IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ruchira T. N. Rajapakshe

27.12.2023

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Discussion

Executive Summary

- Summary of methodologies
  - Data collection using API
  - Data collection with web scraping
  - Data wrangling
  - Exploratory Data Analysis (EDA) using SQL
  - EDA using Pandas and Matplotlib
  - Interactive visual analysis and dashboard
  - Predictive analysis
- Summary of all results
  - EDA results
  - Interactive analysis results
  - Predictive analysis results

Introduction

- Project background and context

On its website, Space X promotes Falcon 9 rocket launches at a price of 62 million dollars; in comparison, other suppliers charge up to 165 million dollars per launch; a large portion of the cost savings are attributable to Space X's ability to reuse the first stage. Thus, we can calculate the cost of a launch if we can ascertain if the first stage will land. This data may be utilized should another business wish to submit a bid for a rocket launch against Space X. The project's objective is to build a machine learning pipeline that can forecast whether or not the initial stage will land successfully.

- Problems you want to find answers

1. What elements influence the rocket's likelihood of a successful landing?
2. The way different features interact to determine the likelihood of a successful landing.
3. What operational circumstances must exist in order to guarantee the success of the landing program.

Section 1

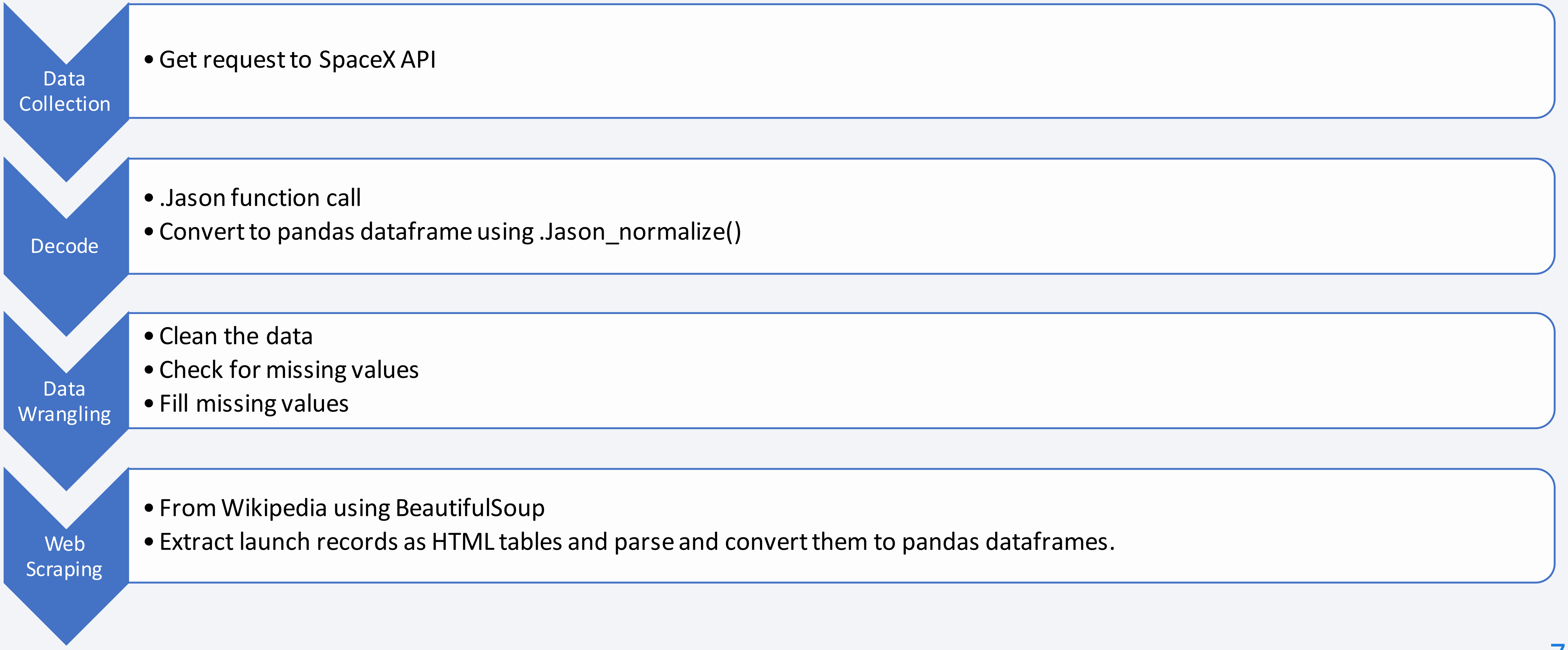
Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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



Data Collection

- Get request to SpaceX API

Decode

- .Jason function call
- Convert to pandas dataframe using .Jason_normalize()

Data Wrangling

- Clean the data
- Check for missing values
- Fill missing values

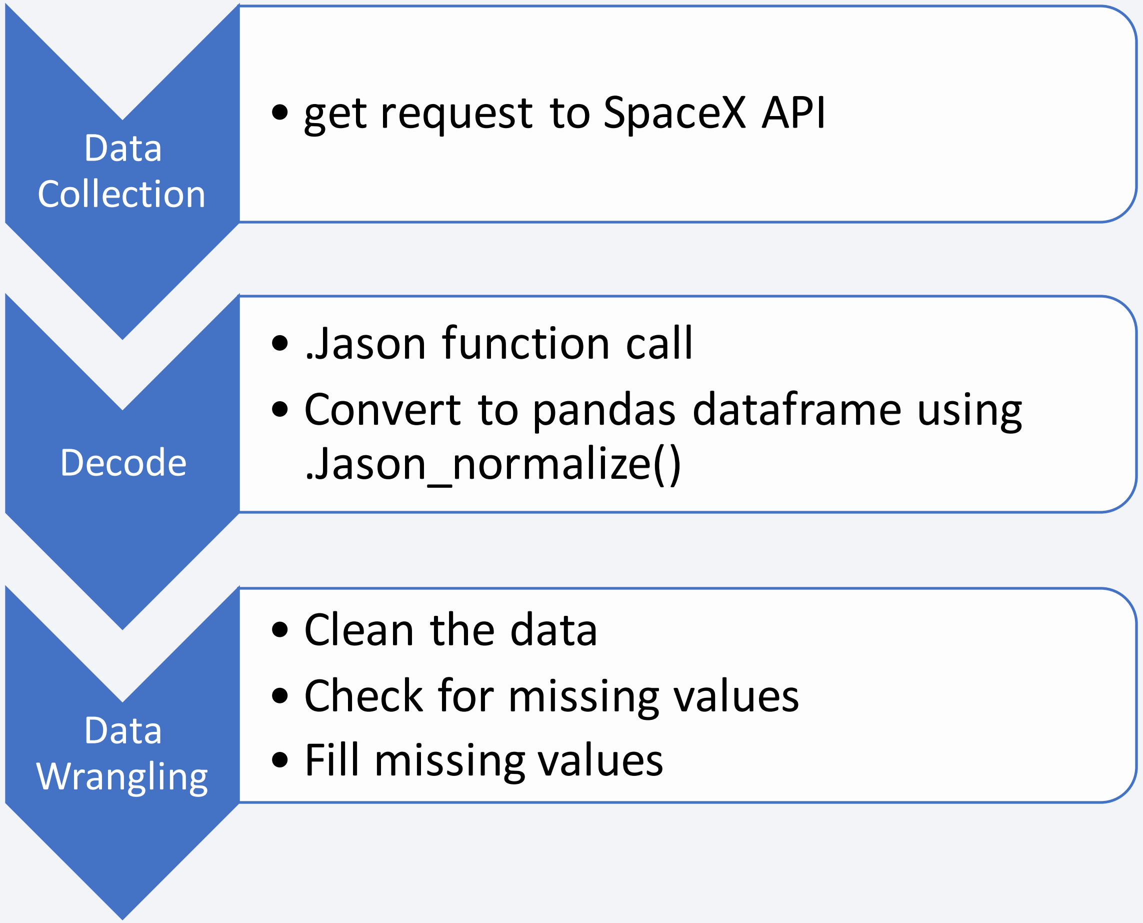
Web Scraping

- From Wikipedia using BeautifulSoup
- Extract launch records as HTML tables and parse and convert them to pandas dataframes.

Data Collection – SpaceX API

- To gather data, sanitize the requested data, and do some simple data wrangling and formatting, we used the get request to the SpaceX API.
- GitHub URL

[DS_Capstone_Ruchira/jupyter-labs-spacex-data-collection-api.ipynb at main · ruchiratn/DS_Capstone_Ruchira (github.com)]



Data Collection

- get request to SpaceX API

Decode

- .Jason function call
- Convert to pandas dataframe using .Jason_normalize()

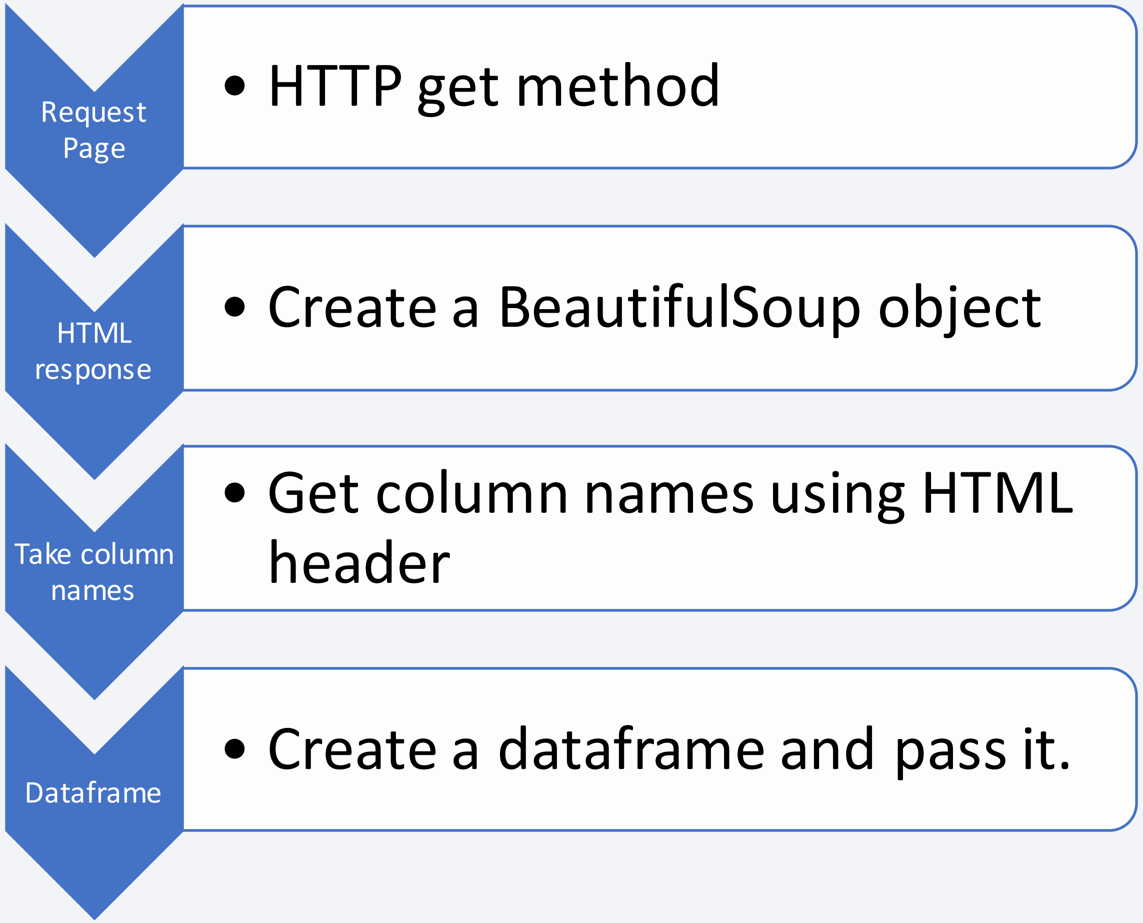
Data Wrangling

- Clean the data
- Check for missing values
- Fill missing values

Data Collection - Scraping

- Using BeautifulSoup, we used web scraping to gather Falcon 9 launch records.The table was parsed, and a pandas dataframe was created.
- GitHub URL

[DS_Capstone_Ruchira/jupyter-labs-webscraping.ipynb at main · ruchiratn/DS_Capstone_Ruchira (github.com)]



Request Page

- HTTP get method

HTML response

- Create a BeautifulSoup object

Take column names

- Get column names using HTML header

Dataframe

- Create a dataframe and pass it.

