IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Dhruv Singh
02-11-25

Outline

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

Executive Summary

Click to add text

- Supervised learning methods (classification) were utilized to predict the outcome of SpaceX launches, depending multiple factors; such as payload, launch site, orbit type, etc.
- The details and the outcome of each rocket launch was scraped from Wikipedia.
- In the end, a reliable Decision Tree algorithm with a 83% accuracy (with a peak accuracy of 87%) was chosen as the predictive algorithm.

Introduction

- SpaceX values one launch at $62 millions, whereas similar competitors quote upwards of $165 millions for a launch
- SpaceX can cut costs and outvalue its competitors due to the cost differences in a launch.
- SpaceX can cut so much cost because it can reuse the first stage.
- Therefore, if we can determine whether the first stage will land, we can determine the cost of a launch.
- We aim to build a classifier model that predicts the outcome of the first stage of each launch.
- Our end goal is to use this information to predict the possibility of success of a launch in case we decide to bid against SpaceX.

Section 1

Methodology

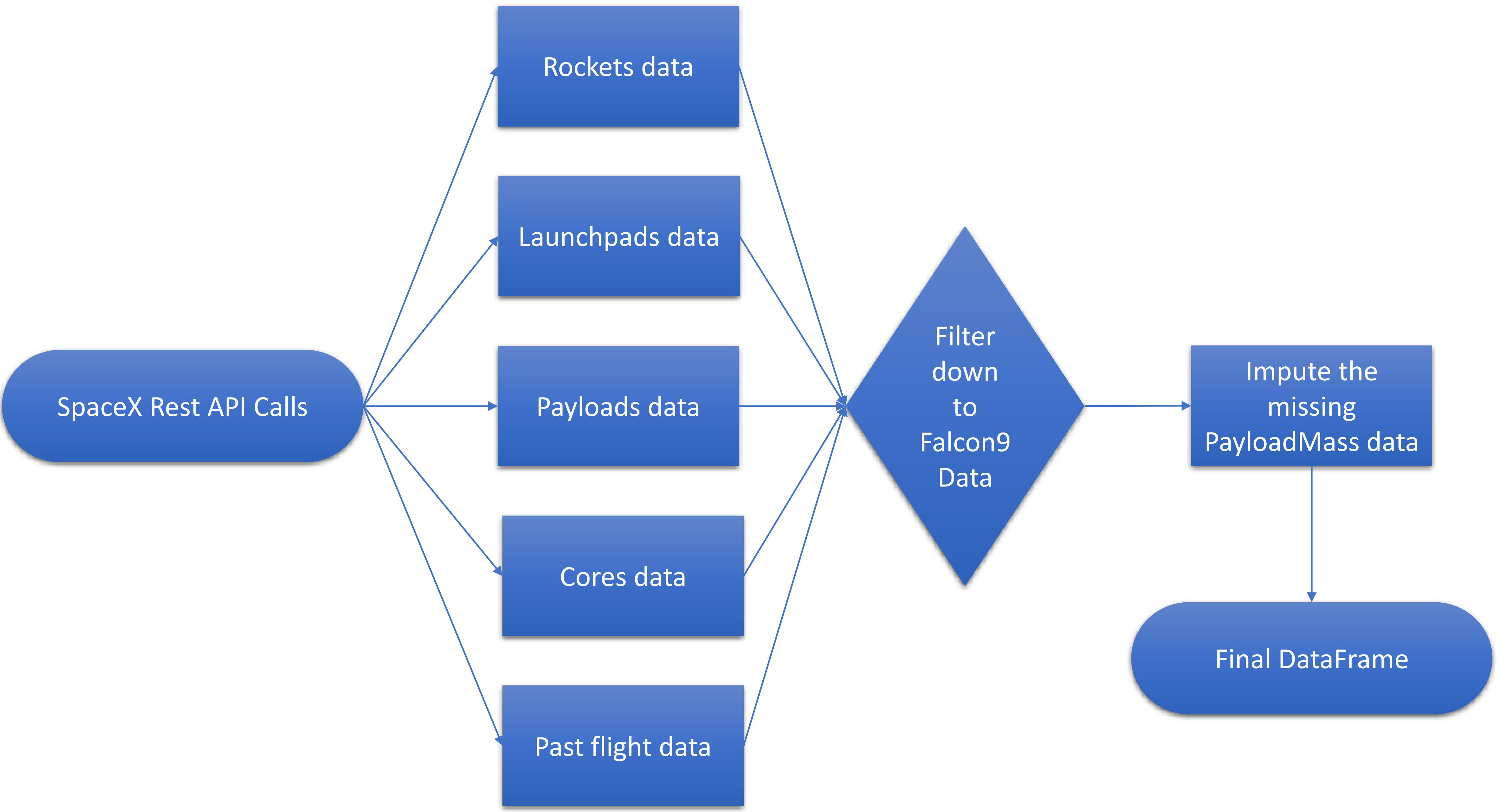
Methodology

Executive Summary

- Data collection methodology:
  - The data used in building the ML model was through calls to the SpaceX API
- Perform data wrangling
  - The obtained data was put a uniform data frame format, the outcome of each record was labeled properly
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection – SpaceX API

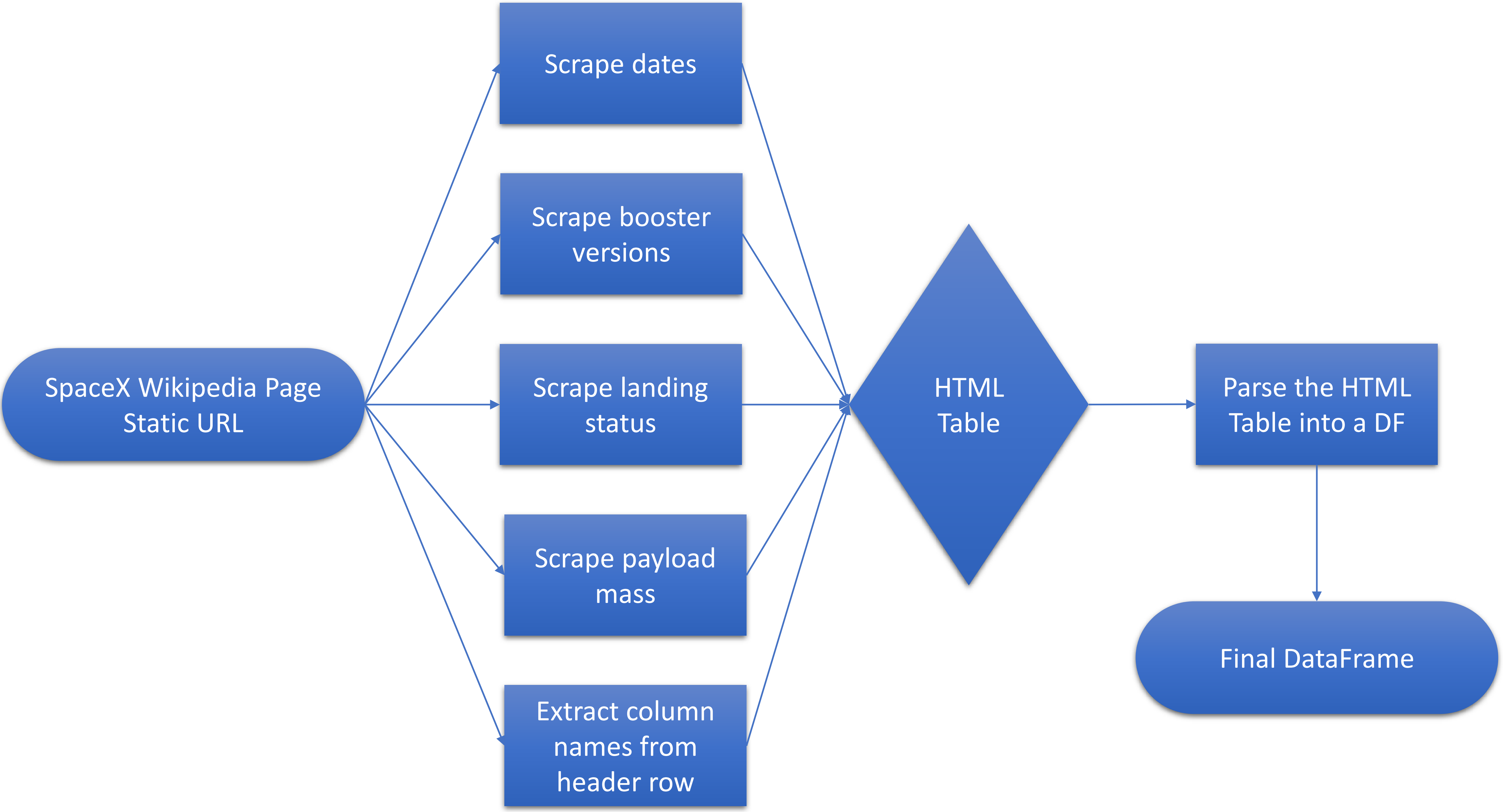
- Data was collected through making calls to the SpaceX REST API and populating a dataframe through Python functions.
- The GitHub URL of the completed SpaceX API calls notebook: https://github.com/uft93/Coursera_Capstone_Space_Y/blob/main/1%20-%20SpaceX-data-collection-api.ipyn


SpaceX Rest API Calls
Rockets data
Launchpads data
Payloads data
Cores data
Past flight data
Filter down to Falcon9 Data
Impute the missing PayloadMass data
Final DataFrame

Data Collection - Scraping

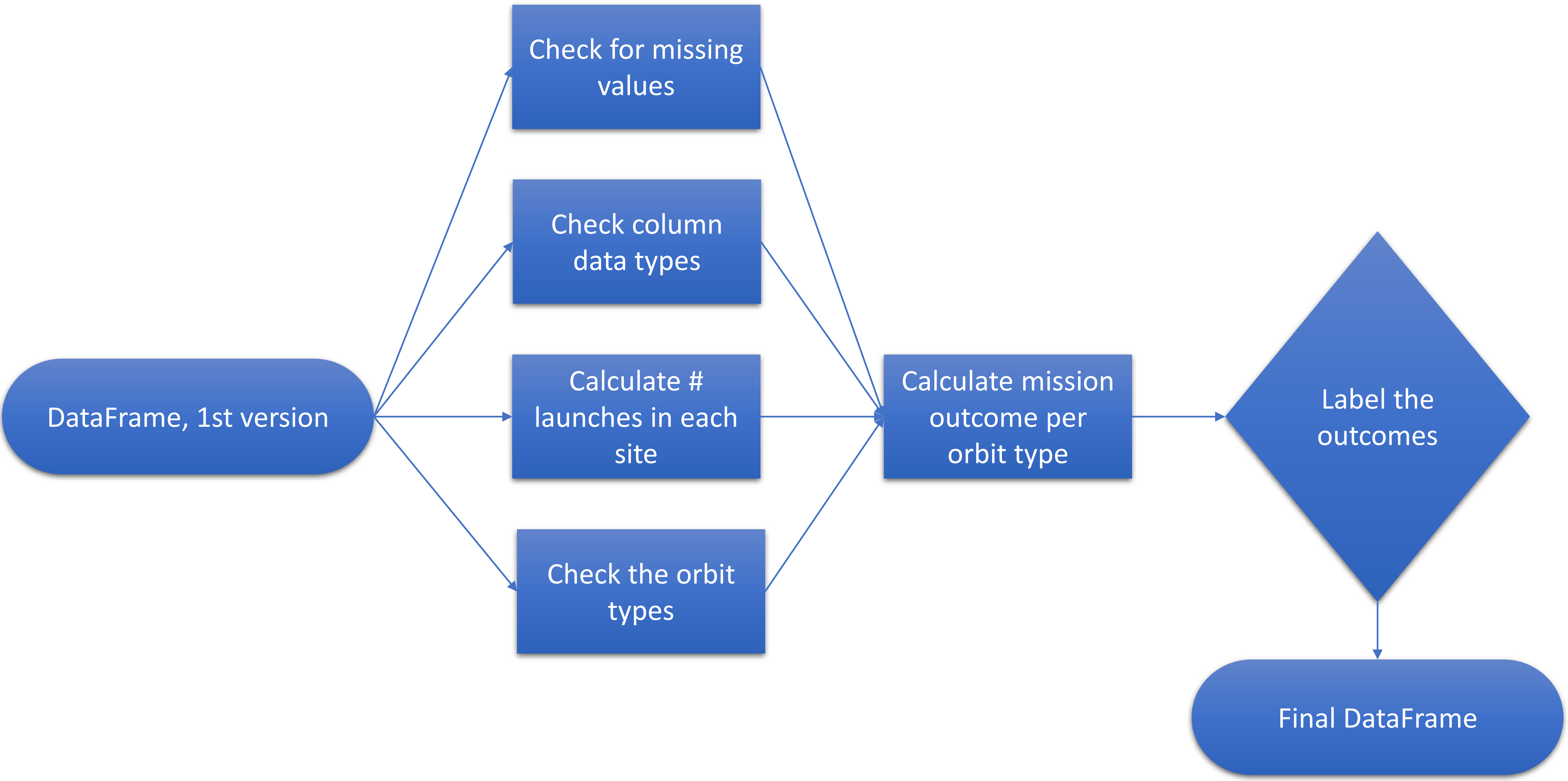
- The webscraping for SpaceX data was done through Requests and BeautifulSoup libraries. The static URL was converted to a Pandas DataFrame.

