IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Maksym
Dmukhovskyy
06/07/2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection with API
  - Data collection with web scraping
  - Data wrangling
  - Exploratory data analysis with SQL
  - Exploratory data analysis with data visualisation
  - Interactive visual analytics with folium
  - Machine learning prediction
- Summary of all results
  - Exploratory data analysis
  - Interactive analytics
  - Predictive analytics

Introduction

- Project background and context

  Space X lists Falcon 9 rocket launches on its website for a price of 62 million dollars, significantly less than other providers who charge over 165 million dollars per launch. This cost efficiency largely stems from Space X's ability to reuse the first stage of their rockets. Thus, by predicting whether the first stage will successfully land, we can estimate the overall launch costs. Such information could be valuable if another company intends to compete with Space X in bidding for a rocket launch contract. The project objective is to develop a machine learning system capable of forecasting successful first-stage landings.

- Problems you want to find answers
  - How can we determine if the rocket will land successfully?
  - The correlation between different data points that determine the success rate of a landing.
  - Which conditions increase the success rate of landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - API request towards https://api.spacexdata.com/v4/rockets/
  - Web Scraping the webpage of https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- Perform data wrangling
  - The API data was retrieved and normalised as a JSON
  - The Web Scraped data was used in a frame by parsing the Wiki launch HTML tables
- Perform exploratory data analysis (EDA) using visualisation and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - The data was evaluated by different classification models
  - Each model evaluated the data using different combinations of parameters

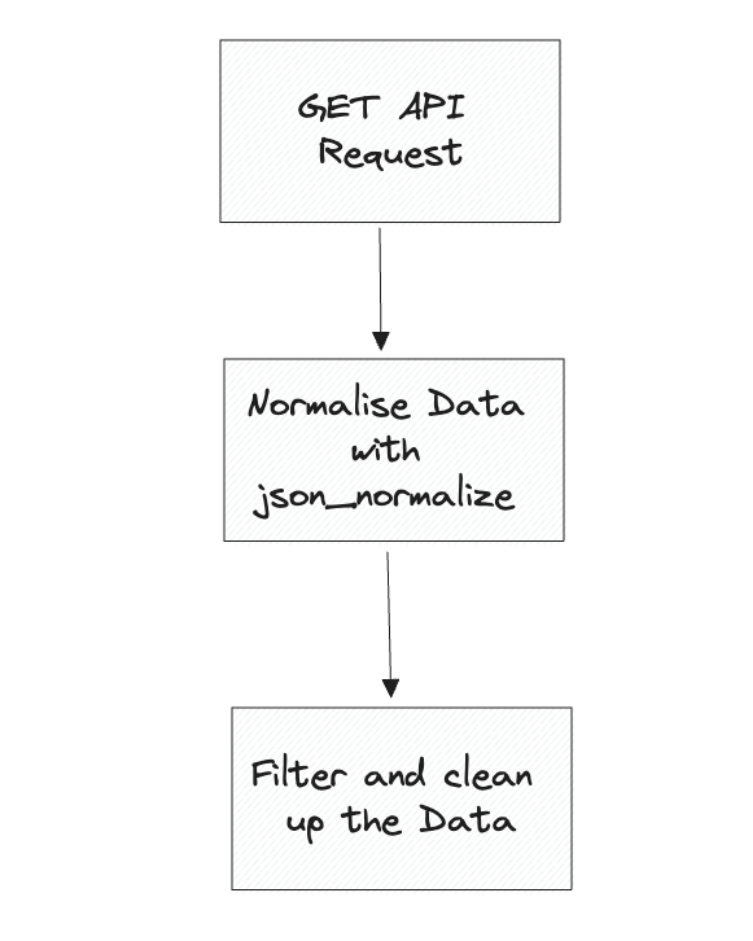
Data Collection

- Describe how data sets were collected.
  - API request towards https://api.spacexdata.com/v4/rockets/
  - Web Scraping the webpage of https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Data Collection – SpaceX API

- Using SpaceX open API that provides data in JSON

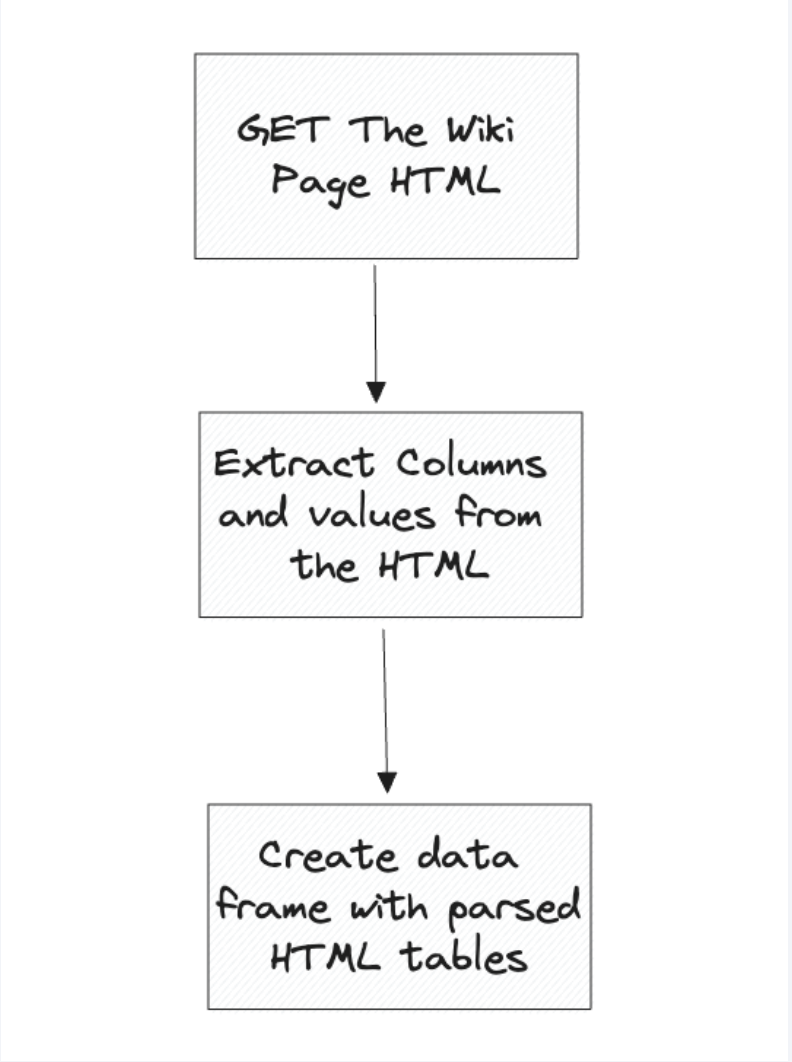
- GitHub URL https://github.com/maxdyy/IBM-Data-Science/blob/master/capstone-project/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Web Scraping the webpage of https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