Data Wrangling

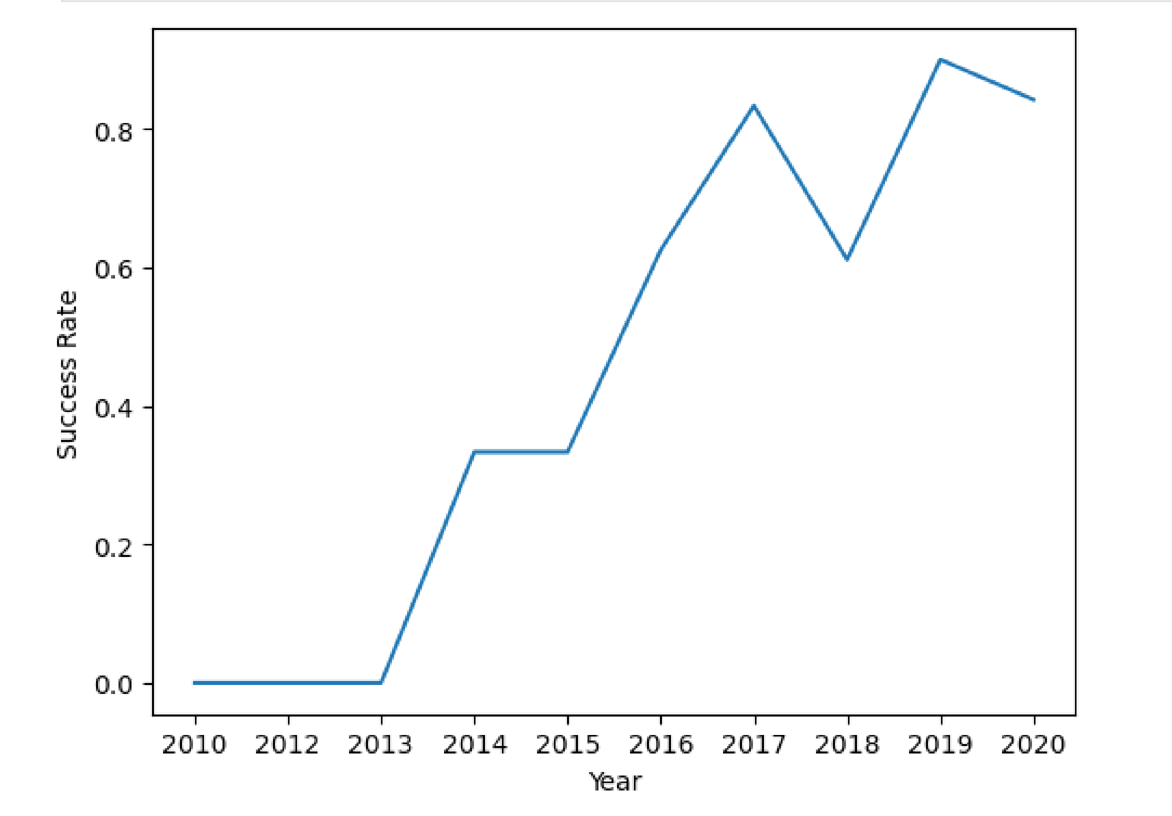
- We identified the training labels by doing an exploratory data analysis.
- We determined the quantity of launches at every location as well as the quantity and frequency of each orbit.
- From the outcome column, we generated the landing outcome label and exported the data to CSV.
- GitHub URL

DS_Capstone_Ruchira/labs-jupyter-spacex-Data wrangling.ipynb at main · ruchiratn/DS_Capstone_Ruchira (github.com)

EDA with Data Visualization

- The relationship between the flight number and the launch site, the payload and the launch site, the success rate of each orbit type, the flight number and the orbit type, and the annual trend of launch success were all visualized as we investigated the data.
- GitHub URL

https://github.com/ruchiratn/DS_Capstone_Ruchira/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb



EDA with SQL

- Without leaving the Jupyter notebook, we loaded the SpaceX dataset into a SQL database.
- We used SQL and EDA to extract knowledge from the data.
- For example, we created queries to determine:
  - The names of the space mission's distinct launch sites.
  - The total mass of payload carried by NASA's launched boosters (CRS)
  - The mass of the payload that booster type F9 v1.1 typically carries
  - The total number of mission outcomes that were successful or unsuccessful
  - The drone ship's unsuccessful landing results, along with the names of the launch sites and booster version.
- GitHub URL

https://github.com/ruchiratn/DS_Capstone_Ruchira/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- On the folium map, we annotated every launch site and added map elements like circles, markers, and lines to indicate the success or failure of launches at every location.
- Class 0 and 1 were given the feature launch outcomes (success or failure).that is, one for achievement and zero for failure.
- We were able to determine which launch sites have a comparatively high success rate by using the color-labeled marker clusters.
- We determined the separations between a launch location and its surrounding areas. We responded to a few inquiries, such as:
  - Are launch locations close to highways, railroads, and coasts.
  - Do launch locations maintain a specific separation from urban areas?
- GitHub URL

DS_Capstone_Ruchira/lab_jupyter_launch_site_location.jupyterlite.ipynb at main · ruchiratn/DS_Capstone_Ruchira (github.com)

Build a Dashboard with Plotly Dash

- We used Plotly dash to create an interactive dashboard.
- We created pie charts that displayed each site's total launches.
- For each booster version, we created a scatter graph that displayed the link between the outcome and payload mass (kg).
- GitHub URL

DS_Capstone_Ruchira/spacex_dash_app.py at main · ruchiratn/DS_Capstone_Ruchira (github.com)

Predictive Analysis (Classification)

- Using pandas and numpy, we loaded the data, processed it, and divided it into training and testing sets.
- Using GridSearchCV, we constructed several machine learning models and adjusted various hyperparameters.
- Our model's accuracy served as its metric, and it was enhanced through feature engineering and algorithm tuning.
- The top-performing categorization model has been identified.
- GitHub URL

DS_Capstone_Ruchira/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb at main · ruchiratn/DS_Capstone_Ruchira (github.com)

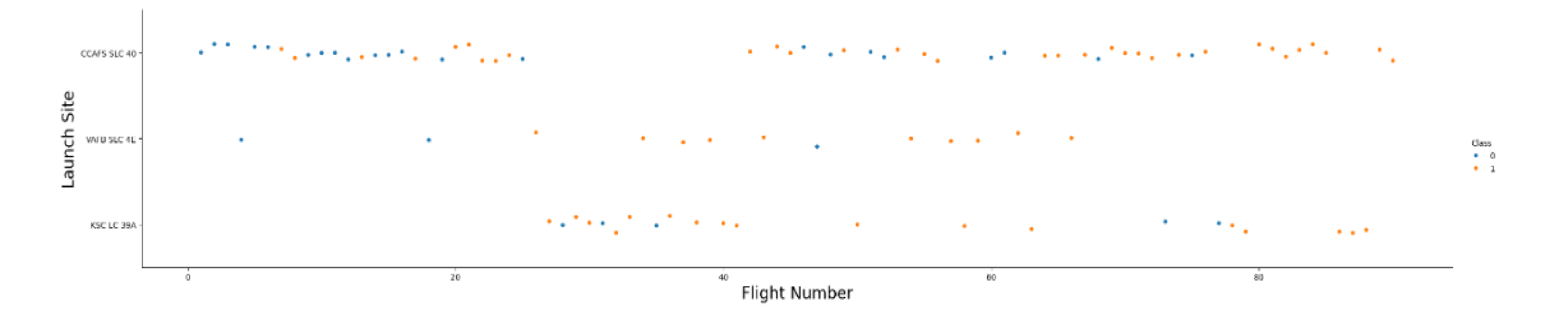
Results

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

Section 2

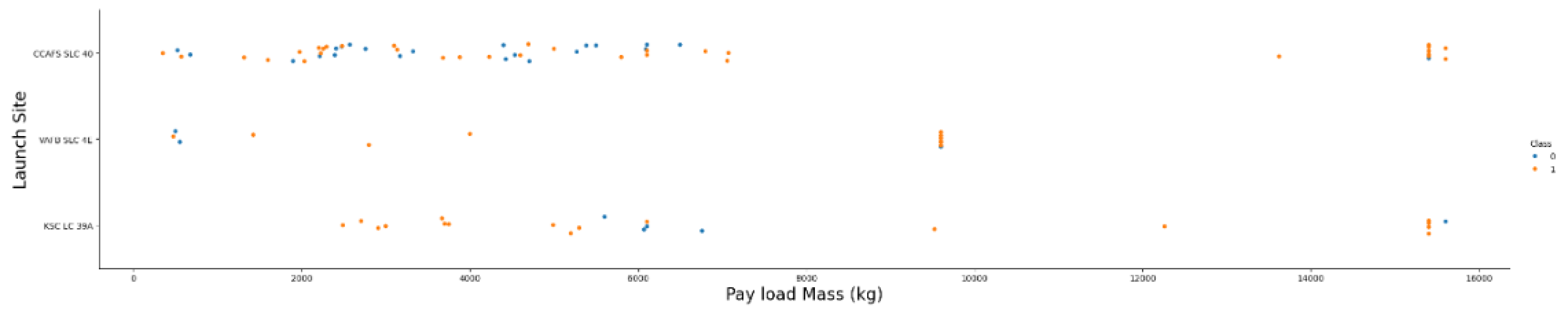
Insights drawn from EDA

Flight Number vs. Launch Site



We deduced from the plot that a launch site's success rate increases with the number of flights conducted there.