- The GitHub URL of the completed web scraping notebook: https://github.com/uft93/Coursera_Capstone_Space_Y/blob/main/2%20-%20Webscraping.ipynb


SpaceX Wikipedia Page Static URL
Scrape dates
Scrape booster versions
Scrape landing status
Scrape payload mass
Extract column names from header row
HTML Table
Parse the HTML Table into a DF
Final DataFrame

Data Wrangling

- The data was processed step by step to do several checks:
  - Check for missing values
  - Check for correct data type in each column
  - Calculating the number of launches for each launch site
  - Getting the value counts for each orbit type
- The mission outcome per orbit type was then calculated, and outcomes were labeled. This concluded the preparation of the DataFrame.
- The GitHub URL of your completed data wrangling related notebooks: https://github.com/uft93/Coursera_Capstone_Space_Y/blob/main/3%20-%20Spacex-Data%20wrangling.ipynb


DataFrame, 1st version
Check for missing values
Check column data types
Calculate # launches in each site
Check the orbit types
Calculate mission outcome per orbit type
Label the outcomes
Final DataFrame

EDA with Data Visualization

1. Categorical plot: Launch outcome based on payload mass vs. flight number
2. Categorical plot: Launch outcome based on launch site vs. flight number
3. Scatter plot: Launch outcome based on launch site vs. payload mass
4. Bar chart: Relationship between launch success rate of each orbit type

EDA with Data Visualization

5. Scatter plot: Launch outcome based on flight number vs. orbit type

6. Scatter plot: Launch outcome based payload mass vs. orbit type

7. Line plot: Launch success yearly trend

GitHub link:
https://github.com/uft93/Coursera_Capstone_Space_Y/blob/main/5%20-%20Eda-dataviz.ipynb

EDA with SQL

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

   SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;

2. Display 5 records where launch sites begin with the string ‘CCA’

   SELECT * FROM SPACEXTBL WHERE SUBSTRING(LAUNCH_SITE, 1, 3) = ‘CCA’ LIMIT 5;

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

   SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = ‘NASA (CRS)’ LIMIT 5;

EDA with SQL

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

```
        SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE
BOOSTER_VERSION = ‘F9 v1.1’;
```

5. List the date when the first successful landing outcome in ground pad was achieved

```
        SELECT MIN(DATE) FROM SPACEXTBL WHERE LANDING__OUTCOME =
‘Success (ground pad)’
```

EDA with SQL

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

```
SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE
LANDING__OUTCOME = ‘Success (drone ship)’ AND PAYLOAD_MASS__KG_
BETWEEN 4000 AND 6000;
```

7. List the total number of successful and failure mission outcomes

```
SELECT COUNT(*) FROM SPACEXTBL WHERE MISSON_OUTCOME LIKE
‘Success%;
```

```
SELECT COUNT(*) FROM SPACEXTBL WHERE MISSON_OUTCOME LIKE
‘Failure%;
```

EDA with SQL

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

SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL;

9. List the failed landing_outcomes in drıne ship, their booster versions, and launch site names for in year 2015

SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE

WHERE LANDING__OUTCOME = ‘Failure (drone ship)’ AND YEAR(DATE) = 2015;

EDA with SQL

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

```
        SELECT COUNT(LANDING__OUTCOME) AS COUNT,
LANDING__OUTCOME
        FROM (SELECT * FROM SPACEXTBL WHERE DATE BETWEEN ‘2010-06-
04’ AND ‘2017-03-20’)
        GROUP BY LANDING__OUTCOME
        ORDER BY COUNT DESC;
```

GitHub link:
https://github.com/uft93/Coursera_Capstone_Space_Y/blob/main/4%20-%20Eda-sql-coursera.ipynb

Build an Interactive Map with Folium

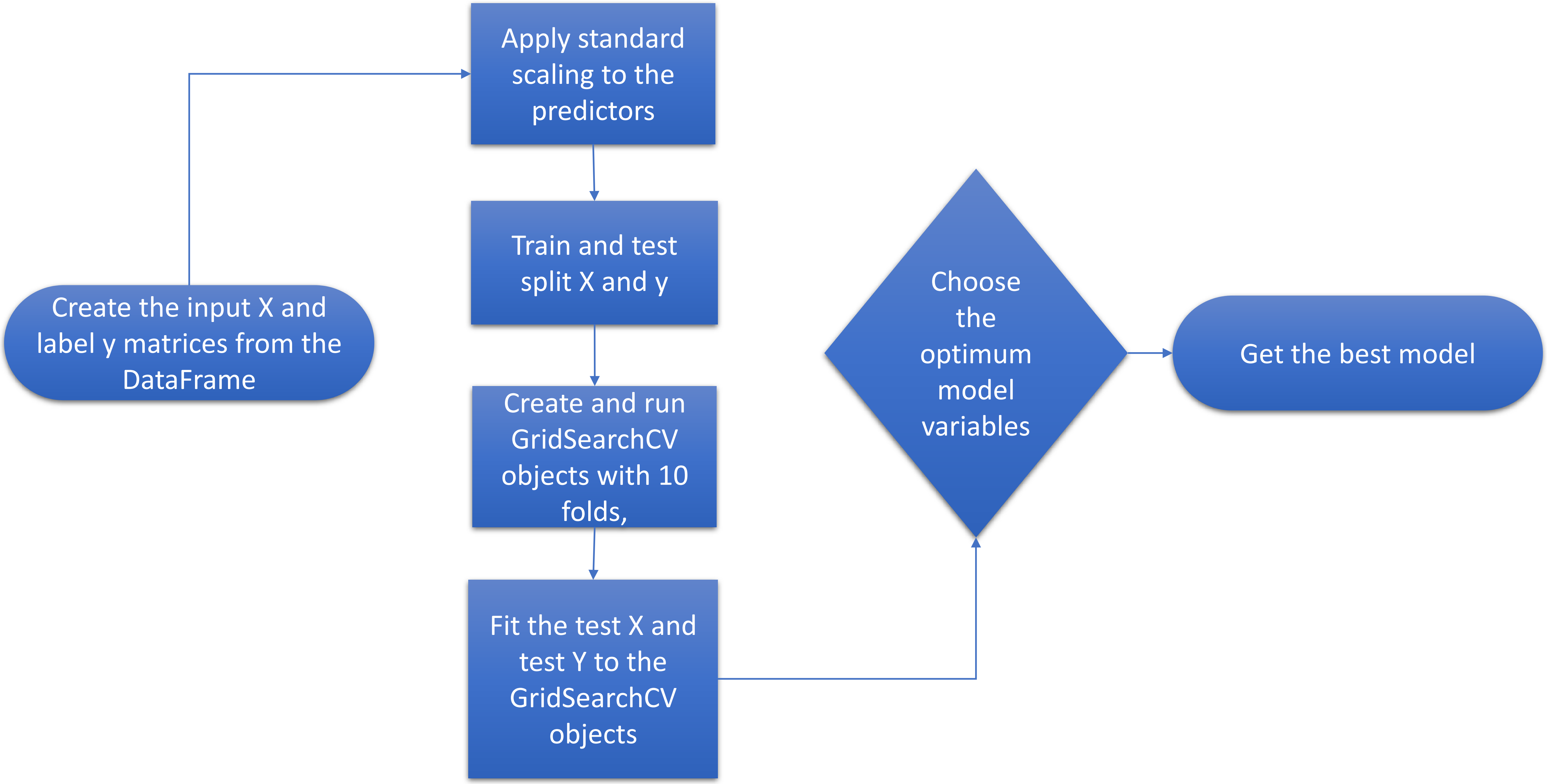
- Map 1: Folium Marker and Circle objects added to the basemap to denote the location of unique launch sites in California and Florida.
- Map 2: Folium MarkerCluster object added to the basemap to display the launch outcome for each site.
- Map 3: Folium MousePosition gets map coordinates for a Mouse over a point on the map. PolyLine object was added to illustrate the distance calculated between the closest railroad and the launch site SLC-40.
- GitHub URL: https://github.com/uft93/Coursera_Capstone_Space_Y/blob/main/6%20-%20Launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- The dashboard consists of two interactive plots: a pie chart on the upper level and a scatter plot on the lower level
- The charts are done on a launch site basis. If no launch site is selected, the pie chart will display success count for all launch sites. If a launch site is selected, it will display total success launches for that launch site.
- The scatter plot displays the relationship between payload mass and the launch outcome. If no launch site is selected, it will display all outcomes for all booster types. If a launch site is selected, it will narrow down to the booster version category used in that launch site.
- GitHub URL: https://github.com/uft93/Coursera_Capstone_Space_Y/blob/main/7%20-%20Dashboards.ipynb

Predictive Analysis (Classification)

- The clean dataframe was separated into predictor matrix X and labels y.
- After scaling the X and splitting both the X and y using sklearn library, one GridSearchCV object was created for each different classification algorithm.
- These objects were fit with the training data. Optimal parameters for each classification algorithm was then extracted from corresponding GridSearchCV object.
- Scores of different GridSearchCV objects were stored in a Python list and the highest one and the corresponding algorithm (Decision Tree) was chosen.
- GitHub URL: https://github.com/uft93/Coursera_Capstone_Space_Y/blob/main/8%20-%20SpaceX_Machine%20Learning%20Prediction.ipynb

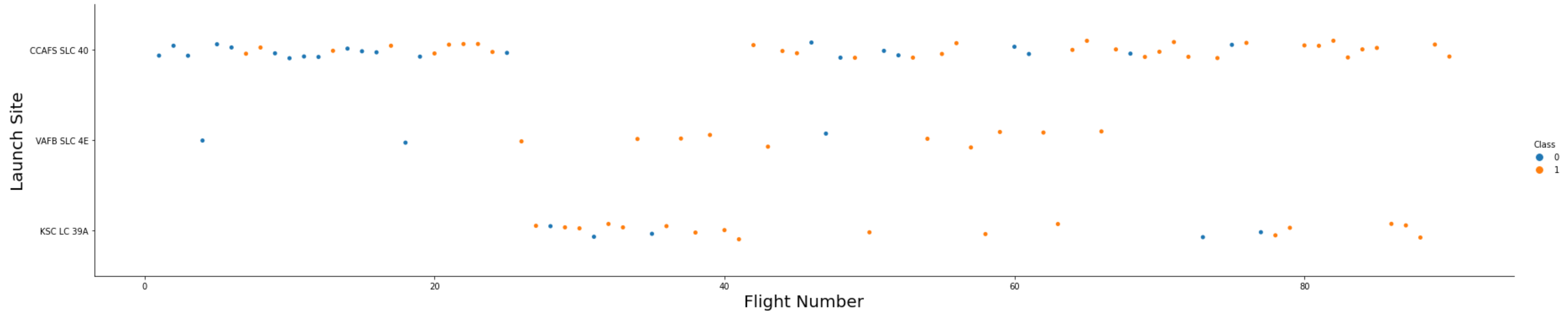

Create the input X and label y matrices from the DataFrame
Apply standard scaling to the predictors
Train and test split X and y
Create and run GridSearchCV objects with 10 folds,
Fit the test X and test Y to the GridSearchCV objects
Choose the optimum model variables
Get the best model

