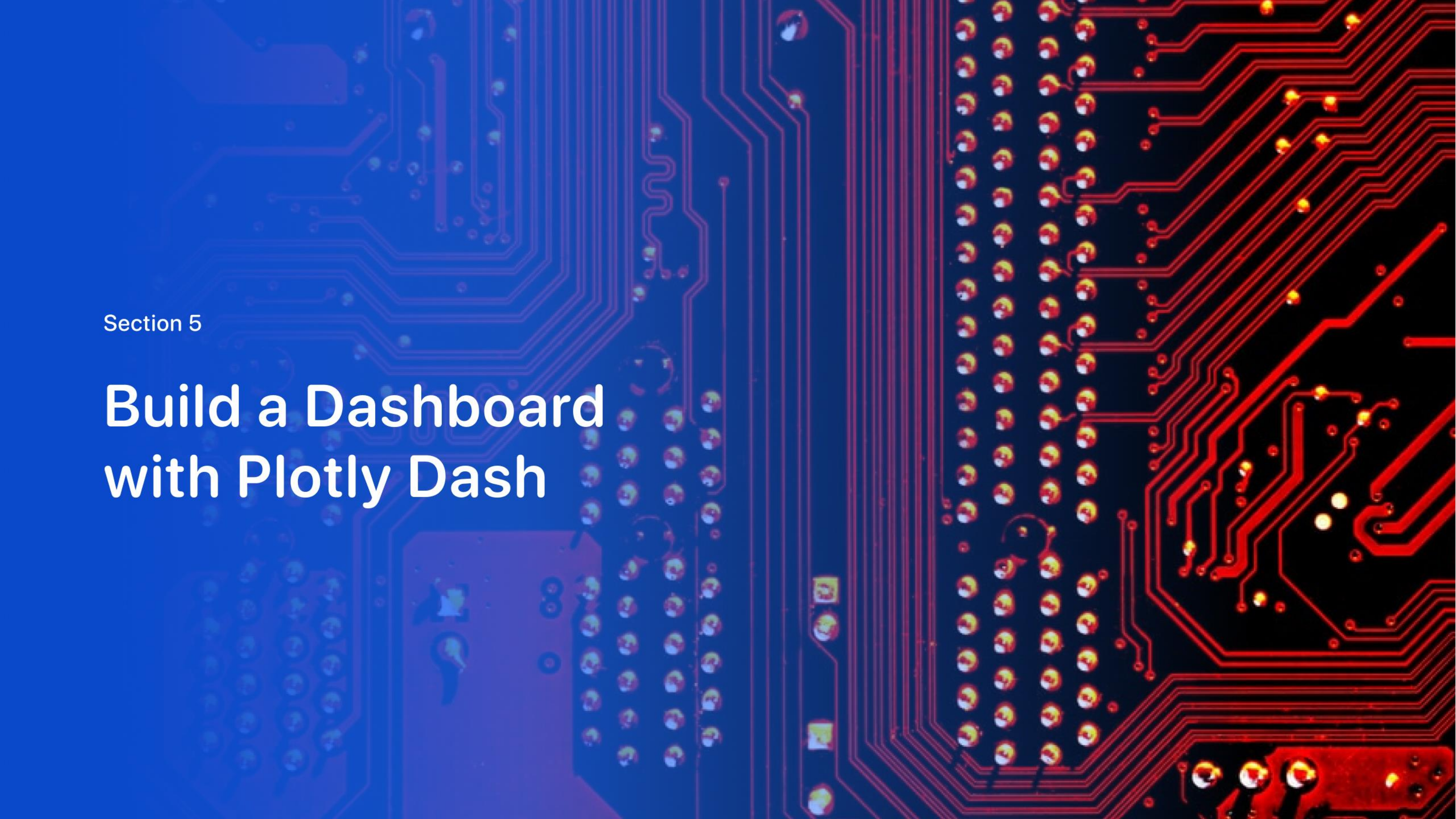
Launch Site Proximities


Lat: 28.63094 Long: -80.99052
+
−
Seminole Ranch Conservation Area
Wildlife Management Area
Titusville
Merritt Island National Wildlife Refuge
Shuttle Landing Facility
KSC LC-39A
CCAFS SLC-40
FL 50
US 1
FL 405
Tosohatchee Wildlife Management Area
Space Coast Regional Airport
Cape Canaveral Space Force Station
Cape Canaveral
Merritt Island
FL 520
26.02 KM
Merritt Island Airport
Rockledge
Cocoa Beach
Lake Poinsett
Taylor Creek Reservoir
Viera
Florida
Leaflet | Data by © OpenStreetMap, under ODbL.

