Winning Space Race with Data Science

Aziz Imanov
27.11.2021

Outline

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

Executive Summary

- **Methodologies applied in this project**:

  1) Data Collection (API, Web-scraping, CSV file available online);

  2) Data Preprocessing / Wrangling;

  3) Exploratory Data Analysis (EDA) and analytical visualization;

  4) Building predictive Machine Learning models and evaluation.

- **Summary of all results**:

  After rigorous analysis of historical data it becomes certain that some mutually unrelated parameters of a rocket launch can predetermine the mission's landing outcome.

Introduction

- **The background of the project**:

  As you may or may not be aware, as any other industry space industry is also becoming more and more commercialized by private companies like: SpaceX, Rocket Lab, Blue Origin, Boeing, etc. The increasing number of private space companies builds up a competition for services and products that can be provided for the industry. This competition among the companies amplifies their will to come up with better or new services and products in terms of factors like: quality, efficiency and cost. For a company that wants to participate and lead the market of space industry it is crucial to find the 'Golden Ratio' between these factors. To achieve this goal a company needs to use modern tools and technologies. One of the main tools on this endeavor and one which I am going to use extensively to get insights from a competitive company is Data Science.

- **Key question to answer**:

  The goal of this project is – based on historical data to determine to a certain level of accuracy whether the landing of the SpaceX rocket's first stage is going to be successful or not during the upcoming launch. The outcome of this research will potentially enable us to set a competitive price for the rocket launch provided by our company.

Section 1

Methodology

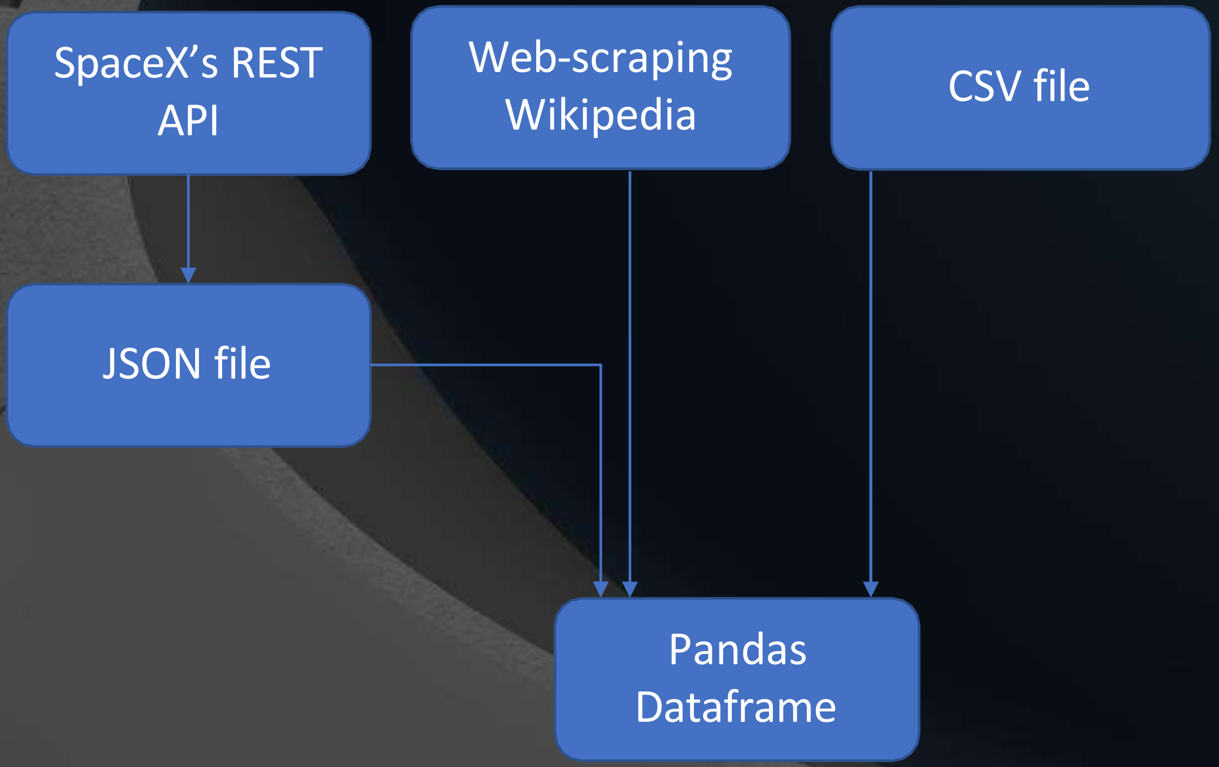
Methodology

Executive Summary

- Data collection methodology:
  - SpaceX REST API;
  - Web-scraping from SpaceX’s Wikipedia page dedicated to Falcon 9 and Falcon Heavy launches;
  - A CSV file available online.
- Perform data wrangling:
  - Data wrangling / preprocessing is done using Python’s Pandas library.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models:
  - Logistic Regression, Support Vector Machine, Decision Tree, K-Nearest Neighbors are the machine learning models used for classification and prediction.

Data Collection

The data for this project was collected using an API, Web-scraping technique and a downloaded CSV file available online.



SpaceX’s REST API

Web-scraping Wikipedia

CSV file

JSON file

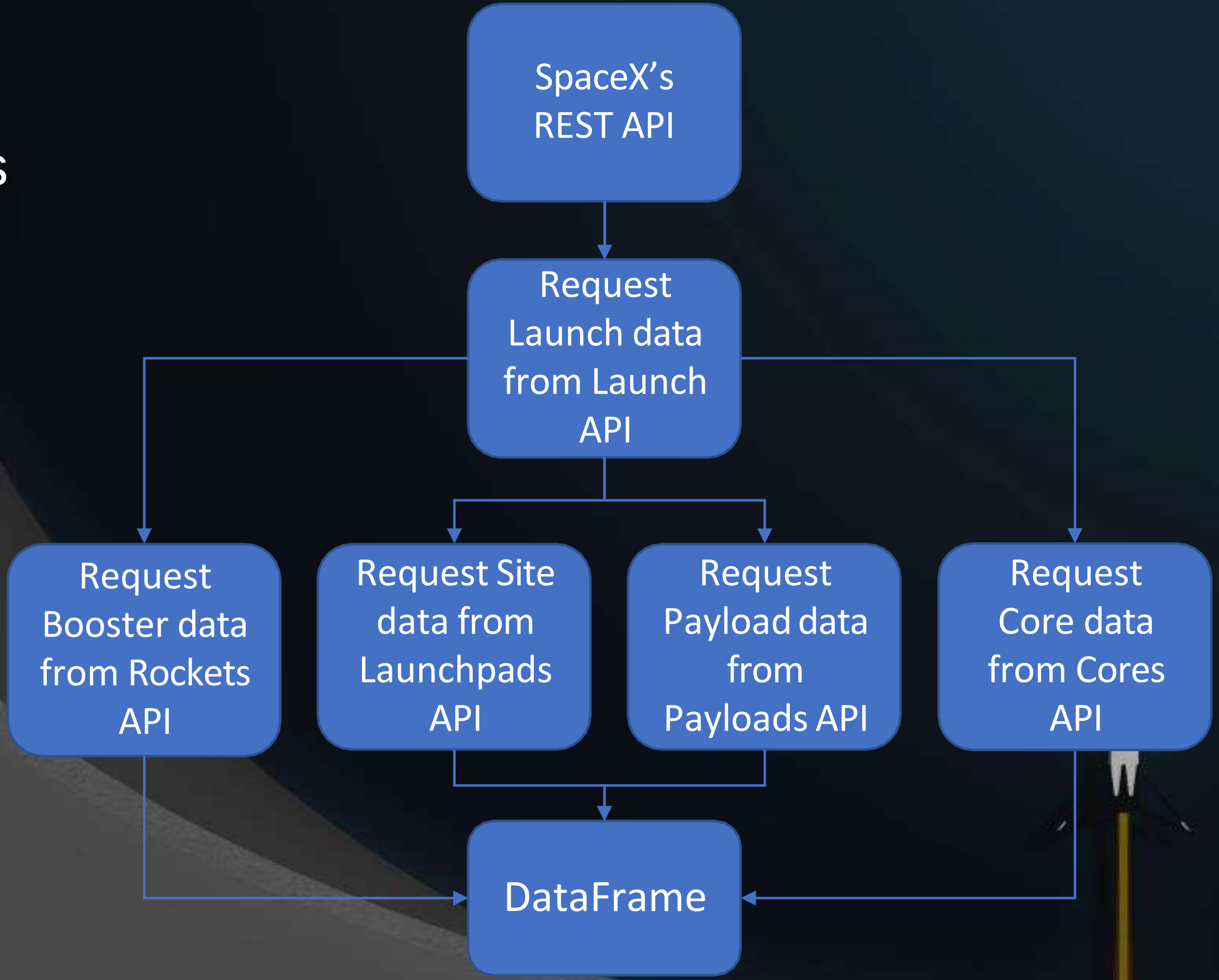
Pandas Dataframe

Data Collection – SpaceX API

This flowchart represents requests from SpaceX's various APIs to access the appropriate data for further preprocessing.

Link to the file on GitHub:

- https://bit.ly/31229Vg

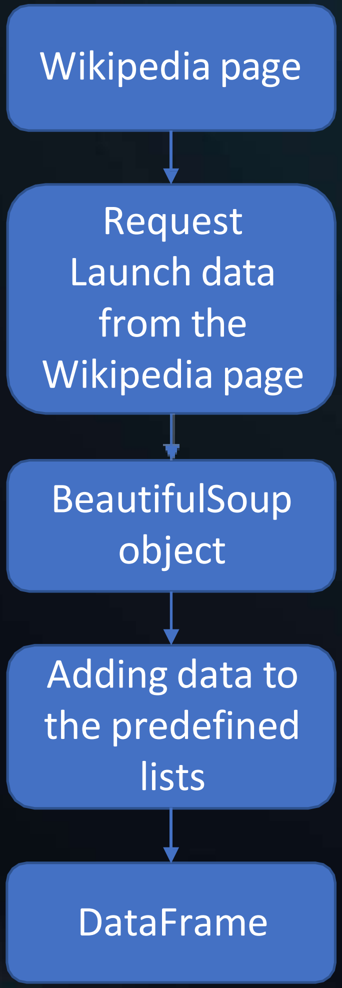

SpaceX's REST API
Request Launch data from Launch API
Request Booster data from Rockets API
Request Site data from Launchpads API
Request Payload data from Payloads API
Request Core data from Cores API
DataFrame

Data Collection - Scraping

This flowchart represents the process of Web-scraping from SpaceX’s Wikipedia page dedicated to the Falcon 9 and Falcon Heavy launches.

Link to the file on GitHub:

- https://bit.ly/3lbCSix

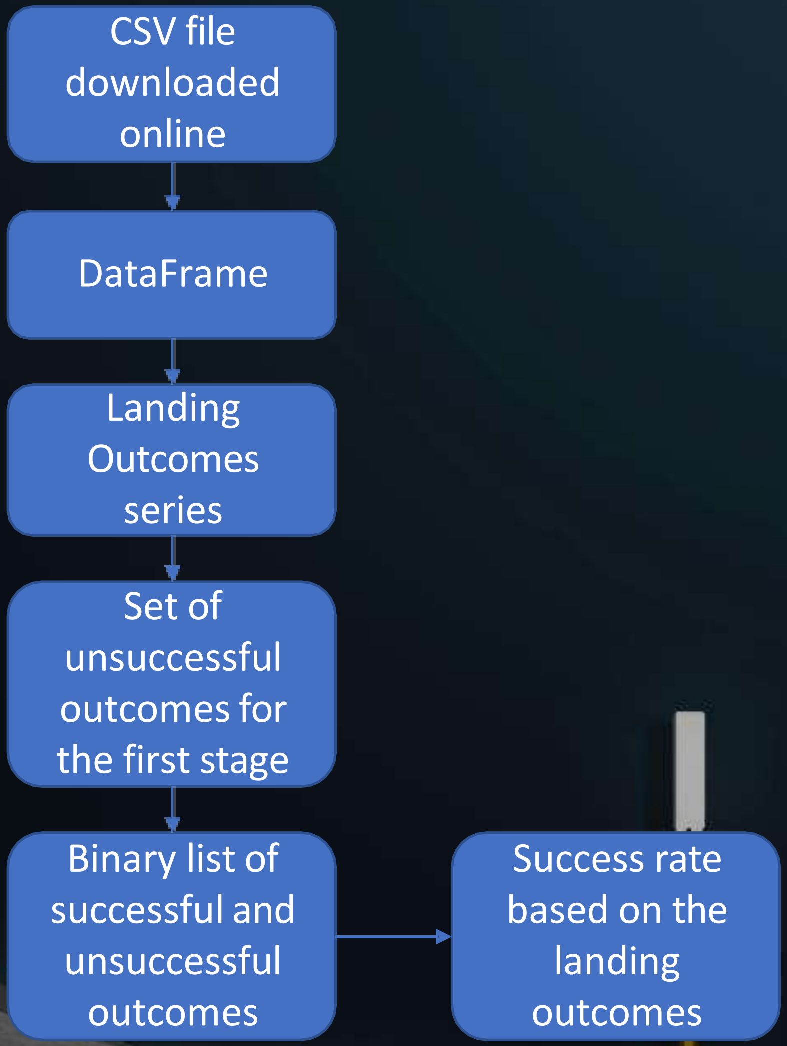

Wikipedia page
Request Launch data from the Wikipedia page
BeautifulSoup object
Adding data to the predefined lists
DataFrame

Data Wrangling

- The data wrangling/preprocessing was done using Pandas library;
- First we got a Series object that includes unique outcome names and the respective quantities from the data frame;
- Then we created determined we divided the outcomes into successful and unsuccessful;
- Then we created a new column in our data frame and added the divided successful and unsuccessful outcomes in a binary format;
- Then we calculated the success rate through finding the average of all the outcome values.