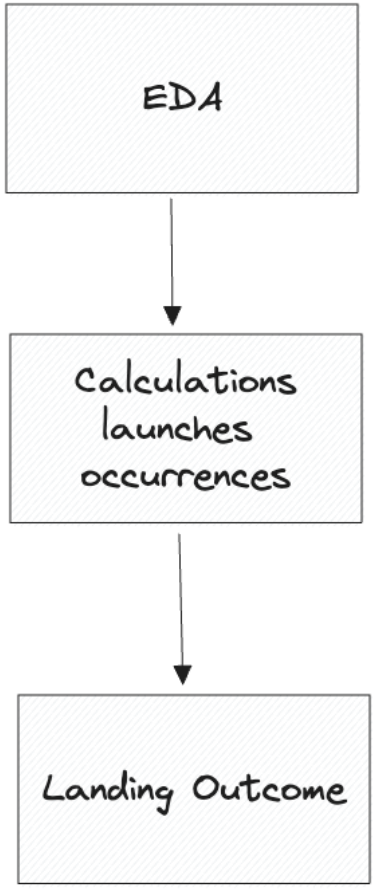
- GitHub URL https://github.com/maxdyy/IBM-Data-Science/blob/master/capstone-project/jupyter-labs-webscraping.ipynb



Data Wrangling

- Initial Exploratory Data Analysis
  - Calculate percentage of missing values
  - Check data types
- Calculations
  - Number of launches on each site
  - Number and occurrence of each orbit
- Create a landing outcome

- GitHub URL https://github.com/maxdyy/IBM-Data-Science/blob/master/capstone-project/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb



EDA with SQL

- Display the names of the unique launch sites in the space mission
  - %sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL;
- Display 5 records where launch sites begin with the string ‘CCA'
  - %sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
- Display the total payload mass carried by boosters launched by NASA (CRS)
  - %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)’;
- Display average payload mass carried by booster version F9 v1.1
  - %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';
- List the date when the first succesful landing outcome in ground pad was acheived.
  - %sql SELECT MIN(DATE) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)';
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - %sql SELECT DISTINCT(BOOSTER_VERSION) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
- List the total number of successful and failure mission outcomes
  - %sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) FROM SPACEXTBL GROUP BY MISSION_OUTCOME;
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - %sql SELECT DISTINCT(BOOSTER_VERSION) FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
- GitHub URL https://github.com/maxdyy/IBM-Data-Science/blob/master/capstone-project/jupyter-labs-eda-sql-coursera_sqllite.ipynb

EDA with Data Visualization

- A scatter point chart for the relationship between Flight Number and Launch Site
- A scatter point chart for the relationship between Payload and Launch Site
- A bar chart to visualise the relationship between success rate of each orbit type
- A scatter point chart for the relationship between FlightNumber and Orbit type
- A scatter point chart for the relationship between Payload and Orbit type
- A line chart to visualise the launch success yearly trend
- GitHub URL https://github.com/maxdyy/IBM-Data-Science/blob/master/capstone-project/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

Build an Interactive Map with Folium

- We used markers, circles and lines on the folium map
  - Markers for points like launch sites
  - Circles for areas around points like NASA
  - Lines for distance visualisation between points
- GitHub URL https://github.com/maxdyy/IBM-Data-Science/blob/master/capstone-project/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- We developed a dashboard with Plotly Dash
- The dashboard has dropdown where we can select the Sites
- We plotted a pie chart showing the total launches based on the selected sites
- We plotted a scatter graph for showing the correlation between payload and success of the mission for a specific site
- GitHub URL https://github.com/maxdyy/IBM-Data-Science/blob/master/capstone-project/spacex_dash_app.py

Predictive Analysis (Classification)

- We loaded the data with pandas and numpy
- We built the machine learning models for landing predictions
- We used accuracy metrics for evaluating the model performance
- We found the best model based on the accuracy
- Add the GitHub URL https://github.com/maxdyy/IBM-Data-Science/blob/master/capstone-project/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

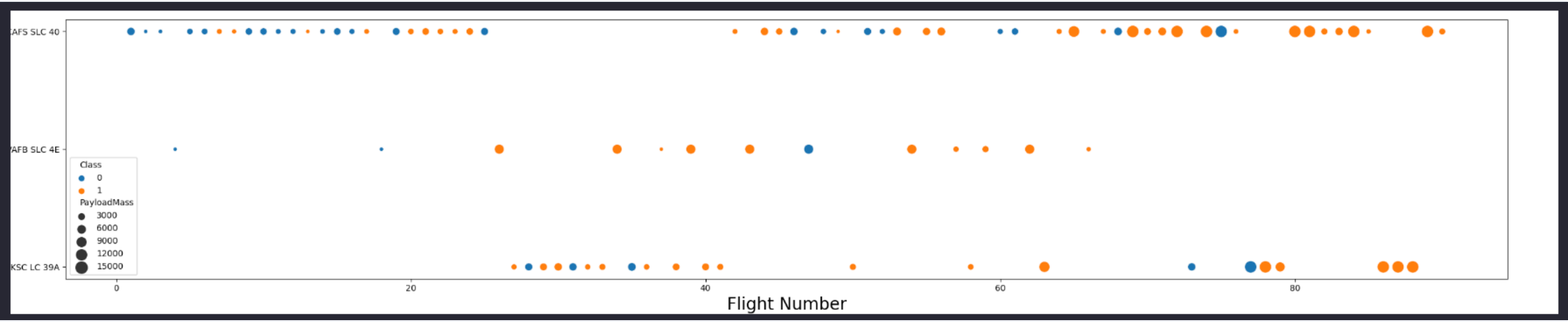
Results

- The data has 4 different launch sites
- In 2015 was the first successful landing
- The number of successful landings improved over the years
- The launches are performed form sites on west and east coast of US
- Most launches were performed from the east coast
- The payload of the Falcon 9 rocket can impact the success of landing
- The data shows two failed attempts at landing in 2018