Results

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

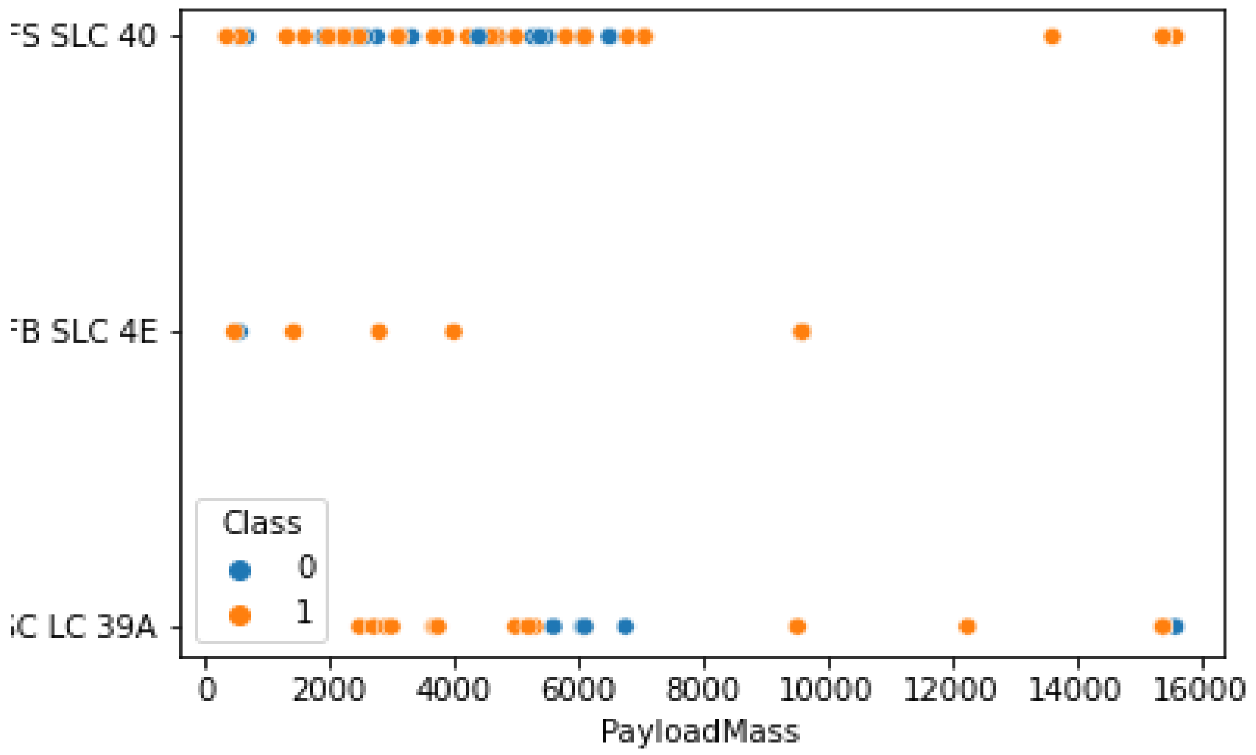
Section 2

Insights drawn from EDA



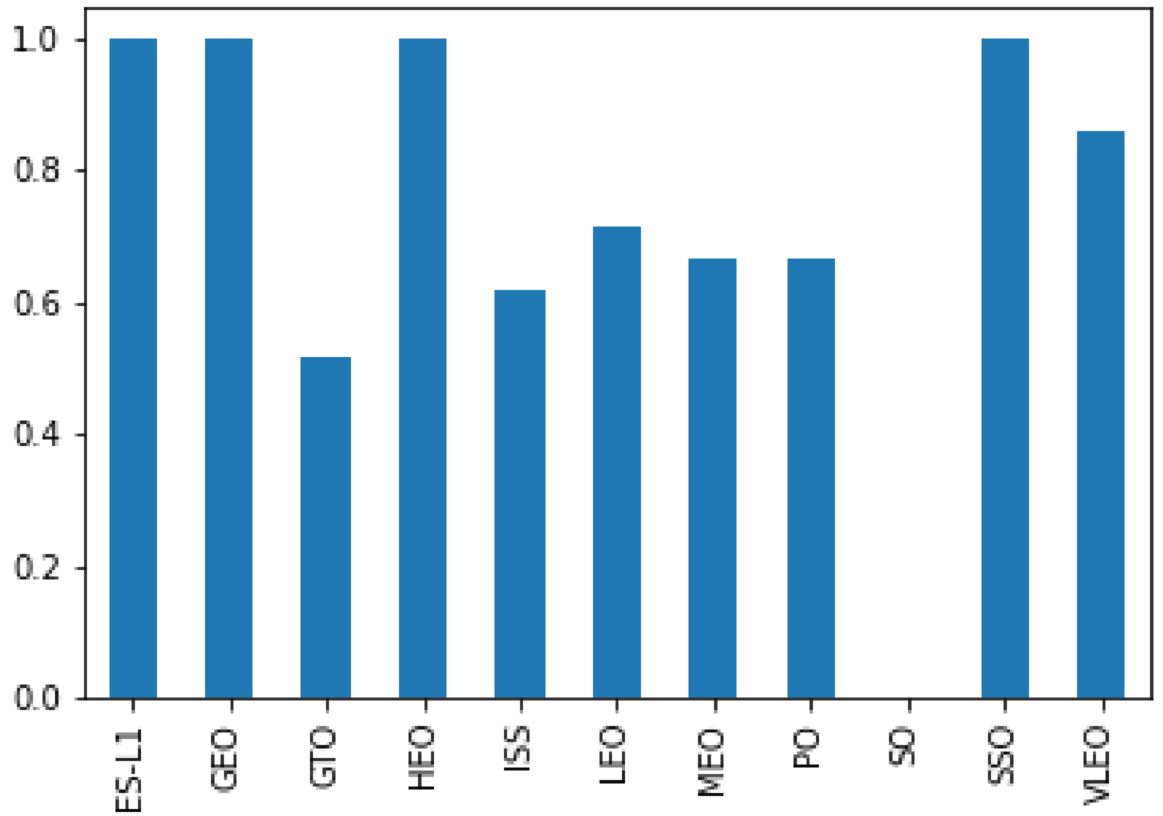
Flight Number vs. Launch Site

- Launch site CCAFS SLC 40 is by far the most used site and its success chance seems to improve with the increasing flight numbers. This flight number vs. success chance correlation seems to hold for VAFB SLC 4E too. It’s hard to draw this type of relationship for KSC LC 39A.



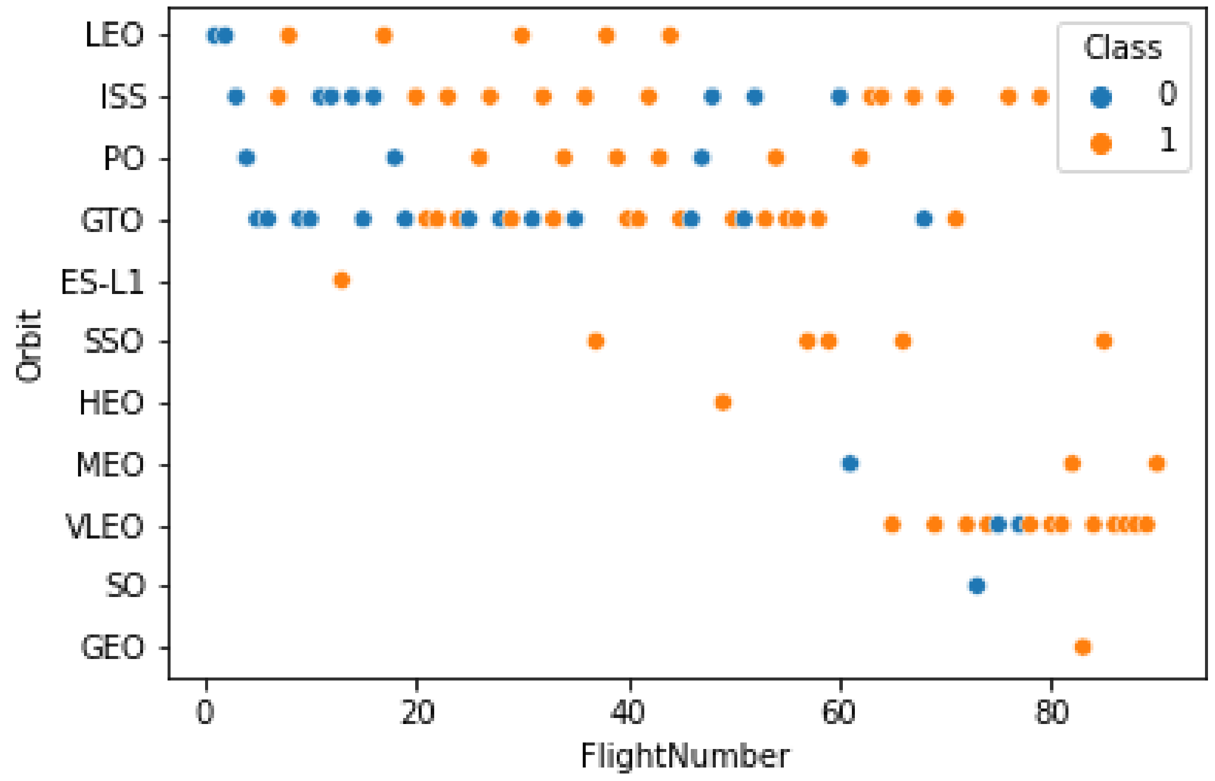
Payload vs. Launch Site

- There seems to be no relationship between launch site vs. payload mass before 8 tons of payload – and for VAFB SLC 4E, correlation between to factors is almost non-existent. For the other two stations, however, beyond 8 tons of payload mass, the success chance seems to increase.



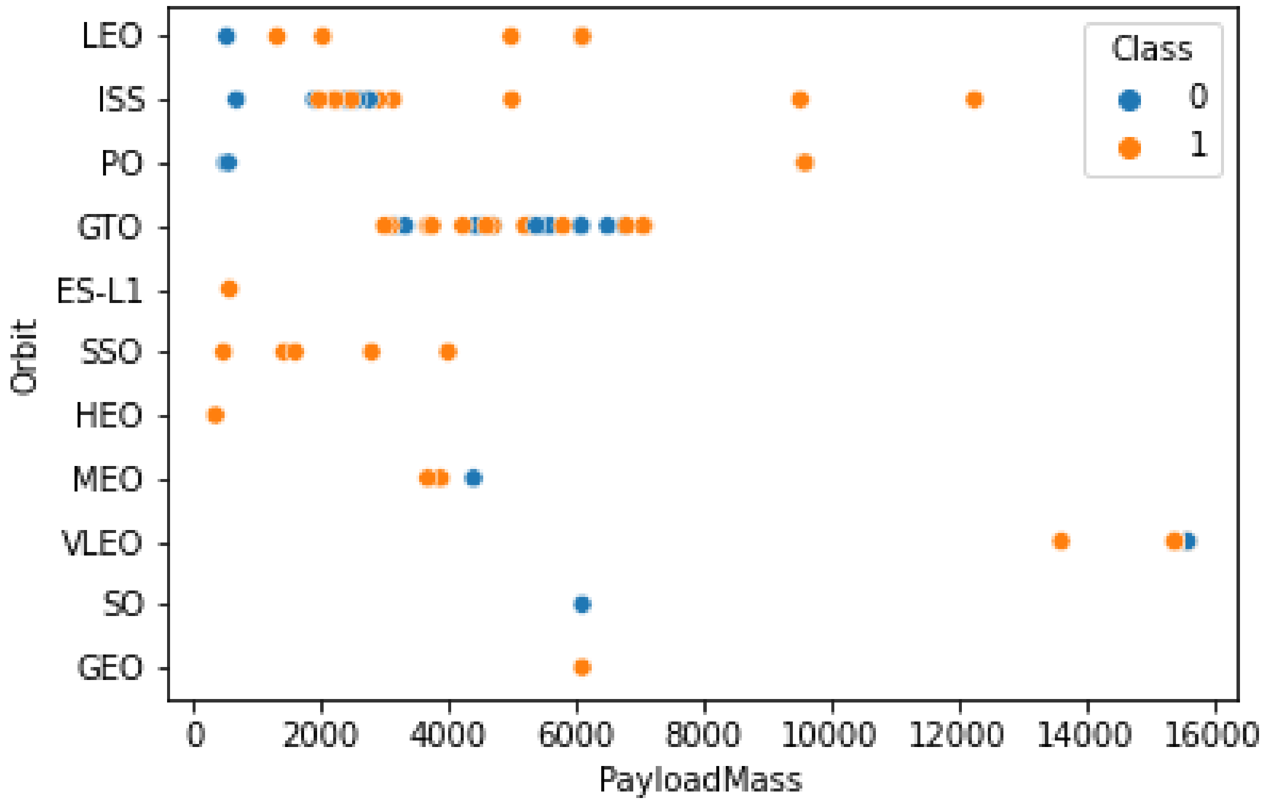
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, and SSO are all perfect candidates. GTO should be avoided (success rate 51%).



