IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Hui Zhang

08/14/2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection via API, Web Scraping
  - Exploratory Data Analysis (EDA) with Data Visualization
  - EDA with SQL
  - Interactive Map with Folium
  - Dashboards with Plotly Dash
  - Predictive Analysis with Machine learning algorithm
- Summary of all results
  - Exploratory Data Analysis results
  - Interactive maps and dashboard
  - Predictive results

Introduction

- Project background and context
  - In this capstone, we will predict if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward 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. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.
- Problems you want to find answers
  - What are the main characteristics of a successful or failed landing ?
  - What are the effects of each relationship of the rocket variables on the success or failure of a landing ?
  - What are the conditions which will allow SpaceX to achieve the best landing success rate ?

Section 1

Methodology

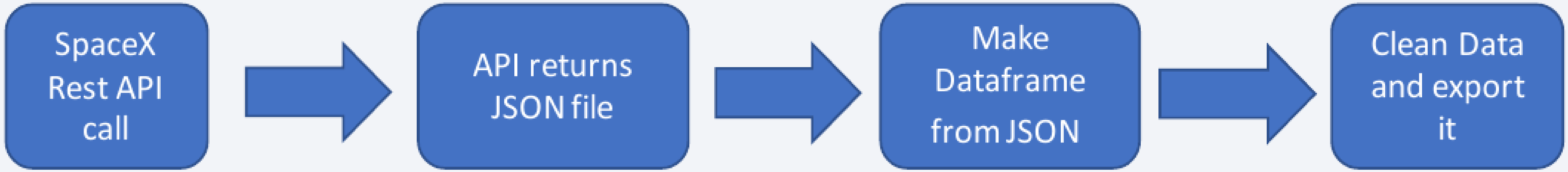
Methodology

Executive Summary

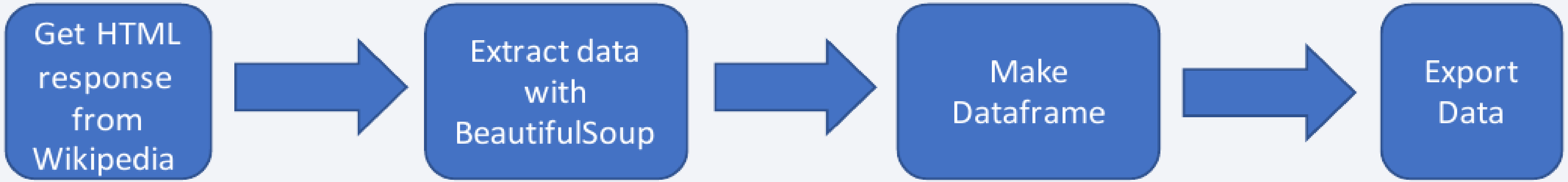
- Data collection methodology:
  - SpaceX REST API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Dropping unnecessary columns and handle NAN (I.e., feature engineering)
  - One Hot Encoding for classification models
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Using Grid search method in Scikit-learn on different models (k nearest Neighbours/Decision trees/Support vector machine/Logistic regression) to find the best parameters for each model

Data Collection

- Describe how data sets were collected.
  - The information obtained by the API are rocket, launches, payload information.
  - The Space X REST API URL is api.spacexdata.com/v4/



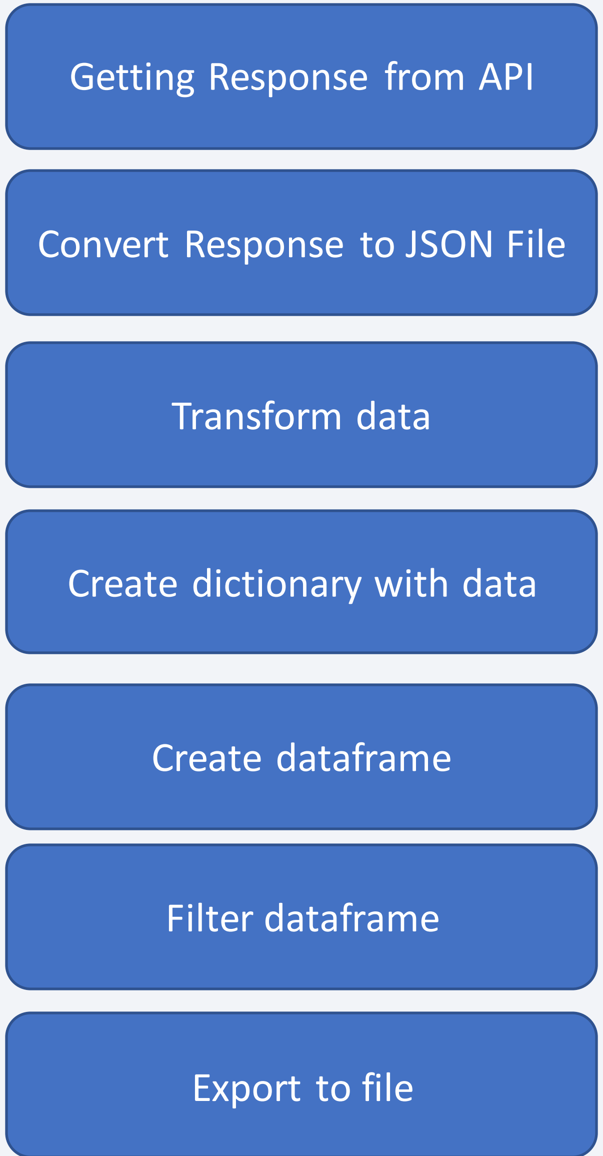
- The information obtained by the webscrapping of Wikipedia are launches, landing, payload information.
  - URL is https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922


Get HTML response from Wikipedia
Extract data with BeautifulSoup
Make Dataframe
Export Data

Data Collection – SpaceX API

- The flow chart for Data collection from SpaceX API is in the right:

- Link to the code:
  - Link_to_code



Getting Response from API

Convert Response to JSON File

Transform data

Create dictionary with data

Create dataframe

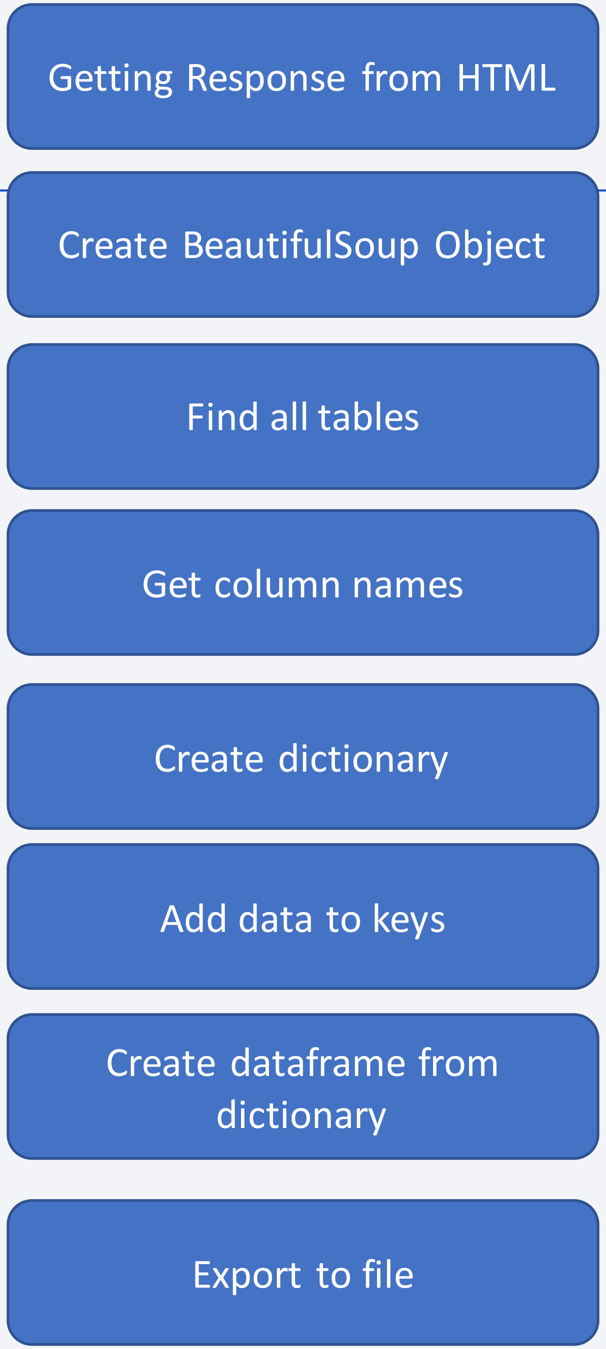
Filter dataframe

Export to file

Data Collection - Scraping

- The flow chart for Data collection of Scraping is in the right:

- Link to the code:
  - Link to code



Getting Response from HTML

Create BeautifulSoup Object

Find all tables

Get column names

Create dictionary

Add data to keys

Create dataframe from dictionary

Export to file

Data Wrangling

- In the dataset, there are several cases where the booster did not land successully.
  - True Ocean, True RTLS, True ASDS means the mission has been successful.
  - False Ocean, False RTLS, False ASDS means the mission was a failure.
- We need to transform string variables into categorical variables where 1 means the mission has been successful and 0 means the mission was a failure.

- [Link_to_code]

Calculate launches number for each site

Calculate the number and occurence of each orbit

Calculate number and occurrence of mission outcome per orbit type

Create landing outcome label from Outcome column

Export to file

EDA with Data Visualization

- Scatter Graphs
  - Flight Number vs. Payload Mass/Flight Number vs. Launch Site/Payload vs. Launch Site/Orbit vs. Flight Number/Payload vs. Orbit Type/Orbit vs. Payload Mass
  - Scatter plots show relationship between variables. This relationship is called the correlation.
- Bar Graph
  - Success rate vs. Orbit
  - Bar graphs show the relationship between numeric and categoric variables.
- Line Graph
  - Success rate vs. Year
  - Line graphs show data variables and their trends.
  - Line graphs can help to show global behavior and make prediction for unseen data.

Link_to_code

EDA with SQL

- We performed SQL queries to gather and understand data from dataset:
  - Displaying the names of the unique lauunch sites in the space mission.
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS).
  - Display average payload mass carried by booster version F9 v1.1.
  - List the date when the first successful landing outcome in ground pad was achieved.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
  - List the total number of successful and failure mission outcomes.
  - List the names of the booster_versions which have carried the maximum payload mass.
  - List the records which will display the month names, faiilure landing_ouutcomes in drone ship, booster versions, launch_site for the months in year 2015.
  - Rank the count of successful landiing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.
- Link_to_code

Build an Interactive Map with Folium

- Folium map object is a map centered on NASA Johnson Space Center at Houson, Texas
  - Red circle at NASA Johnson Space Center's coordinate with label showing its name (folium.Circle, folium.map.Marker).
  - Red circles at each launch site coordinates with label showing launch site name (folium.Circle, folium.map.Marker, folium.features.DivIcon).
  - The grouping of points in a cluster to display multiple and different information for the same coordinates (folium.plugins.MarkerCluster).
  - Markers to show successful and unsuccessful landings. Green for successful landing and Red for unsuccessful landing. (folium.map.Marker, folium.Icon).
  - Markers to show distance between launch site to key locations (railway, highway, coastway, city) and plot a line between them .(folium.map.Marker, folium.PolyLine, folium.features.DivIcon)

- These objects are created in order to understand better the problem and the data. We can show easily all launch sites, their surroundings and the number of successful and unsuccessful landings.