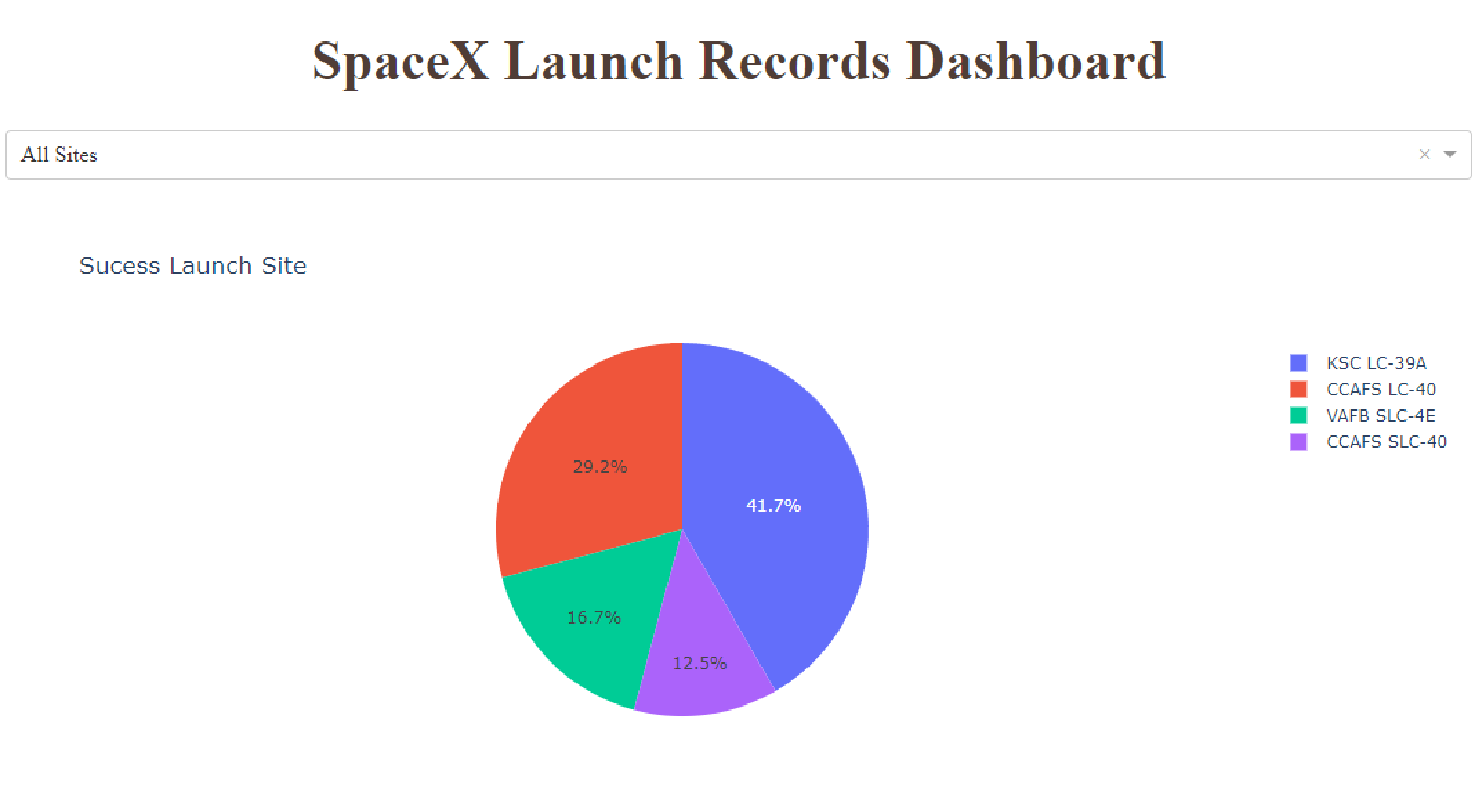
- As you see in the map, you will see distance between Launch Site KSC LC-39A and Merritt Island Airport

