IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Ella Parker
14.06.22

# Outline

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

# Executive Summary

- Performed Data Collection using an API and Web Scraping. Then, Data Wrangling was performed to convert the collected data into a compatible and usable format for analysis. Exploratory data analysis (EDA) using visualization and SQL was completed to gain an understanding of the entire dataset to be worked with. Interactive visual analytics were created using Folium and Plotly Dash, to demonstrate the success of certain launch sites and payloads. Finally predictive analysis was completed using classification to create models which predict if the Falcon 9 First Stage Landing will be successful.
- Several statistics highlighted that the success rates for the Landing of the Falcon 9 First Stage relied upon the payload, launch site, and the success rates seem to be increasing as more rockets are sent into orbit. There is also a correlation between payload mass, orbit and success rates.

# Introduction

- SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards 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. Spaces X's Falcon 9 launch like regular rockets. This information can be used if an alternate company wants to bid against space X for a rocket launch.
- As a competing company we want to predict if the first stage will land, so we can determine the cost of a launch and provide competitive prices with our own rockets.

Section 1

# Methodology

# Methodology

## Executive Summary

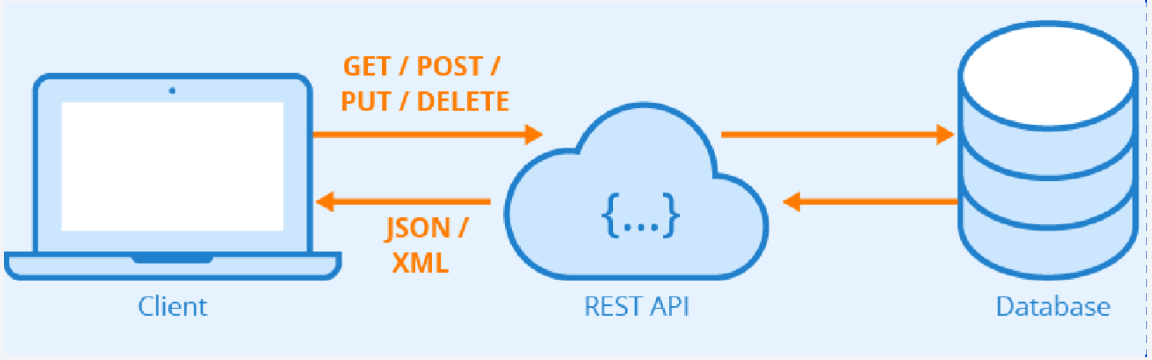
- Data collection methodology:
  - Data was collecting using the SpaceX API and from a Wikipedia page titled ‘List of Falcon 9 and Falcon Heavy launches
- Perform data wrangling
  - The data was processed to convert outcomes into Training Labels (with 1 meaning a successful land and 0 meaning an unsuccessful outcome).
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - The best Hyperparameter for SVM, Decision Tree, Logistic Regression and K Nearest Neighbors was determined and the most accurate model was found.

# Data Collection

- Data was collecting using the SpaceX API and from a Wikipedia page titled ‘List of Falcon 9 and Falcon Heavy launches. This data was then extracted into a data frame, so that it could be transformed and analysed at later stages.
- Using the API the ‘rocket’ column told us about the booster name, the ‘launchpad’ column gave us the name of the launch site being used, the longitude and the latitude, the ‘payload’ demonstrated the mass of the payload and the orbit the rocket is going to, and the ‘cores’ gave us the outcome and type of landing, number of flights with that core, if gridfins were used, if the core is reused, if legs were used, the landing pad type used, the block of the core, the number of times that core has been reused and the serial of the core.
- Using webscraping allowed us to gain this same information using BeautifulSoup objects.

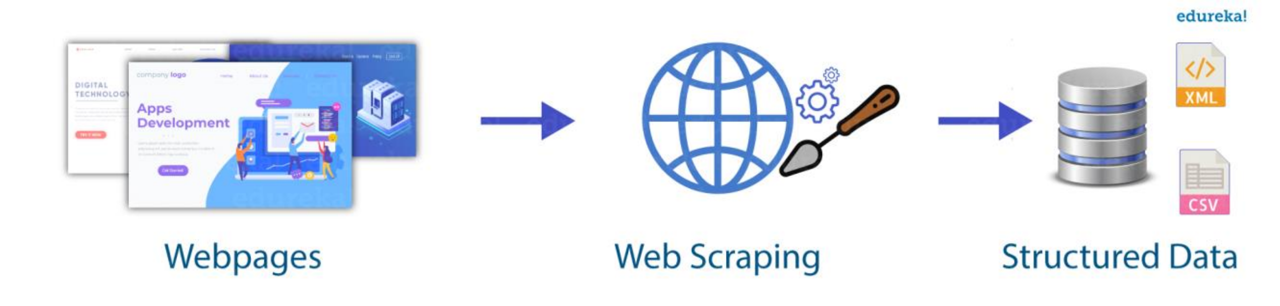
# Data Collection – SpaceX API

- SpaceX REST API works in the same way as any other REST API, as demonstrated in the diagram to the right. The SpaceX REST API allows the client to request information about rocket launches from the SpaceX database
- https://github.com/ellxparker/Applied-Data-Science-Capstone/blob/ee360781e0255d347f978967dcc03beaae9bb4bb/Data%20Collection%20API%20Lab.ipynb



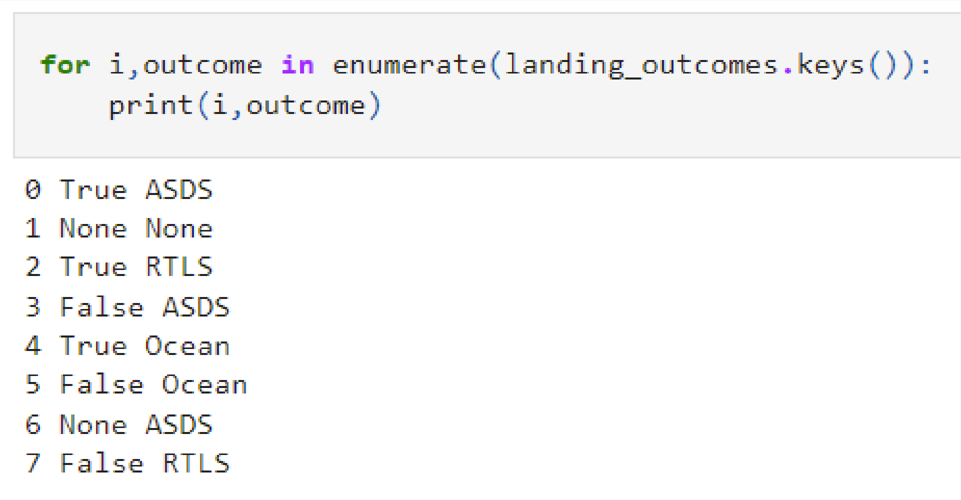
# Data Collection - Scraping

- The flowchat on the right shows the method of using web scraping, in conjunction with BeautifulSoup to obtain and structure the information from a website – used in this project to gain information from a Wikipedia page about rocket launches.
- https://github.com/ellxparker/Applied-Data-Science-Capstone/blob/ee360781e0255d347f978967dcc03beaae9bb4bb/Data%20Collection%20with%20Web%20Scraping.ipynb



# Data Wrangling

- The data was explored using informal exploratory data analysis, and the values for non-integer columns were transformed into Training Labels. An additional column was also added, “class”, which represents a successful or unsuccessful landing (1=successful, 0=unsuccessful), extracted from the landing_outcomes column, with the possible seen below
- Link: https://github.com/ellxparker/Applied-Data-Science-Capstone/blob/ee360781e0255d347f978967dcc03beaae9bb4bb/Data%20Wrangling.ipynb



```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```