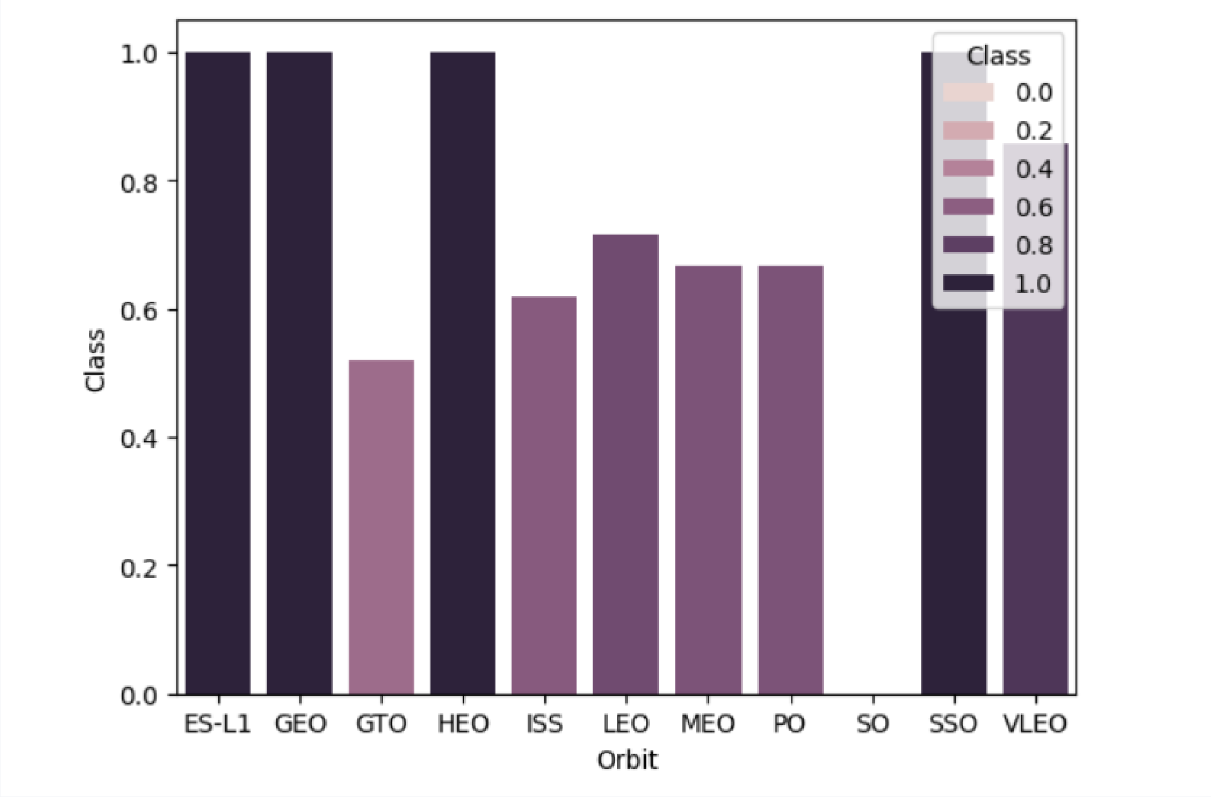
Payload vs. Launch Site



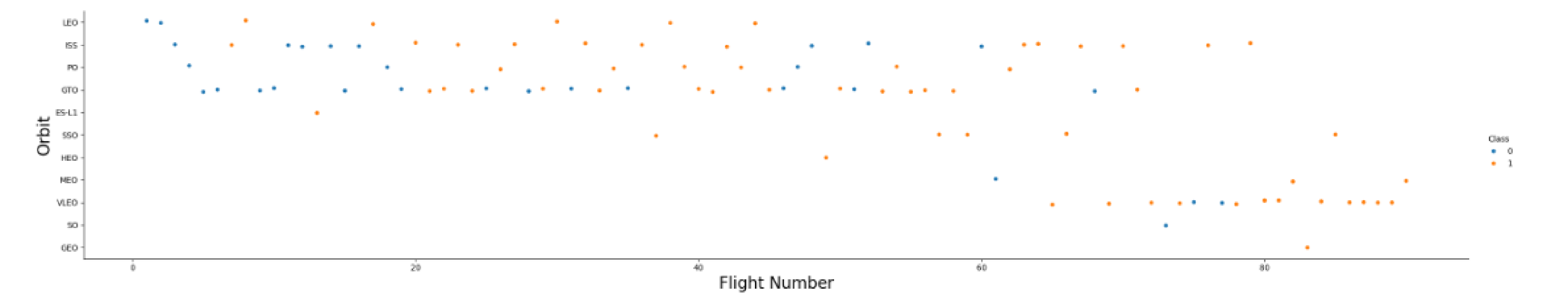
For the VAFB-SLC launchsite there are no rockets launched for heavy payload mass(greater than 10000).

Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- GTO has the lowest success rate.

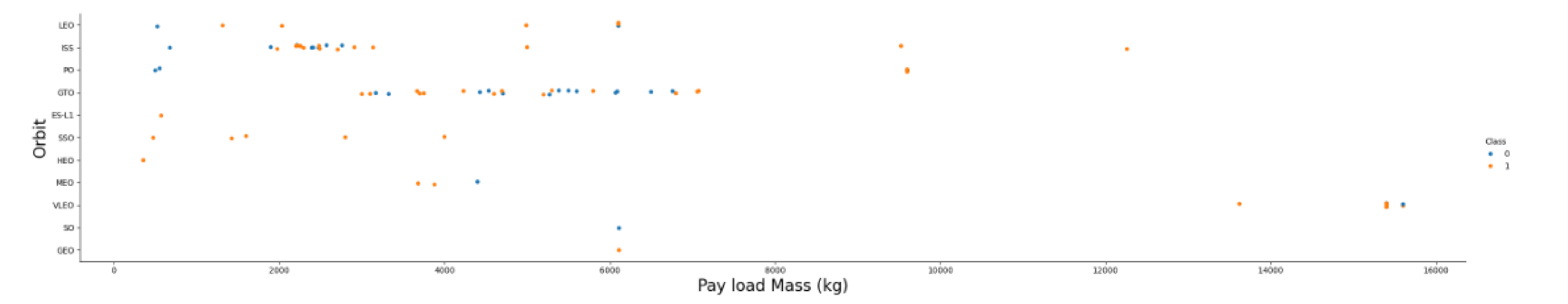


Flight Number vs. Orbit Type



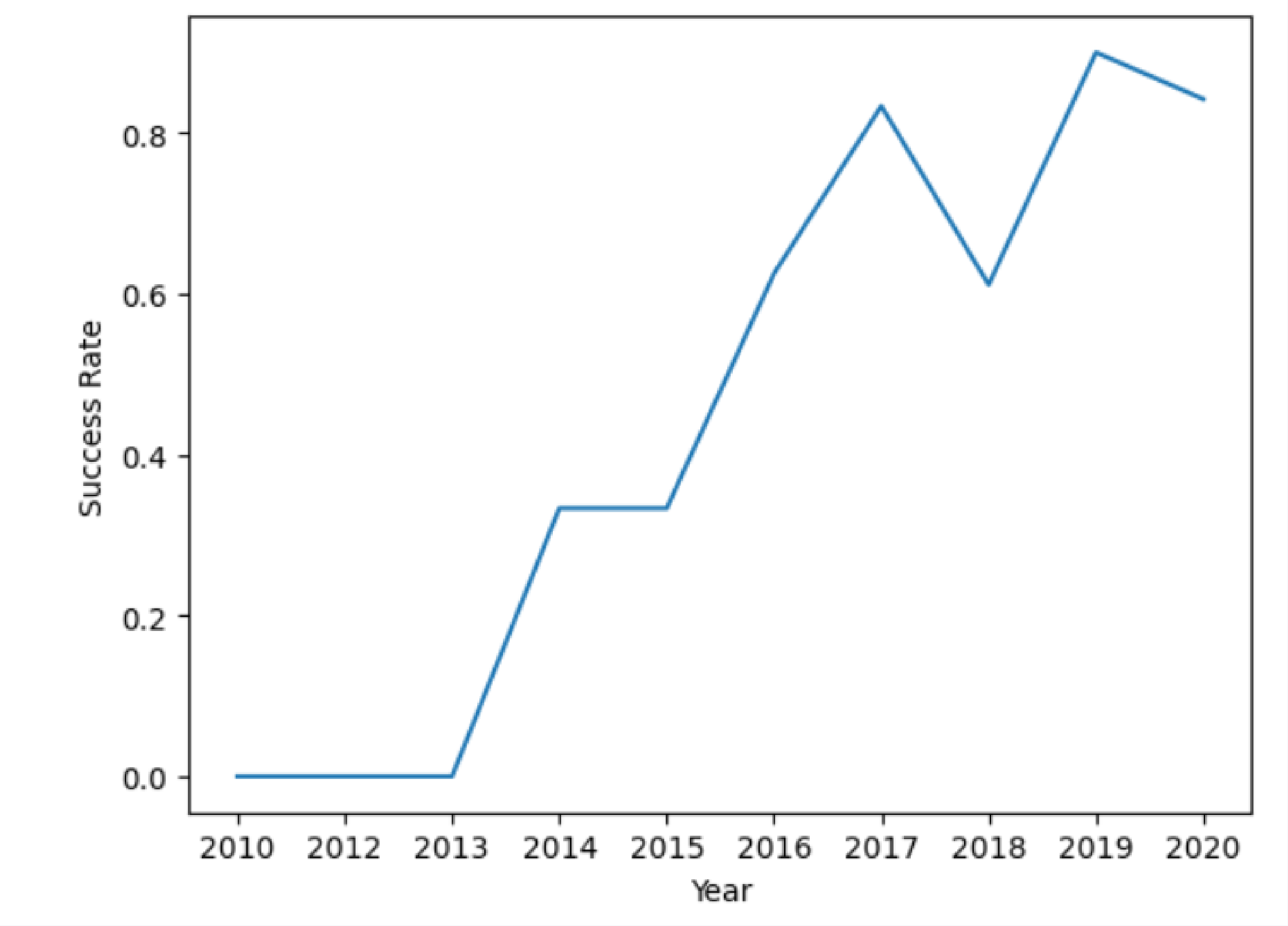
- In the LEO orbit the Success appears related to the number of flights.
- On the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS orbits.
- However, for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

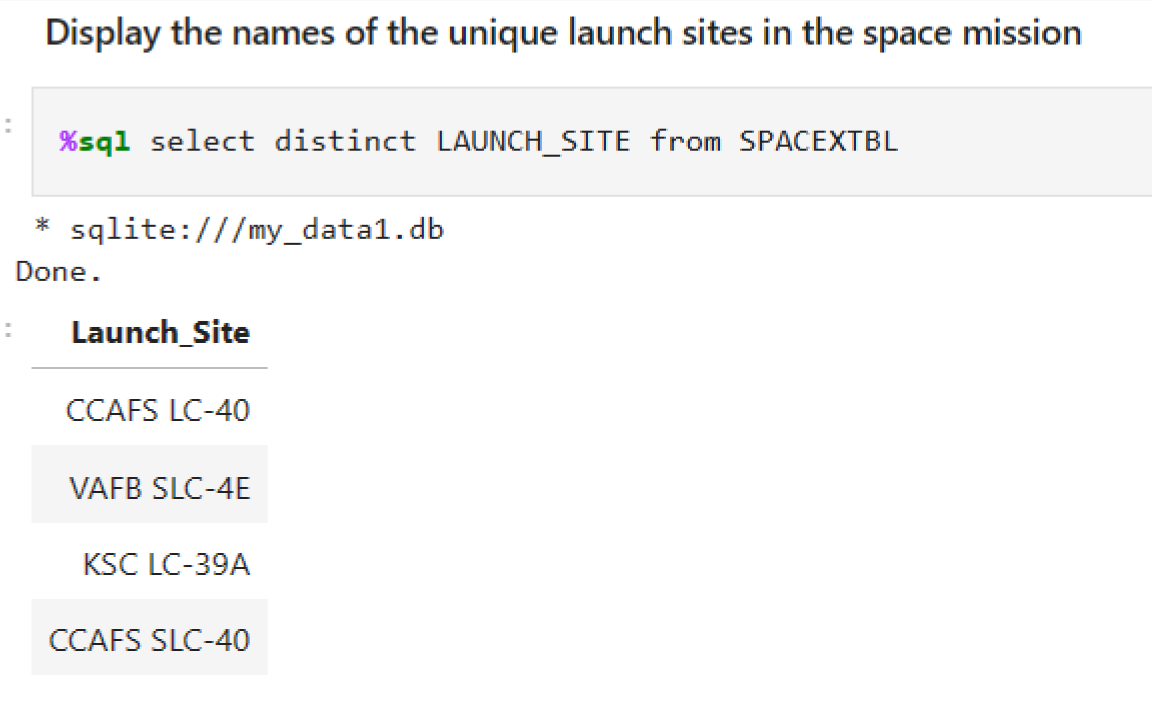
Launch Success Yearly Trend



- The success rate since 2013 kept increasing till 2020

All Launch Site Names

- We selected only unique launch sites from the SpaceX data by using the keyword DISTINCT.