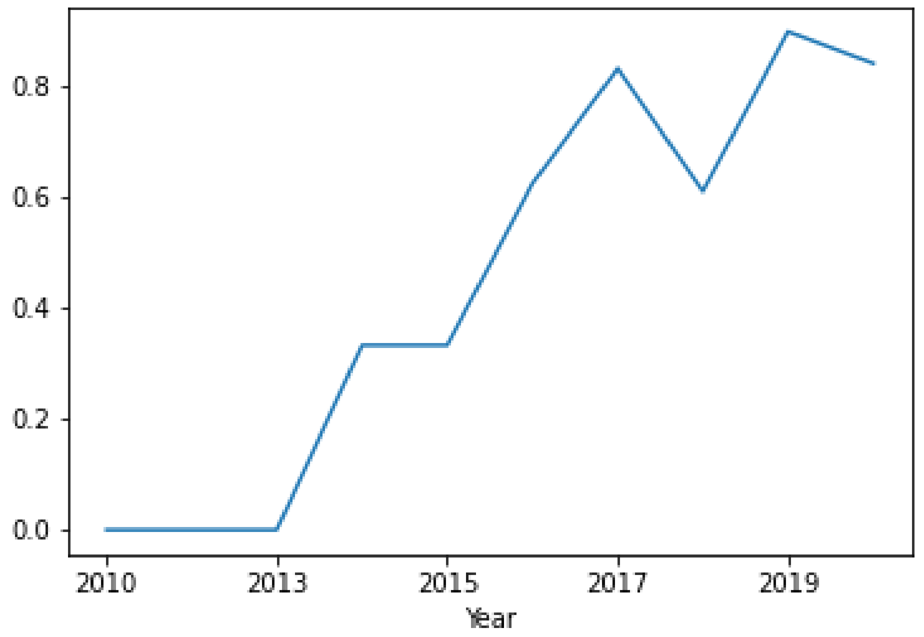
Flight Number vs. Orbit Type

- For the orbit types with adequate number of samples and an establishable correlation with flight number, success chance seems to increase with flight number. Examples: LEO, VLEO.
- No correlation examples: GTO, ISS
- SSO seems to be very successful but the sample size is rather small.



Payload vs. Orbit Type

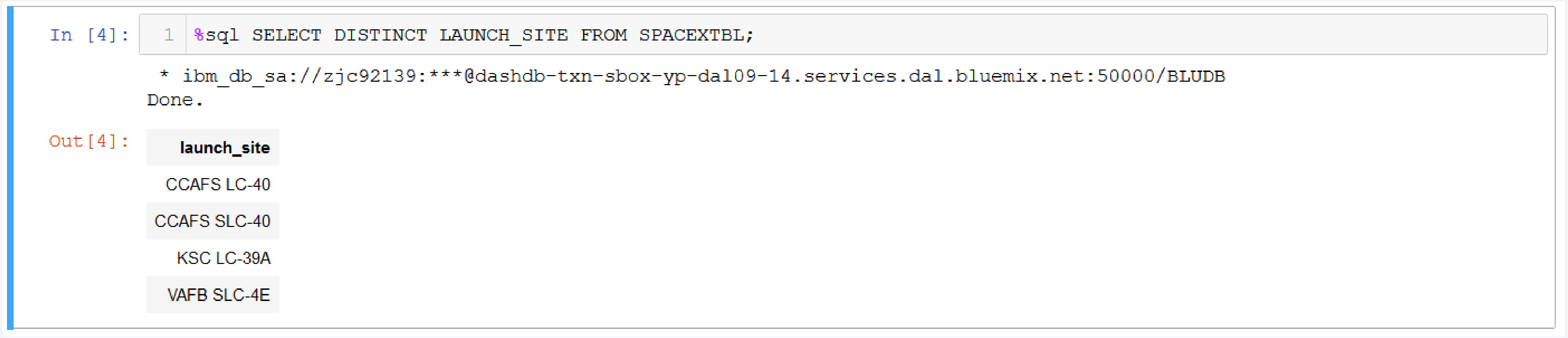
- For LEO and ISS, there's an apparent correlation between payload mass and success chance. A correlation for GTO seems to be hard to establish, and the sample size for other orbit types is rather small.



Launch Success Yearly Trend

- The overall success trend is very satisfactory. After and almost 2-year stagnation period, the success rate quickly reached 80%. Although a small dip followed the following year, the upward trend continued into the year 2020.

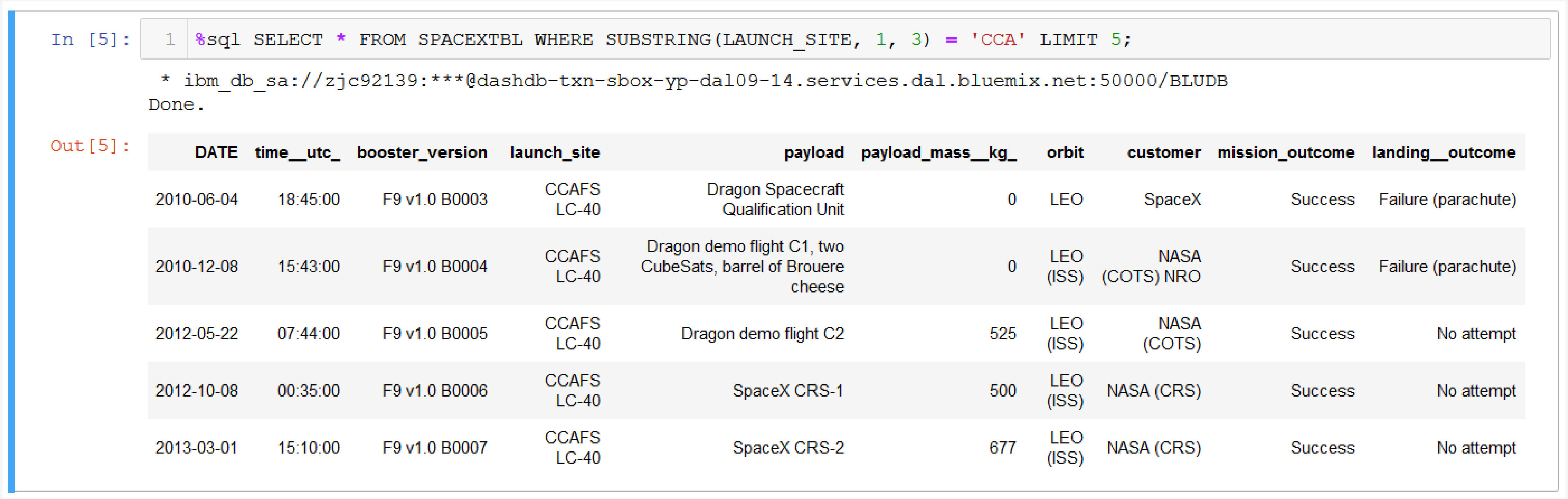
All Launch Site Names



```
In [4]:  %sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;

 * ibm_db_sa://zjc92139:***@dashdb-txn-sbox-yp-dal09-14.services.dal.bluemix.net:50000/BLUDB
Done.

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

- SELECT DISTINCT statement is used to pull only the unique launch site names.

Launch Site Names Begin with 'CCA'



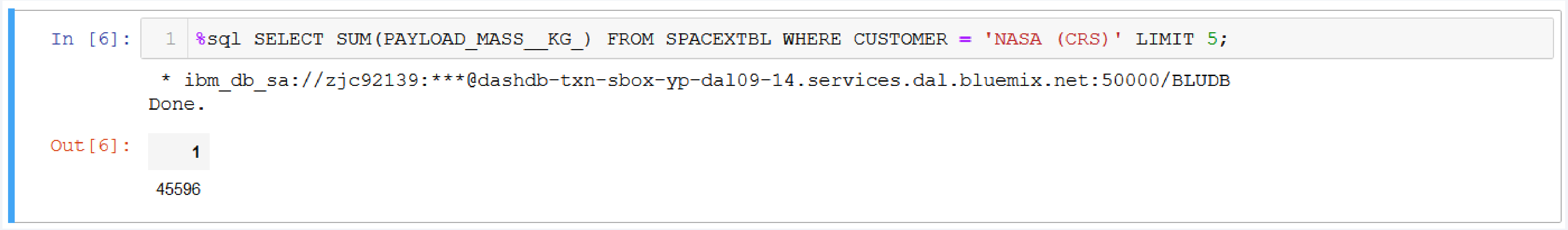
```
In [5]:  %sql SELECT * FROM SPACEXTBL WHERE SUBSTRING(LAUNCH_SITE, 1, 3) = 'CCA' LIMIT 5;

 * ibm_db_sa://zjc92139:***@dashdb-txn-sbox-yp-dal09-14.services.dal.bluemix.net:50000/BLUDB
Done.
```

Out[5]:

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

- I used the SUBSTRING function in this query to filter the launch sites down to ones that start with the substring ‘CCA’.

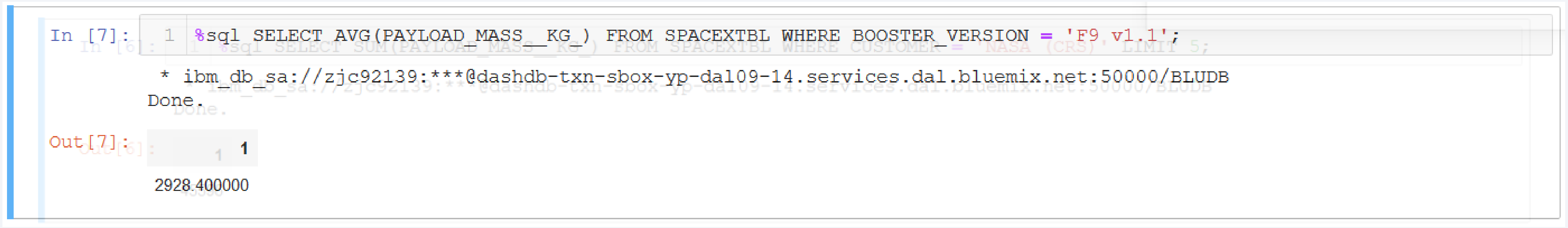
Total Payload Mass



```
In [6]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)' LIMIT 5;

 * ibm_db_sa://zjc92139:***@dashdb-txn-sbox-yp-dal09-14.services.dal.bluemix.net:50000/BLUDB
Done.

Out[6]:     1
        45596
```

- SUM function in conjunction with WHERE statement (to filter the customer down to NASA) makes the query calculate the total payload mass carried by boosters launched by NASA (CRS).

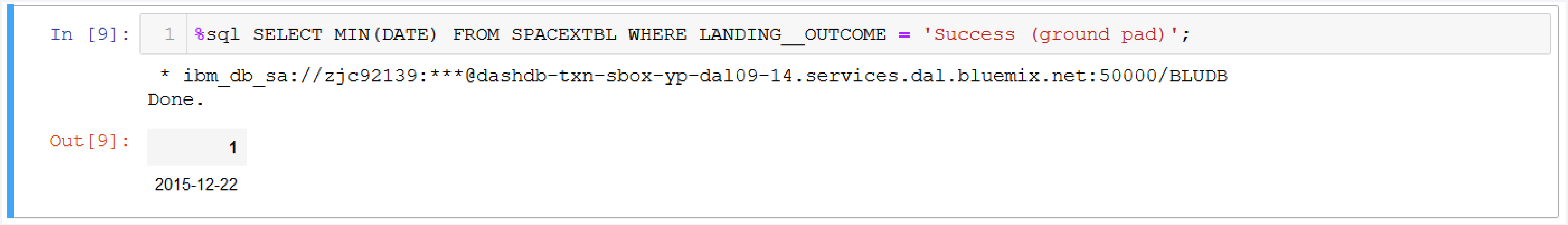
Average Payload Mass by F9 v1.1



```
In [7]:  %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';

          * ibm_db_sa://zjc92139:***@dashdb-txn-sbox-yp-dal09-14.services.dal.bluemix.net:50000/BLUDB
         Done.

