The Battle for Space Domain

Danilo Rezende Teófilo

10 August 2022

IBM Developer

SKILLS NETWORK

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Conclusion

EXECUTIVE SUMMARY

- Summary of Methodologies
  - Data Collection
  - Data Wrangling
  - Exploratory Data Analysis
  - Interactive Visual Analytics
  - Predictive Analysis
- Summary of All Results
  - Exploratory Data Analysis with Visualization Result
  - Exploratory Data Analysis with SQL Result
  - Interactive Map with Folium Results
  - Plotly Dash Dashboard Results
  - Predictive Analysis Result

INTRODUCTION

Actually, SpaceX is the most successful space travel company in the world. This success can be explained in part by its cheaper costs in comparison with other companies that offer the same services.

One advantage in SpaceX's Falcon 9 rockets is the possibility in first stage reuse, what keeps SpaceX's lower costs. The success rate in Falcon 9 first stage landing is one of the main reasons to determine if it can be reused, but it's not the only one.

To make SpaceY competitive in comparison to SpaceX, we're going to analyze SpaceX data and discover what are the reasons which determine first state reuse rate. This analysis is fundamental to SpaceY development.

Let the battle begins!

METHODOLOGY

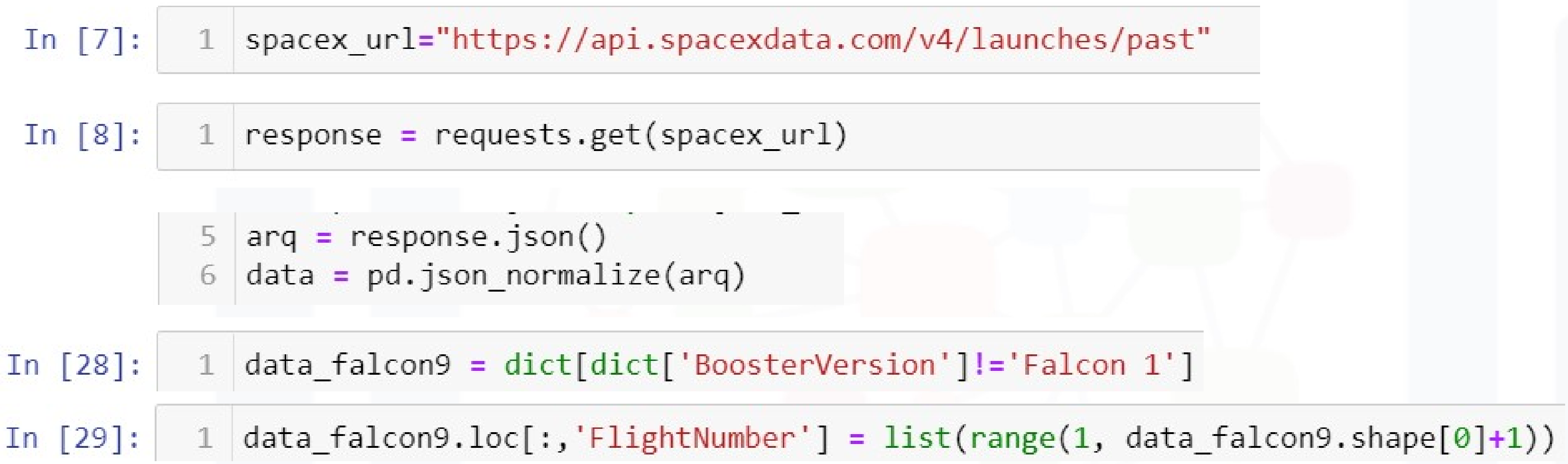
- Data Collection
  - Data Collection API
  - Data Collection with Web Scrapping
- Data Wrangling
- Exploratory Data Analysis
  - EDA with Visualization
  - EDA using SQL
- Interactive Visual Analytics
  - Interactive Visual Analytics with Folium
  - Interactive Dashboard with Plotly Dash
- Predictive Analysis

DATA COLLECTION

Data collection is the process of collecting, measuring and analyzing different types of information using a set of known validated techniques. All this information, when correctly treated and analyzed, can be a powerful input to make critical business decisions.

In this paper SpaceX information was gathered through two different ways: using an API (SpaceX REST API) and through Web Scrapping from Wikipedia. When using SpaceX REST API, the response was decoded with json(), turned into a pandas dataframe using json_normalize() and then treated. To collect data using web scrapping, Beautiful Soup was used to extract the information from HTML, which was converted into a dictionary and later to a dataframe.

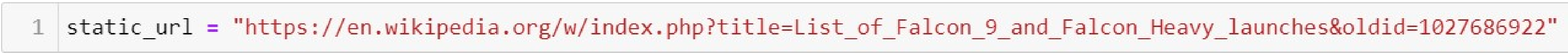
DATA COLLECTION WITH API



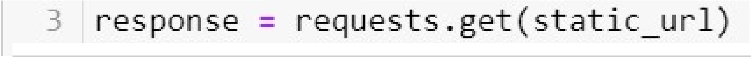
```
In [7]:  spacex_url="https://api.spacexdata.com/v4/launches/past"

In [8]:  response = requests.get(spacex_url)

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

In [28]: data_falcon9 = dict[dict['BoosterVersion']!='Falcon 1']

In [29]: data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
```

https://github.com/danrezt/Applied_Data_Science_Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

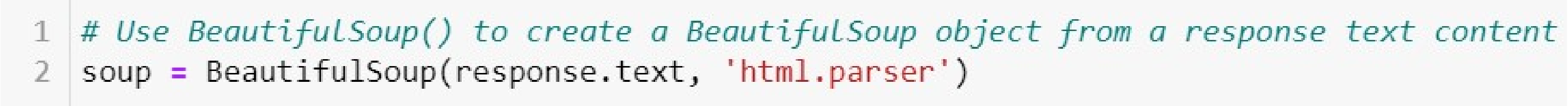
DATA COLLECTION WITH WEB SCRAPPING



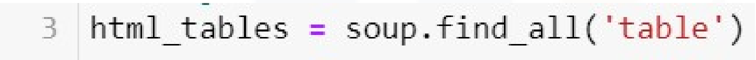
```python
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```



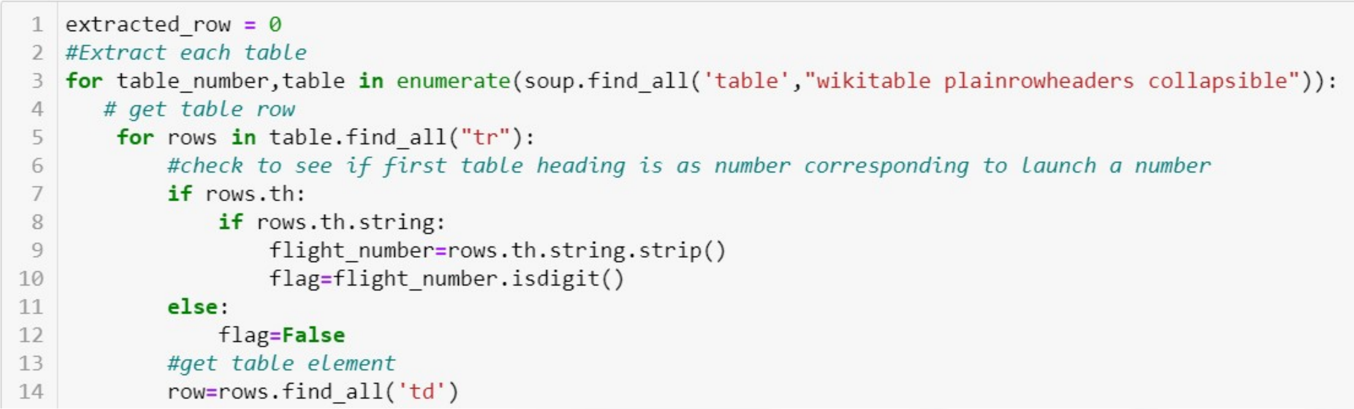
```python
response = requests.get(static_url)
```



```python
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'html.parser')
```



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