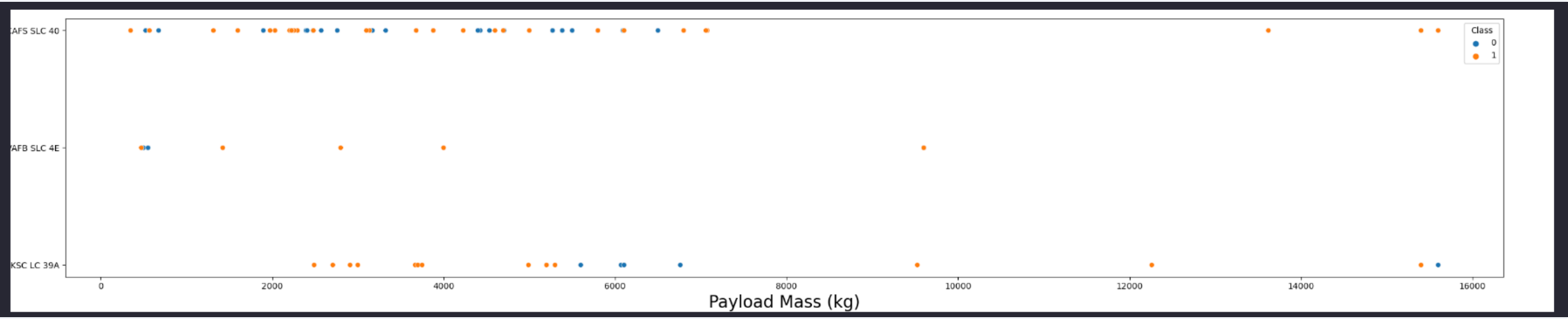
Section 2

Insights drawn from EDA

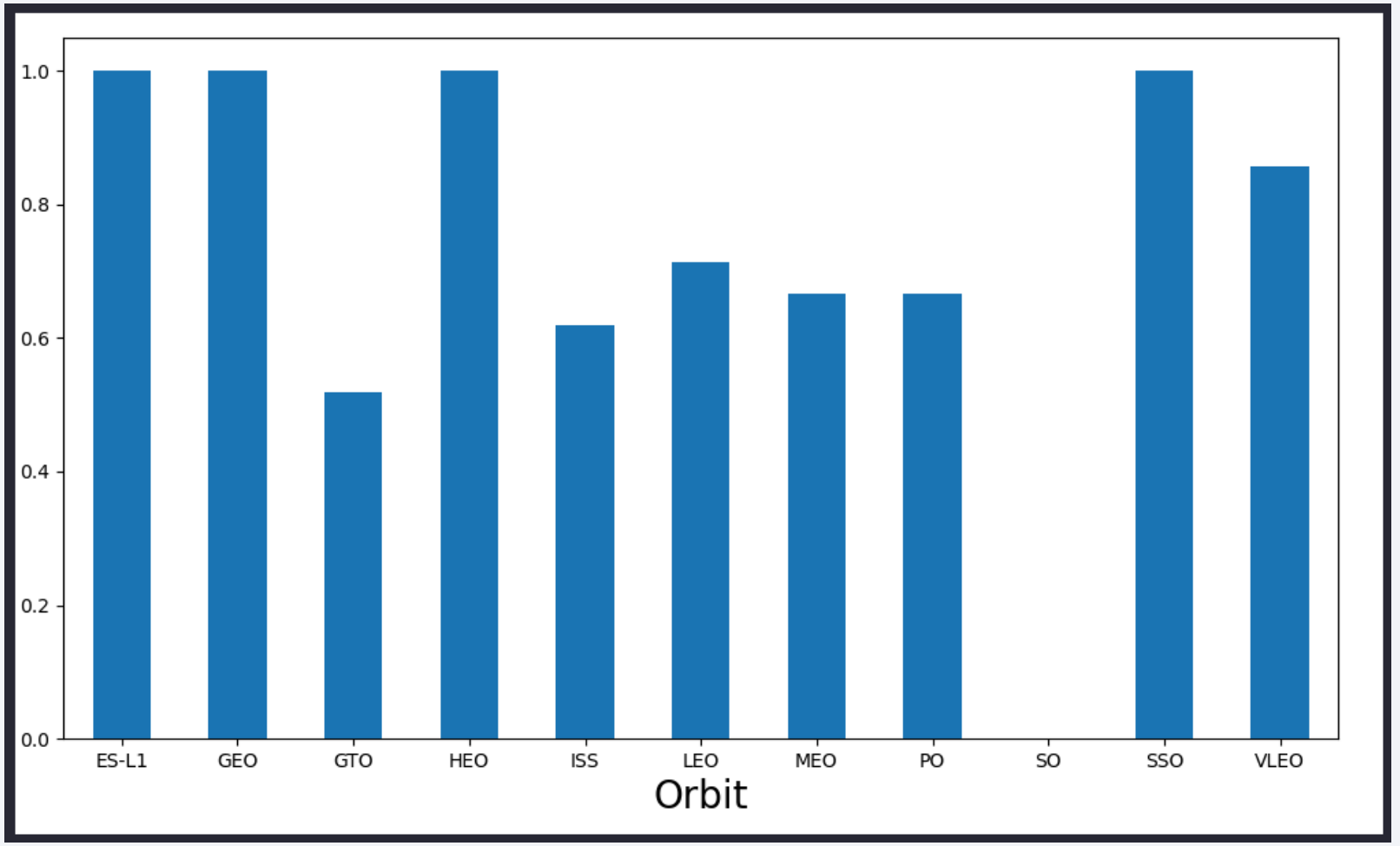
Flight Number vs. Launch Site



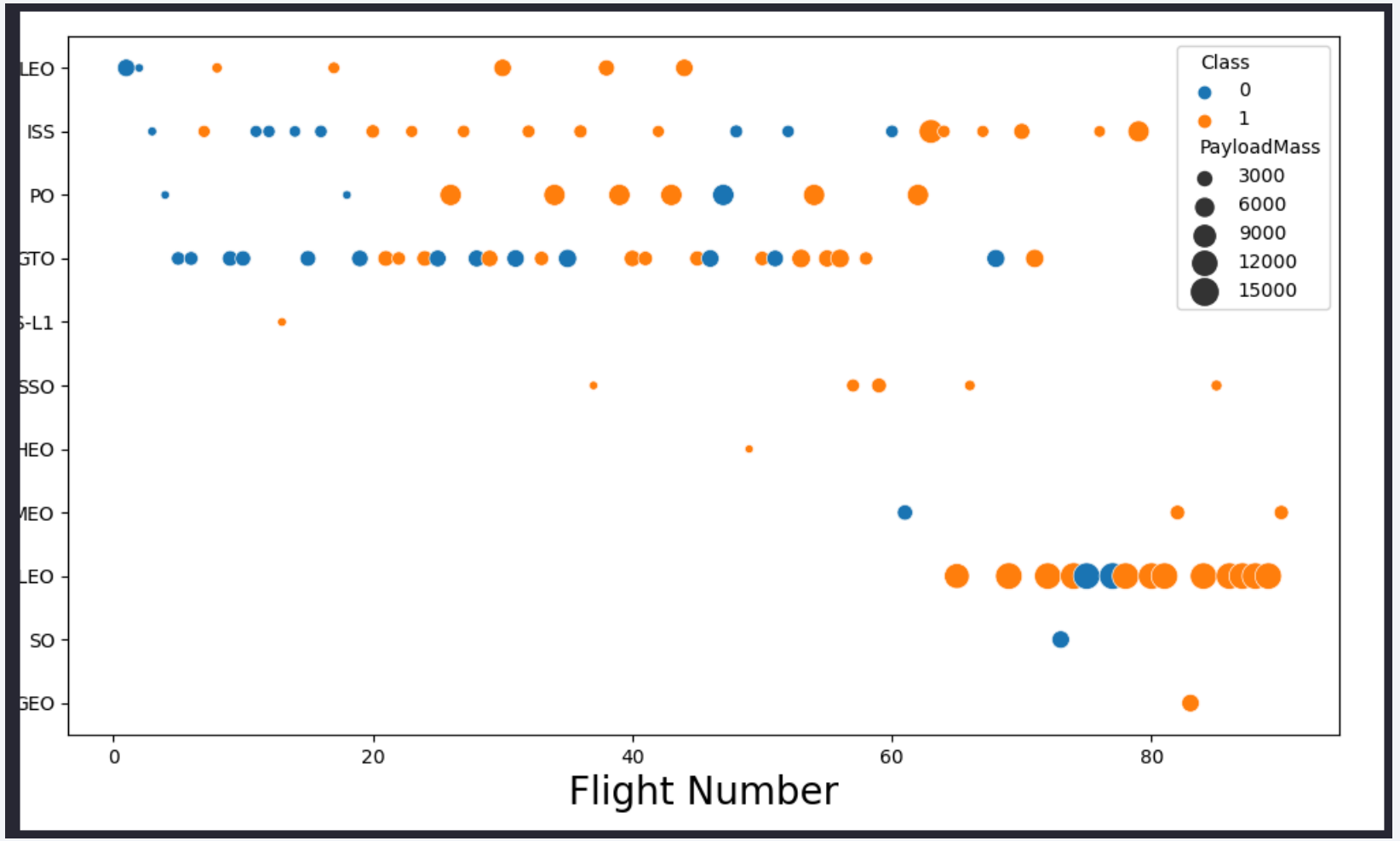
Payload vs. Launch Site



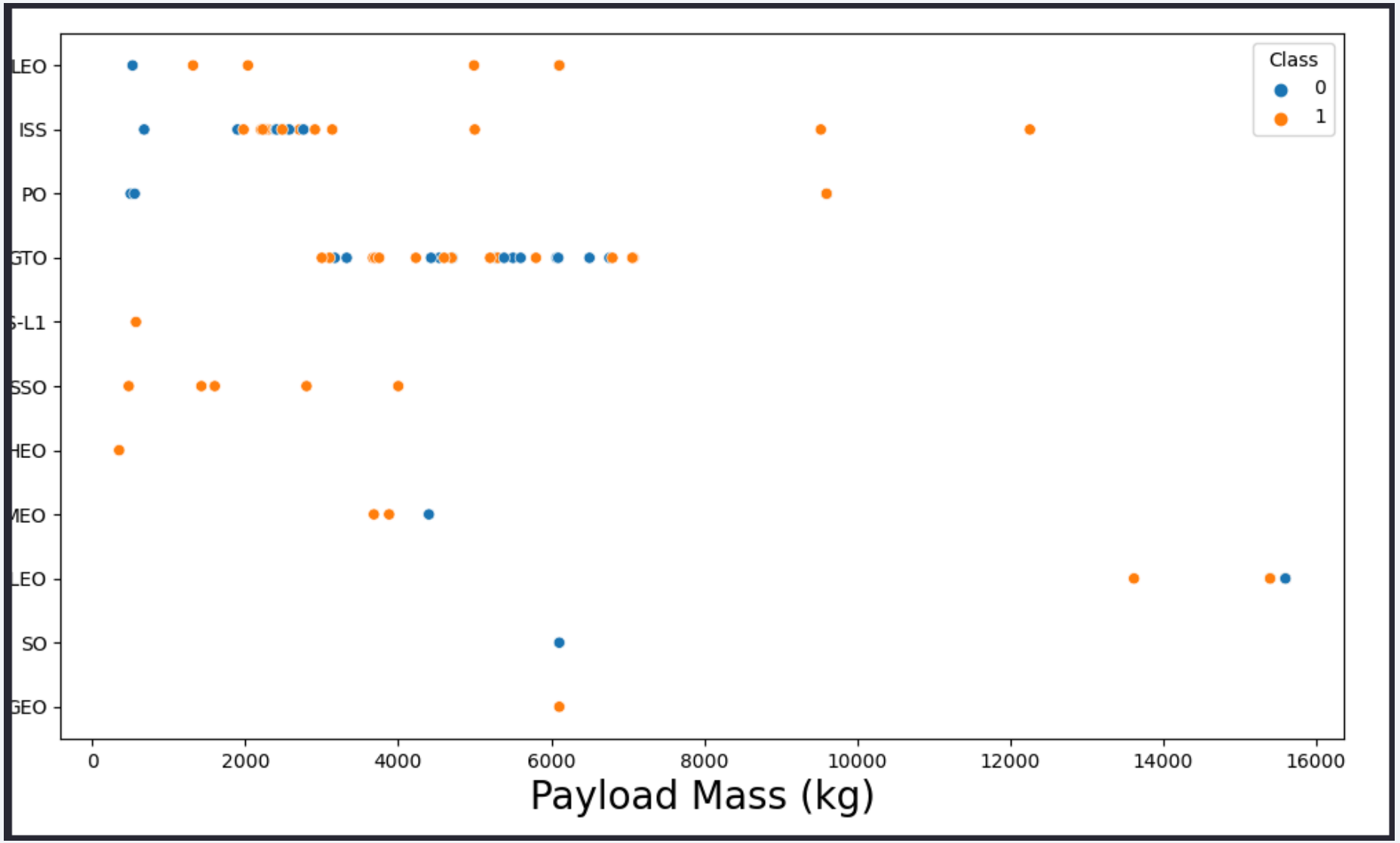
Success Rate vs. Orbit Type



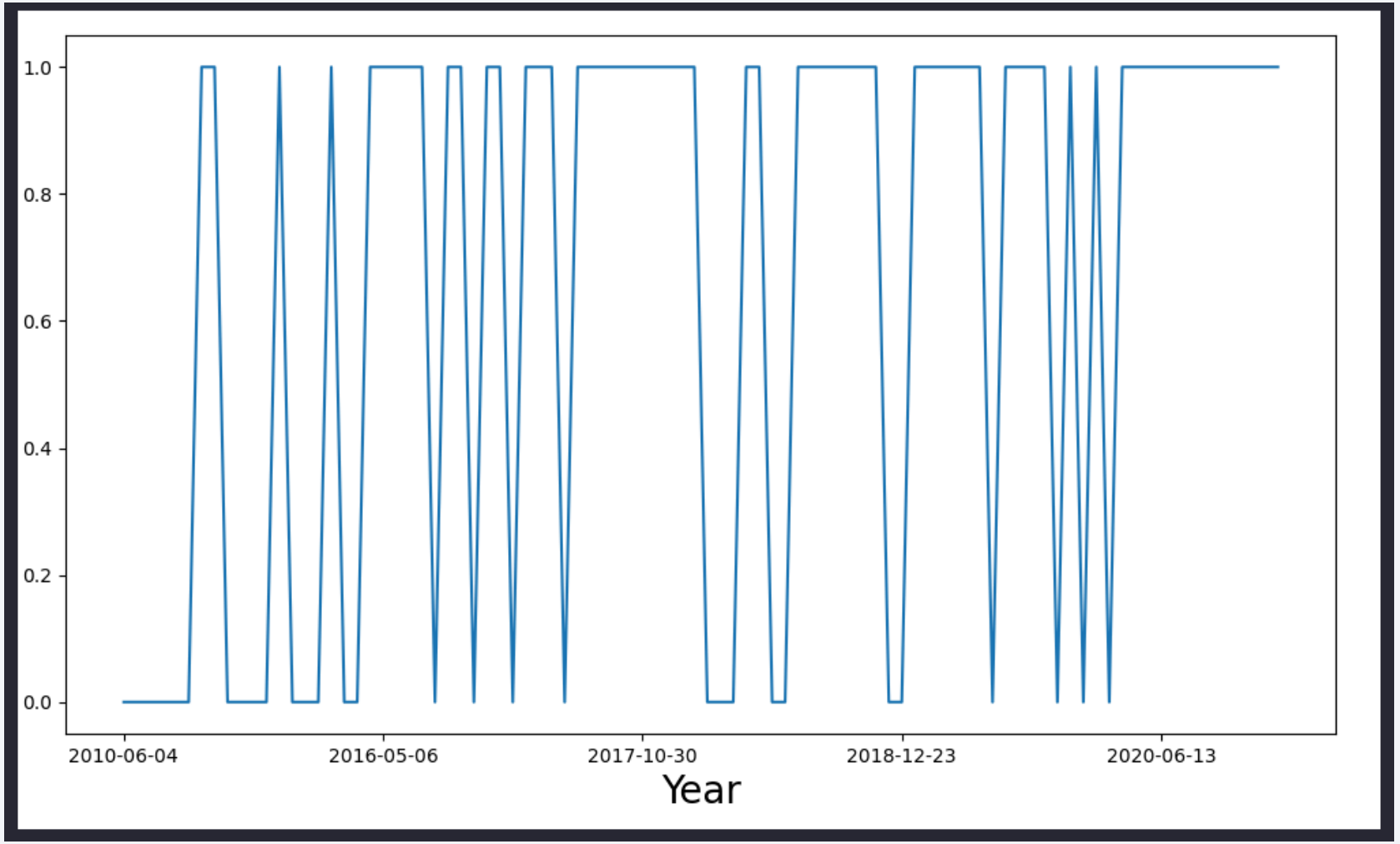
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

- We used the DISTINCT query to select unique names

```
In [7]: %sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL;
 * sqlite:///my_data1.db
Done.
Out[7]:
```

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

Launch Site Names Begin with 'CCA'

- We used CCA% to select the sites that begin with CCA

Display 5 records where launch sites begin with the string 'CCA'

```
In [8]: %sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
 * sqlite:///my_data1.db
Done.
```

Out[8]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Lan
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Fai