Link_to_code

Build a Dashboard with Plotly Dash

- Dashboard has dropdown, pie chart, rangeslider and scatter plot components
  - Dropdown allows a user to choose the launch site or all launch sites (dash_core_components.Dropdown).
  - Pie chart shows the total success and the total failure for the launch site chosen with the dropdown component (plotly.express.pie).
  - Rangeslider allows a user to select a payload mass in a fixed range (dash_core_components.RangeSlider) .
  - Scatter chart shows the relationship between two variables, in particular Success vs Payload Mass (plotly.express.scatter) .

  - [Link_to_code]()

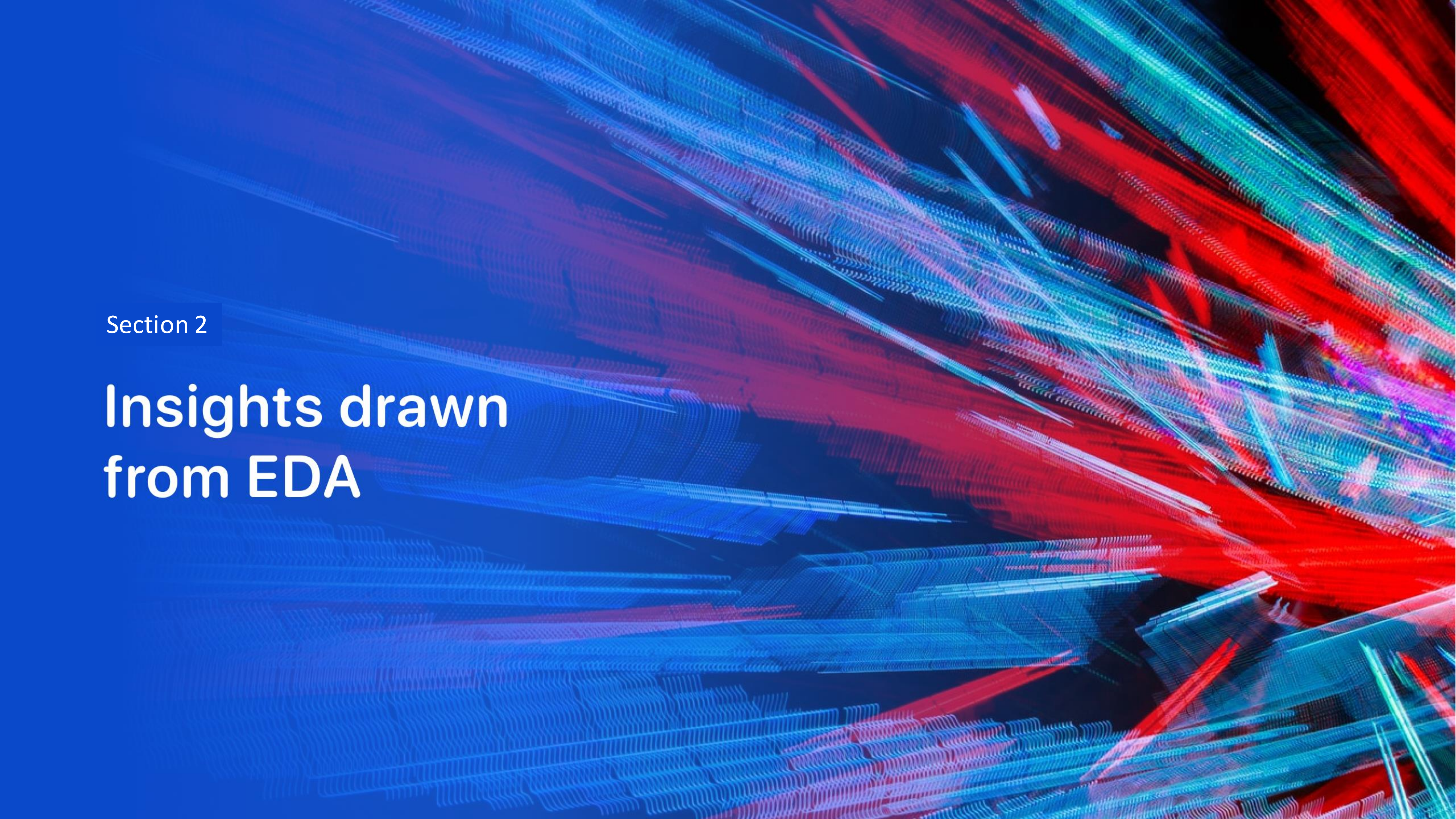
Predictive Analysis (Classification)

- Data preparation
  - Load dataset
  - Normalize data
  - Split data into training and test sets.
- Model preparation
  - Selection of machine learning algorithms
  - Set parameters for each algorithm to GridSearchCV
  - Training GridSearchModel models with training dataset
- Model evaluation
  - Get best hyperparameters for each type of model
  - Compute accuracy for each model with test dataset
  - Plot Confusion Matrix
- Model comparison
  - Comparison of models according to their accuracy
  - The model with the best accuracy will be chosen (see Notebook for result)

  - Link_to_code

Results

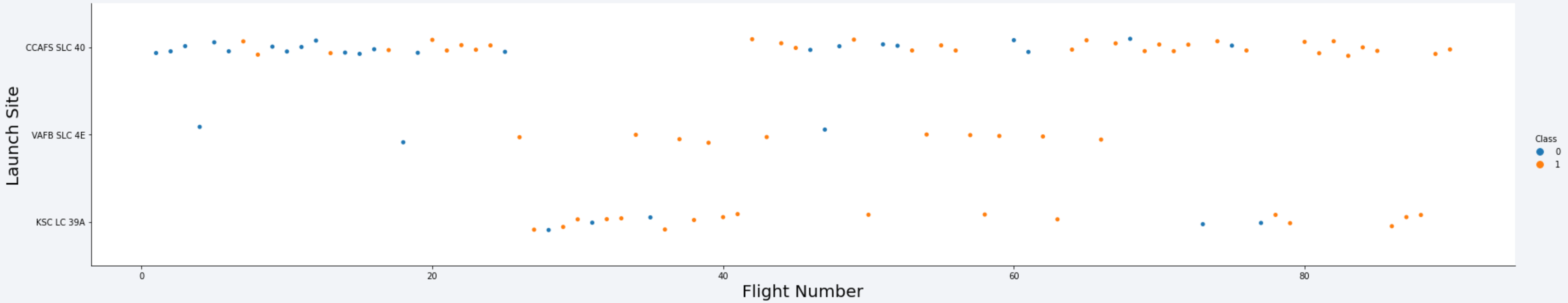
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results



Section 2

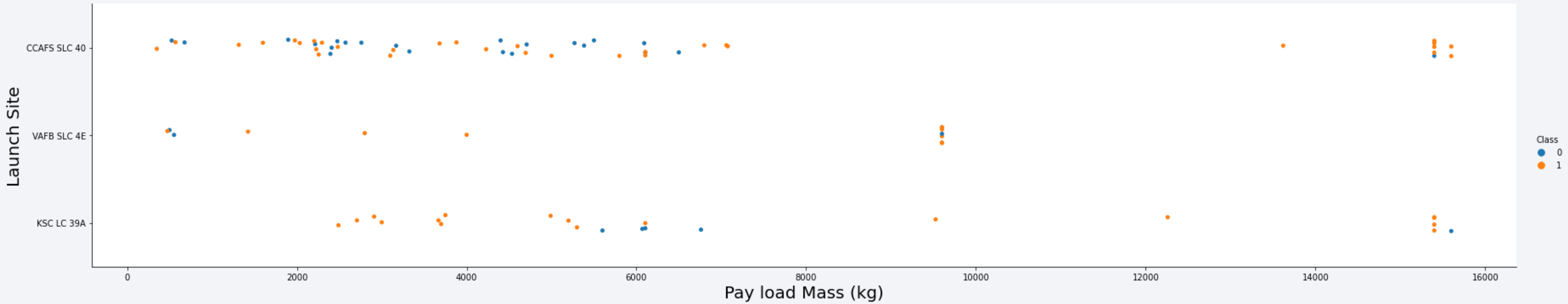
Insights drawn from EDA

Flight Number vs. Launch Site



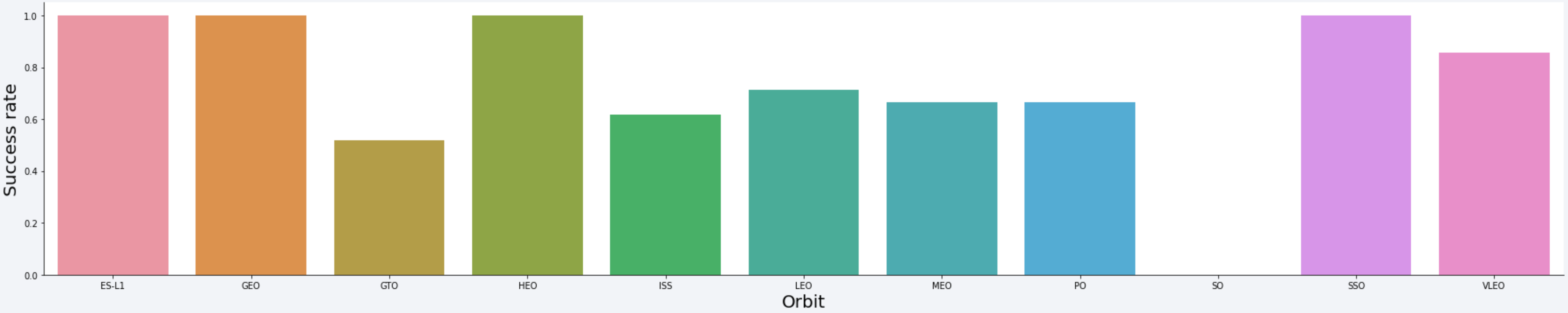
- For each site, the success rate is increasing.

Payload vs. Launch Site



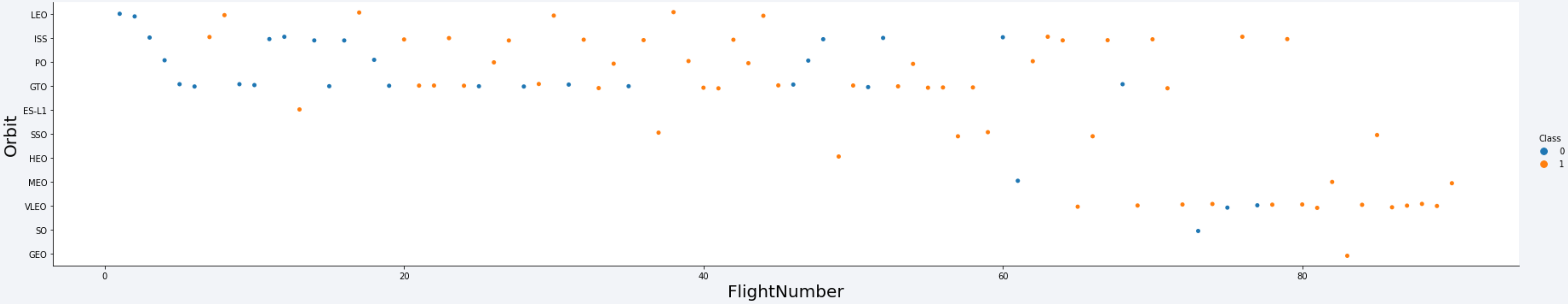
- Depending on the launch site, a heavier payload may be a consideration for a successful landing.
- On the other hand, a too heavy payload can make a landing fail.