Link to the file on GitHub:

- https://bit.ly/3nToPzY


CSV file downloaded online

DataFrame

Landing Outcomes series

Set of unsuccessful outcomes for the first stage

Binary list of successful and unsuccessful outcomes

Success rate based on the landing outcomes

EDA with Data Visualization

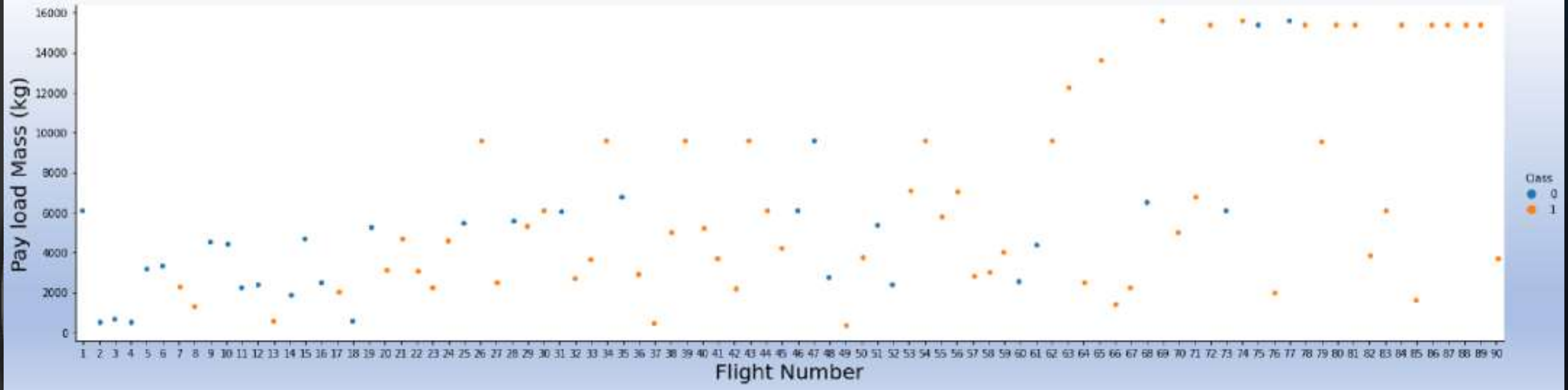
In the Exploratory Data Analysis stage of this project there were several types of plots used to get insights from the data:

- Seaborn Catplot (seaborn.catplot());
- Matplotlib Pyplot Plot (matplotlib.pyplot.plot());
- Seaborn Lineplot (seaborn.lineplot()).

Link to the file on GitHub:

- https://bit.ly/3p1CCDL

As an example, here is Seaborn Catplot (scatter) plot that is used to identify the relationship between FlightNumber and PayloadMass variables:



EDA with SQL

SQL queries used to gain insight from the data for Exploratory Data Analysis:

- Selected unique Launch Site values;
- Selected 5 rows in which the values of the Launch Site column start with ‘CCA’ letters;
- Calculated the total payload mass (kg) carried for the ‘NASA (CRS)’ customer;
- Calculated the average payload mass (kg) carried by the ‘F9 v1.1’ booster version;
- Selected the date when SpaceX successfully landed their first stage booster on a ground pad;
- Selected unique mission outcome names with respective quantities.

Link to the file on GitHub:

https://bit.ly/3xoVK2s

Build an Interactive Map with Folium

Features used to getter a better view of the launch sites' location on the Folium road-map:

- Circle – to see the location of the object and it's surroundings;
- Marker – to indicate the name of the object on the map;
- Mouse Positon Coordinates – to find the location of the mouse on the map;
- Line – to calculate the distance from the main object to some point on the map

Link to the file on GitHub:

- https://bit.ly/3xApedS

Build a Dashboard with Plotly Dash

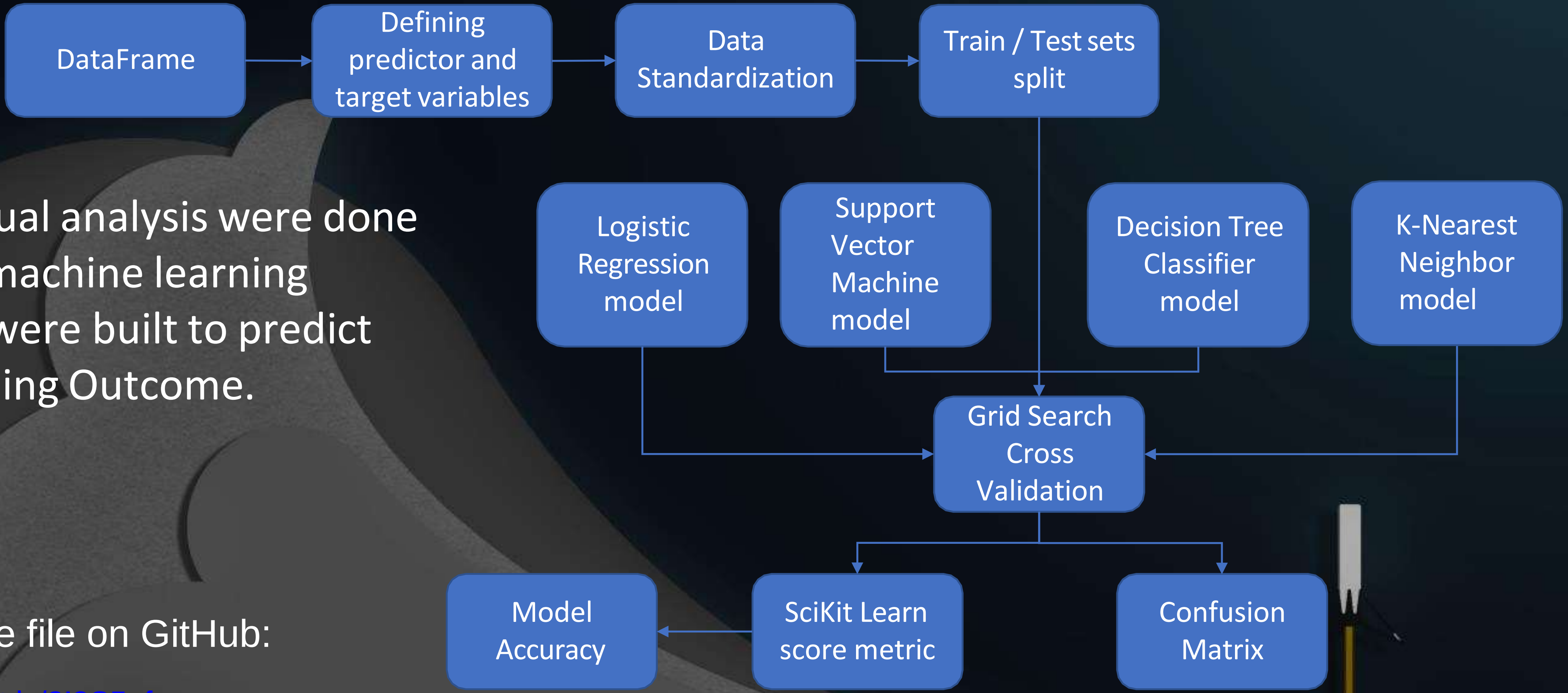
Plotly Dash tools used for creating an interactive dashboard consisting of plots and interaction tools like:

- Dropdown Box – for choosing one or all of the Launch Sites to be depicted on the Piechart;
- Piechart - depicting the Launch Sites in a form of proportional sectors and corresponding number of Launches;
- Range Slider – for choosing the range of the Payload Mass (kg) to be depicted on the Scatter Plot;
- Scatter Plot – depicting the relationship between the Payload Mass (kg) and Classes of the rockets;

Link to the file on GitHub:

- https://bit.ly/3xw4fZt

Predictive Analysis (Classification)


DataFrame
Defining predictor and target variables
Data Standardization
Train / Test sets split
Logistic Regression model
Support Vector Machine model
Decision Tree Classifier model
K-Nearest Neighbor model
Grid Search Cross Validation
Model Accuracy
SciKit Learn score metric
Confusion Matrix

After visual analysis were done several machine learning models were built to predict the Landing Outcome.

Link to the file on GitHub:

- https://bit.ly/3I8S5ef

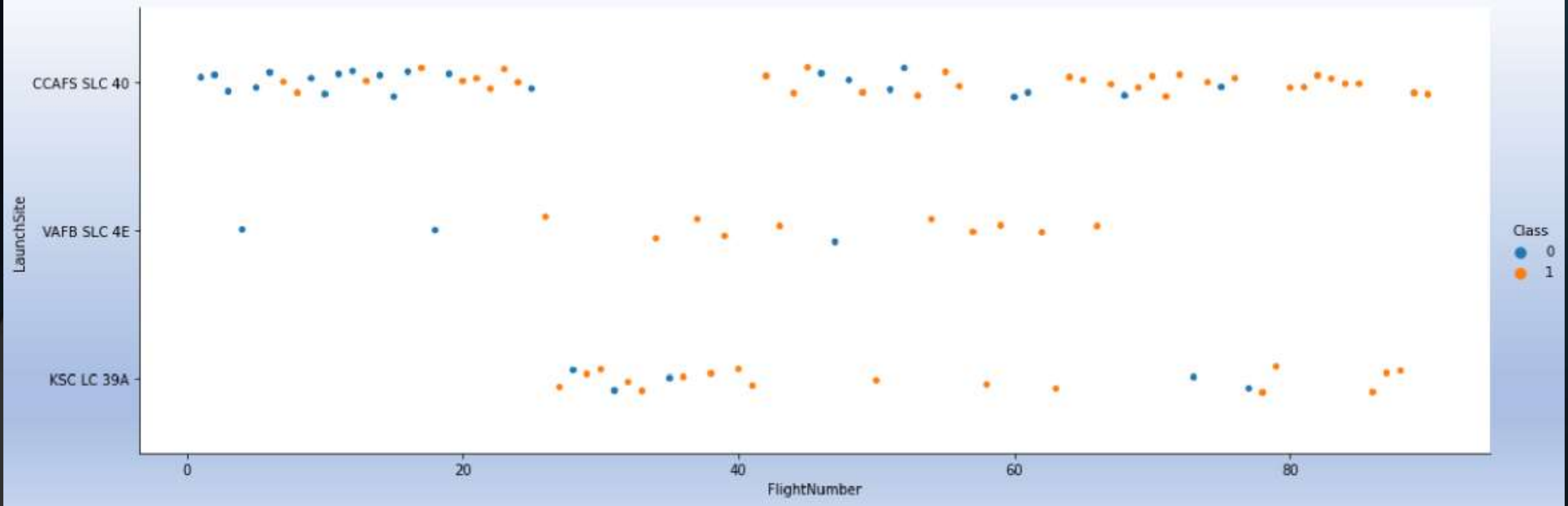
Results

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

Section 2

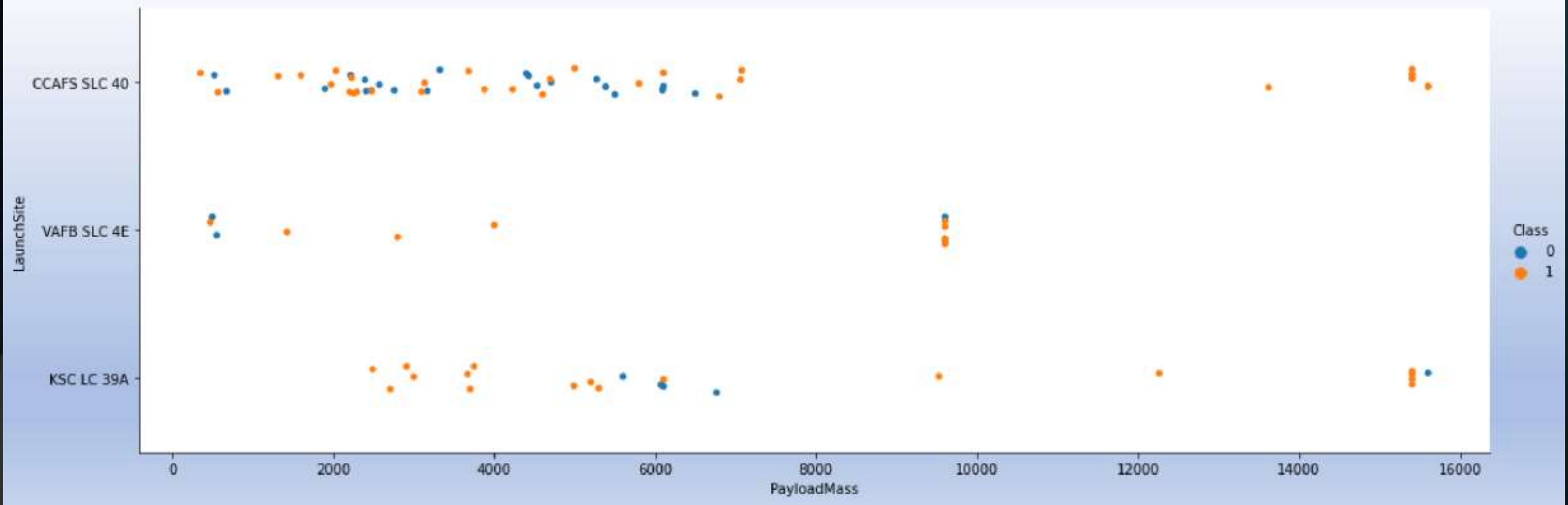
Insights drawn from EDA

Flight Number vs. Launch Site



Observing the plot above that depicts the relationship between the ‘Flight Number’ and ‘Launch Site’ variables we can conclude that the landing outcomes of the missions that took place on ‘CCAFS SLC 40’ and ‘VAFB SLC 4E’ sites have improved as the number of flights increased. Hence the probability of future mission landing outcomes being successful also increases.

