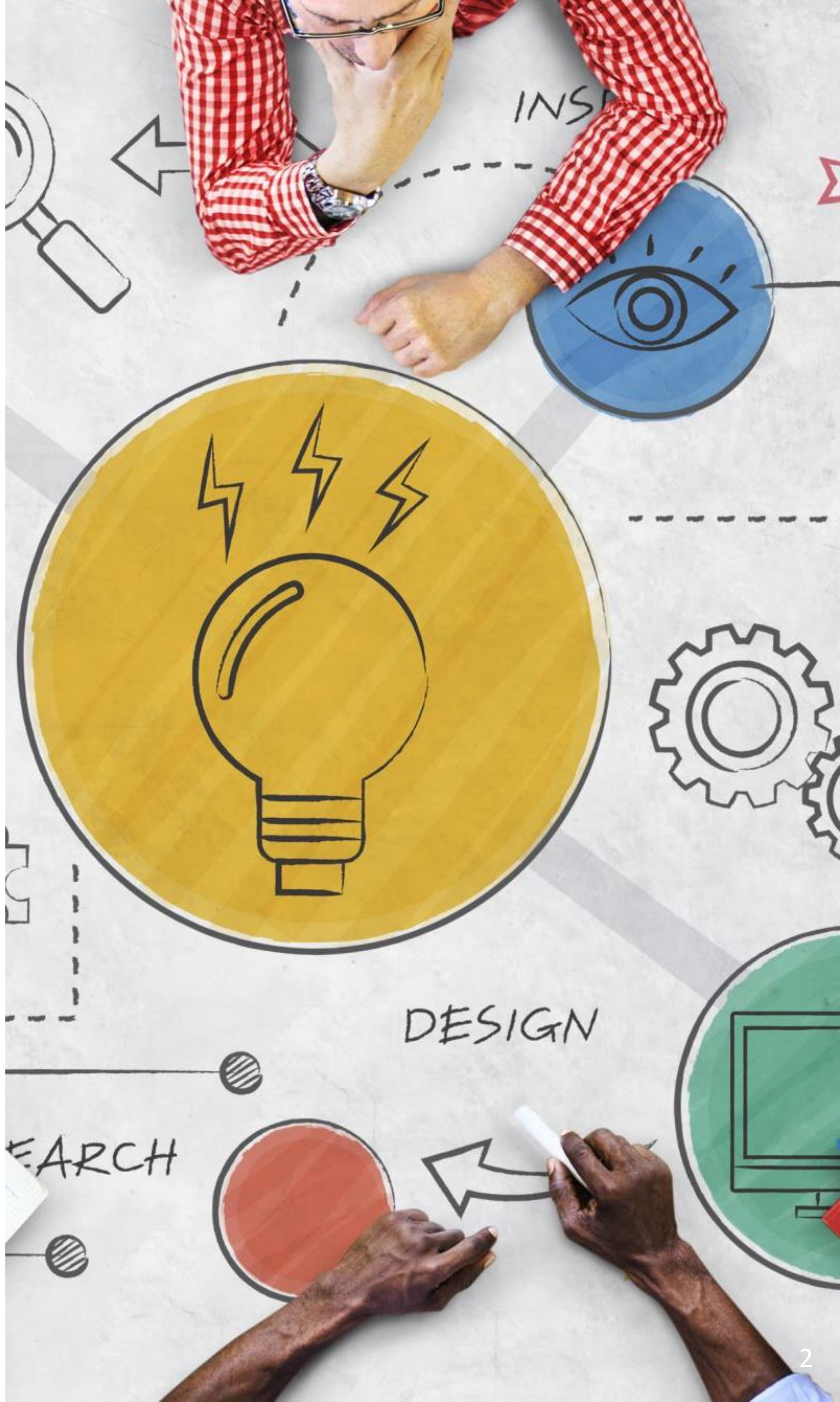
IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Erick Rudelman M.
July 26, 2023

Outline



Executive Summary

Summary of methodologies

- **Data Collection**: Gathering data from API
- **Web Scrapping**: BeautifulSoup and Parsing data
- **Data Wrangling**: Converting outcomes to classes
- **Exploratory Data Analysis**: Use of Machine Learning and SQL
- **Visual Analytics**: Explore and manipulate data in an interactive way with Folium and Plotly Dash
- **Predictive Analysis**: Test Logistic Regression, Support Vector Machine, Decision Tree Classifier and K-nearest Neighbors

Summary of all results

- Data frames, Graphic Visualizations, statistics and insights about SpaceX rocket launching and landing
- Predictive results were obtained to execute analysis on the data

Introduction

Project background and context

SpaceX is the only private company to ever return a spacecraft from low-earth orbit. This is due to their capability to reuse the first stage of their rocket. For SpaceX, rocket launches are relatively inexpensive. For this project, data science techniques are applied in order to build a pipeline and determine meaningful insights if another company wants to compete against Space X. Relevant data will be obtained to predict outcomes such as:

- Price of launch
- If first stage will land successfully or not

Problems you want to find answers

- Cost of each launch

- Landing success rate in each site

- Determine if SpaceX will reuse the first stage

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology
  - Data was collected using the get.response() function from the SpaceX API
- Perform data wrangling
  - Cleaning, transforming, and organizing raw data. Also, creating new features or modifying existing ones to extract relevant information from the data
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build and train the classification model to make predictions on the dataset. Models such as Logistic Regression, Support Vector Machine, Decision Tree and K-Nearest Neighbors are evaluated

Data Collection

Data Collection from SpaceX API

Data was gathered from an API by sending a **requests.get(url)**

URL was used to target endpoint of data: api.spacexdata.com/v4/

To view data the function **response.json()** was used. The function **json_normalize()** was used to build data frames.

Web Scrapping Falcon 9 launch records with BeautifulSoup

Falcon 9 launch records HTML table extracted from Wikipedia

Table was parsed and converted into a DataFrame using BeautifulSoup

Data Collection

Collecting Data

Response = requests.get(“https://api.spacexdata.com/v4/rockets/”)

response.json()

data=pd.json_normalize(response.json())

Web Scrapping with BeautifulSoup

soup= BeautifulSoup(response.text,"html.parser")

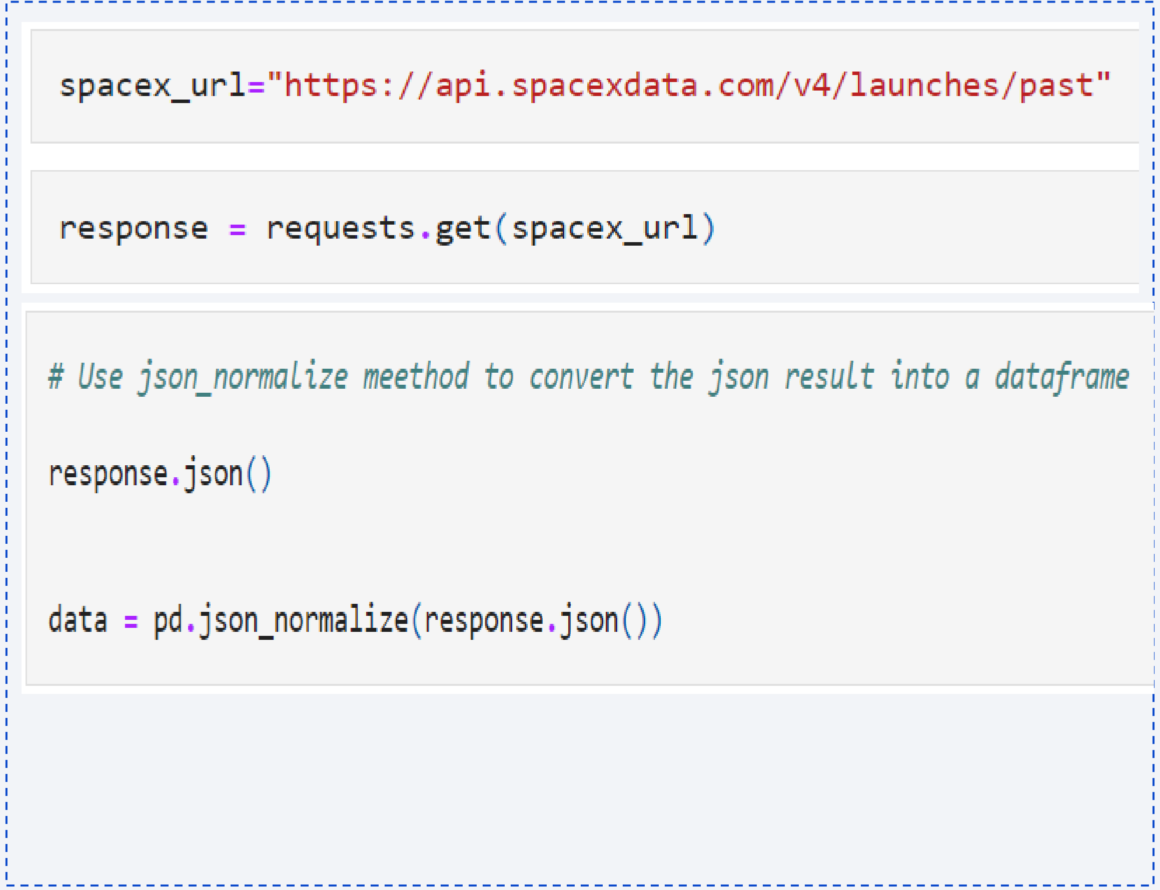
print(soup.title.string)

data=pd.json_normalize(response.json())

Data Collection – SpaceX API

Data is gathered, then decoded and finally normalized to convert into a DataFrame using the Pandas Library.

LINK TO CODE:
https://github.com/rudicr/DataScience-Projects/blob/Winning-Space-Race-with-Data-Science/1.jupyter-labs-spacex-data-collection-api.ipynb



```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

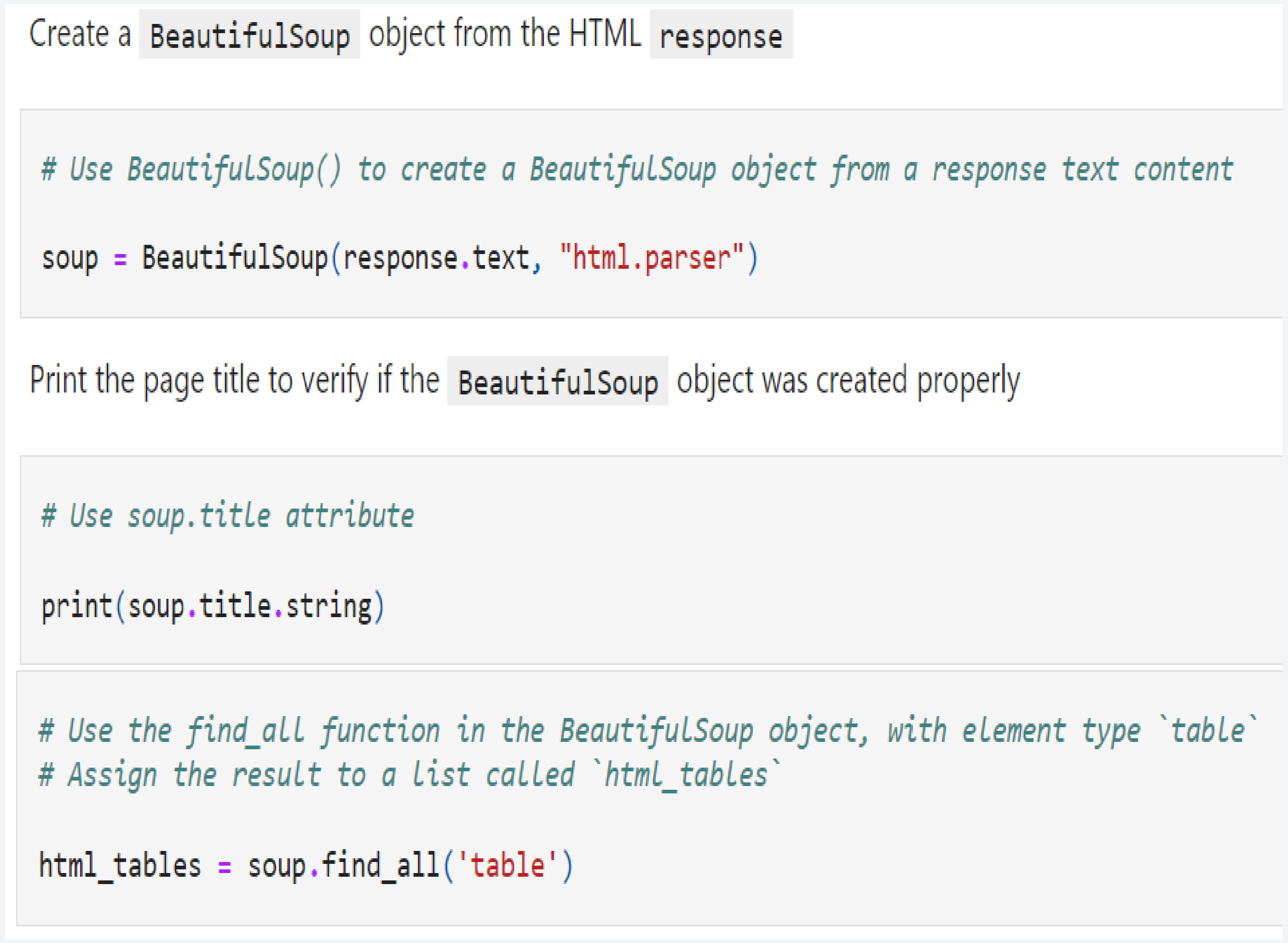
```
response = requests.get(spacex_url)
```

```
# Use json_normalize meethod to convert the json result into a dataframe

response.json()


data = pd.json_normalize(response.json())
```

Data Collection - Scraping

BeautifulSoup object created in order to collect relevant data from the HTML response. Tables and DataFrames are created to work with specific data columns.

LINK TO CODE:
https://github.com/rudicr/DataScience-Projects/blob/Winning-Space-Race-with-Data-Science/2.jupyter-labs-webscraping.ipynb



Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content

soup = BeautifulSoup(response.text, "html.parser")
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
# Use soup.title attribute

print(soup.title.string)
```

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`

html_tables = soup.find_all('table')
```

Data Wrangling

- Processing Data
  - Firstly, the method **value_counts()** was used to determine the number of launches per site and the number of occurrences of each orbit. Same method was used to determine the number of landing outcomes. The landing outcomes were classified with the values 0 or 1.