Success Rate vs. Orbit Type



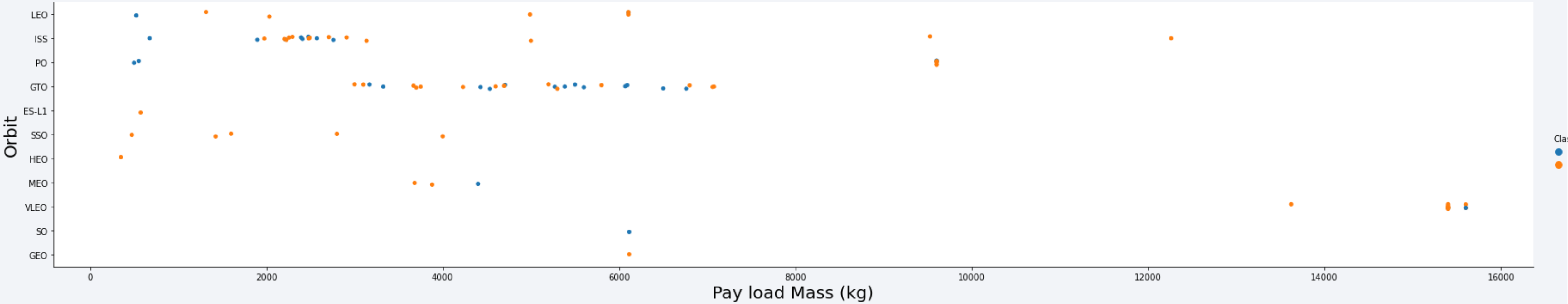
- success rate varies for different orbit types. We note that ES-L1, GEO, HEO, SSO have the best success rate.

Flight Number vs. Orbit Type



- the success rate increases with the number of flights for the LEO orbit. For some orbits like GTO, there is no relation between the success rate and the number of flights. But we can suppose that the
- high success rate of some orbits like SSO or HEO is due to the knowledge learned during former launches for other orbits.

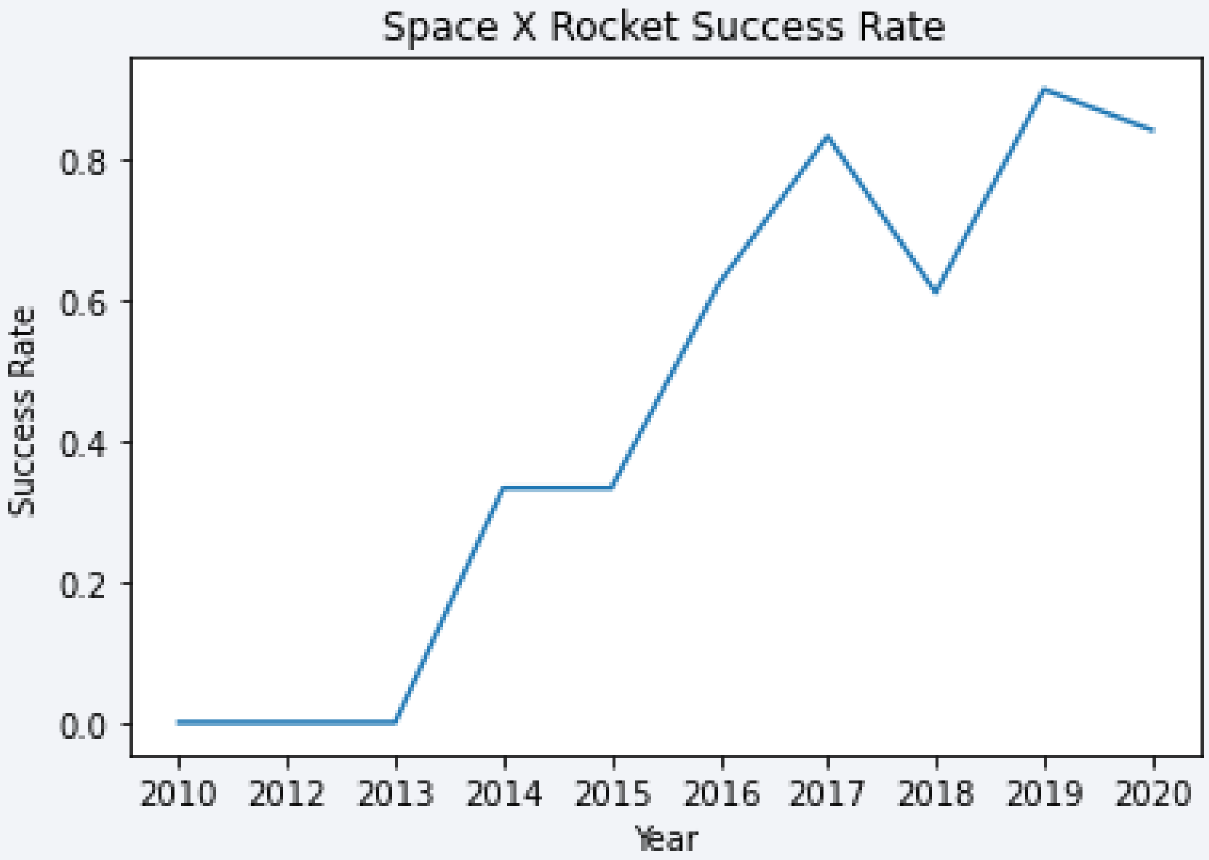
Payload vs. Orbit Type



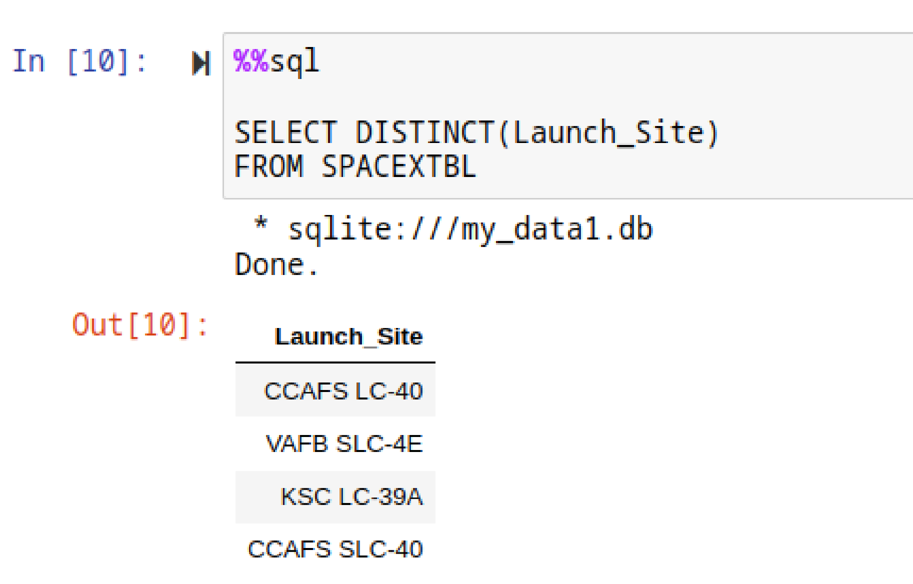
- The weight of the payloads can have a great influence on the success rate of the launches in certain orbits. For example, heavier payloads improve the success rate for the LEO orbit.
- Another finding is thatdecreasing the payload weight for a GTO orbit improves the success of a launch.

Launch Success Yearly Trend

- Since 2013, we can see an increase in the Space X Rocket success rate.



All Launch Site Names



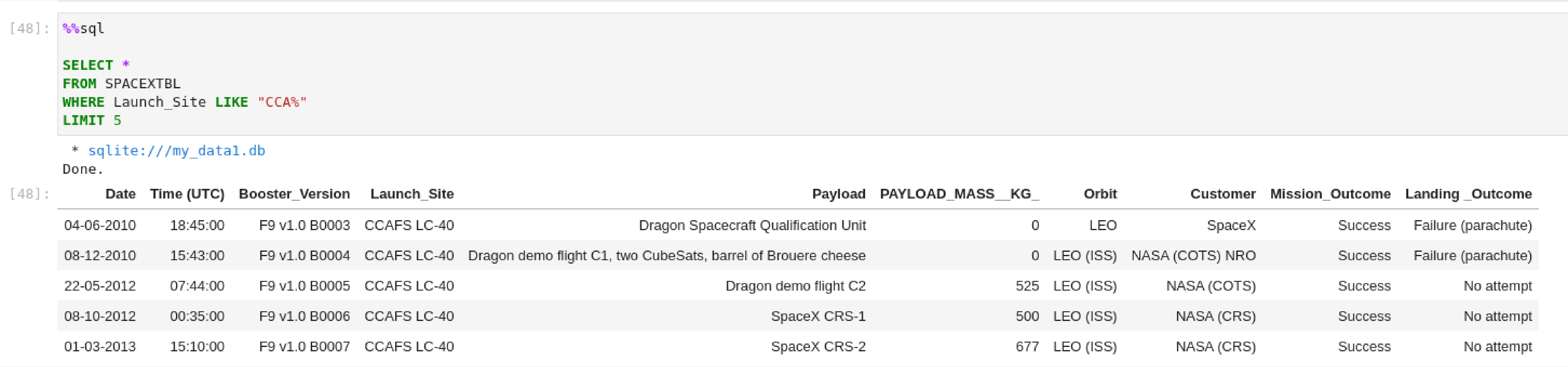
```
In [10]:  %%sql

          SELECT DISTINCT(Launch_Site)
          FROM SPACEXTBL

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

Out[10]:

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

- DISTINCT gives the unique output and remove the duplicates

Launch Site Names Begin with 'CCA'