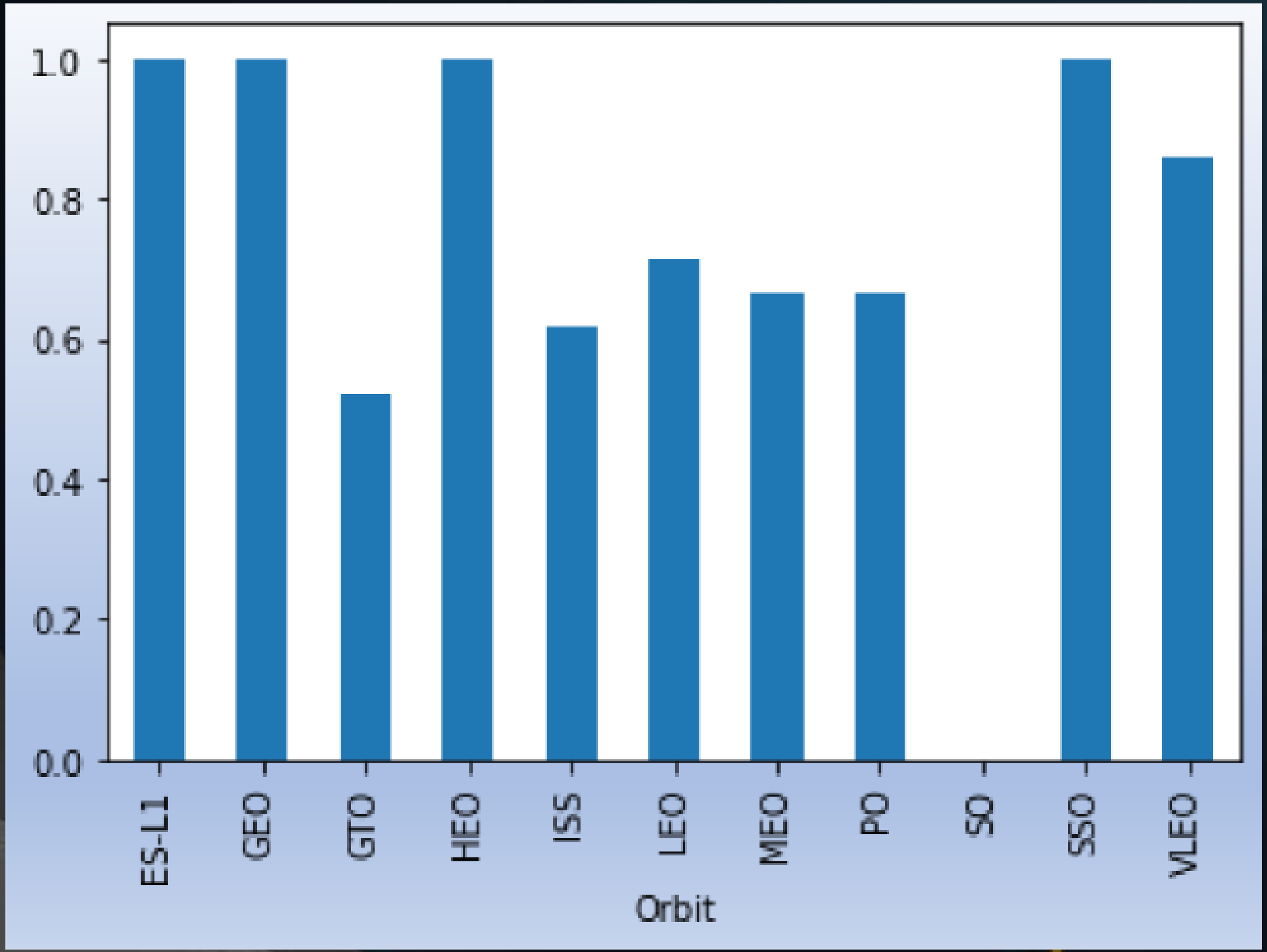
Payload vs. Launch Site



- The plot above illustrates that the launches with the largest Payload Mass (14000 – 16000 kg) have taken off of ‘CCAFS SLC 40’ and ‘VAFB SLC 4E’ launch sites.
- We can also see that the landing outcomes of the missions with Payload Mass above 8000 kg are mostly successful.
- It is necessary to note that in missions where the Payload Mass is below 8000 kg the landing outcome is more likely to be successful for the launch site ‘KSC LC 39A’

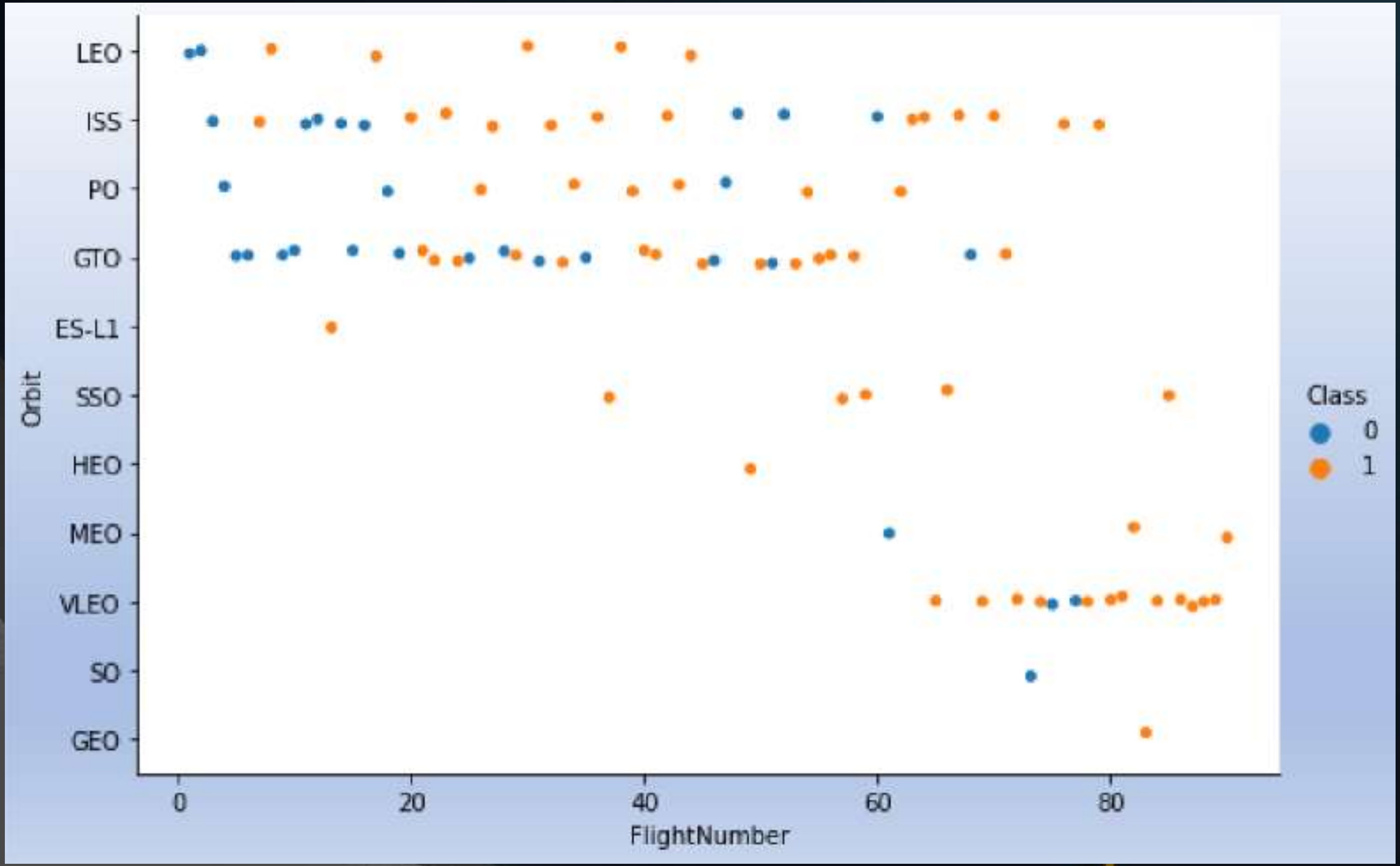
Success Rate vs. Orbit Type

- On the bar chart to the right we can see that the orbit types with the highest Success Rate (100%) of landing outcomes are ‘ES-L1’, ‘GEO’, ‘HEO’ and ‘SSO’.
- The lowest Success Rate of landing outcomes on this chart is for SO (Sub-Orbital) orbit type. This is because the goal of this mission was to test the first Crew Dragon (C205) with in-flight abort system. The first stage was destroyed after Crew Dragon detached, which was expected. This same first stage booster has completed 4 successful landings before this mission.

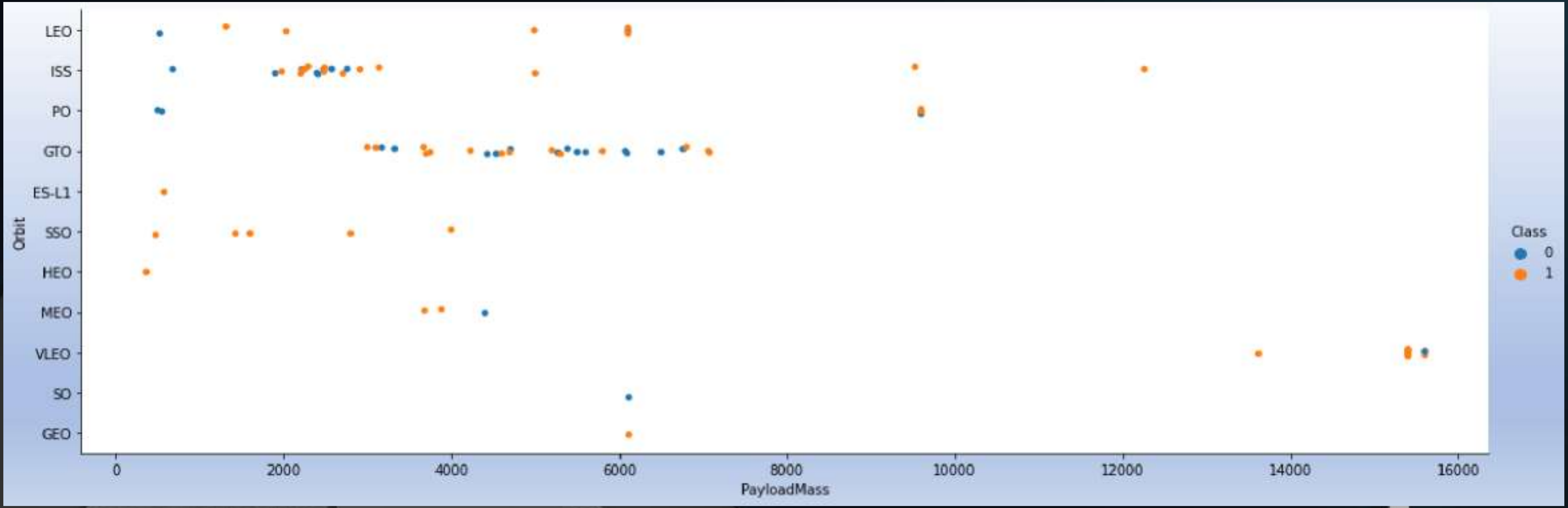


Flight Number vs. Orbit Type

- On this scatter plot we can see that the Success Rates of landing outcomes for ‘LEO’, ‘SSO’ and ‘VLEO’ orbit types are relatively higher and more consistent.
- It is worth noting that as the number of flights of the company increased the VLEO (Very Low Earth Orbit) orbit type has been used more frequently for the missions. This is because SpaceX has started deploying their own Starlink internet satellites to this orbit.