Out[7]:          1
         2928.400000
```

- This query is very similar to the previous one. One needs to use AVG function instead of SUM. The result is the average payload mass carried by booster version F9 v1.1.

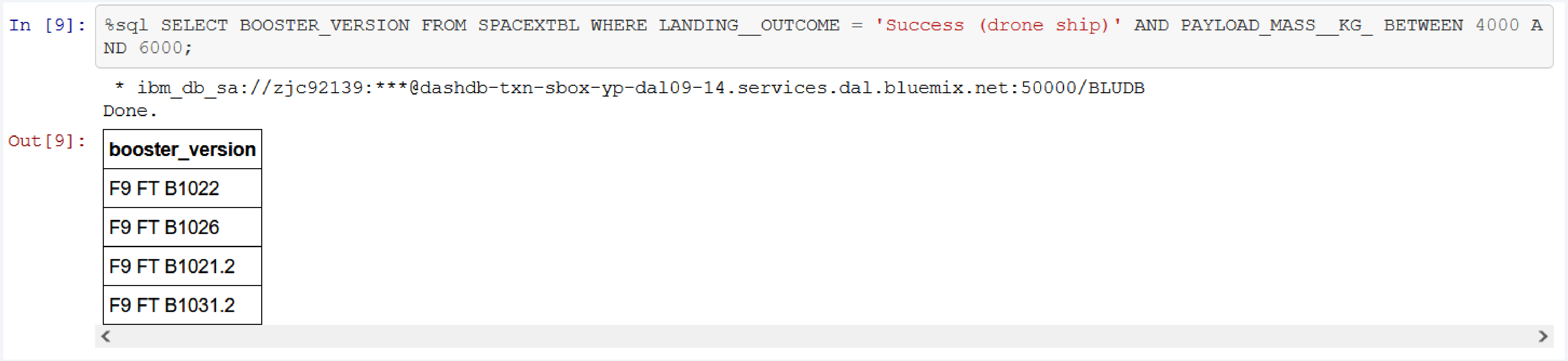
First Successful Ground Landing Date



```
In [9]:  %sql SELECT MIN(DATE) FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (ground pad)';

 * ibm_db_sa://zjc92139:***@dashdb-txn-sbox-yp-dal09-14.services.dal.bluemix.net:50000/BLUDB
Done.

Out[9]:          1
        2015-12-22
```

- I used MIN function in conjunction with a WHERE statement in this query to retrieve the date when the first successful landing outcome in ground pad achieved.

Successful Drone Ship Landing with Payload between 4000 and 6000

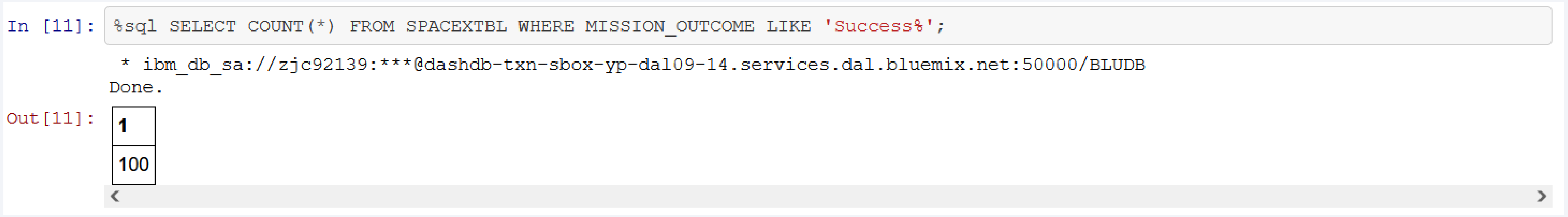


```
In [9]: %sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 A
        ND 6000;

 * ibm_db_sa://zjc92139:***@dashdb-txn-sbox-yp-dal09-14.services.dal.bluemix.net:50000/BLUDB
Done.
Out[9]:
```

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

- I used the BETWEEN operator in combination with a WHERE statement in this query to list the names of the boosters which have success in drone ship landing with a payload mass between 4000 and 5000 kg's.

Total Number of Successful and Failure Mission Outcomes

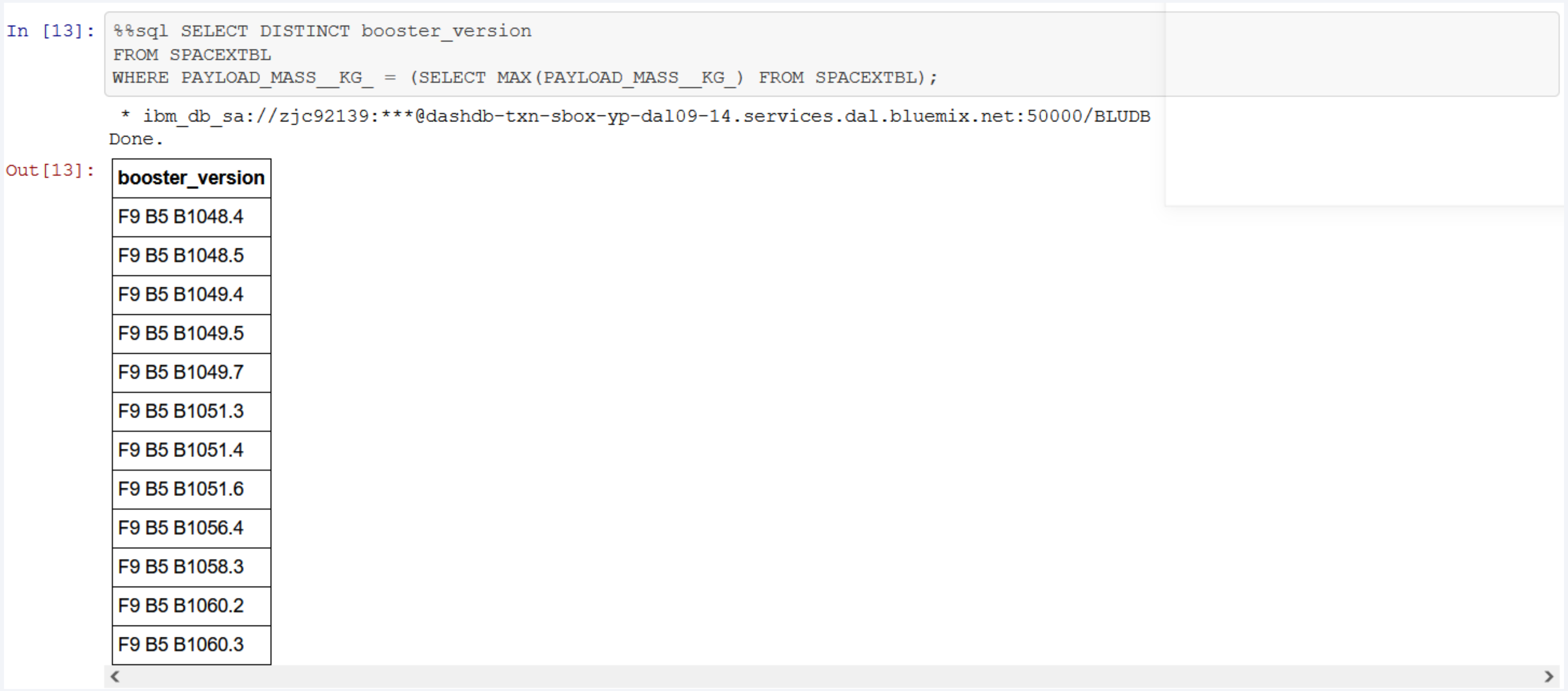


```
In [11]: %sql SELECT COUNT(*) FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Success%';

          * ibm_db_sa://zjc92139:***@dashdb-txn-sbox-yp-dal09-14.services.dal.bluemix.net:50000/BLUDB
         Done.
Out[11]:
```

1
100

- I've used a simple COUNT function with LIKE operator to get the total number of successful mission outcomes. I haven't included the failure missions in this presentation to keep things short; one just needs to change 'Success%' to 'Failure%'.

Boosters Carried Maximum Payload

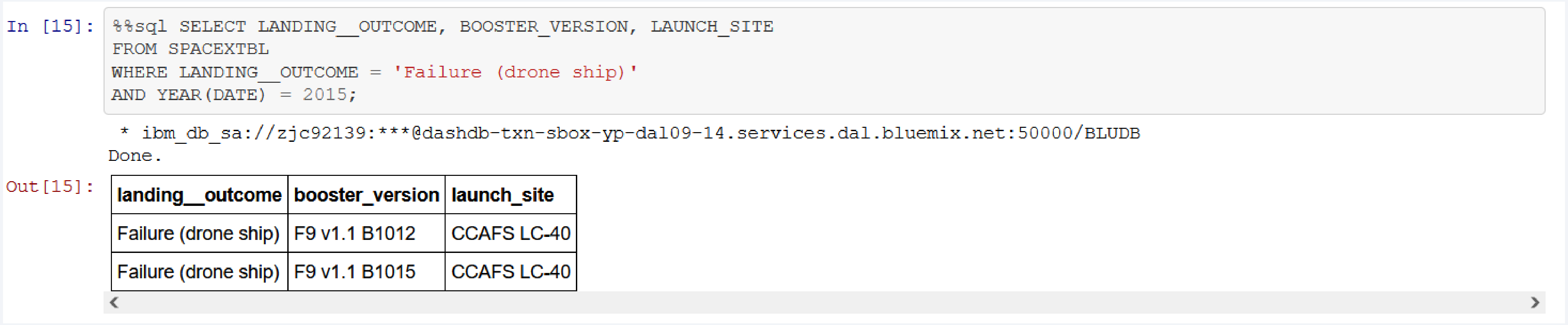


```
In [13]: %%sql SELECT DISTINCT booster_version
         FROM SPACEXTBL
         WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);

          * ibm_db_sa://zjc92139:***@dashdb-txn-sbox-yp-dal09-14.services.dal.bluemix.net:50000/BLUDB
         Done.
Out[13]:
```

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

- This one required a subquery for the WHERE clause. Everything else is in SELECT-WHERE-FROM format.

2015 Launch Records

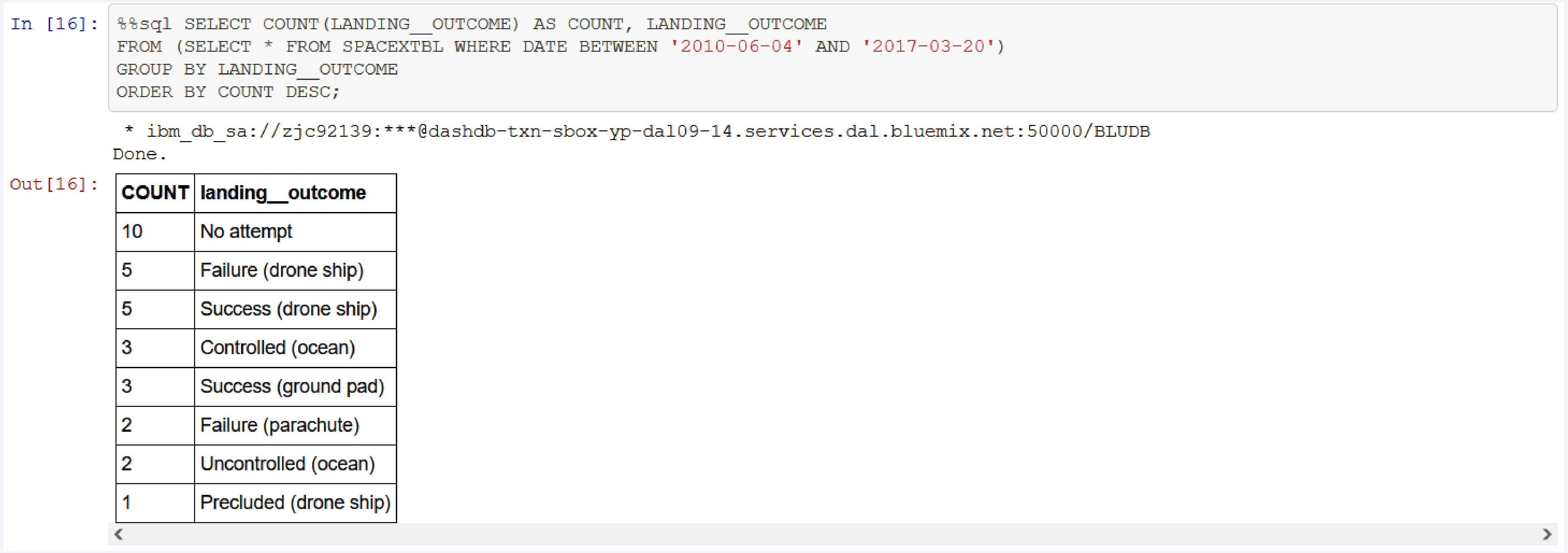


