IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ricardo Dexter Mendoza
07-02-2022

Outline

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

Executive Summary

The research sets out to prove if it is possible to predict whether the first stage of the launch would be successful based on the launch parameter. After the data was gathered, processed, and predicting models were ran, it was found that it is possible to predict the success of the Mission’s first phase up to a 95% accuracy

Introduction

The Research focuses around possible for the launch parameters as Launch site, Payload Mass, etc having an effect on the success rate of the first stage of the launch. The success rate in this instance means that the first phase trustors can be unused in a future launch. These parameters vary with every launch and the data is publicly available.

The research sets out to explore which feature set to be used for the gathered data and which statistical model to yield the best results when predicting to success rate.

Section 1

Methodology

Methodology

Executive Summary

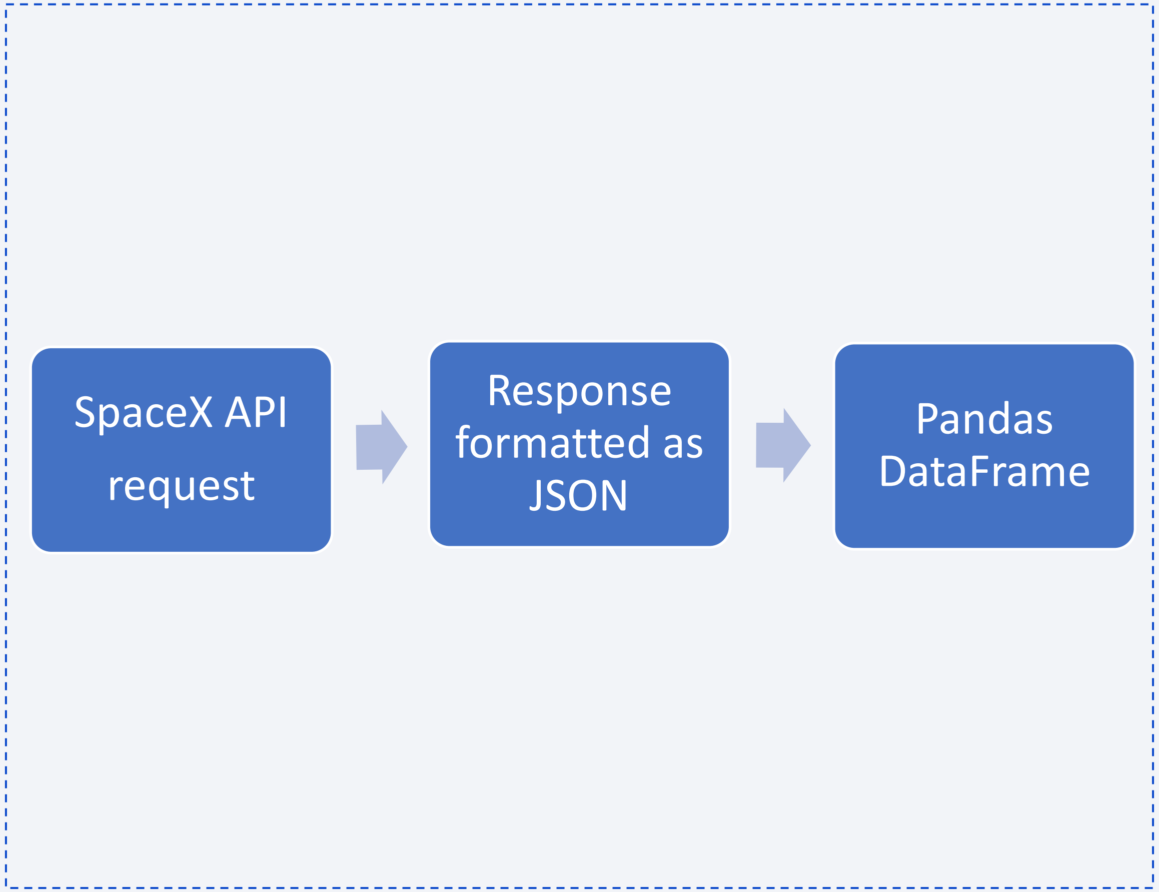
- Data collection methodology:
    - Data was collected via API calls and web-scrapping on publicly available data.
- Perform data wrangling
    - The Data was examined for nulls and datatype inconsistences and streamlining the Outcome column
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - KNN, SVM , Decision tree and Logistic Regressions methods were tested using multiple parameters to find the best performing model.

Data Collection

The first set of data was retrieved by using the Space API. The JSON Response was parsed and turned into a Data Frame. The data was then filtered to only contain Falcon 9 data. The second set of data was scraped from the website of the Falcon 9 Wiki page. The data was parsed using Beautiful soup to eventually produce a Data Frame.

Data Collection – SpaceX API

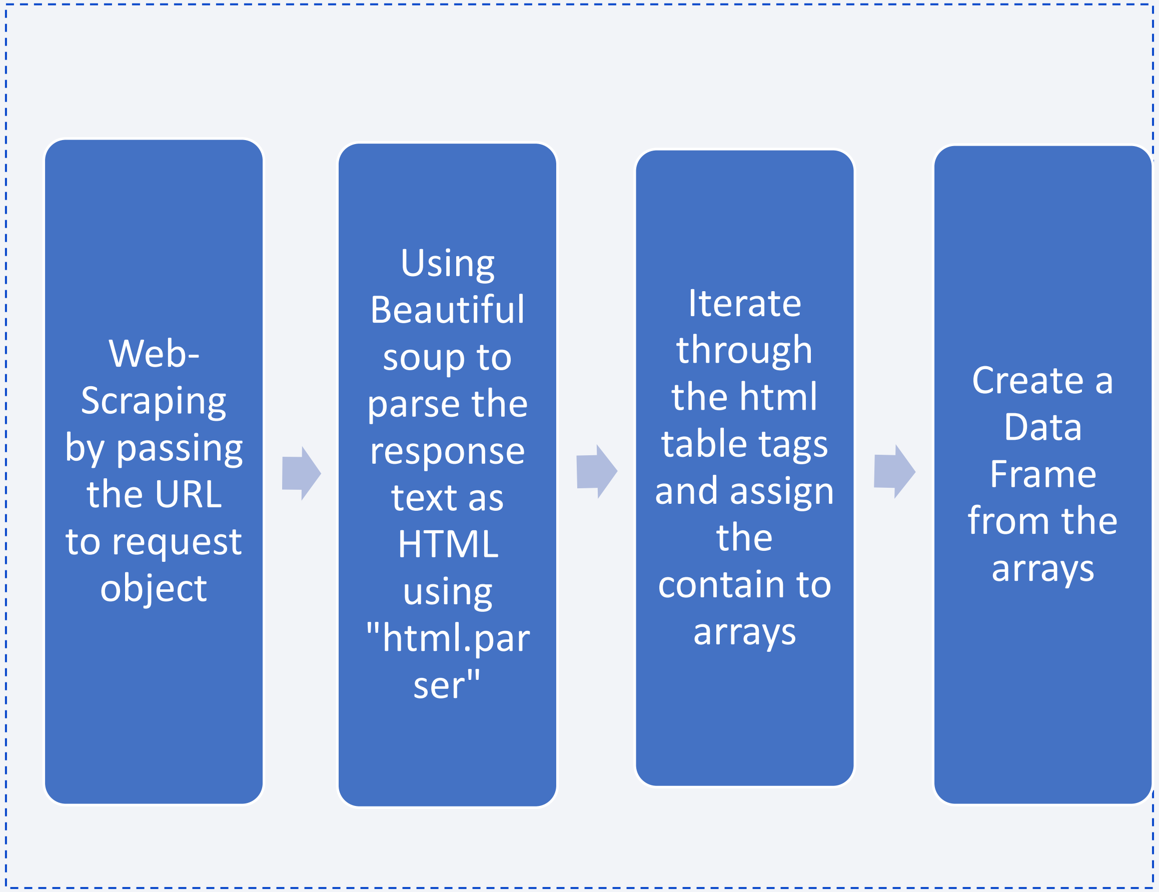
- Data collection with SpaceX REST calls using the following endpoint: "https://api.spacexdata.com/v4/launches/past"
- Link to Completed SpaceX API calls notebook (Click here) as an external reference and peer-review purpose



Data Collection - Scraping

- Webscrapping the Falcon9 Launch Wiki page from its URL

- The completed web scraping notebook can be accessed Here, as an external reference and peer-review purpose



Web-Scraping by passing the URL to request object

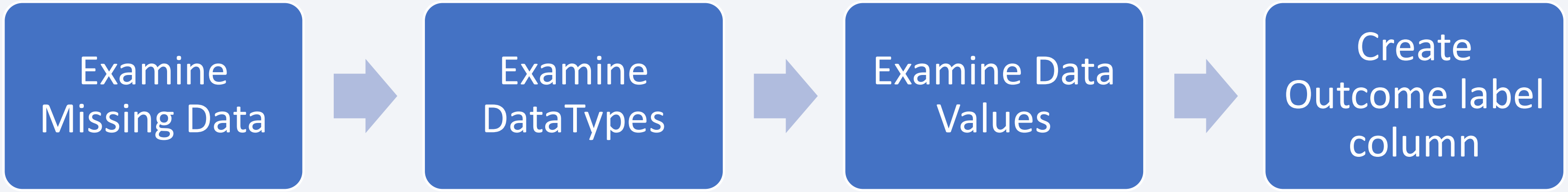
Using Beautiful soup to parse the response text as HTML using "html.parser"

Iterate through the html table tags and assign the contain to arrays

Create a Data Frame from the arrays

Data Wrangling

- The Data was first examined for nulls and datatype inconsistences. After examining the values of the columns a new binary Outcome column was created to indicate if the launch was successful or not based on multiple different outcomes.
- The completed data wrangling related notebooks can be accessed Here, as an external reference and peer-review purpose



Examine Missing Data

Examine DataTypes

Examine Data Values

Create Outcome label column

EDA with Data Visualization

The Relationship between some of the features were examined using Scatter plots and bar graphs. These Relationship include Payload and Launch Site, success rate of each orbit type, success rate of each orbit type, Payload and Orbit type, launch success and date to get the yearly trend. After this was done the Feature set was selected and categorical features were flattened using OneHotEncoding

- The completed EDA with data visualization notebook can be accessed HERE, as an external reference and peer-review purpose

EDA with SQL

Summarize the SQL queries performed

- Display the names of the unique launch Sites
- Display the total payload mass carried by boosters launched by NASA
- Find the date when the first successful landing outcome in ground pad
- List the names of the booster_versions which have carried the maximum payload mass
- Etc
- The completed EDA with SQL notebook can be accessed HERE, as an external reference and peer-review purpose

Build an Interactive Map with Folium

The Folium Maps were used to give a visual preceptive of the Launch sites and its proximate. Markers were used to denote the different Launches and the related success

- The completed interactive map with Folium map can be accessed HERE , as an external reference and peer-review purpose

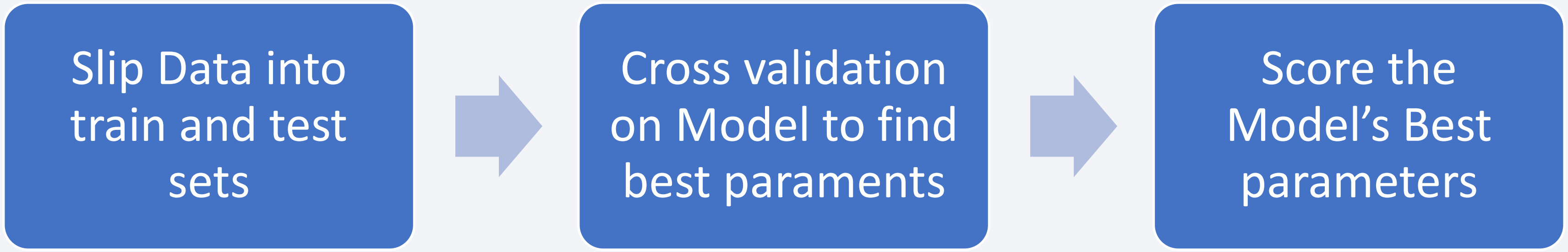
Build a Dashboard with Plotly Dash

The Dashboard contained a dropdown box of all the Launch sites which affected a Pie Chart and a Scatter plot base on it's selection. In addition the Scatter plot was controlled using a Payload Mass slider.