```
[48]: %%sql

SELECT *
FROM SPACEXTBL
WHERE Launch_Site LIKE "CCA%"
LIMIT 5
```

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

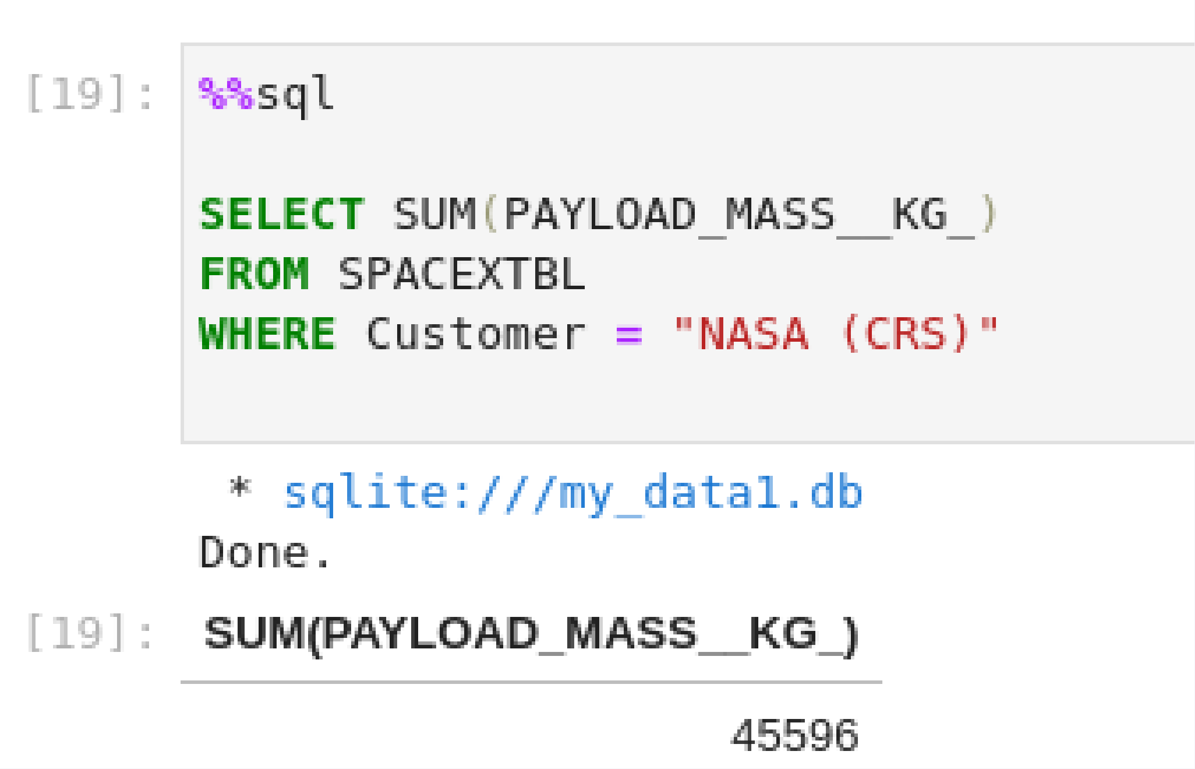
[48]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- The WHERE clause followed by LIKE clause filters launch sites that contain the substring CCA.
- LIMIT 5 shows 5 records from filtering.

Total Payload Mass

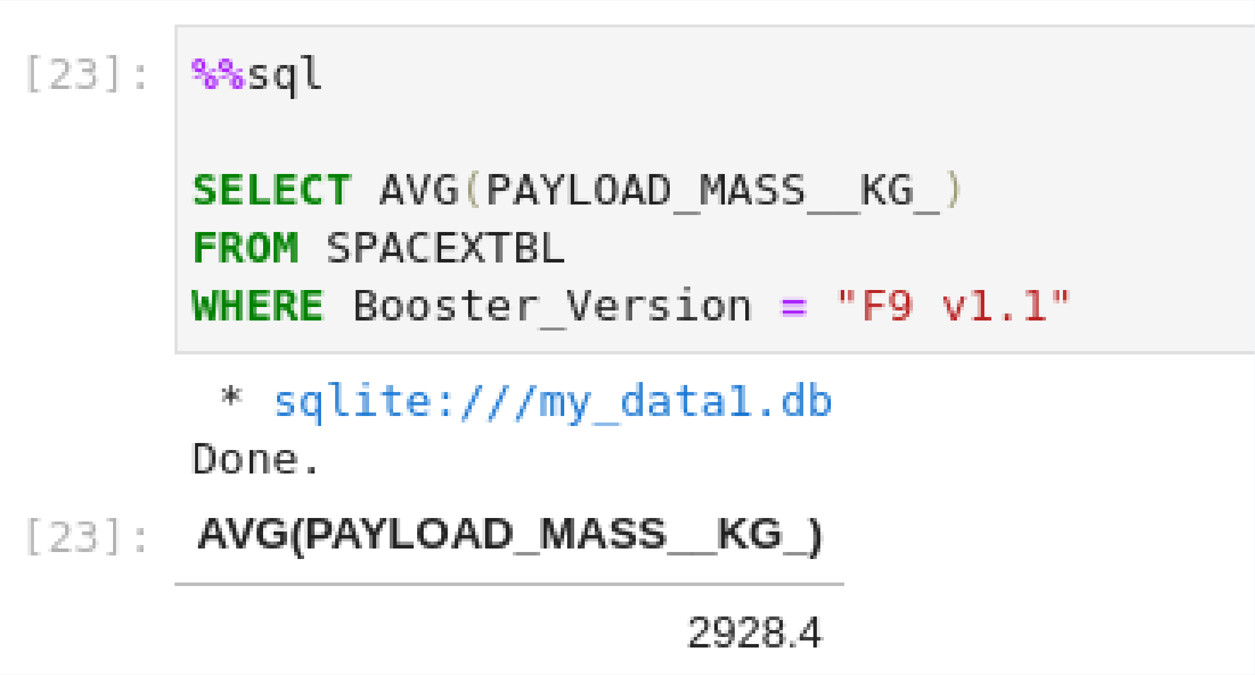
- This query returns the sum of all payload masses where the customer is NASA (CRS).



```
[19]: %%sql

      SELECT SUM(PAYLOAD_MASS__KG_)
      FROM SPACEXTBL
      WHERE Customer = "NASA (CRS)"

       * sqlite:///my_data1.db
      Done.
[19]: SUM(PAYLOAD_MASS__KG_)
                       45596
```

Average Payload Mass by F9 v1.1

- This query returns the average of all payload masses where the booster version is F9 v1.1



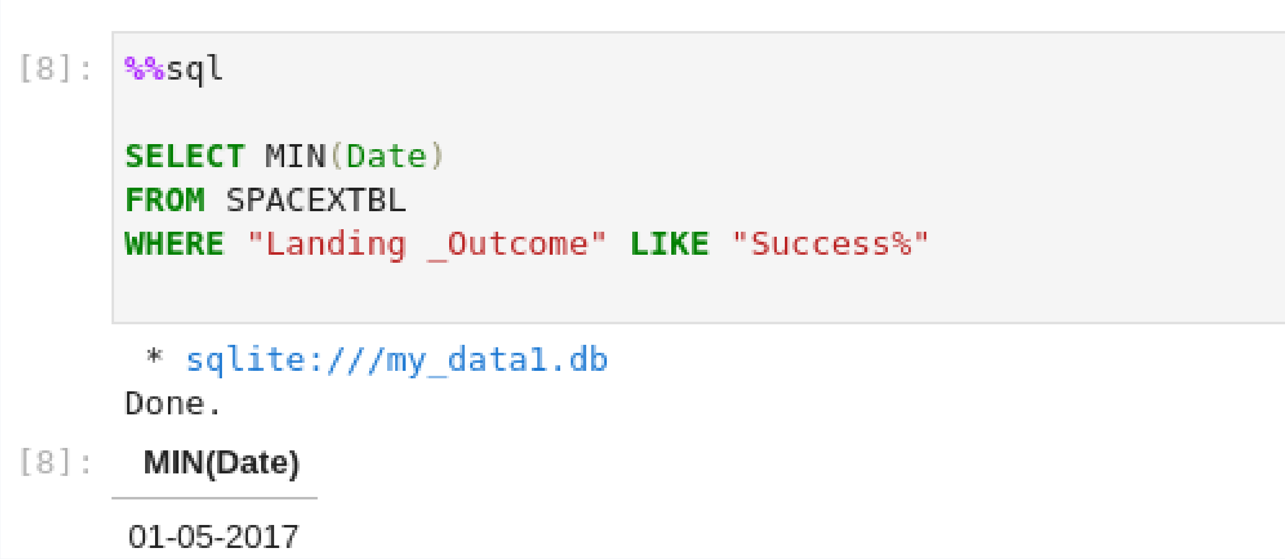
```
[23]: %%sql

      SELECT AVG(PAYLOAD_MASS__KG_)
      FROM SPACEXTBL
      WHERE Booster_Version = "F9 v1.1"

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

[23]:

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- The WHERE clause filters dataset in order to keep only records where landing was successful.
- With the MIN function, we select the record with the oldest date.



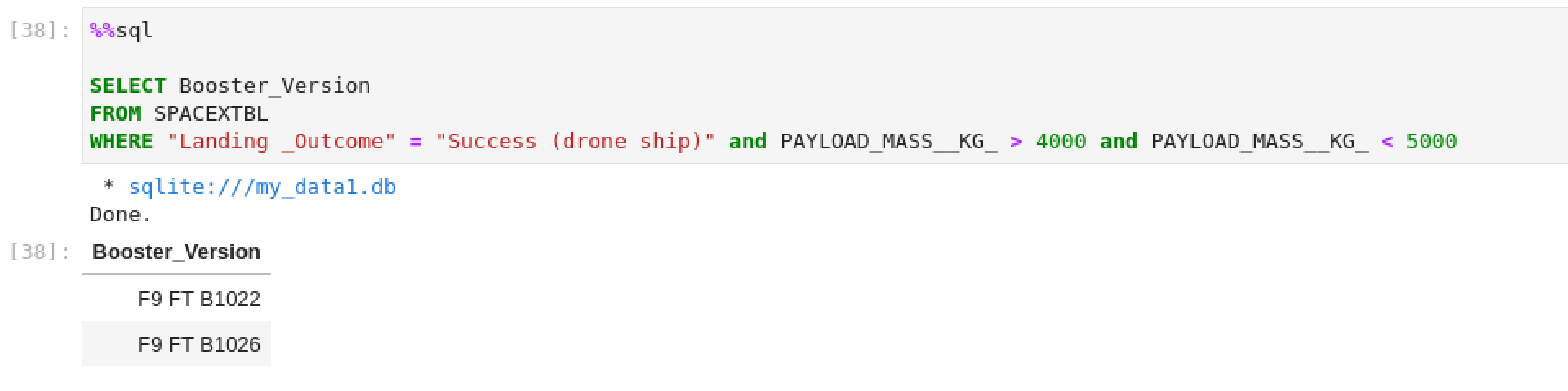
```
[8]: %%sql

     SELECT MIN(Date)
     FROM SPACEXTBL
     WHERE "Landing _Outcome" LIKE "Success%"
```

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

[8]:

MIN(Date)
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000



```
[38]: %%sql

      SELECT Booster_Version
      FROM SPACEXTBL
      WHERE "Landing _Outcome" = "Success (drone ship)" and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 5000
```

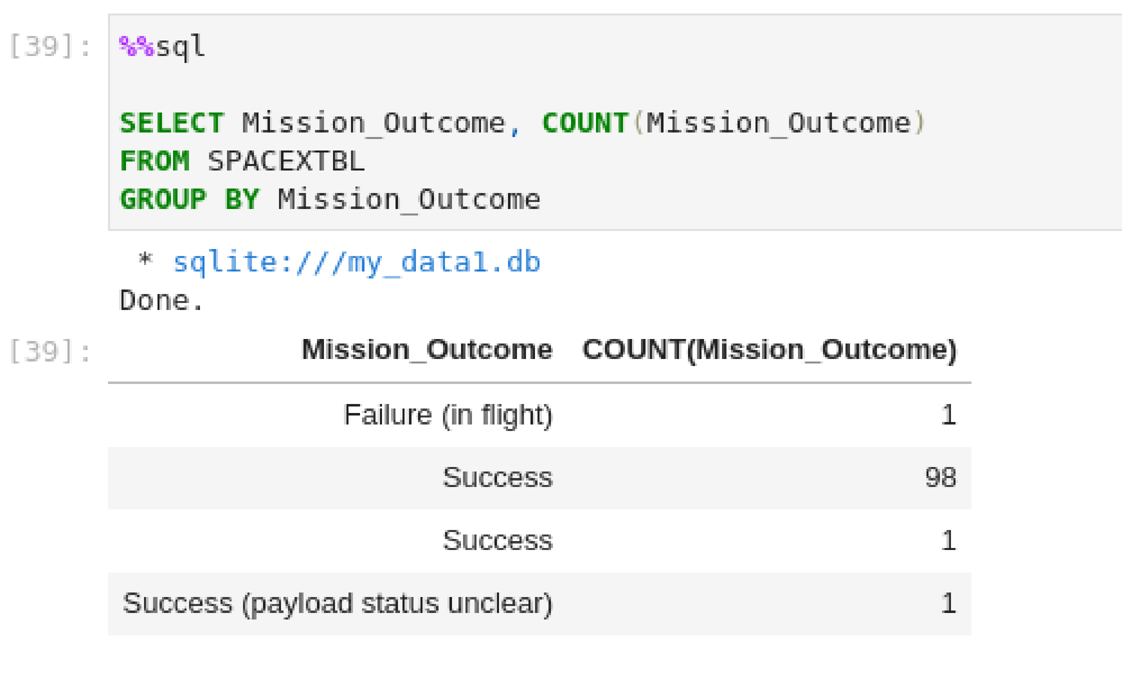
```
 * sqlite:///my_data1.db
Done.
```

[38]:

Booster_Version
F9 FT B1022
F9 FT B1026

- This query returns the booster version where landing was successful and payload mass is between 4000 and 6000 kg.
- The WHERE and AND clauses filter the dataset.