```
df.LaunchSite.value_counts()
```

```
df.Orbit.value_counts()
```

```
landing_outcomes = df.Outcome.value_counts()
```

LINK TO CODE:
https://github.com/rudicr/DataScience-Projects/blob/Winning-Space-Race-with-Data-Science/3.%20IBM-DS0321EN-SkillsNetwork_labs_module_1_L3_labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

EDA with Data Visualization

- The following charts were used:
  - Categorical Plots were used to display the relation between a numerical and a categorical value.
    - Bar Plot is used since its effective for comparing the quantities of different categories.
  - Scatter Plots are used to portray how one continuous variable can affect another.
  - A Line Chart was used to portray how the success rate changes over time.

LINK TO CODE:
https://github.com/rudicr/DataScience-Projects/blob/Winning-Space-Race-with-Data-Science/5.%20IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- SQL queries were used to obtain the following:
  - Display unique launch sites
  - Display 5 records where launch sites begins with ‘CCA’
  - Display total payload mass
  - Display average payload mass by specific booster F9 v1.1
  - List dates of successful landing
  - Boosters' names with drone ship success and a payload mass specification.
  - Total number of successful and failure mission outcomes
  - Rank count of landing outcomes

LINK TO CODE:
https://github.com/rudicr/Data Science-Projects/blob/Winning-Space-Race-with-Data-Science/4.jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Circles were created to enclose launch sites and the NASA Johnson Space Center.
- Markers were used to show these exacts points; They were also used to mark selected coastlines, highways, cities and railways.
- The PolyLine was added as a line to show the distance between each launch site and different objects.
- The MousePosition was added to get a coordinate for a mouse over a point.
- Marker clusters with colors were used to portray successful launches per site.

LINK TO CODE:
https://github.com/rudicr/DataScience-Projects/blob/Winning-Space-Race-with-Data-Science/6.IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite%20(1).ipynb

Build a Dashboard with Plotly Dash

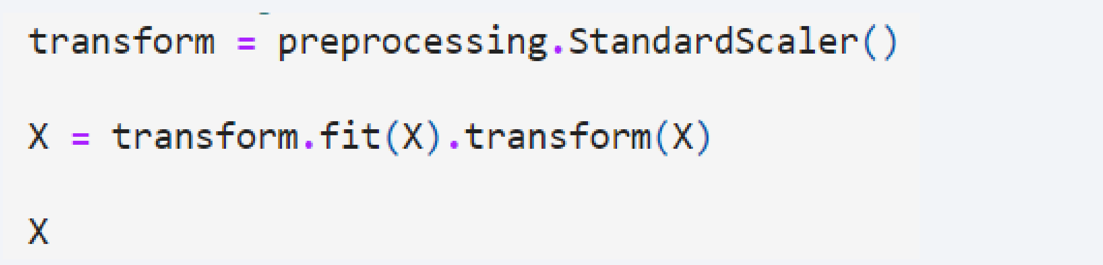
- A pie chart was created to portray the launches per launch site.
- The scatter plot was created to portray the relationship between Payload Mass and Success Outcome for the several booster versions.

LINK TO CODE:
https://github.com/rudicr/DataScience-Projects/blob/Winning-Space-Race-with-Data-Science/7.%20spacex_dash_app.py

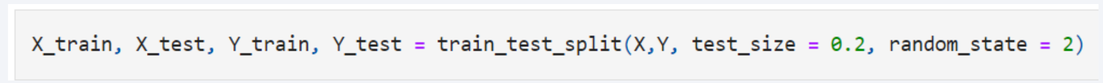
Predictive Analysis (Classification)

- Predictive analysis was done by building and training the classification model to make predictions on the dataset.
- Models such as Logistic Regression, Support Vector Machine, Decision Tree and K-Nearest Neighbors are evaluated.
- GridSearchCV is used to turn hyperparameters into machine learning models. These are parameters that were not learned during the training but need to be set. The Grid Search is used to evaluate the performance of a model by splitting the data and training the model to obtain a more robust estimate of the model's performance and avoid overfitting.

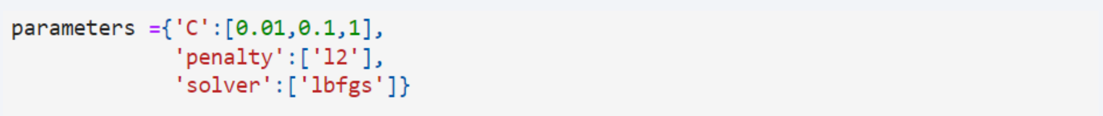
<u>LINK TO CODE:</u>
https://github.com/rudicr/DataScience-Projects/blob/Winning-Space-Race-with-Data-Science/8.%20IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb



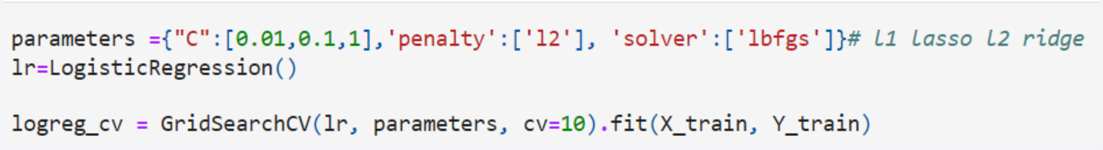
```
transform = preprocessing.StandardScaler()

X = transform.fit(X).transform(X)

X
```



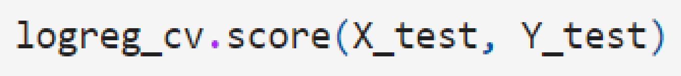
```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.2, random_state = 2)
```



```
parameters ={'C':[0.01,0.1,1],
             'penalty':['l2'],
             'solver':['lbfgs']}
```



```
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge
lr=LogisticRegression()

logreg_cv = GridSearchCV(lr, parameters, cv=10).fit(X_train, Y_train)
```



```
logreg_cv.score(X_test, Y_test)
```

Results

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

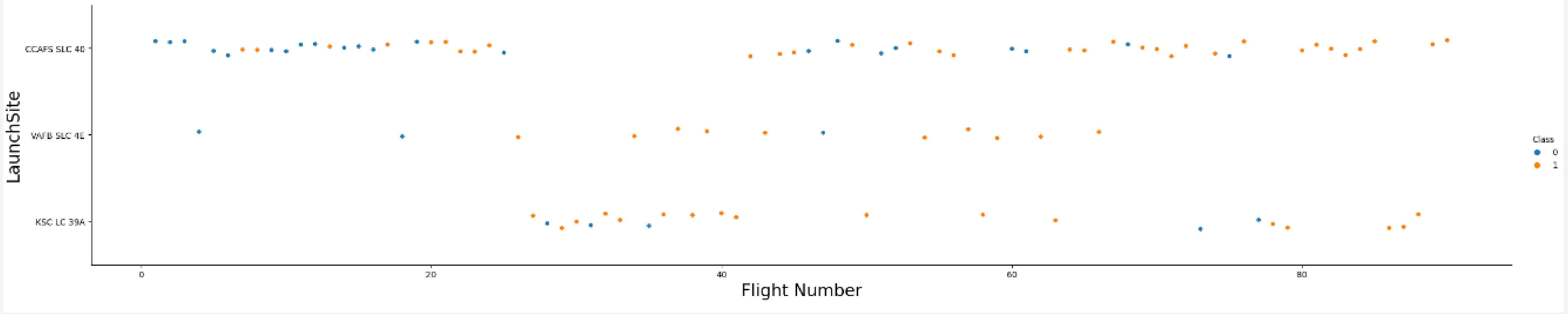
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

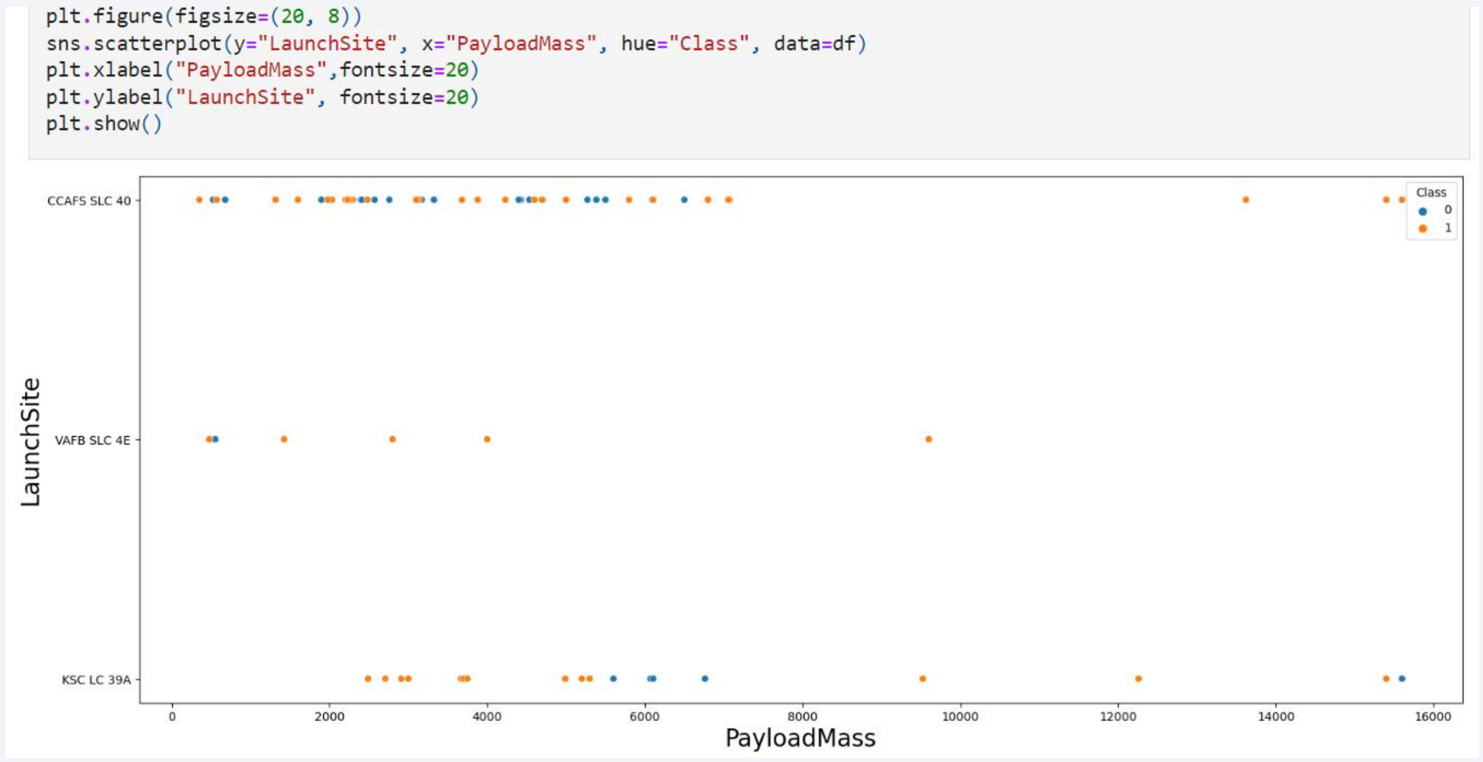
- We can see a scatter plot between FlightNumber and Payload Mass. As the flight number increases, the first stage is more likely to land successfully. Payload Mass is also relevant and the bigger it is, the less likely the first stage will return.