- The completed Plotly Dash notebook can be accessed HERE, as an external reference and peer-review purpose

Predictive Analysis (Classification)

- A combination of models were used and best one was selected. Each model was training to find its optimal hyperparameters using GridSearchCV's cross-validation on the training and testing sets. The Four models used were: KNN, SVM Decision Tree and Logistic Regression.



Slip Data into train and test sets

Cross validation on Model to find best paraments

Score the Model's Best parameters

- The completed predictive analysis lab can be accessed HERE, as an external reference and peer-review purpose

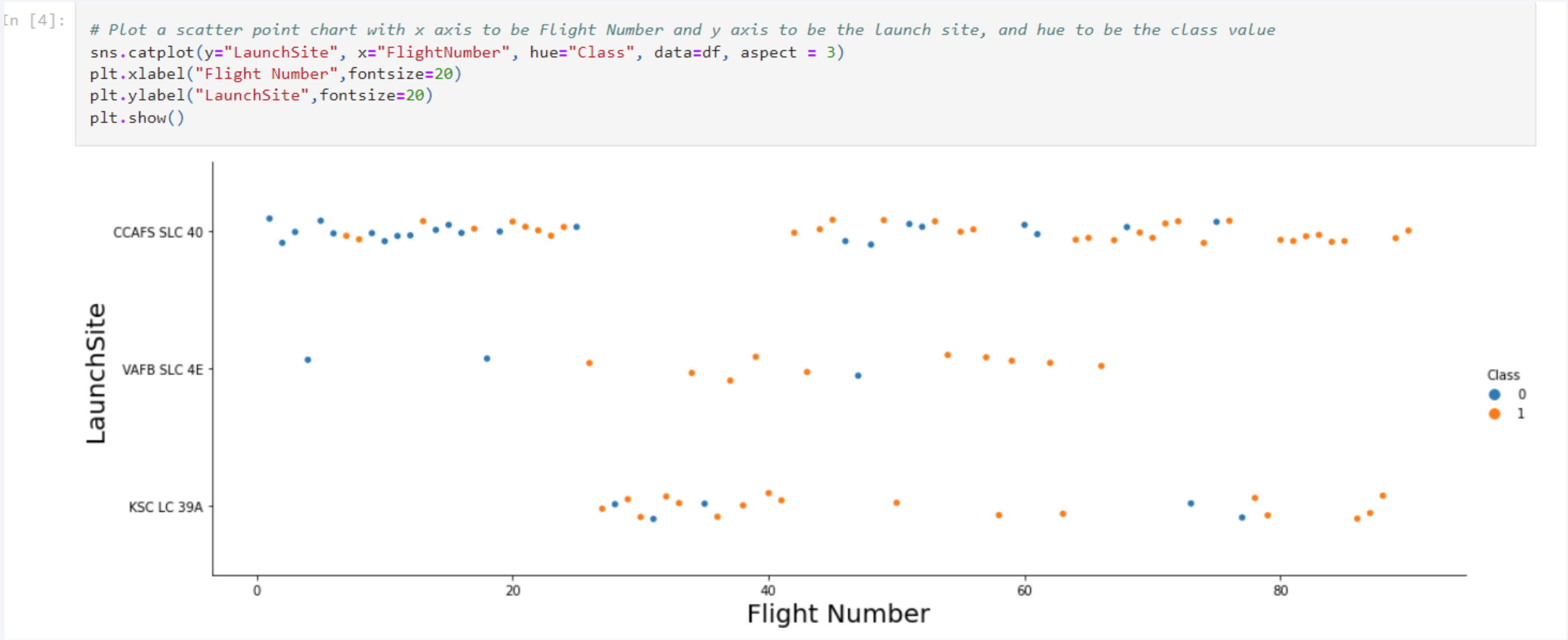
Results

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

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

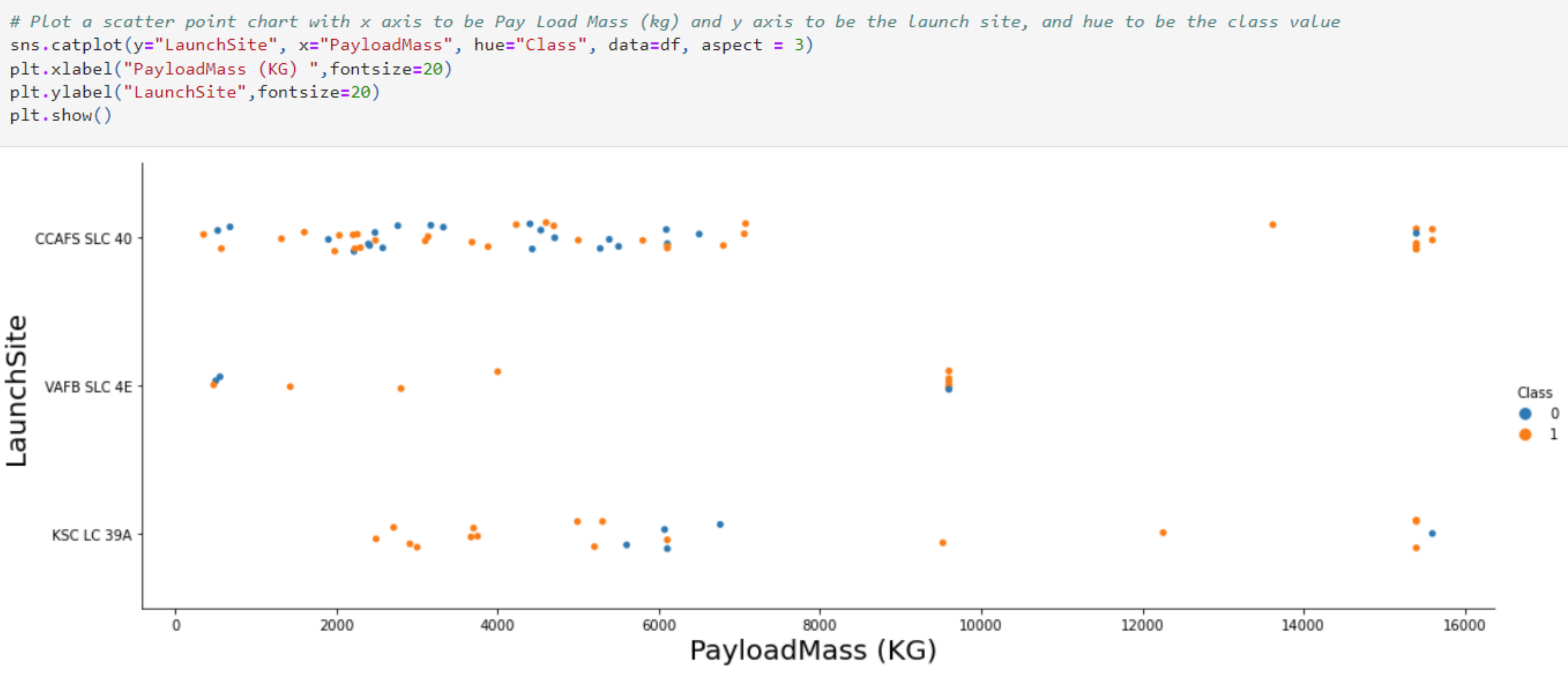


In [4]:

```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 3)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```

The graph shows that the CCAFS SLC 40 Launch pad had the Most Flights and also had the most Bad outcomes.

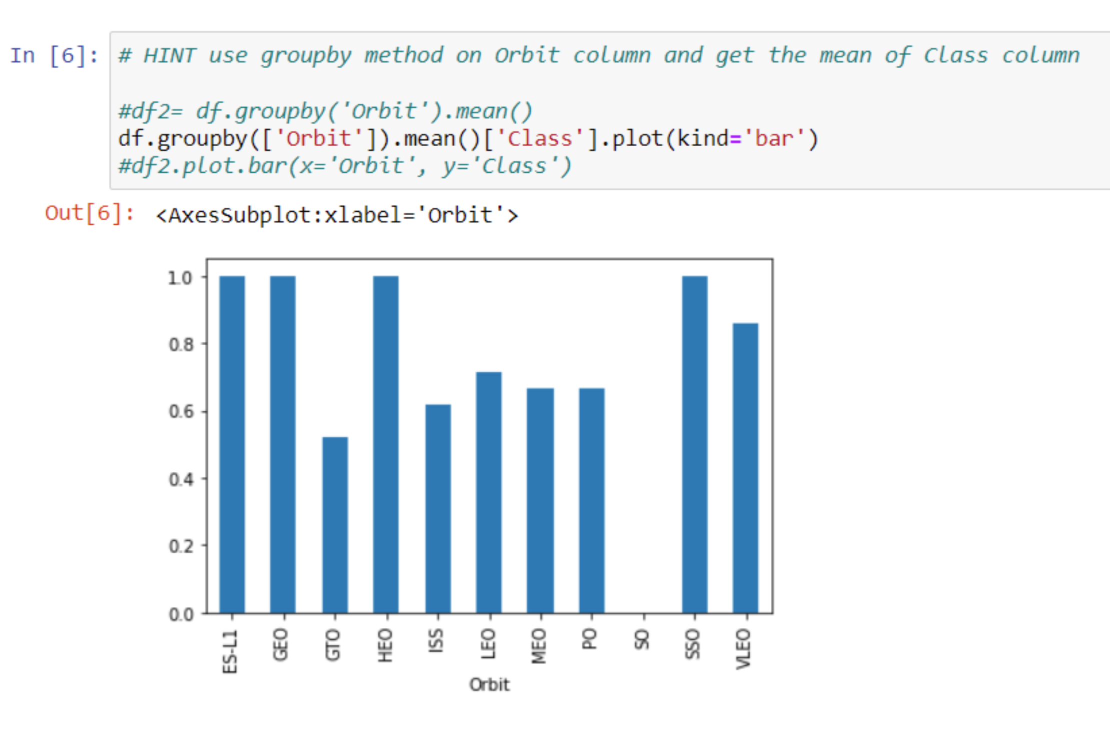
Payload vs. Launch Site



```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 3)
plt.xlabel("PayloadMass (KG) ",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```

The graph shows that the VAFB-SLC site there are no rockets launched for heavy payload mass (greater than 10000).