```python
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
```

https://github.com/danrezt/Applied_Data_Science_Capstone/blob/main/jupyter-labs-webscraping.ipynb

DATA WRANGLING

Data wrangling is the process of transforming and mapping data from one primitive and untreated data form into another format with the intent of making it more appropriate and valuable for a variety of downstream purposes such as analytics. The goal of data wrangling is to assure quality and useful data.

In this study, the SpaceX data gathered in the previous steps is treated through the following stages:

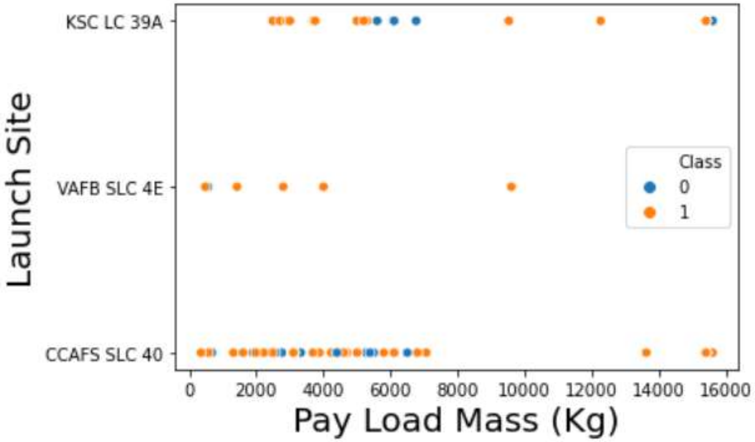
- To calculate the number of launches of each sites;
- To calculate the number and occurrence of each orbit;
- To calculate the number and occurrence of mission outcome per orbit type;
- To create a new binary attribute, related to the landing outcome success of each launch;

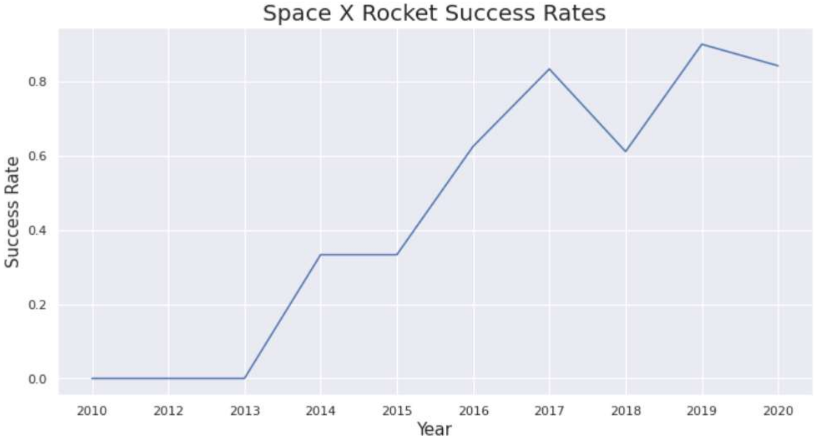
https://github.com/danrezt/Applied_Data_Science_Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA WITH DATA VISUALIZATION

The Exploratory Data Analysis is capable of making us understand the data and the relationships between all the gathered information.

Analyzing SpaceX data, it was found the influence in launch outcome through relationships between flight number and launch site, payload mass and launch site, success rate and orbit type, flight number and orbit type, payload mass and orbit type, launch success yearly trend.





https://github.com/danrezt/Applied_Data_Science_Capstone/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA USING SQL

SQL is a consolidated language to manage and manipulating databases. Using its queries and structures it's possible to do a powerful EDA.

In this study it was applied to achieve some insights into the data. Among them are the items below:

- The unique launch sites in the space mission;
- The total payload mass carried by boosters launched by NASA (CRS);
- The average payload mass carried by booster version F9 v1.1;
- The date of the first successful landing outcome in ground pad;
- The total number of successful and failure mission outcomes;

https://github.com/danrezt/Applied_Data_Science_Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

INTERACTIVE VISUAL ANALYSIS

To visualize the launch sites and the related launch data was used Folium, through which is possible to create a interactive map.

First, the coordinates of each launch site was used to draw circle markers in the map with the name of its names as labels. Then, launch outcomes info was used to draw markers (green – success / red – failure) in these circles.

Finally, it was calculated the distance of the launch sites to various landmark to find answer to questions like:

- How close are the launch sites to railways, highways, cities and coastlines?

It was used too Ploty Dash to build an interactive dashboard, which allow the user to create graphs and analyze data dynamically, through a colorful and interesting way. In the Results Section is possible to see pie charts and scatter graphs created with it.

https://github.com/danrezt/Applied_Data_Science_Capstone/blob/main/lab_jupyter_launch_site_location.ipynb

https://github.com/danrezt/Applied_Data_Science_Capstone/blob/main/spacex_dash_app.py

PREDICTIVE ANALYSIS

Predictive analysis is the area in data science where the data basis is used to create models that try to explain the relationship between dependent and independent variables. There are different techniques to build those models, but the process is similar to all them.

The data is loaded, then treated, later divided into training and test sets. A machine learning technique is chosen to build the model with a set of parameters, and fitted to the data training set.

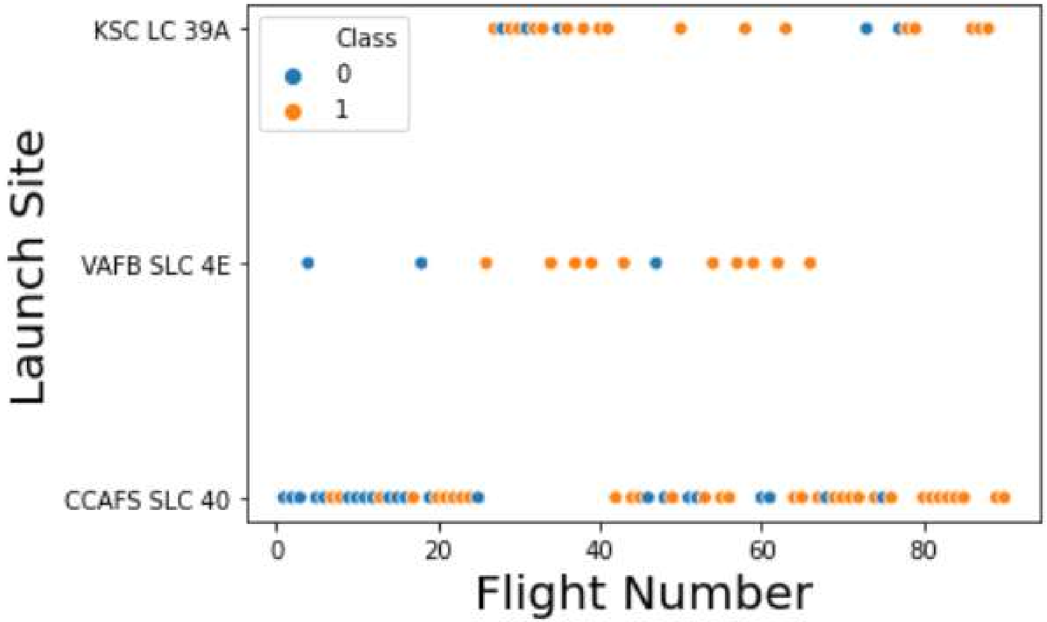
In this paper, some techniques are tested, evaluated and compared. The models are improved through feature engineering and algorithm tuning. The model with the best accuracy score is the one which has the best performance.

https://github.com/danrezt/Applied_Data_Science_Capstone/blob/main/lab_jupyter_launch_site_location.ipynb

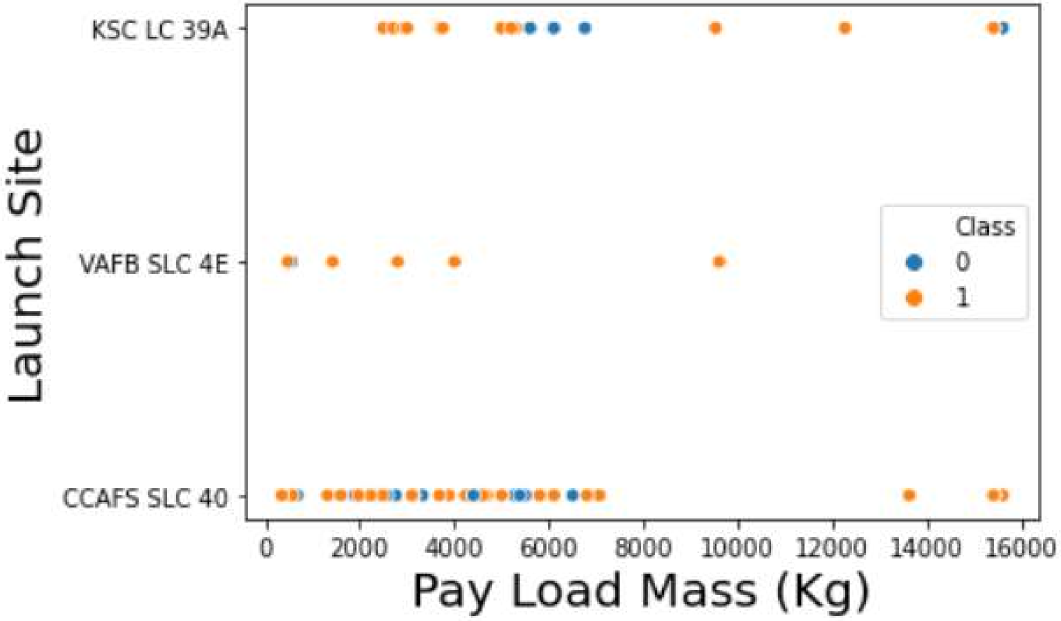
RESULTS

EXPLORATORY DATA ANALYSIS RESULTS

Flight Number x Launch Site

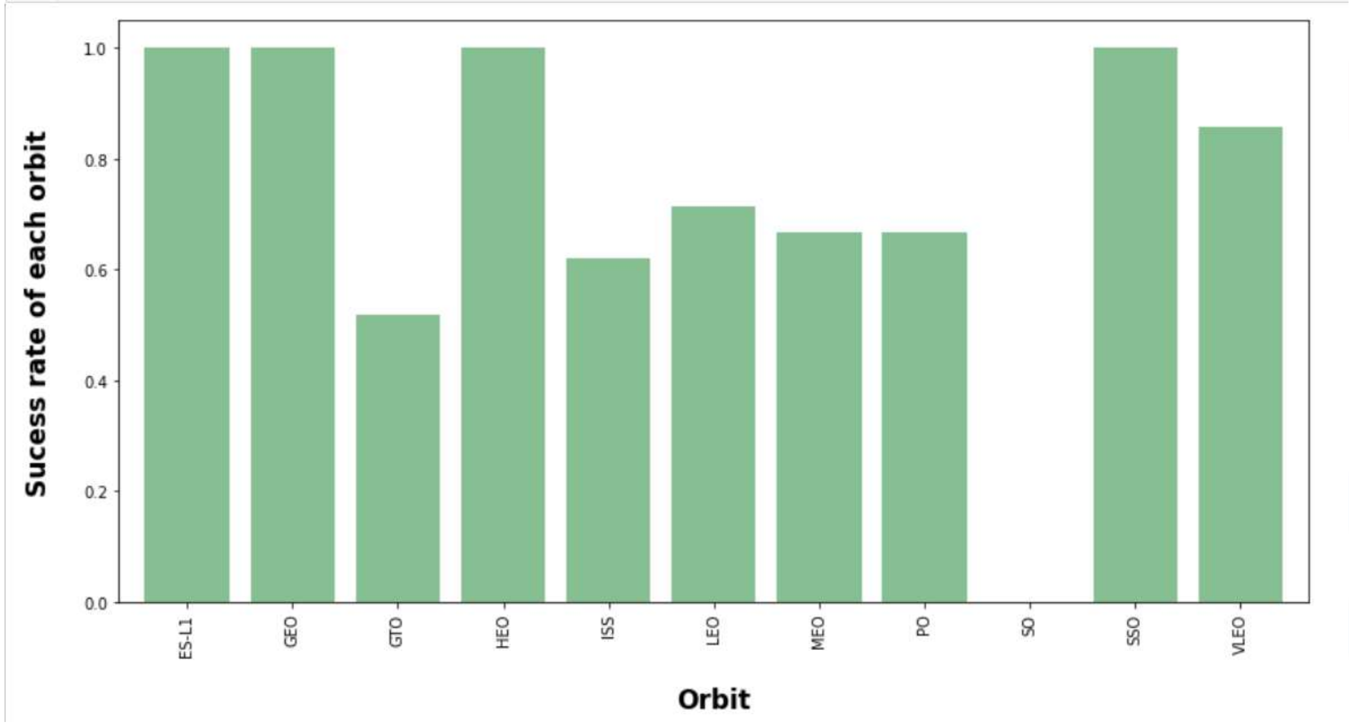


Launch Site x Payload Mass



EXPLORATORY DATA ANALYSIS RESULTS

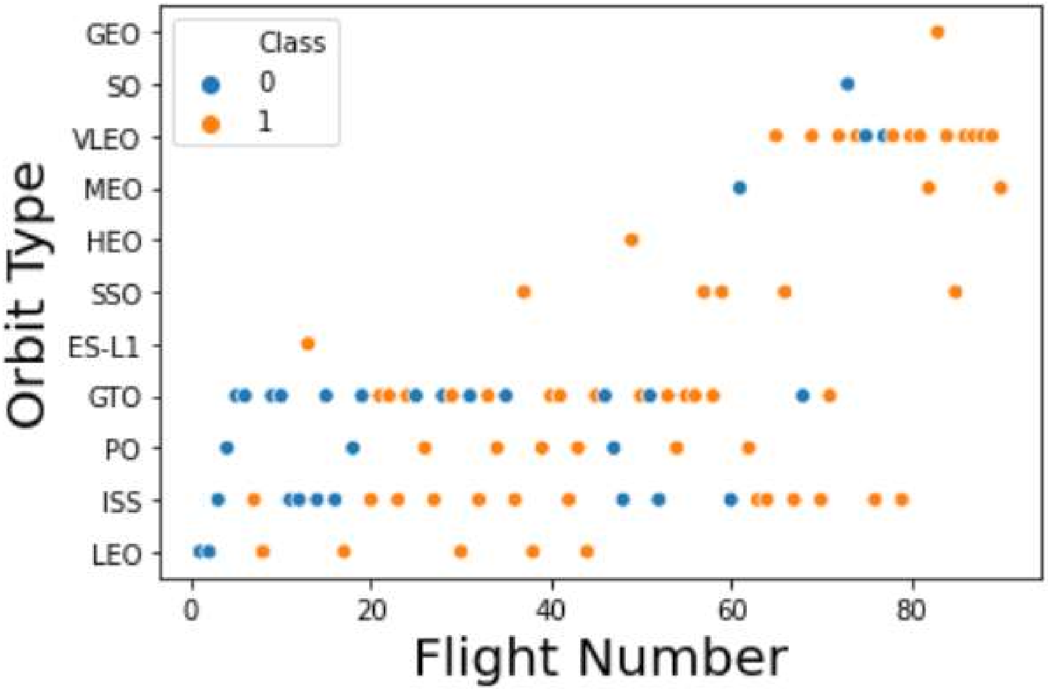
Success Rate x Orbit Type



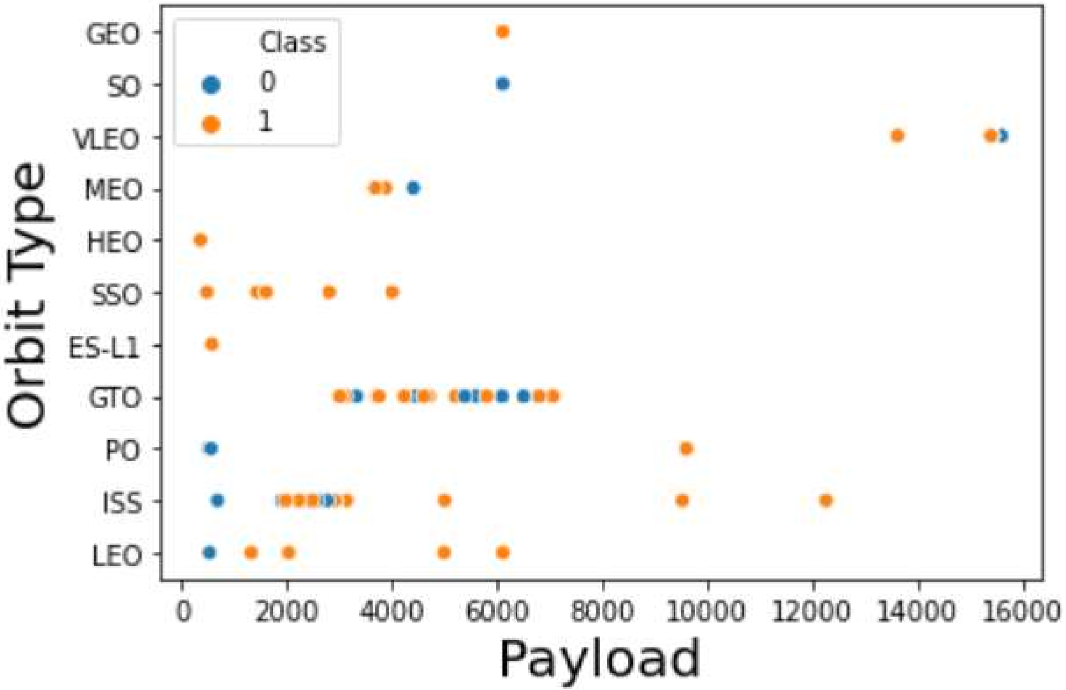
- Launch sites with more flights have a greater success rate;
- Between a payload mass of 4000 and 6000 kg, the fail rate is higher;
- The most successful orbits are ES-L1, GEO, HEO and SSO.

EXPLORATORY DATA ANALYSIS RESULTS

Flight Number x Orbit Type

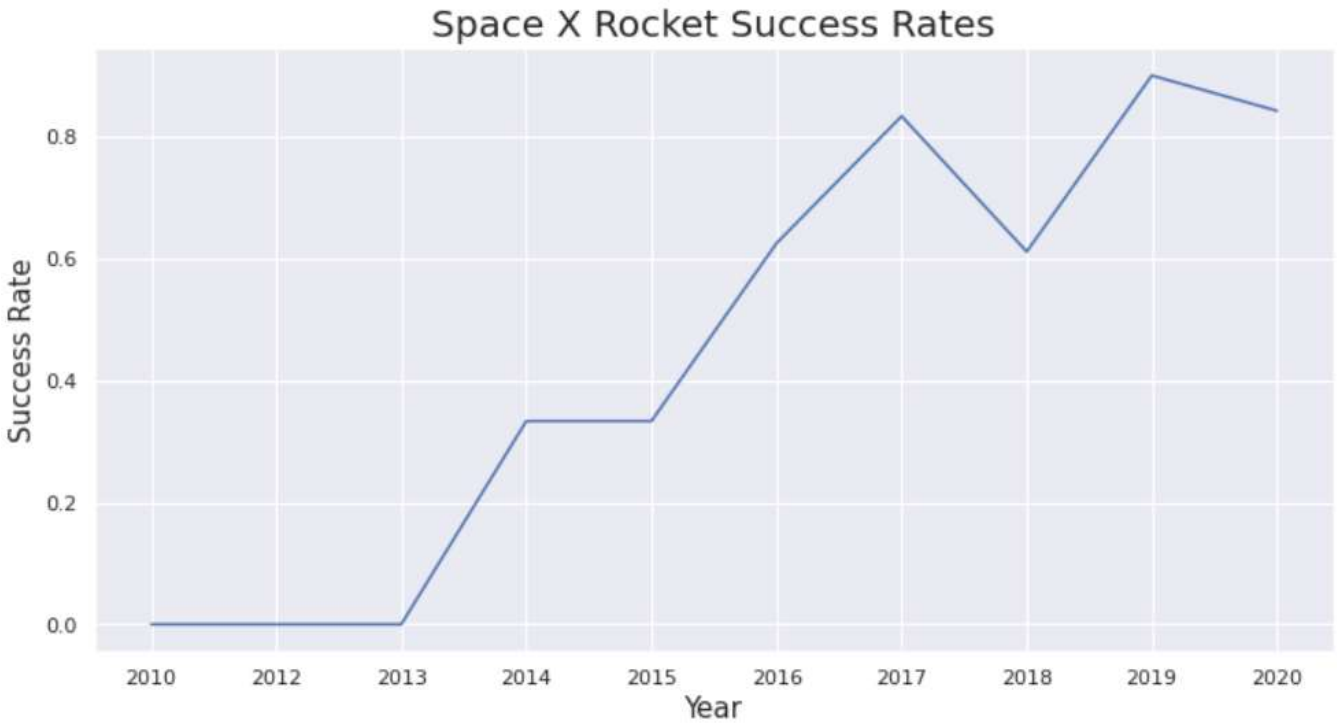


Launch Site x Payload Mass



EXPLORATORY DATA ANALYSIS RESULTS

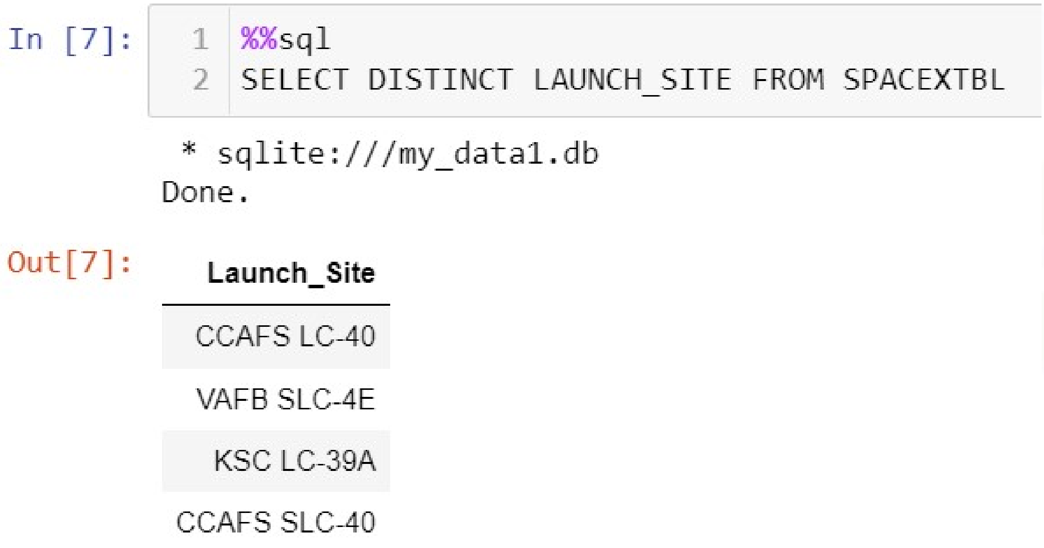
Launch Success Yearly Trend



- In GTO and ISS orbits apparently there’s no relation between success rate and flight number;
- Since 2013 launch success rate has been increasing quickly. Only in 2018 it was lower than in the year before.

EXPLORATORY DATA ANALYSIS RESULTS

Launch Site Names