```
In [15]: %%sql SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE
         FROM SPACEXTBL
         WHERE LANDING__OUTCOME = 'Failure (drone ship)'
         AND YEAR(DATE) = 2015;

          * ibm_db_sa://zjc92139:***@dashdb-txn-sbox-yp-dal09-14.services.dal.bluemix.net:50000/BLUDB
         Done.
Out[15]:
```

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

- This query is also easy to write and is in the SELECT-WHERE-FROM format; the only hook is here is that one should combine two WHERE conditions with an AND operator and should make use of YEAR function for the second WHERE clause condition.

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



```
In [16]: %%sql SELECT COUNT(LANDING__OUTCOME) AS COUNT, LANDING__OUTCOME
         FROM (SELECT * FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20')
         GROUP BY LANDING__OUTCOME
         ORDER BY COUNT DESC;
```

```
 * ibm_db_sa://zjc92139:***@dashdb-txn-sbox-yp-dal09-14.services.dal.bluemix.net:50000/BLUDB
Done.
```

Out[16]:

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

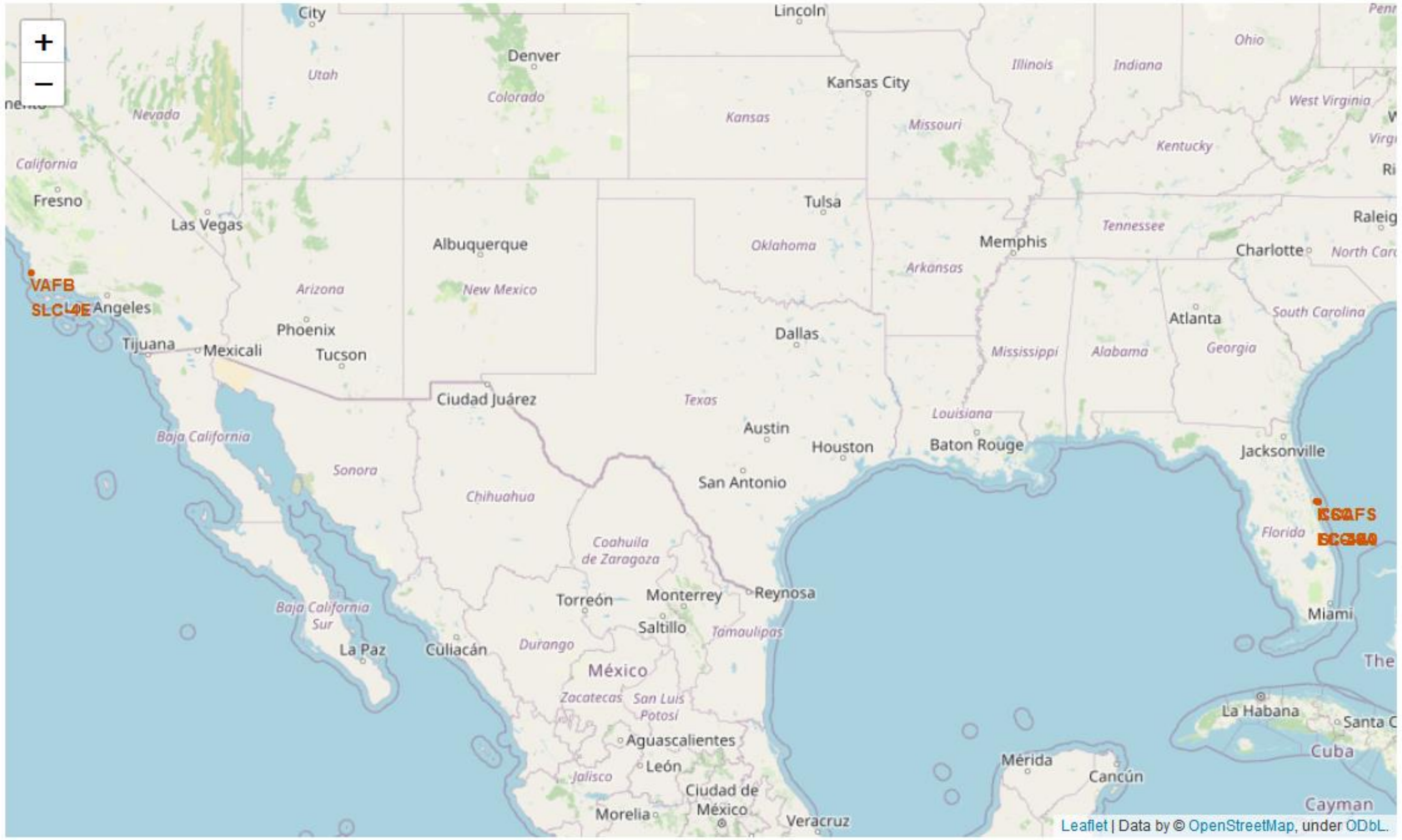
- For this query, I combined a GROUP BY and ORDER BY with a subquery for the WHERE clause to narrow down the table down to only the launches between the specified dates.

Section 4

Launch Sites Proximities Analysis

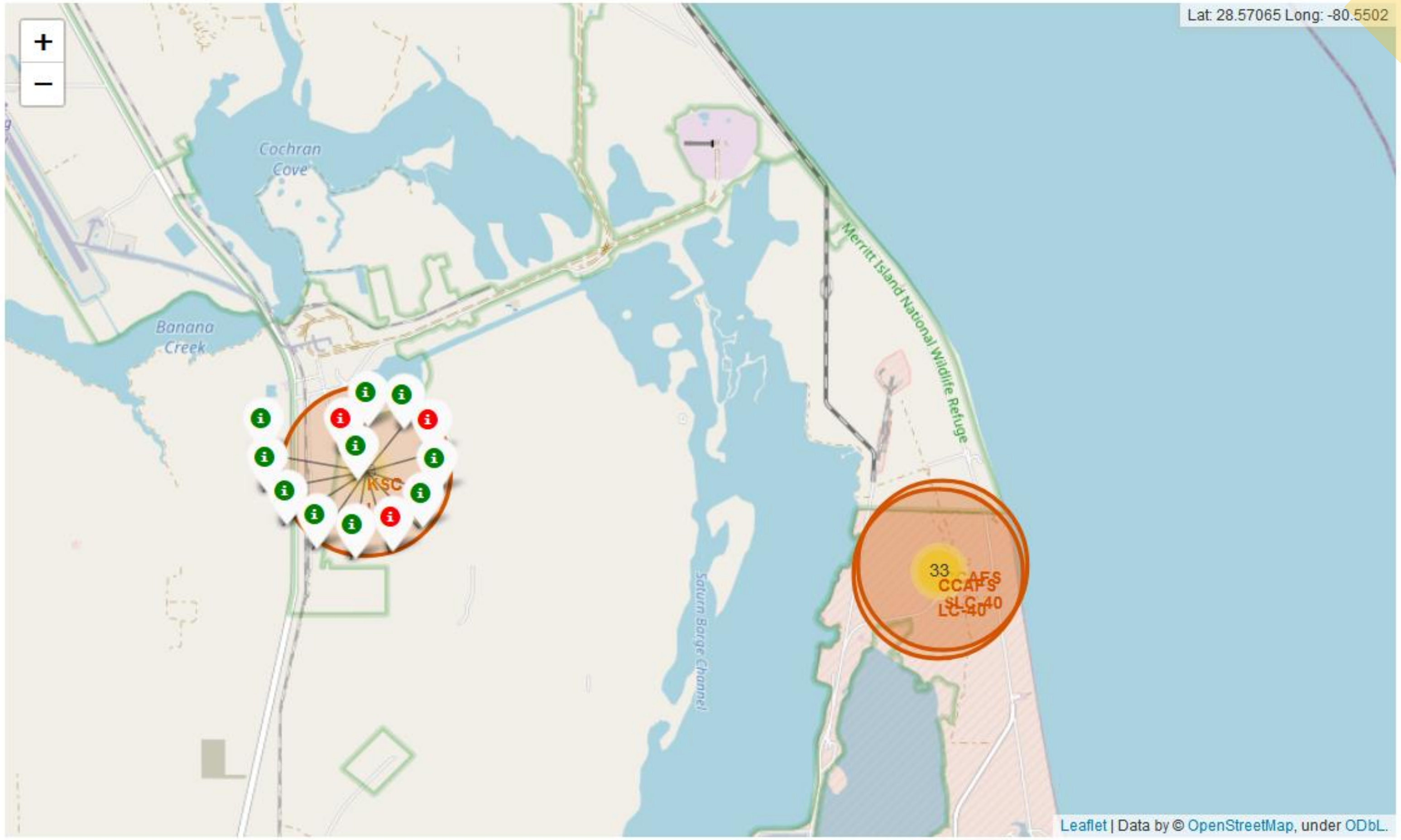
Mark All Launch Sites

- This map displays every launch site's location on worl map. To enhance the visuals, the launch sites are highlighted with a text label on a specific coordinate. There's a bit of an overlap in Florida because 3 stations (KSC LC-39A, CCAFS SLC-40, and CCAFS LC-40) are very close to each other.



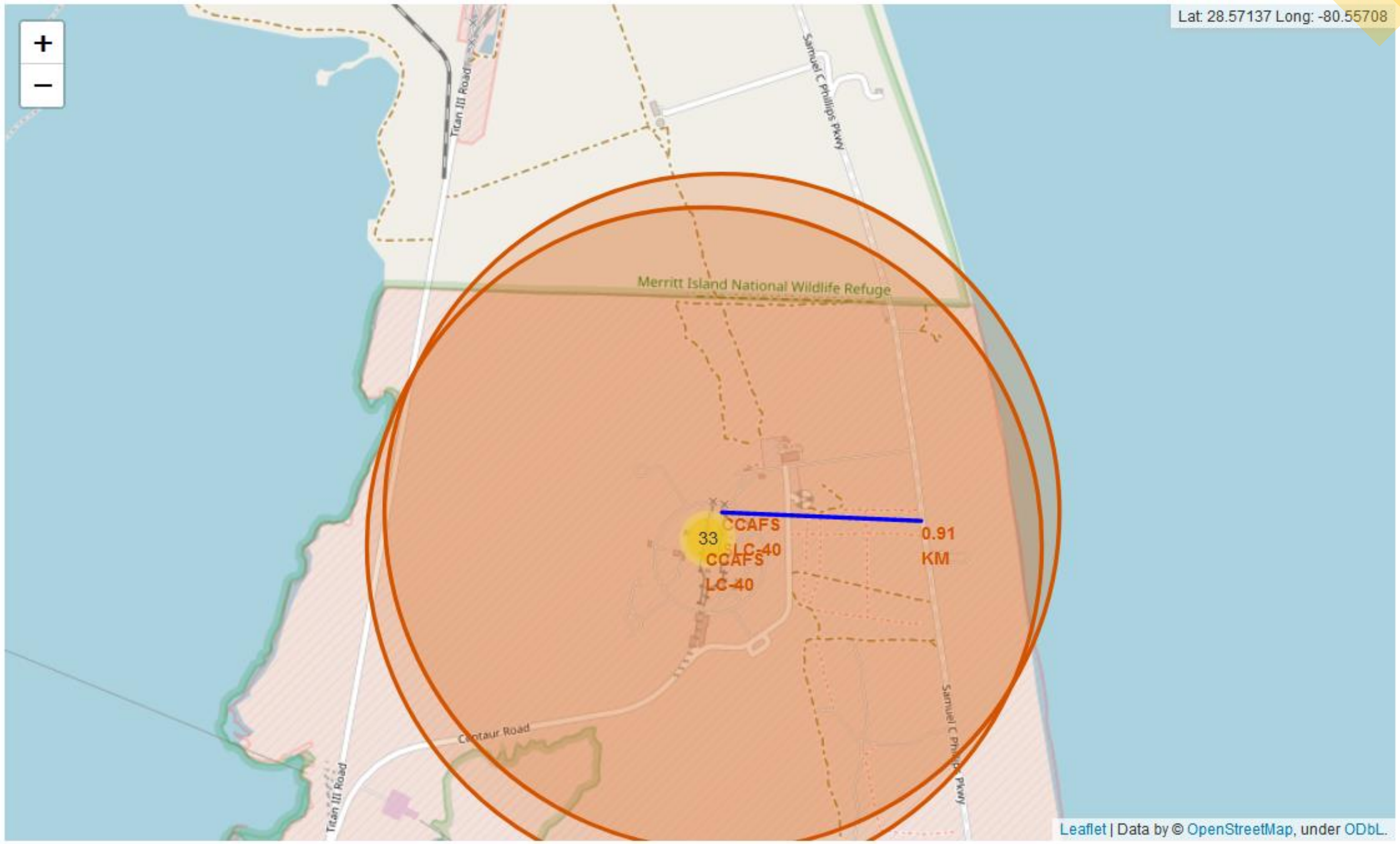
Mark the Success/Failed Launches for Each

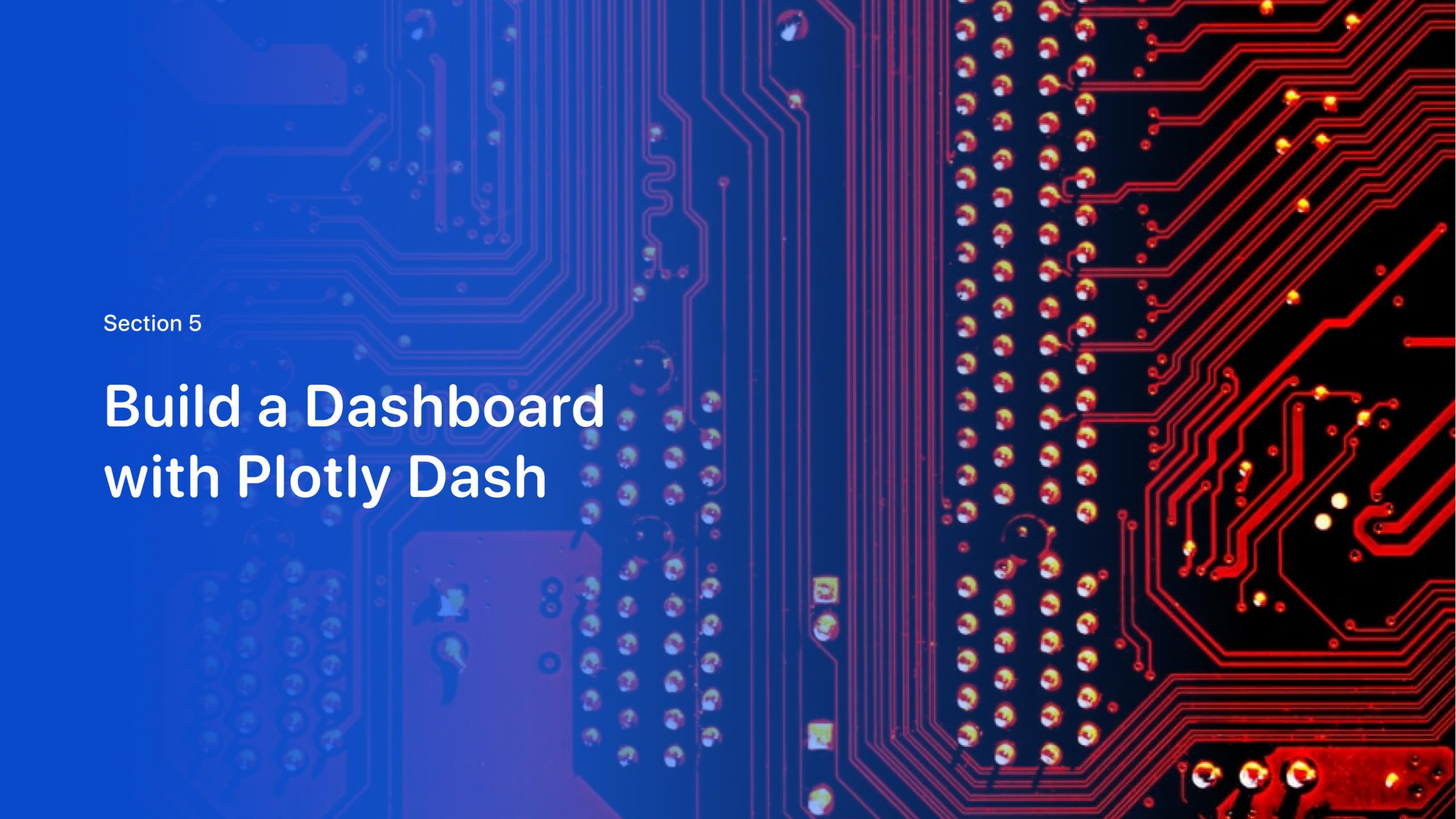