Success Rate vs. Orbit Type



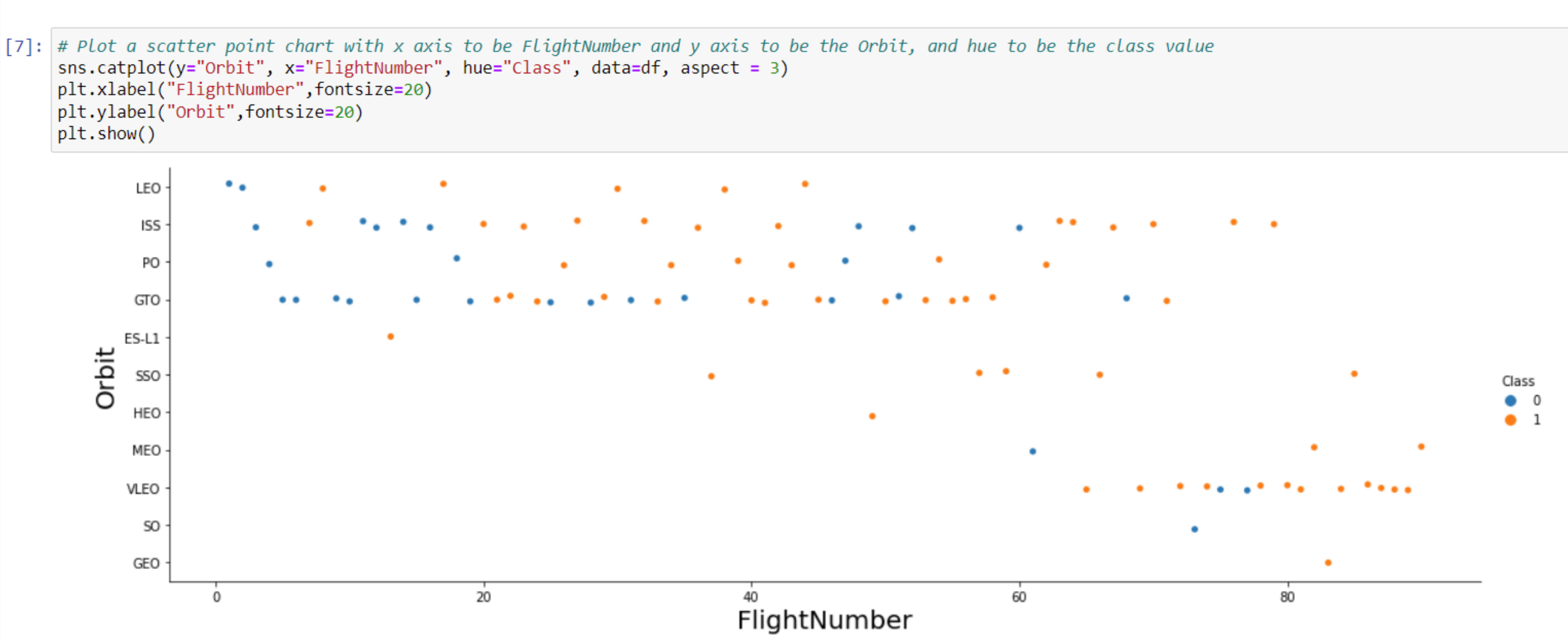
```
In [6]: # HINT use groupby method on Orbit column and get the mean of Class column

        #df2= df.groupby('Orbit').mean()
        df.groupby(['Orbit']).mean()['Class'].plot(kind='bar')
        #df2.plot.bar(x='Orbit', y='Class')

Out[6]: <AxesSubplot:xlabel='Orbit'>
```

The Graph shows that the orbits of ES-L1, GEO, HEO & SSO all have high rates of the Mission success, However these can be due to few flight numbers at these orbits.

Flight Number vs. Orbit Type

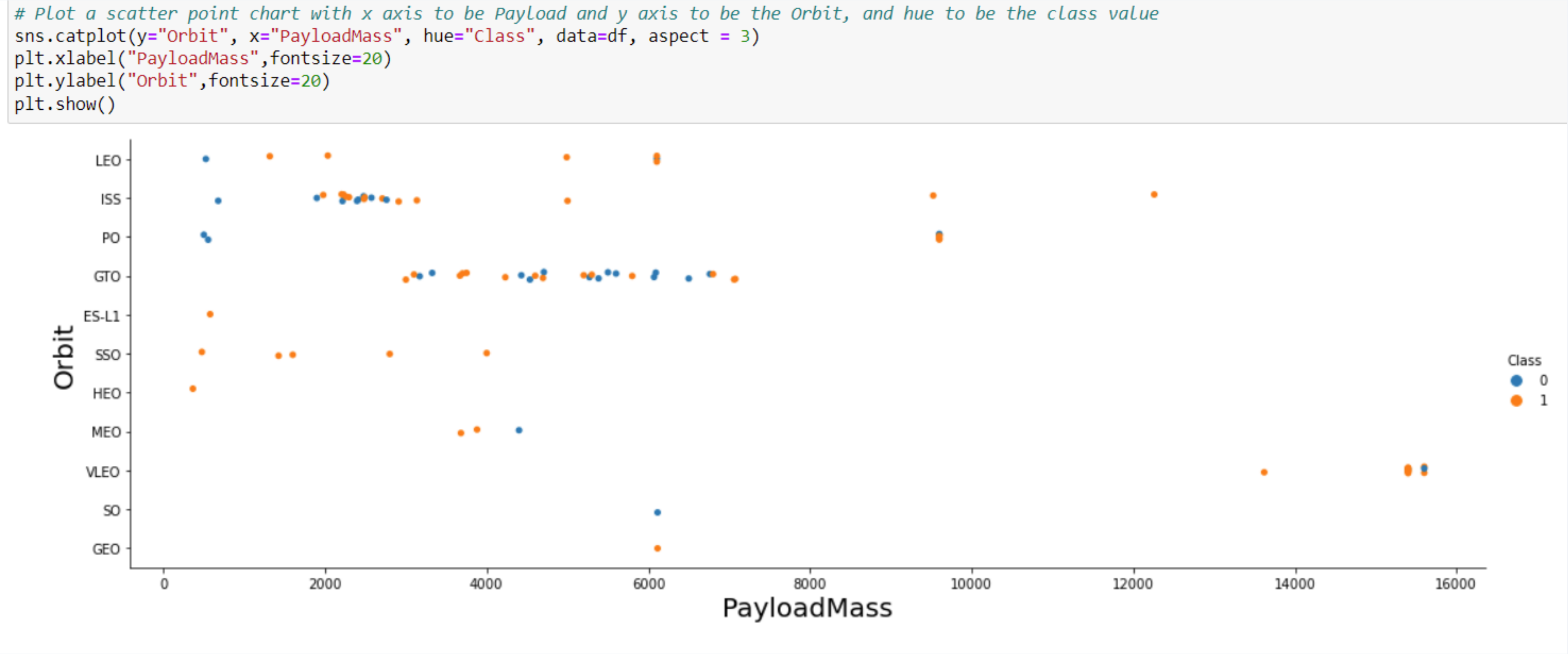


[7]:

```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 3)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

The graph shows 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.

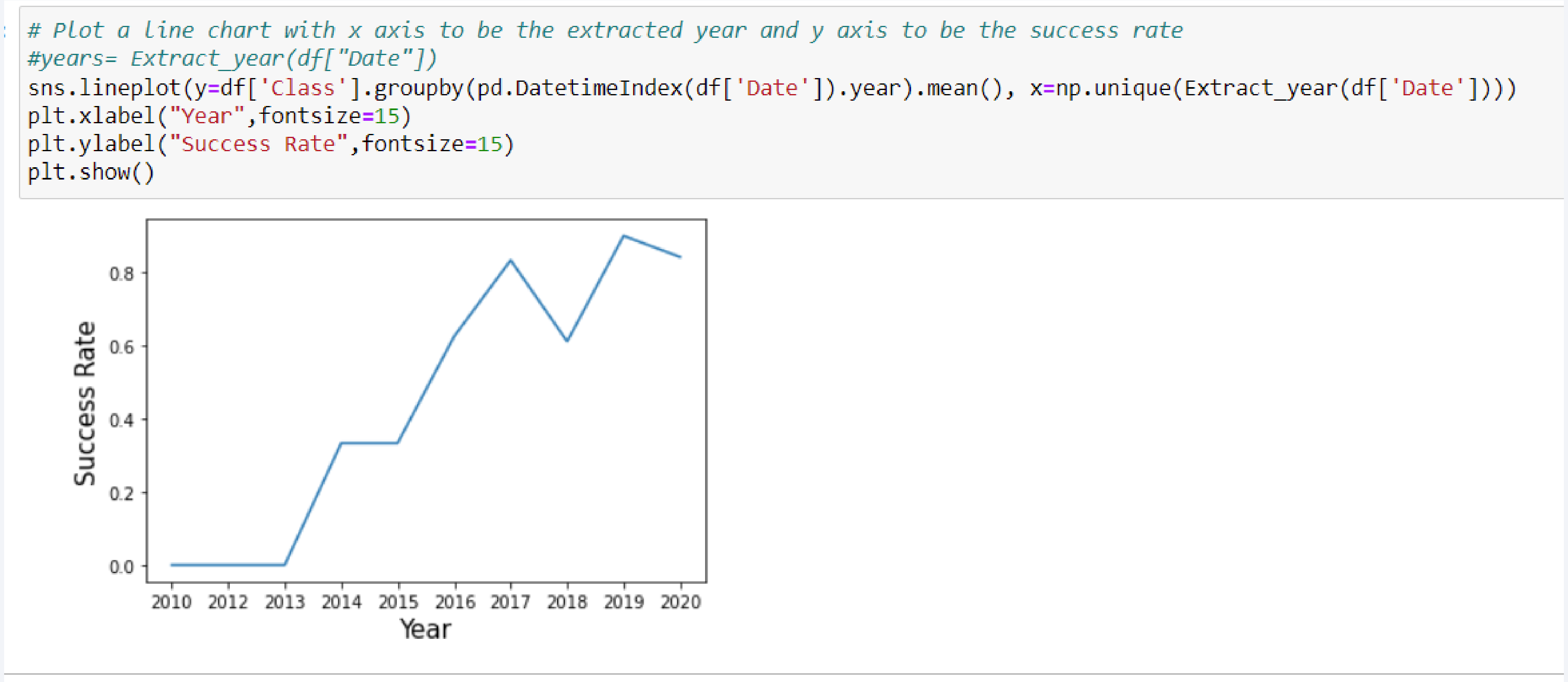
Payload vs. Orbit Type



```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 3)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

- The graph shows that with heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.