```
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("LaunchSite", fontsize=20)
plt.show()
```



Payload vs. Launch Site

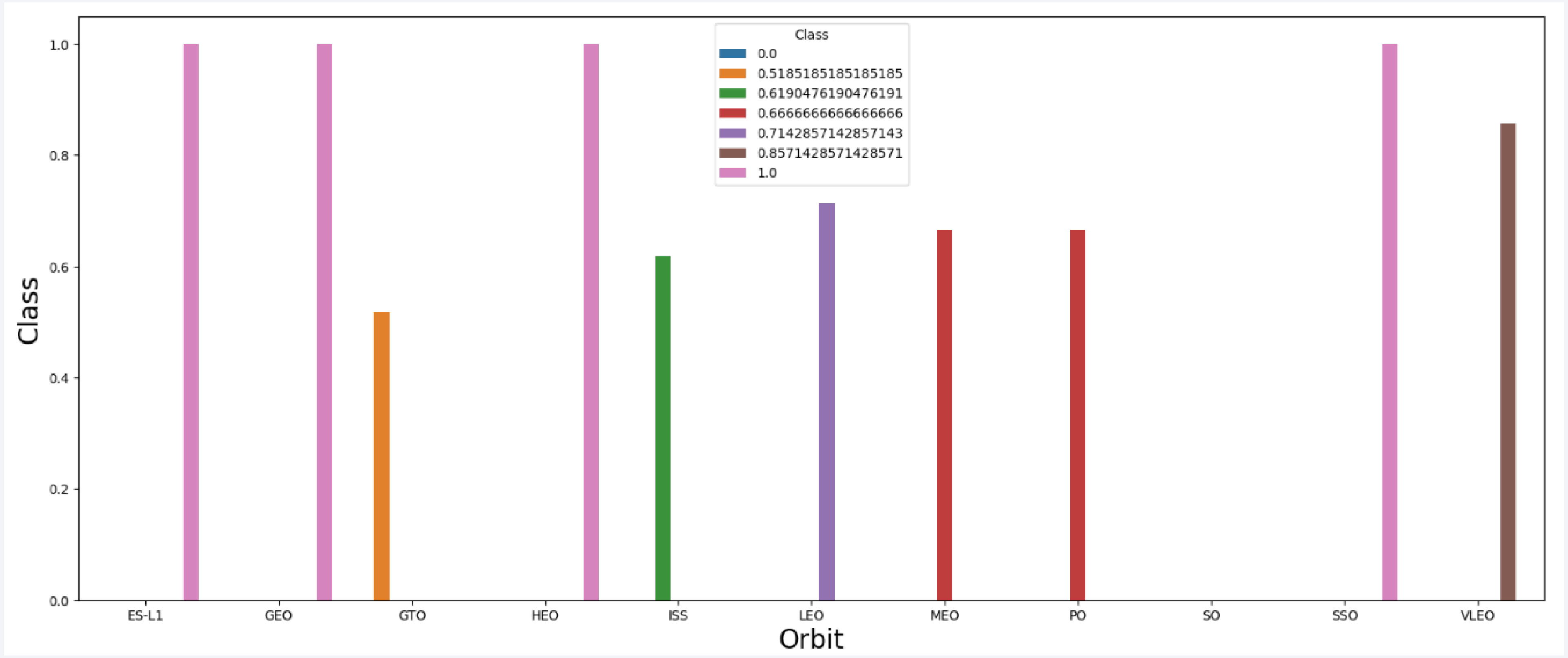
- It can be observed that for the VAFB-SLC site, there are no rockets launched for heavy Payload Mass greater than 10,000 kg.
- For the KSC LC-39A there are no launches for Payload Mass lower than 2,000 Kg.
- For CCAFS SLC-40 an increase in the rate of successful launches can be observed as the Payload Mass increases.



```
plt.figure(figsize=(20, 8))
sns.scatterplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("LaunchSite", fontsize=20)
plt.show()
```

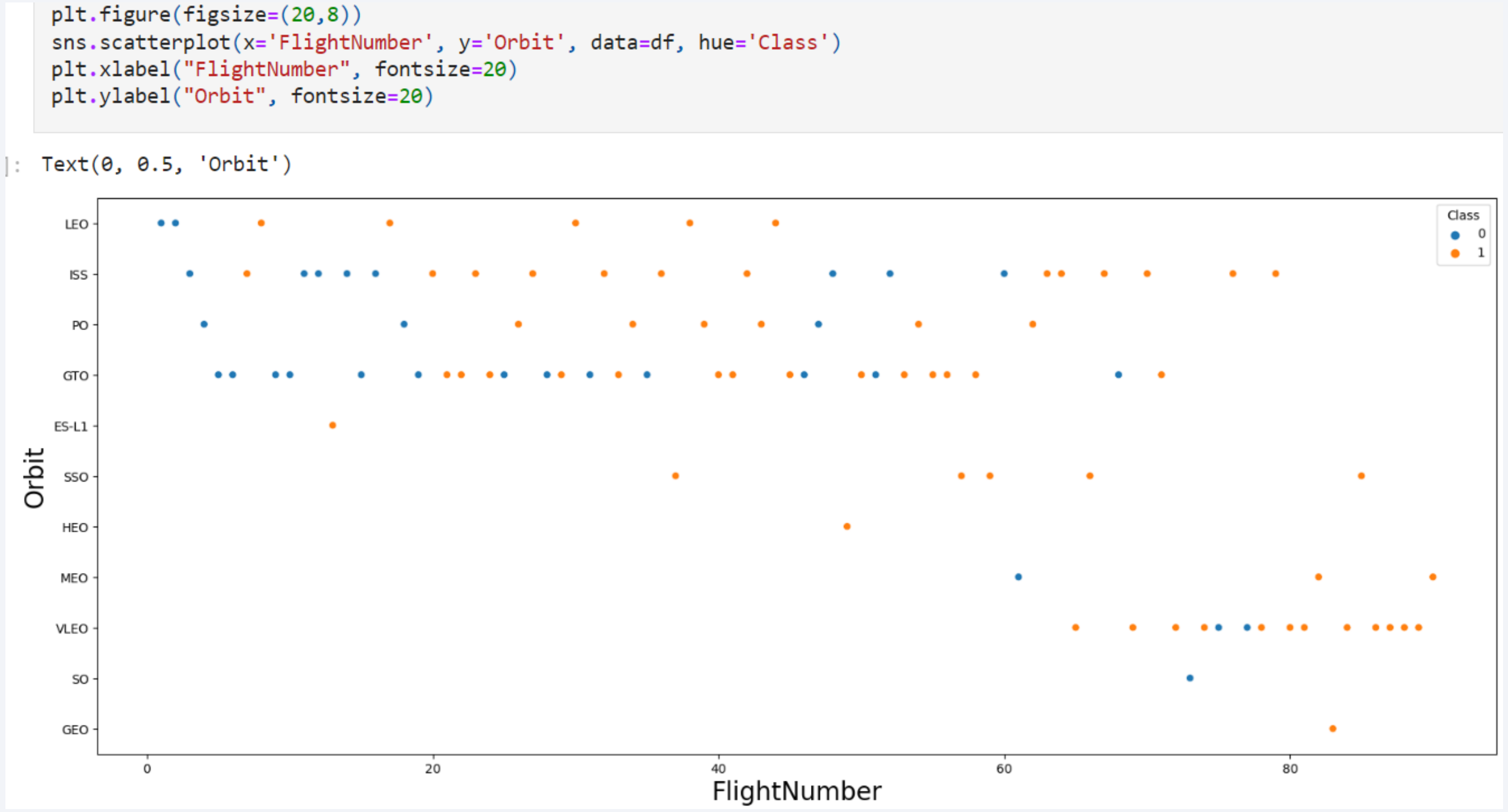
Success Rate vs. Orbit Type

- A bar chart is used to portray the success rate grouped by orbit type. It can be observed that ES-L1, GEO, HEO, and SEO are the orbits with the highest success rate.



Flight Number vs. Orbit Type

- It can be observed that in the LEO and SSO orbit the success appears related to the number of flights.
- For the GTO orbit there seems to be no relation with the number of flights.

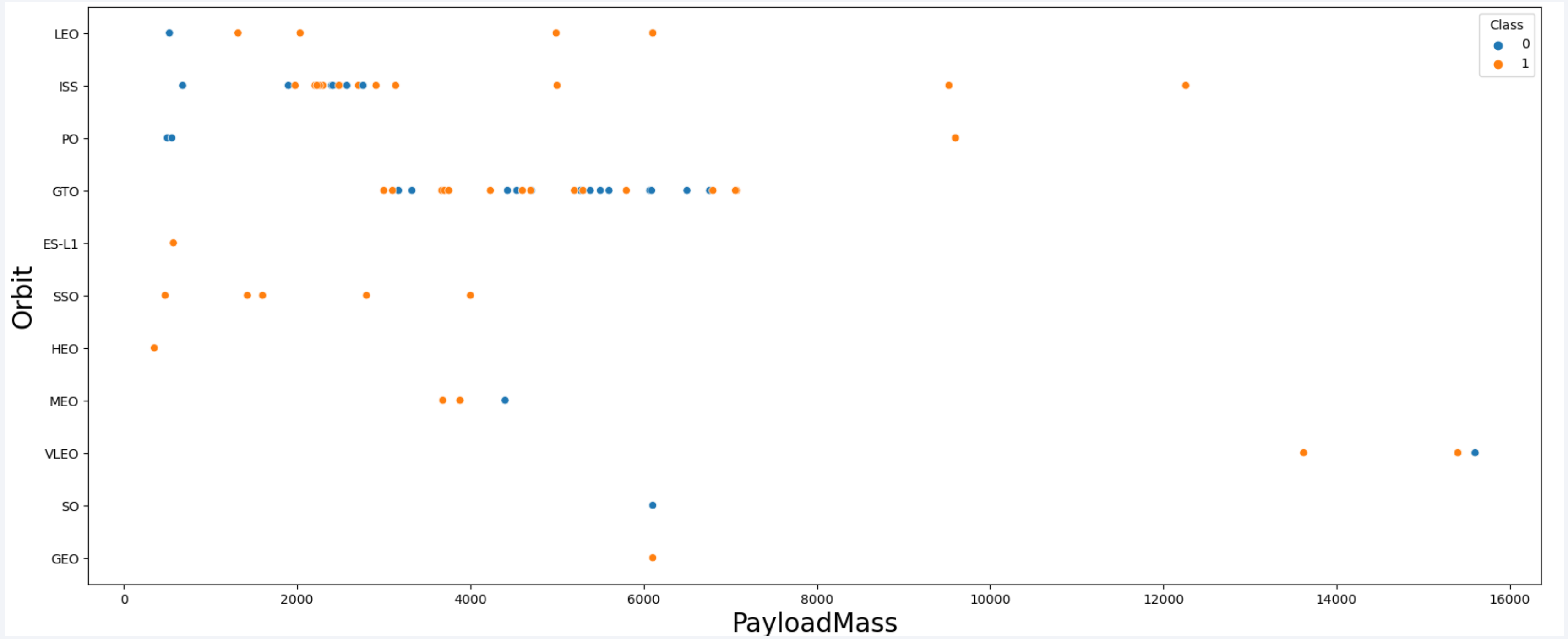


```
plt.figure(figsize=(20,8))
sns.scatterplot(x='FlightNumber', y='Orbit', data=df, hue='Class')
plt.xlabel("FlightNumber", fontsize=20)
plt.ylabel("Orbit", fontsize=20)
```