```
In [7]:
%%sql
SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL
```

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

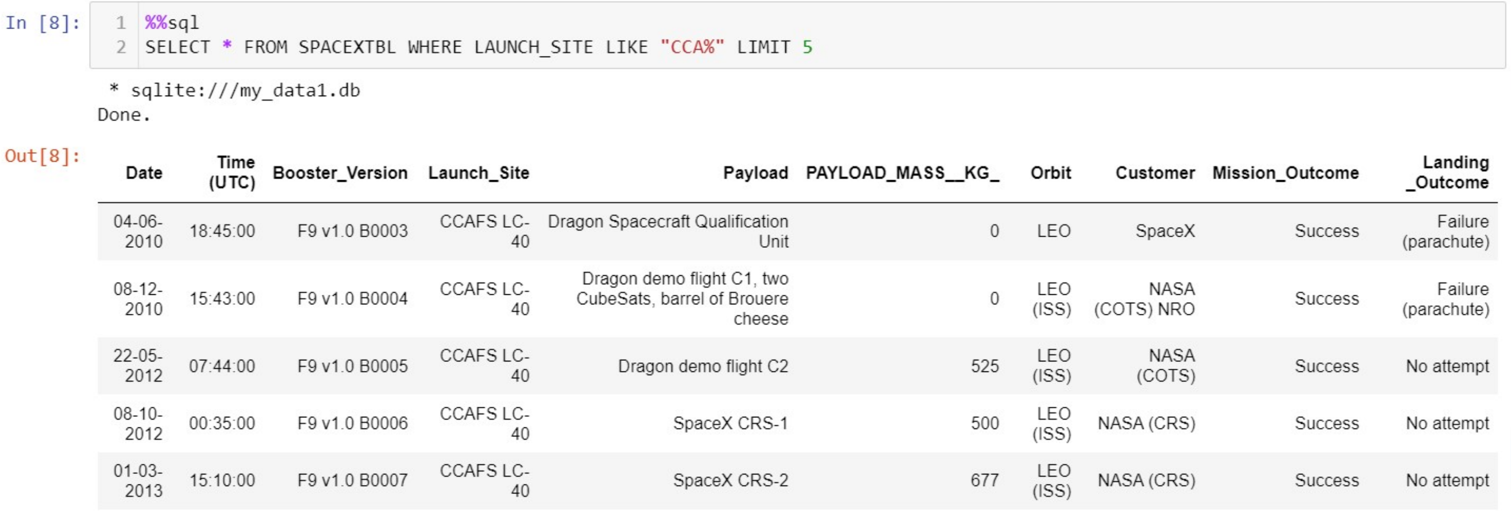
Out[7]:

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

- To display all the launch site names in the space mission it was used the key word DISTINCT in the SQL query

EXPLORATORY DATA ANALYSIS RESULTS

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



In [8]:

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

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

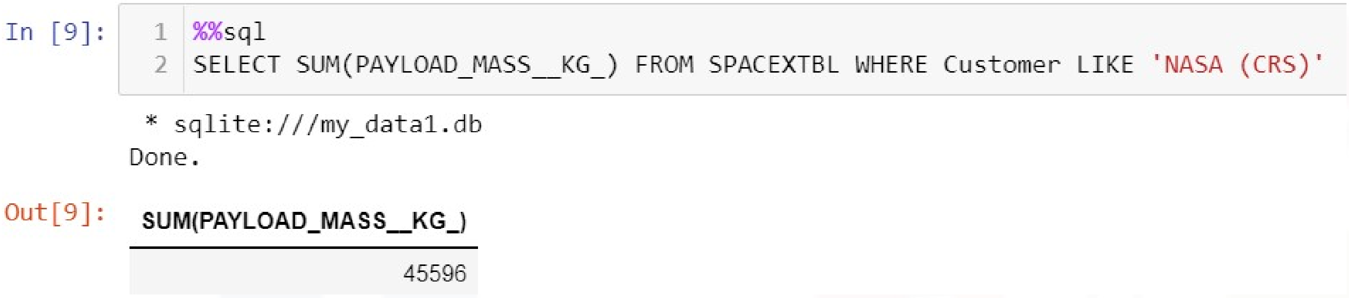
Out[8]:

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

- To display only 5 records where launch sites begin with the string ‘CCA’ it was used the key word LIMIT

EXPLORATORY DATA ANALYSIS RESULTS

Display total payload mass carried by boosters launched by NASA



In [9]:

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer LIKE 'NASA (CRS)'
```