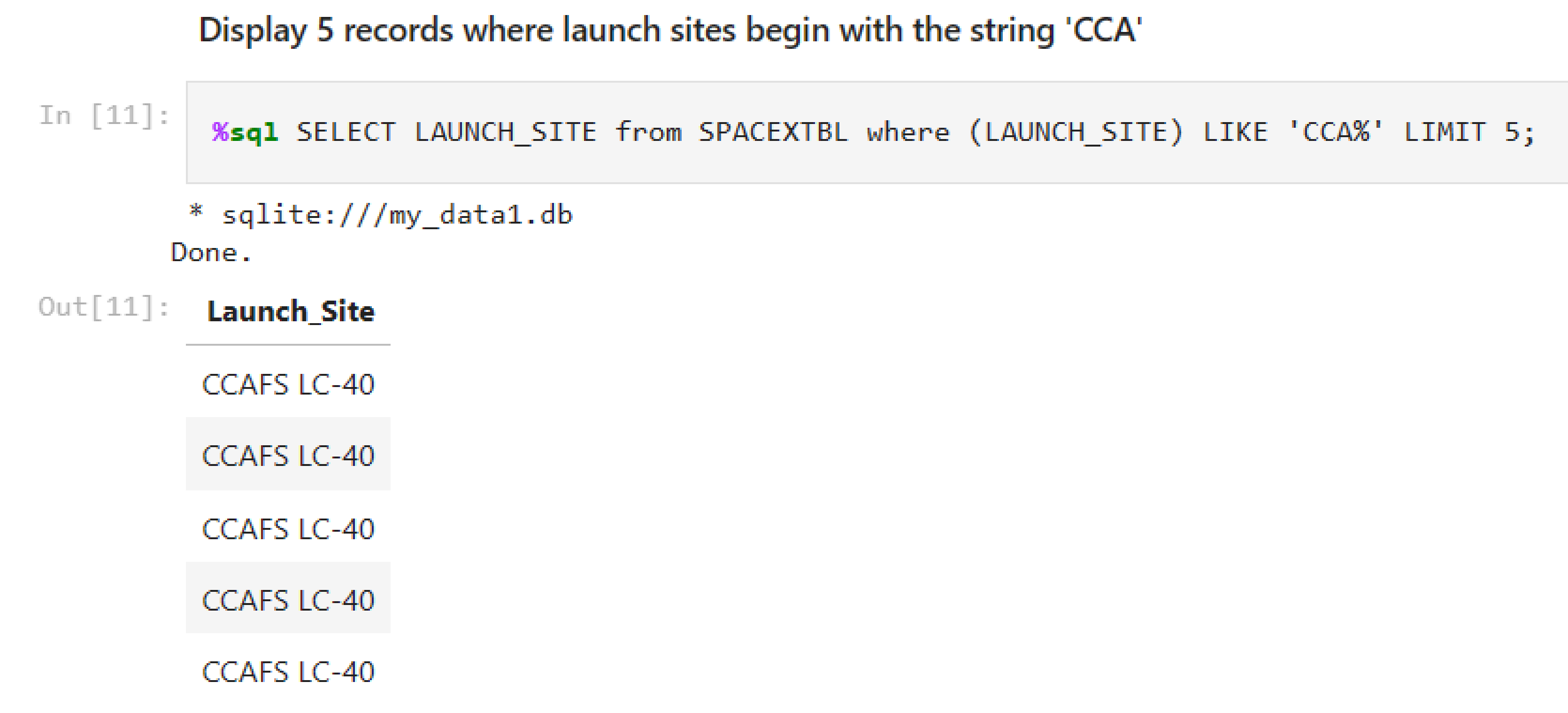
Display the names of the unique launch sites in the space mission

```
%sql select distinct LAUNCH_SITE from SPACEXTBL
```

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

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

Launch Site Names Begin with 'CCA'



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

```
In [11]: %sql SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

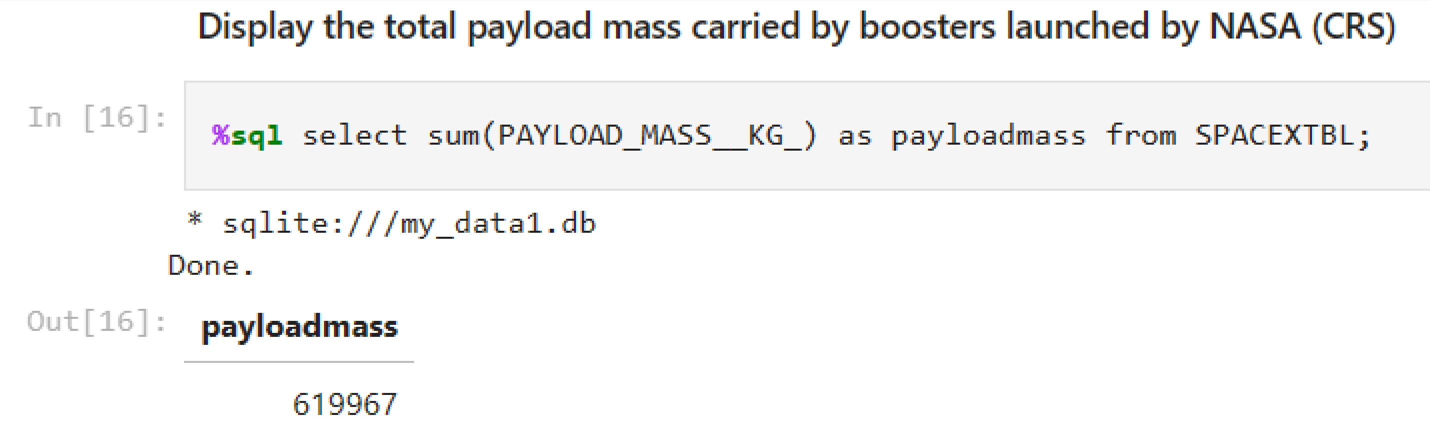
```
 * sqlite:///my_data1.db
Done.
```

Out[11]:

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

- We used select method to display 5 records where launch sites begin with `CCA`

Total Payload Mass

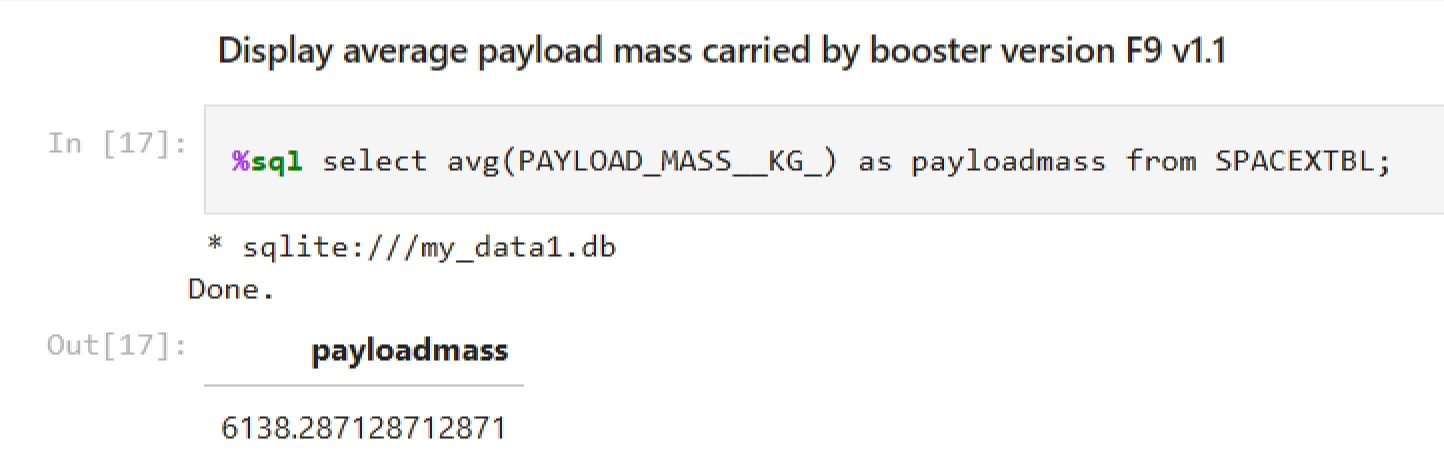


Display the total payload mass carried by boosters launched by NASA (CRS)

```
In [16]:  %sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;

           * sqlite:///my_data1.db
          Done.

Out[16]:  payloadmass

              619967
```

- The total payload carried by boosters from NASA was calculated using the query above.

Average Payload Mass by F9 v1.1

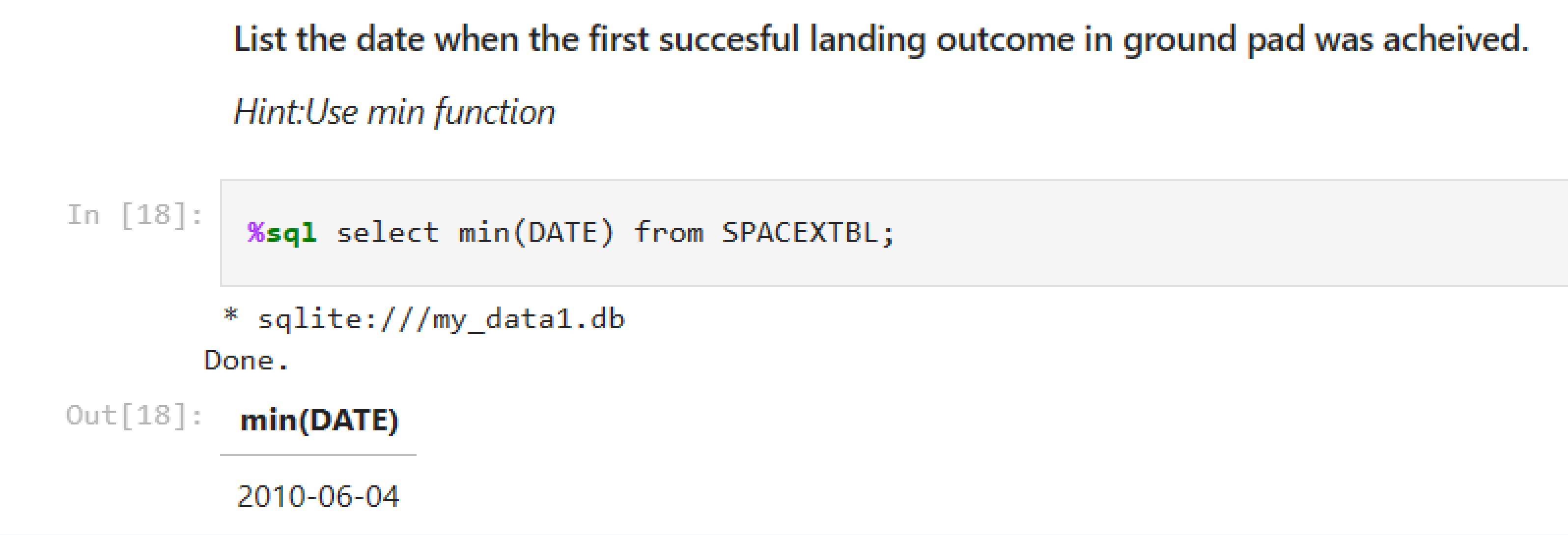


Display average payload mass carried by booster version F9 v1.1

```
In [17]:  %sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;

           * sqlite:///my_data1.db
          Done.

Out[17]:        payloadmass
           ------------------
            6138.287128712871
```

- The average payload mass carried by booster version F9 v1.1 was calculated as above.

First Successful Ground Landing Date



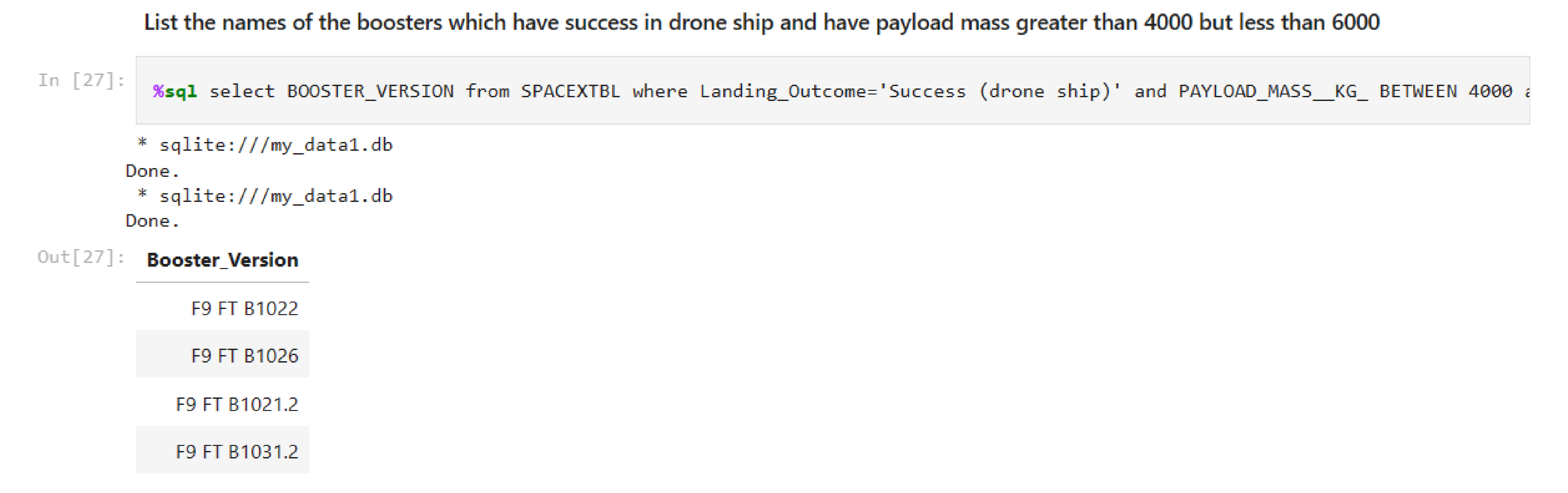
List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
In [18]:  %sql select min(DATE) from SPACEXTBL;

 * sqlite:///my_data1.db
Done.

Out[18]:  min(DATE)

          2010-06-04
```

- The first successful landing outcome on ground pad was 2010-06-04.