- The aim of this map is to display the sites that have high success rates. In addition to using marker clusters again, this map also contains marker colors for success/failure. Launch sites with relatively high success rate can be identified from these markers.


Lat: 28.57065 Long: -80.5502
+
−
Cochran Cove
Banana Creek
KSC
Merritt Island National Wildlife Refuge
Saturn Barge Channel
33
CCAFS
SLC-40
LC-40
Leaflet | Data by © OpenStreetMap, under ODbL.

Calculate the Distances Between a Launch Site to its Proximites

- This map displays the distance from CCAFS SLC-40 to the nearest railway road. On top the of map markers used in the second map, the enhancement here is the addition of a polyline to illustrate the distance.


+
−
Lat: 28.57137 Long: -80.55708
Merritt Island National Wildlife Refuge
Titan III Road
Samuel C Phillips Pkwy
Centaur Road
33
CCAFS LC-40
CCAFS LC-40
0.91 KM
Leaflet | Data by © OpenStreetMap, under ODbL.

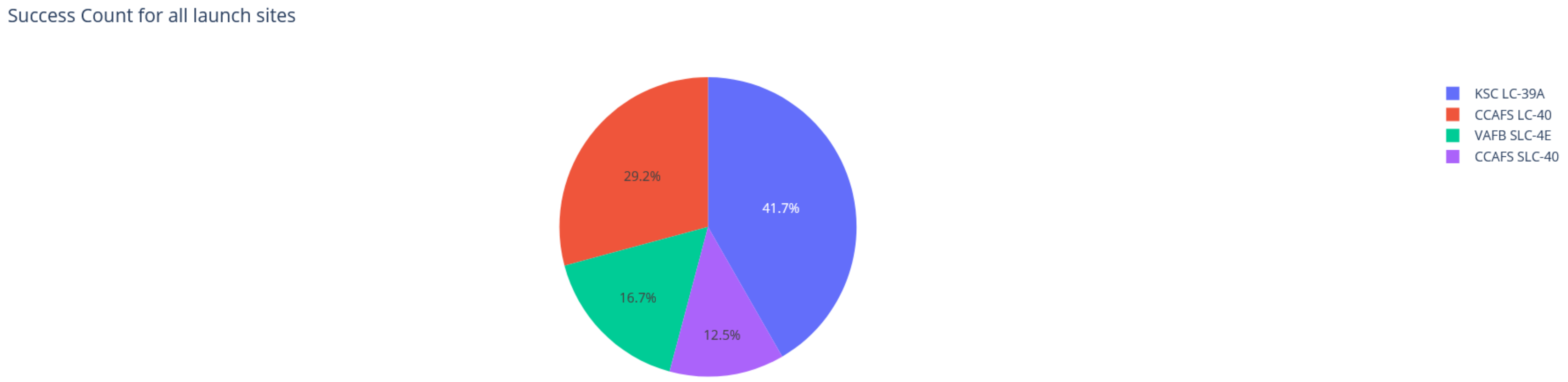


Section 5

Build a Dashboard with Plotly Dash

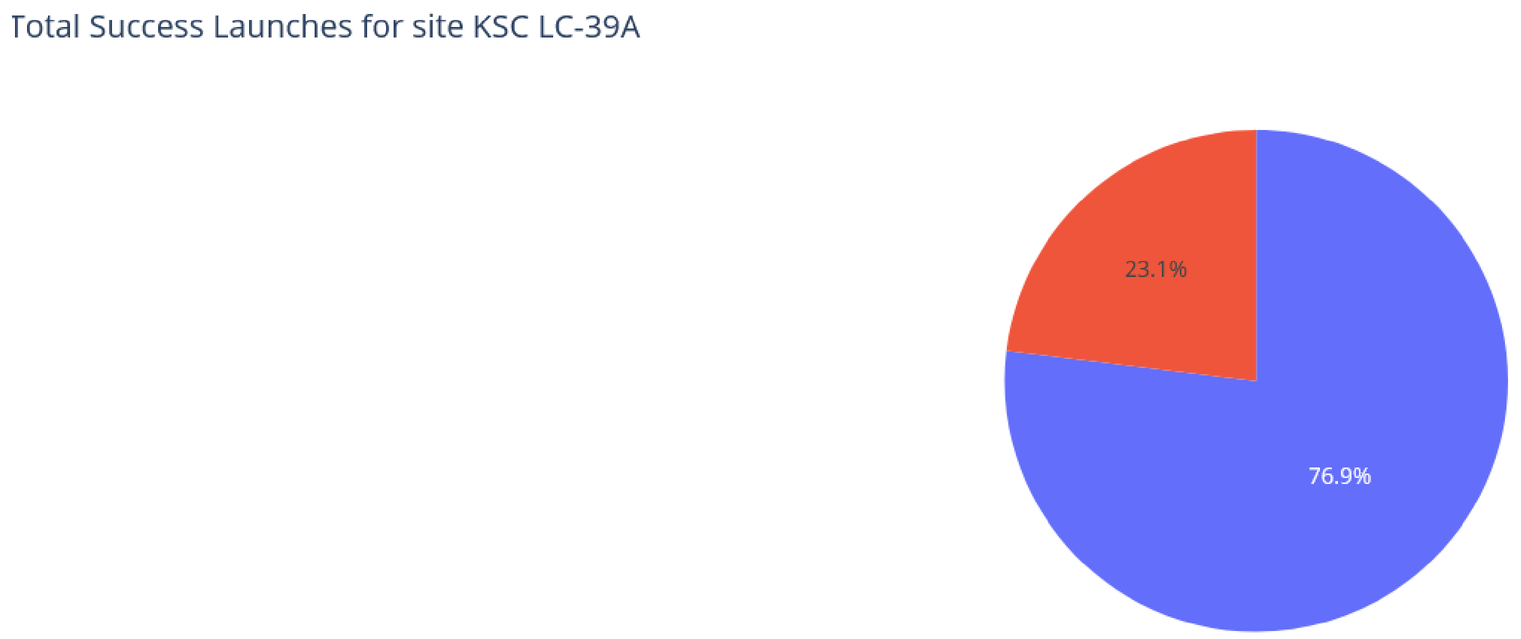
Success Count for All Launch Sites

- This pie chart shows the rate of success for all four launch sites.



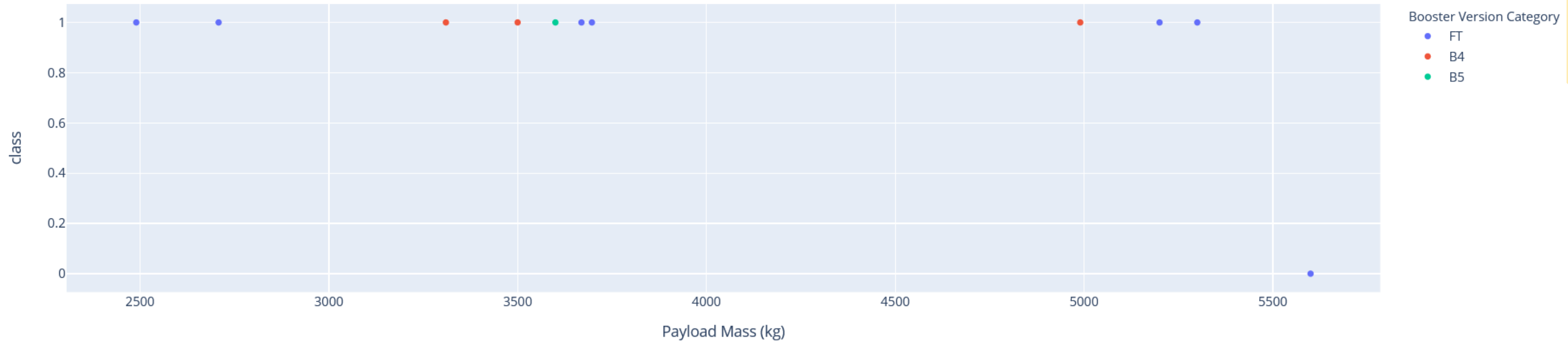
Success Count for the Most Successful Launch Site

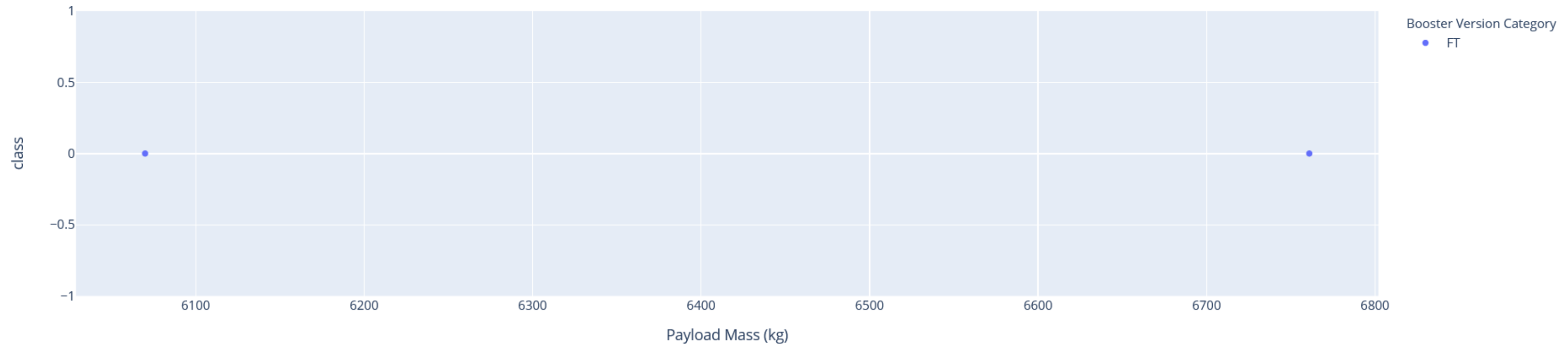
- This pie chart shows the percentage of successful launches (76.9%) for the most successful launc site (KSC LC-39A).



Payload vs. Launch Outcome

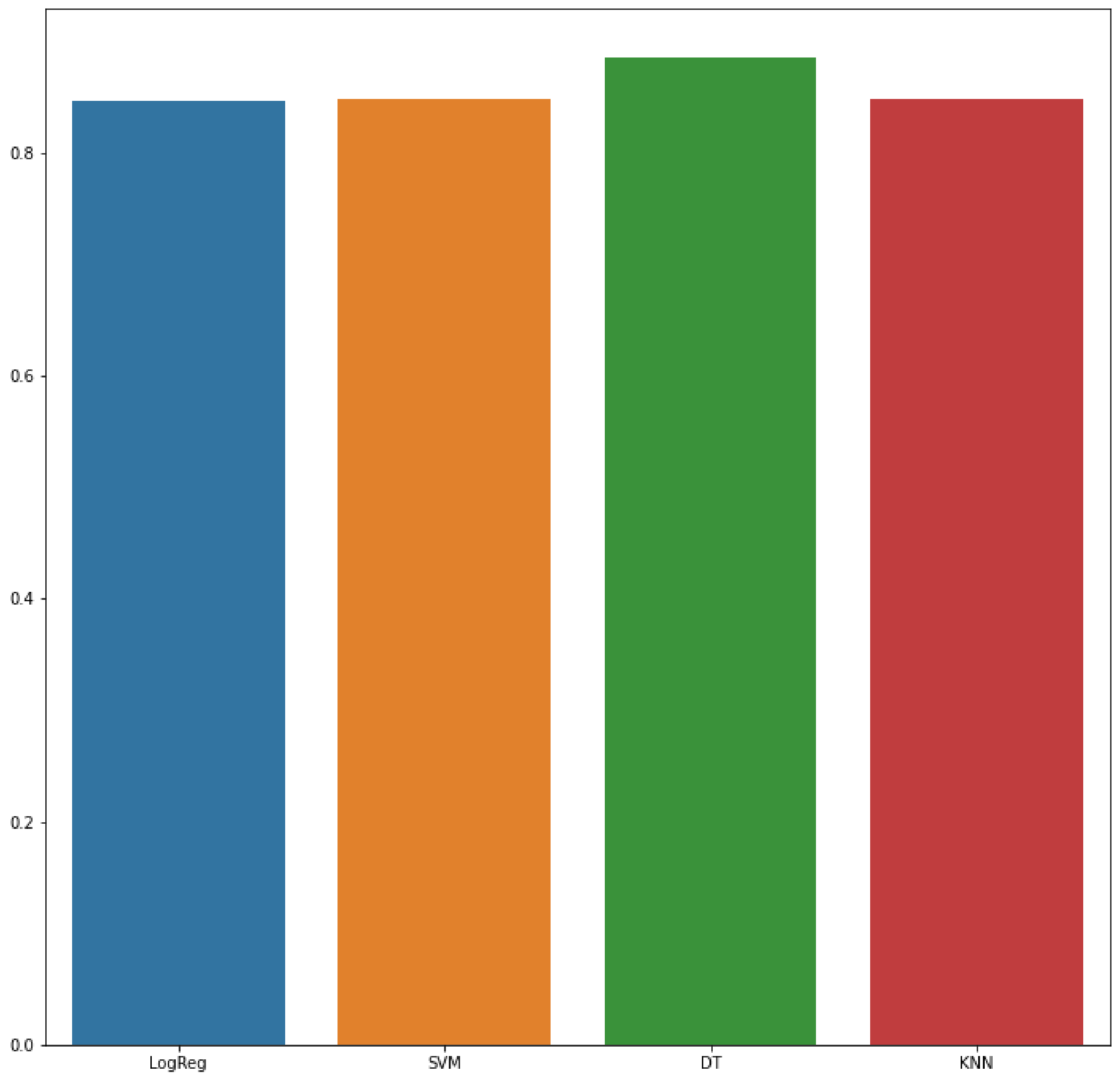
- These two scatter plots display Payload vs. Launch Outcome.
- First one is for payloads between 0-5500 kg.
- The second one is for 5500-7000kg.
- In the first range, FT is the most frequently used booster type and boasts 85% success rate. B4 and B5 both have a 100% success rate, but their sample size is rather small (three and one, respectively).
- In the second range, only FT booster was used, and it has no successful attempts. We could interpret this as success probability is non-existent for FT beyond 5500 kg, but sample size is only n=2.