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Fai
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	

Total Payload Mass

- We used the SUM in the query to calculate the total mass

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

```
In [9]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)';
 * sqlite:///my_data1.db
Done.
Out[9]: SUM(PAYLOAD_MASS__KG_)
        45596.0
```

Average Payload Mass by F9 v1.1

- We used the AVG query to calculate the average payload mass carried by booster version F9 v1.1

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

```
In [10]: %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';
 * sqlite:///my_data1.db
Done.
Out[10]: AVG(PAYLOAD_MASS__KG_)
         2928.4
```

First Successful Ground Landing Date

- We used the MIN(DATE) query to find the first successful landing outcome on ground pad

```
In [12]: %sql SELECT MIN(DATE) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)';
 * sqlite:///my_data1.db
Done.
Out[12]: MIN(DATE)
         01/08/2018
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- We used WHERE together with AND to filter the successful landings

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

```
In [13]: %sql SELECT DISTINCT(BOOSTER_VERSION) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_M
 * sqlite:///my_data1.db
Done.
```

Out[13]:

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

Total Number of Successful and Failure Mission Outcomes

- Here we created a query that groups by MISSION_OUTCOME

List the total number of successful and failure mission outcomes

```
In [14]: %sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) FROM SPACEXTBL GROUP BY MISSION_OUTCOME;
 * sqlite:///my_data1.db
Done.
```

Out[14]:

Mission_Outcome	COUNT(MISSION_OUTCOME)
None	0
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- The query goes through the BOOSTER_VERSION and selects the MAX of PAYLOAD_MASS__KG_

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