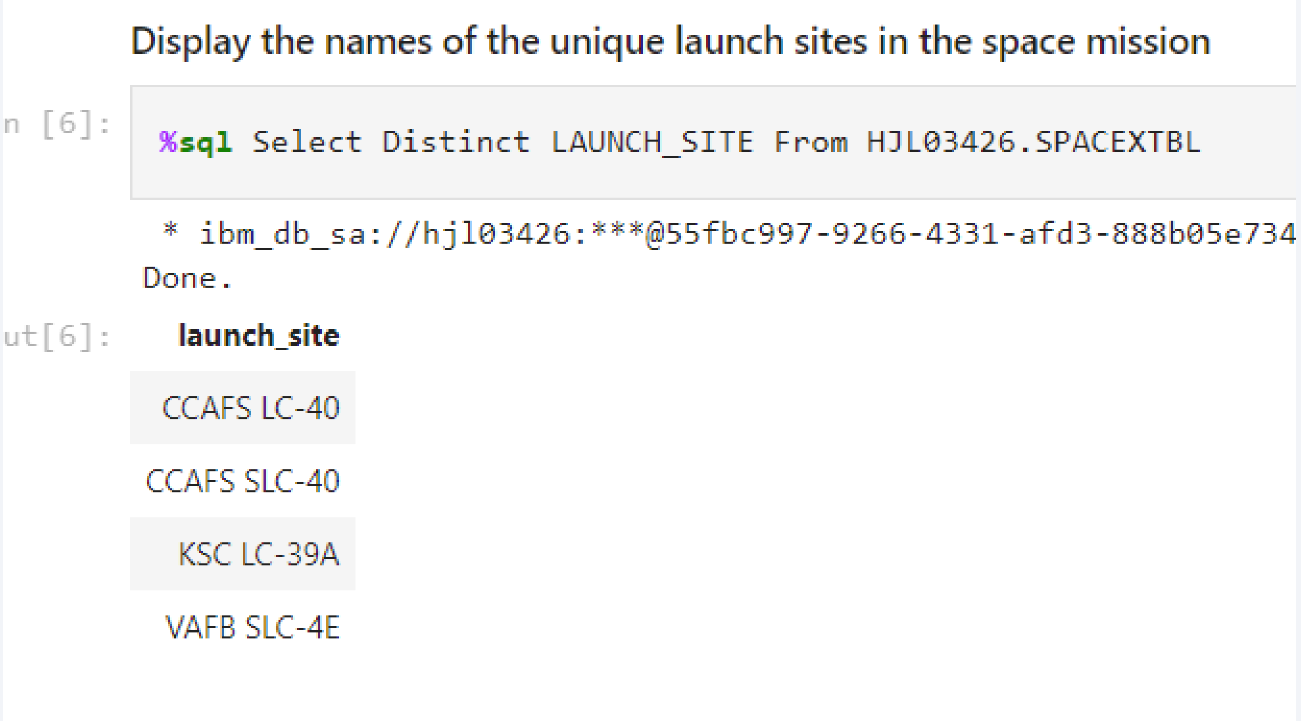
Launch Success Yearly Trend



```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
#years= Extract_year(df["Date"])
sns.lineplot(y=df['Class'].groupby(pd.DatetimeIndex(df['Date']).year).mean(), x=np.unique(Extract_year(df['Date'])))
plt.xlabel("Year",fontsize=15)
plt.ylabel("Success Rate",fontsize=15)
plt.show()
```

The graph shows that the success rate kept increasing since 2013 till 2020

All Launch Site Names



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

```
n [6]:  %sql Select Distinct LAUNCH_SITE From HJL03426.SPACEXTBL
```

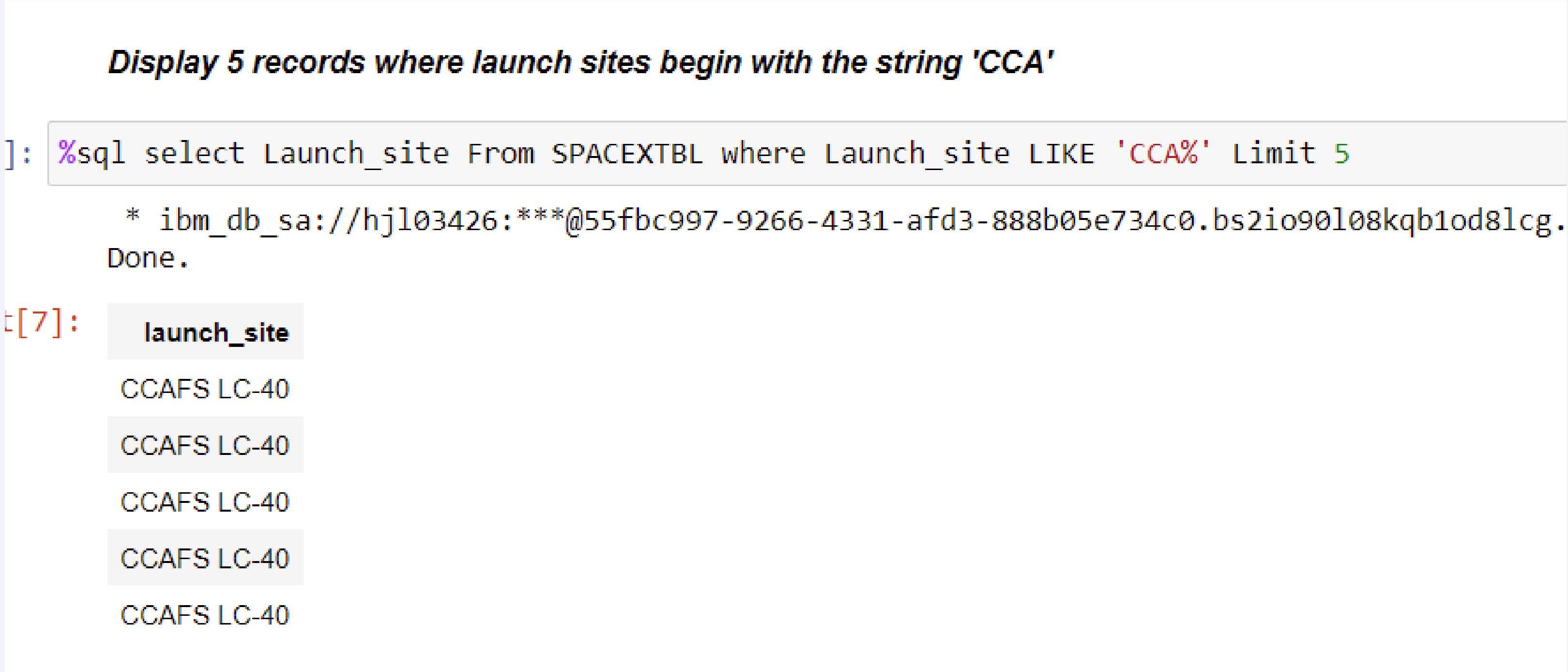
```
 * ibm_db_sa://hjl03426:***@55fbc997-9266-4331-afd3-888b05e734
Done.
```

ut[6]:

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

The List of Unique Launch Sites

Launch Site Names Begin with 'CCA'



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

```
]: %sql select Launch_site From SPACEXTBL where Launch_site LIKE 'CCA%' Limit 5
```

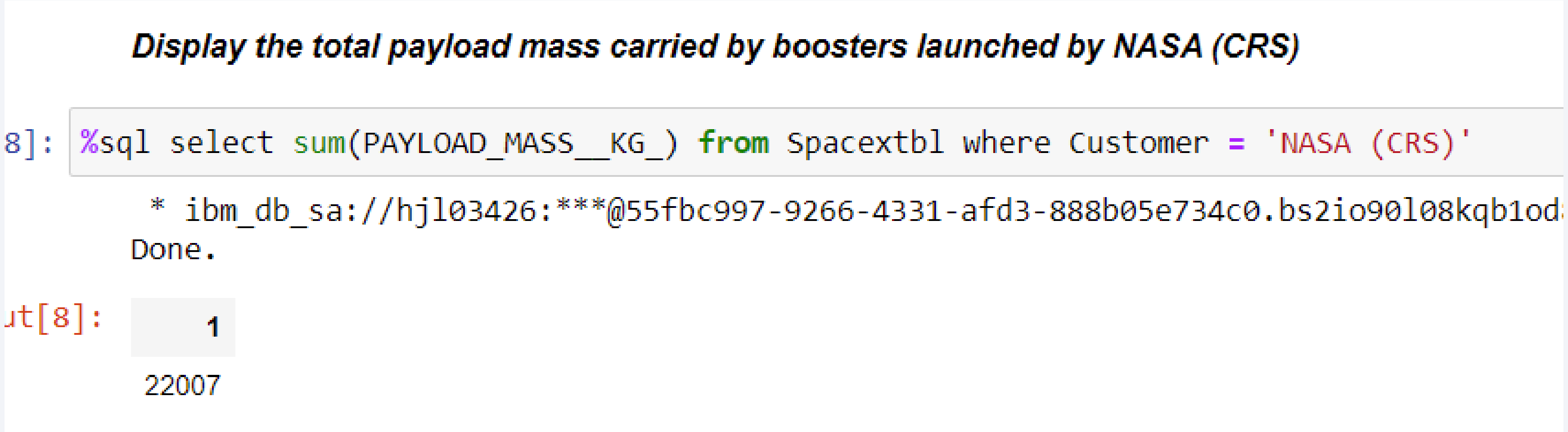
```
 * ibm_db_sa://hjl03426:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.
Done.
```

t[7]:

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

The first 5 Results in the table

Total Payload Mass



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

```
8]: %sql select sum(PAYLOAD_MASS__KG_) from Spacextbl where Customer = 'NASA (CRS)'
```

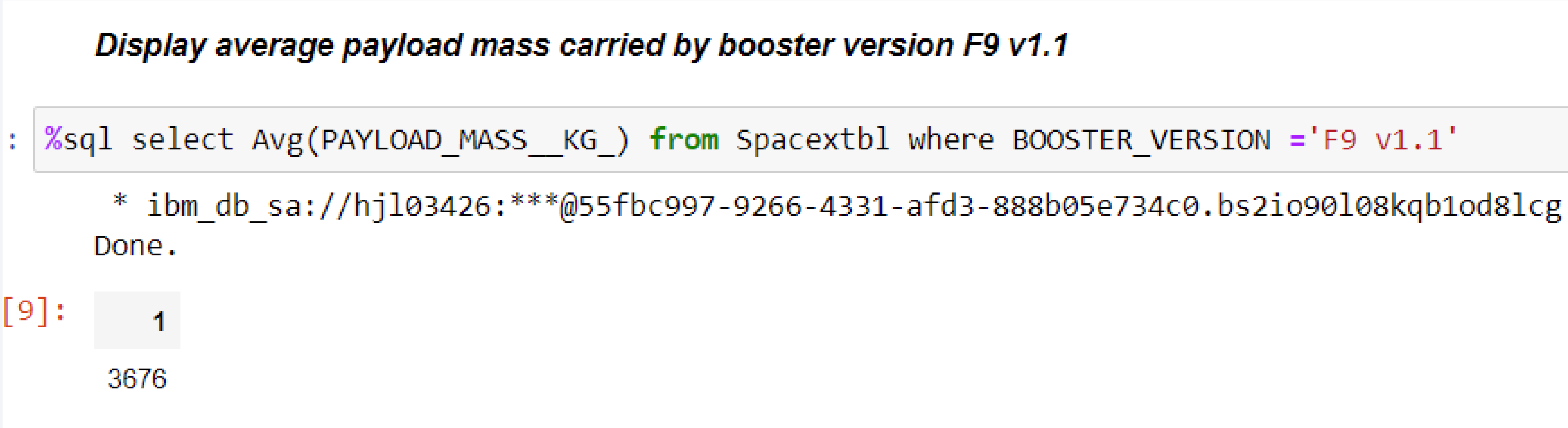
```
 * ibm_db_sa://hjl03426:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od
Done.
```

ut[8]:

1
22007

The Sum of the Payload Mass in the Table

Average Payload Mass by F9 v1.1



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

```
: %sql select Avg(PAYLOAD_MASS__KG_) from Spacextbl where BOOSTER_VERSION ='F9 v1.1'
```