```
0 True ASDS
1 None None
2 True RTLS
3 False ASDS
4 True Ocean
5 False Ocean
6 None ASDS
7 False RTLS
```

# EDA with Data Visualization

- Cat plots were used to plot FlightNumber against PayloadMass, FlightNumber against LaunchSite, LaunchSite against PayloadMass, FlightNumber against Orbit Type, PayloadMass against Orbit Type. These Cat plots were used to demonstrate the relationship between categorical variables. A bar chat was used to plot average success rate for each Orbit Type, which demonstrated the orbit types which had a higher landing success rate. Finally, a line graph was used to show the launch success yearly trend, showing that from 2013 the success rate increased until 2020.
- Link: https://github.com/ellxparker/Applied-Data-Science-Capstone/blob/ee360781e0255d347f978967dcc03beaae9bb4bb/Exploratory%20Data%20Analysis%20with%20Vizualisation.ipynb

# EDA with SQL

SQL Queries:

- Display the distinct launch sites
- Display 5 records with launch sites starting with ‘CCA’
- Display total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- Display the date of first successful landing on a ground pad
- Display the booster names which have success in drone ship and have payload mass between 4000 and 6000 kg
- Display the total number of success and failure mission outcomes
- Display the names of the booster versions which have carried the max payload mass
- Display the failed landing outcomes in drone ship, their booster versions and launch sites for 2015
- Rank the count of landing outcomes between 2010-06-04 and 2017-03-20 in descending order

Link: https://github.com/ellxparker/Applied-Data-Science-Capstone/blob/ee360781e0255d347f978967dcc03beaae9bb4bb/Exploratory%20Data%20Analysis%20with%20SQL.ipynb

# Build an Interactive Map with Folium

- All launch sites were added on a map with the start location as NASA Johnson Space Centre. This was added to give a visual representation of all the areas that the rockets could be launched from. You can observe that all locations are by the coast in Southern USA. For each location a circle was added with a marker was also added for each location to show the name of the launch site, to make the map easier to read. Then, I plotted the success and failed launches on the map with colour coded points, so I could visually see if there was any pattern between success/failure and location. These points were part of MarkerClusters to group similar launch sites together, so the map did not become overcrowded. Finally, I used the map to calculate the distances between a launch site and the proximities to it. This allowed me to visually see the distance to the ocean, cities and other major landmarks.
- Link (As the Watson Studio Ran out of storage I had to complete these in the Skills Lab and the maps are not showing, instead these maps are shown in the launch site proximity section) https://github.com/ellxparker/Applied-Data-Science-Capstone/blob/ee360781e0255d347f978967dcc03beaae9bb4bb/Interactive%20Visual%20Analytics%20with%20Folium%20Fixed%20Errors.ipynb

# Build a Dashboard with Plotly Dash

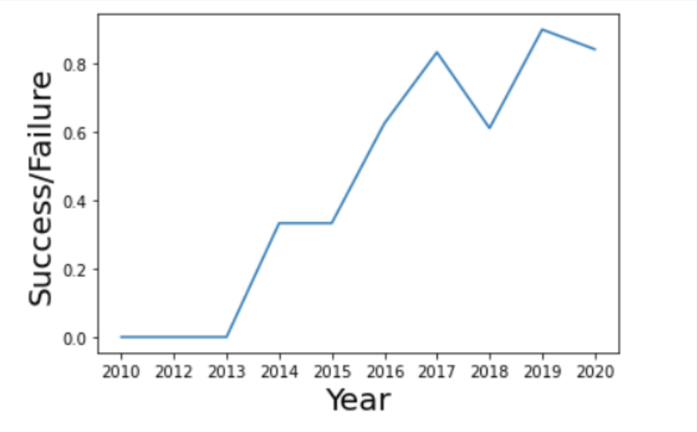
- A pie chart was created to show the proportion of success or failure outcomes for a certain (or all) launch site. A cat chart was also created to show the correlation between payload and success for a certain (or all) launch site, dependent on the booster version. There was a drop down menu to select the launch site to analyse and a slider to select the range of payload masses.
- These plots were added to show and compare the average success rates of each launch site visually. It also demonstrated if there was any correlation between the payload mass and success rate, or the booster version and success rate.
- Link: https://github.com/ellxparker/Applied-Data-Science-Capstone/blob/ee360781e0255d347f978967dcc03beaae9bb4bb/spacex_dash_app.py

# Predictive Analysis (Classification)

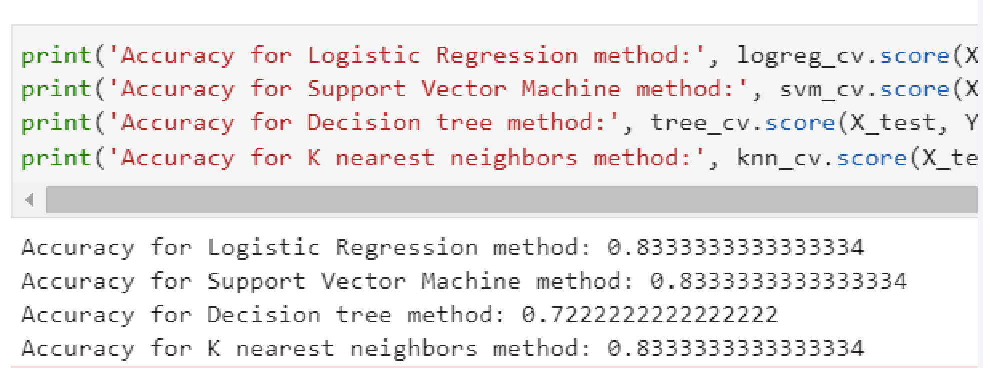
- Four models were created (SVM, Decision Tree, Logistic Regression and K Nearest Neighbors) and the best hyperparameter for each was determined using a GridSearchCV object. The accuracy for each model (with the optimized parameters) was then calculated using the accuracy method, and the models were compared to see which would be the best fit for the scenario.
- Created in Skills Lab and Uploaded to GitHub – the Notebook sometimes does not open on GitHub but if you put the link into nbviewer.org then it will load (ignore all the errors). Link: https://github.com/ellxparker/Applied-Data-Science-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5%20(1).ipynb

# Results

Exploratory data analysis results:



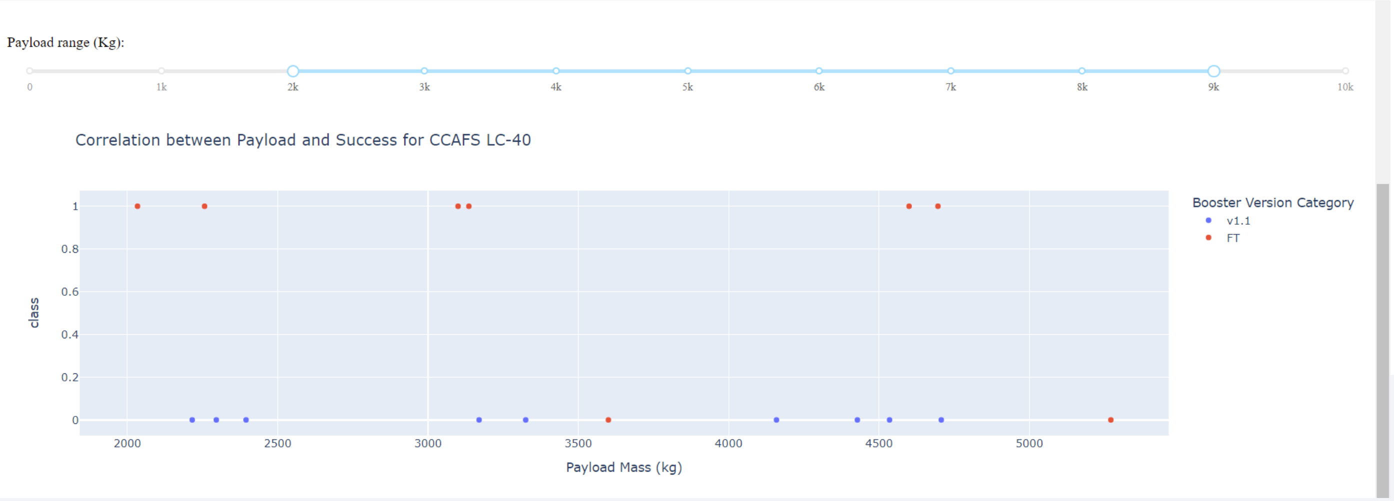
Predictive Analytics Model Accuracy:

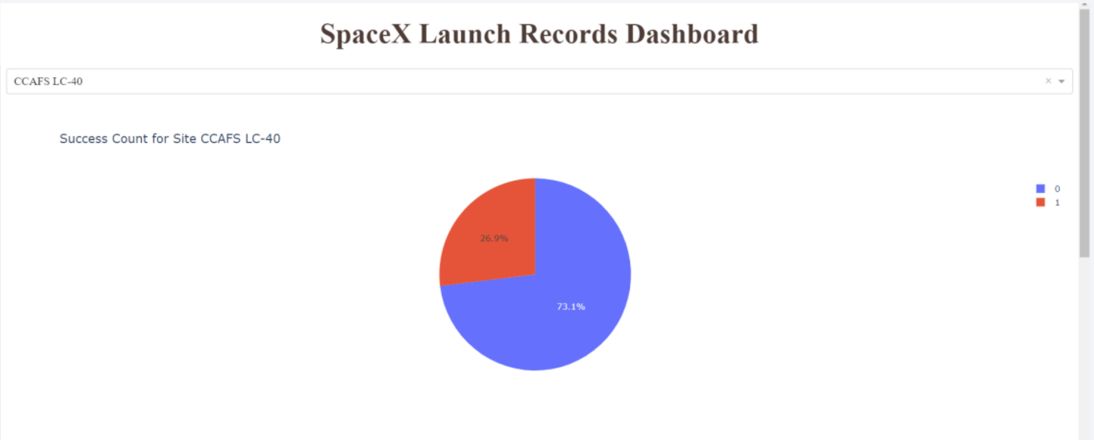


```
print('Accuracy for Logistic Regression method:', logreg_cv.score(X
print('Accuracy for Support Vector Machine method:', svm_cv.score(X
print('Accuracy for Decision tree method:', tree_cv.score(X_test, Y
print('Accuracy for K nearest neighbors method:', knn_cv.score(X_te
```