```
Text(0, 0.5, 'Orbit')
```

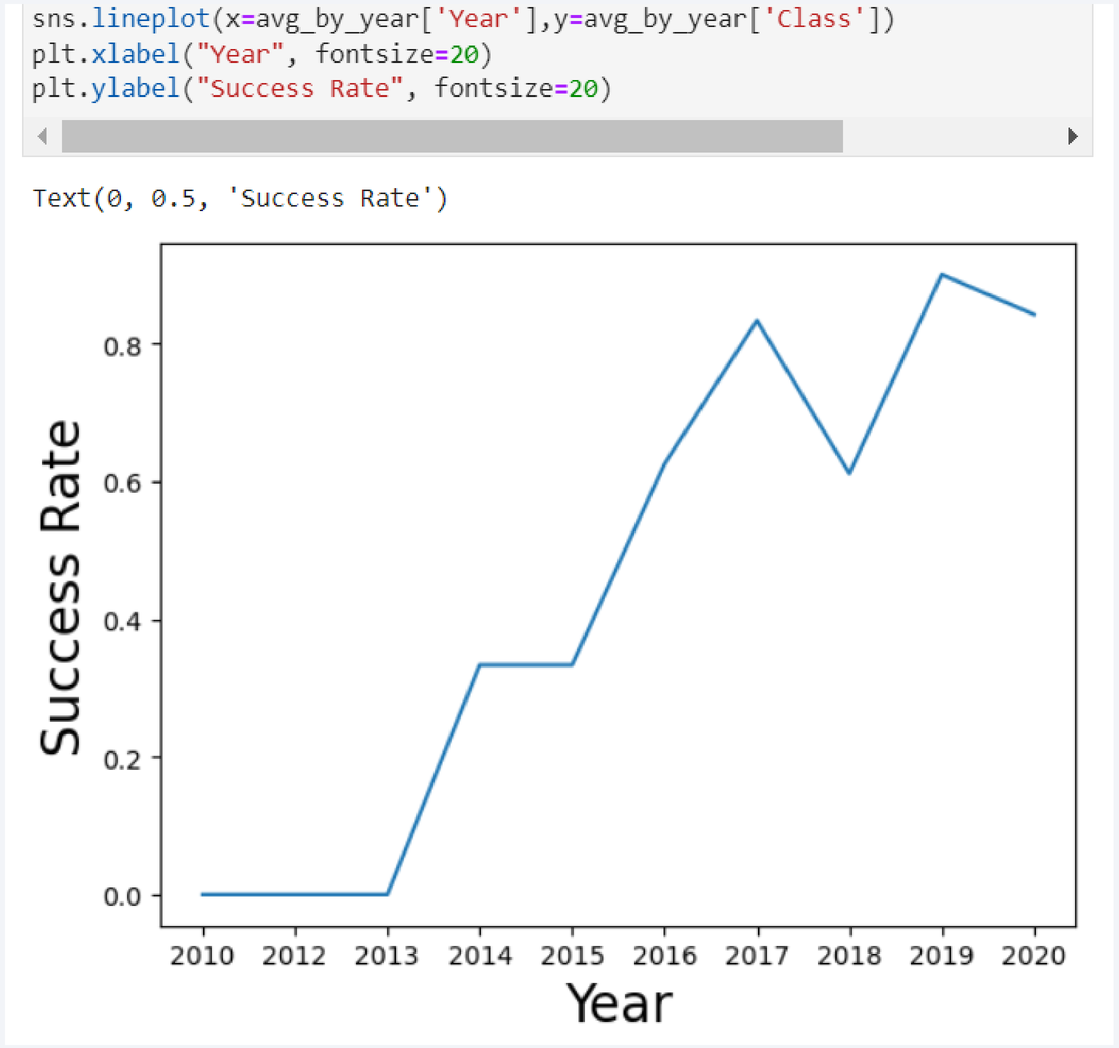
Payload vs. Orbit Type

- It can be observed that for Polar, LEO and ISS the landing success rate increases as the Payload Mass increases.
- For SSO orbit there seems to be higher success rate as the Payload Mass decreases.
- No meaningful insights can be deduced for the GTO orbit.



Launch Success Yearly Trend

- This line graph is useful to portray how the success rate changes over time. It can be observed that it has been increasing over time.
- It first dropped after 2017 and resurged to peak in the year 2019.

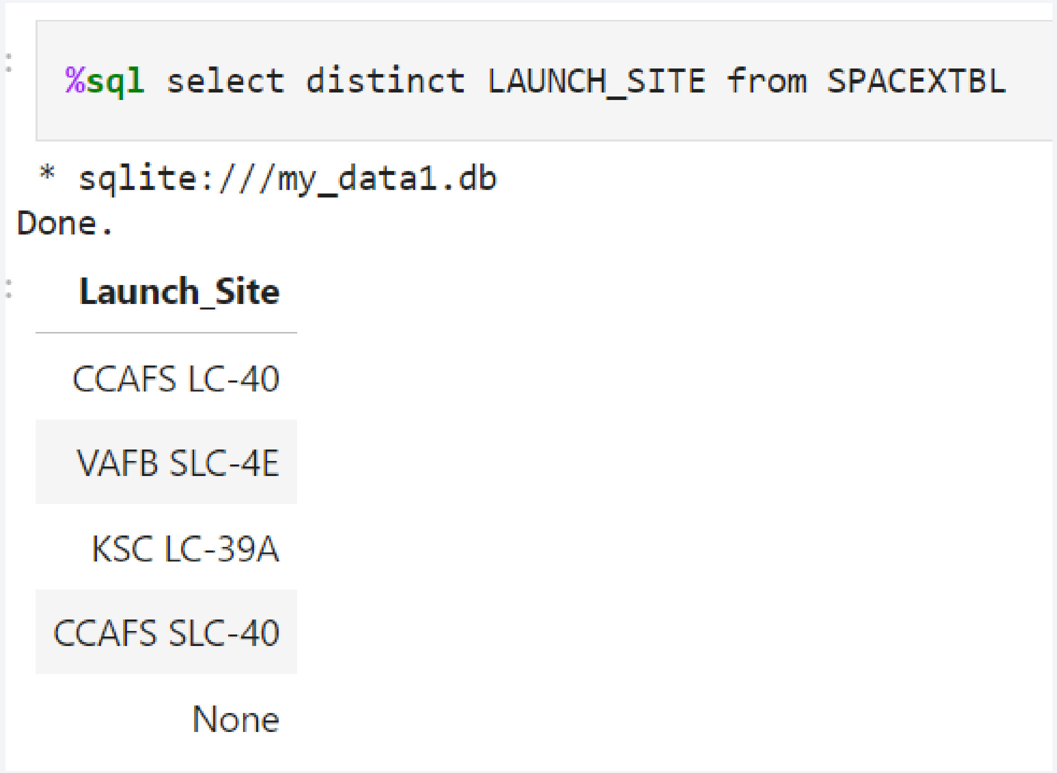


```
sns.lineplot(x=avg_by_year['Year'],y=avg_by_year['Class'])
plt.xlabel("Year", fontsize=20)
plt.ylabel("Success Rate", fontsize=20)
```

```
Text(0, 0.5, 'Success Rate')
```

All Launch Site Names

- Unique launch sites
  - CCAFS LC-40
  - CCAFS SLC-40
  - KSC LC-39A
  - VAFB SLC-4E
- Unique launch sites were obtained by using the DISTINCT query.



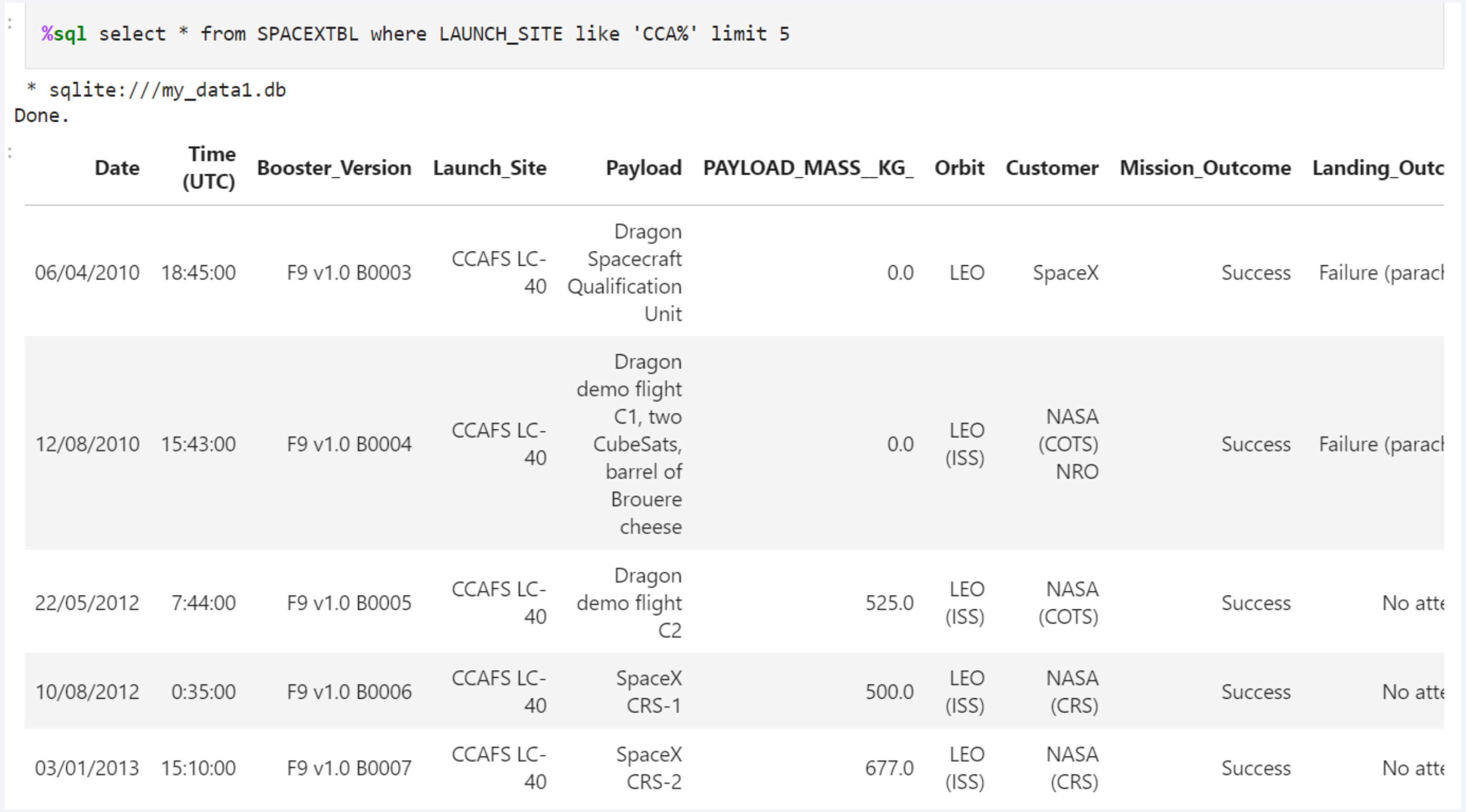
```
%sql select distinct LAUNCH_SITE from SPACEXTBL
```

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

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Launch Site Names Begin with 'CCA'

- The keywords WHERE, LIKE and LIMIT were used to extract 5 records where launch sites begin with ‘CCA’.



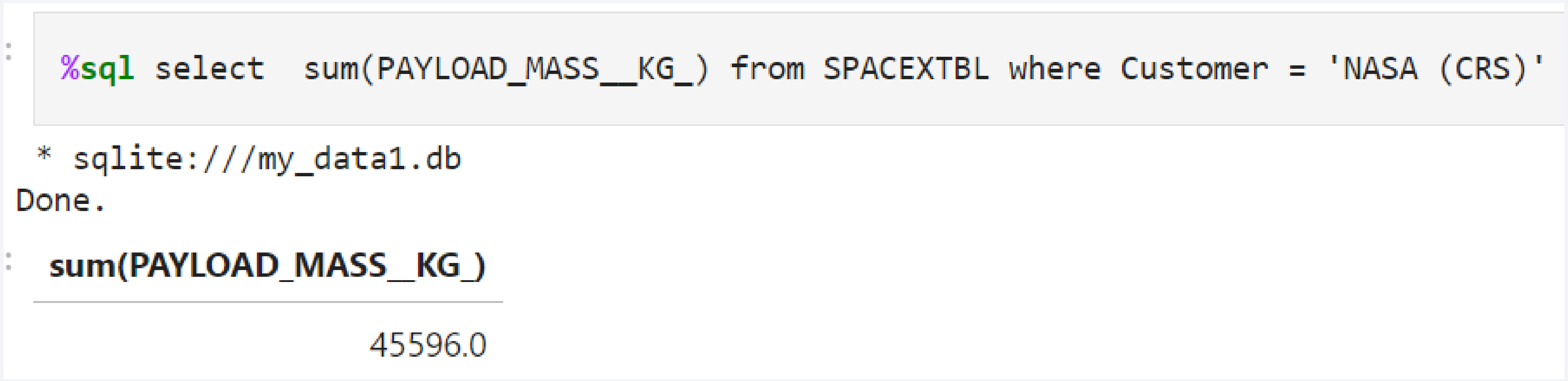
```
%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outc
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (paracl
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (paracl
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No atte
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No atte
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No atte

Total Payload Mass

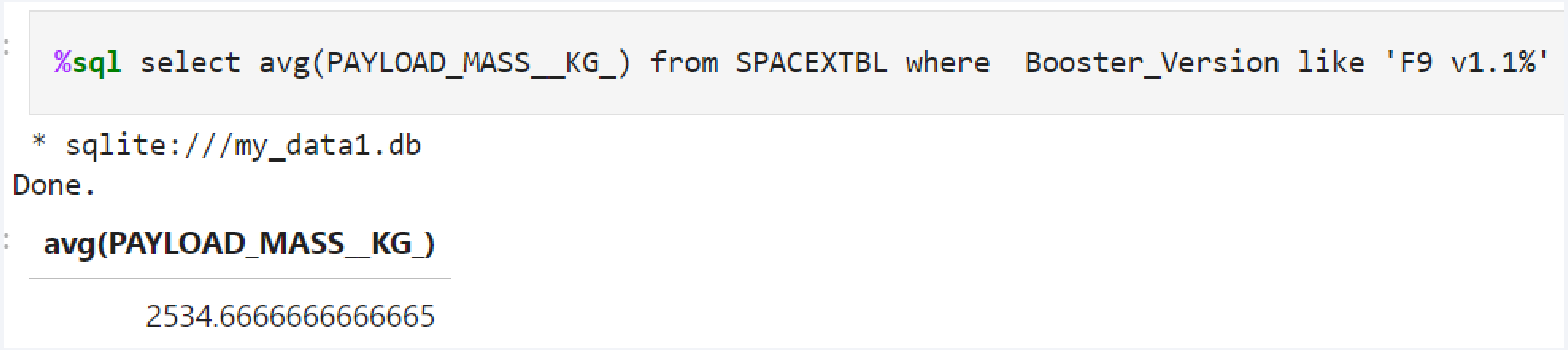
- The total Payload Mass carried by boosters from NASA was obtained by querying the sum() function and WHERE keyword.
- Total of 45,596 Kg.



```
%sql select  sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer = 'NASA (CRS)'

 * sqlite:///my_data1.db
Done.

sum(PAYLOAD_MASS__KG_)
               45596.0
```

Average Payload Mass by F9 v1.1

- The average Payload Mass carried by booster version F9 v1.1 was obtained using the avg() function and WHERE keyword.
- Average of 2534.6 Kg.