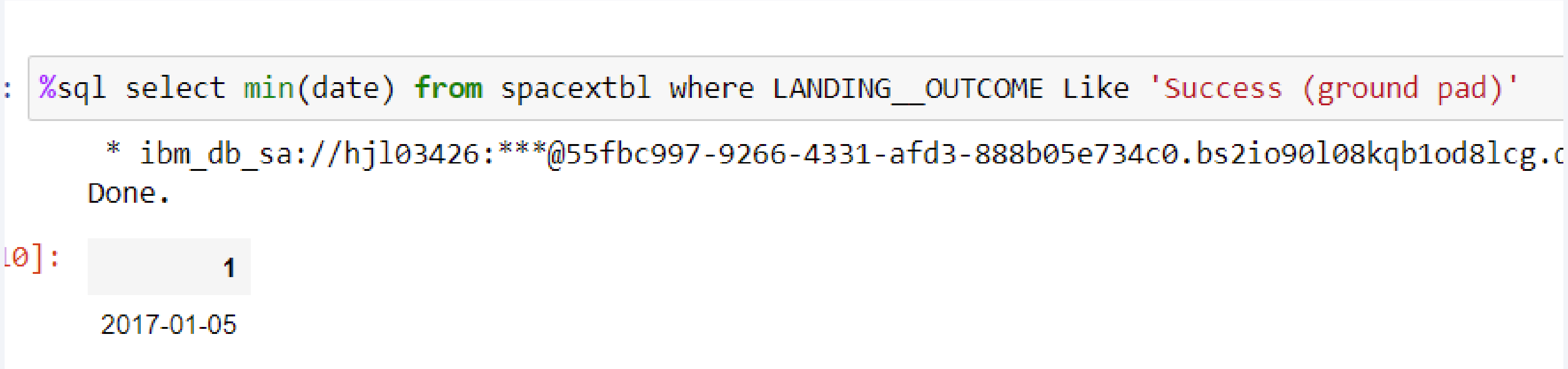
```
 * ibm_db_sa://hjl03426:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg
Done.
```

[9]:

1
3676

- the average payload mass carried by booster version F9 v1.1 in the Table

First Successful Ground Landing Date



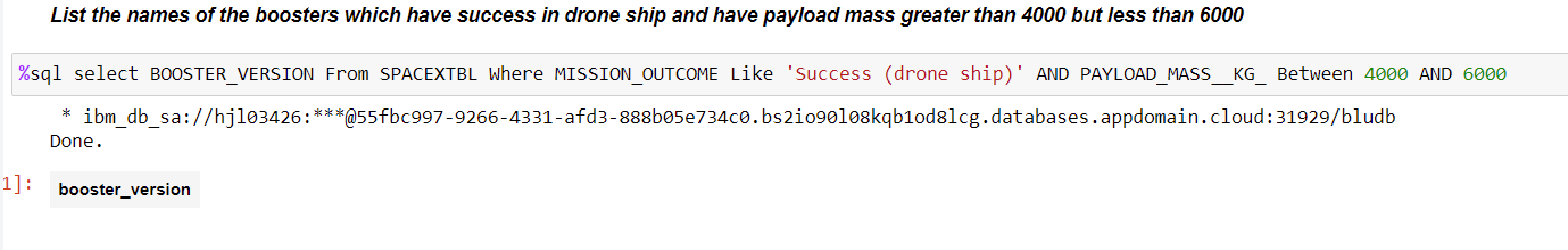
```
: %sql select min(date) from spacextbl where LANDING__OUTCOME Like 'Success (ground pad)'
```

```
 * ibm_db_sa://hjl03426:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.c
Done.
```

```
l0]:          1
      2017-01-05
```

The dates of the first successful landing outcome on ground pad in the Table

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

```
%sql select BOOSTER_VERSION From SPACEXTBL Where MISSION_OUTCOME Like 'Success (drone ship)' AND PAYLOAD_MASS__KG_ Between 4000 AND 6000
```

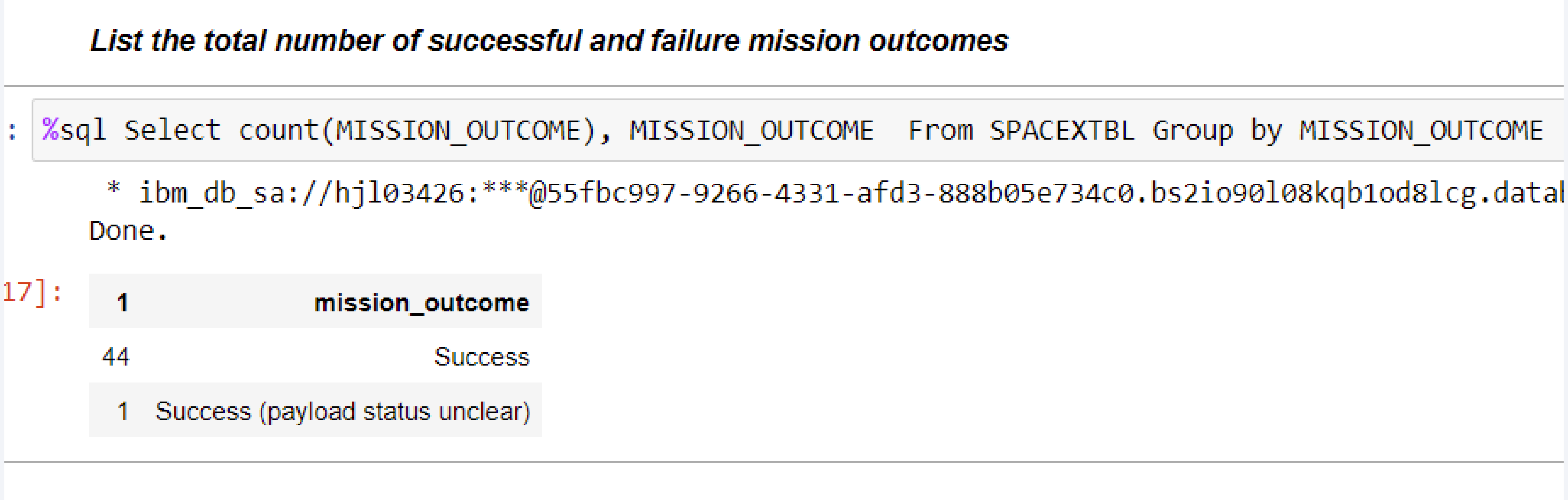
```
 * ibm_db_sa://hjl03426:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

1]:

booster_version

- Errors loading data in to the db2 database can account for the results

Total Number of Successful and Failure Mission Outcomes



List the total number of successful and failure mission outcomes

```
: %sql Select count(MISSION_OUTCOME), MISSION_OUTCOME  From SPACEXTBL Group by MISSION_OUTCOME
```

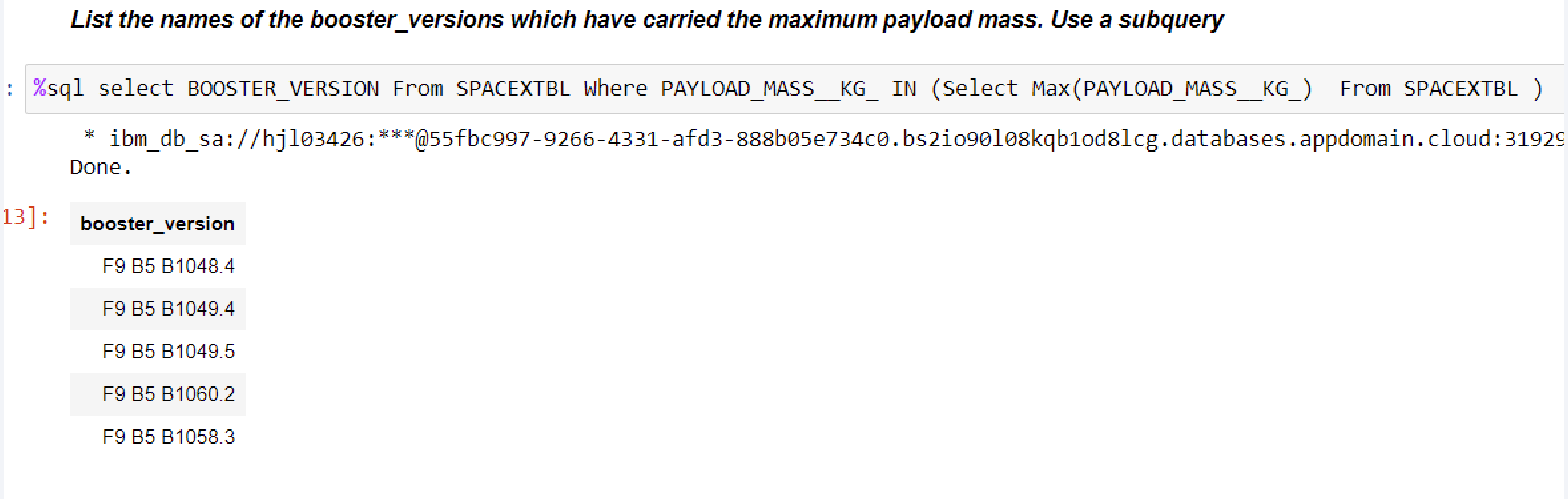
```
 * ibm_db_sa://hjl03426:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.datal
Done.
```

17]:

1	mission_outcome
44	Success
1	Success (payload status unclear)

total number of successful and failure mission outcomes in the table

Boosters Carried Maximum Payload



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

```
: %sql select BOOSTER_VERSION From SPACEXTBL Where PAYLOAD_MASS__KG_ IN (Select Max(PAYLOAD_MASS__KG_)  From SPACEXTBL )
```

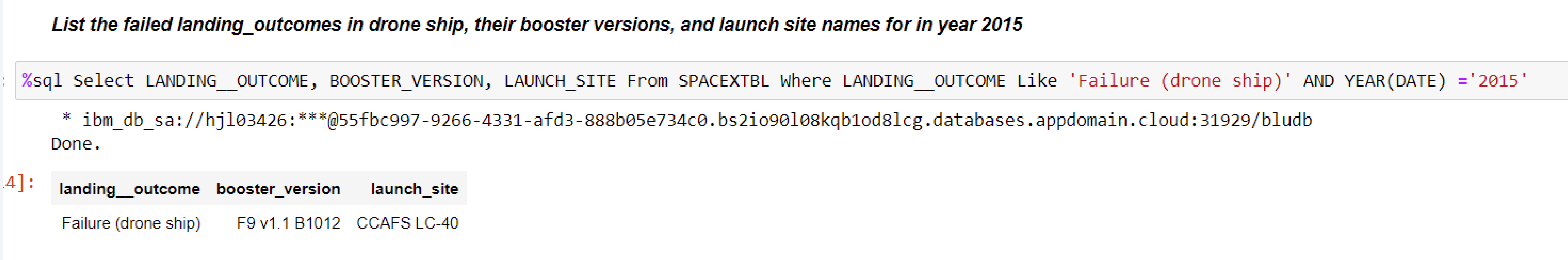
```
 * ibm_db_sa://hjl03426:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929
Done.
```

13]:

booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3

- List the names of the booster which have carried the maximum payload mass

2015 Launch Records



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

```
%sql Select LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE From SPACEXTBL Where LANDING__OUTCOME Like 'Failure (drone ship)' AND YEAR(DATE) ='2015'
```

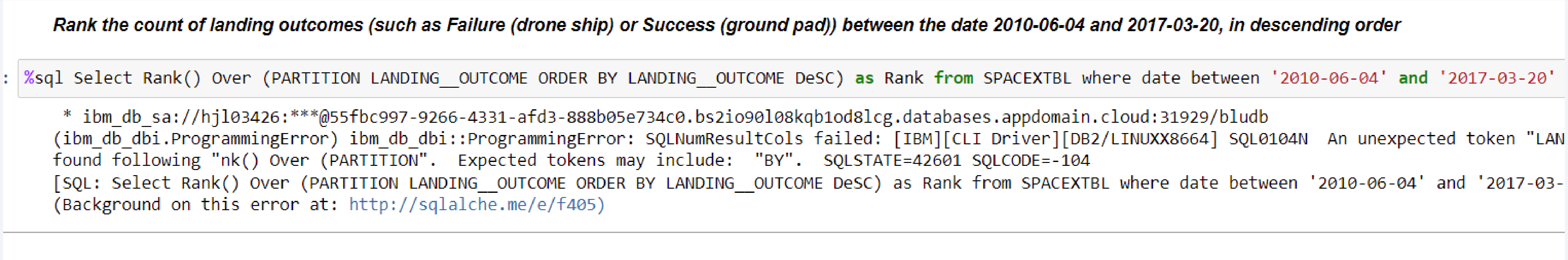
```
 * ibm_db_sa://hjl03426:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