```
Accuracy for Logistic Regression method: 0.8333333333333334
Accuracy for Support Vector Machine method: 0.8333333333333334
Accuracy for Decision tree method: 0.7222222222222222
Accuracy for K nearest neighbors method: 0.8333333333333334
```

Interactive Analytics:





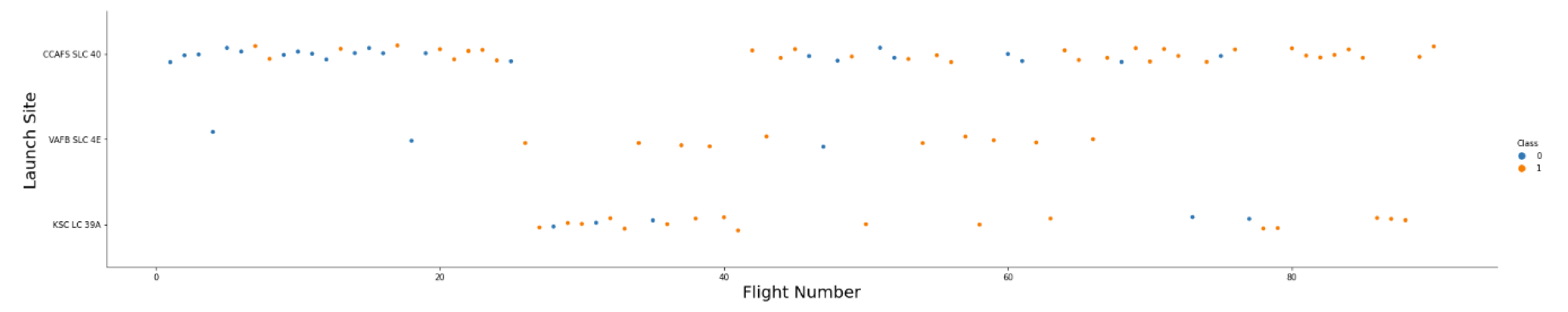
SpaceX Launch Records Dashboard
CCAFS LC-40
Success Count for Site CCAFS LC-40
26.9%
73.1%
0
1

Section 2

# Insights drawn from EDA

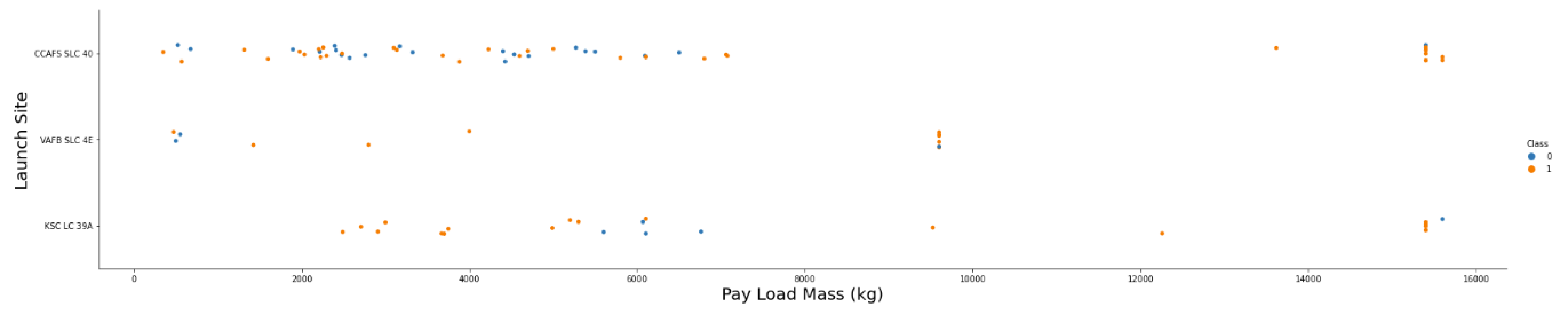
# Flight Number vs. Launch Site

- The scatter point graph shows the flight numbers with the corresponding launch sites that the rockets took off at. The points are also colour coded by success/failure of landing (Blue is failure and Yellow is success. One thing it shows is that CCAFS SLC 40 has a higher success rate with later flight numbers.



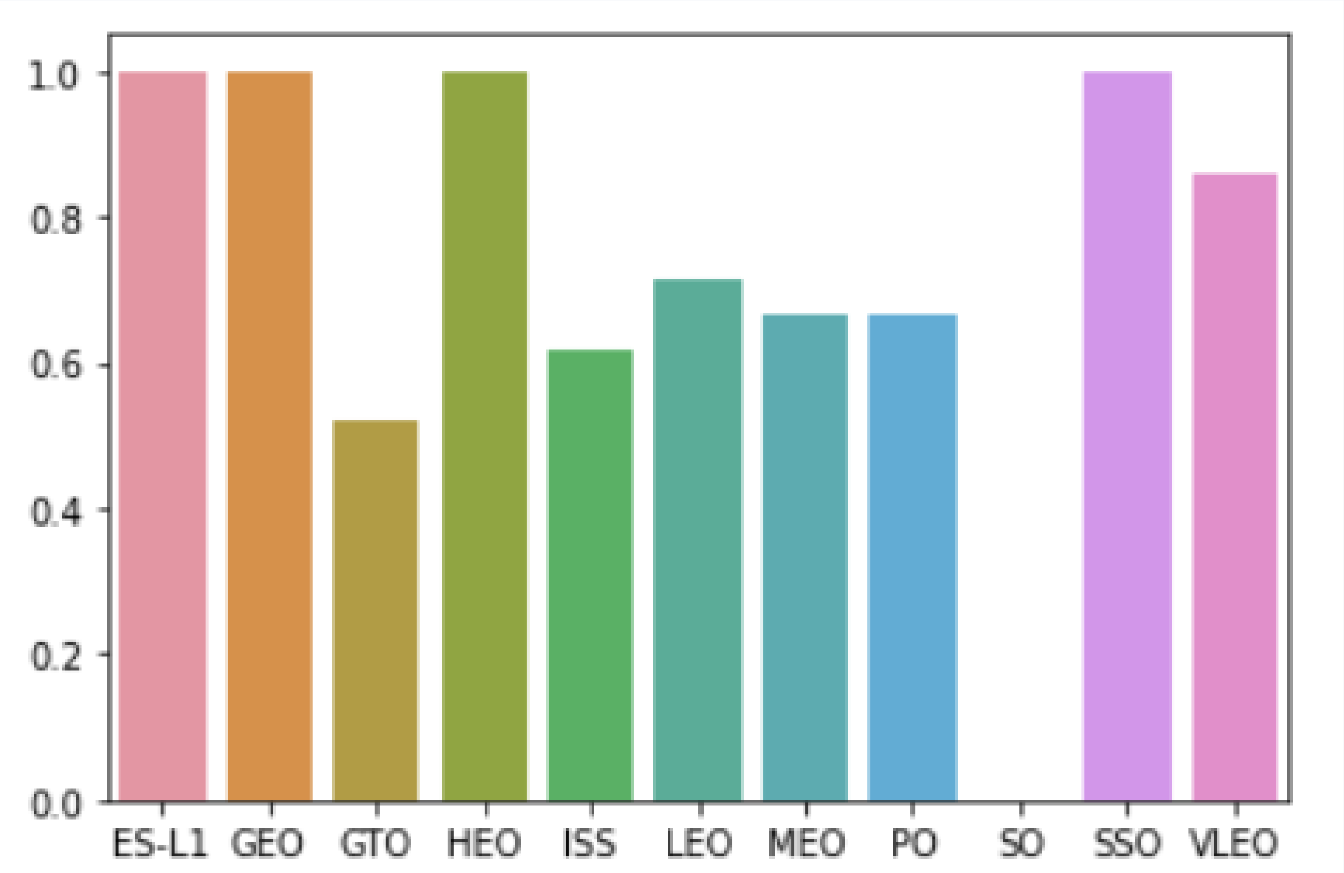
# Payload vs. Launch Site

- The scatter point graph shows the launch sites with the corresponding payload mass of each rocket. The points are also colour coded by success/failure of landing (Blue is failure and Yellow is success. One thing it shows is that payloads with a higher mass have a greater success at the CCAFS SLC 40 launch site.



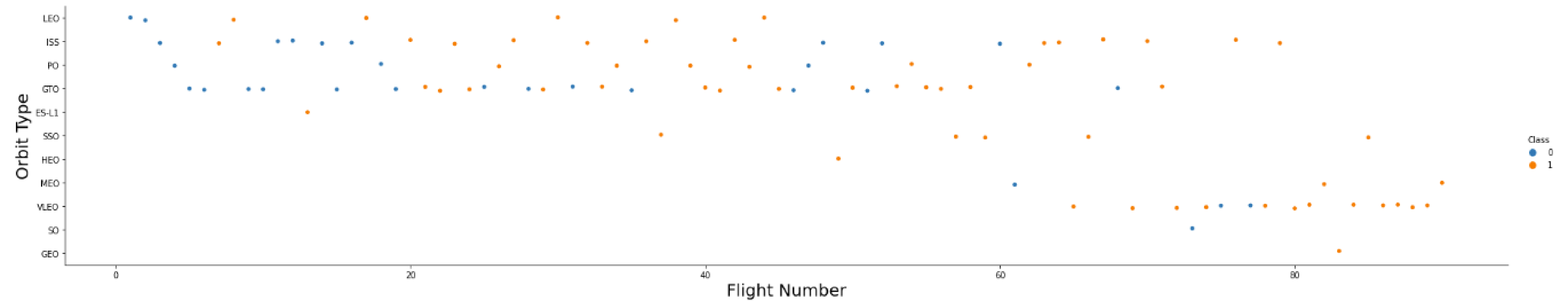
# Success Rate vs. Orbit Type

- The bar chart shows the different orbit types with their corresponding average success rate. One thing it shows is that ES-L1, GEO, HEO, and SSO orbits have had no failed landings, demonstrated by the average success rate of 1. However, the SO orbit type has not had a successful landing as the average success rate is 0.



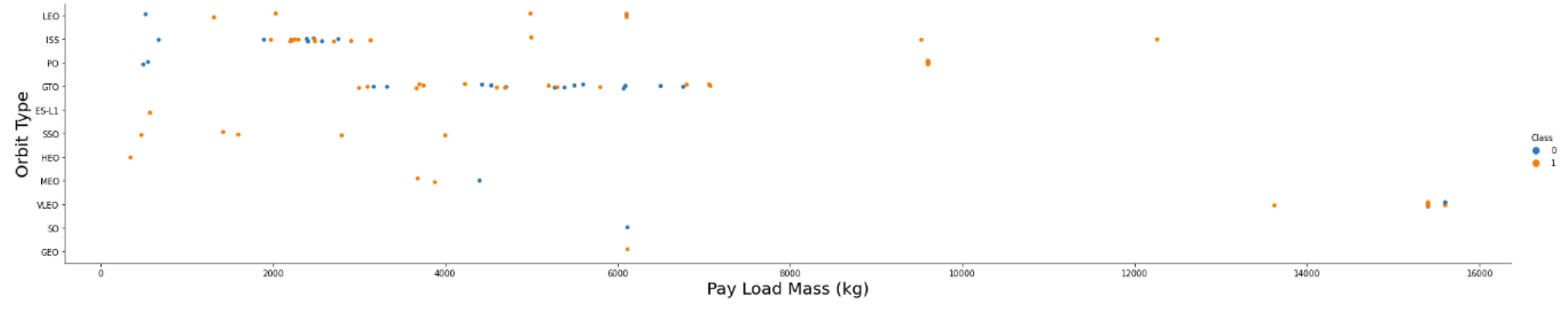
# Flight Number vs. Orbit Type

- The scatter point graph shows the flight numbers with their corresponding orbit types. The points are also colour coded by success/failure of landing (Blue is failure and Yellow is success. One thing it shows is that in the LEO orbit the Success appears related to the number of flights.



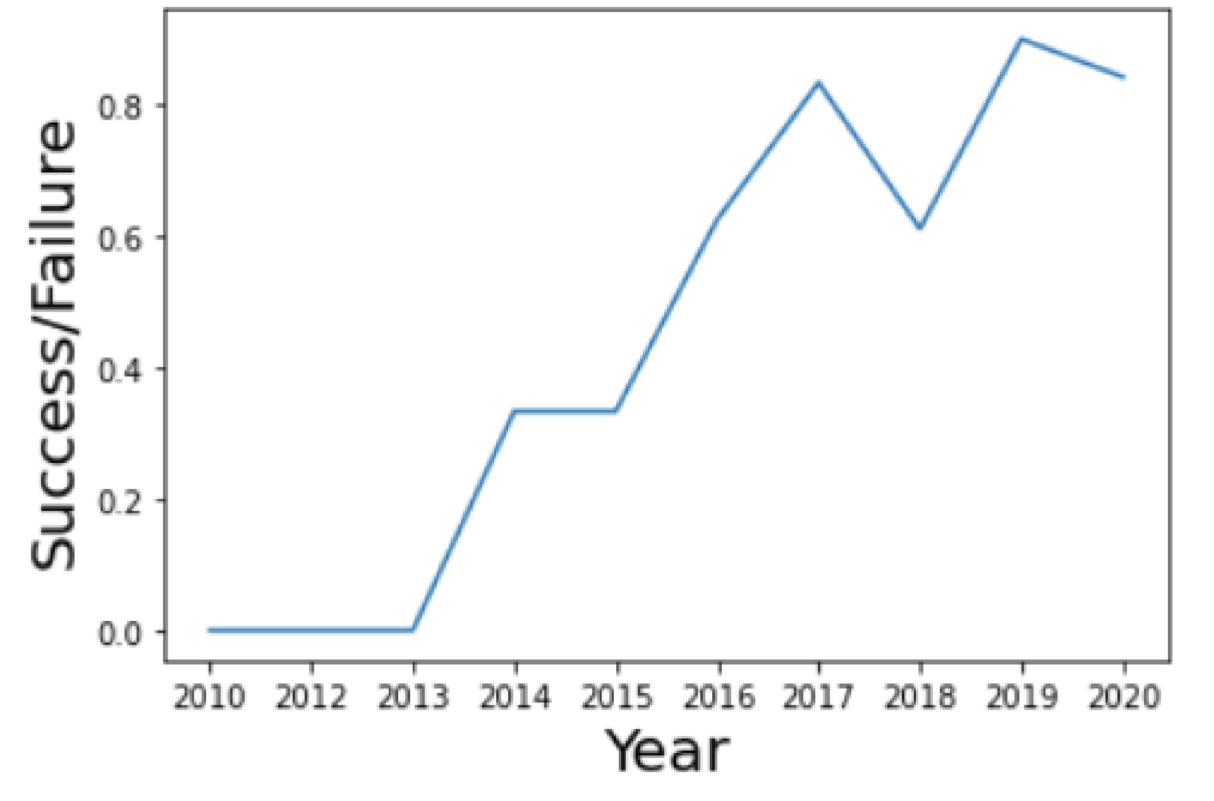
# Payload vs. Orbit Type

- The scatter point graph shows the orbit types with their corresponding payload mass. The points are also colour coded by success/failure of landing (Blue is failure and Yellow is success. One thing it shows is that the success rate with heavy payloads is greater for Polar, LEO and ISS.



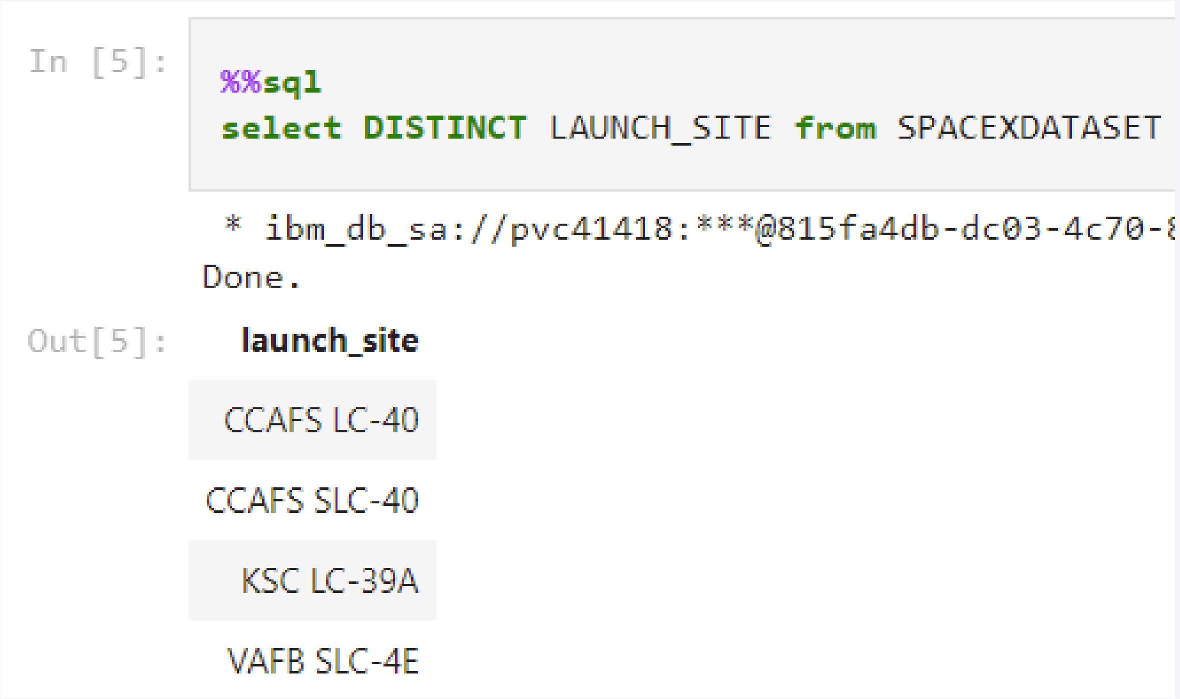
# Launch Success Yearly Trend

- The line chart shows the average launch success trend over the years that SpaceX has been launching the Falcon9. One thing it shows is that the success rate since 2013 kept increasing till 2017, before dropping slightly in 2018, however the success rate then increased again in 2019, before dropping slightly down in 2020.



# All Launch Site Names

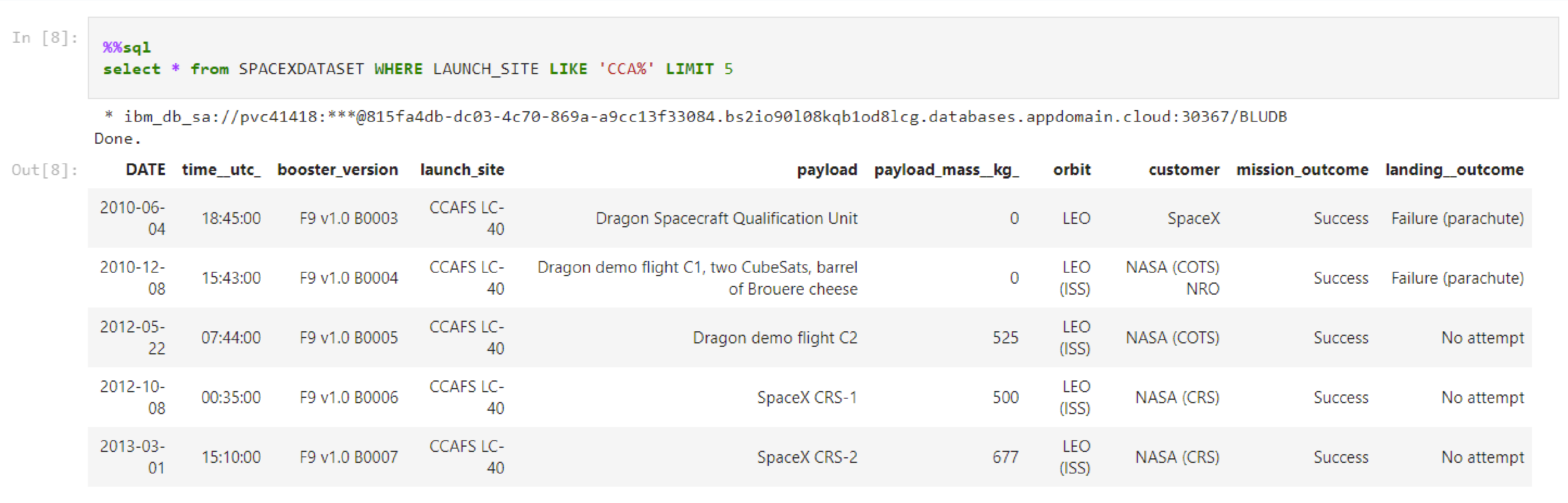
- The SQL query on the right performs a search on the database containing the SpaceX Falcon9 data, returning the list of all distinct launch sites. You can see that there are 4 distinct launch sites which the Falcon9 rockets were launched from (CCAFS LC-40, CCAFS SLC-40, KSC LC-39A, and VAFB SLC-4E).