Section 6

Predictive Analysis (Classification)

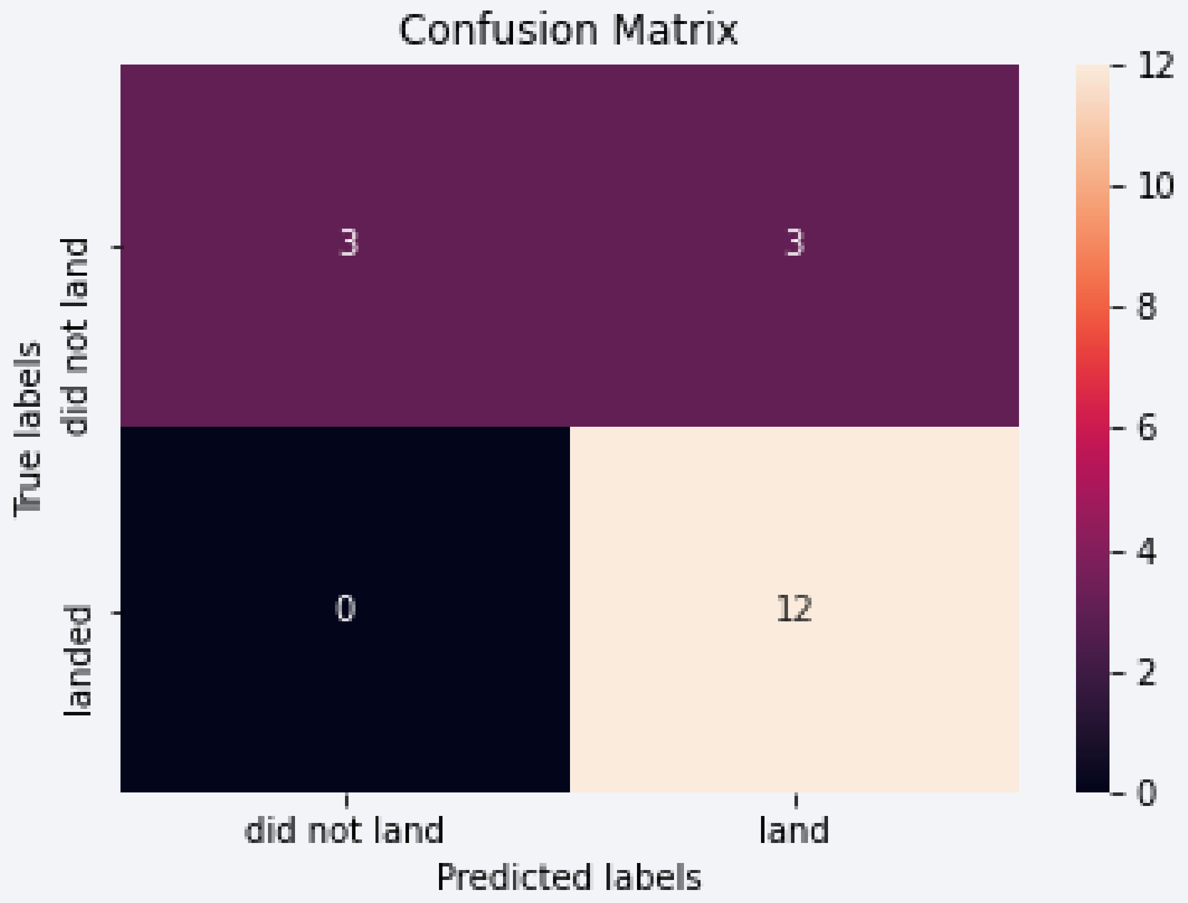


Classification Accuracy

- Almost all models perform the same, however a Decision Tree model pulls forward slightly (88.6%).

Confusion Matrix

- The DT model correctly identifies all positive labels; we have no false negatives. All 12 successful attempts have been labeled as true positive.
- On the historic data, the performance falls drastically in the negative label; with 3 true negatives, we have 3 false positives.
- At the cost of 100% sensitivity, we have only 50% specificity.
- Our model knows very well how to predict a launch that will be successful (0% Type II Error chance), but half the time it will classify a launch that will be a failure as successful (50% Type I Error chance).
- The model may need further tweaking to prevent Space Y from bidding on these failures the model falsely predicts to be successful.



Conclusions

- The best performing model among the four classifications algorithm that were utilized, was the decision tree model.
- The DT model outperforms other models by about 5%, at a performance of 88% accuracy.
- However, accuracy alone doesn’t count for the intricacies of the decision-making process. The false positive rate is 0, whereas false negative rate is 0.5.
- For the historic data, the model was able to predict all successful launches depending on the inputs and missed half of the launches that failed and classified them as successes.
- Depending on the needs of the company, this model could be improved through incorporating ensemble learning methods (like Random Forest instead of just one tree).

Appendix

- All the Python and SQL code used in the making of this slide could be found under the following GitHub repo link:
- https://github.com/uft93/Coursera_Capstone_Space_Y

Thank you!