Total Number of Successful and Failure Mission Outcomes



```
%%sql

SELECT Mission_Outcome, COUNT(Mission_Outcome)
FROM SPACEXTBL
GROUP BY Mission_Outcome
```

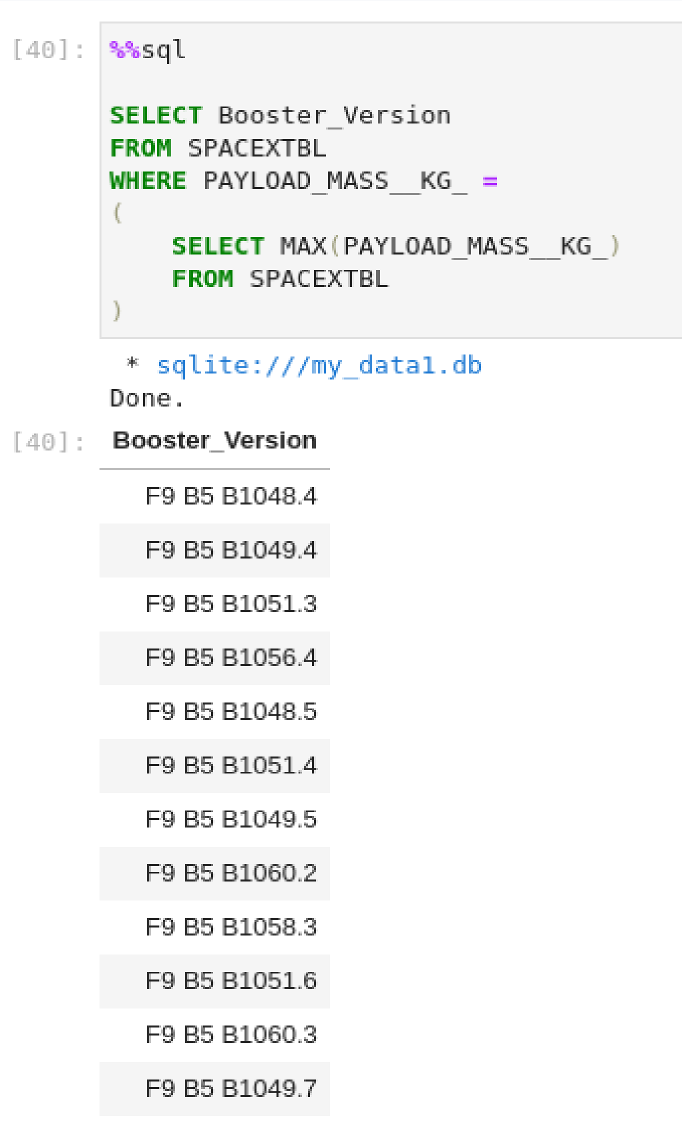
```
 * sqlite:///my_data1.db
Done.
```

Mission_Outcome	COUNT(Mission_Outcome)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

- Using GROUP BY and count to count the total number of successful/failure mission outputs

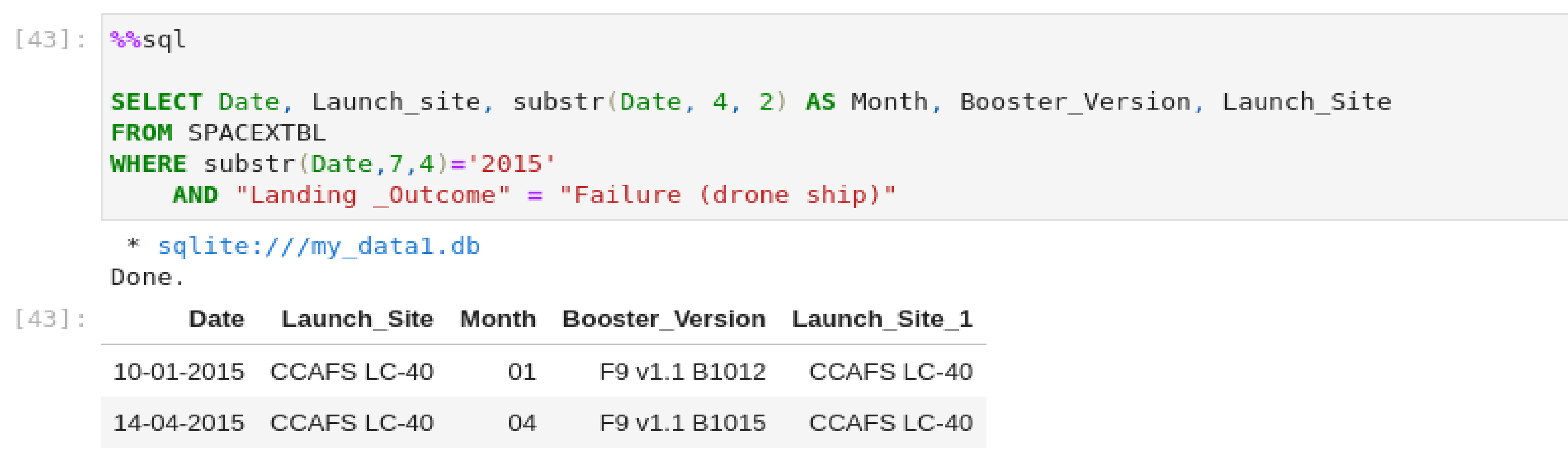
Boosters Carried Maximum Payload

- We used a subquery to filter data by returning only the heaviest payload mass with MAX function.
- The main query uses subquery results and returns unique booster version (SELECT DISTINCT) with the heaviest payload mass.



```
[40]: %%sql

      SELECT Booster_Version
      FROM SPACEXTBL
      WHERE PAYLOAD_MASS__KG_ =
      (
          SELECT MAX(PAYLOAD_MASS__KG_)
          FROM SPACEXTBL
      )

       * sqlite:///my_data1.db
      Done.
[40]: 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



```
[43]: %%sql

      SELECT Date, Launch_site, substr(Date, 4, 2) AS Month, Booster_Version, Launch_Site
      FROM SPACEXTBL
      WHERE substr(Date,7,4)='2015'
          AND "Landing _Outcome" = "Failure (drone ship)"

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

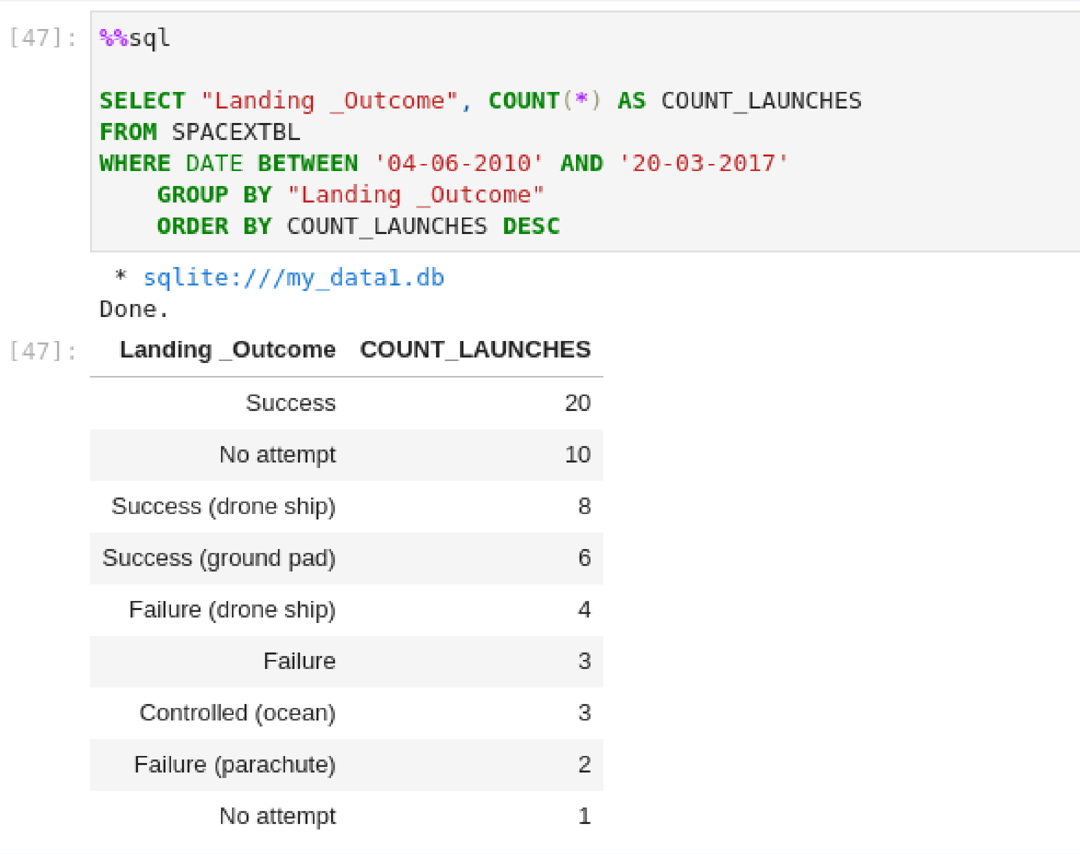
[43]:

Date	Launch_Site	Month	Booster_Version	Launch_Site_1
10-01-2015	CCAFS LC-40	01	F9 v1.1 B1012	CCAFS LC-40
14-04-2015	CCAFS LC-40	04	F9 v1.1 B1015	CCAFS LC-40

- This query returns month, booster version, launch site where landing was unsuccessful and landing date took place in 2015.
- Substr function process date in order to take month or year. Substr(DATE, 4, 2) shows month. Substr(DATE,7, 4) shows year.