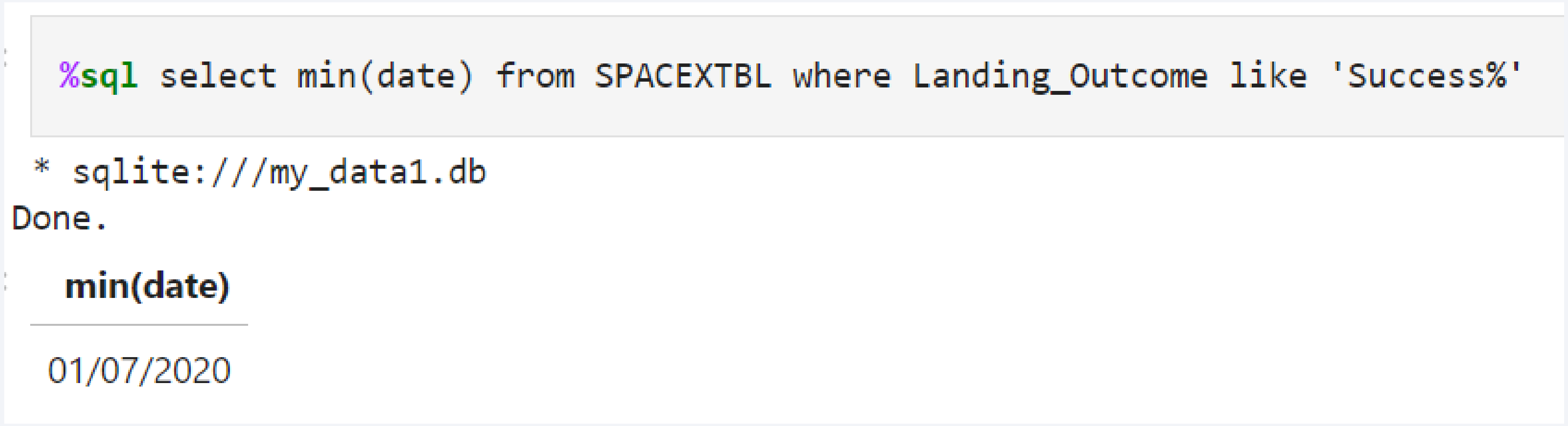
```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where  Booster_Version like 'F9 v1.1%'
```

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

avg(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

- A query with the min() function was used to find the date of the first successful landing outcome on ground pad. Also, the WHERE and LIKE keywords were used.

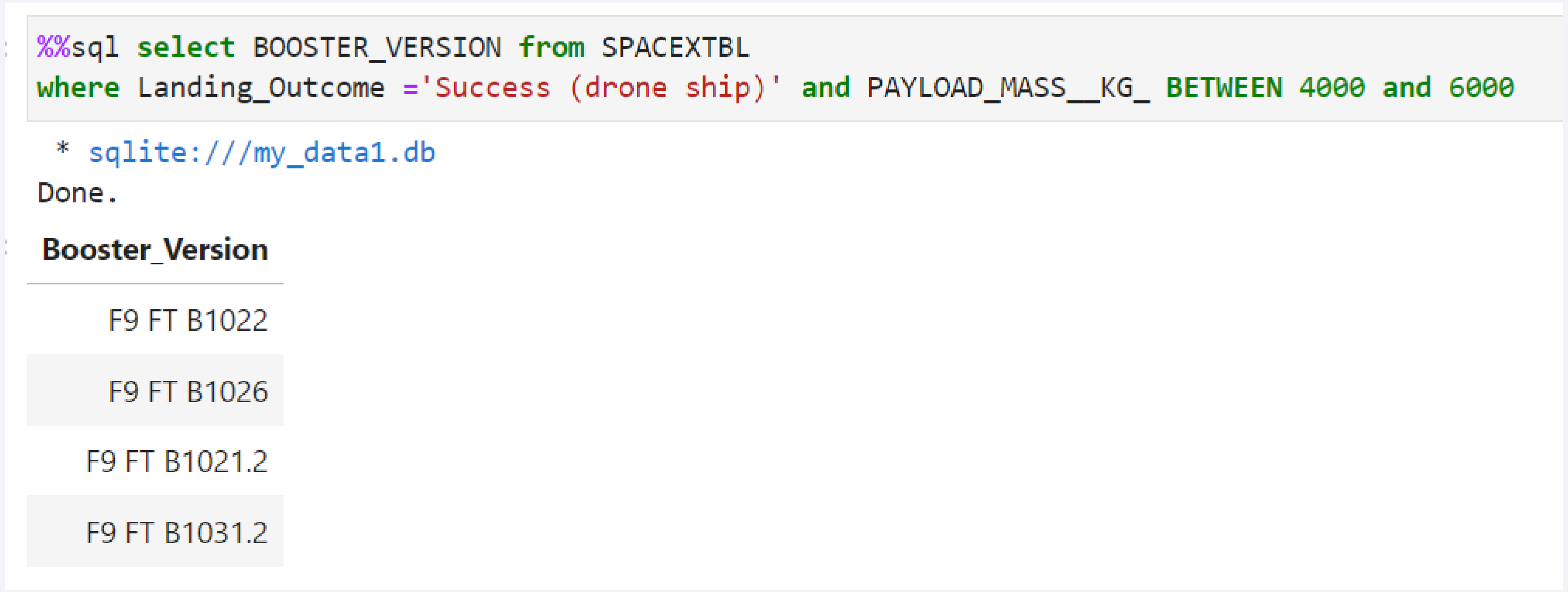


```
%sql select min(date) from SPACEXTBL where Landing_Outcome like 'Success%'

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

min(date)
01/07/2020

Successful Drone Ship Landing with Payload between 4000 and 6000

- The WHERE keyword is used to select the names of boosters which have successfully landed on drone ship.
- The BETWEEN and AND keyword were used in this query to filter Payload Mass greater than 4000 but less than 6000 kg.



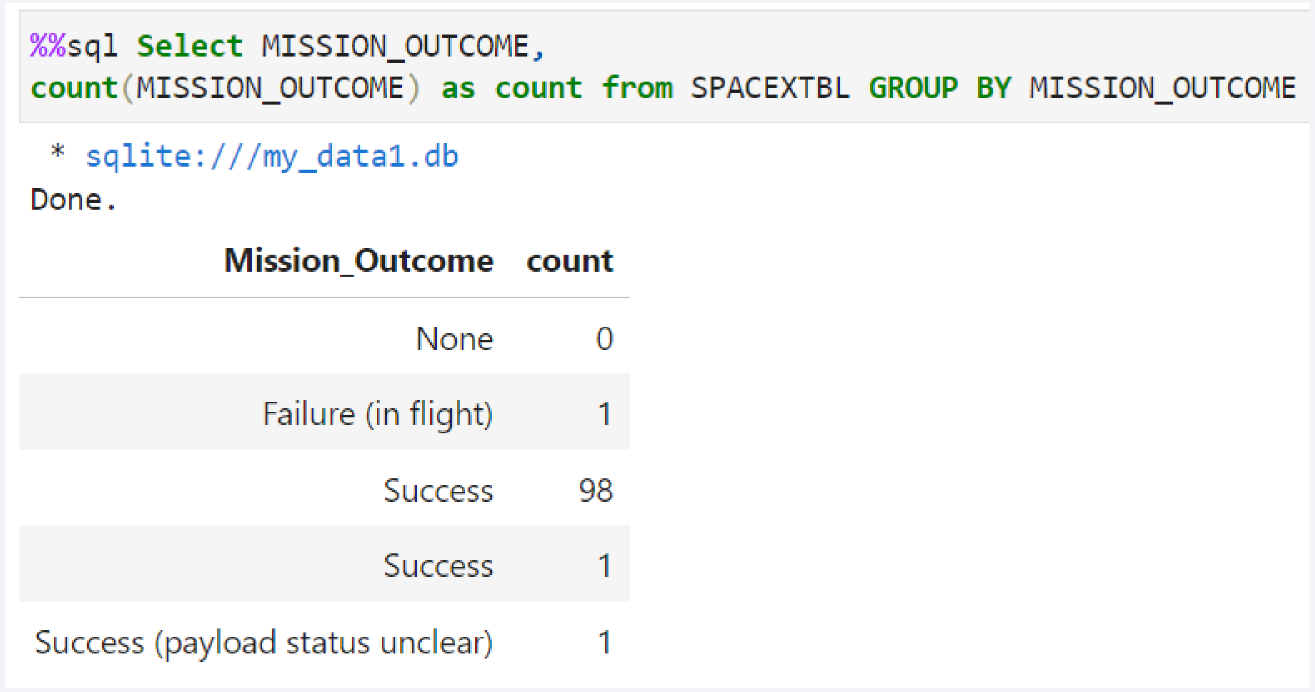
```
%%sql select BOOSTER_VERSION from SPACEXTBL
where Landing_Outcome ='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000
```

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

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

Total Number of Successful and Failure Mission Outcomes

- In this query the count() function is used to obtain the number of landings per Mission Outcome group.
- A total of 99 successful mission outcomes, 1 failed and 1 success with payload status unclear can be observed.



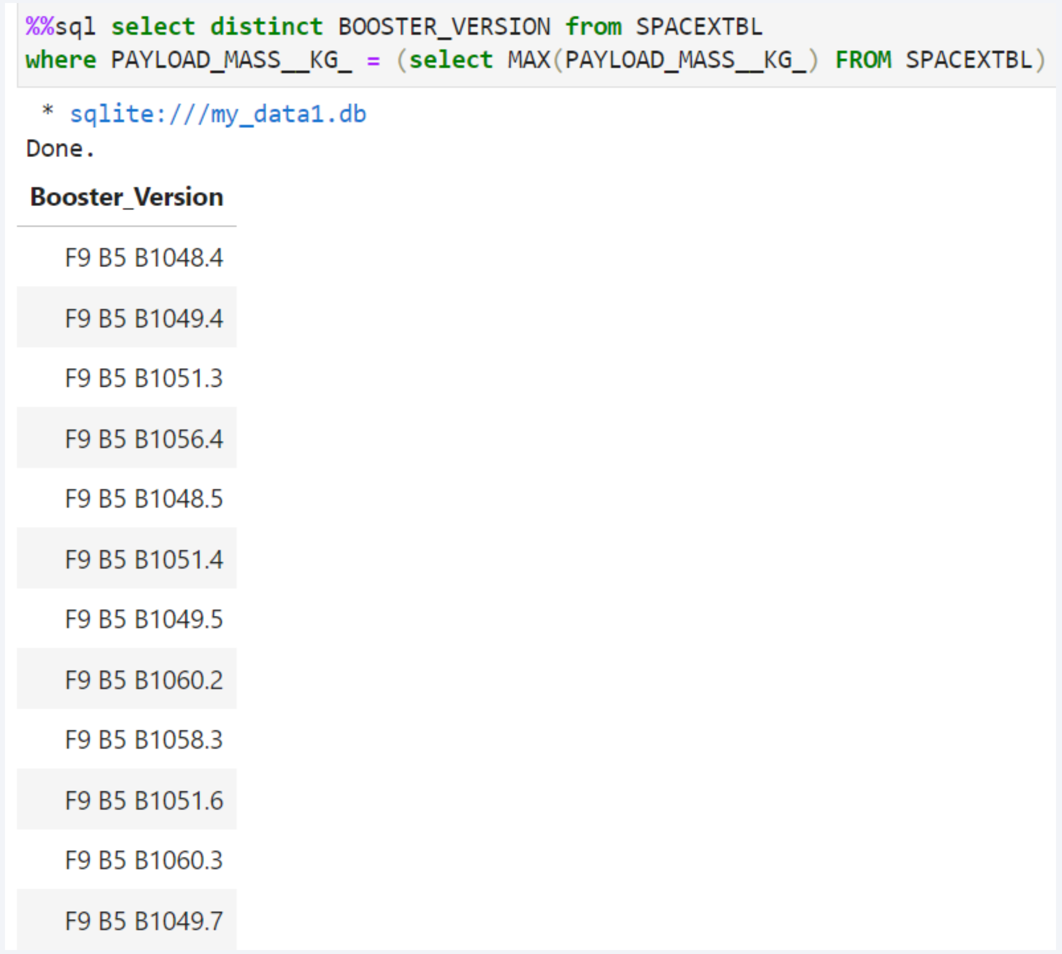
```
%%sql Select MISSION_OUTCOME,
count(MISSION_OUTCOME) as count from SPACEXTBL GROUP BY MISSION_OUTCOME
```

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

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

Boosters Carried Maximum Payload

- A subquery is used within a WHERE function in order to list the booster versions which have carried the maximum Payload Mass.



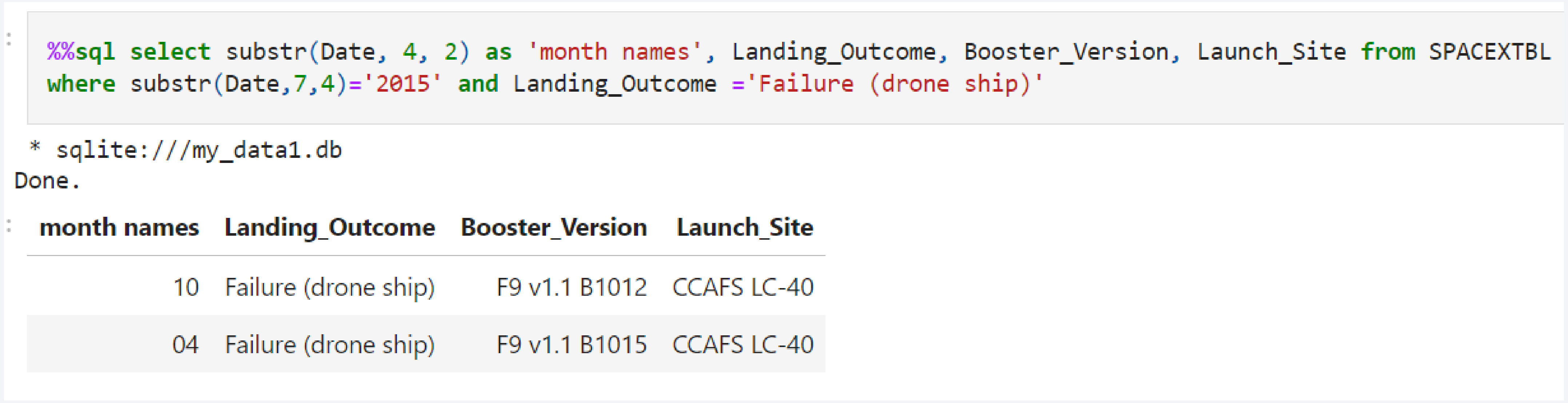
```
%%sql select distinct BOOSTER_VERSION from SPACEXTBL
where PAYLOAD_MASS__KG_ = (select MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