4]:

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40

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

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

```
: %sql Select Rank() Over (PARTITION LANDING__OUTCOME ORDER BY LANDING__OUTCOME DeSC) as Rank from SPACEXTBL where date between '2010-06-04' and '2017-03-20'
```

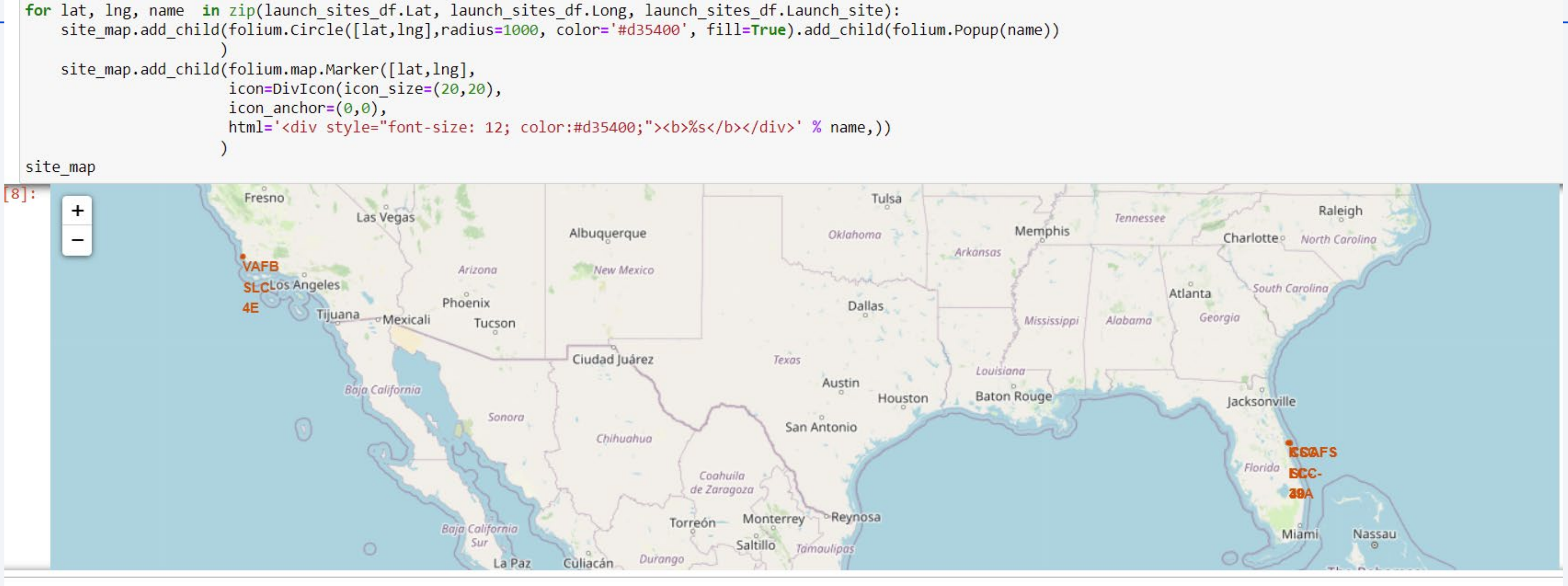
```
 * ibm_db_sa://hjl03426:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
(ibm_db_dbi.ProgrammingError) ibm_db_dbi::ProgrammingError: SQLNumResultCols failed: [IBM][CLI Driver][DB2/LINUXX8664] SQL0104N  An unexpected token "LAN
found following "nk() Over (PARTITION".  Expected tokens may include:  "BY".  SQLSTATE=42601 SQLCODE=-104
[SQL: Select Rank() Over (PARTITION LANDING__OUTCOME ORDER BY LANDING__OUTCOME DeSC) as Rank from SPACEXTBL where date between '2010-06-04' and '2017-03-
(Background on this error at: http://sqlalche.me/e/f405)
```

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

Section 3

Launch Sites Proximities Analysis

Folium Map showing the Launch Site Locations

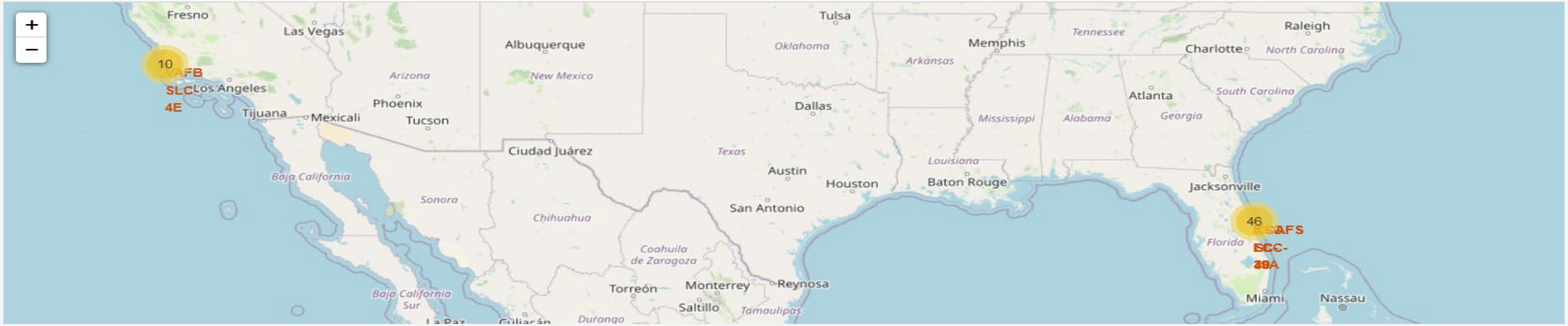


```
for lat, lng, name  in zip(launch_sites_df.Lat, launch_sites_df.Long, launch_sites_df.Launch_site):
    site_map.add_child(folium.Circle([lat,lng],radius=1000, color='#d35400', fill=True).add_child(folium.Popup(name))
                       )
    site_map.add_child(folium.map.Marker([lat,lng],
                        icon=DivIcon(icon_size=(20,20),
                        icon_anchor=(0,0),
                        html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % name,))
                       )
site_map
```

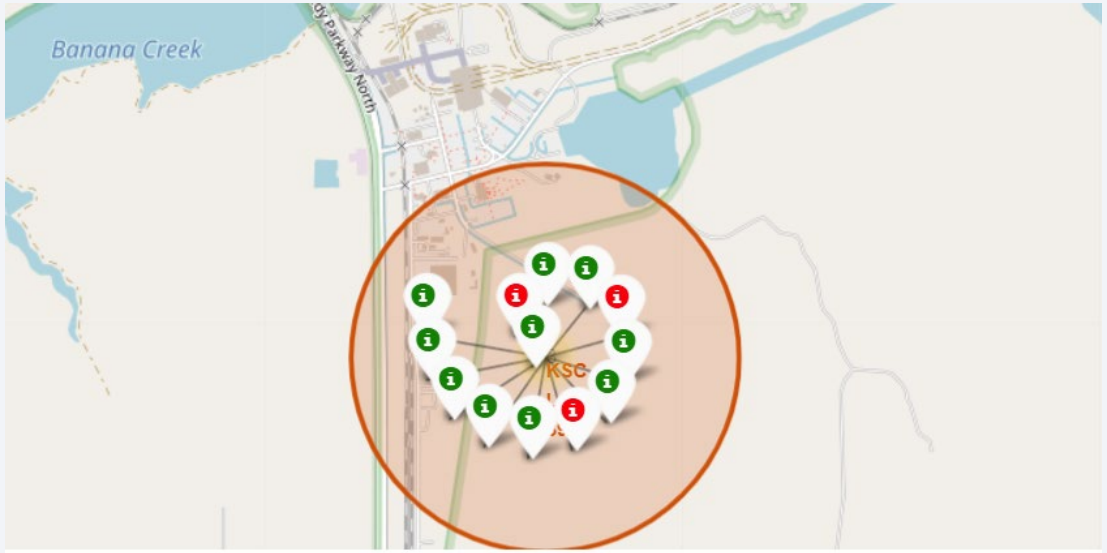
[8]:

The maps revealed that the launch site are all very close to a coastline of some sort.

Folium Map Showing Success/Failed launches per Launch Site

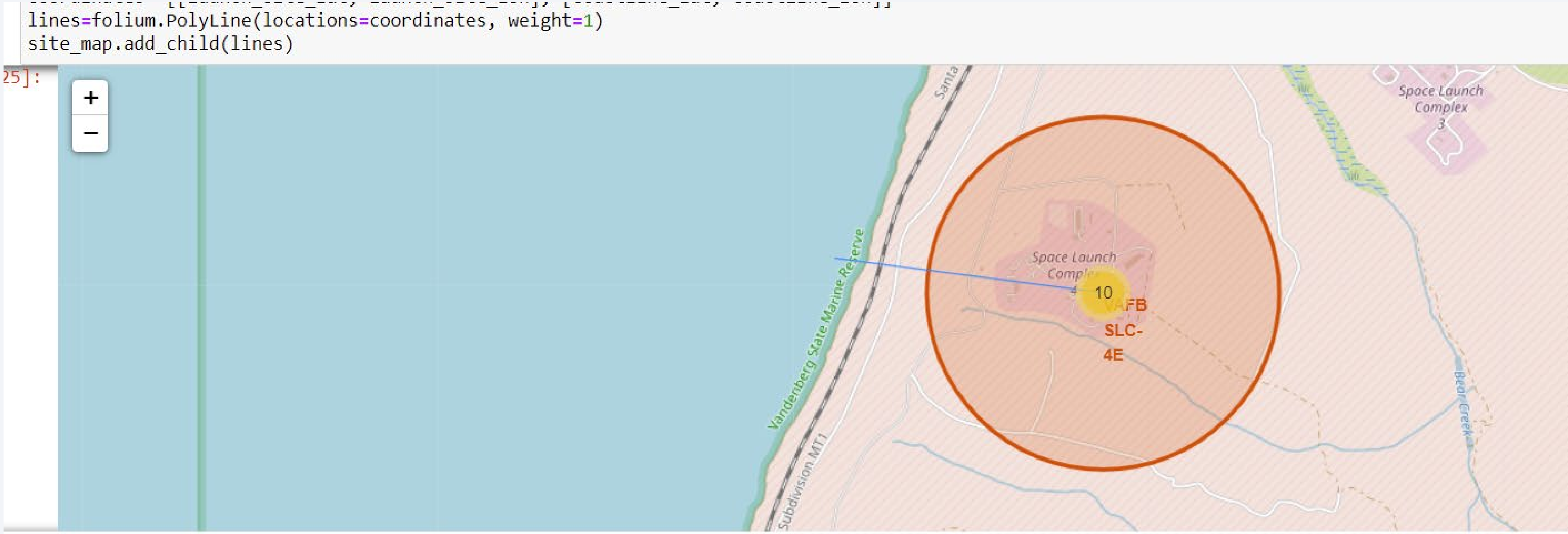


- The Maps shows the number of flights per Launch site with Markers indicating the success of the Flight. With, green being Successful and red being Failed.
- See the zoomed in Markers RSC LC 39A on the right.