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

- This query returns landing outcomes and their count where mission was successful and date is between 04/06/2010 and 20/03/2017.
- The GROUP BY clause groups results by landing outcome and ORDER BY COUNT DESC shows results in decreasing order.



```
[47]: %%sql

      SELECT "Landing _Outcome", COUNT(*) AS COUNT_LAUNCHES
      FROM SPACEXTBL
      WHERE DATE BETWEEN '04-06-2010' AND '20-03-2017'
          GROUP BY "Landing _Outcome"
          ORDER BY COUNT_LAUNCHES DESC
```

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

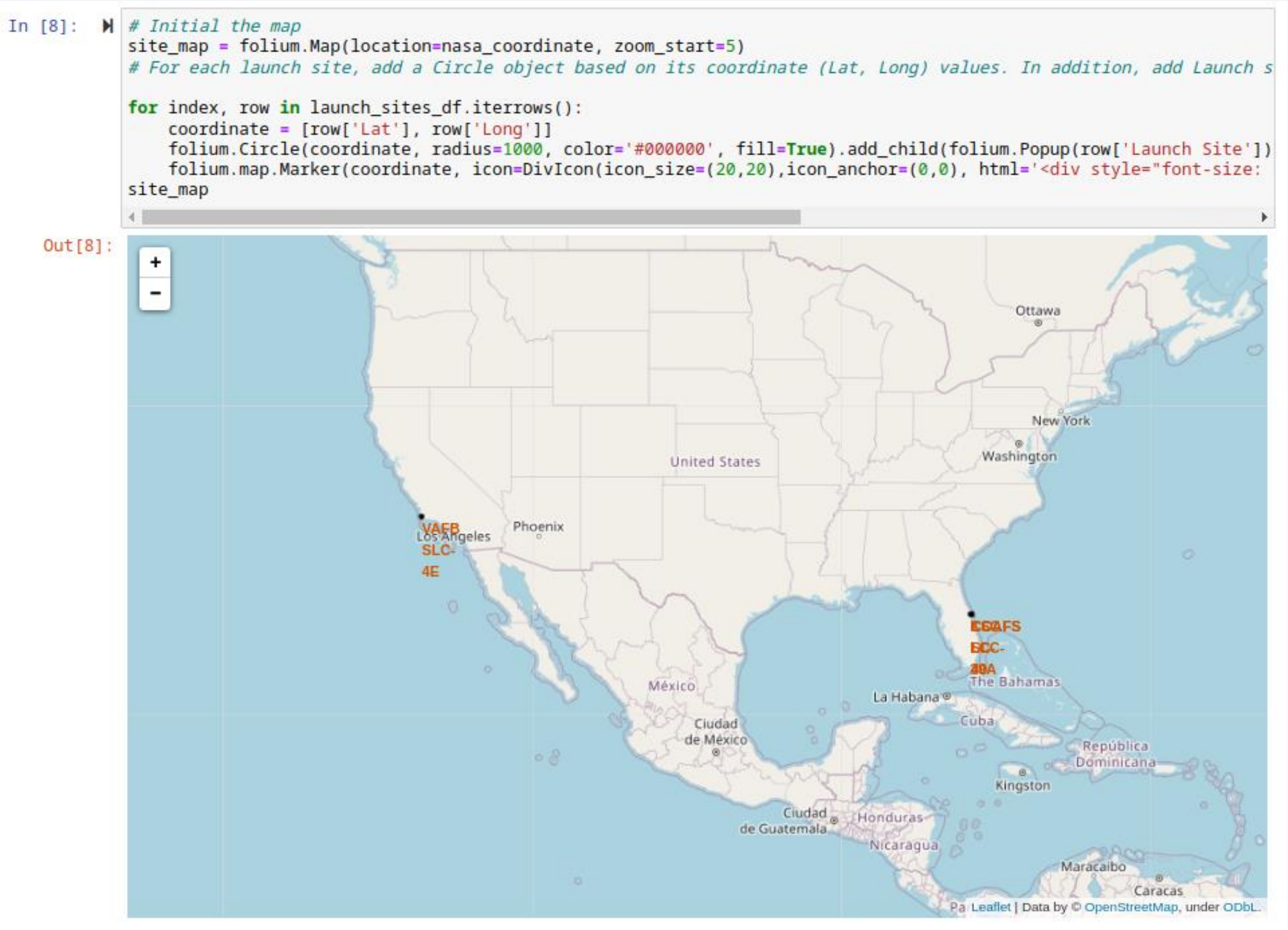
[47]:

Landing _Outcome	COUNT_LAUNCHES
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Failure	3
Controlled (ocean)	3
Failure (parachute)	2
No attempt	1

Section 3

Launch Sites Proximities Analysis

Folium map – Ground stations



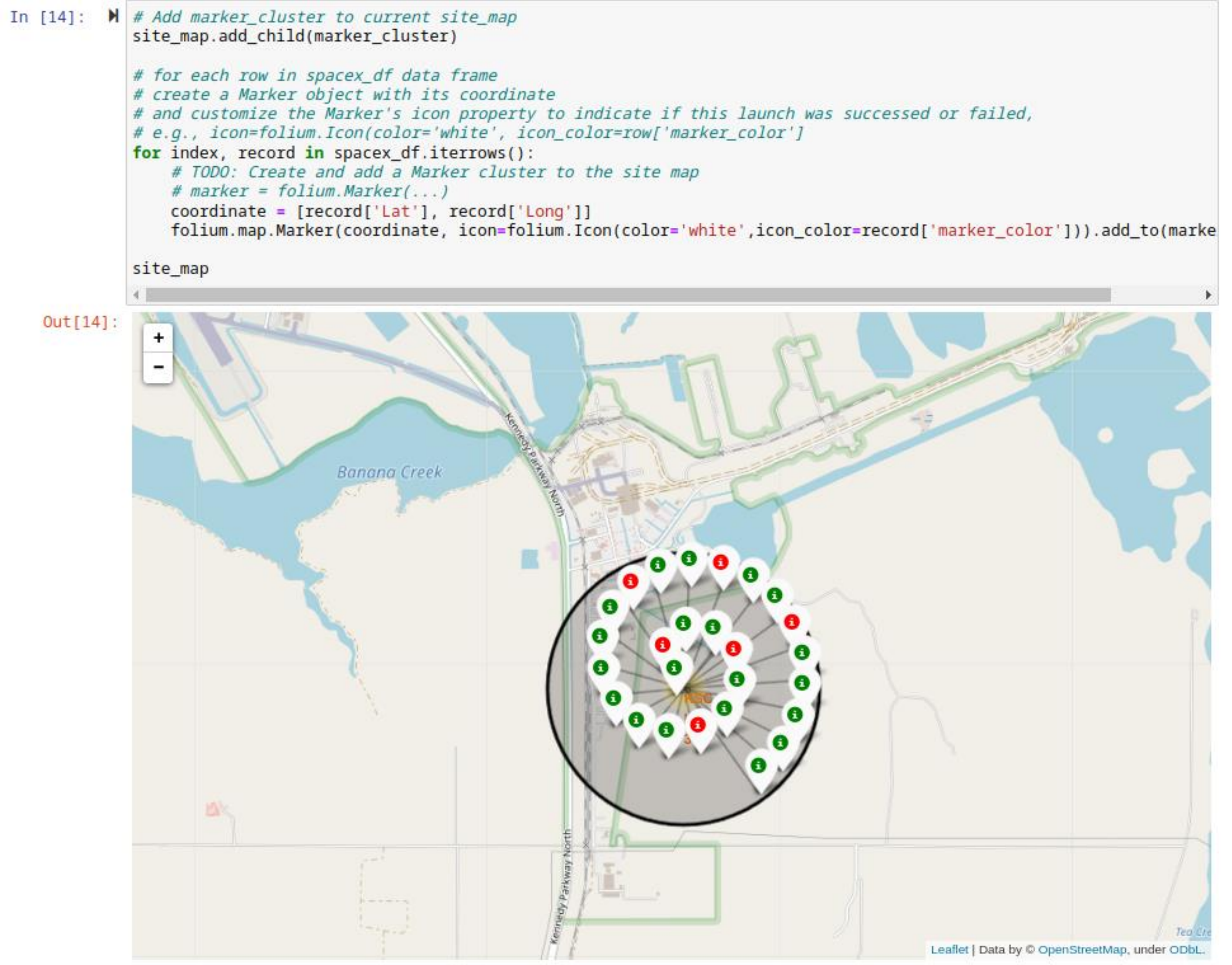
In [8]:

```
# Initial the map
site_map = folium.Map(location=nasa_coordinate, zoom_start=5)
# For each launch site, add a Circle object based on its coordinate (Lat, Long) values. In addition, add Launch s

for index, row in launch_sites_df.iterrows():
    coordinate = [row['Lat'], row['Long']]
    folium.Circle(coordinate, radius=1000, color='#000000', fill=True).add_child(folium.Popup(row['Launch Site'])
    folium.map.Marker(coordinate, icon=DivIcon(icon_size=(20,20),icon_anchor=(0,0), html='<div style="font-size:
site_map
```

Out[8]:

- We see that Space X launch sites are located on the coast of the United States

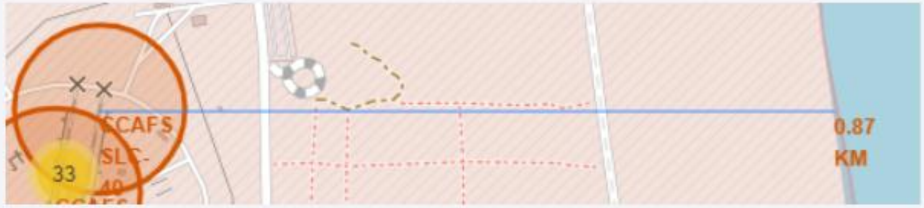
Folium map – Color Labeled Markers

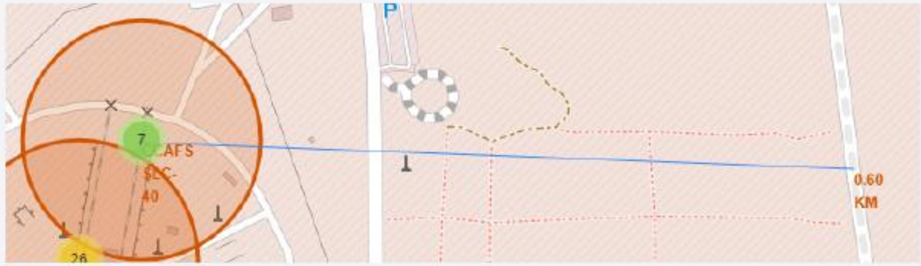


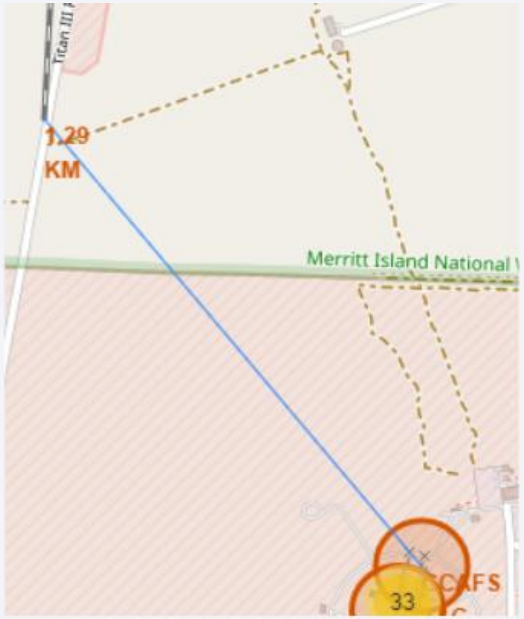
```
# Add marker_cluster to current site_map
site_map.add_child(marker_cluster)

# for each row in spacex_df data frame
# create a Marker object with its coordinate
# and customize the Marker's icon property to indicate if this launch was successed or failed,
# e.g., icon=folium.Icon(color='white', icon_color=row['marker_color']
for index, record in spacex_df.iterrows():
    # TODO: Create and add a Marker cluster to the site map
    # marker = folium.Marker(...)
    coordinate = [record['Lat'], record['Long']]
    folium.map.Marker(coordinate, icon=folium.Icon(color='white',icon_color=record['marker_color'])).add_to(marke

site_map
```

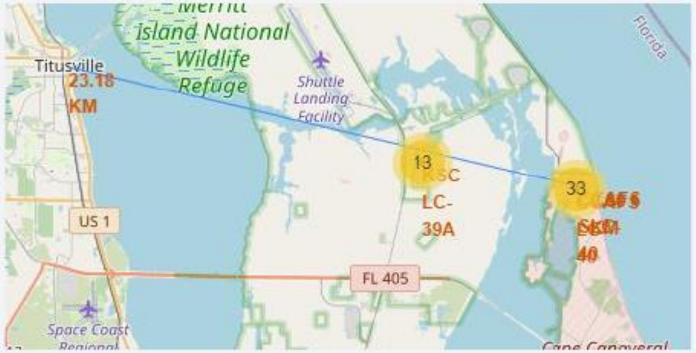
- Green marker represents successful launches. Red marker represents unsuccessful launches.