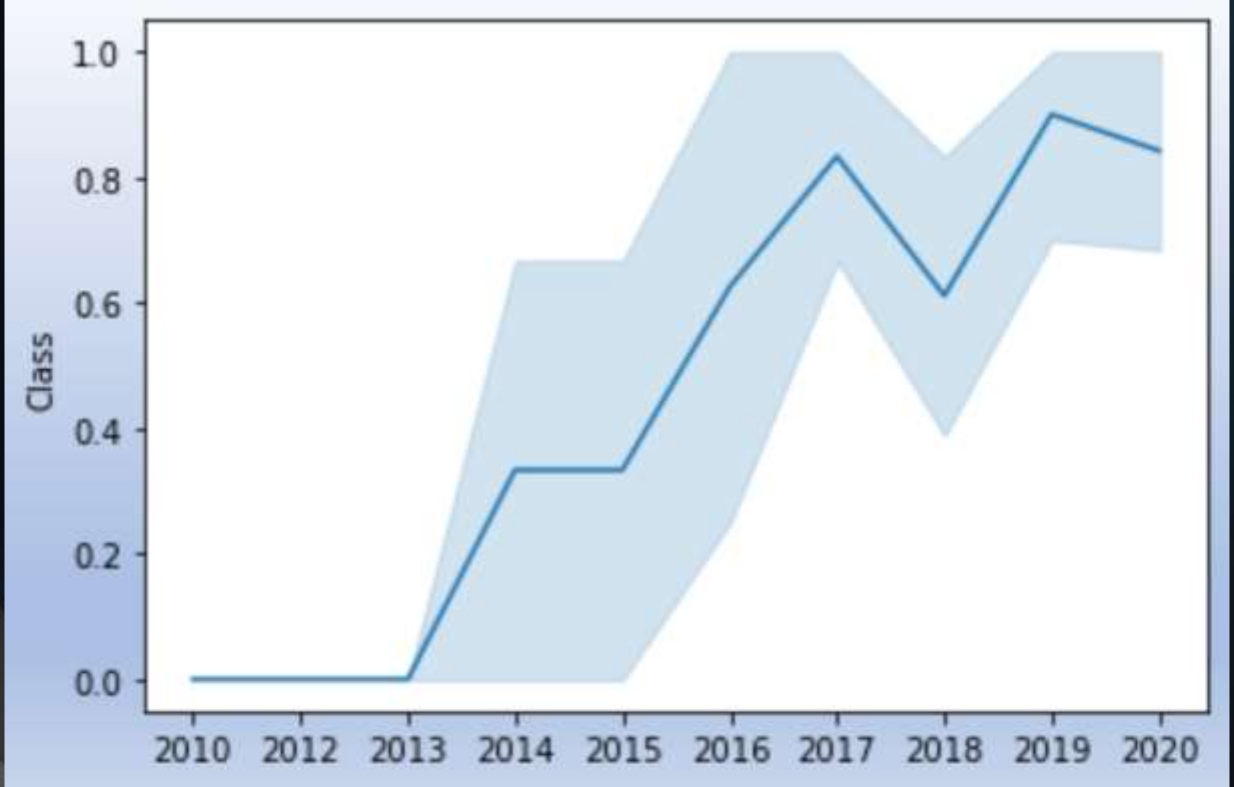
Payload vs. Orbit Type



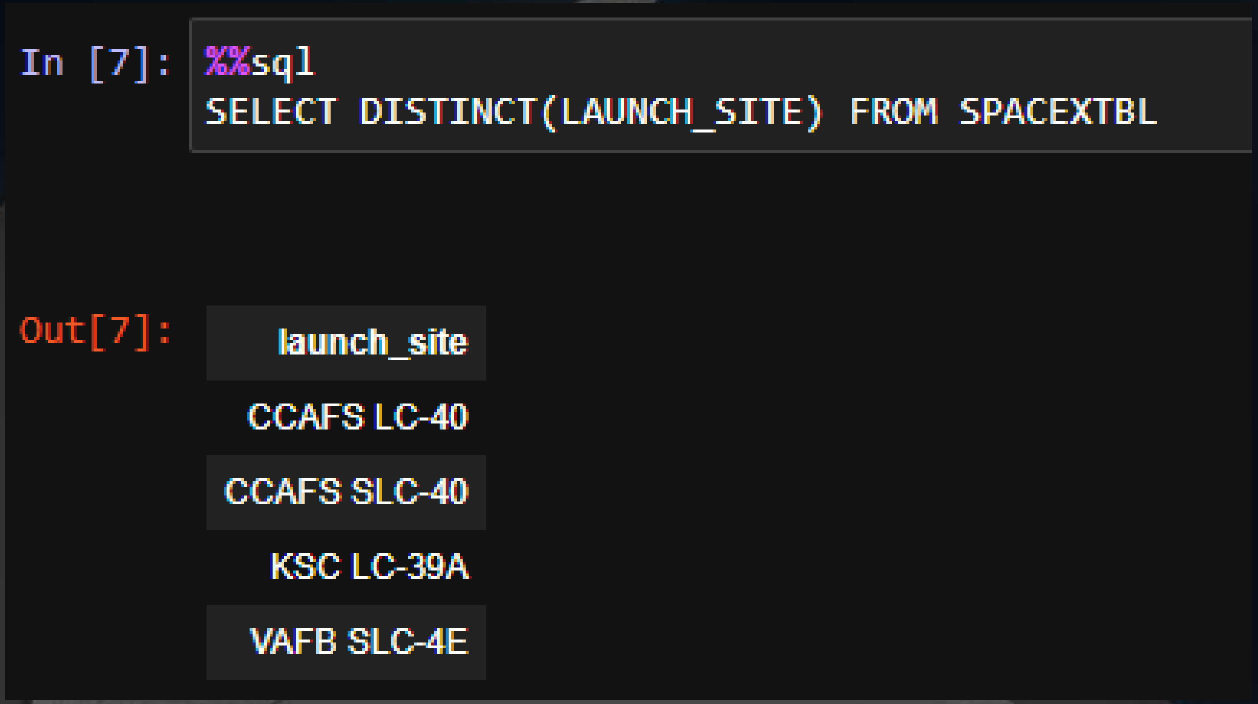
On this scatter plot we can see that the Success Rate for the ‘LEO’ and ‘SSO’ orbit types is relatively higher with launch Payload Mass below 8000 kg, although most of the launches are aimed toward ‘ISS’ and ‘GTO’ obit types

Launch Success Yearly Trend

On this line graph we can see that from 2013 to 2020 SpaceX has significantly improved in terms of Success Rate of landing outcomes. It took only 7 years for SpaceX to go from 0% to ≈85% Success Rate of landing outcomes, which is extremely impressive.



All Launch Site Names

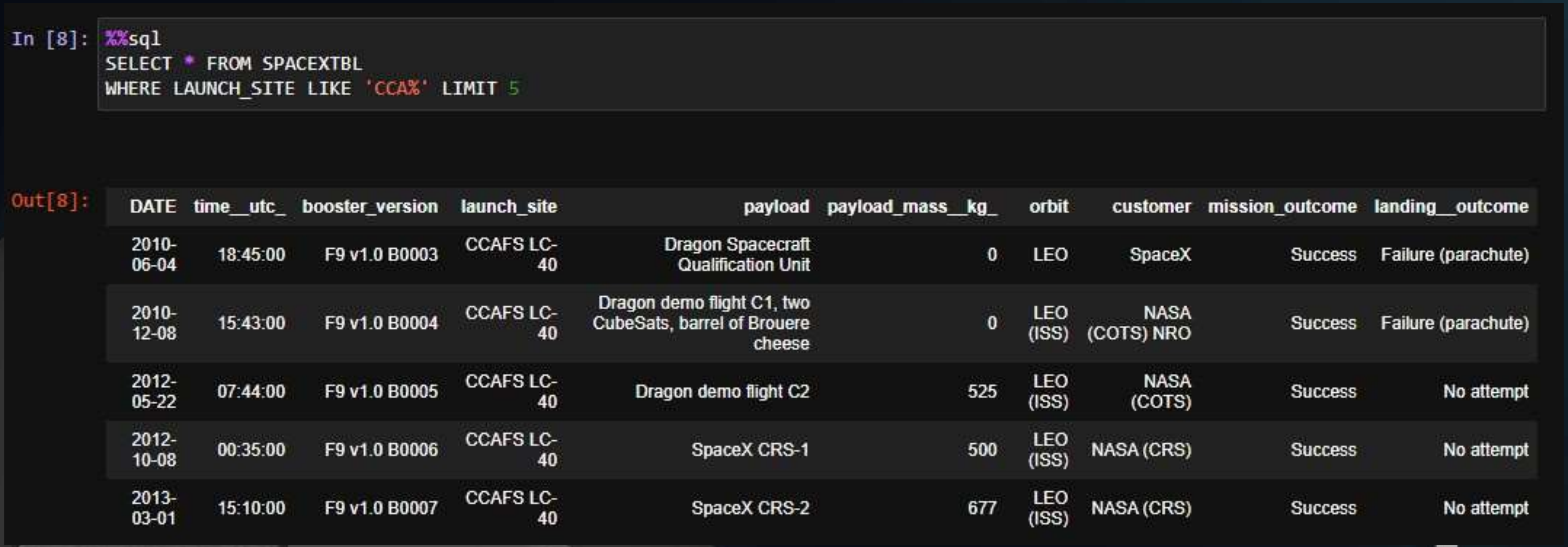


```
In [7]: %%sql
        SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL
```

```
Out[7]:  launch_site
         CCAFS LC-40
        CCAFS SLC-40
         KSC LC-39A
        VAFB SLC-4E
```

Selecting the rows using ‘DISTINCT()’ clause to return all unique values from the ‘LAUNCH_SITE’ column of the ‘SPACEXTBL’ table

Launch Site Names Begin with 'CCA'



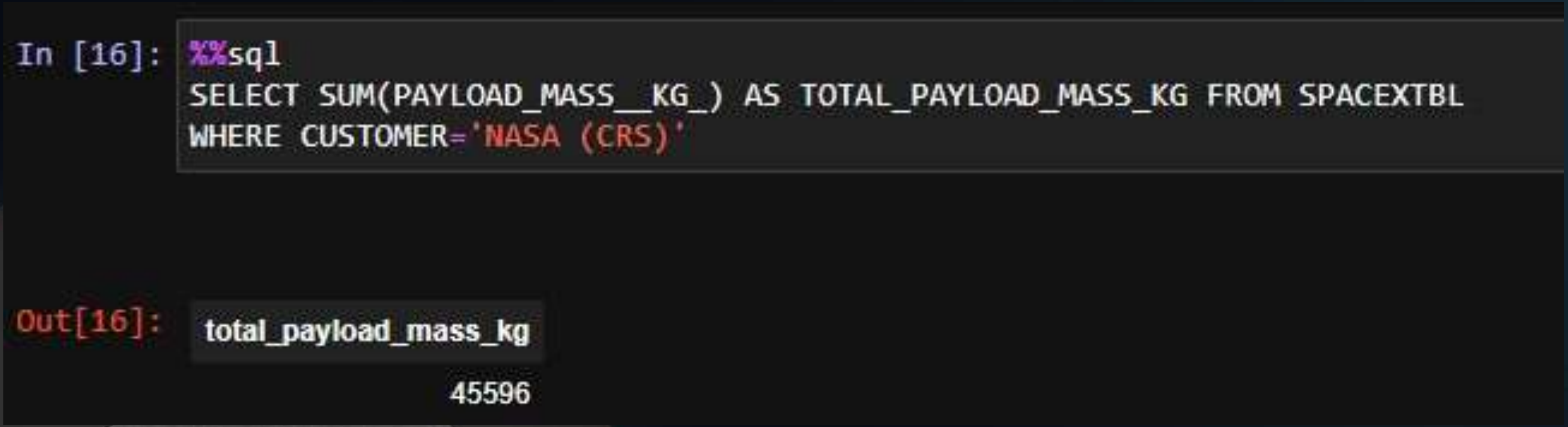
```
In [8]: %%sql
        SELECT * FROM SPACEXTBL
        WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

Out[8]:

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

- Using ‘*’ sign gets all the rows from the ‘SPACEXTBL’;
- We use ‘WHERE’ clause and ‘LIKE’ operator to filter out the result to get only rows that have ‘LAUNCH_SITE’ values starting with “CCA” letters;
- We use ‘LIMIT’ clause to limit the number of returned rows to 5

Total Payload Mass

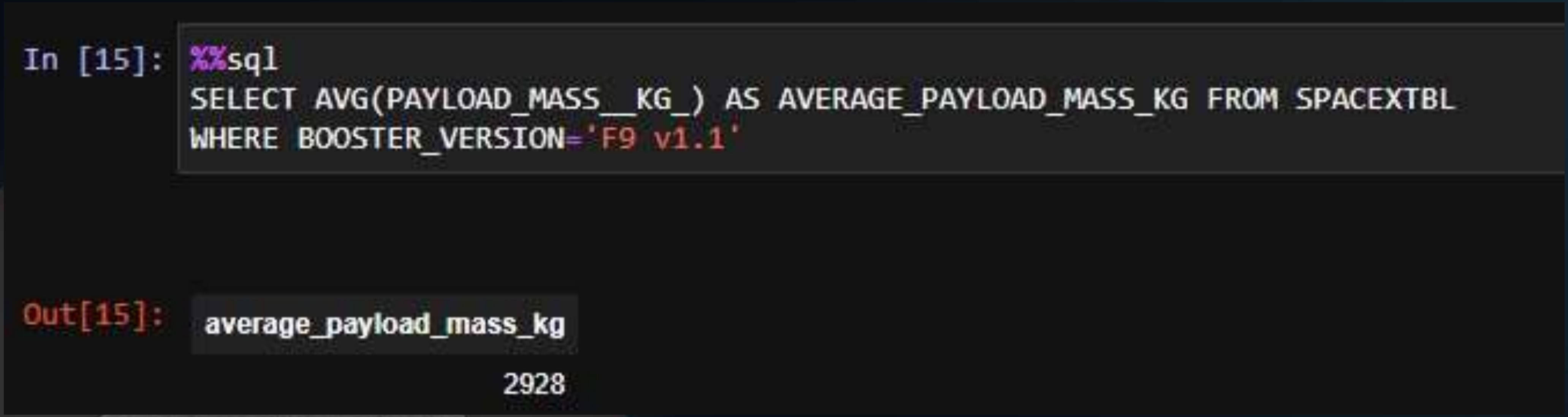


```
In [16]: %%sql
         SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD_MASS_KG FROM SPACEXTBL
         WHERE CUSTOMER='NASA (CRS)'
```

```
Out[16]: total_payload_mass_kg
                         45596
```

- Using ‘SUM()’ aggregate function to calculate the total of the values in the ‘PAYLOAD_MASS__KG_’ column of the ‘SPACEXTBL’ table
- Using ‘AS’ command to create a ‘TOTAL_PAYLOAD_MASS_KG’ alias for the new column
- Using ‘WHERE’ clause with ‘=‘ operator to filter out the result for the calculation of the values where ‘CUSTOMER’ column values are “NASA (CRS)”

Average Payload Mass by F9 v1.1

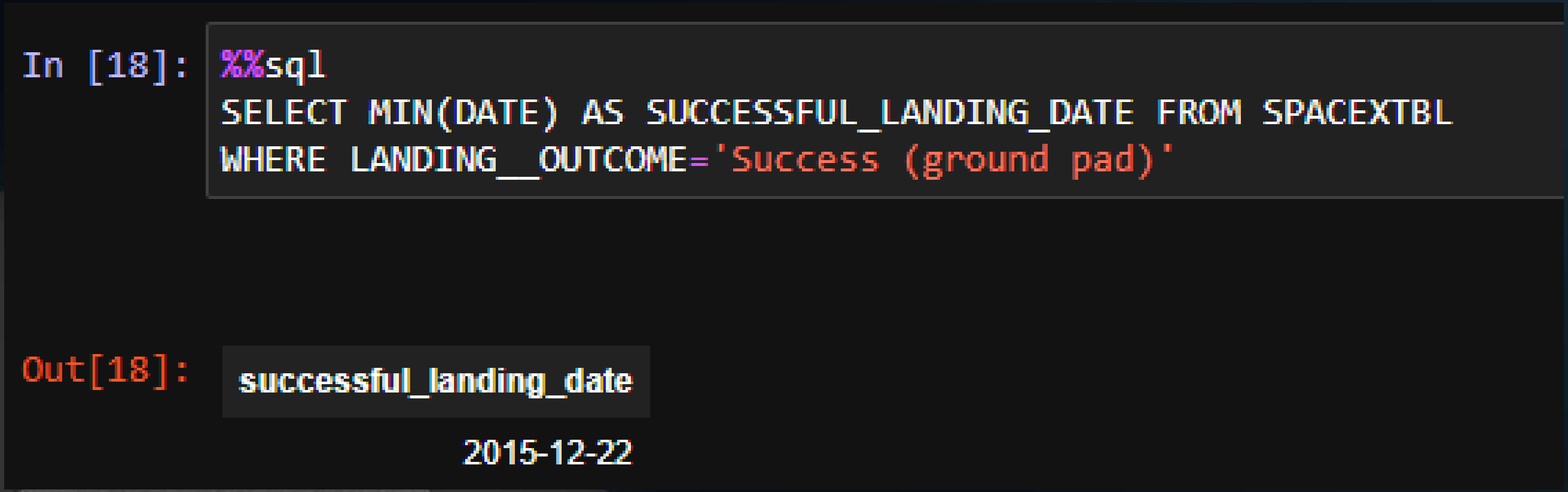


```
In [15]: %%sql
         SELECT AVG(PAYLOAD_MASS__KG_) AS AVERAGE_PAYLOAD_MASS_KG FROM SPACEXTBL
         WHERE BOOSTER_VERSION='F9 v1.1'