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

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 this query a substr() function is used to select a month and year from the Date column.
- A WHERE and AND keyword were used to obtain data from the year 2015 and were landing outcomes in drone ship failed.



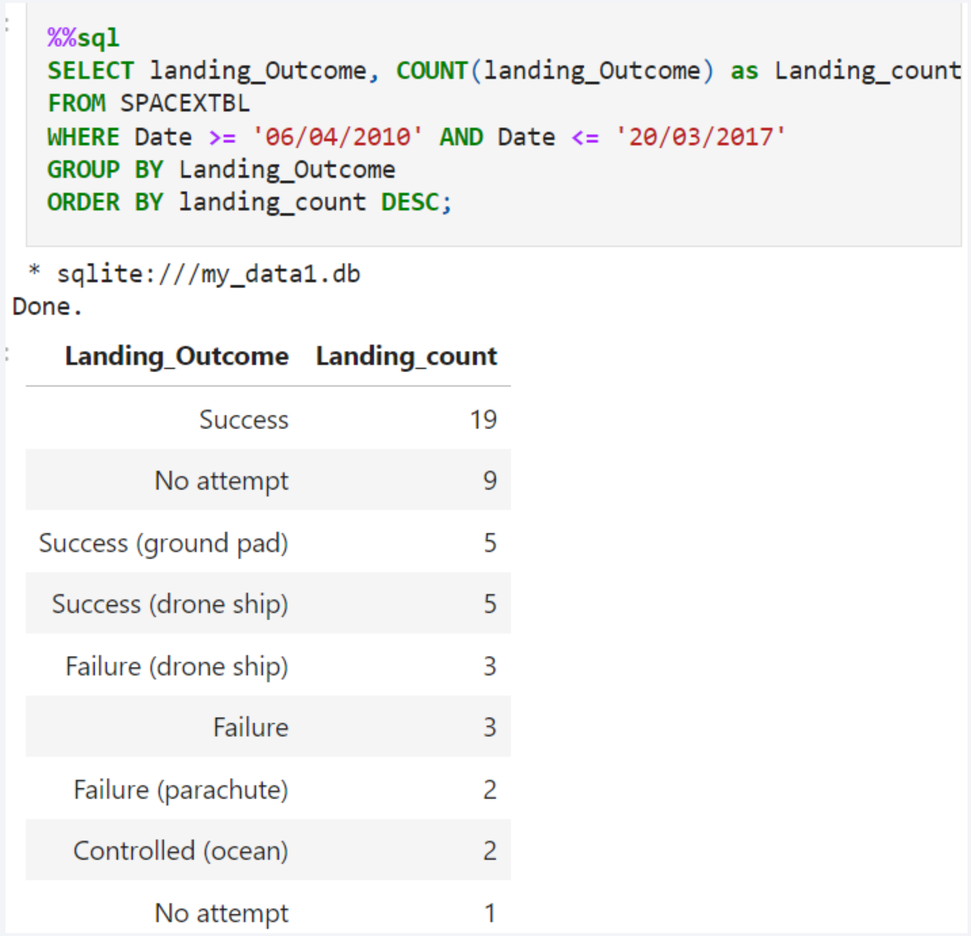
```
%%sql select substr(Date, 4, 2) as 'month names', Landing_Outcome, Booster_Version, Launch_Site from SPACEXTBL
where substr(Date,7,4)='2015' and Landing_Outcome ='Failure (drone ship)'
```

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

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

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

- In this query the count() function is used to count the landings by landing outcome.
- Dates are specified to be between 06/04/2010 and 20/03/2017 by using the WHERE and AND keywords.
- The GROUP BY, ORDER BY and DESC keywords are used to group the counts by landing outcome and then display the landing count by descending order.



```
%%sql
SELECT landing_Outcome, COUNT(landing_Outcome) as Landing_count
FROM SPACEXTBL
WHERE Date >= '06/04/2010' AND Date <= '20/03/2017'
GROUP BY Landing_Outcome
ORDER BY landing_count DESC;
```

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

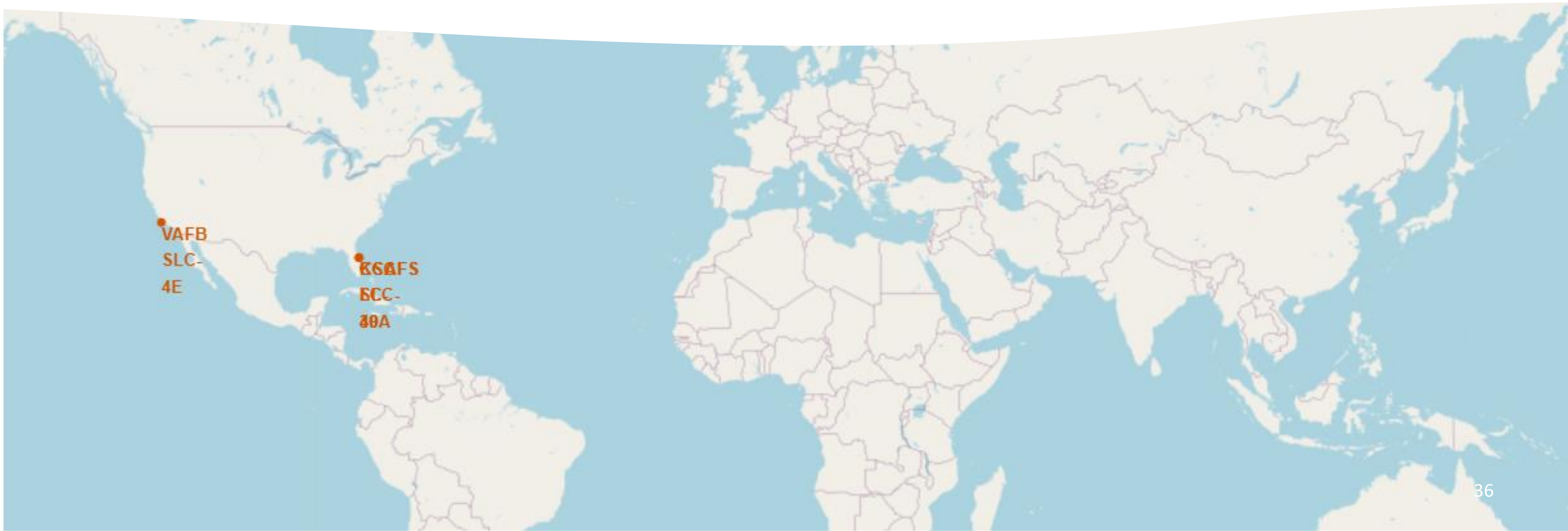
Landing_Outcome	Landing_count
Success	19
No attempt	9
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	3
Failure	3
Failure (parachute)	2
Controlled (ocean)	2
No attempt	1

Section 3

Launch Sites Proximities Analysis

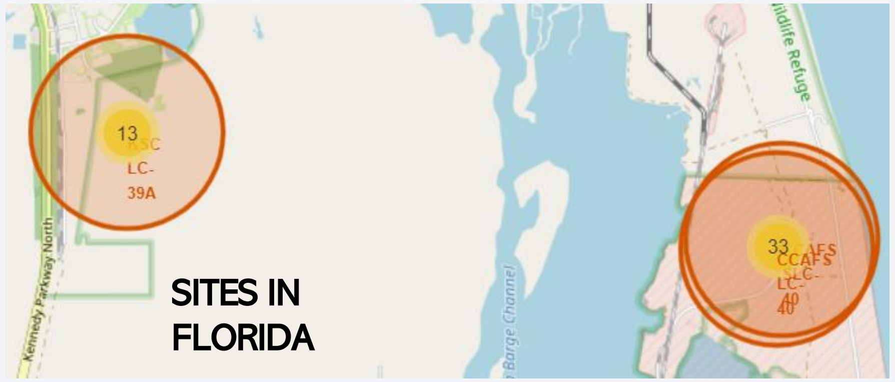
Launch Sites in Global Map

- It is noted that the launch sites for SpaceX are all in the USA, located in the coast of Florida and California.

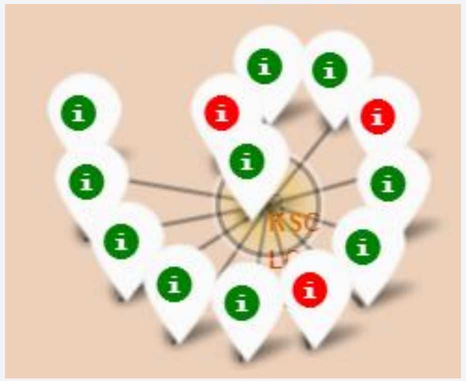


Success Record per Launch Site

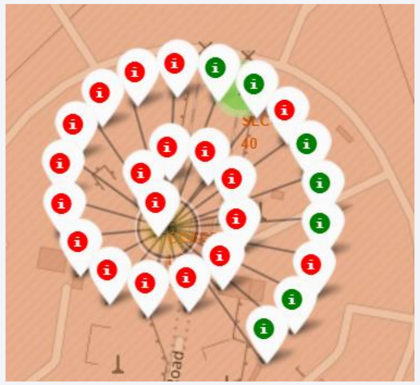
- For each launch site, green marks resemble successful launches and red marks resemble failed launches.