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

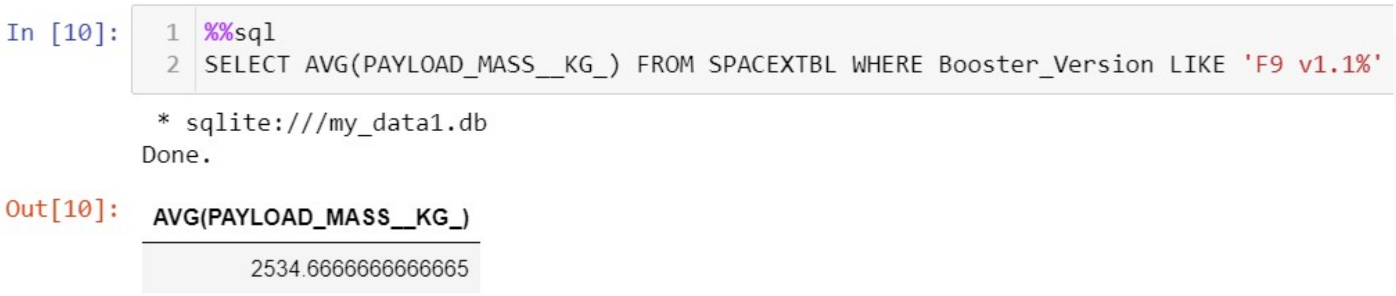
Out[9]:

SUM(PAYLOAD_MASS__KG_)
45596

- To display total payload mass carried by boosters launched by NASA it was used the key word SUM

EXPLORATORY DATA ANALYSIS RESULTS

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

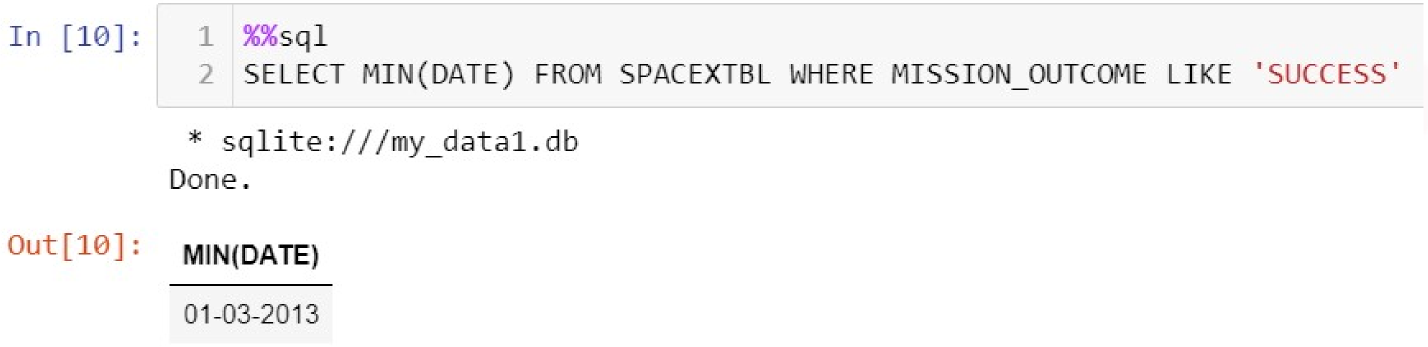


```
In [10]:  %%sql
          SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.1%'

 * sqlite:///my_data1.db
Done.

Out[10]:  AVG(PAYLOAD_MASS__KG_)
                2534.6666666666665
```