```
In [5]:  %%sql
         select DISTINCT LAUNCH_SITE from SPACEXDATASET

          * ibm_db_sa://pvc41418:***@815fa4db-dc03-4c70-8
         Done.
Out[5]:     launch_site
            CCAFS LC-40
           CCAFS SLC-40
             KSC LC-39A
            VAFB SLC-4E
```

# Launch Site Names Begin with 'CCA'

- The SQL query below performs a search on the database containing the SpaceX Falcon9 data, returning the first 5 records with a launch site beginning with ‘CCA’. The query returns all the data for each record with a matching launch site.



In [8]:

```
%%sql
select * from SPACEXDATASET WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

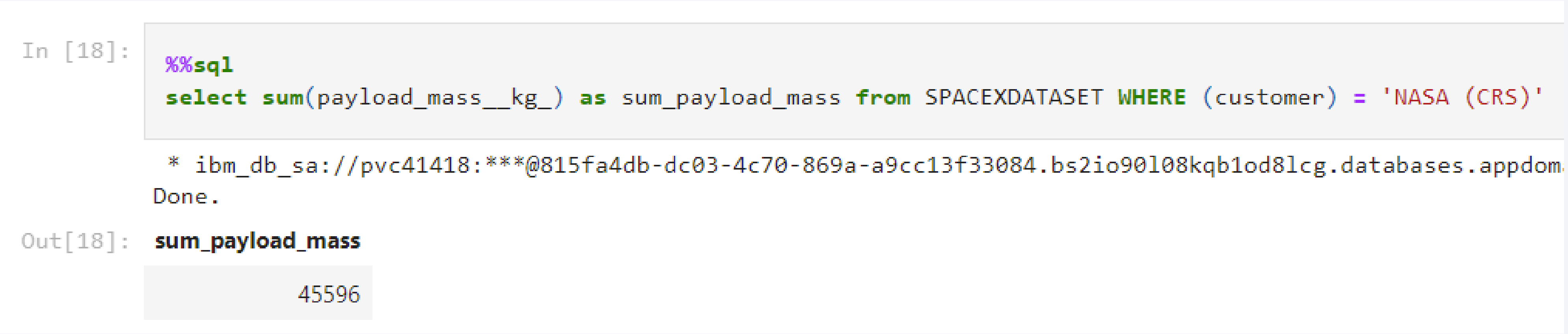
```
 * ibm_db_sa://pvc41418:***@815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30367/BLUDB
Done.
```

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 |

# Total Payload Mass

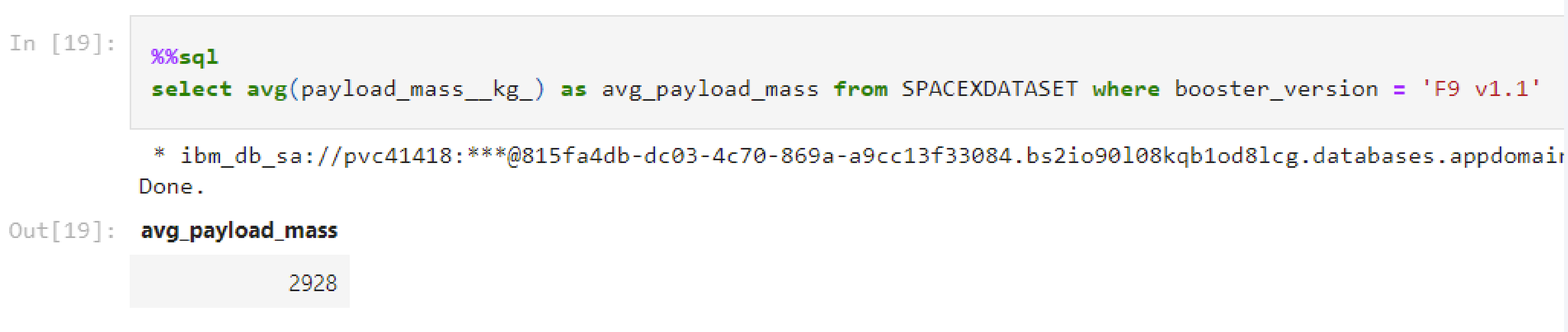
- The SQL query below performs a search on the database containing the SpaceX Falcon9 data, using the aggregate function ‘sum’ to generate the total payload mass for all rockets launched by NASA (CRS).



```
In [18]:  %%sql
          select sum(payload_mass__kg_) as sum_payload_mass from SPACEXDATASET WHERE (customer) = 'NASA (CRS)'

           * ibm_db_sa://pvc41418:***@815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdom
          Done.

Out[18]:  sum_payload_mass
                     45596
```

# Average Payload Mass by F9 v1.1

- The SQL query below performs a search on the database containing the SpaceX Falcon9 data, using the aggregate function ‘avg’ calculate the average payload mass carried by all Falcon9 rockets with booster version F9 v1.1



In [19]:

```
%%sql
select avg(payload_mass__kg_) as avg_payload_mass from SPACEXDATASET where booster_version = 'F9 v1.1'
```

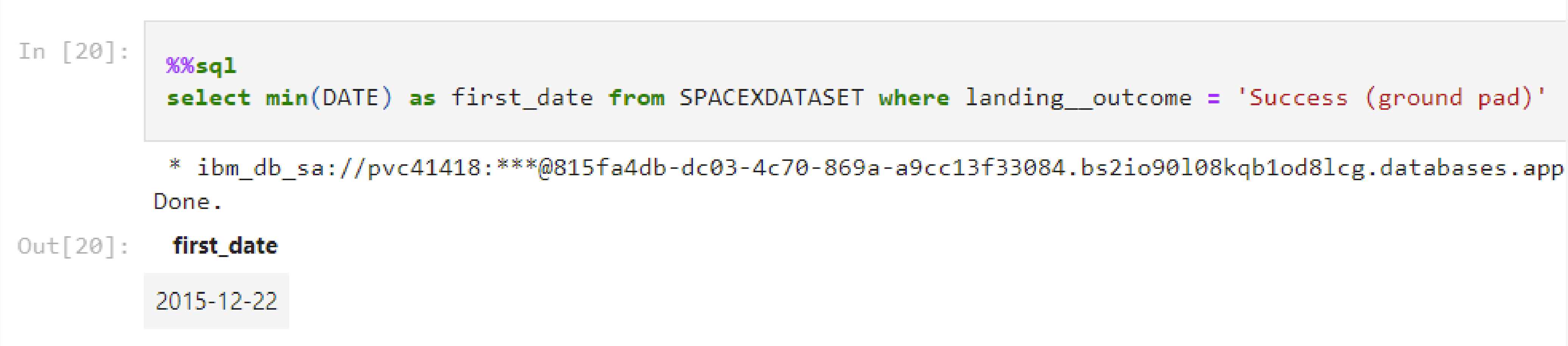
```
 * ibm_db_sa://pvc41418:***@815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomai
Done.
```

Out[19]:

| avg_payload_mass |
|---|
| 2928 |

# First Successful Ground Landing Date

- The SQL query below performs a search on the database containing the SpaceX Falcon9 data, using the aggregate function ‘min’ to select the row with the smallest (earliest) date from the dataset, and then returning the date in a column called ‘first_date’



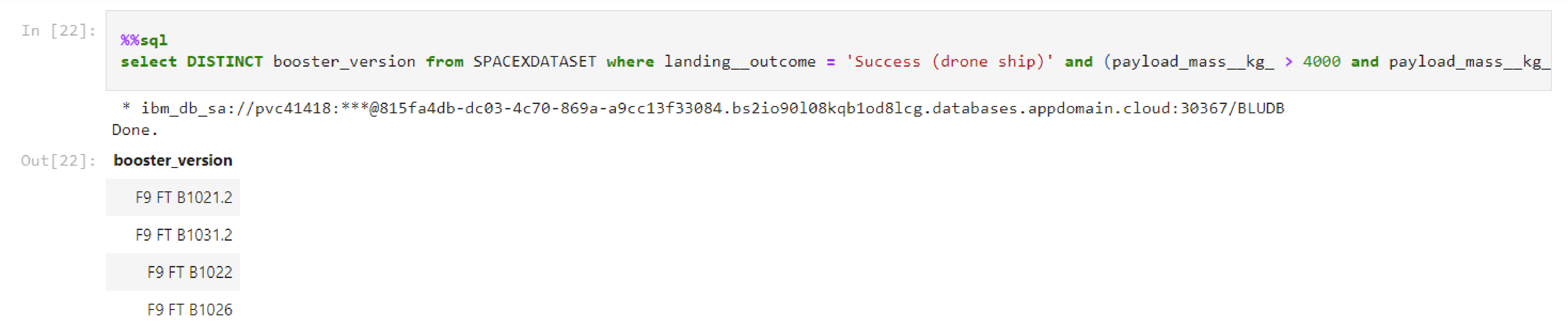
```
In [20]:  %%sql
          select min(DATE) as first_date from SPACEXDATASET where landing__outcome = 'Success (ground pad)'

           * ibm_db_sa://pvc41418:***@815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.app
          Done.
```

Out[20]:

| first_date |
|---|
| 2015-12-22 |

# Successful Drone Ship Landing with Payload between 4000 and 6000

- The SQL query below performs a search on the database containing the SpaceX Falcon9 data, using the distinct function in conjunction with multiple where clauses to select the booster versions which had a successful drone ship landing with a payload between 4000 and 6000 kg.