Folium Map – Distances between CCAFS SLC-40 and its proximities

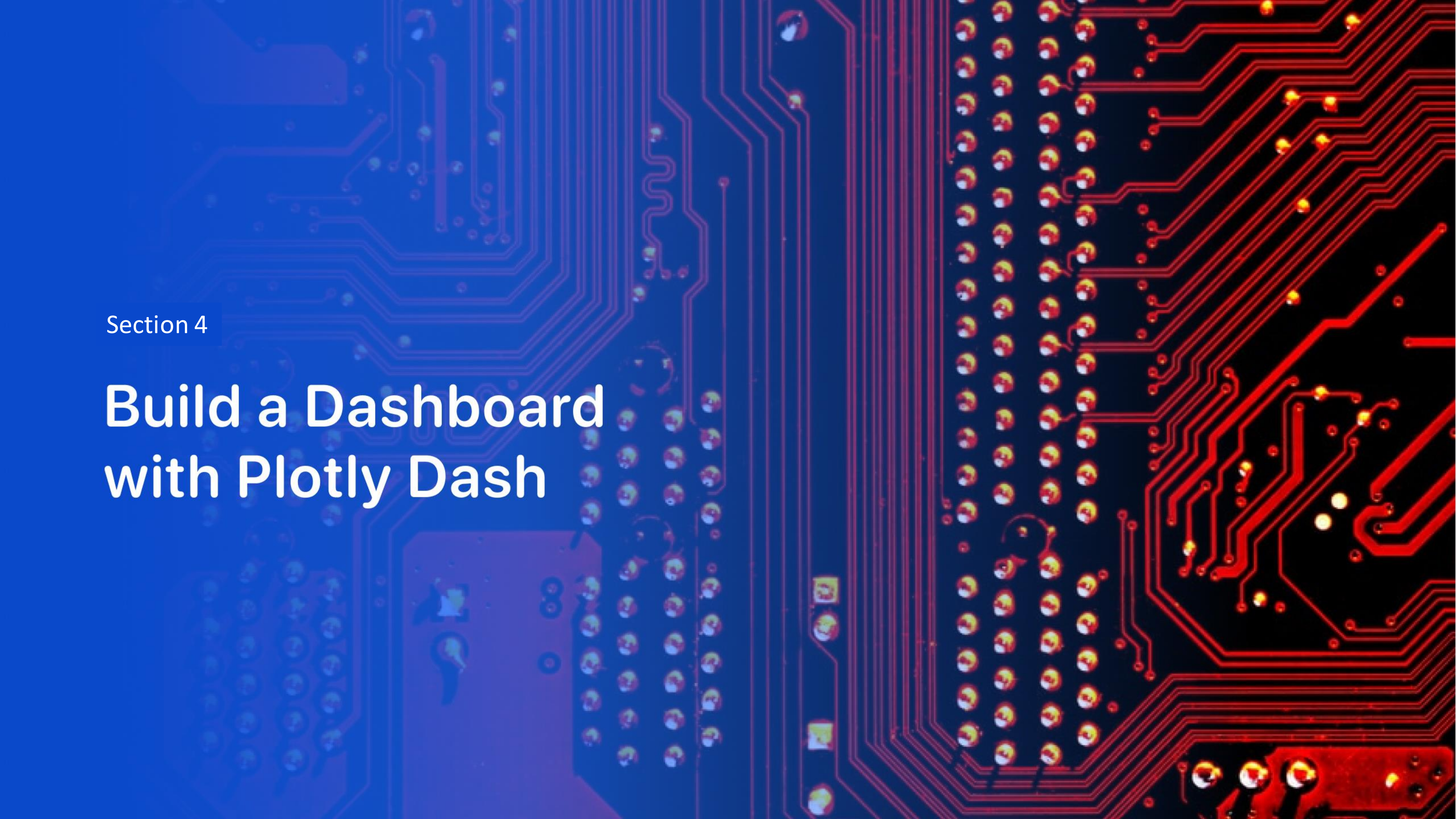








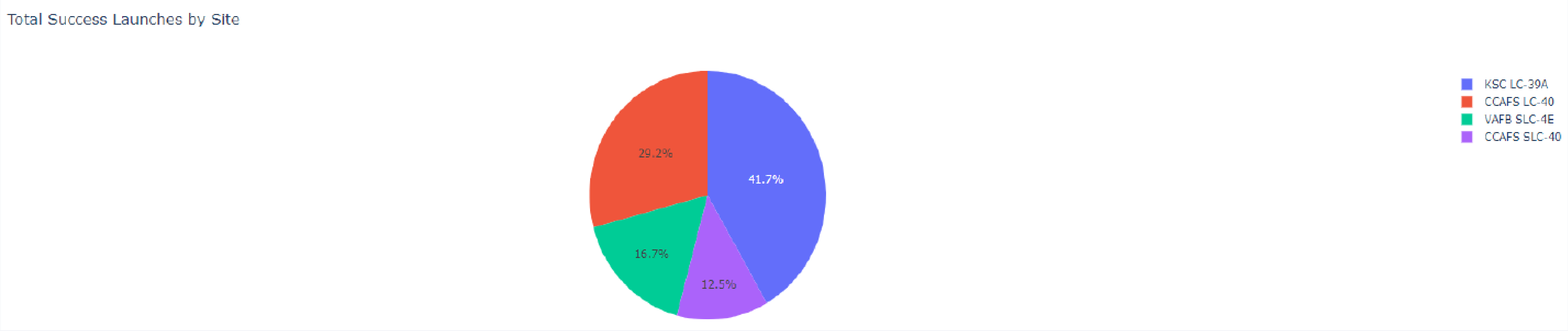
- Is CCAFS SLC-40 in close proximity to railways ? Yes
- Is CCAFS SLC-40 in close proximity to highways ? Yes
- Is CCAFS SLC-40 in close proximity to coastline ? Yes
- Do CCAFS SLC-40 keeps certain distance away from cities ? No



Section 4

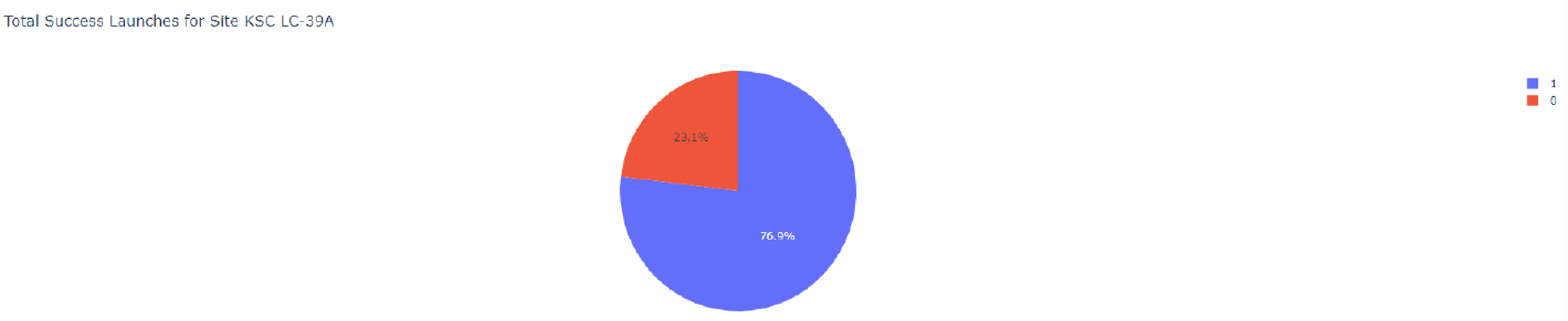
Build a Dashboard with Plotly Dash

Dashboard – Total success by Site



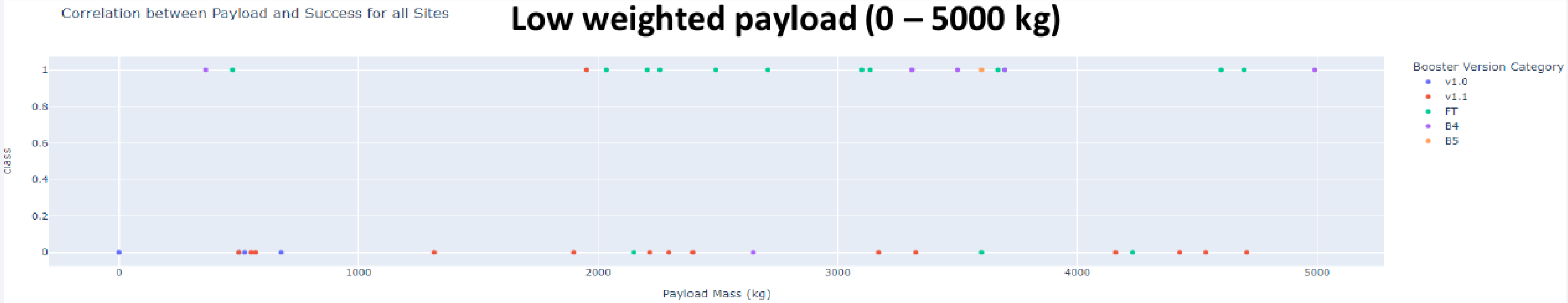
- KSC LC-39A has the best success rate of launches.

• Dashboard – Total success launches for Site KSC LC-39A

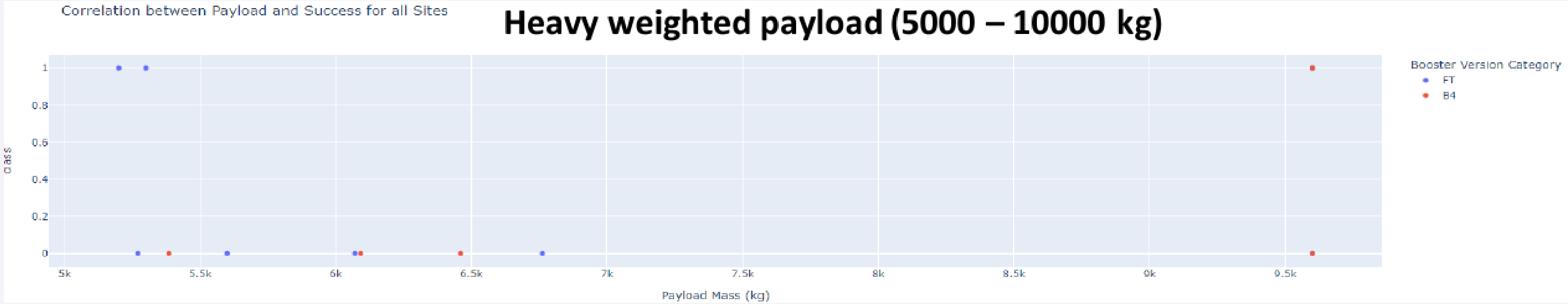


- KSC LC-39A has achieved a 76.9% success rate while getting a 23.1% failure rate.