Out[15]: average_payload_mass_kg
                            2928
```

- Using ‘AVG()’ function to calculate the average of the values in the ‘PAYLOAD_MASS__KG_’ column of the ‘SPACEXTBL’ table
- Using ‘AS’ command to create a ‘AVERAGE_PAYLOAD_MASS_KG’ alias for the new column
- Using ‘WHERE’ clause with ‘=‘ operator to filter out the result for the calculation of the values where ‘BOOSTER_VERSION’ column values are “F9 v1.1”

First Successful Ground Landing Date

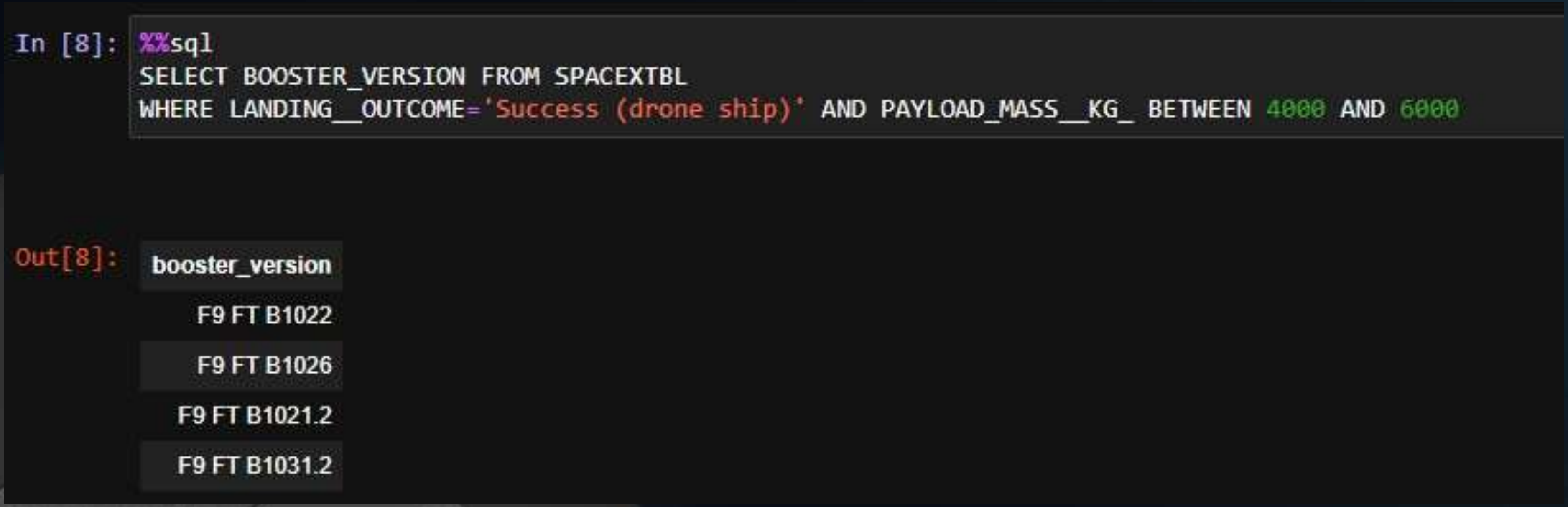


```
In [18]: %%sql
         SELECT MIN(DATE) AS SUCCESSFUL_LANDING_DATE FROM SPACEXTBL
         WHERE LANDING__OUTCOME='Success (ground pad)'
```

```
Out[18]: successful_landing_date
                      2015-12-22
```

- Using ‘MIN()’ function to the oldest date from the ‘DATE’ column of the ‘SPACEXTBL’ table
- Using ‘AS’ command to create a ‘SUCCESSFUL_LANDING_DATE’ alias for the new column
- Using ‘WHERE’ clause with ‘=‘ operator to filter out the result for the values where ‘LANDING_OUTCOME’ column values are “Success (ground pad)”

Successful Drone Ship Landing with Payload between 4000 and 6000



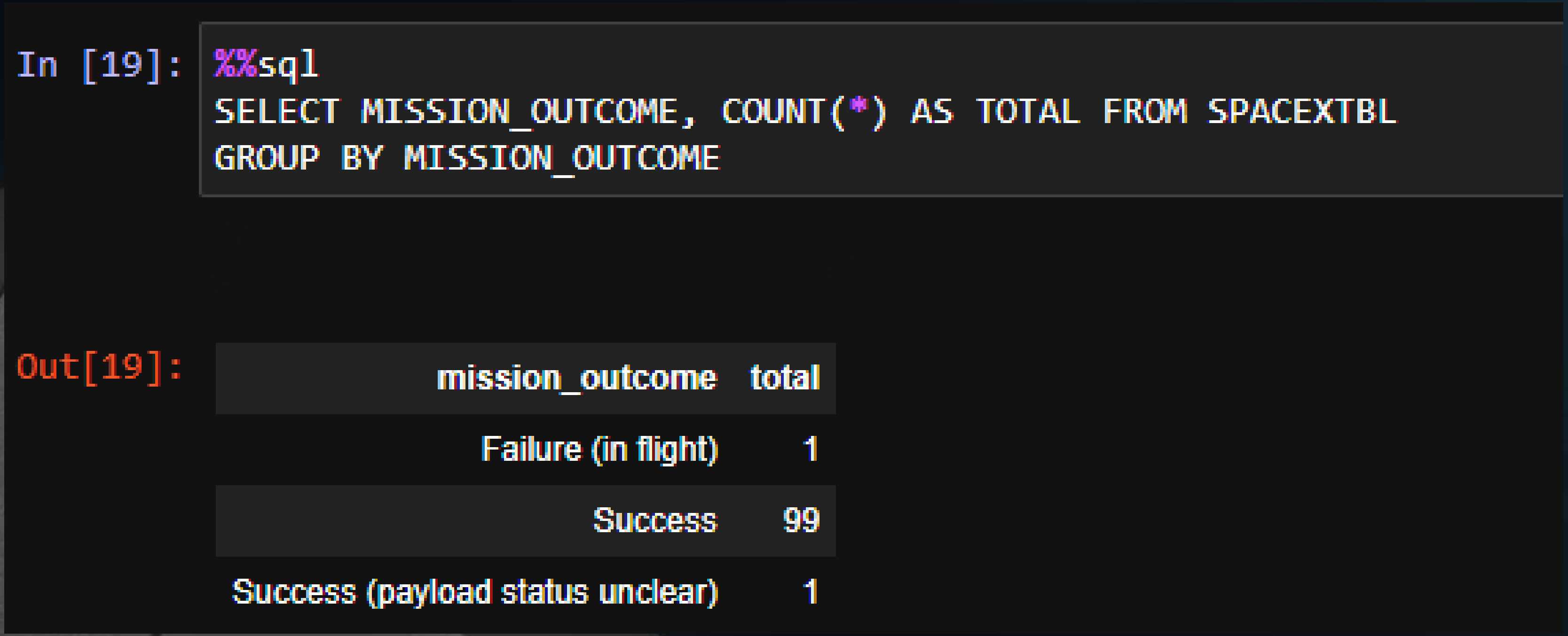
In [8]:
```
%%sql
SELECT BOOSTER_VERSION FROM SPACEXTBL
WHERE LANDING__OUTCOME='Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

Out[8]:

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

- Selecting values from the ‘BOOSTER_VERSION’ column of the ‘SPACEXTBL’ table
- Using ‘WHERE’ clause with ‘=‘, ‘AND’ and ‘BETWEEN’ operators to filter out the result for the values where ‘LANDING_OUTCOME’ column values are “Success (drone ship)” and ‘PAYLOAD_MASS__KG_’ values are between ‘4000’ and ‘6000’

Total Number of Successful and Failure Mission Outcomes



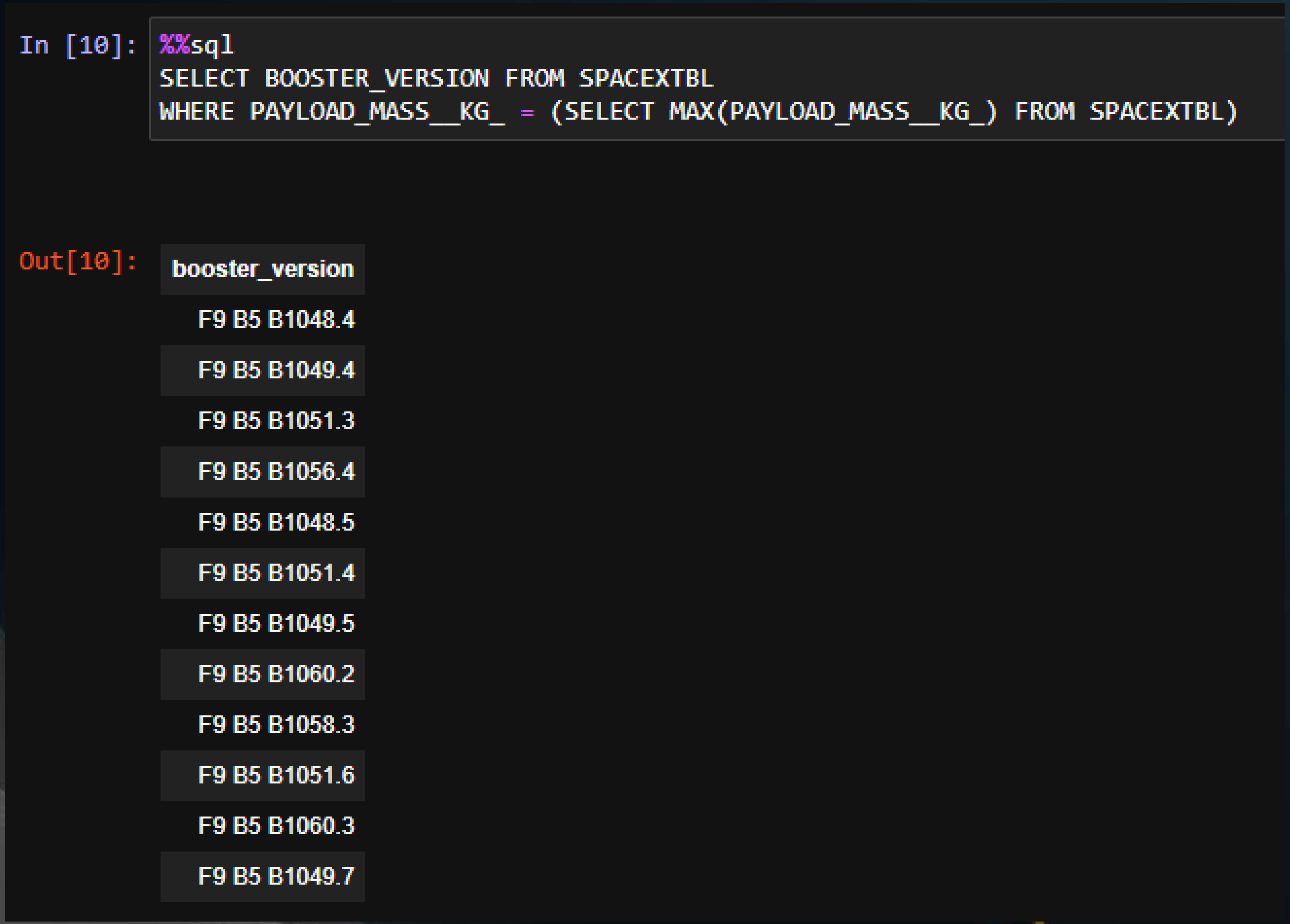
```
In [19]: %%sql
         SELECT MISSION_OUTCOME, COUNT(*) AS TOTAL FROM SPACEXTBL
         GROUP BY MISSION_OUTCOME
```

```
Out[19]:
                  mission_outcome   total
                Failure (in flight)     1
                            Success    99
   Success (payload status unclear)     1
```

- Selecting ‘MISSION_OUTCOME’ values using ‘COUNT()’ and ‘*’ sign to calculate the number of missions in the ‘SPACEXTBL’ table
- Using ‘AS’ command to create a ‘TOTAL’ alias for the new column
- Using ‘GROUP BY’ clause to group the results by ‘MISSION_OUTCOME’ values

Boosters Carried Maximum Payload

- Selecting values from the ‘BOOSTER_VERSION’ column of the ‘SPACEXTBL’ table
- Using ‘WHERE’ clause with ‘=‘ operator to filter out the result for the values where ‘PAYLOAD_MASS__KG_’ column values are same as the largest value in this column selected using ‘MAX()’ function