Section 5

Build a Dashboard with Plotly Dash



Success Rate Outcomes

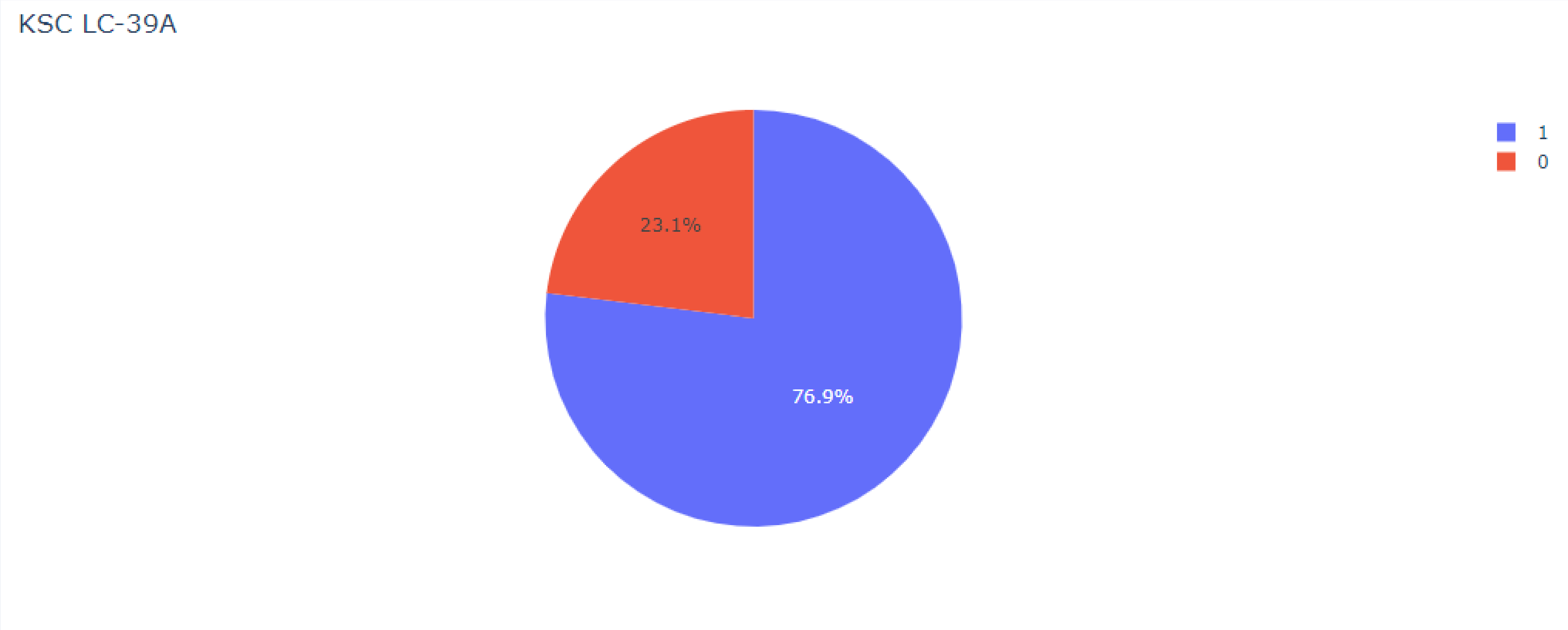


SpaceX Launch Records Dashboard

All Sites

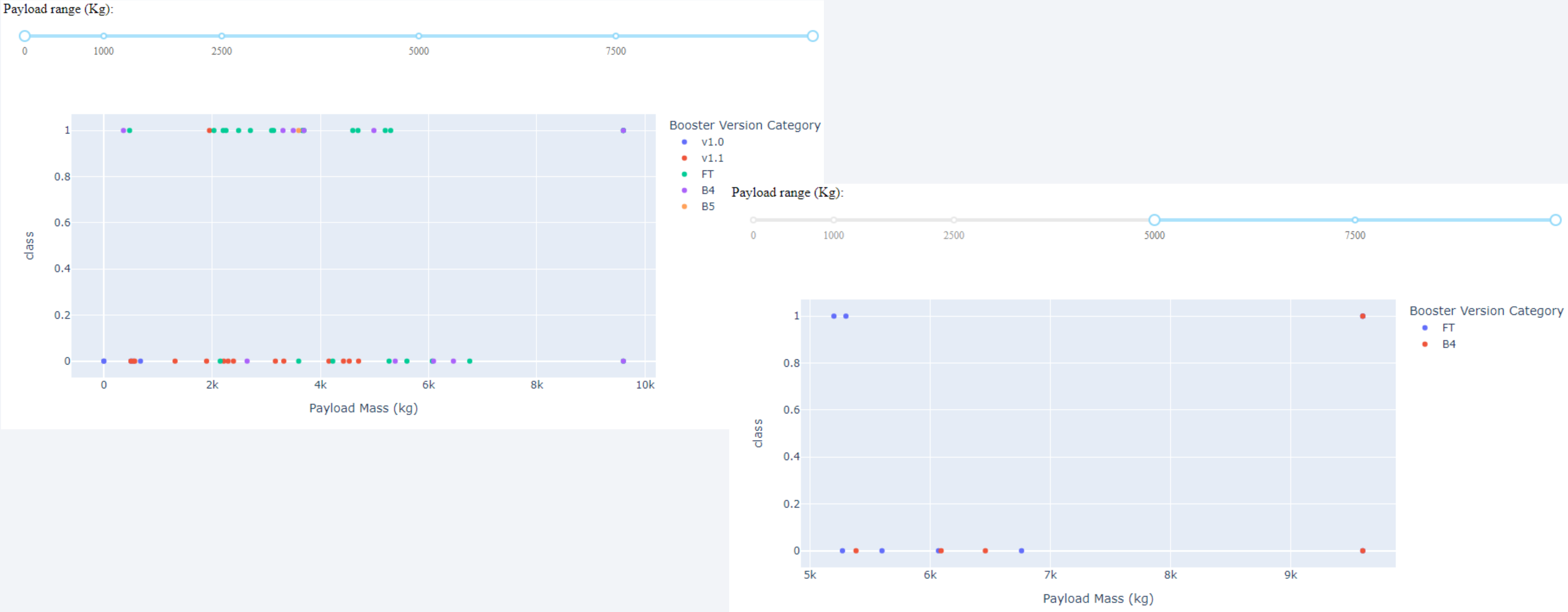
- With pie chart, you will see overall success rate for each Launch Site