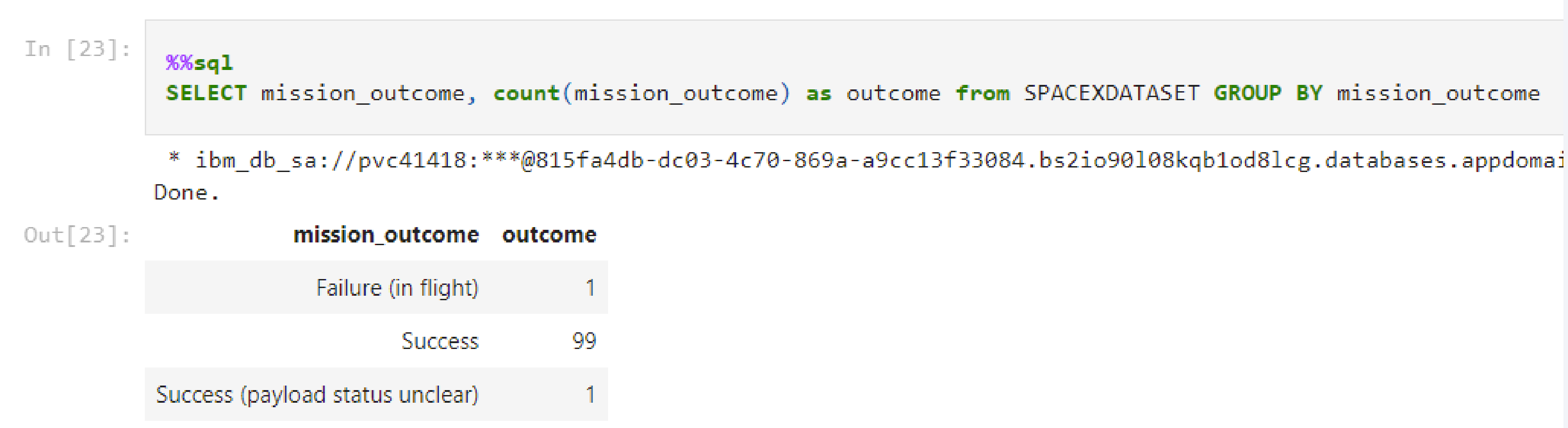
```
In [22]: %%sql
         select DISTINCT booster_version from SPACEXDATASET where landing__outcome = 'Success (drone ship)' and (payload_mass__kg_ > 4000 and payload_mass__kg_

          * ibm_db_sa://pvc41418:***@815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30367/BLUDB
         Done.
```

Out[22]:

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

# Total Number of Successful and Failure Mission Outcomes

- The SQL query below performs a search on the database containing the SpaceX Falcon9 data, using the ‘count’ aggregate function in conjunction with the ‘group by’ function to select the different mission outcomes and the count of these outcomes.



In [23]:

```
%%sql
SELECT mission_outcome, count(mission_outcome) as outcome from SPACEXDATASET GROUP BY mission_outcome
```

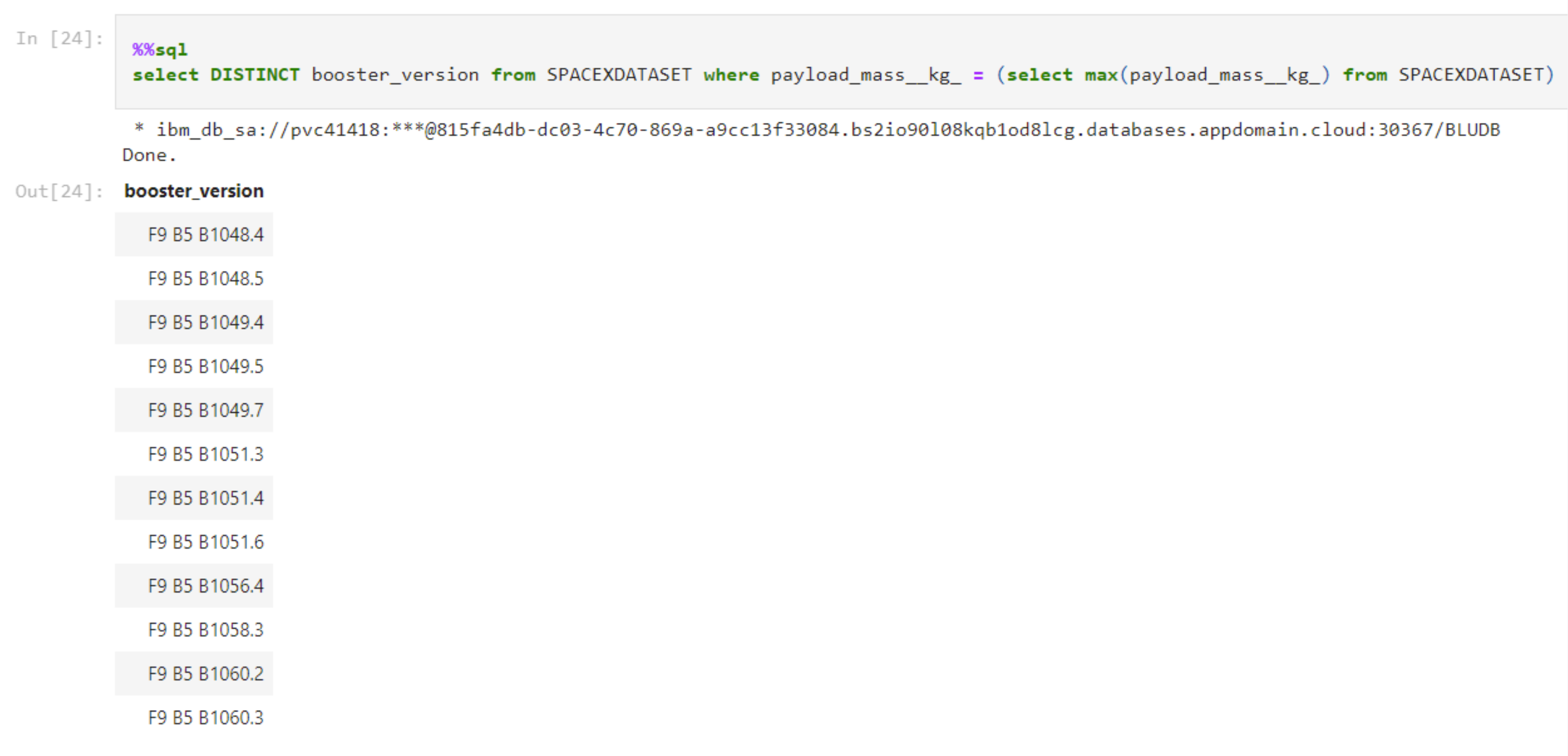
```
 * ibm_db_sa://pvc41418:***@815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomai
Done.
```

Out[23]:

| mission_outcome | outcome |
|---:|---:|
| Failure (in flight) | 1 |
| Success | 99 |
| Success (payload status unclear) | 1 |

# Boosters Carried Maximum Payload

- The SQL query below performs a search on the database containing the SpaceX Falcon9 data, using a subquery to find the maximum payload, and then selecting all distinct booster versions which have carried the maximum payload mass which was found in the sub query.



In [24]:

```
%%sql
select DISTINCT booster_version from SPACEXDATASET where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXDATASET)
```

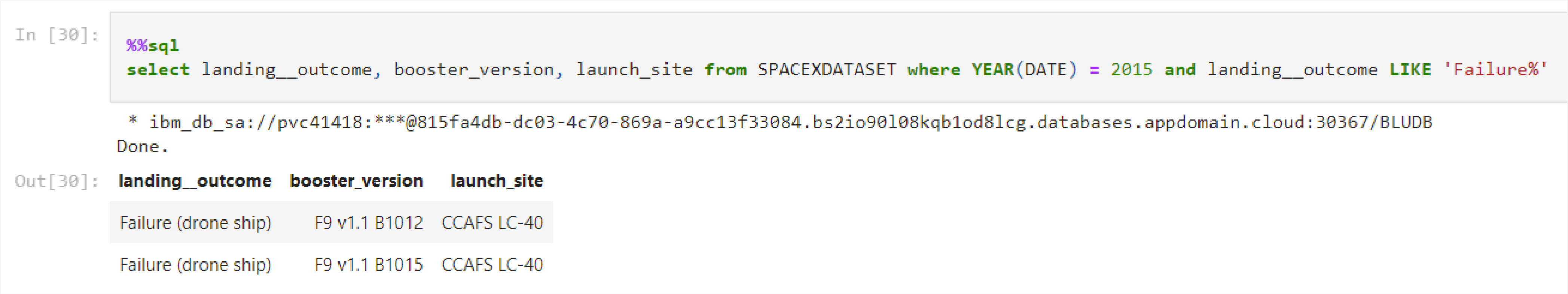
```
 * ibm_db_sa://pvc41418:***@815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30367/BLUDB
Done.
```

Out[24]:

| 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 |

# 2015 Launch Records

- The SQL query below performs a search on the database containing the SpaceX Falcon9 data, using multiple where clauses to list the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015.



In [30]:

```
%%sql
select landing__outcome, booster_version, launch_site from SPACEXDATASET where YEAR(DATE) = 2015 and landing__outcome LIKE 'Failure%'
```

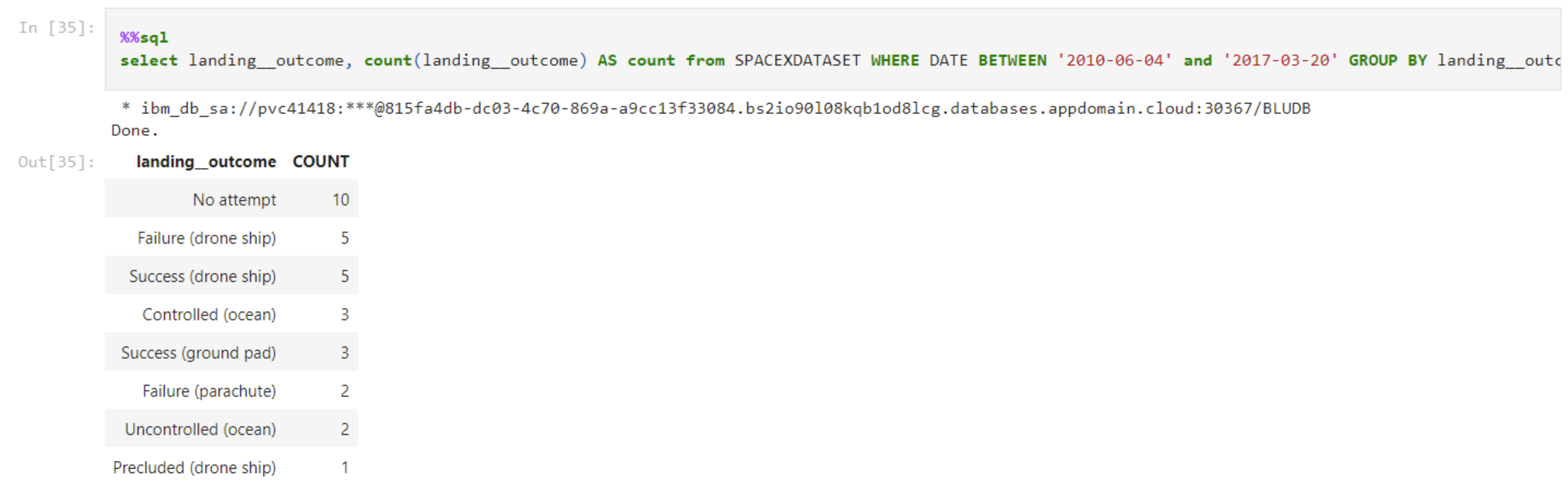
```
 * ibm_db_sa://pvc41418:***@815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30367/BLUDB
Done.
```

Out[30]:

| 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 |

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

- The SQL query below performs a search on the database containing the SpaceX Falcon9 data, where clauses find the landing between 2010-06-04 and 2017-03-20 and then using the ‘group by’ aggregate function to produce a count of each landing outcome for this time period. The query then uses the ‘ORDER BY’ function to sort the outcomes in order of their corresponding count, descending.