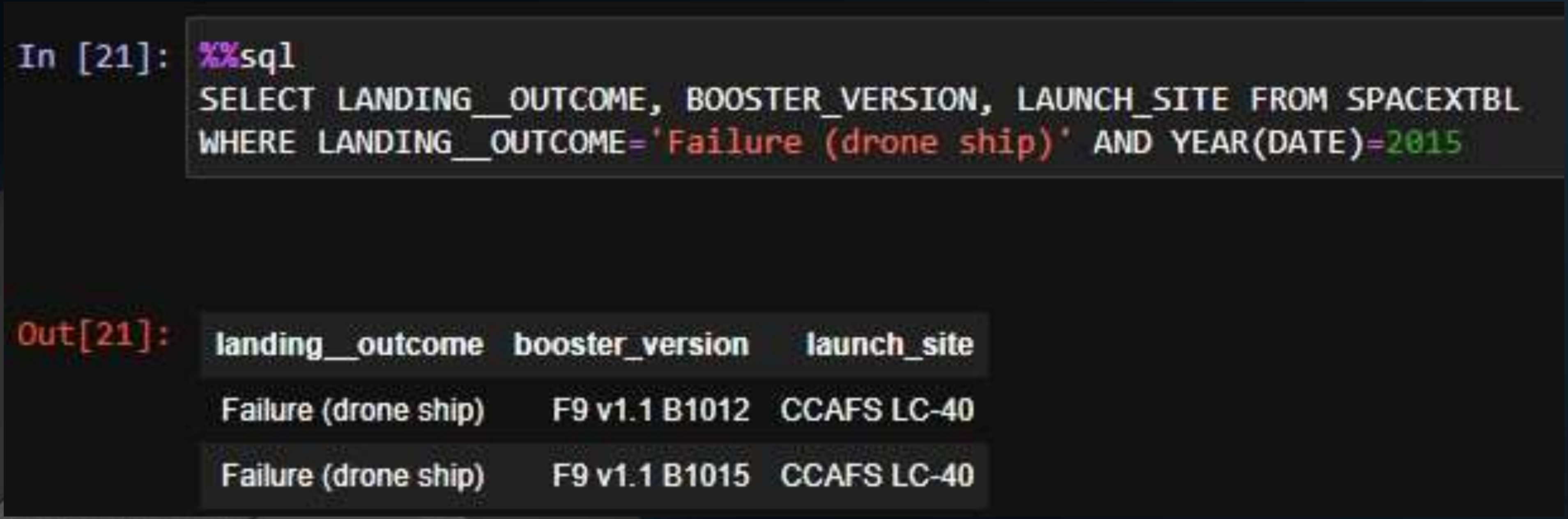


```
In [10]: %%sql
         SELECT BOOSTER_VERSION FROM SPACEXTBL
         WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

Out[10]:

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



In [21]:

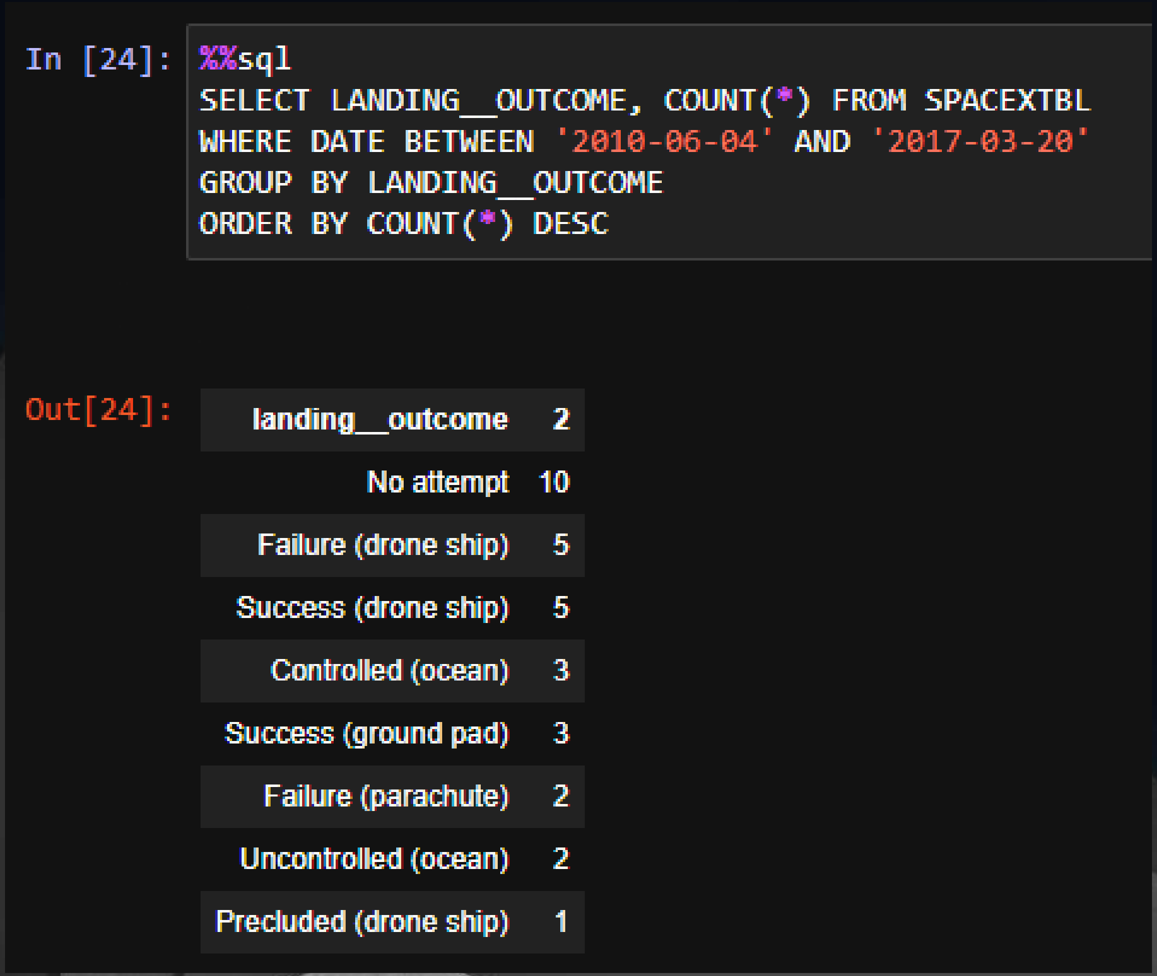
```
%%sql
SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL
WHERE LANDING__OUTCOME='Failure (drone ship)' AND YEAR(DATE)=2015
```

Out[21]:

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- Selecting values from the ‘LANDING_OUTCOME’, ‘BOOSTER_VERSION’ and ‘LAUNCH_SITE’ columns of the ‘SPACEXTBL’ table
- Using ‘WHERE’ clause with ‘=‘, ‘AND’ operator and YEAR() function to filter out the results where ‘LANDING_OUTCOME’ column values are “Failure (drone ship)” and the year values of the ‘DATE’ column are 2015

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



In [24]:

```
%%sql
SELECT LANDING__OUTCOME, COUNT(*) FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING__OUTCOME
ORDER BY COUNT(*) DESC
```

Out[24]:

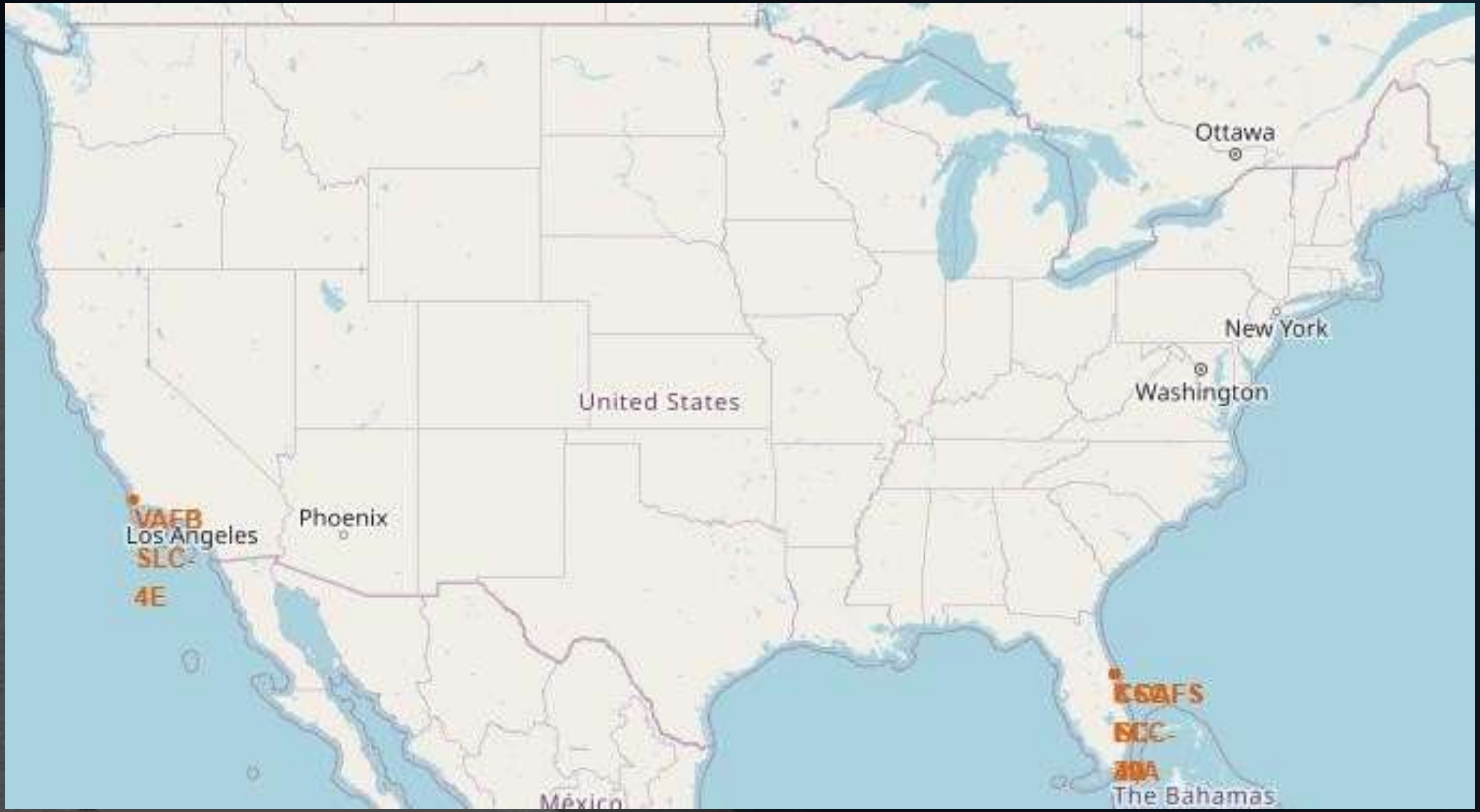
landing__outcome	2
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

- Selecting values from the ‘LANDING__OUTCOME’ and using ‘COUNT()’ function with ‘*’ sign to calculate the number of rows in the ‘SPACEXTBL’ table
- Using ‘WHERE’ clause with ‘=‘, ‘AND’ and ‘BETWEEN’ operators to filter out the result for the values where ‘DATE’ column values are between ‘2010-06-04’ and ‘2017-03-20’
- Using ‘GROUP BY’ clause to group the values of the ‘LANDING_OUTCOME’ column
- Using ‘ORDER BY’ clause to sort the values by the number of rows in each group

Section 4

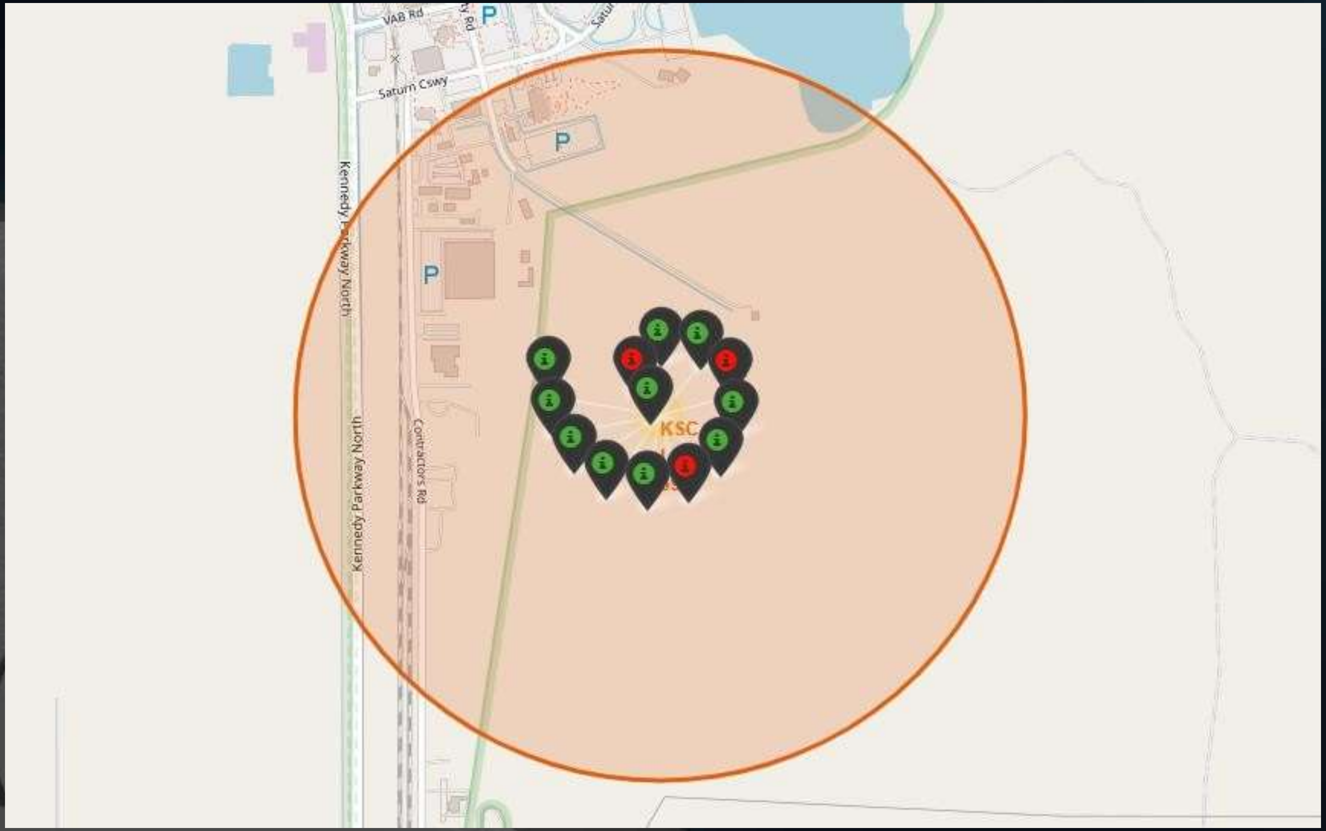
Launch Sites Proximities Analysis

Partial road-map of the USA with Launch Site locations



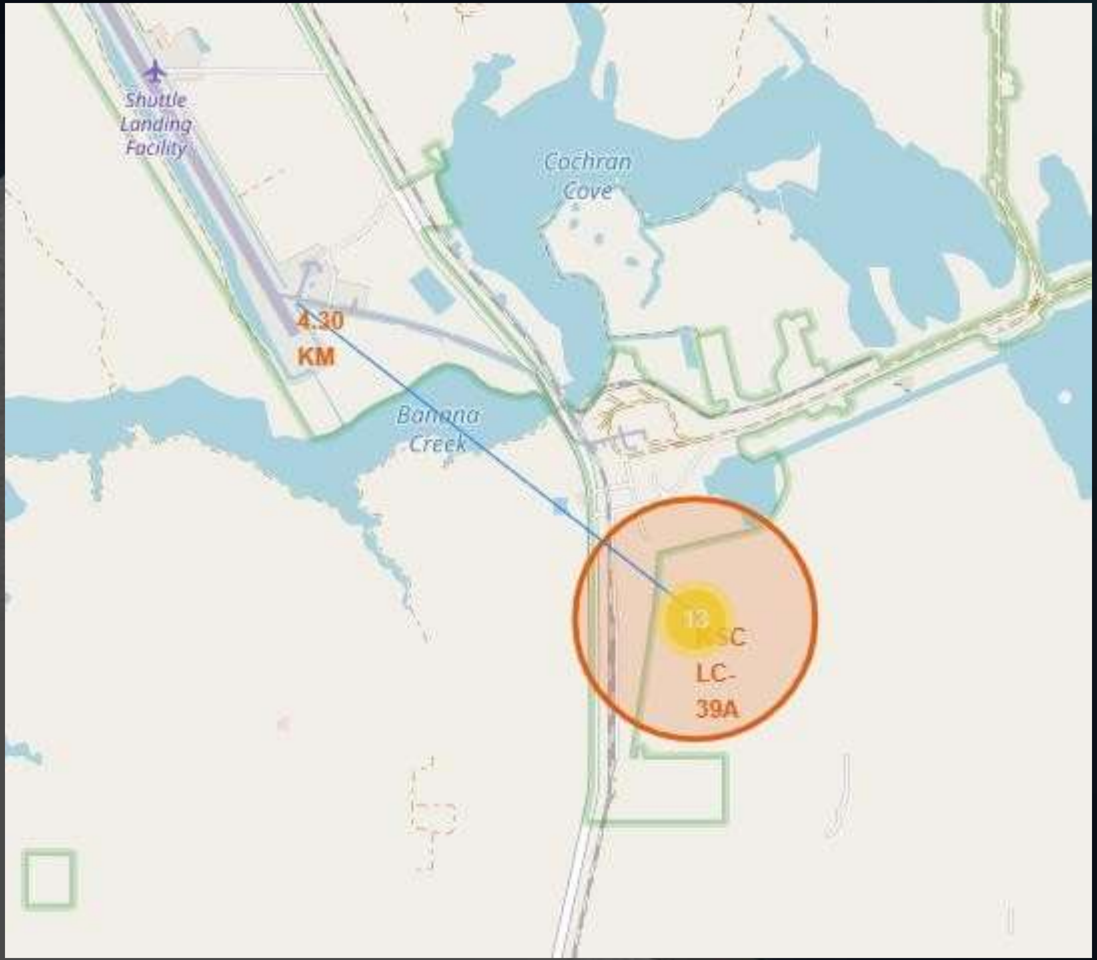
As you can see on this map, the ‘VAFB SLC-4E’ launch site is located in California. As the launch sites ‘KSC LC-39A’, ‘CCAFS SLC-40’ and ‘CCAFS LC-40’ are located close to each in Florida the location looks like one spot on this map.

Road-map with landing outcomes



This road-map depicts an area with ‘KSC LC-39A’ launch site located in the center and landing outcomes with green/red colored markers surrounding it

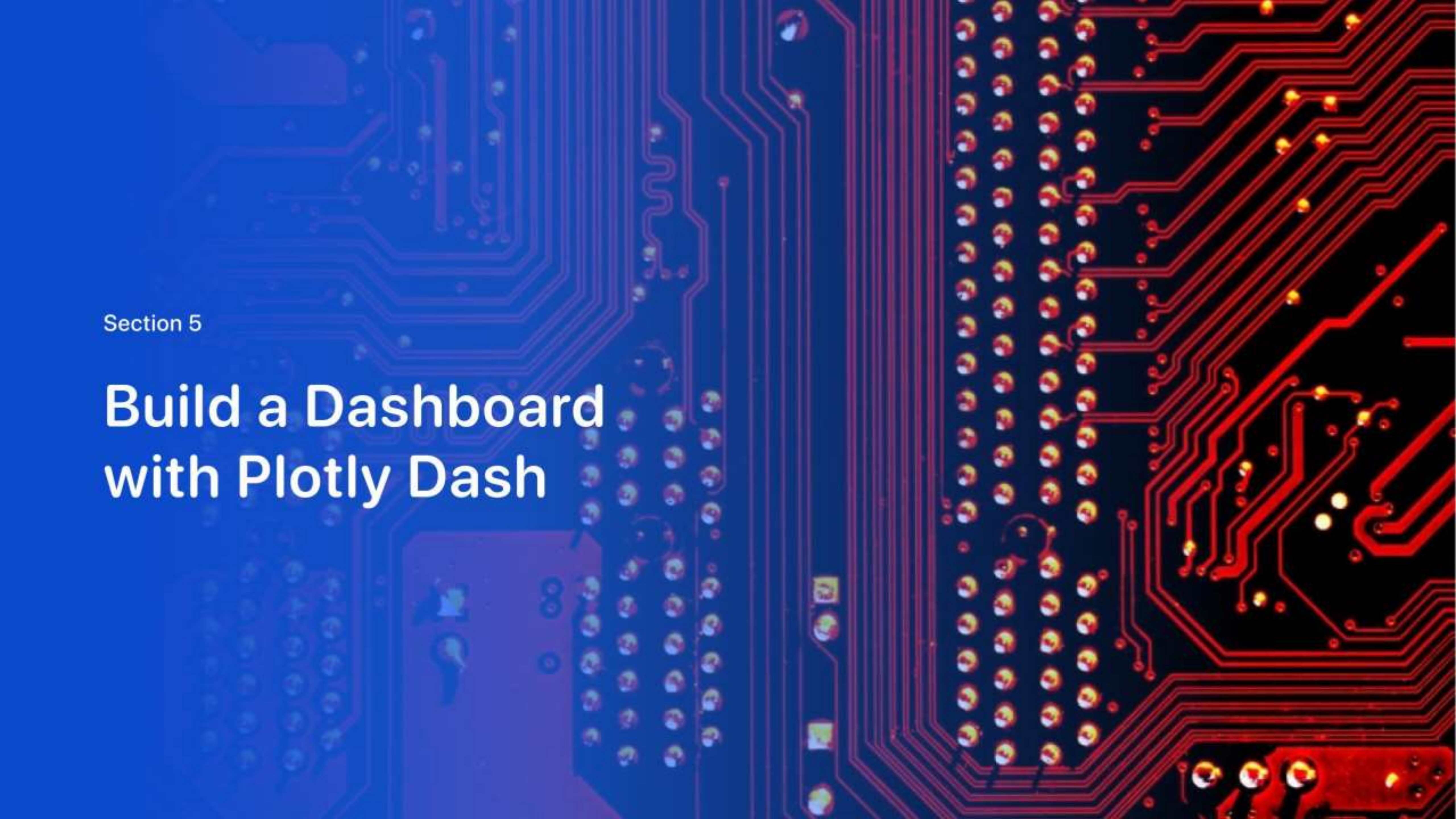
Road-map with calculated distance



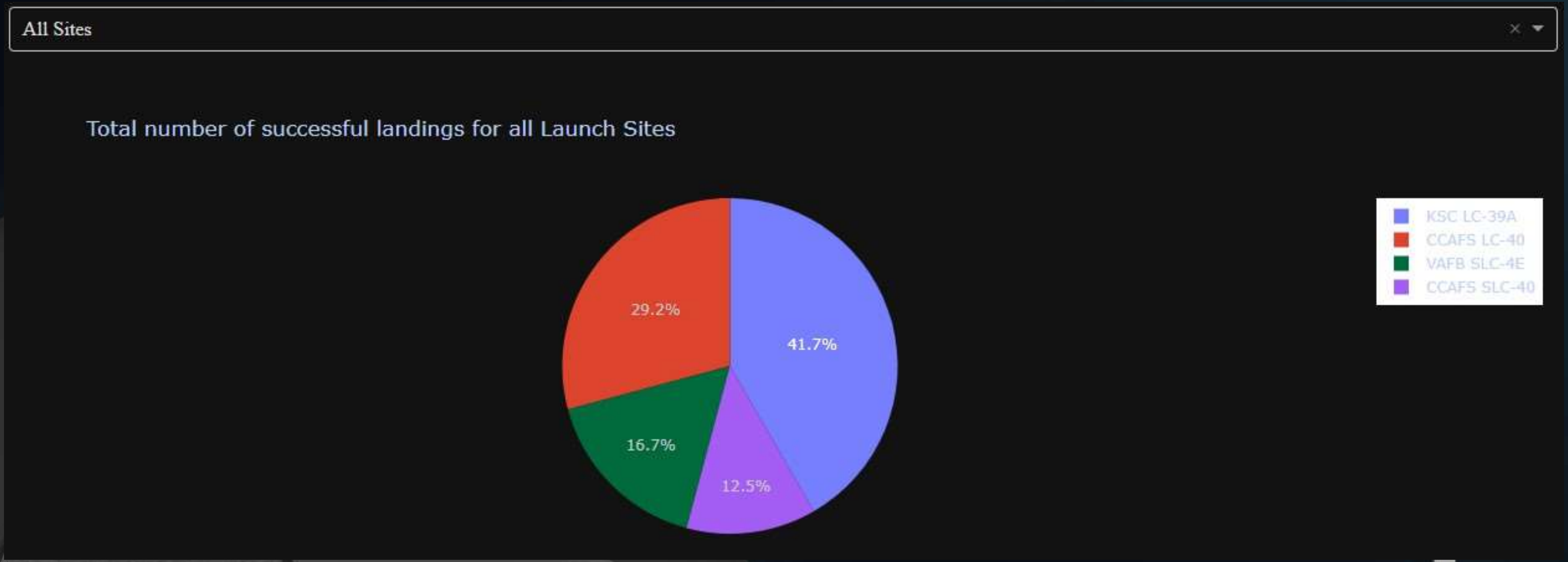
This road-map depicts the calculated distance (4.3 km) between ‘KSC LC-39A’ and ‘Shuttle Landing Facility’.

Section 5

Build a Dashboard with Plotly Dash



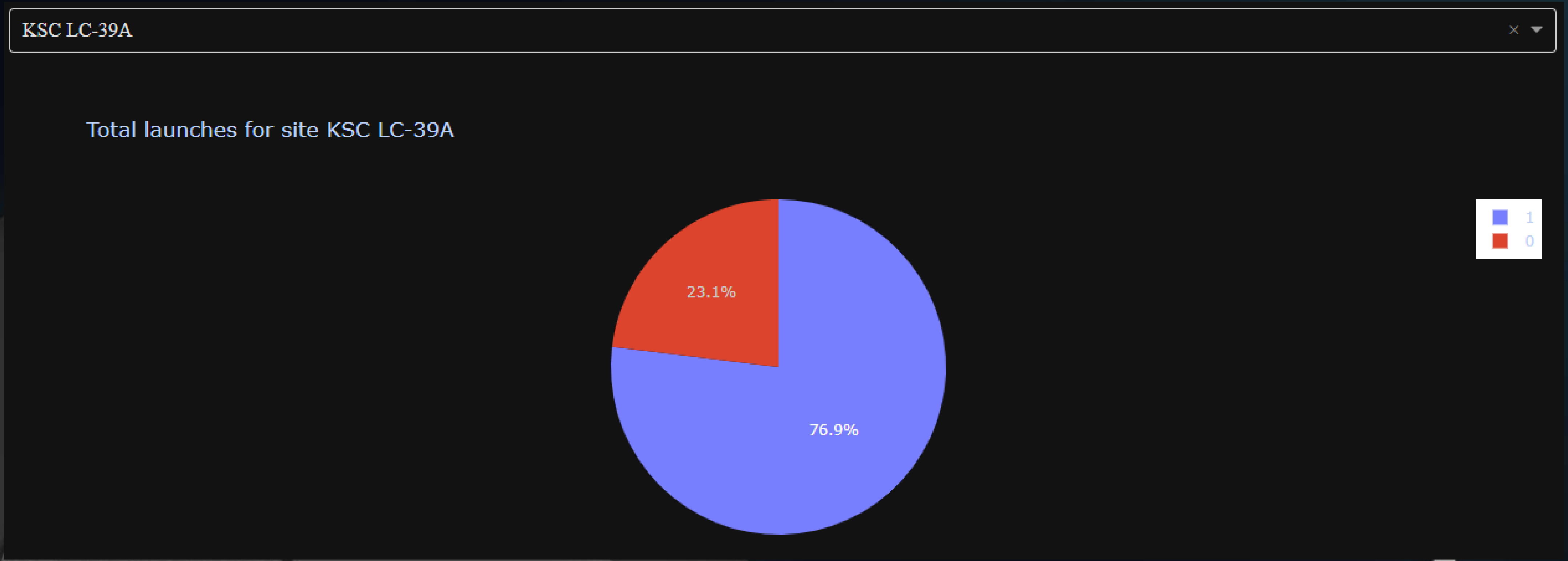
Pie-chart with total number of successful landing outcomes



All Sites

This Pie-chart makes it obvious that the ‘KSC LC-39A’ launch site has the largest share of successful landing outcomes among other launch sites

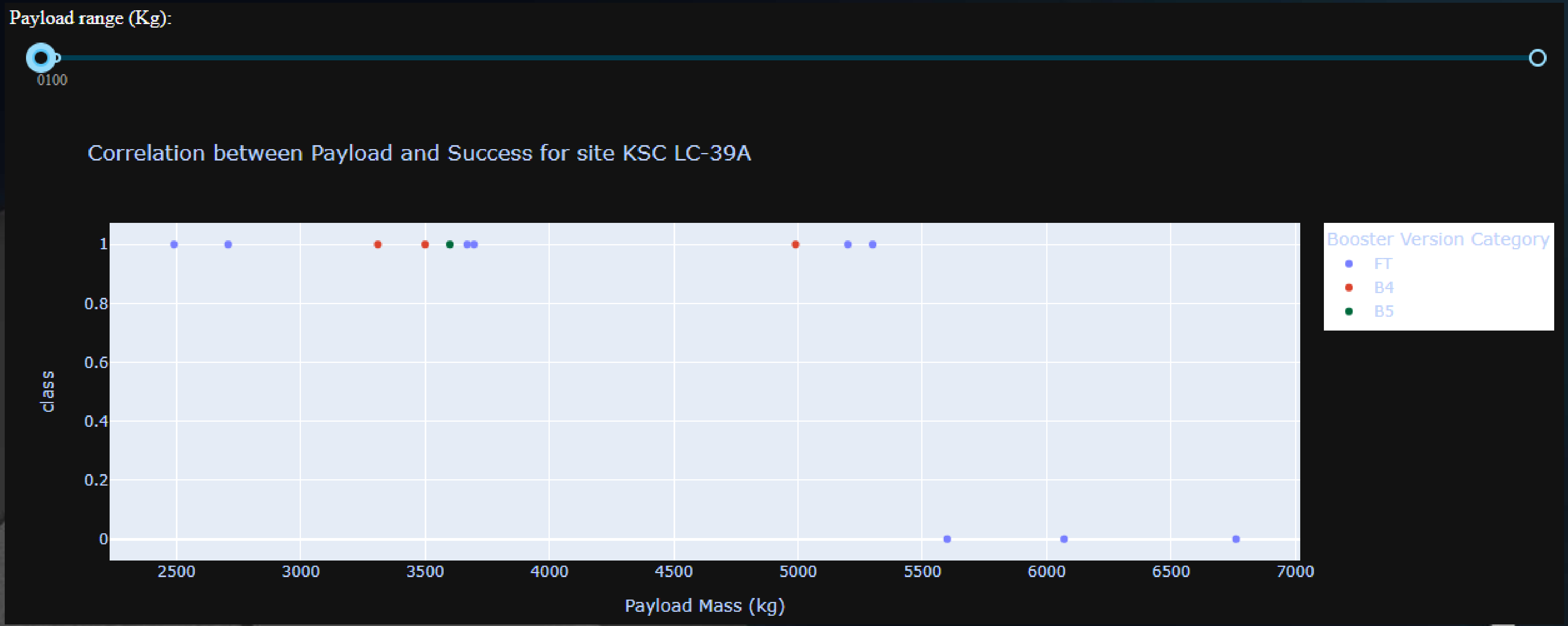
Pie-chart with the success rate of landing outcome for one launch site



KSC LC-39A

This Pie-chart depicts that the success rate of landing outcomes for the ‘KSC LC-39A’ launch site is 76.9%.

Scatter-plot with Success Rate depicted using Booster Version Category and Payload Mass (kg) relationship



Payload range (Kg):

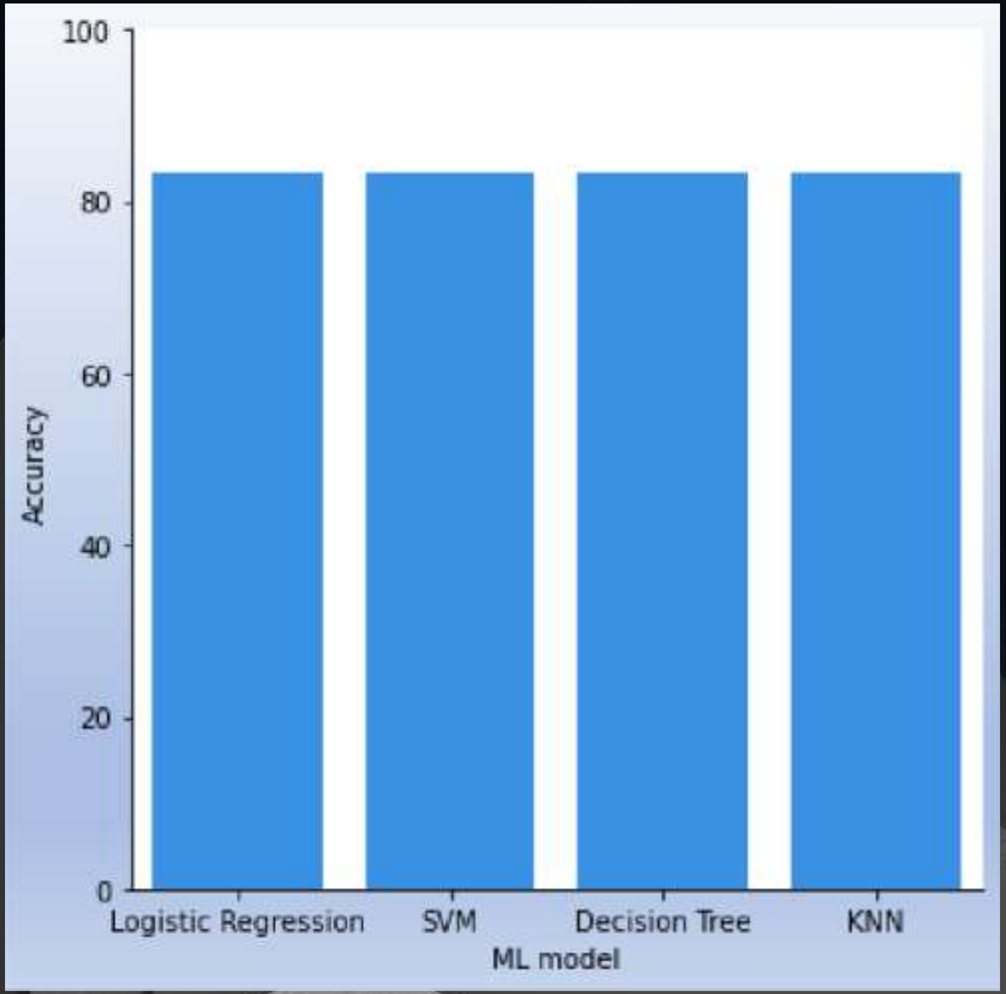
0100

- On this scatter plot we can see that the rocket with ‘FT’ Booster Version Category has performed all launches with Payload Mass above 5500 kg and they were all unsuccessful in terms of landing outcomes.
- At the same time most of the successful landing outcomes with Payload Mass below 5500 kg belong to the same booster version.

Section 6

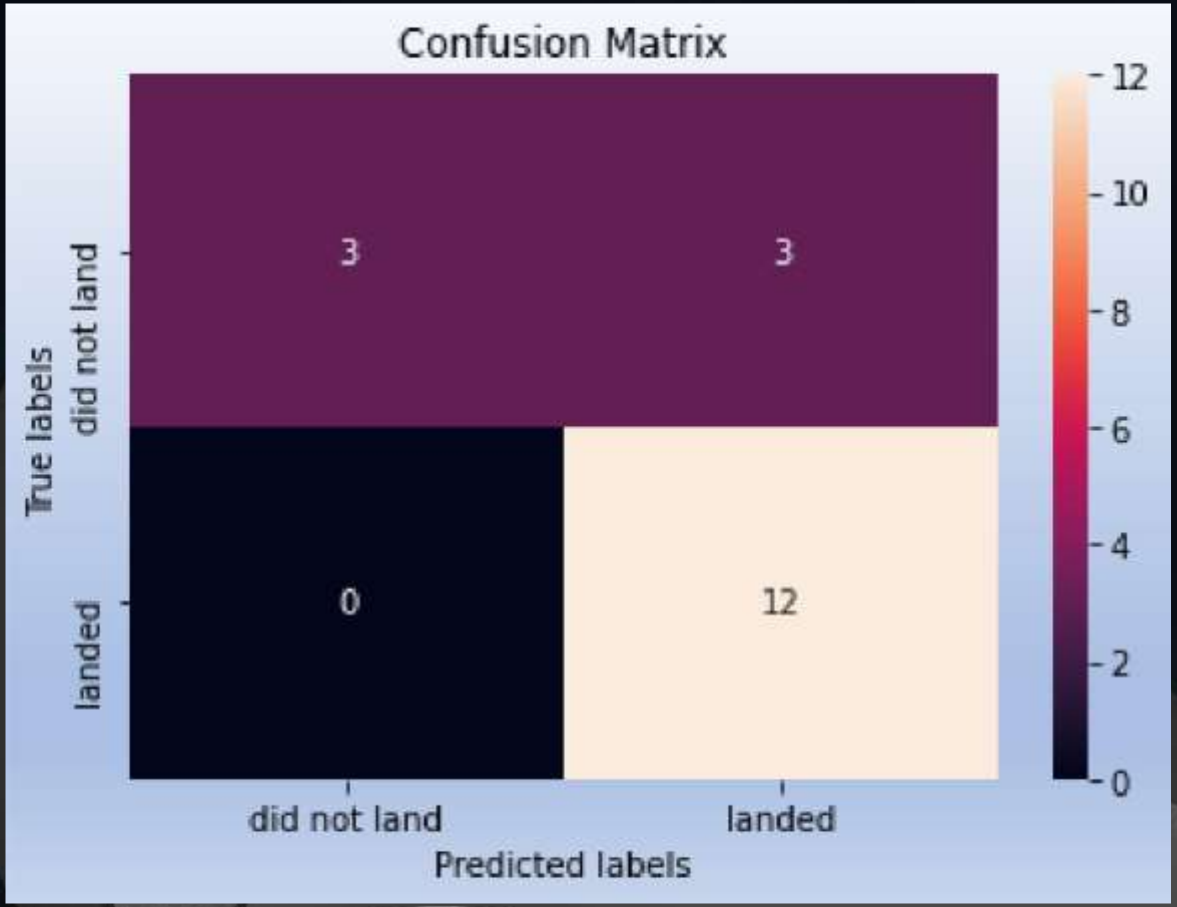