- To display average payload mass carried by booster version F9 v1.1 it was used the key word AVG

EXPLORATORY DATA ANALYSIS RESULTS

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



```
In [10]:
%%sql
SELECT MIN(DATE) FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'SUCCESS'
```

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

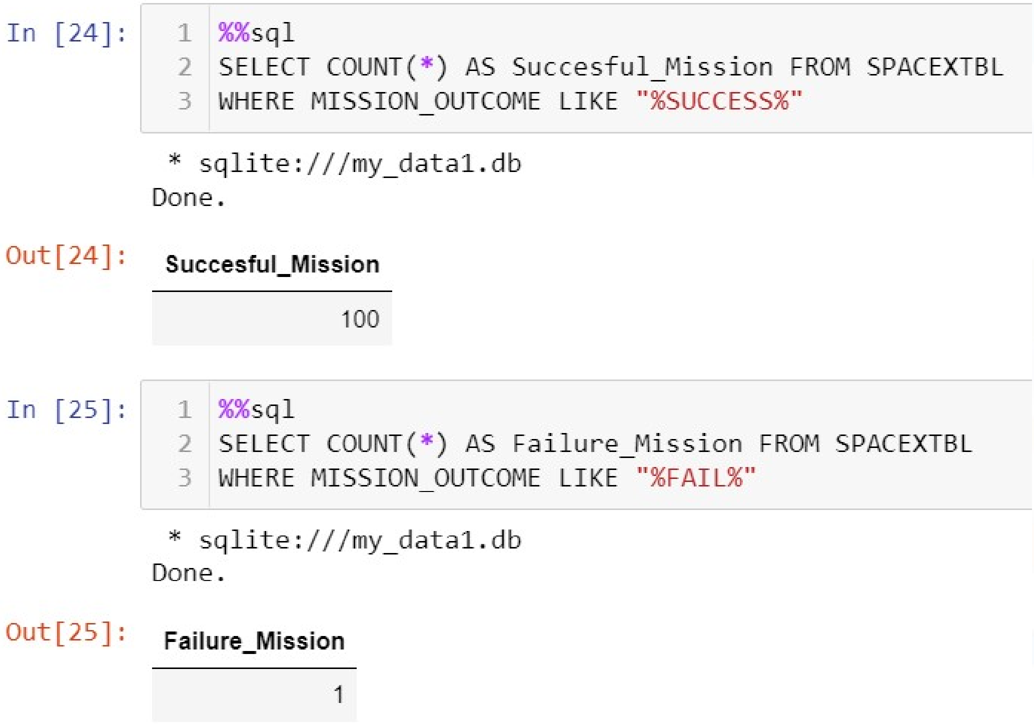
Out[10]:

MIN(DATE)
01-03-2013

- To list the date when the first successful landing outcome in ground pad was achieved it was used the key word MIN

EXPLORATORY DATA ANALYSIS RESULTS

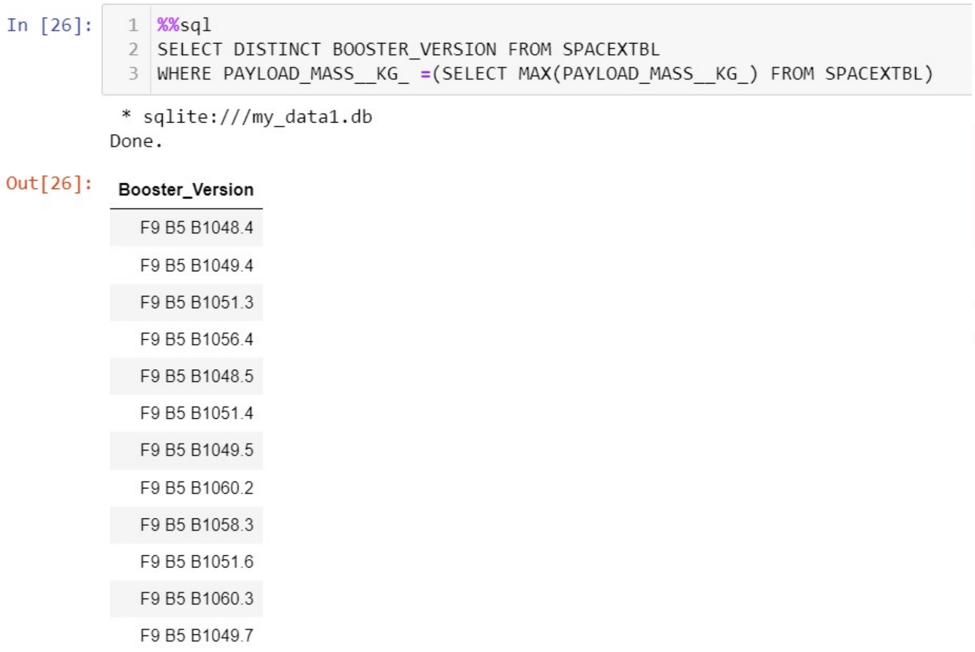
List the total number of successful and failure mission outcomes



```
In [24]:
%%sql
SELECT COUNT(*) AS Succesful_Mission FROM SPACEXTBL
WHERE MISSION_OUTCOME LIKE "%SUCCESS%"

 * sqlite:///my_data1.db
Done.

Out[24]:
Succesful_Mission
              100

In [25]:
%%sql
SELECT COUNT(*) AS Failure_Mission FROM SPACEXTBL
WHERE MISSION_OUTCOME LIKE "%FAIL%"

 * sqlite:///my_data1.db
Done.

Out[25]:
Failure_Mission
              1
```

EXPLORATORY DATA ANALYSIS RESULTS

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



In [26]:

```
%%sql
SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

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

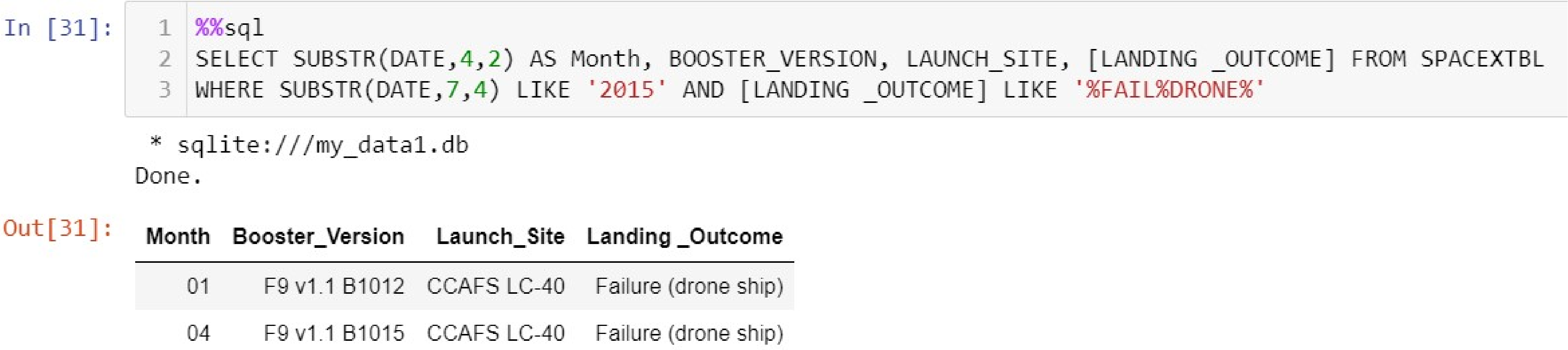
Out[26]:

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

- For this task it was used a subquery with the key word MAX

EXPLORATORY DATA ANALYSIS RESULTS

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



In [31]:

```
%%sql
SELECT SUBSTR(DATE,4,2) AS Month, BOOSTER_VERSION, LAUNCH_SITE, [LANDING _OUTCOME] FROM SPACEXTBL
WHERE SUBSTR(DATE,7,4) LIKE '2015' AND [LANDING _OUTCOME] LIKE '%FAIL%DRONE%'
```

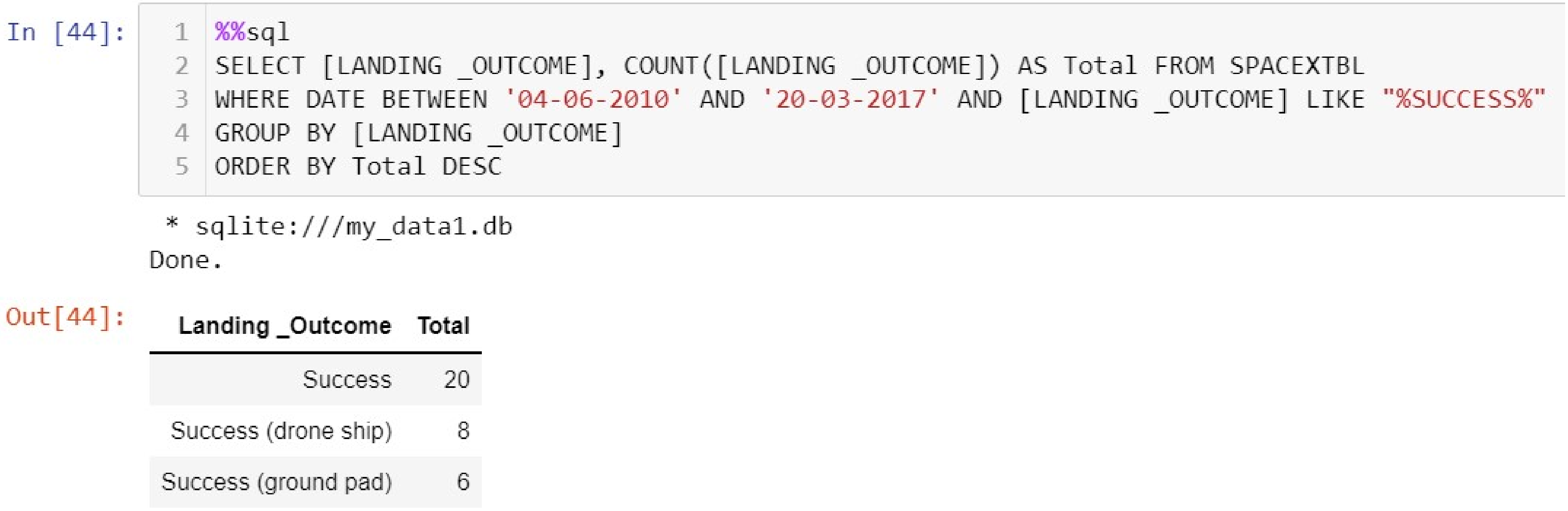
```
 * sqlite:///my_data1.db
Done.
```

Out[31]:

Month	Booster_Version	Launch_Site	Landing _Outcome
01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

EXPLORATORY DATA ANALYSIS RESULTS

Rank the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order



In [44]:

```
%%sql
SELECT [LANDING _OUTCOME], COUNT([LANDING _OUTCOME]) AS Total FROM SPACEXTBL
WHERE DATE BETWEEN '04-06-2010' AND '20-03-2017' AND [LANDING _OUTCOME] LIKE "%SUCCESS%"
GROUP BY [LANDING _OUTCOME]
ORDER BY Total DESC
```

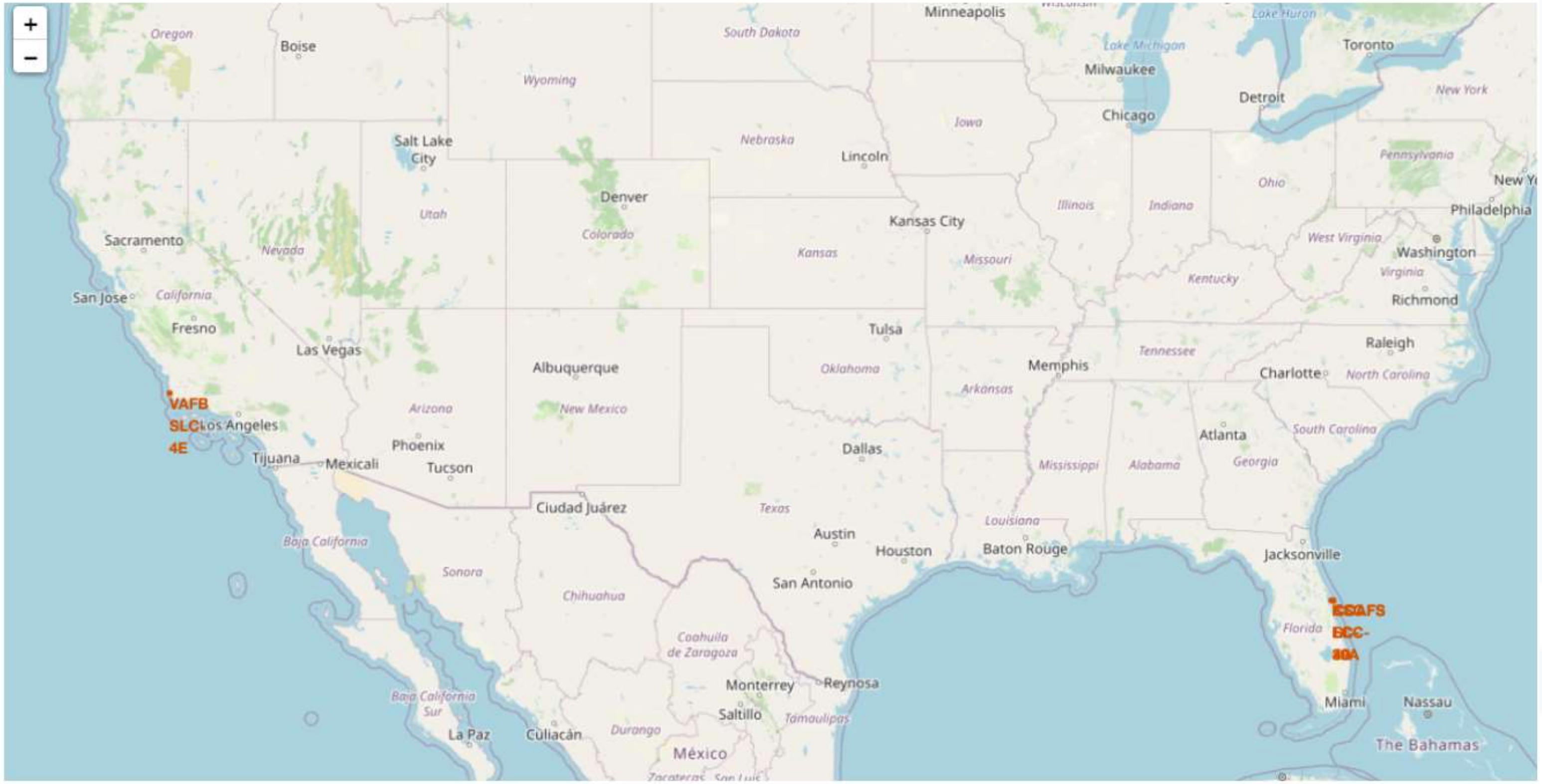
```
 * sqlite:///my_data1.db
Done.
```

Out[44]:

Landing _Outcome	Total
Success	20
Success (drone ship)	8
Success (ground pad)	6

INTERACTIVE VISUAL ANALYSIS RESULTS

Mark all the launch sites on a map

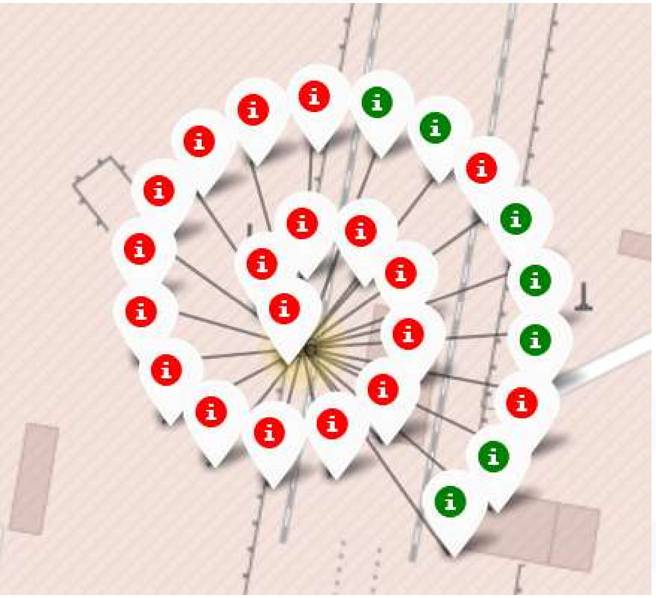


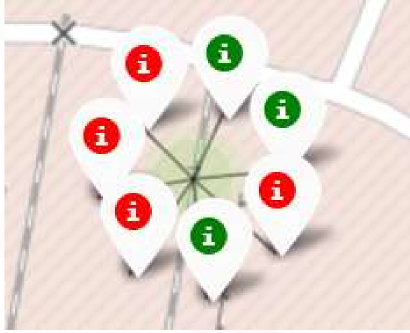
- It’s possible to see that the launch sites are concentrated in areas in the US west and east coasts

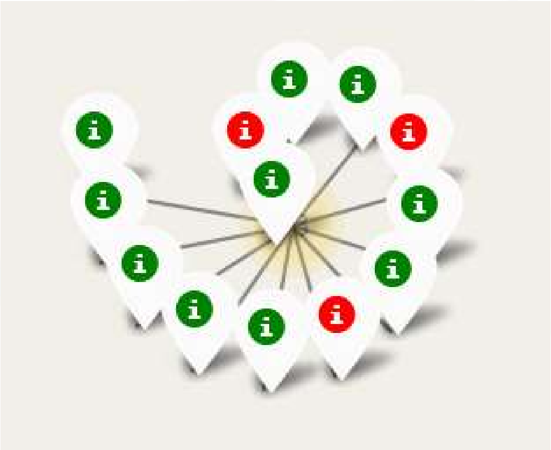
INTERACTIVE VISUAL ANALYSIS RESULTS

Mark the success/failed launches for each site on the map

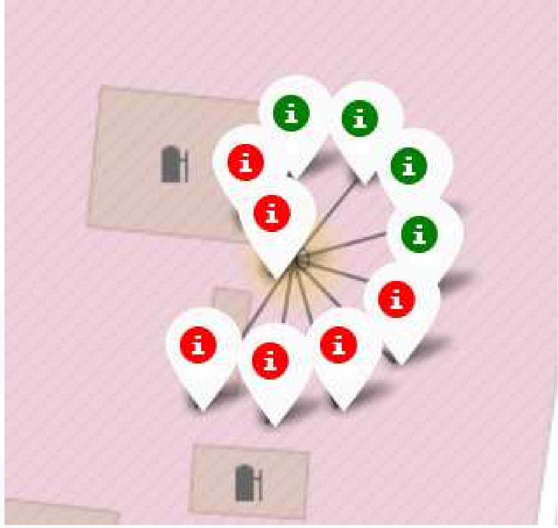
Florida Launch Sites





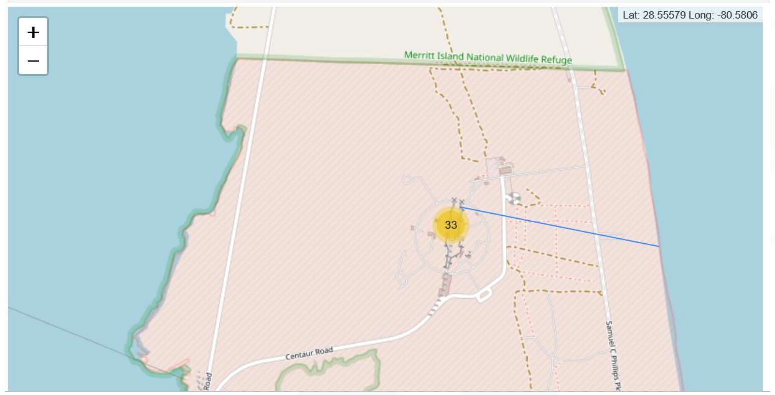


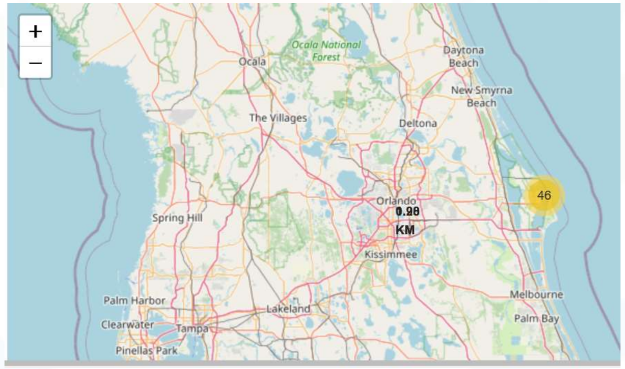
California Launch Sites



INTERACTIVE VISUAL ANALYSIS RESULTS

Calculate the distances between launch sites and its proximities


+
−
Lat: 28.55579 Long: -80.5806
Merritt Island National Wildlife Refuge
33
Centaur Road
Samuel C Phillips Pkwy

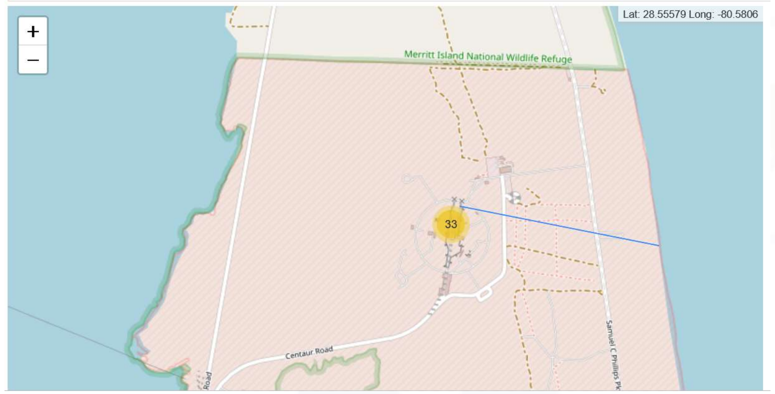


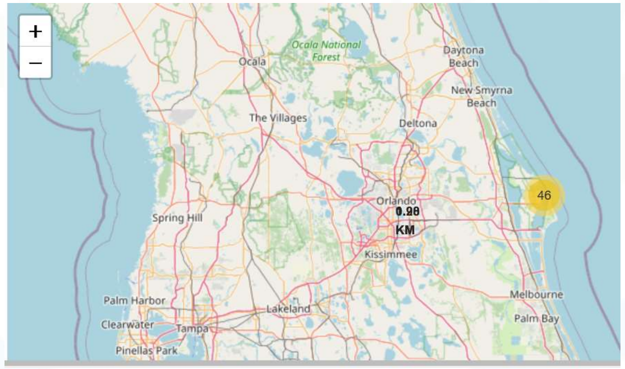
It was calculated the distance between the launch sites and some landmarks.

- To the coast, the distance found was 0,90km
- To a near city, the distance found was 0,98km

INTERACTIVE VISUAL ANALYSIS RESULTS

Calculate the distances between launch sites and its proximities


Lat: 28.55579 Long: -80.5806
+
−
Merritt Island National Wildlife Refuge
33
Centaur Road
Samuel C Phillips Pkwy

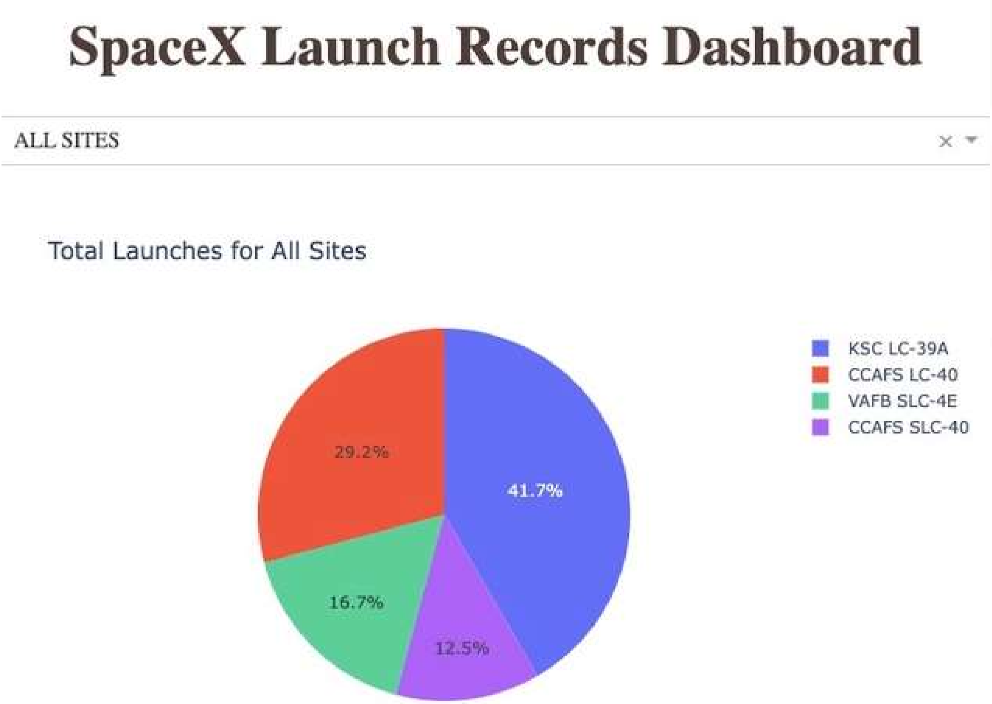


It was calculated the distance between the launch sites and some landmarks.

- To the coast, the distance found was 0,90km
- To a near city, the distance found was 0,98km

BUILD A DASHBOARD WITH PLOTLY DASH

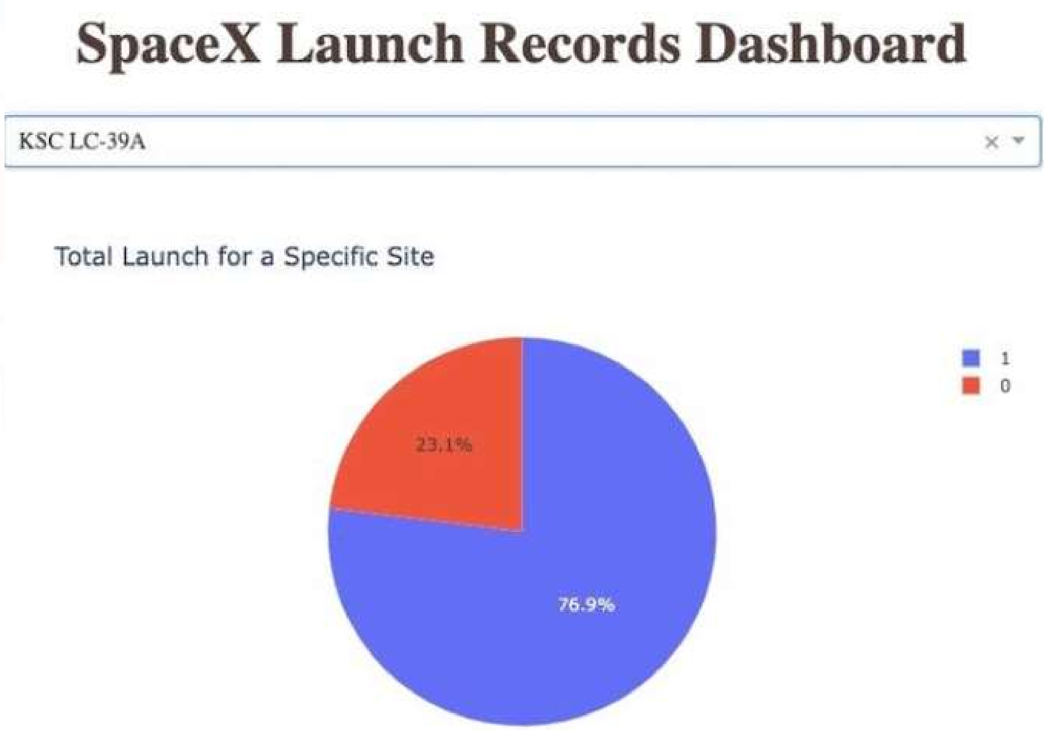
Pie chart showing the success rate by launch site



SpaceX Launch Records Dashboard

ALL SITES

Pie chart showing the launch with the highest success rate



SpaceX Launch Records Dashboard

KSC LC-39A

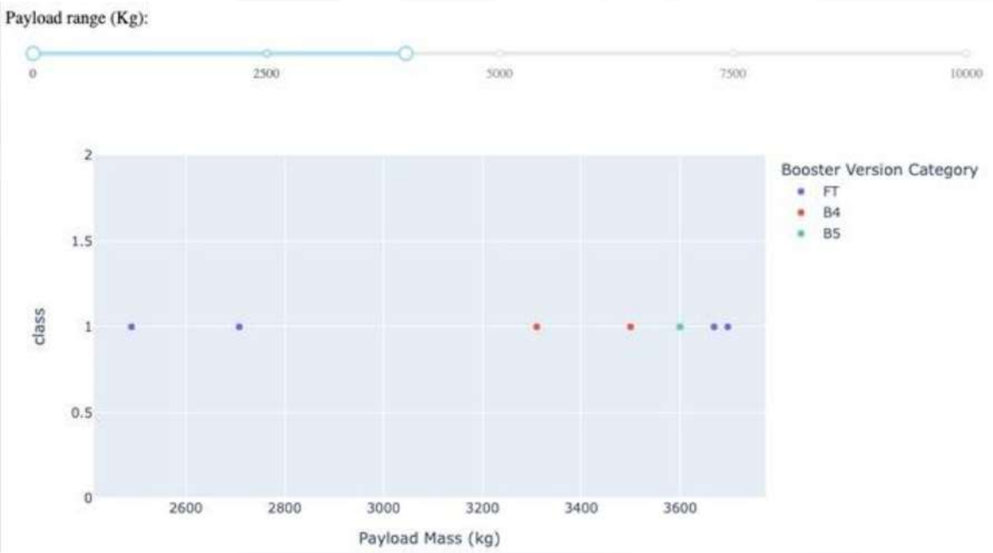
IBM Developer

SKILLS NETWORK

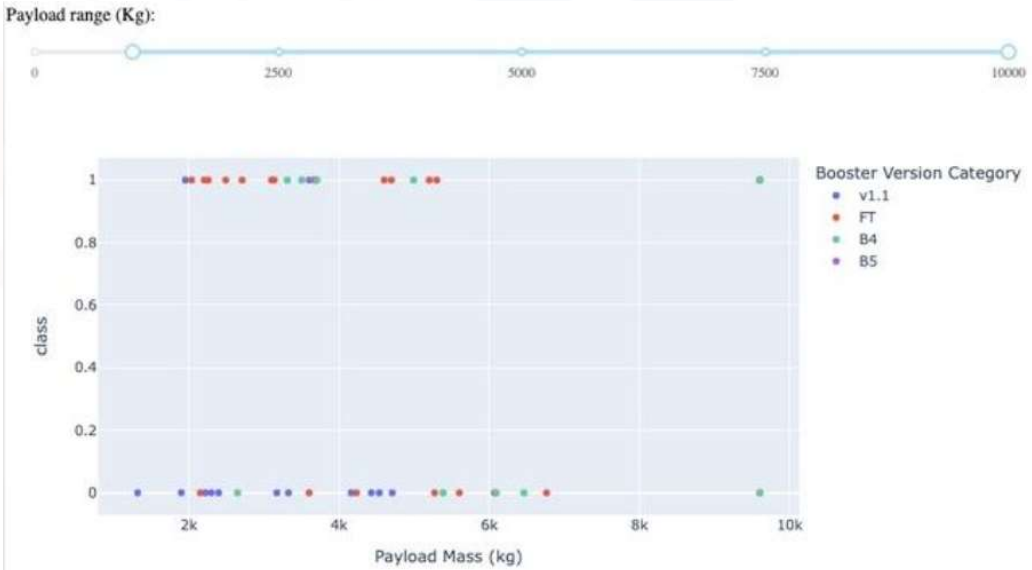
BUILD A DASHBOARD WITH PLOTLY DASH

Scatter plot payload mass vs launch outcome for all launch sites

Payload mass below 4000kg



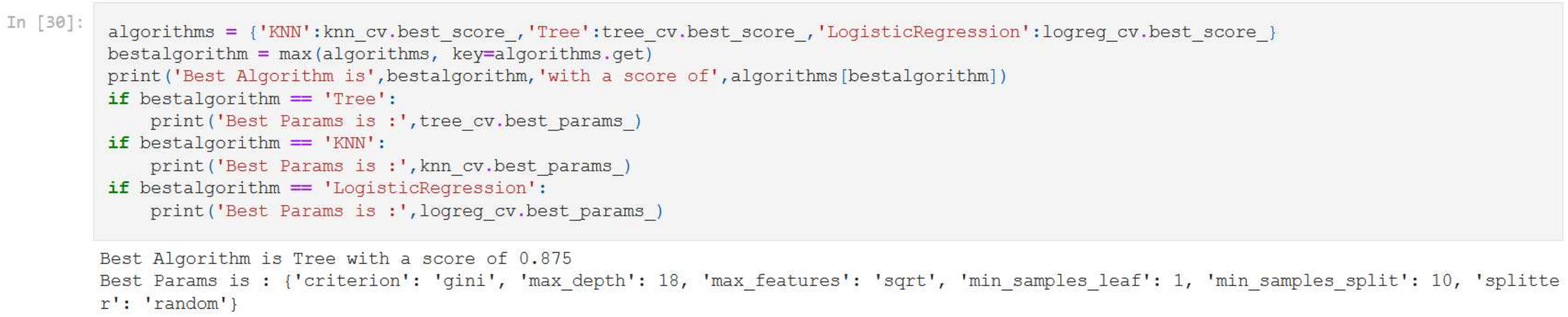
Payload mass between 4000 and 10000kg



PREDICTIVE ANALYSIS RESULTS

Find the method with the best performance

With the code below, it was compared the performance between Logistic Regression, K Nearest Neighbors and Decision Tree. The best method for this dataset is the Decision Tree.



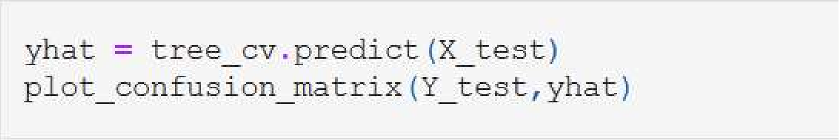
In [30]:

```python
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

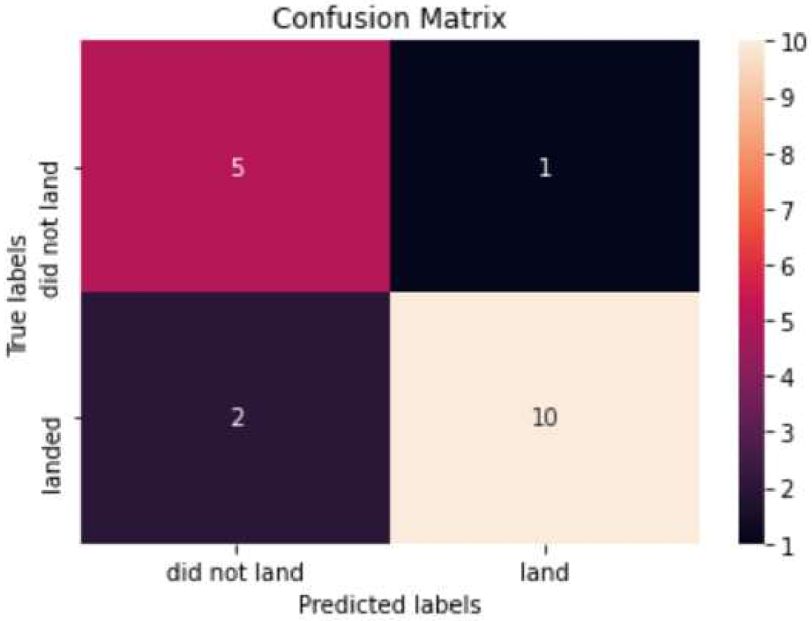
```
Best Algorithm is Tree with a score of 0.875
Best Params is : {'criterion': 'gini', 'max_depth': 18, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitte
r': 'random'}
```

PREDICTIVE ANALYSIS RESULTS

Confusion Matrix



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



The confusion matrix to the dataset shows that the Tree Classifier model can distinguish the different classes.

Only three times it gave a wrong result – one false positive and two false negative.

CONCLUSION

- Through years, since 2013, F9 landing success rate is continuously increasing. It’s probable that it continues to improve even more as time passes;
- The SpaceX launch sites are all in the US and close to a coastline;
- The best Machine Learning method to predict the landing outcome success for this SpaceX dataset is the Decision Tree Classifier;
- The most successful orbits are ES-L1, GEO, HEO and SSO;
- The success rate is lower when payload mass is over 4000kg.

APPENDIX

Thank you!