In [35]:

```
%%sql
select landing__outcome, count(landing__outcome) AS count from SPACEXDATASET WHERE DATE BETWEEN '2010-06-04' and '2017-03-20' GROUP BY landing__outc
```

```
 * ibm_db_sa://pvc41418:***@815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30367/BLUDB
Done.
```

Out[35]:

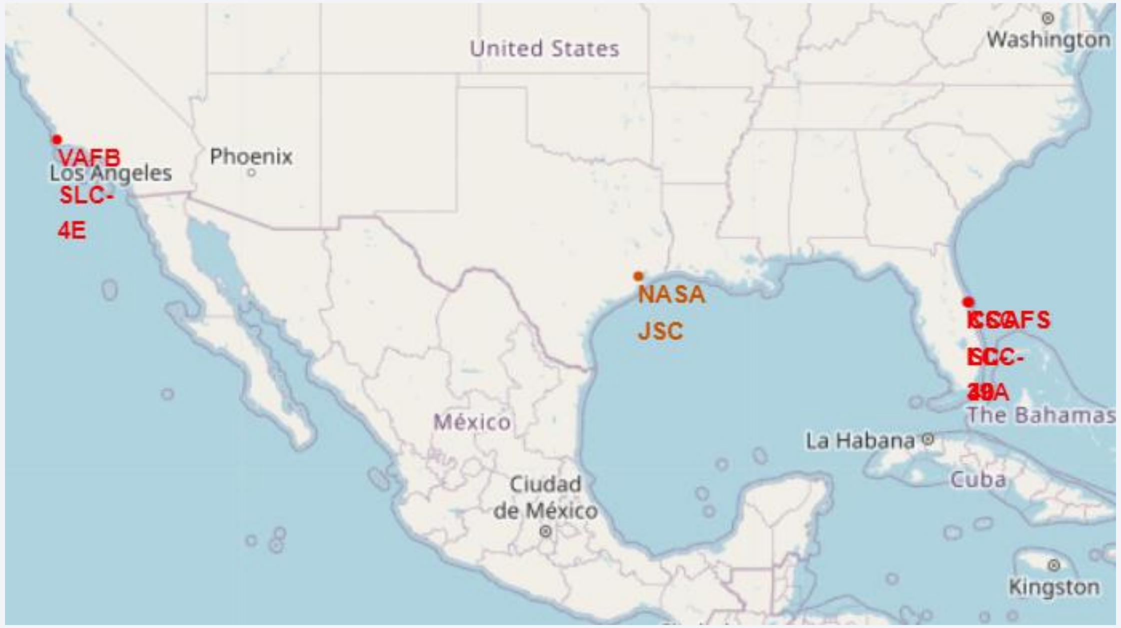
| landing__outcome | COUNT |
|---|---|
| No attempt | 10 |
| Failure (drone ship) | 5 |
| Success (drone ship) | 5 |
| Controlled (ocean) | 3 |
| Success (ground pad) | 3 |
| Failure (parachute) | 2 |
| Uncontrolled (ocean) | 2 |
| Precluded (drone ship) | 1 |

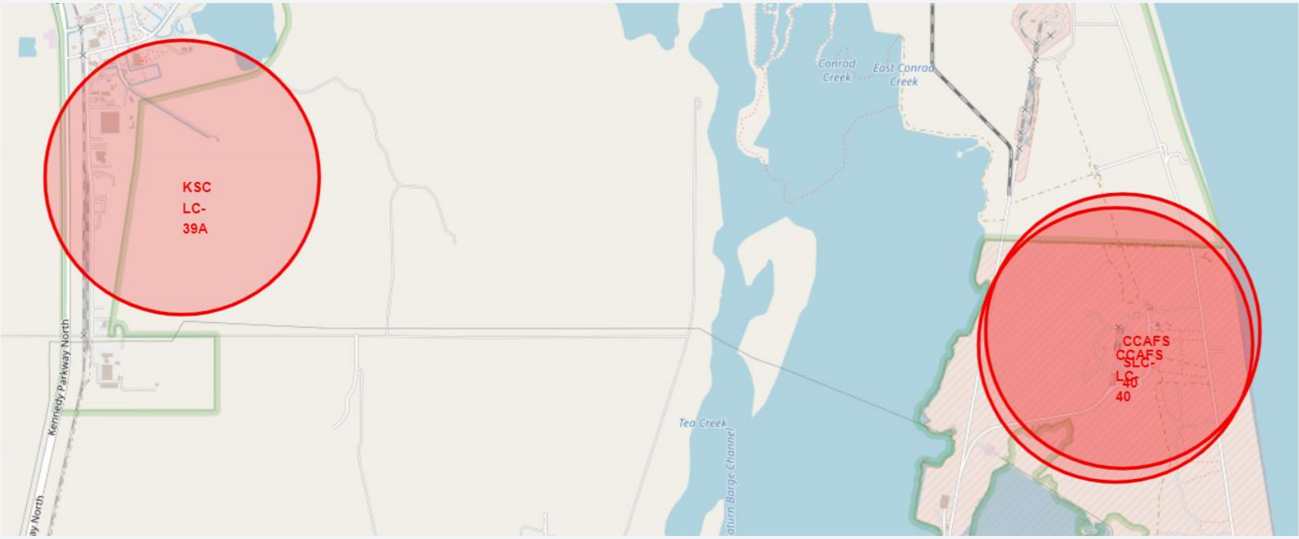
Section 3

# Launch Sites Proximities Analysis

# Launch Site Location

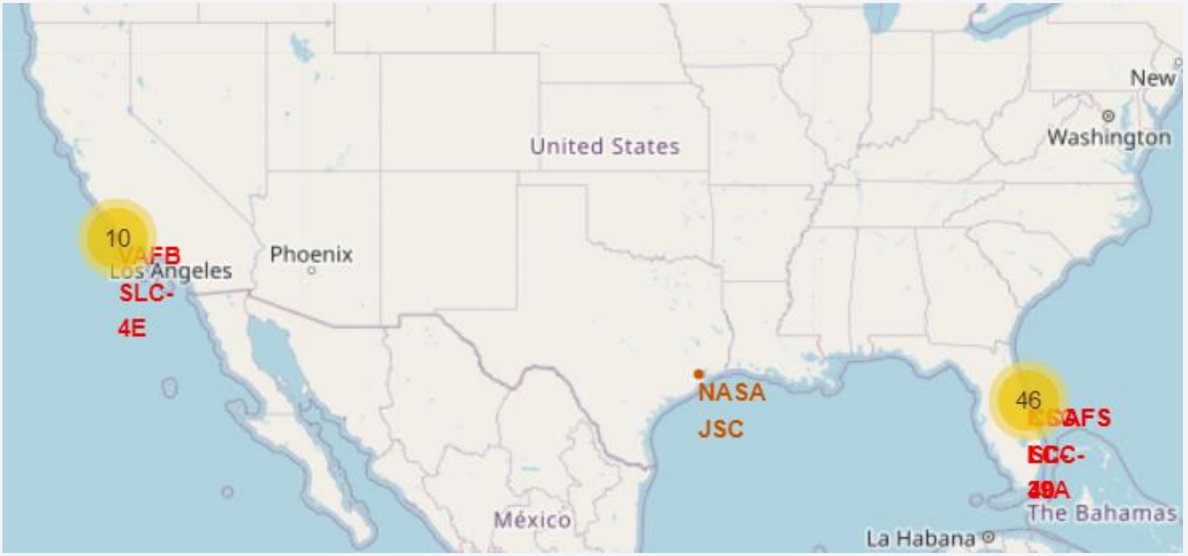
- The Folium map on the right shows the launch sites' location in relation to the NASA Johnson's Space Centre. The map on the left shows a zoomed in version of the map, focused on the group of launch sites in Florida. We can see that there are 3 launch sites in close proximity to each other in Florida, and a fourth near to Los Angeles in California. We can also see that all launch sites are close to one of the coasts.

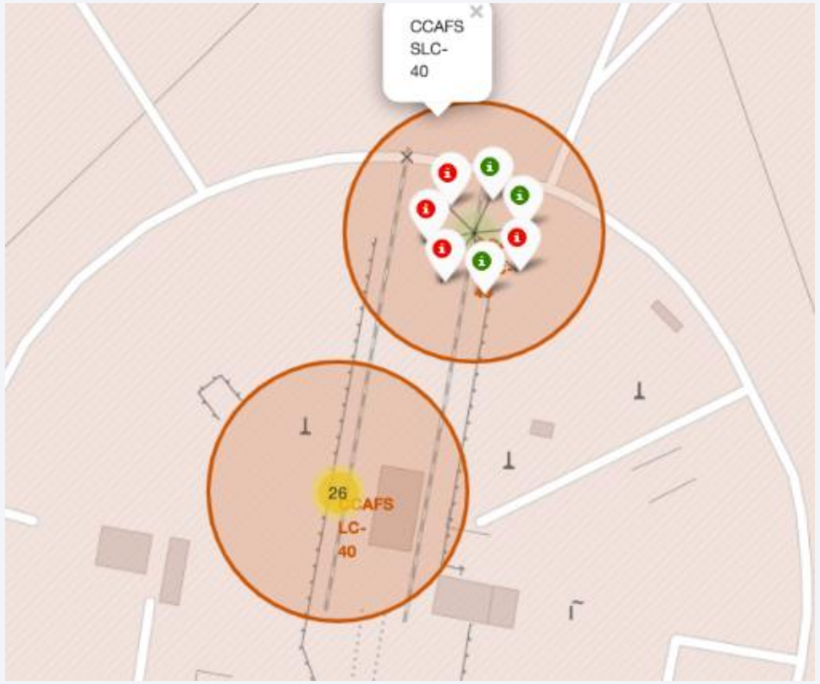




# Launch Outcomes for each Launch Site

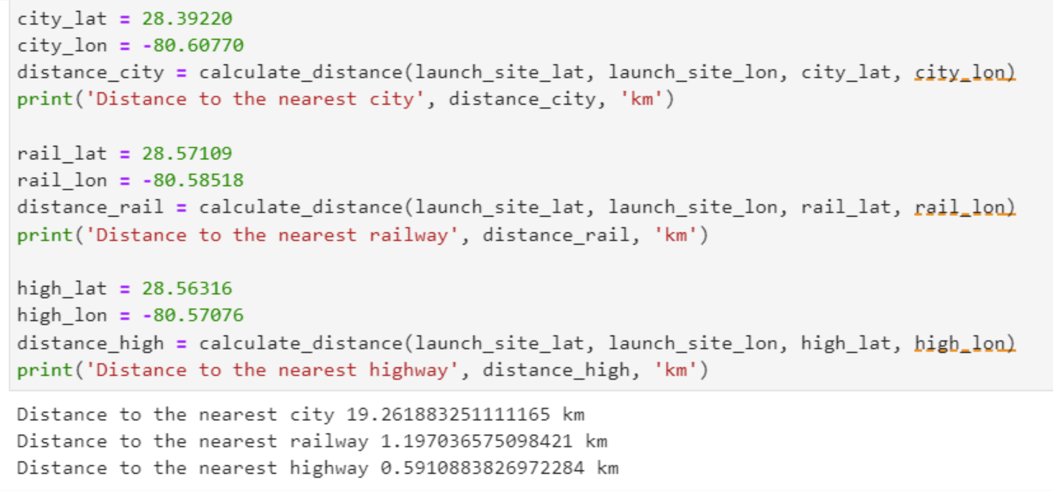
- The Folium map on the left shows the clusters (MarkerClusters) of launches in relation to the NASA Johnson’s Space Centre and the launch sites. The map on the right shows a zoomed in version of the map, focused on the group of launch sites in Florida. We can see that the successful launches have been highlighted with a green marker and the unsuccessful with a red marker. The marker cluster also shows the name of each launch site.