```
In [15]: %sql SELECT DISTINCT(BOOSTER_VERSION) FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FRO
 * sqlite:///my_data1.db
Done.
```

Out[15]:

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

- The query select the year 2015 with WHERE substr(Date,7,4)='2015' and filters the failures with AND Landing_Outcome LIKE 'Failure%'

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

In [17]:
```
%sql SELECT substr(Date, 4, 2) AS month, Landing_Outcome, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE subst
```

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

Out[17]:

month	Landing_Outcome	Booster_Version	Launch_Site
10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- The query ranks the landing outcomes with the booster version using RANK() OVER(ORDER BY PAYLOAD_MASS__KG_ DESC) AS RANK FROM SPACEXTBL

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, in descending order.

In [18]:
```
%sql SELECT BOOSTER_VERSION, RANK() OVER(ORDER BY PAYLOAD_MASS__KG_ DESC) AS RANK FROM SPACEXTBL;
```

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

Out[18]:

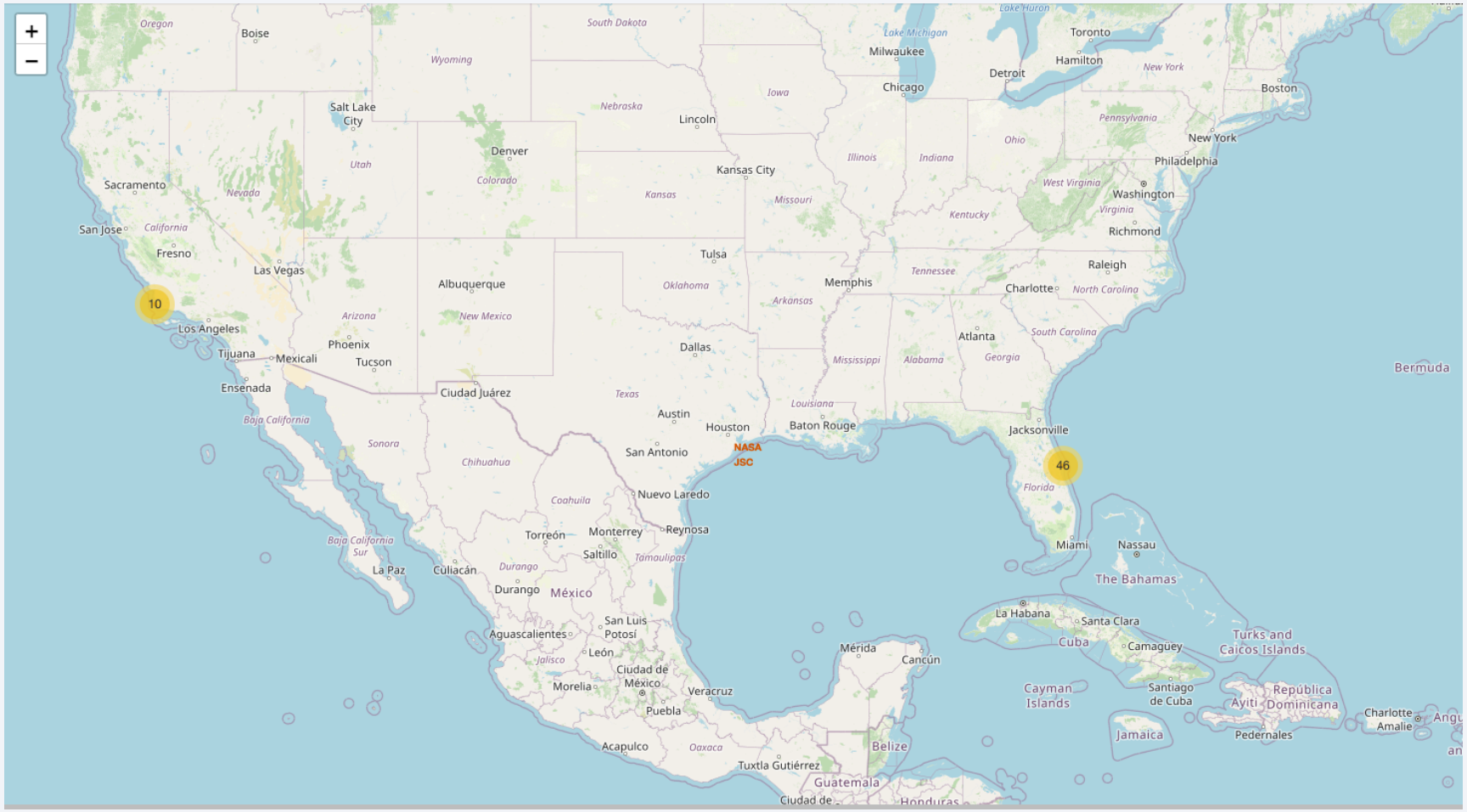
Booster_Version	RANK