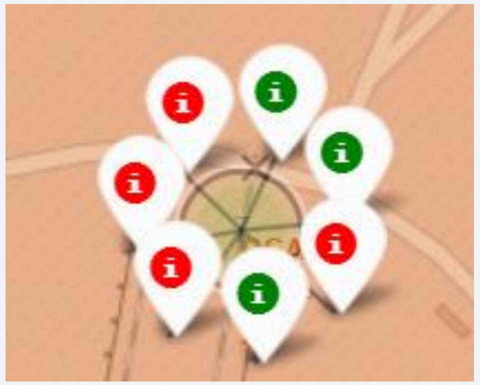
KSC LC -39A

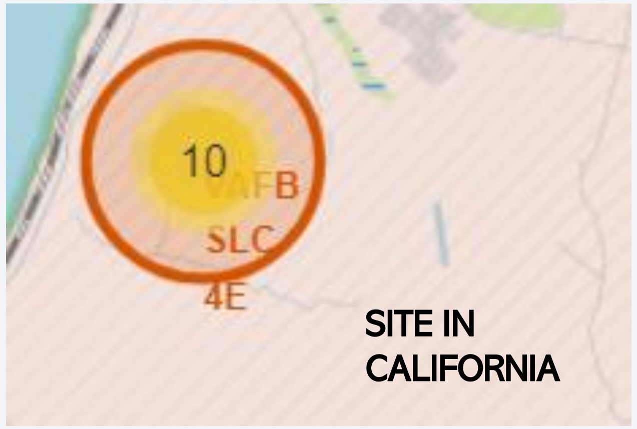


CCAFS LC-40

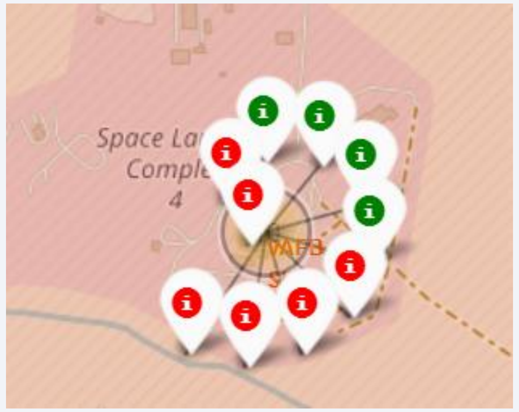


CCAFS SLC-40





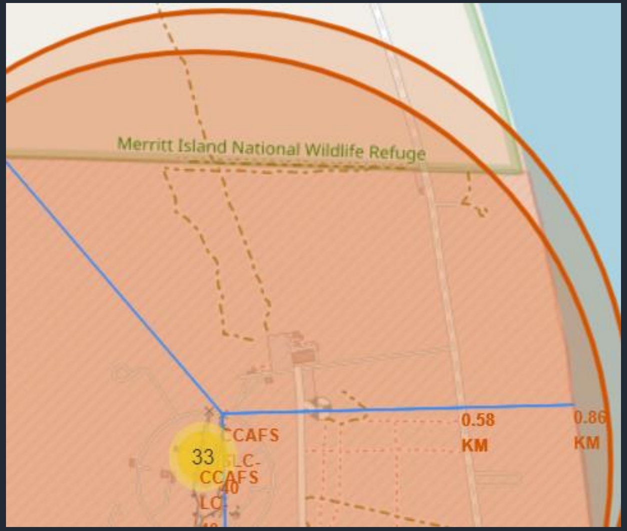
VAFB SLC-40



Distance from Launch Site to Coastline

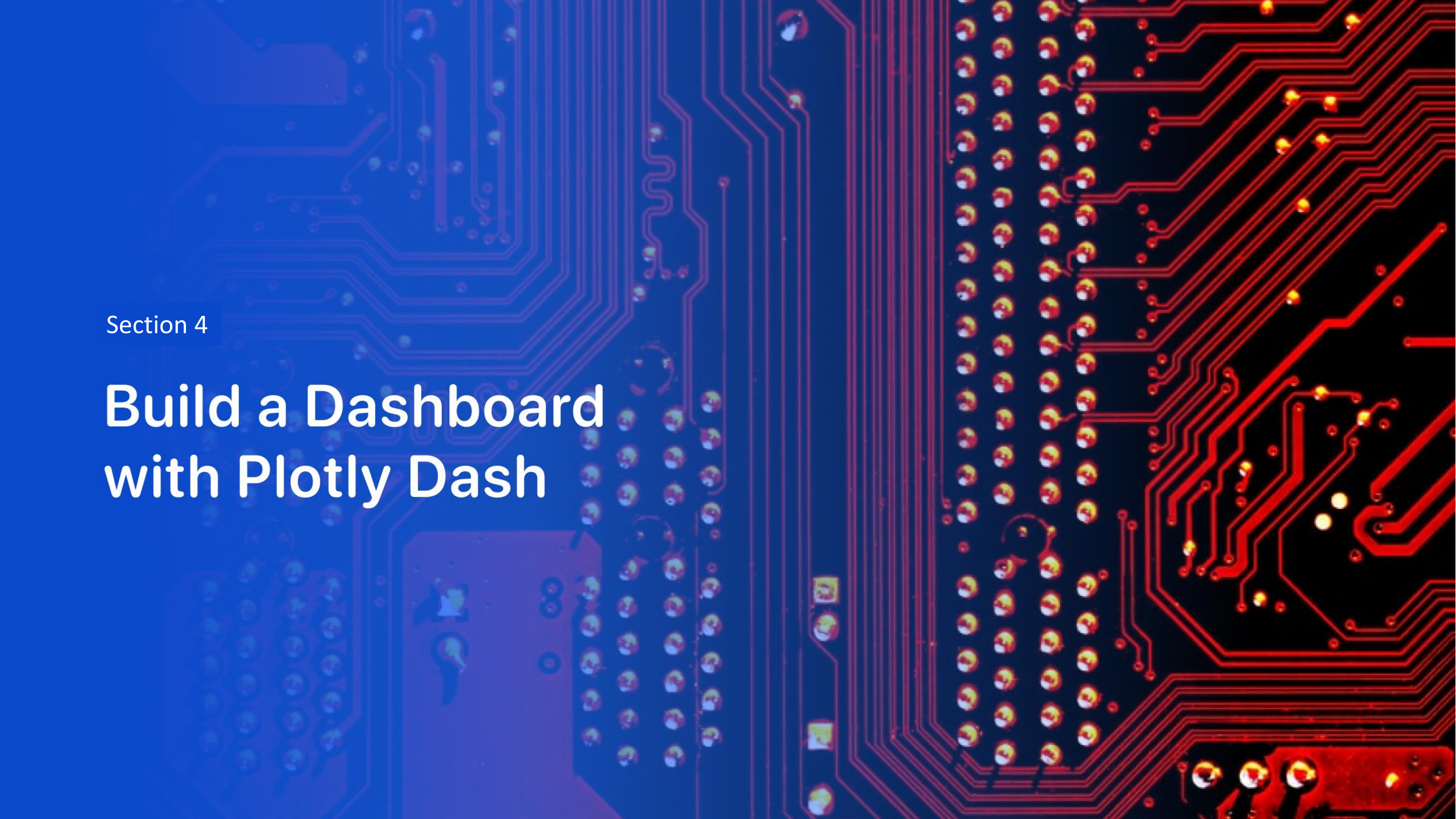
- The Florida site CCAFS SLC- 40 has the following distances:
  - 0.86 km to the nearest coastline
  - 0.58 km to the nearest highway
  - 1.27 km to the nearest Railway
  - 54.9 km to the nearest city (Melboune Beach)

46
40
Cape Canaveral Space Force Station
Canaveral



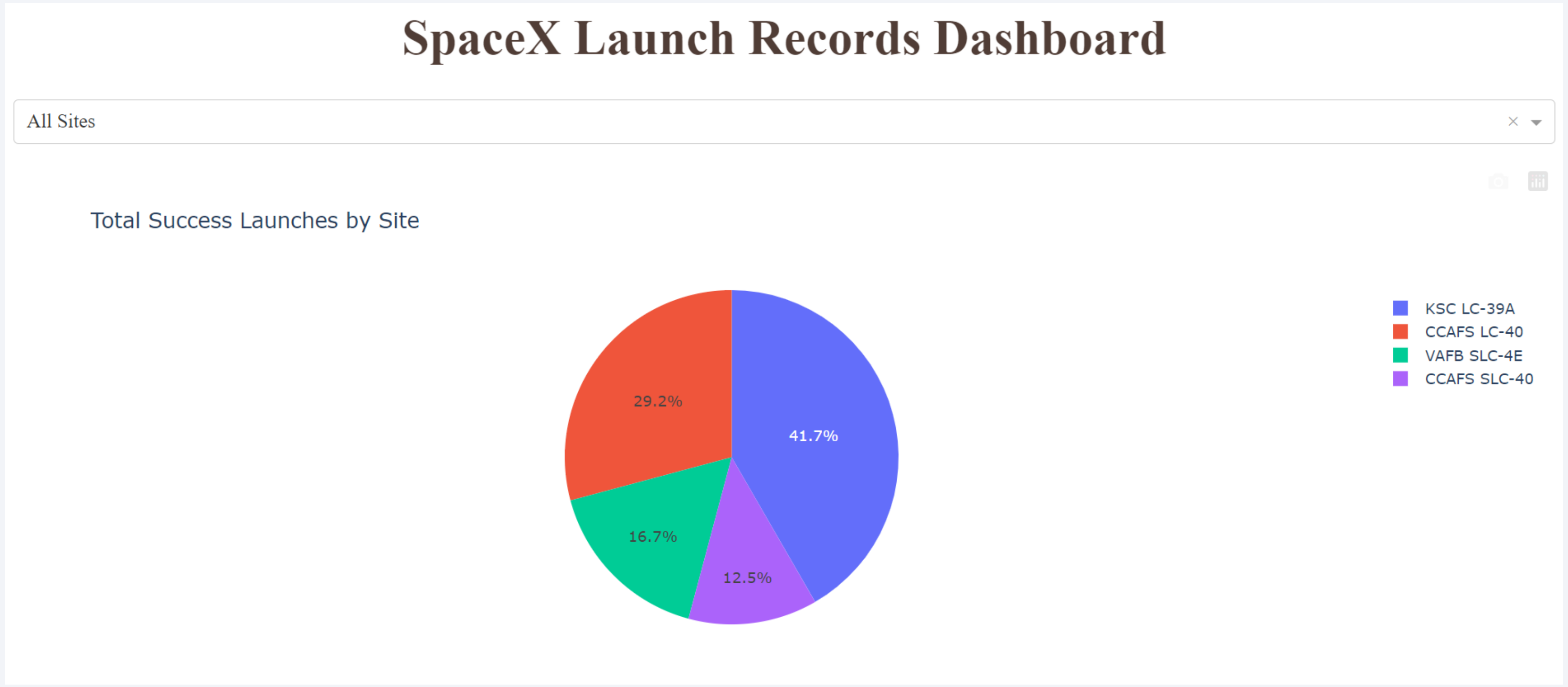
Section 4

Build a Dashboard with Plotly Dash



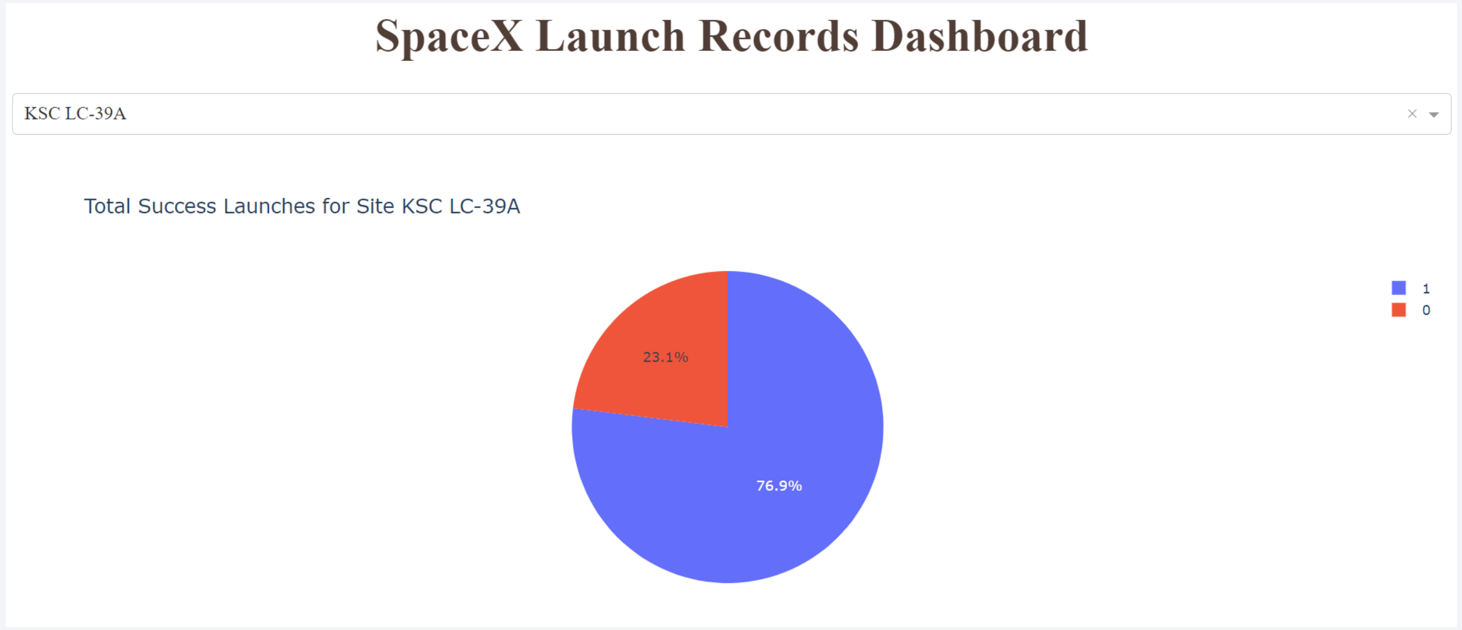
Total Success Launches for All Sites

- In the pie chart it can be observed how the launch site KSC LC-39A is acknowledged for having 41.7% of all the successful launches, being the launch site with the highest amount.