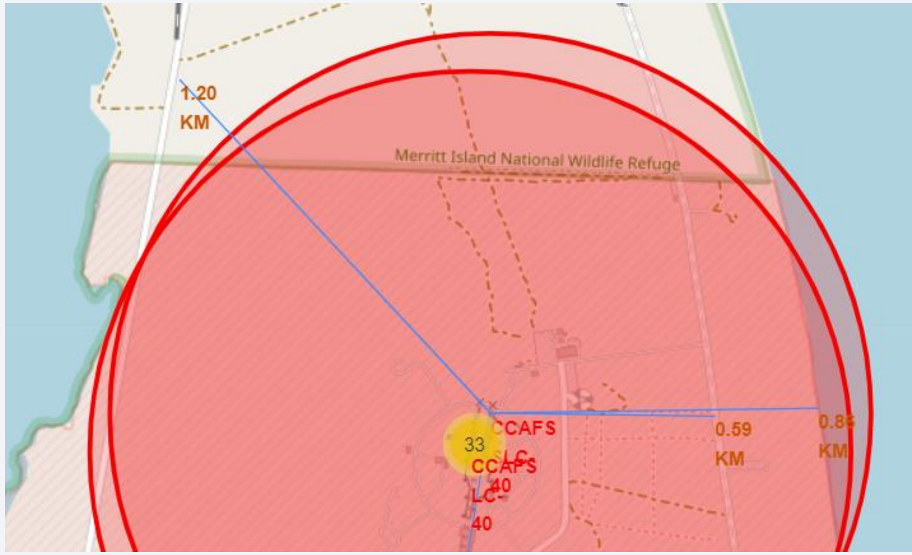
# Launch Site Proximities (CCAFS SLC-40)

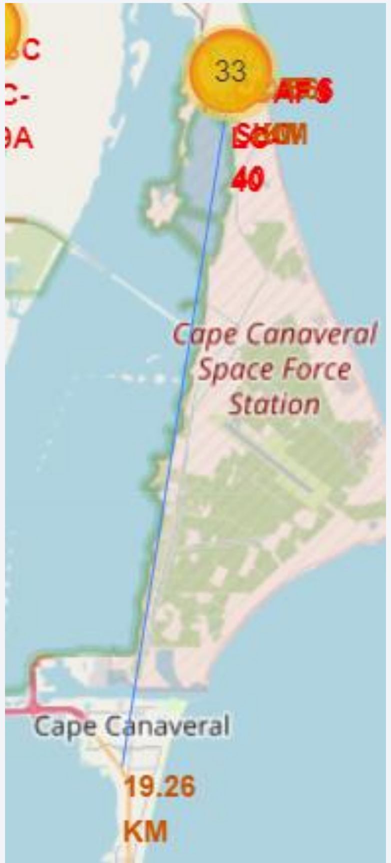
- The code in the bottom right shows the calculation of proximities using the calculate_distance function and latitude and longitude.
- The middle folium map shows the proximity of the railway, coast and highway to launch site CCAFS SLC-40 using lines (1.2km, 0.86km, and 0.59km respectively).
- The right folium map is a zoomed out version of the middle folium map to show the distance to the closest city, Cape Canaveral (19.26km) using a line.

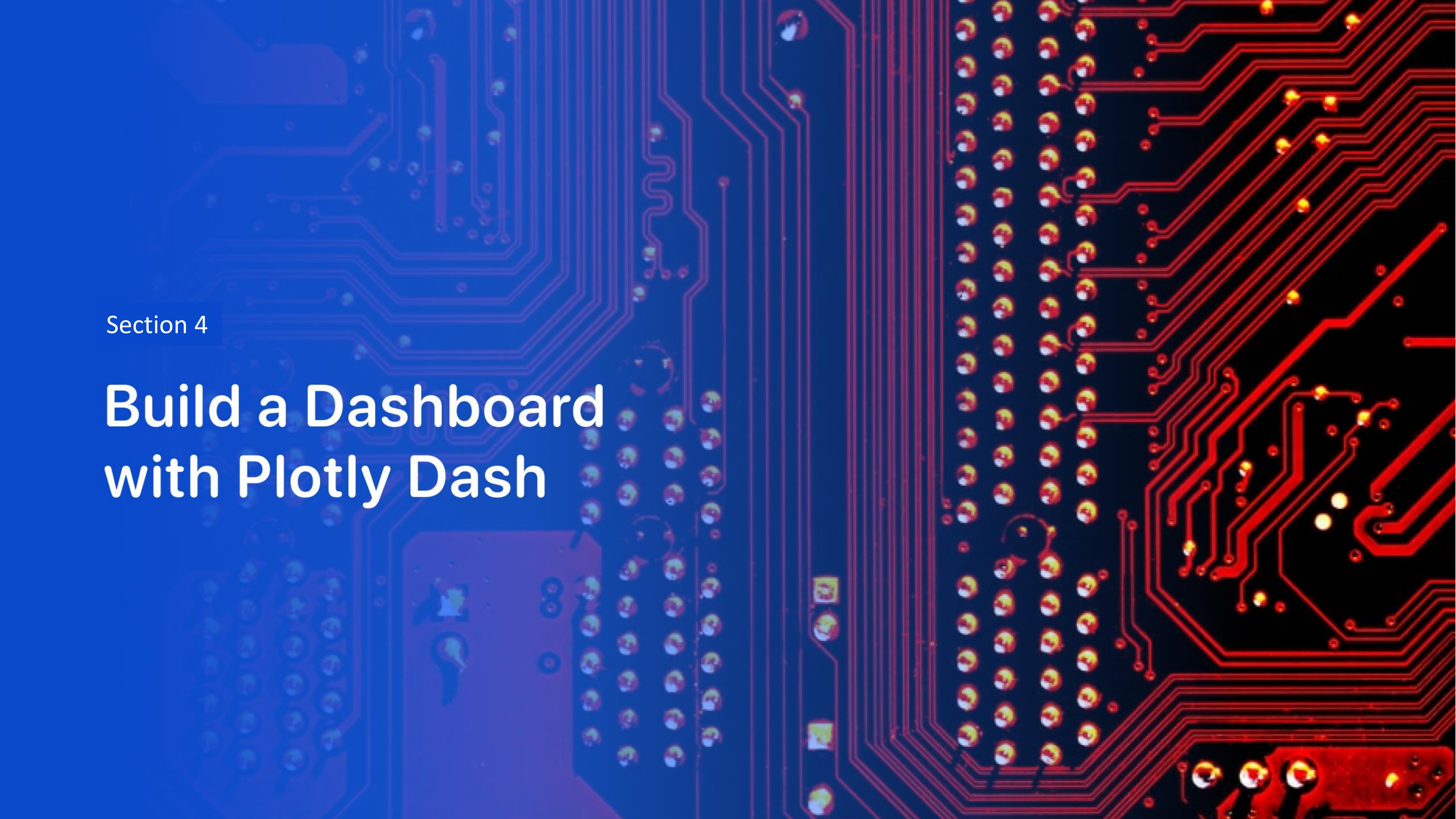


```
city_lat = 28.39220
city_lon = -80.60770
distance_city = calculate_distance(launch_site_lat, launch_site_lon, city_lat, city_lon)
print('Distance to the nearest city', distance_city, 'km')

rail_lat = 28.57109
rail_lon = -80.58518
distance_rail = calculate_distance(launch_site_lat, launch_site_lon, rail_lat, rail_lon)
print('Distance to the nearest railway', distance_rail, 'km')

high_lat = 28.56316
high_lon = -80.57076
distance_high = calculate_distance(launch_site_lat, launch_site_lon, high_lat, high_lon)
print('Distance to the nearest highway', distance_high, 'km')
```

```
Distance to the nearest city 19.261883251111165 km
Distance to the nearest railway 1.197036575098421 km
Distance to the nearest highway 0.5910883826972284 km
```





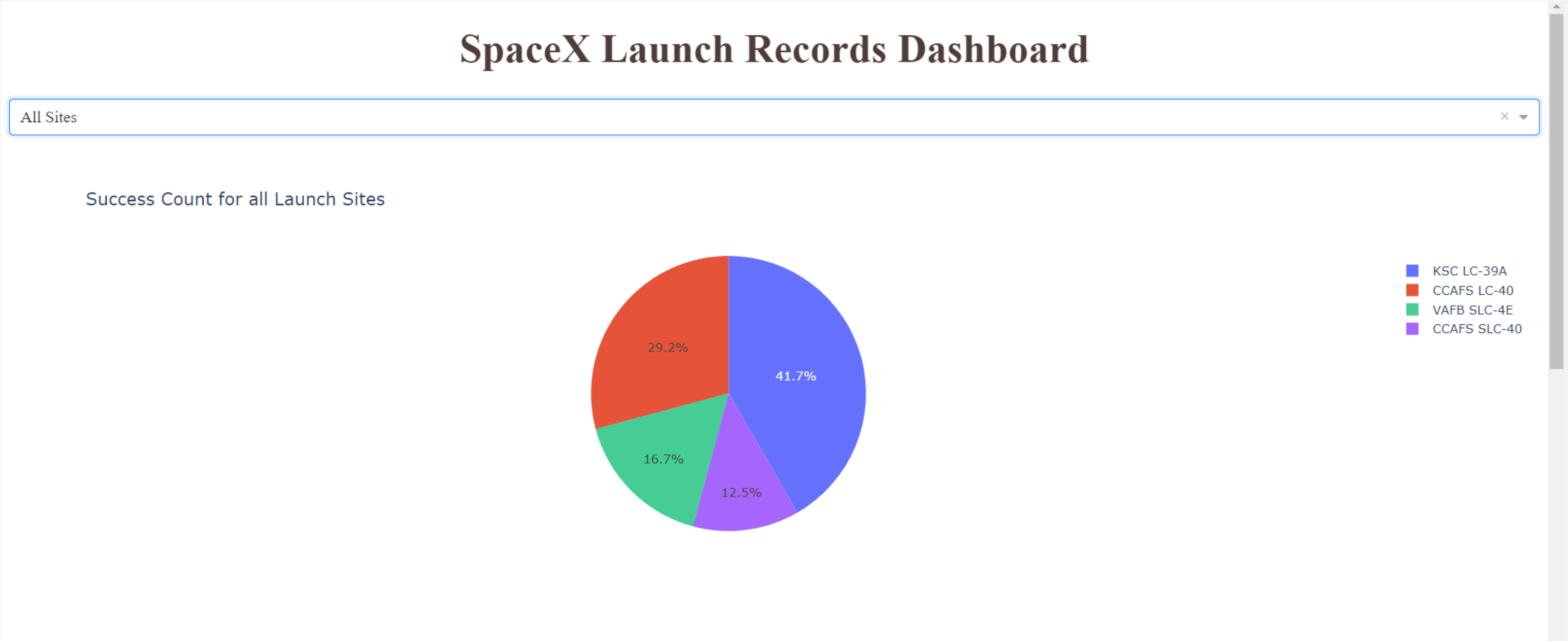


Section 4

# Build a Dashboard with Plotly Dash

# All Sites Launch Success - Piechart

- The piechart below shows the success proportion for each launch site, it shows that KSC LC-39A has the highest success proportion at 41.7% and CCAFS SLC-40 has the lowest at 12.5%.