Successful Drone Ship Landing with Payload between 4000 and 6000



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

```
In [27]: %sql select BOOSTER_VERSION from SPACEXTBL where Landing_Outcome='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 a
```

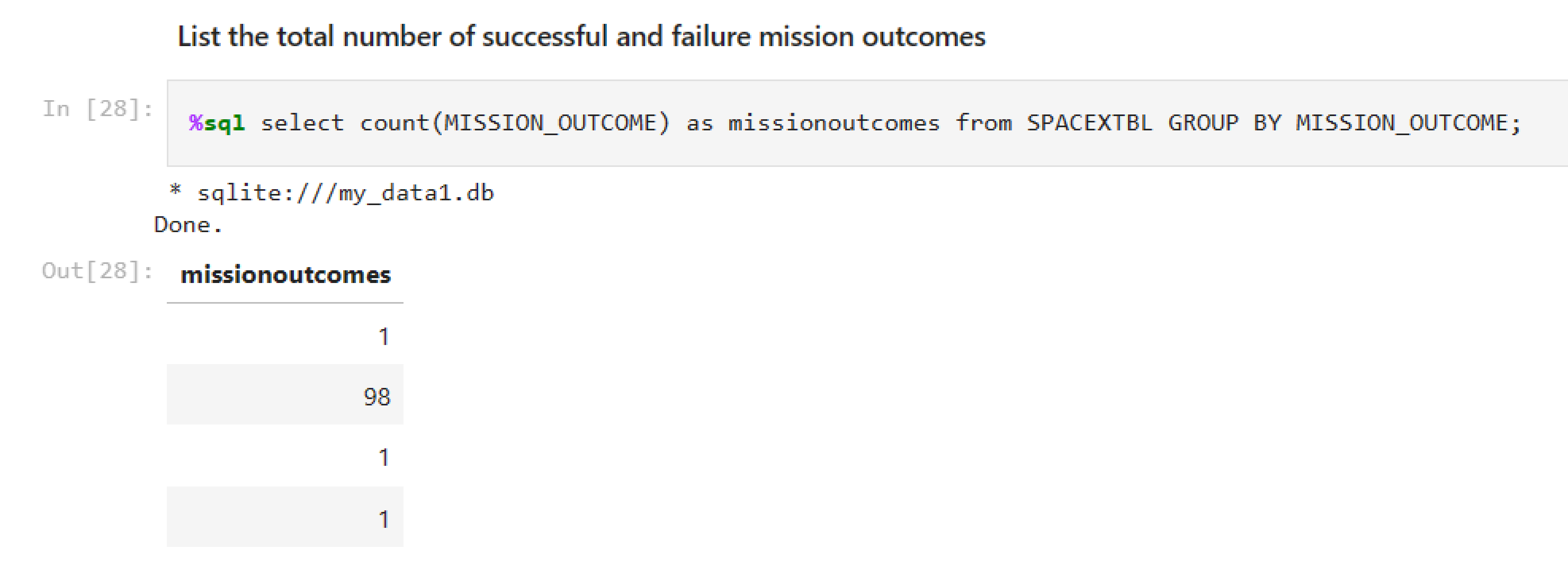
```
 * sqlite:///my_data1.db
Done.
 * sqlite:///my_data1.db
Done.
```

Out[27]:

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

In order to identify successful landings with payload masses larger than 4000 but less than 6000, we employed the AND condition after using the WHERE clause to filter for boosters that have successfully landed on drone ships.

Total Number of Successful and Failure Mission Outcomes



List the total number of successful and failure mission outcomes

```
In [28]:  %sql select count(MISSION_OUTCOME) as missionoutcomes from SPACEXTBL GROUP BY MISSION_OUTCOME;
```

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

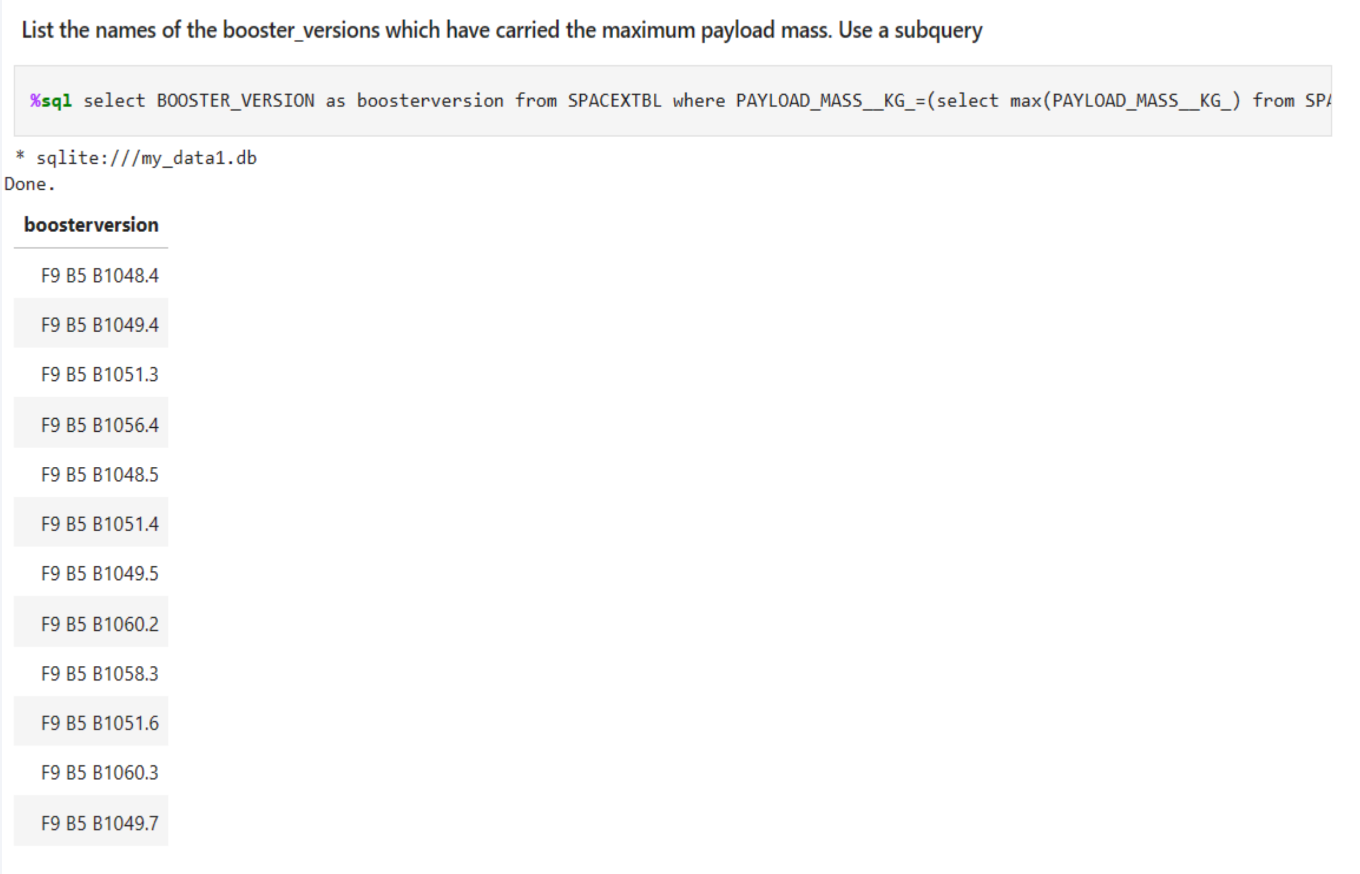
Out[28]:

missionoutcomes
1
98
1
1

- We used wildcard like ‘%’ to filter for **WHERE** MissionOutcomes was a success or a failure.

Boosters Carried Maximum Payload

- The booster that have carried
  the maximum payload was determined using a subquery in
  the **WHERE** clause and
  the **MAX()** function.