Launch Site With Highest Success Ratio

- The launch site KSC LC-39A has the highest success ratio in its launches, having a record of 76.9% successful launches.



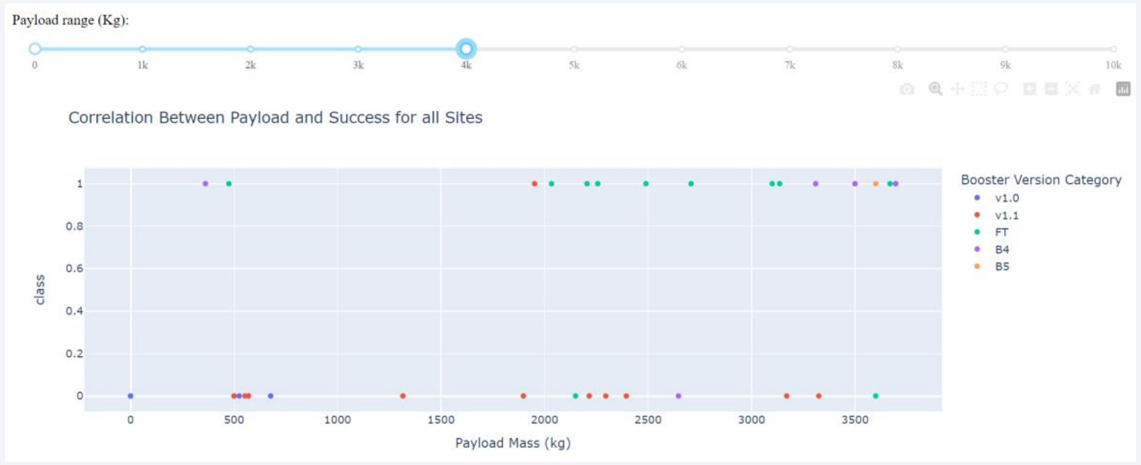
SpaceX Launch Records Dashboard

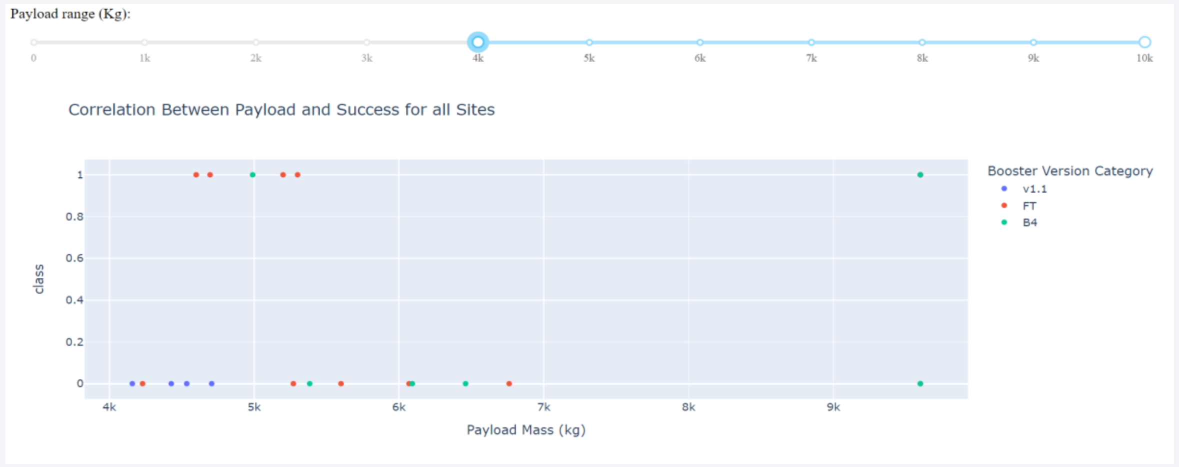
KSC LC-39A

Total Success Launches for Site KSC LC-39A

Correlation Between Payload and Success

- For a Payload Mass of 0 - 4,000 kg there is an apparent success rate in the landings of the rocket.
- For a Payload Mass of 4,000 – 10,000 kg there is decrease in the number of successful landings of the rocket.



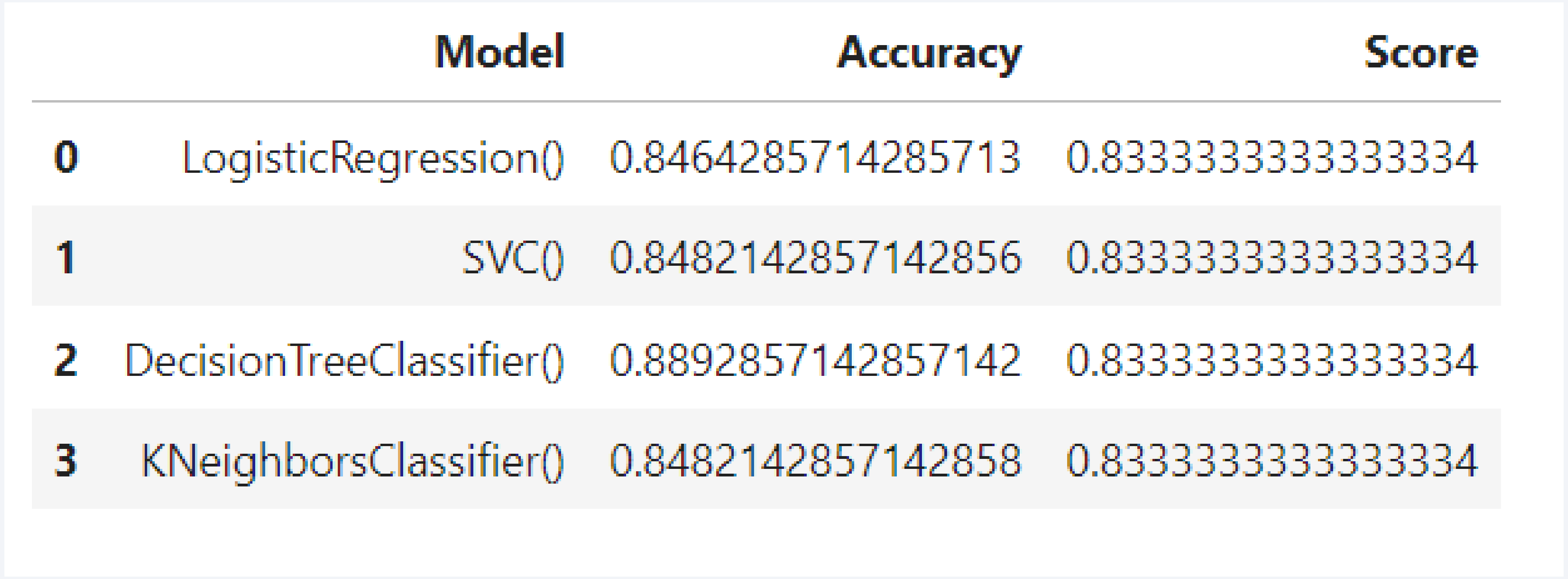


Section 5

Predictive Analysis (Classification)

Classification Accuracy

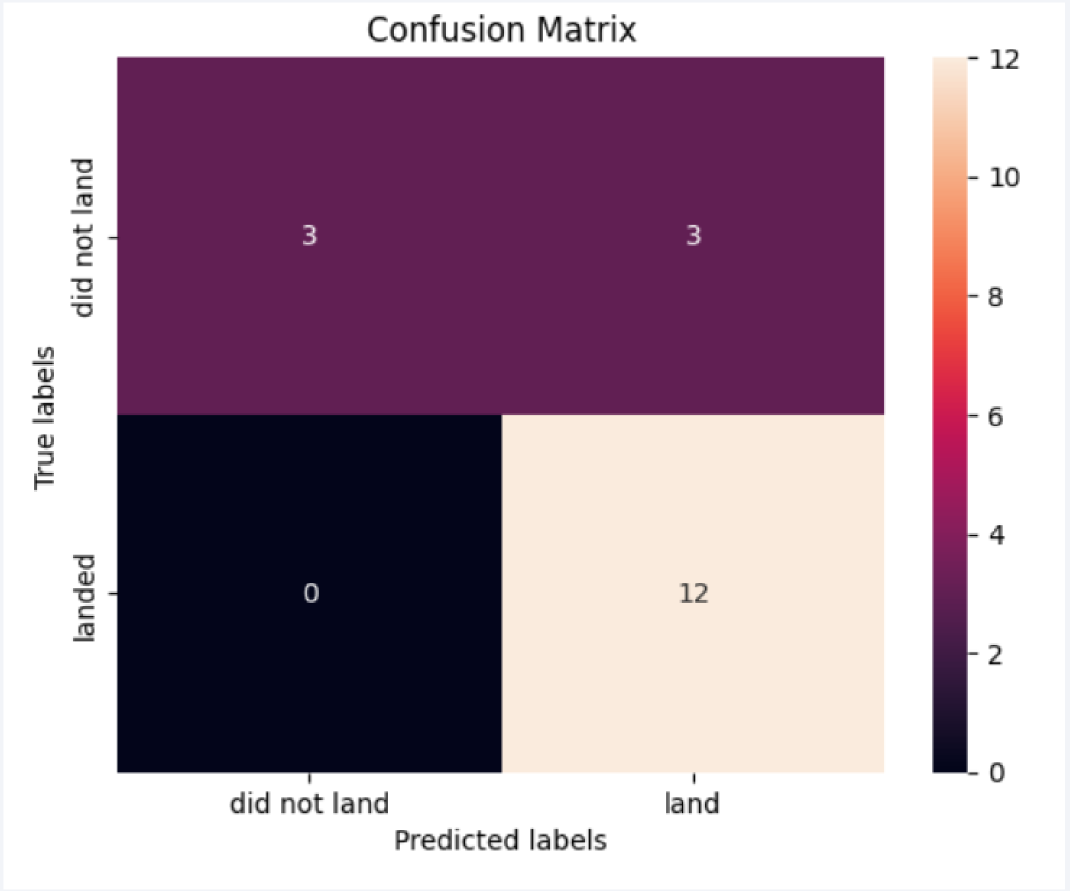
- All 4 models have a very similar accuracy classification



	Model	Accuracy	Score
0	LogisticRegression()	0.8464285714285713	0.8333333333333334
1	SVC()	0.8482142857142856	0.8333333333333334
2	DecisionTreeClassifier()	0.8892857142857142	0.8333333333333334
3	KNeighborsClassifier()	0.8482142857142858	0.8333333333333334

Confusion Matrix

- For all models with a very similar accuracy classifier the Confusion Matrix can be observed in the following way.
- 12 predicted landings that didn't land are shown.



Conclusions

- The success rate of the launches has increased through time.
- As the flight number increases, the first stage is more likely to land successfully.
- For a Payload Mass of 4,000 kg or less, there is a higher number of successful landings.
- ES-L1, GEO, HEO, and SEO are the orbits with the highest success rate.
- All 4 models have a very similar accuracy classification for predicting a successful landing.
- The site with the greatest success rate in landings is KSC LC -39A.

Thank you!