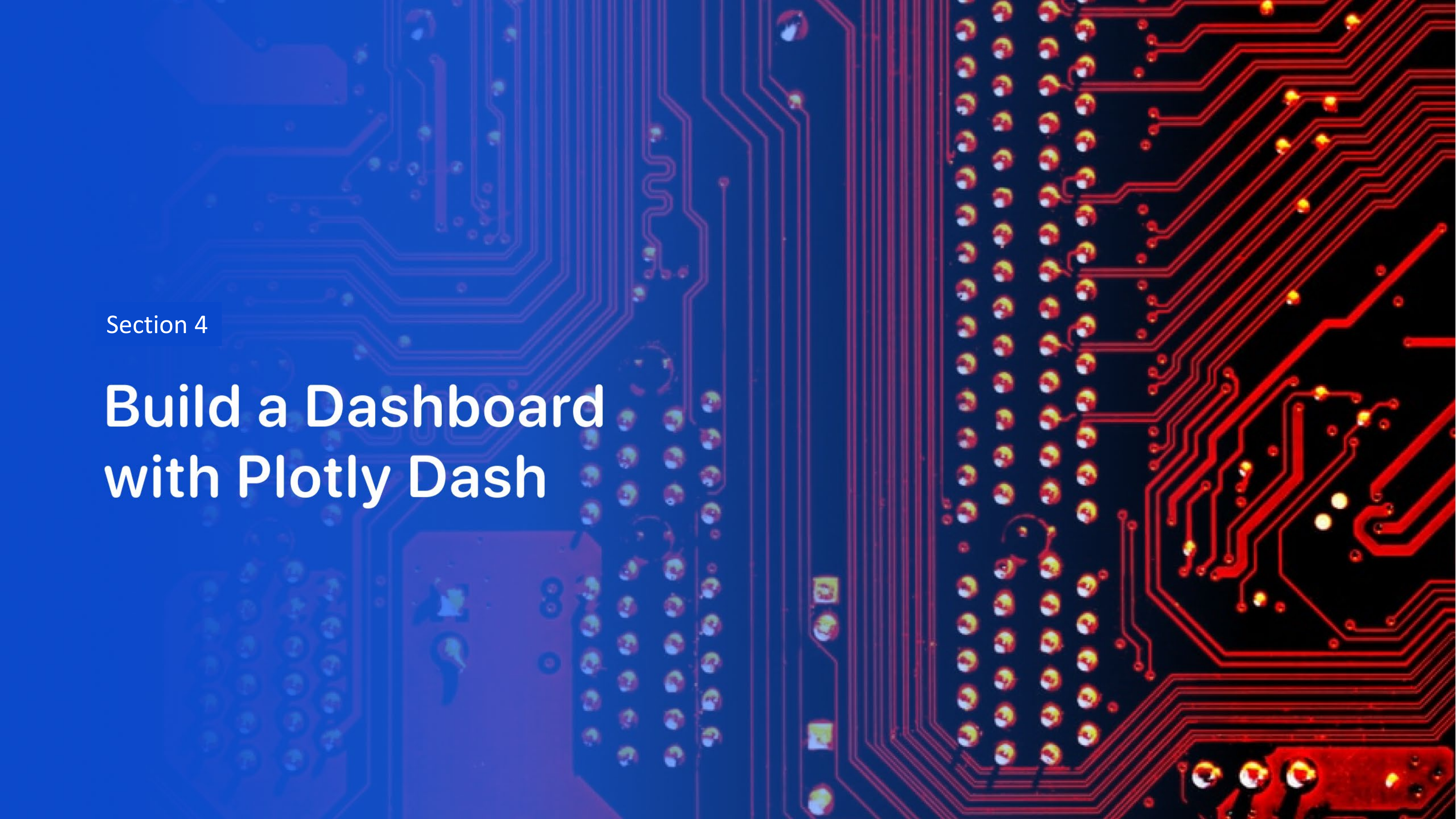
shots:

Folium Map Showing distances between a launch site to its proximities



```
lines=folium.PolyLine(locations=coordinates, weight=1)
site_map.add_child(lines)
```

- The Map shows that the site VAF BSLC-4E is in close proximity to the coast, a railway and a major road but is far form any cities or towns

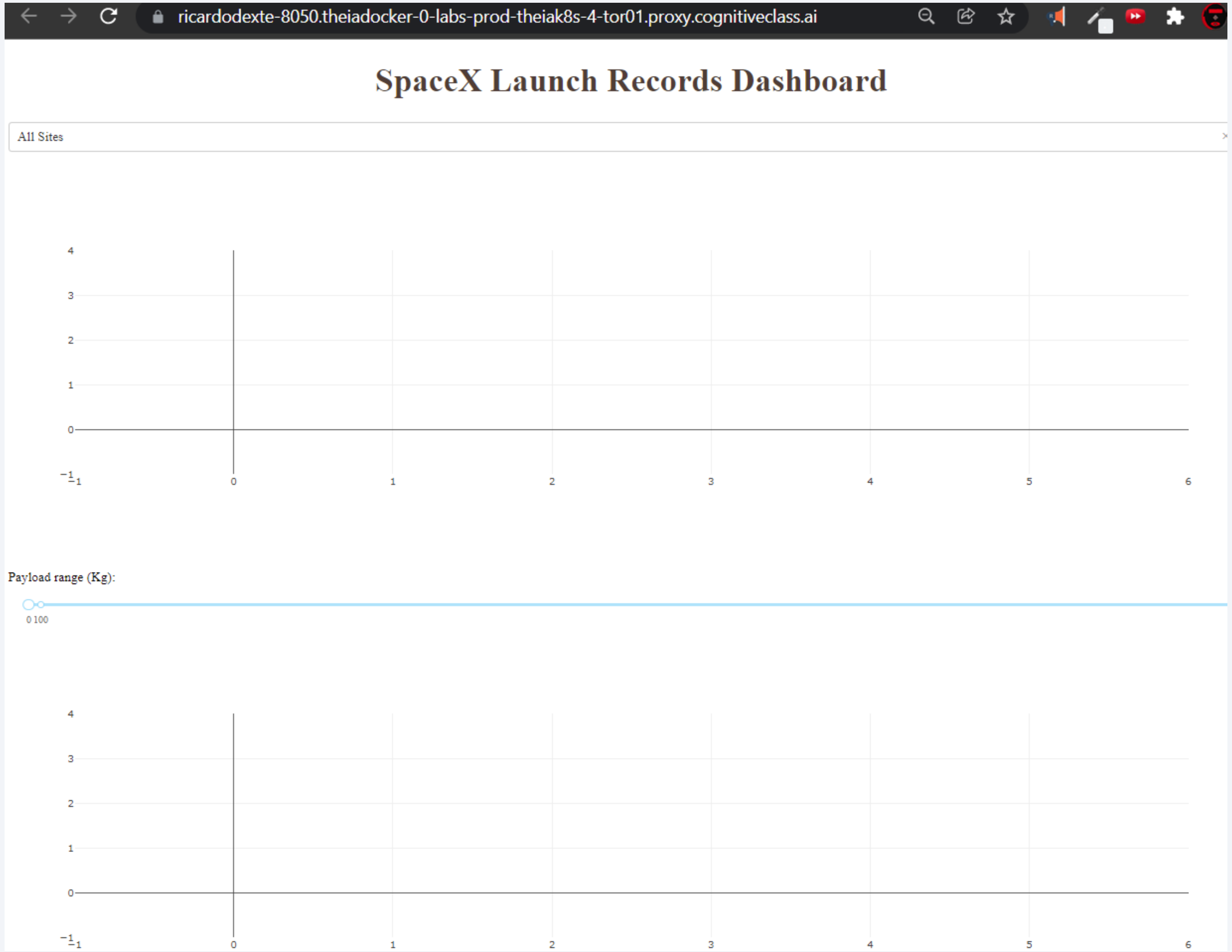


Section 4

Build a Dashboard with Plotly Dash

Dashboard Showing data for all Sites

- This screenshot was supposed to show the data for all sites but my plots were not displaying



ricardodexte-8050.theiadocker-0-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai

SpaceX Launch Records Dashboard

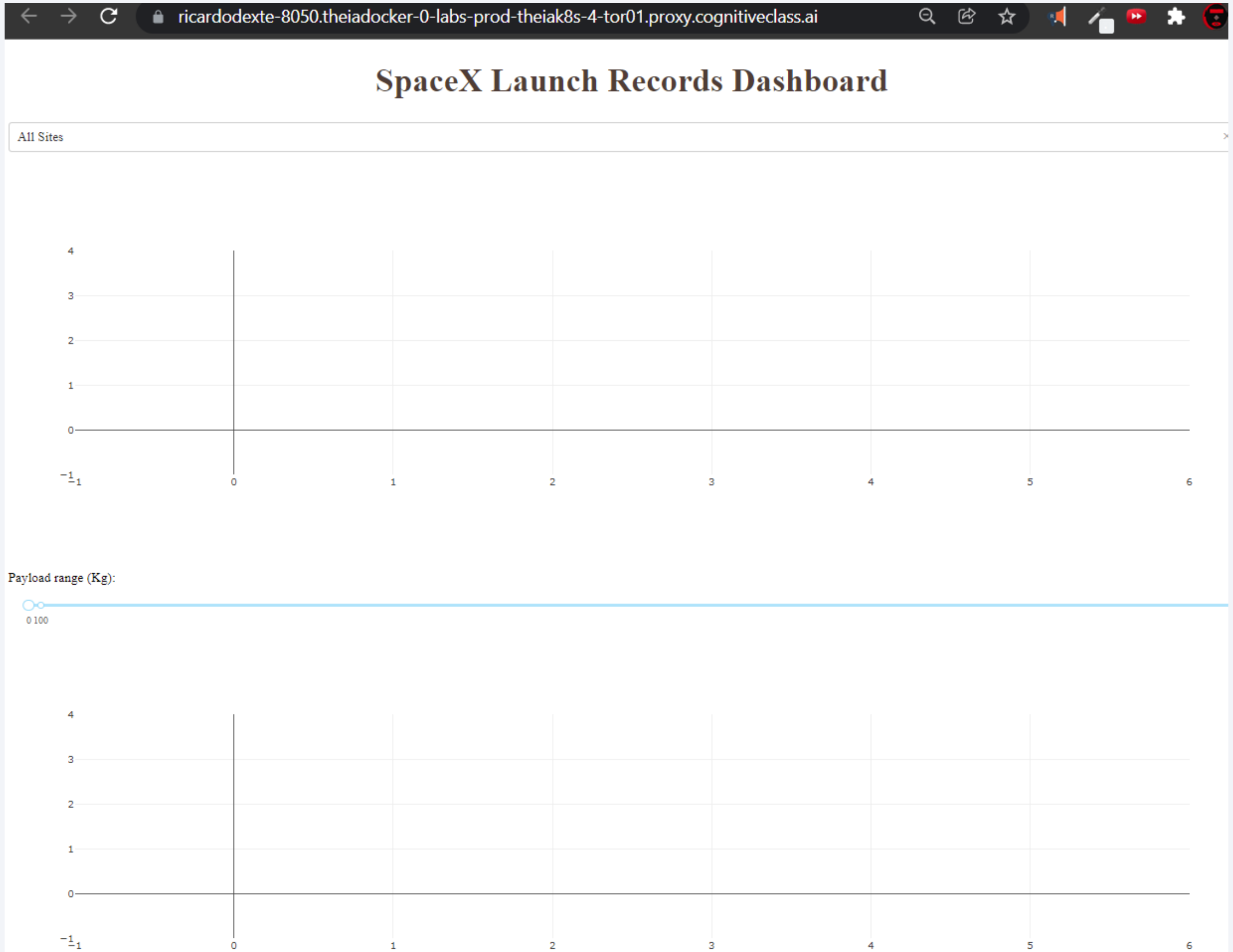
All Sites

Payload range (Kg):