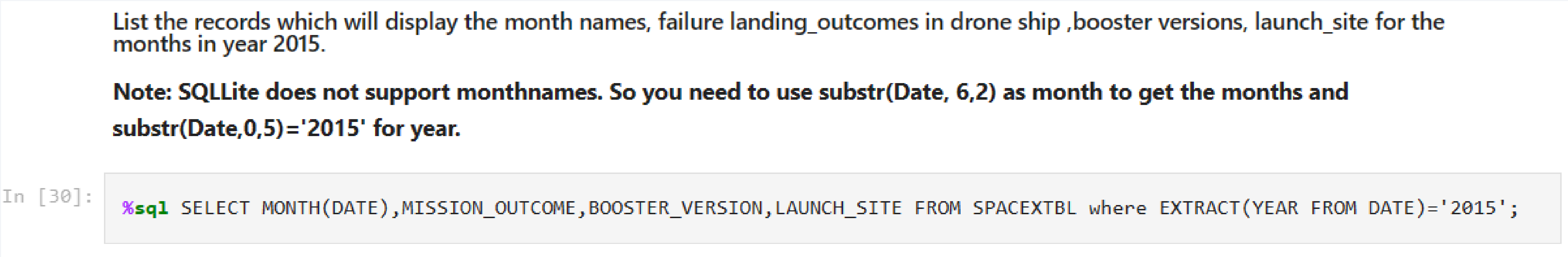
List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPA
```

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

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

2015 Launch Records



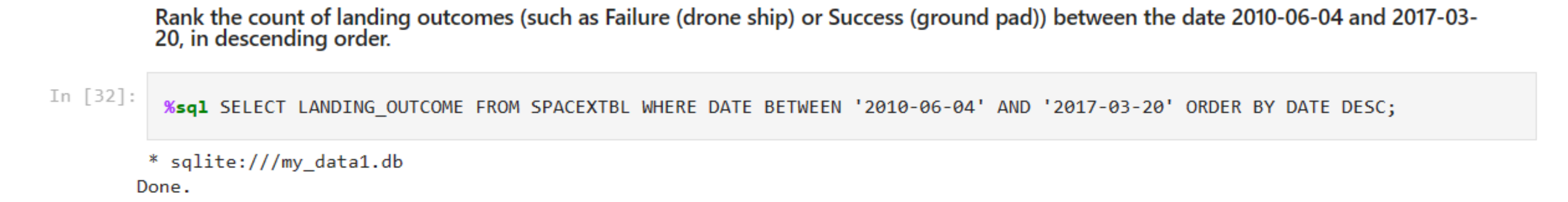
List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

In [30]:
```
%sql SELECT MONTH(DATE),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL where EXTRACT(YEAR FROM DATE)='2015';
```

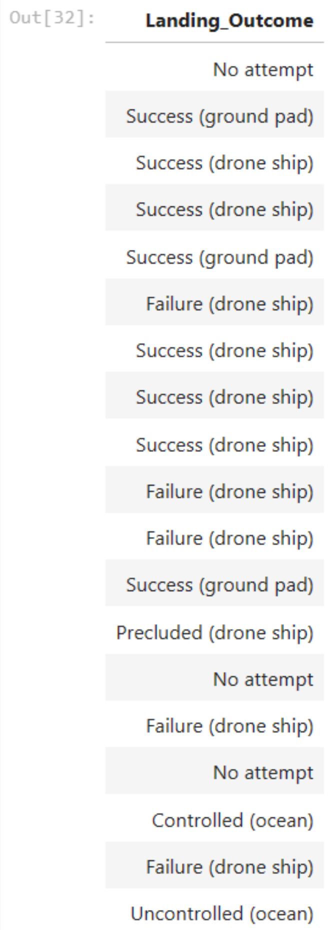
- In order to filter for failure landing outcomes in drone ship, their booster versions, and launch site names for the year 2015, we combined the usage of the WHERE clause, LIKE, AND, and BETWEEN conditions.

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



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 [32]: %sql SELECT LANDING_OUTCOME FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC;

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



Out[32]:

Landing_Outcome
No attempt
Success (ground pad)
Success (drone ship)
Success (drone ship)
Success (ground pad)
Failure (drone ship)
Success (drone ship)
Success (drone ship)
Success (drone ship)
Failure (drone ship)
Failure (drone ship)
Success (ground pad)
Precluded (drone ship)
No attempt
Failure (drone ship)
No attempt
Controlled (ocean)
Failure (drone ship)
Uncontrolled (ocean)

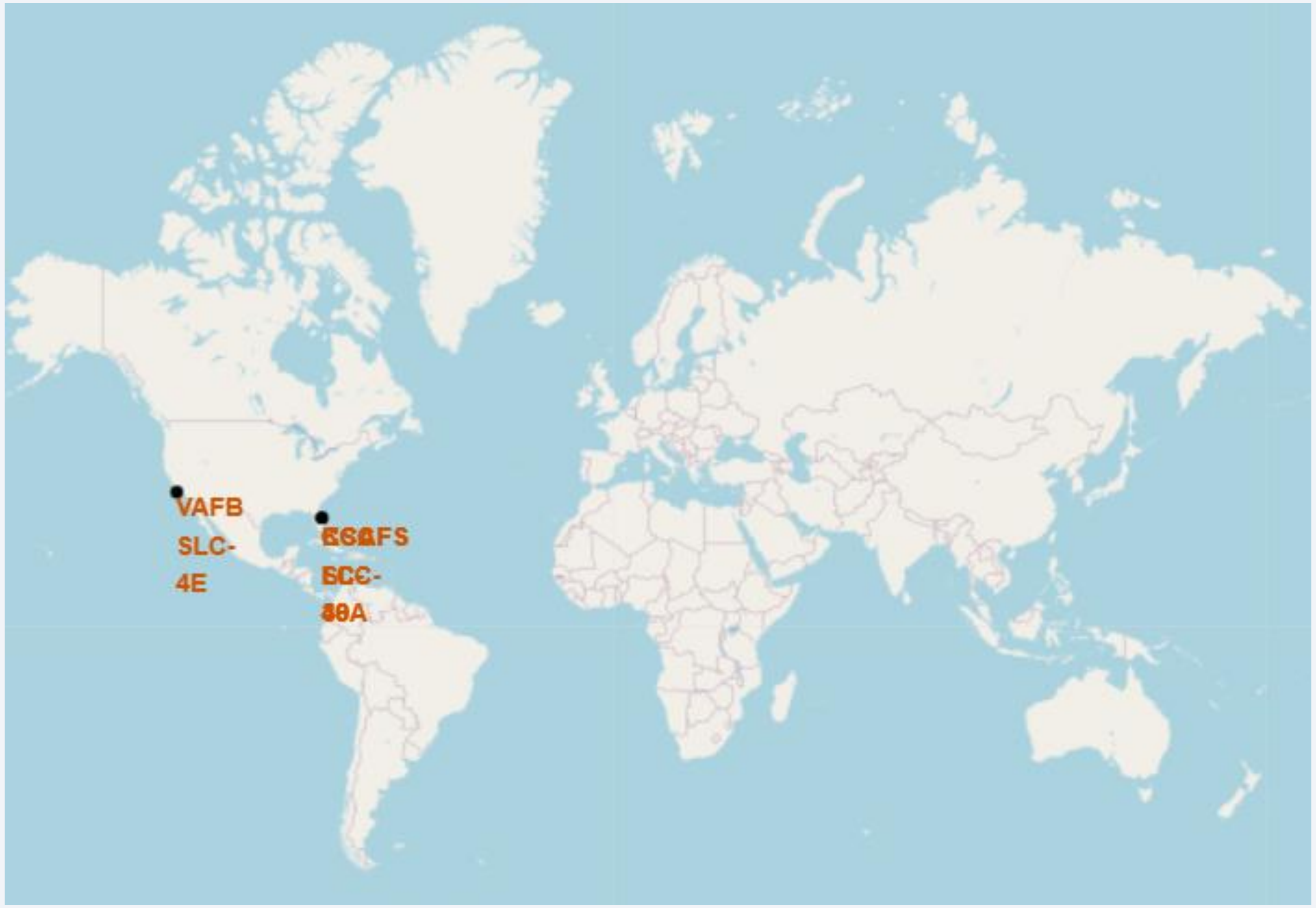
- Using the WHERE clause, we filtered the data for landing outcomes BETWEEN 2010-06-04 and 2010-03-20. We also picked the landing outcomes and the COUNT of landing outcomes.
- The landing outcomes were sorted using the GROUP BY clause, and the grouped landing outcomes were then arranged in descending order using the ORDER BY clause.

Section 3

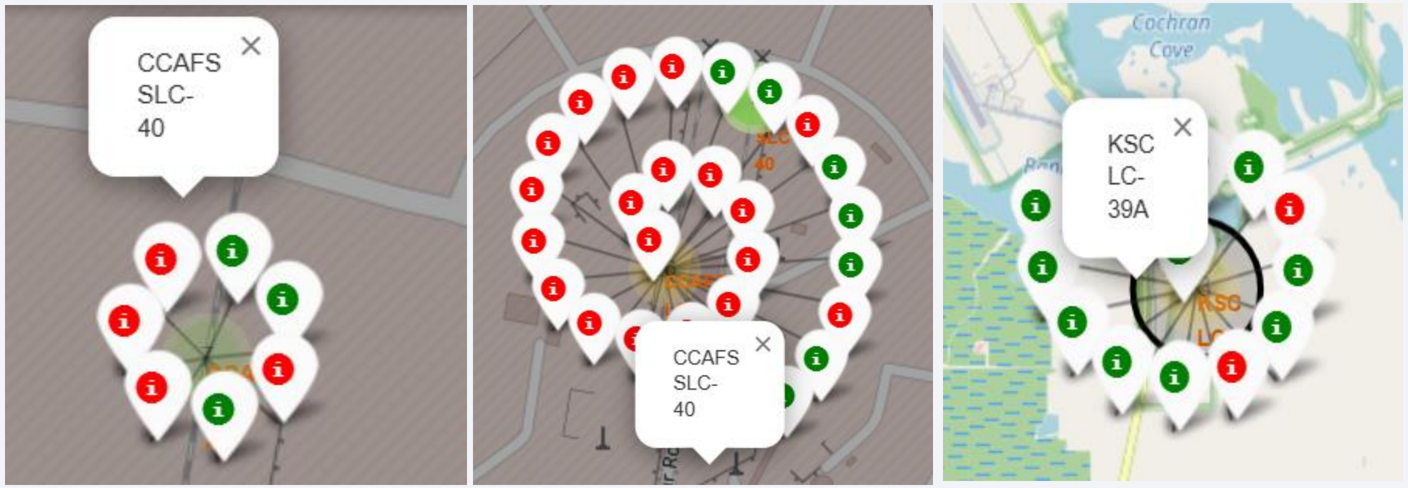
Launch Sites Proximities Analysis

All Launch Sites in the World Map

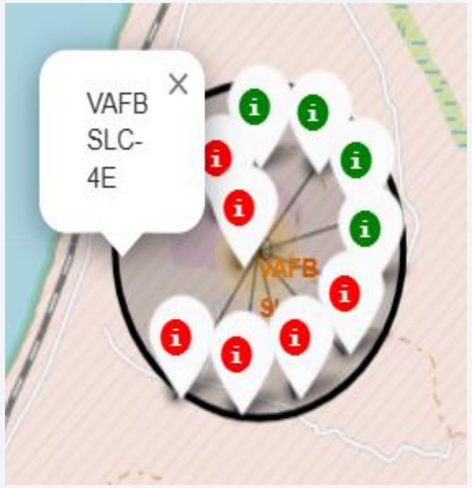
- All launch sites are in proximity to the Equator line.
- All launch sites are in very close proximity to the coast.