Highest Success Launch Site



- The highest success Launch Site is KSC LC-39A which is 76.9%

Payload vs Launch Outcome



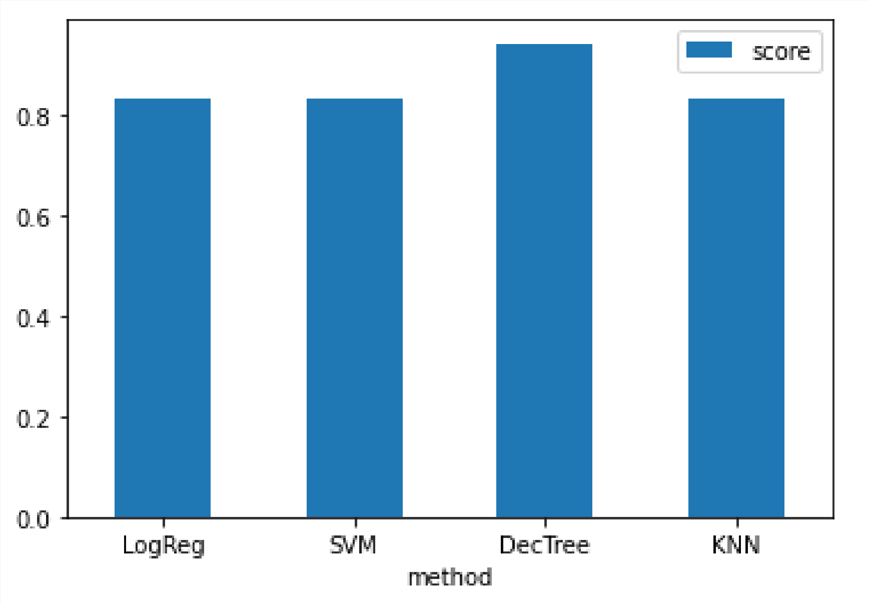
- As scatter chart, you will see that almost success will be payload under 4,000 kg.