Predictive Analysis (Classification)

Classification Accuracy



On this bar chart we can see that all the models built return the same accuracy level of 83.33%.

Confusion Matrix

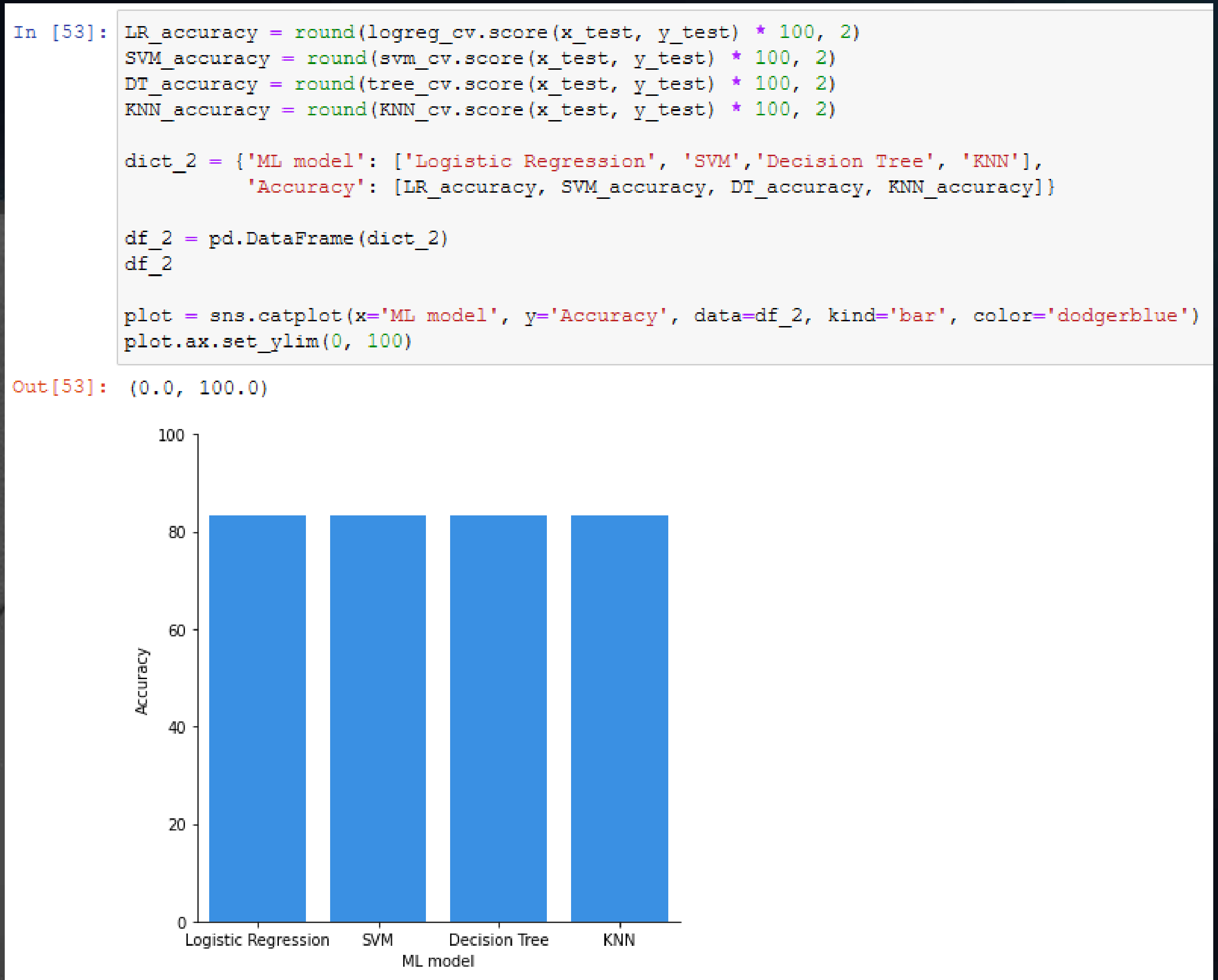


- As all of our models returned the same level of accuracy, we can observe the overall result on one heat-plot. On this confusion matrix heat-plot there was a total of 18 landing outcomes and 6 of them were unsuccessful. Our models could only predict half of 6 unsuccessful outcomes correctly (True Positive).
- On the other hand our models could predict correctly all 12 successful outcomes (True Negative).

Conclusions

- The Success Rate of SpaceX mission’s landing outcomes have gone from 0% (2013) to ≈85% (2020). This is roughly 12% increase every year, which is extremely impressive;
- The orbit types with the highest Success Rate (100%) of the landing outcomes are: ‘ES-L1’, ‘GEO’, ‘HEO’ and ‘SSO’. Although it should be considered that there was only 1 mission launched to the first 3 orbit types;
- The orbit type with the lowest Success Rate (≈52%) is ‘GTO’. We don’t consider ‘SO’ orbit type here because the loss of the first stage in this mission was required for Crew Dragon (C205) abort system testing purposes;
- Most of the missions (≈74%) had payload mass below 8000 kg and the Success Rate of this missions is ≈60%. The Success Rate of the missions with payload mass above 8000 kg is ≈87%;
- The orbit type with highest Success Rate with payload mass below 8000 kg are the ones indicated in the second bullet;
- The ‘KSC LC-39A’ launch sites has a ≈41% portion of all successful landing outcomes among other launch sites. The ‘CAFS SLC-40’ launch sites has a ≈12% portion, which is the lowest;
- All unsuccessful landing outcomes have happened with the first stages within ‘FT’ booster version category and all of these launches had a payload mass above 5500 (kg). The first stages within ‘B4’ and ‘B5’ booster version category had 100% Success Rate.

Appendix



```
In [53]: LR_accuracy = round(logreg_cv.score(x_test, y_test) * 100, 2)
         SVM_accuracy = round(svm_cv.score(x_test, y_test) * 100, 2)
         DT_accuracy = round(tree_cv.score(x_test, y_test) * 100, 2)
         KNN_accuracy = round(KNN_cv.score(x_test, y_test) * 100, 2)

         dict_2 = {'ML model': ['Logistic Regression', 'SVM','Decision Tree', 'KNN'],
                   'Accuracy': [LR_accuracy, SVM_accuracy, DT_accuracy, KNN_accuracy]}

         df_2 = pd.DataFrame(dict_2)
         df_2

         plot = sns.catplot(x='ML model', y='Accuracy', data=df_2, kind='bar', color='dodgerblue')
         plot.ax.set_ylim(0, 100)

Out[53]: (0.0, 100.0)
```

Thank you!