- All launch sites are in United States of America.



Color Labeled Launch Outcomes



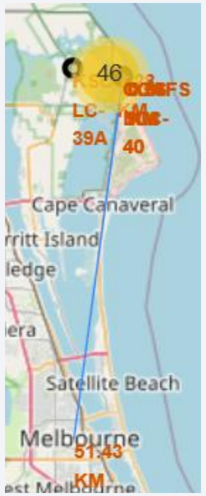
Florida Launch Sites



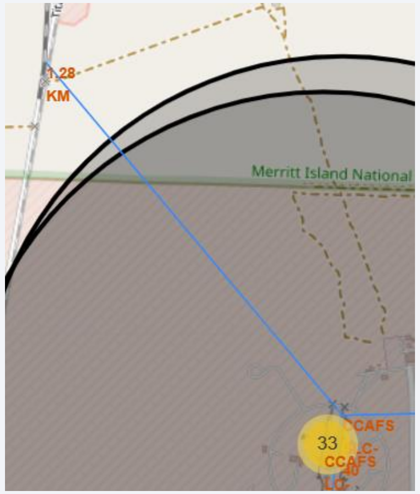
California Launch Sites

Green – Successful Launches
Red – Unsuccessful Launches

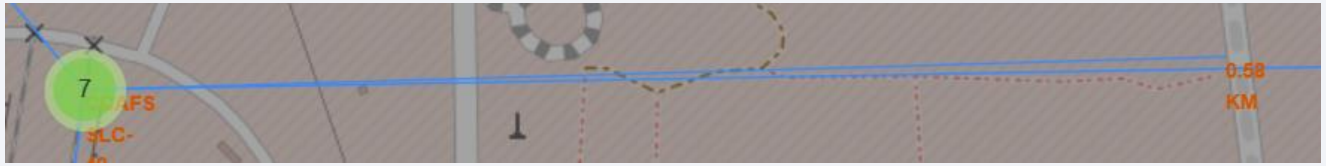
Launch Sites for Proximities



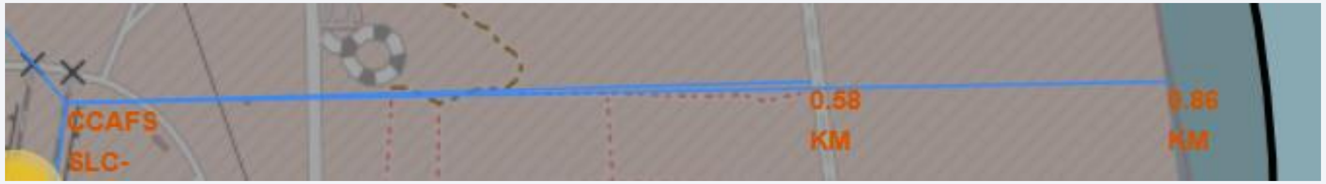
Town



Railway



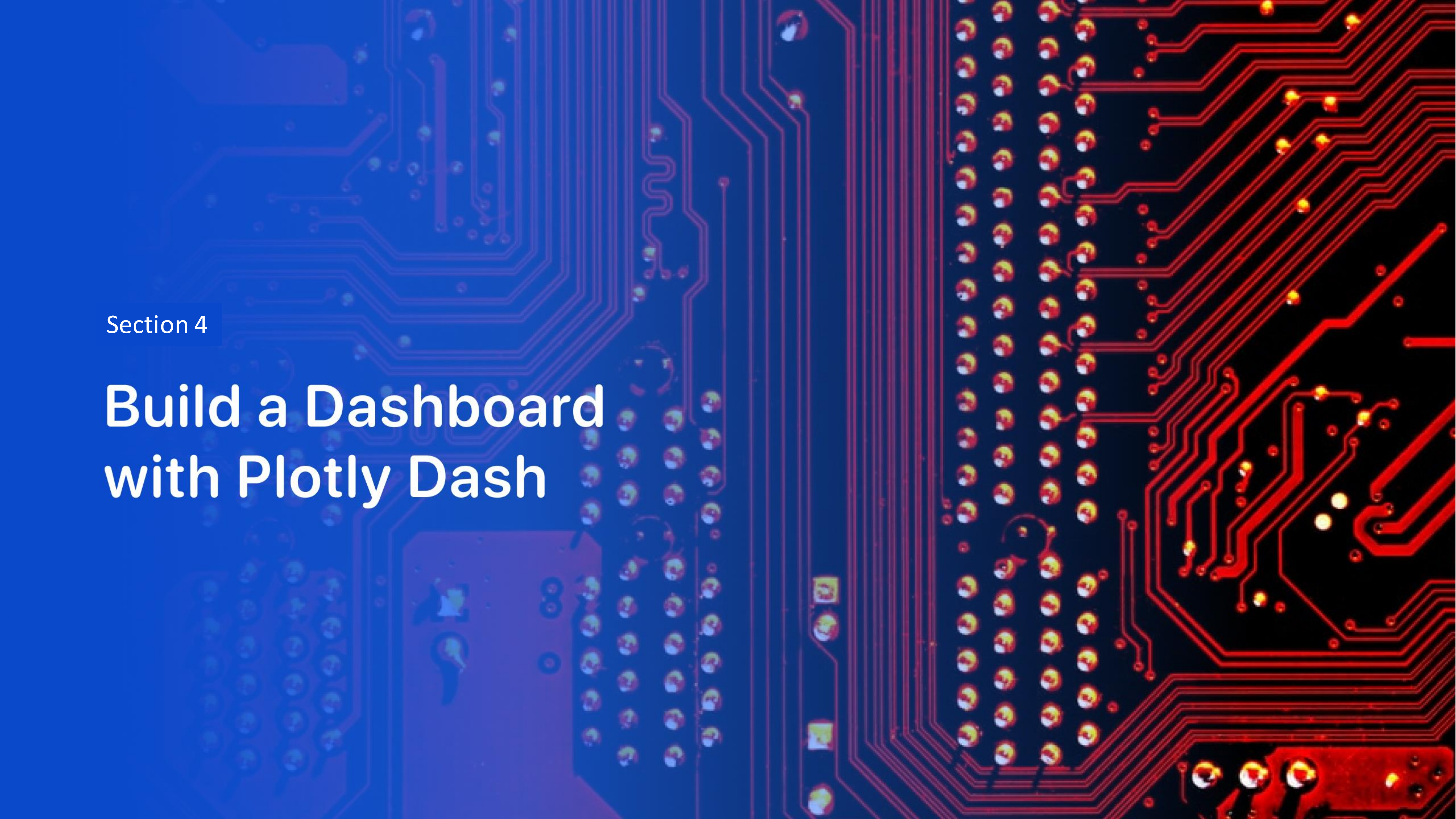
Highway



Coastline

- Distance to railways, highways, coastline and cities from a specific launch site is distributed according to follows;

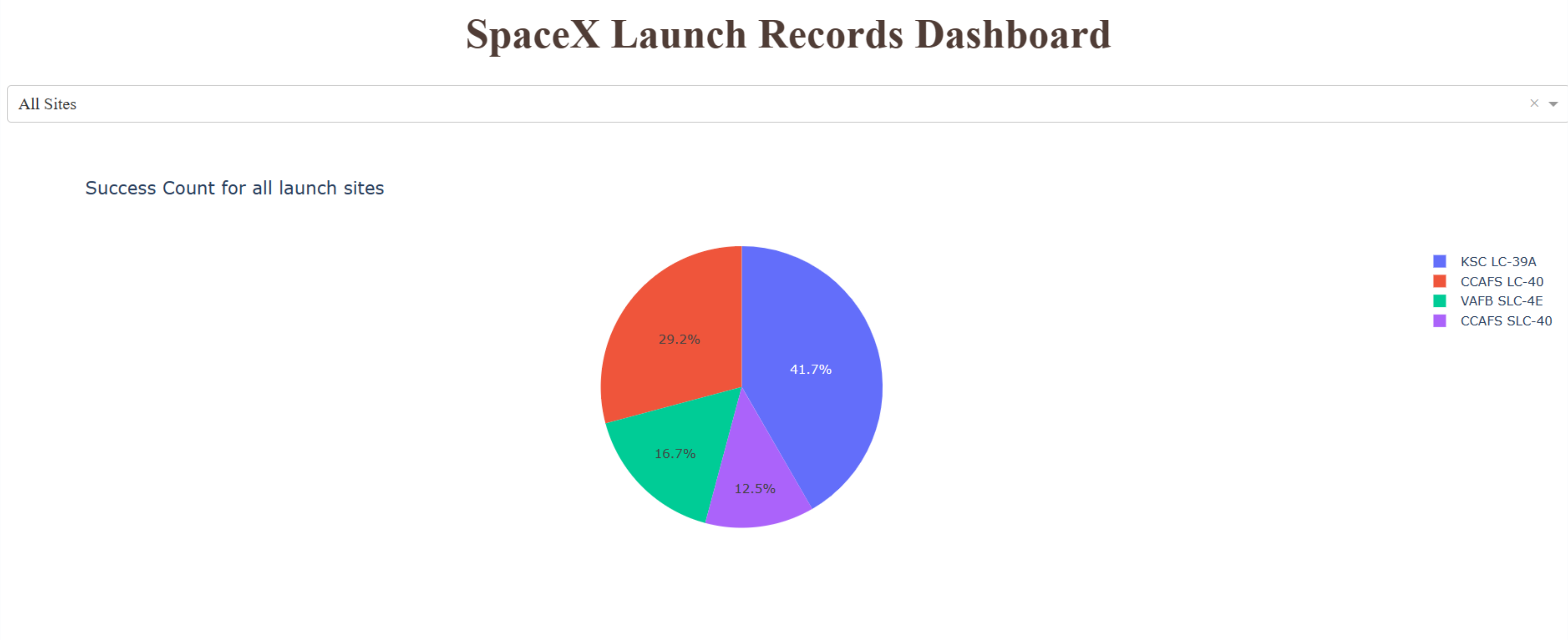
  Distance to cities > Distance to railways > Distance to coastlines > Distance to highways
- Distance to cities from a launch site is higher to prevent the disturbances and adverse effects of the launches.
- Launch sites situated close proximity to the highways and railways to easy transportation purposes.
- Launch sites situate close proximity to the coastline due to need of high loads of water to cooling purposes and also for launch safety.



Section 4

Build a Dashboard with Plotly Dash

Success Count for All Sites

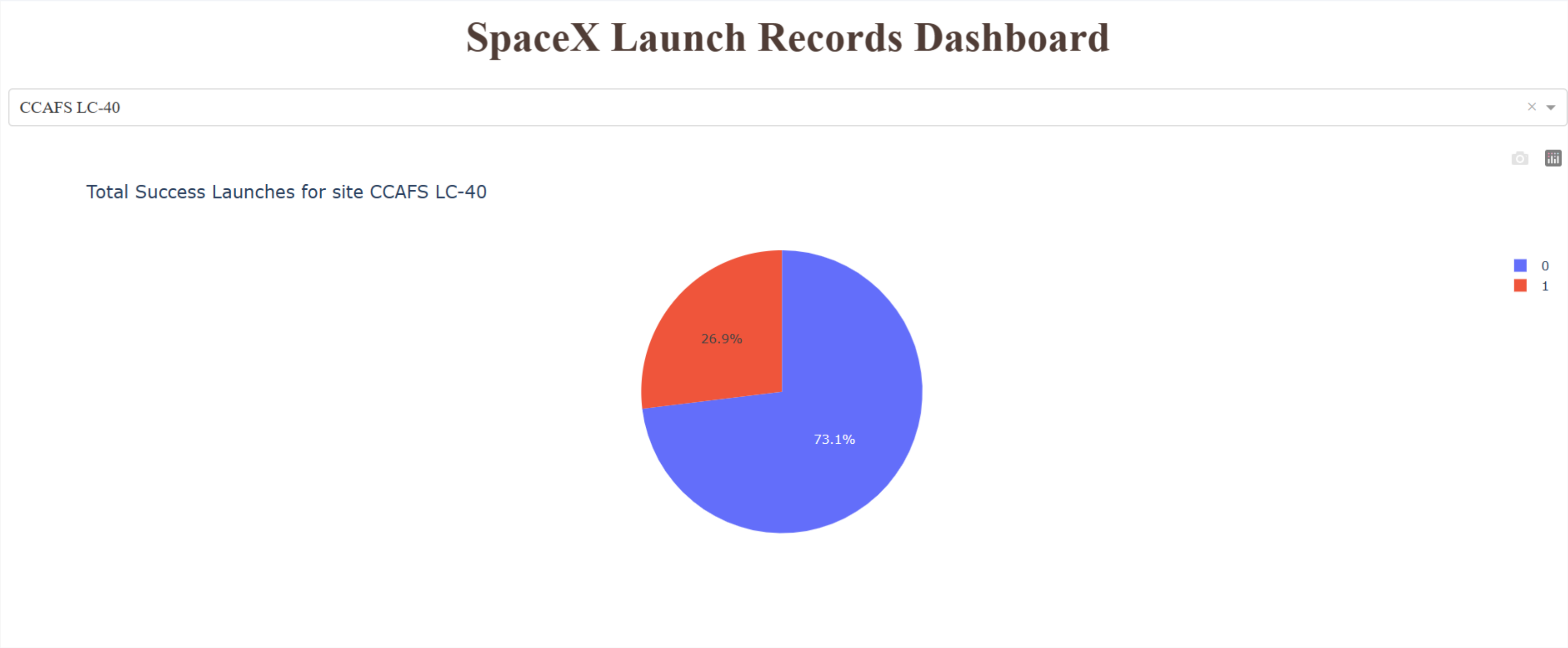


SpaceX Launch Records Dashboard

All Sites

- The highest success count is for KSC LC- 39A Launch site.
- The lowest success is for CCAFS SLC – 40 Launce site.

Highest Launch Success Ratio



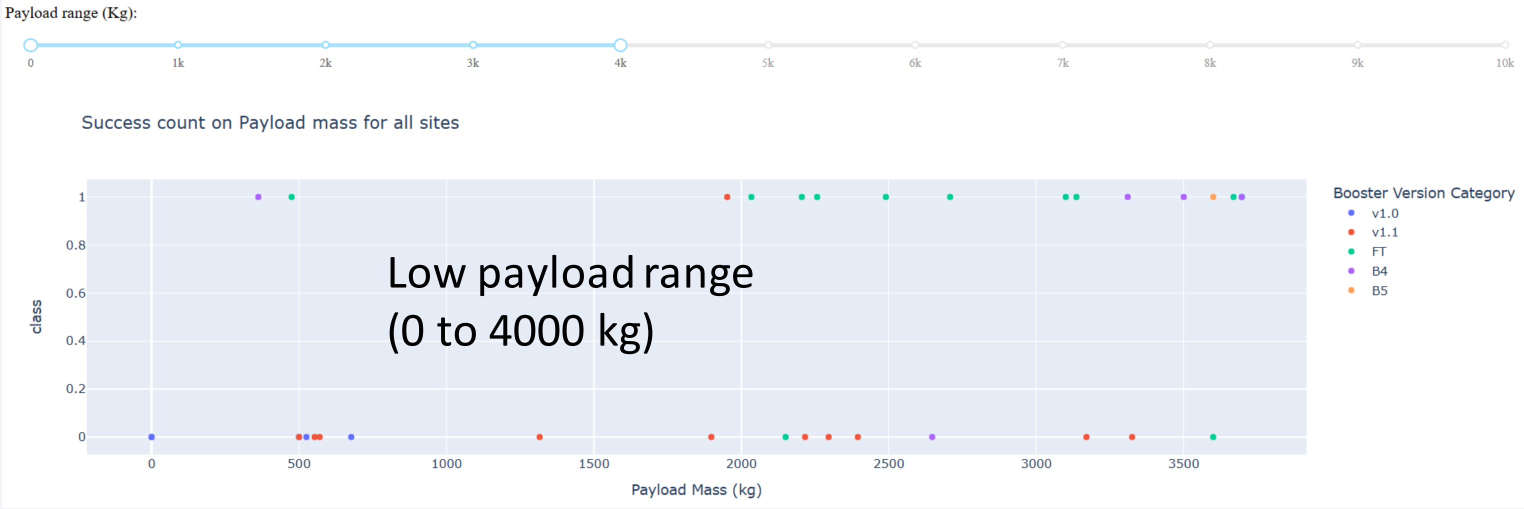
SpaceX Launch Records Dashboard

CCAFS LC-40

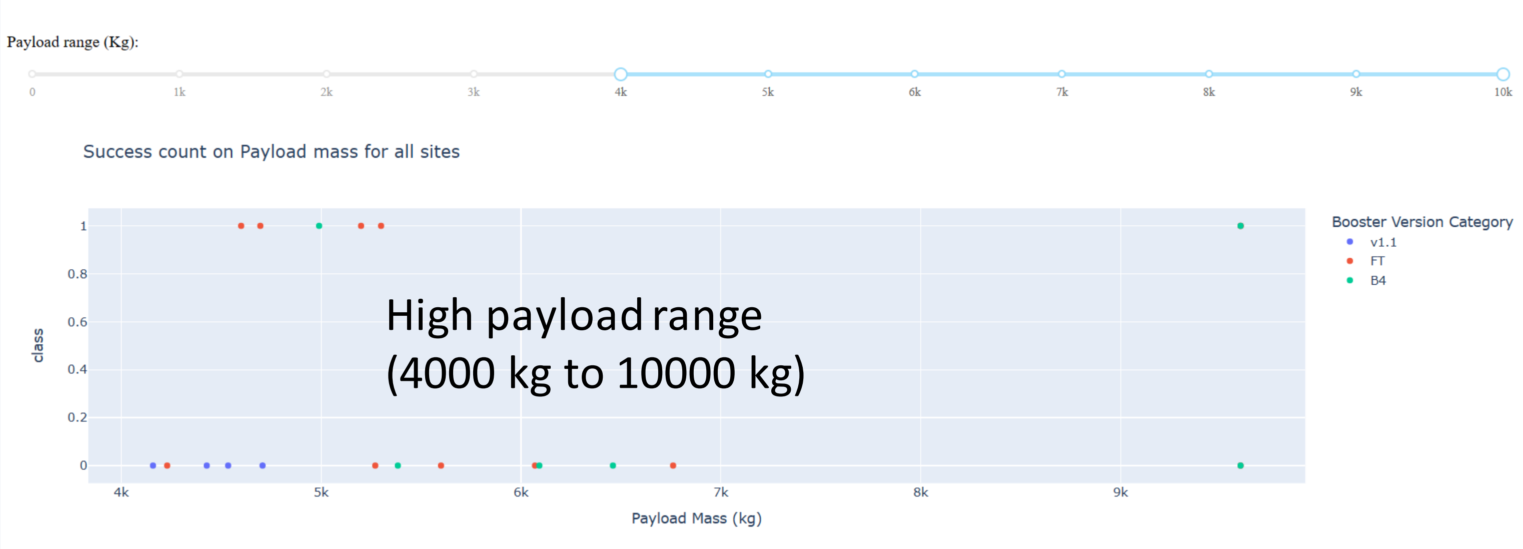
Total Success Launches for site CCAFS LC-40

- CCAFS LC – 40 Launch Site has the highest success ratio which is 73.1%.

Playload Vs. Launch Outcome



Low payload range
(0 to 4000 kg)



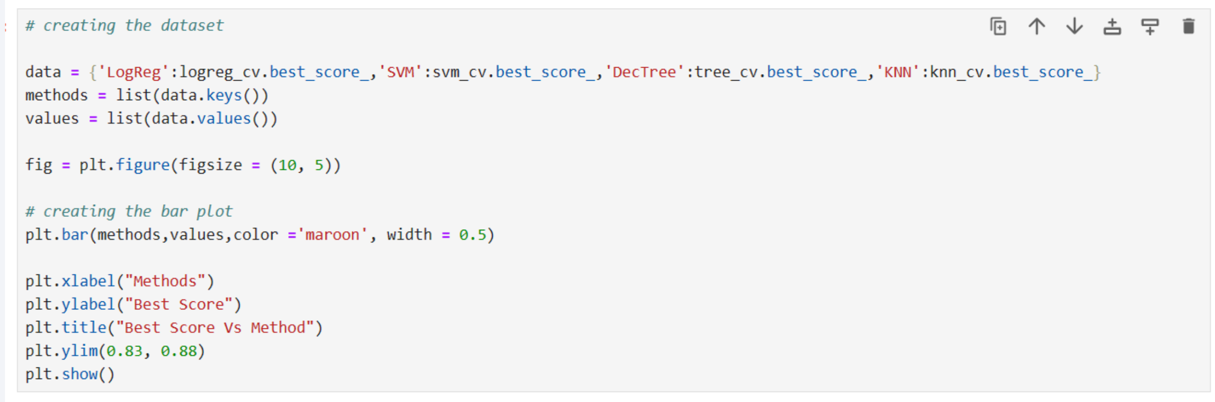
High payload range
(4000 kg to 10000 kg)

Success rate for low payload range (0 – 4000 kg) is higher than the high payload range (4000 kg – 1000 kg)

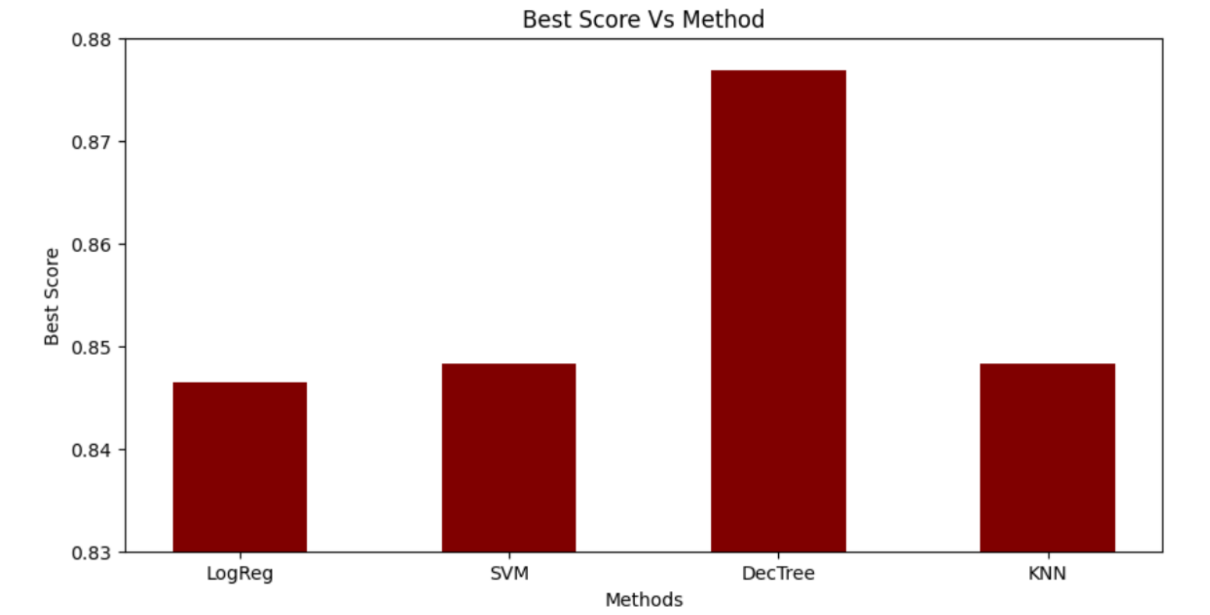
Section 5

Predictive Analysis (Classification)

Classification Accuracy



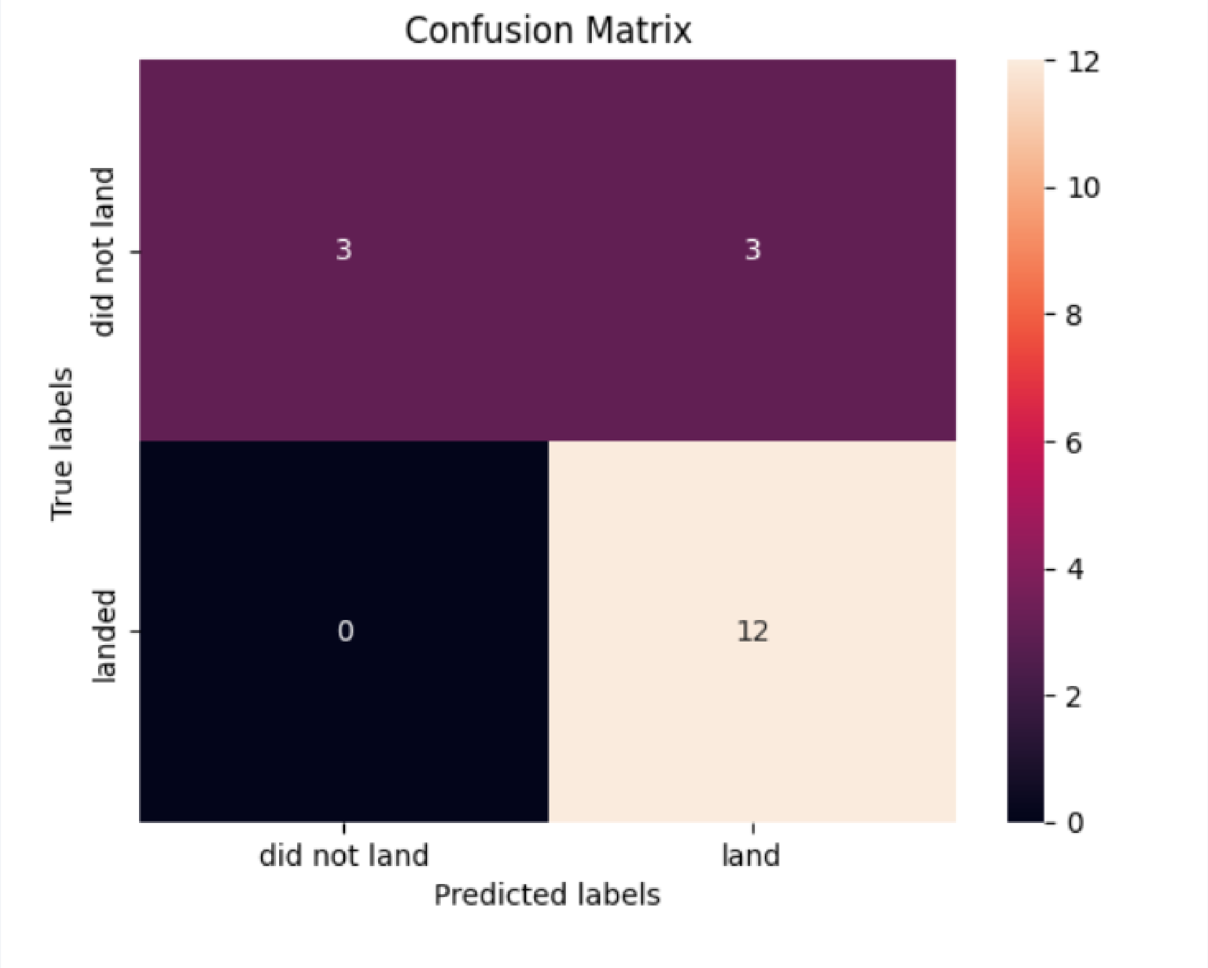
```
# creating the dataset