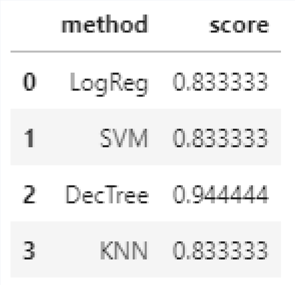
Section 6

Predictive Analysis (Classification)

Classification Accuracy

- After evaluate all classification models:
  - Logistic Regression
  - Support Victor Machine
  - Decision Tree
  - K-Nearest Neighbors
- As see in the chart, Decision Tree has most accuracy.

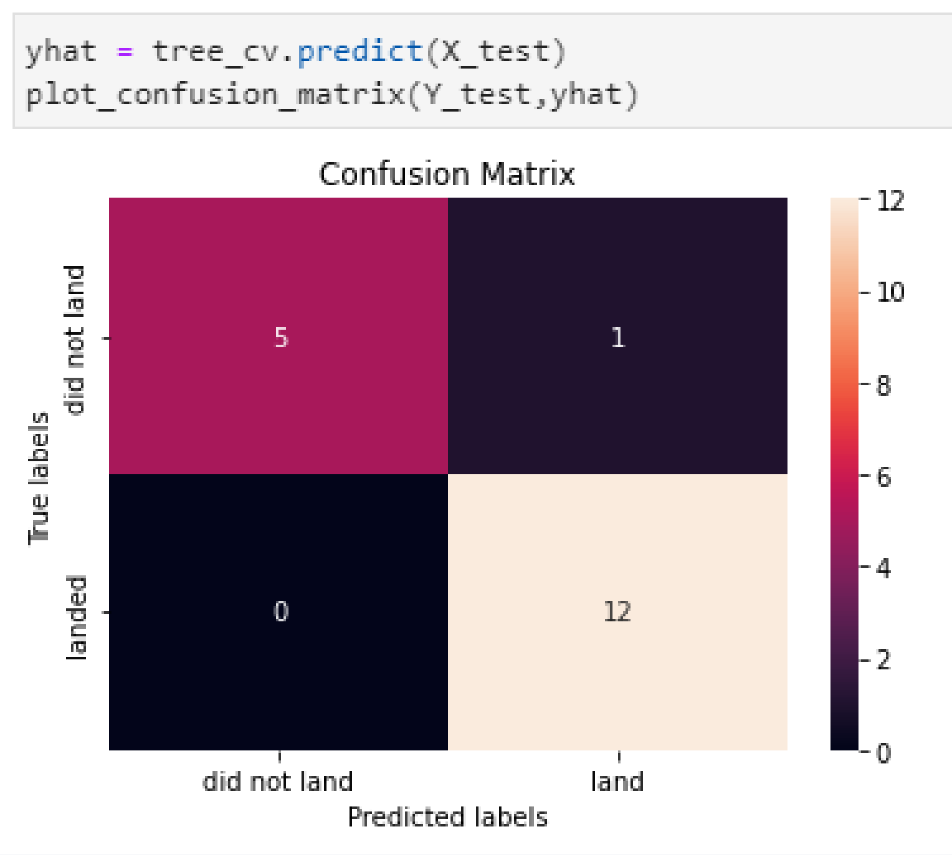




	method	score
0	LogReg	0.833333
1	SVM	0.833333
2	DecTree	0.944444
3	KNN	0.833333

Confusion Matrix

- Decision Tree model has less false positive (1) than other models (3).
- As see in confusion matrix, the prediction from Decision Tree model got only 1 false positive. This will made true positive and true negative higher than another models



```
yhat = tree_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```

Conclusions

- With the result we may use Decision Tree model to determine the first stage will be landed
- The Hyperparameter for Decision Tree model will be:
  - Criterion: gini
  - Max Depth: 6
  - Max Features: auto
  - Min Samples Leaf: 1
  - Min Samples Split: 2
  - Splitter: random
- Next step, we may use Decision Tree model along with this Hyperparameter to predict land success of the first stage as use the prediction to determine cost of a launch.

Appendix

- GitHub Repository

Thank you!