F9 B5 B1048.4	1
F9 B5 B1049.4	1
F9 B5 B1051.3	1
F9 B5 B1056.4	1
F9 B5 B1048.5	1
F9 B5 B1051.4	1
F9 B5 B1049.5	1
F9 B5 B1060.2	1
F9 B5 B1058.3	1
F9 B5 B1051.6	1
F9 B5 B1060.3	1
F9 B5 B1049.7	1
F9 B5 B1049.6	13

Section 3

Launch Sites Proximities Analysis

Launches sites and their frequency

- We can see that there’re two launch sites, on east and west coasts. We can see that the west coast is most popular.

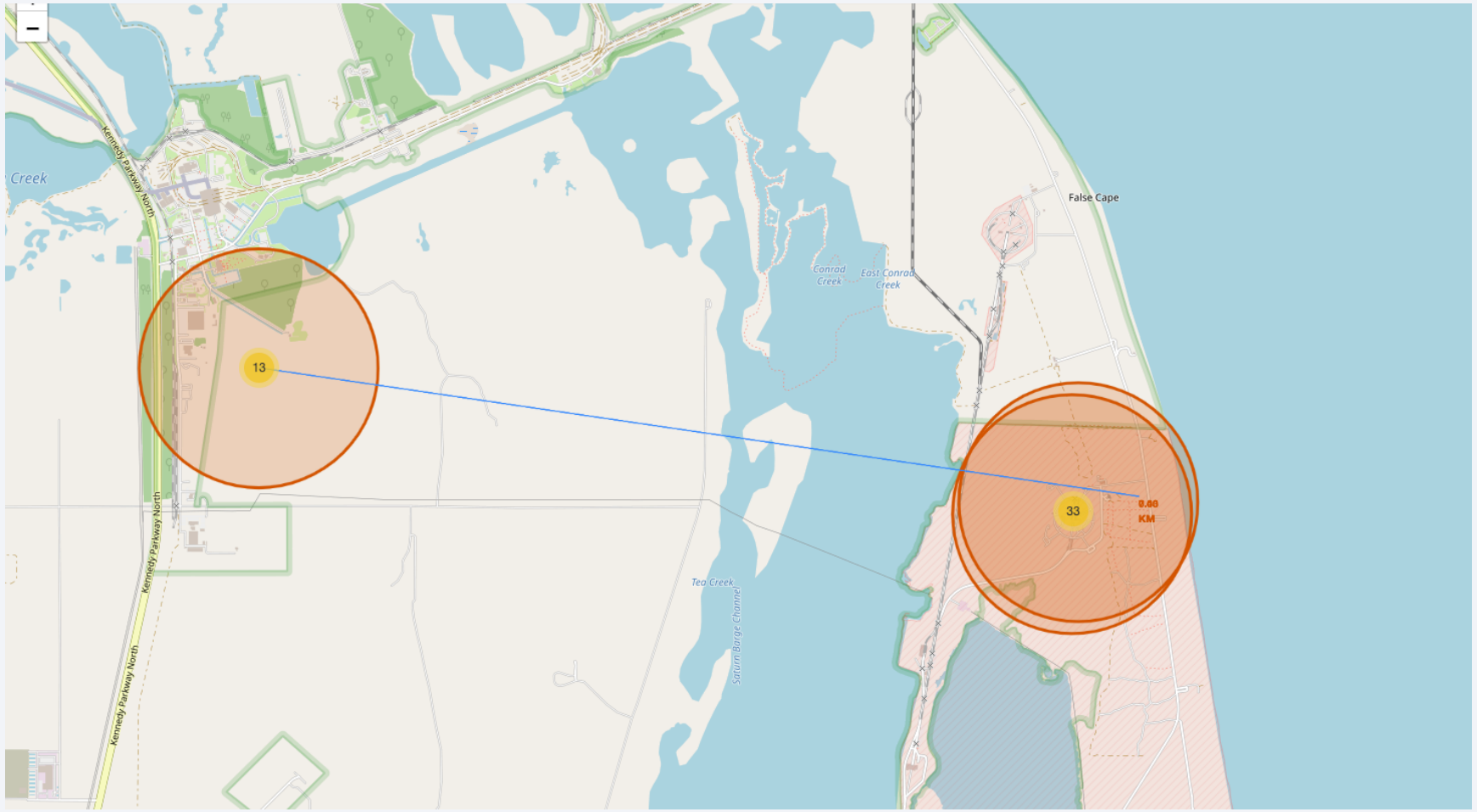


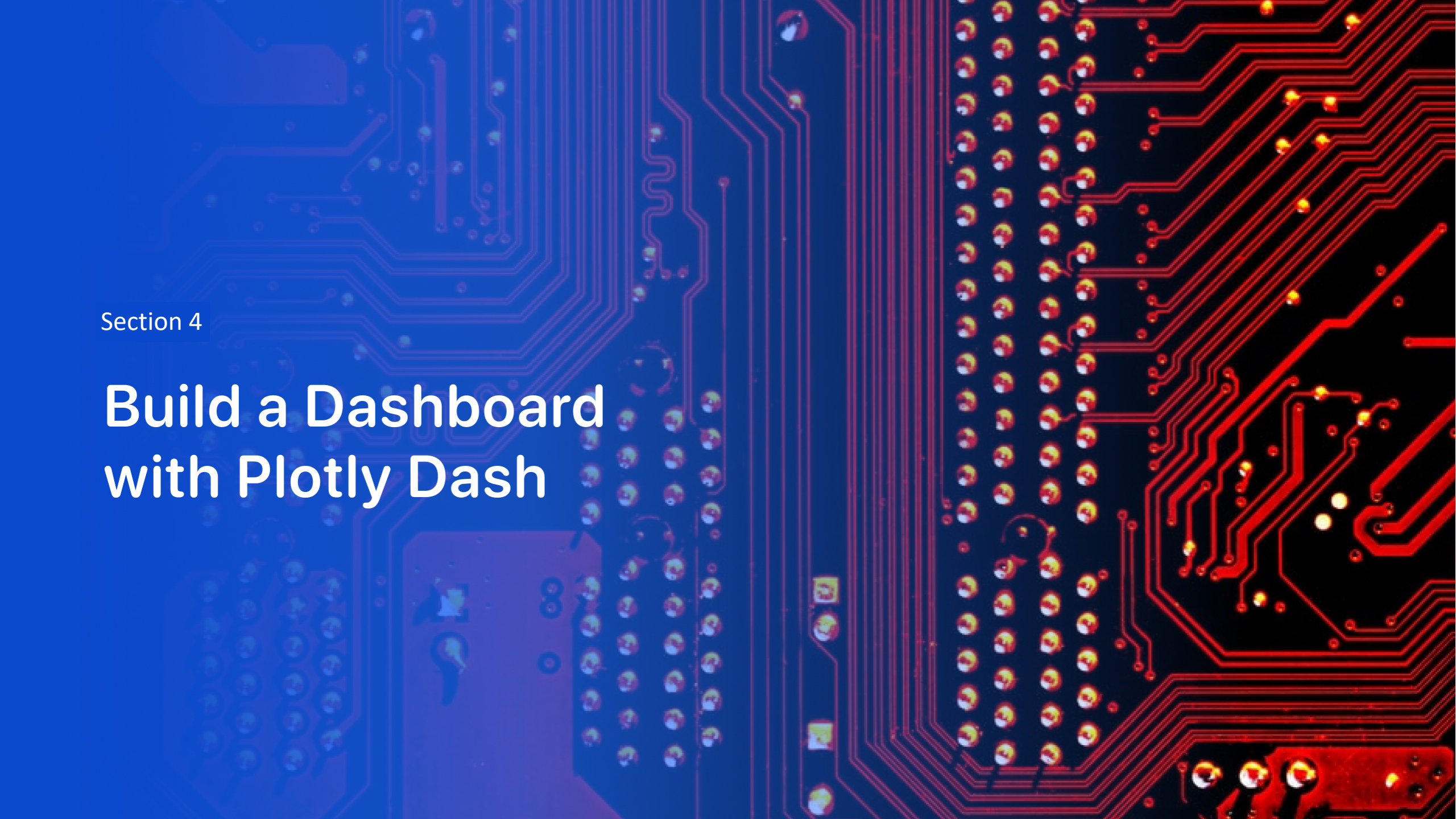
Global map of launch sites

- We can see that all the launches were performed from the United States

The distance between launch sites and the coast line

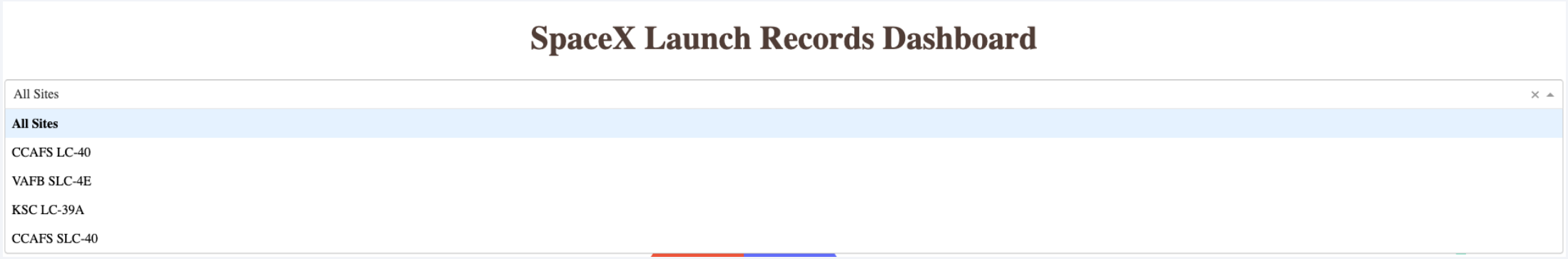
- We can see that the most frequent launch site is the one closest to the coast line




Section 4
Build a Dashboard with Plotly Dash

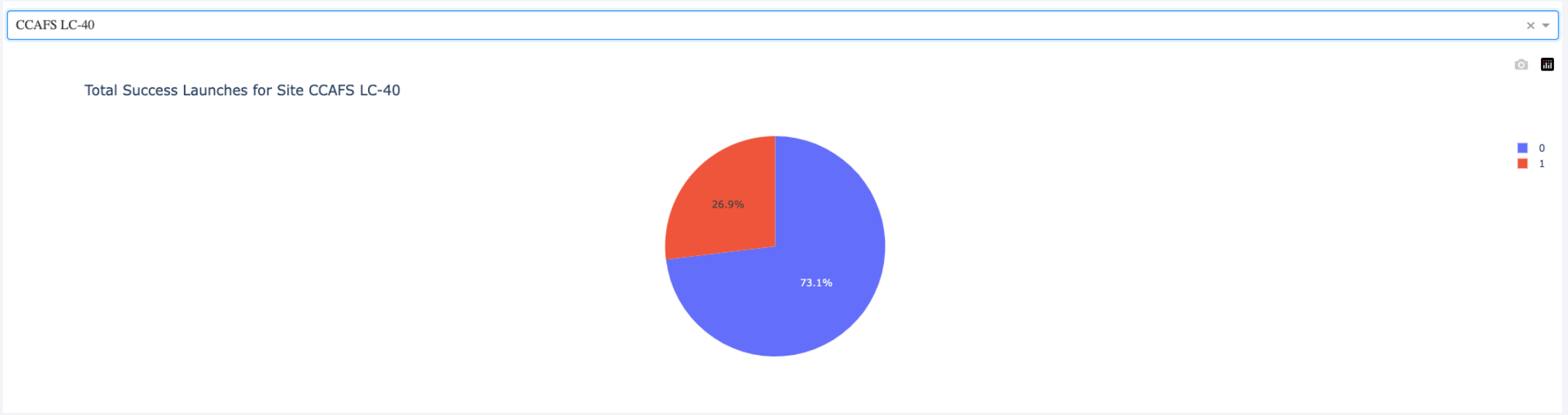
Dashboard options

- The dashboard allows to filter the data by sites with a dropdown



Dashboard success rate chart

- When selecting a launch site, the pie chart below will show the success rate for all the launches from that exact site



Dashboard correlation Payload and Success