data = {'LogReg':logreg_cv.best_score_,'SVM':svm_cv.best_score_,'DecTree':tree_cv.best_score_,'KNN':knn_cv.best_score_}
methods = list(data.keys())
values = list(data.values())

fig = plt.figure(figsize = (10, 5))

# creating the bar plot
plt.bar(methods,values,color ='maroon', width = 0.5)

plt.xlabel("Methods")
plt.ylabel("Best Score")
plt.title("Best Score Vs Method")
plt.ylim(0.83, 0.88)
plt.show()
```



- Decision tree model is better for train accuracy. But test accuracy is similar.

Confusion Matrix

- The decision tree classifier's confusion matrix demonstrates the classifier's ability to discriminate between the various classes.
- False positives are the main issue. That is, the classifier interprets an unsuccessful landing as a successful landing.



Conclusions

- Therefore, we can say that:
  - The success rate at a launch site increases with the number of flights conducted there.
  - The launch success rate increased from 2013 to 2020.
  - Orbits with the highest success rate were ES-L1, GEO, HEO, SSO, and VLEO.
  - Out of all the sites, KSC LC-39A had the most successful launches.
  - For this problem, the optimal machine learning algorithm is the decision tree classifier.

Discussion

During this project, we predicted if the Falcon 9 first stage will land successfully.

Thank you!