Dashboard – Payload mass vs Outcome for all sites with different payload mass selected



Low weighted payload (0 – 5000 kg)



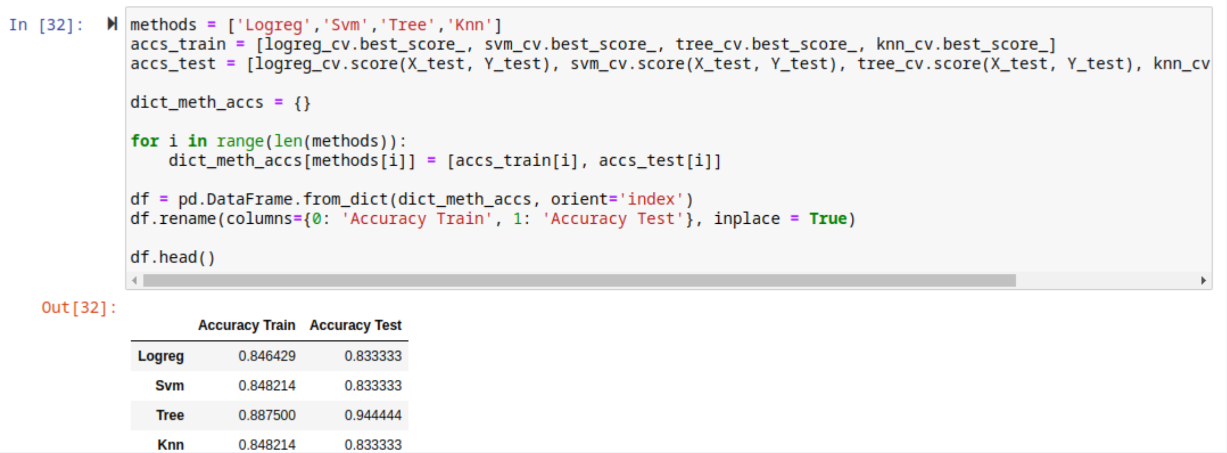
Heavy weighted payload (5000 – 10000 kg)

- Low weighted payloads have a better success rate than the heavy weighted payloads.

Section 5

Predictive Analysis (Classification)

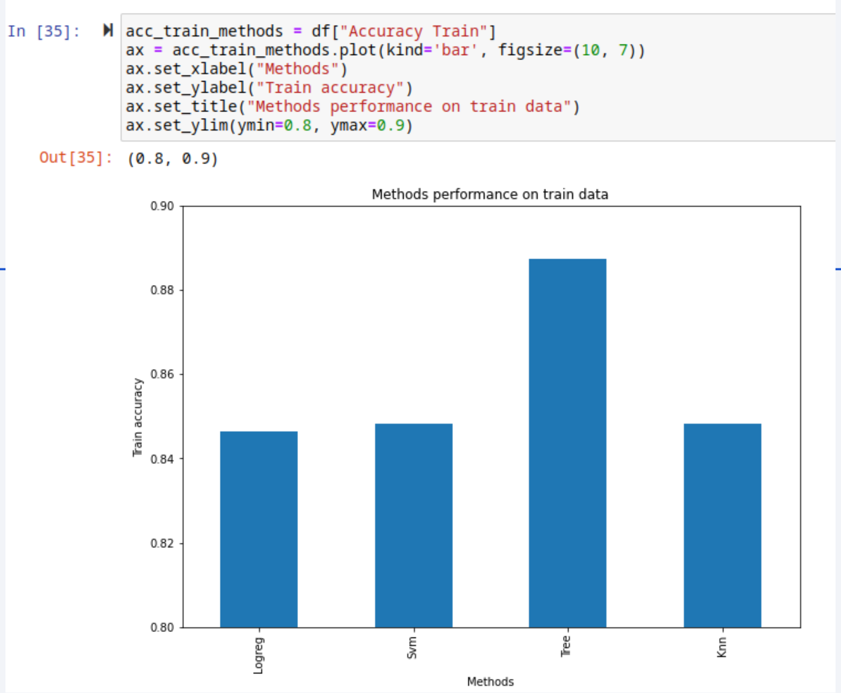
Classification Accuracy



In [32]:
```python
methods = ['Logreg','Svm','Tree','Knn']
accs_train = [logreg_cv.best_score_, svm_cv.best_score_, tree_cv.best_score_, knn_cv.best_score_]
accs_test = [logreg_cv.score(X_test, Y_test), svm_cv.score(X_test, Y_test), tree_cv.score(X_test, Y_test), knn_cv

dict_meth_accs = {}

for i in range(len(methods)):
    dict_meth_accs[methods[i]] = [accs_train[i], accs_test[i]]

df = pd.DataFrame.from_dict(dict_meth_accs, orient='index')
df.rename(columns={0: 'Accuracy Train', 1: 'Accuracy Test'}, inplace = True)

df.head()
```

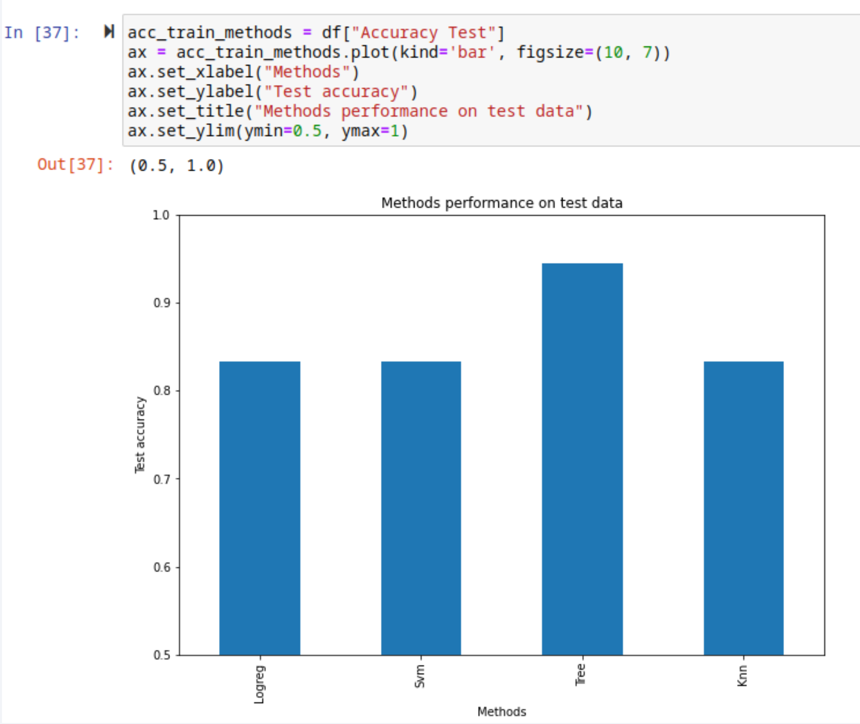
Out[32]:

	Accuracy Train	Accuracy Test
Logreg	0.846429	0.833333
Svm	0.848214	0.833333
Tree	0.887500	0.944444
Knn	0.848214	0.833333



In [35]:
```python
acc_train_methods = df["Accuracy Train"]
ax = acc_train_methods.plot(kind='bar', figsize=(10, 7))
ax.set_xlabel("Methods")
ax.set_ylabel("Train accuracy")
ax.set_title("Methods performance on train data")
ax.set_ylim(ymin=0.8, ymax=0.9)
```

Out[35]:
```
(0.8, 0.9)
```



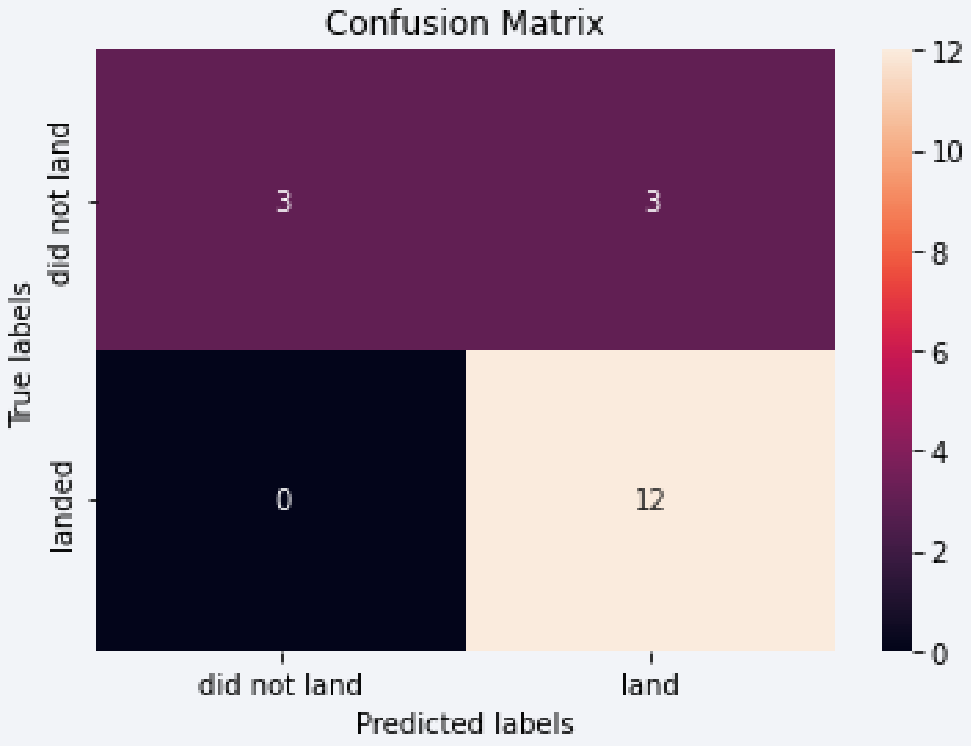
In [37]:
```python
acc_train_methods = df["Accuracy Test"]
ax = acc_train_methods.plot(kind='bar', figsize=(10, 7))
ax.set_xlabel("Methods")
ax.set_ylabel("Test accuracy")
ax.set_title("Methods performance on test data")
ax.set_ylim(ymin=0.5, ymax=1)
```

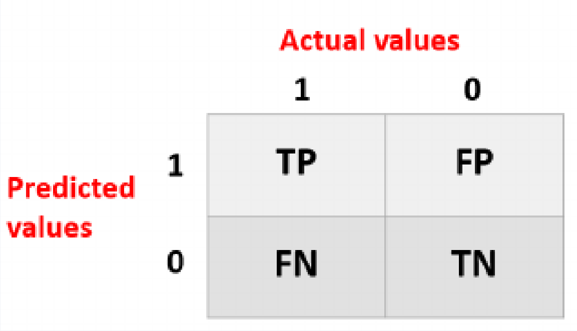
Out[37]:
```
(0.5, 1.0)
```

- The accuracy is very similar across different models. Overall, decision tree is slightly better

Confusion Matrix

- The confusion matrix is the same for all models.
- The main problem of these models are false positives.





		Actual values	
		1	0
Predicted values	1	TP	FP
	0	FN	TN

Conclusions

- The success of a mission can be explained by several factors such as the launch site, the orbit and especially the number of previous launches. Indeed, we can assume that there has been a gain in knowledge between launches that allowed to go from a launch failure to a success.
- The orbits with the best success rates are GEO, HEO, SSO, ES-L1.
- Depending on the orbits, the payload mass can be a criterion to take into account for the success of a mission. Some orbits require a light or heavy payload mass. But generally low weighted payloads perform better than the heavy weighted payloads.
- With the current data, we cannot explain why some launch sites are better than others (KSC LC-39A is the best launch site). To get an answer to this problem, we could obtain atmospheric or other relevant data.
- For this dataset, we choose the Decision Tree Algorithm as the best model even if the test accuracy between all the models used is identical. We choose Decision Tree Algorithm because it has a better train accuracy.

Thank you!