- A scatter point chart will show the correlations between the success of launch and payload mass. The chart also shows the booster versions.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- All models had an accuracy over 0.80, but the Decision Tree model had an accuracy of 0.88

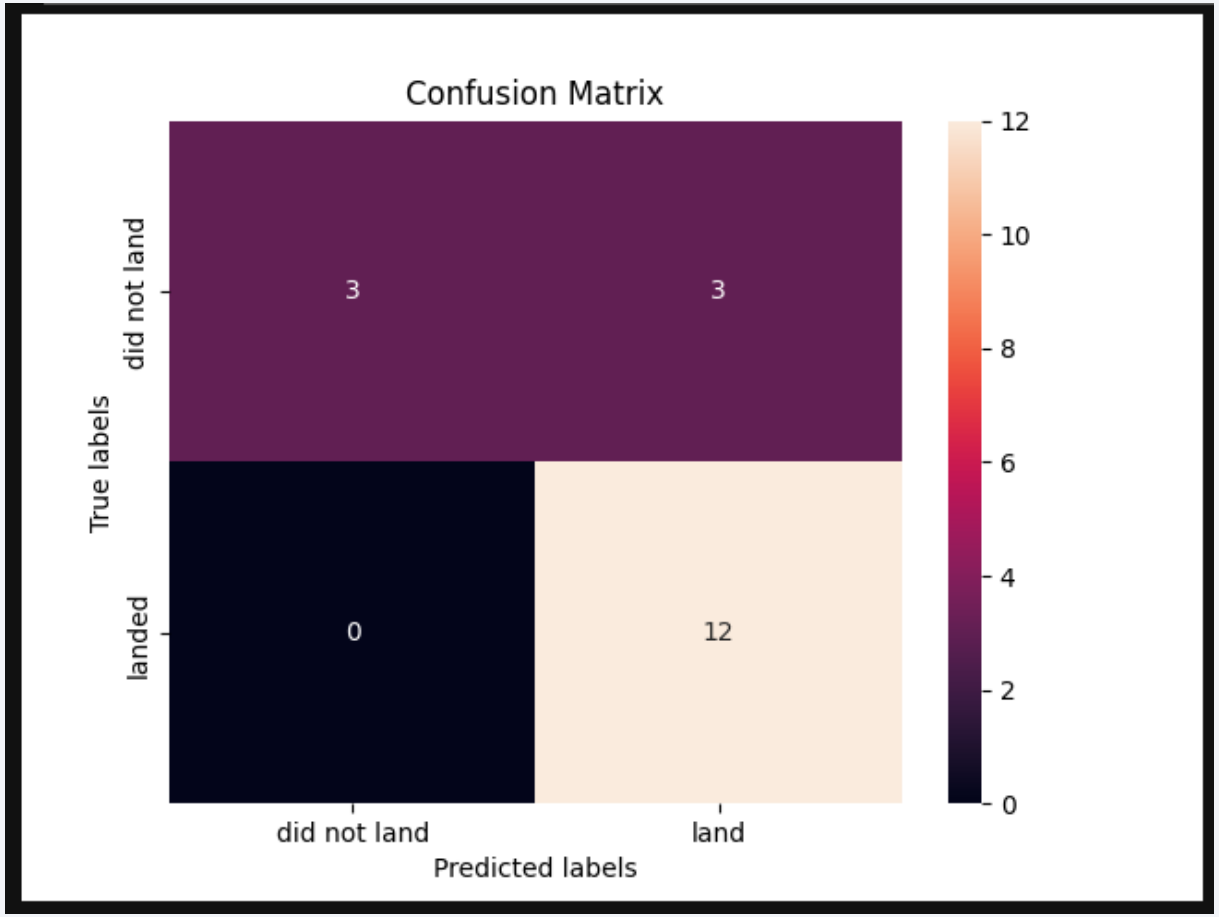
Find the method performs best:

In [32]:
```
print("Logistic Regression: ", logreg_cv.score(X_test, Y_test))
print("SVM: ", svm_cv.score(X_test, Y_test))
print("Decision Tree: ", tree_cv.score(X_test, Y_test))
print("K Nearest Neighbors: ", knn_cv.score(X_test, Y_test))
```

```
Logistic Regression:  0.8333333333333334
SVM:  0.8333333333333334
Decision Tree:  0.8888888888888888
K Nearest Neighbors:  0.8333333333333334
```

Confusion Matrix

- The confusion matrix for the Decision Tree model



Conclusions

- Launch success rates look directly correlated to the amount of launches from the same site
- The decision Tree classifier is the best algorithm to predict the success rate
- The success rates started to increase with the years
- KSC LC-39A has the biggest percentage in success launches
- The KSC LC-39A is also the most popular launch site

Thank you!