0 100

Dashboard Showing launch site with highest launch success ratio

- Show the screenshot of the piechart for the launch site with highest launch success ratio



ricardodexte-8050.theiadocker-0-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai

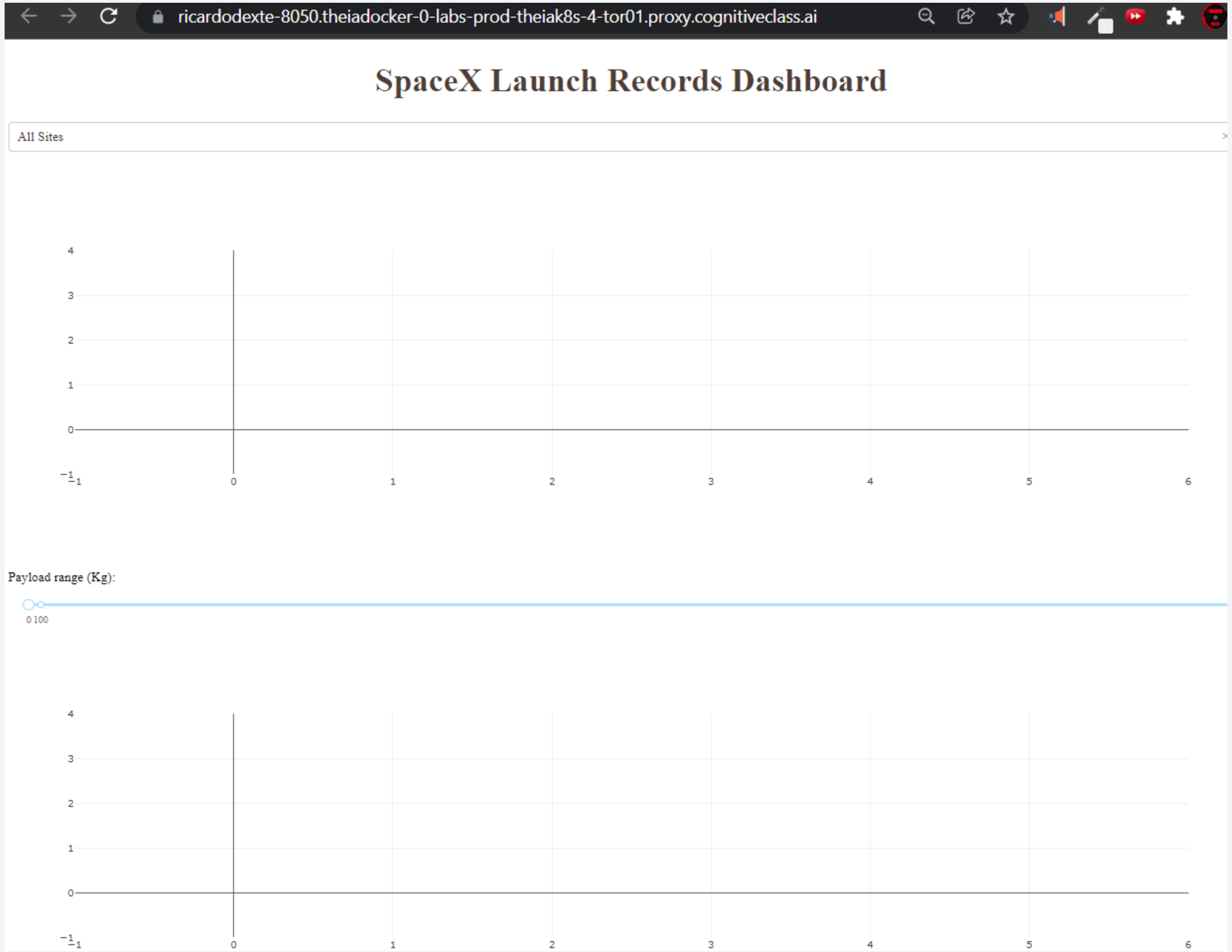
SpaceX Launch Records Dashboard

All Sites

Payload range (Kg):

Dashboard Showing Payload vs. Launch Outcome scatter plot

- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

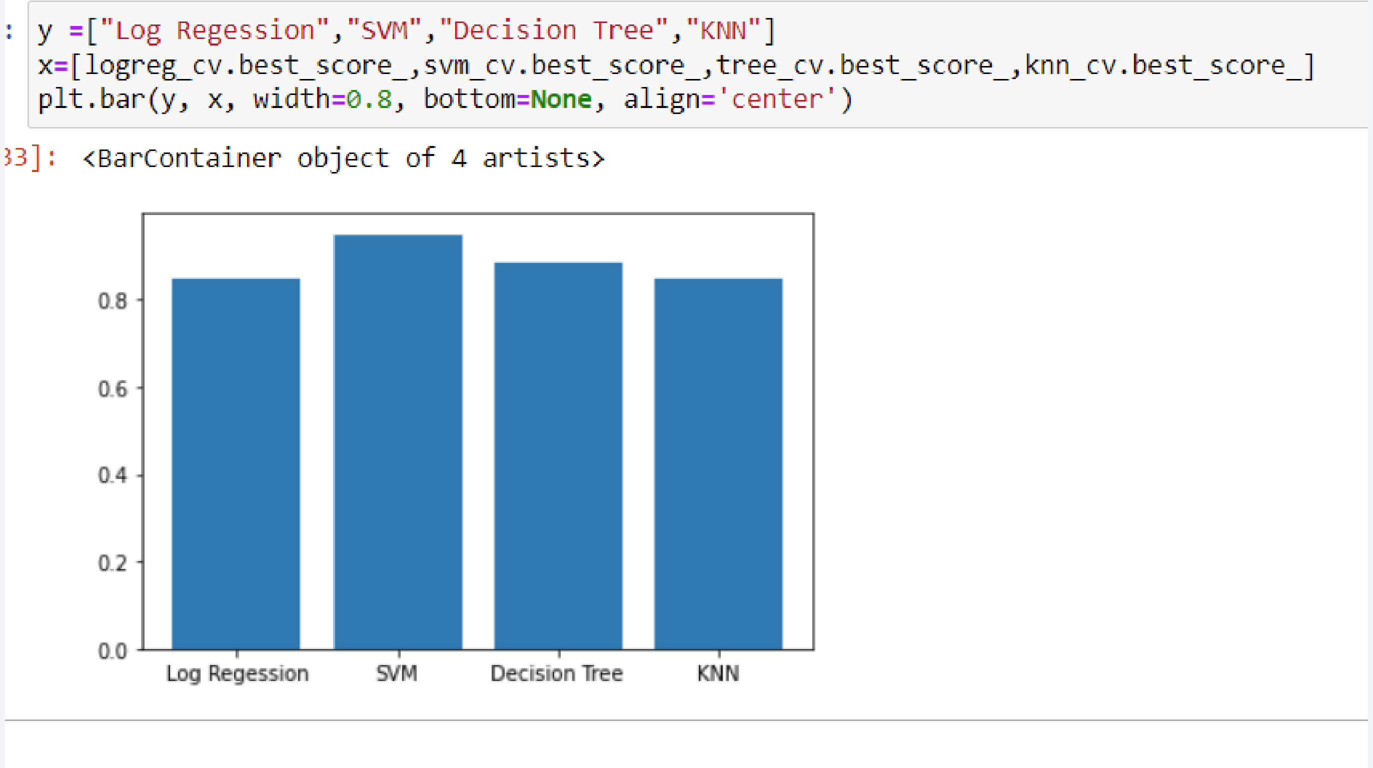


ricardodexte-8050.theiadocker-0-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai
SpaceX Launch Records Dashboard
All Sites
4
3
2
1
0
−1
−1
0
1
2
3
4
5
6
Payload range (Kg):
0 100
4
3
2
1
0
−1
−1
0
1
2
3
4
5
6

Section 5

Predictive Analysis (Classification)

Classification Accuracy

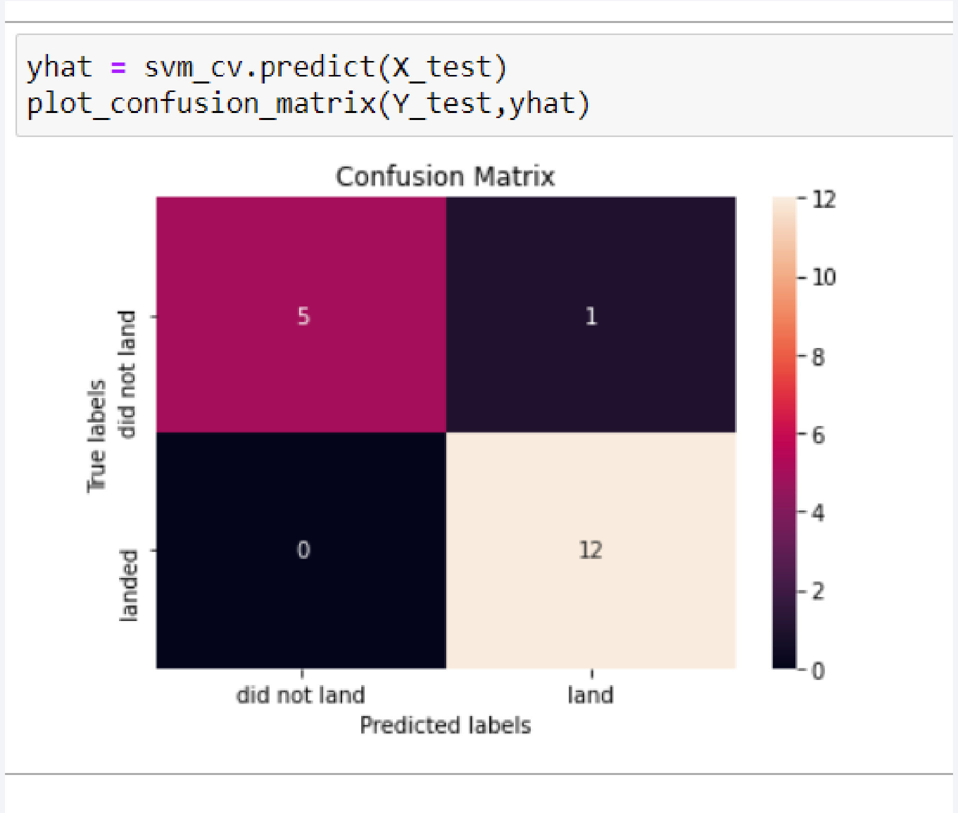


```
: y =["Log Regession","SVM","Decision Tree","KNN"]
  x=[logreg_cv.best_score_,svm_cv.best_score_,tree_cv.best_score_,knn_cv.best_score_]
  plt.bar(y, x, width=0.8, bottom=None, align='center')
```

```
33]:  <BarContainer object of 4 artists>
```

SVM Performed the Best with the following parameters: {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}

Confusion Matrix



```
yhat = svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```

Confusion matrix of the best performing model the SVM. As shown the model only had 1 False positive in the test set.

Conclusions

- In conclusion, the study has shown that it is possible to predict the success of the launch based on a variety of Launch Parameters.
- It was found that the best model for predictions was a Support Vector Machine which gave an accuracy of 95% using the following hyperparameters:

C: =1.0

gamma = 0.03162277660168379

Kernel = 'sigmoid'

Appendix

- All Relevant Notebooks can be accessed at the following github repo : HERE

Thank you!