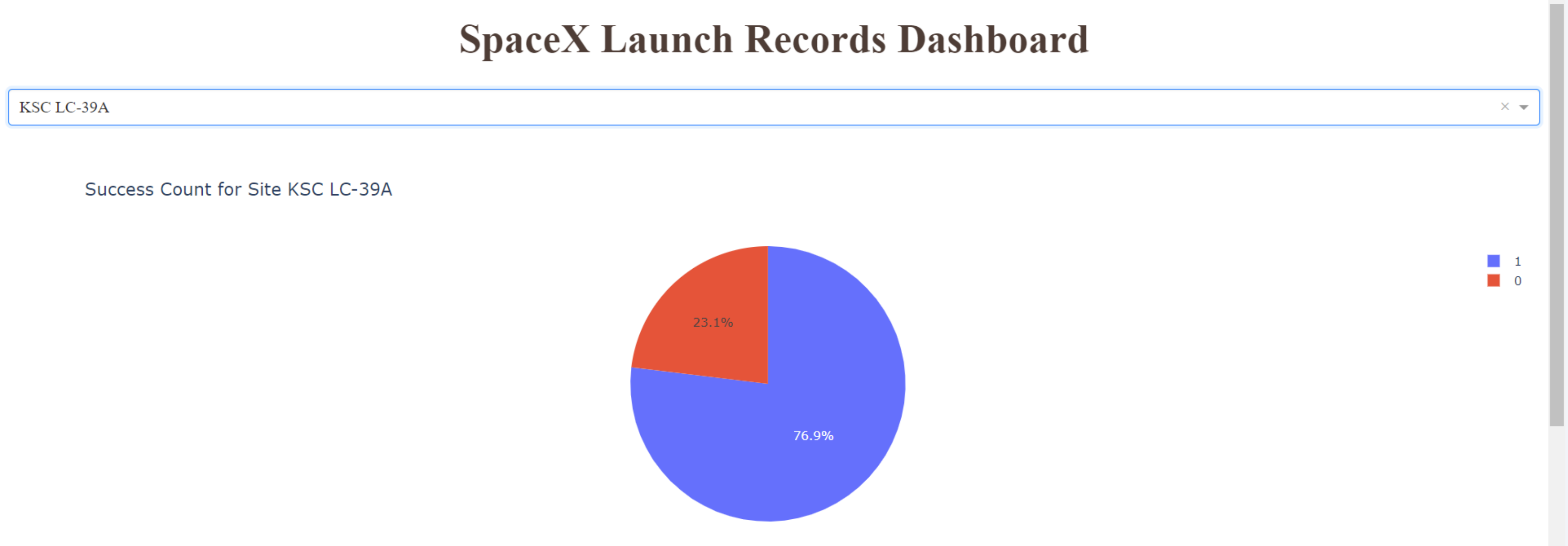
SpaceX Launch Records Dashboard

All Sites

Success Count for all Launch Sites

# Site with Highest Launch Success - Piechart

- The piechart below shows the success proportion specifically for the KSC LC-39A site which has the highest launch success. It shows that in 76.9% of launches the rocket landed successfully, with only 23.1% landing unsuccessfully.

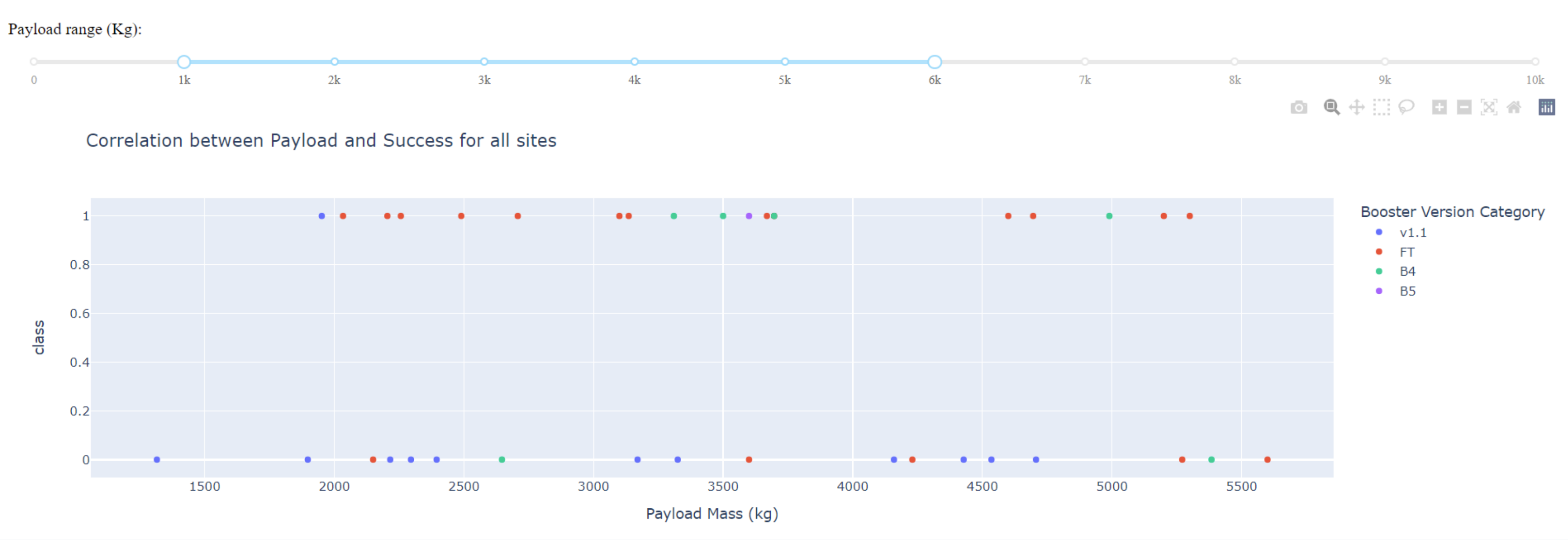


SpaceX Launch Records Dashboard

KSC LC-39A

# Payload vs. Launch Outcome Scatterplot

- The scatterplot below shows the Launch Outcome for all launch sites compared to the payload mass (filtered to only show payloads between 1000 and 6000 kg). It is also colour coded by the booster version, which shows that the FT (red) booster tends to be more successful than the v1.1 (blue) booster.

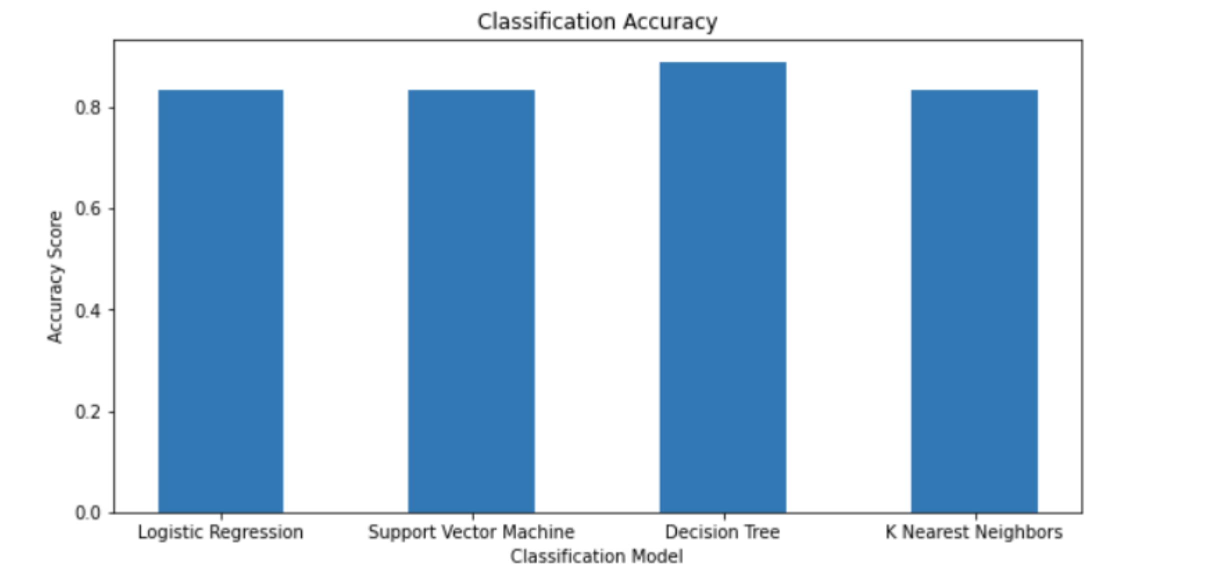


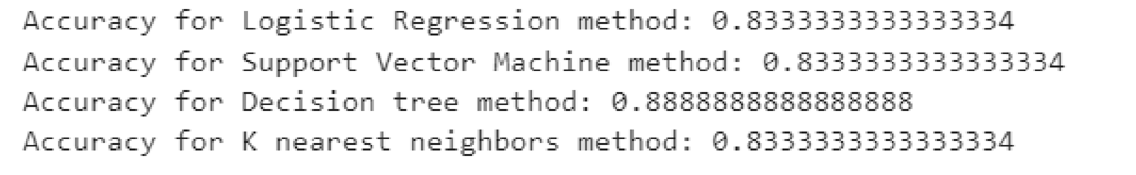
Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

- 4 Classification models were built (a logistic regression model, a Support Vector Machine, a Decision Tree and a K Nearest Neighbors model), and the hyperparameters of each of these models was optimized using a GridSearch object. Accuracy scores were then produced for each model using the testing data, from the test train split that was completed, showing that the Decision Tree has the greatest accuracy at 0.89. The bar chart shows the comparison between the accuracy of the different models, and the values on the right shows the output from the 'score' function used to test the models.

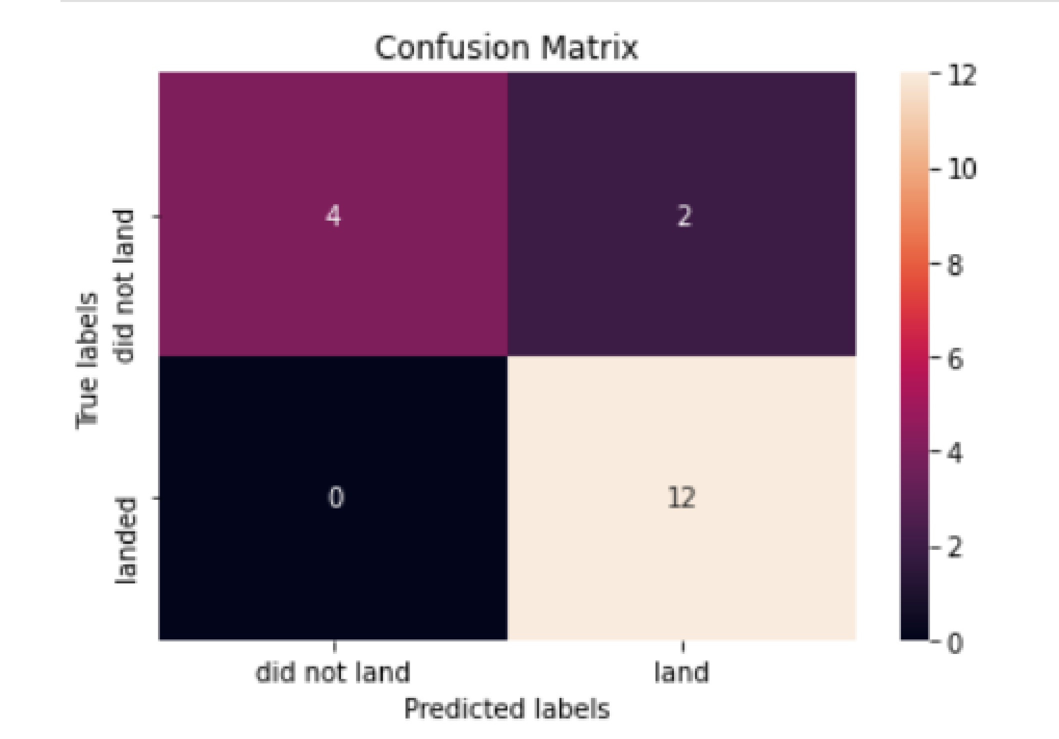




```
Accuracy for Logistic Regression method: 0.8333333333333334
Accuracy for Support Vector Machine method: 0.8333333333333334
Accuracy for Decision tree method: 0.8888888888888888
Accuracy for K nearest neighbors method: 0.8333333333333334
```

# Confusion Matrix

- The diagram below shows the confusion matrix for the highest performing model, the Decision Tree. We can see that the only issue with the model is the two False Positives that it raised (demonstrated in the upper right quadrant), however the model successfully identified all False outcomes, and is the most accuracte model for the dataset.



# Conclusions

- The folium map analysis on the data has shown that the optimal launch locations are close to the coast
- The Exploratory Data Analysis with Visualisation showed that the success rate since 2013 kept increasing till 2017, before dropping slightly in 2018, however the success rate then increased again in 2019, before dropping slightly down in 2020. Overall, the general trend of success rate has been increasing since 2013.
- The Plotly Interactive Dashboard Analysis showed that KSC LC-39A was the launch site with the greatest proportion of successful landings.
- The Machine Learning Classification techniques showed us that the best model to predict if a rocket will land would be the Decision Tree Model with hyperparameters: ‘criterion=‘gini’, ‘max_depth’=6, ‘max_features’=‘sqrt’, ‘min_samples_leaf’=4, ‘min_samples_split’=2, and ‘splitter’=‘best’.

# Appendix

- Code snippets shown throughout the Report, and notebook links given throughout for extra exploration of the reviewer.

Thank you!