IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Ibrahim Volpi>
<March 2023>

Outline

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

Executive Summary

- On this presentation the conclusions extracted after the data collection, wrangling, analysis and model fit will be provided regarding the Success / Failure of launchings for Falcon 9 Orbital Class Rocket activities under SpaceX from four different launch sites.

- The EDA and model fit will be processed to gain insight in terms of the correlation of different variables like Payload mass, orbit selected, landing site, etc.

Data collection methodology:

Both Web Scraping with "requests" and "BeautifulSoup" packages and the use of the SpaceX API. For consistency on the results, an URL to a JSON file was provided *

https://github.com/IbraTebas/Capstone-Watson/tree/master refer to the MASTER GITHUB branch with all the completed activities.

Results:

- Recommendations regarding decrease on success rates.

- Recommendations regarding potential orbits and payload mass with higher success rate

- Recommendation regarding best performing launching sites.

*https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json

Introduction

- This project was developed in the context of the Capstone submission for the IBM Data Science Professional Certificate.
- The idea of the project is to gain insight in terms of the Success / Failure of launches for Falcon 9 by SpaceX and the correlation between different variables involved on each launch.
- The data provided had multiple sources including Wikipedia URLs, the SpaceX API and JSON files.
- The reach of this project is limited as per the volume of data accessed, EDA with visualization and SQL DataBase exploration is included, as well as more interactive exploratory tools as Dashboards
- The reach of the conclusions is measured based on the amount of data provided.
- Future models and EDAs process with a bigger set of data would be useful in terms of gaining more applicability of the recommendations and insights.

Falcon 9 launches Dragon to the International Space Station from Launch Complex 39A

Section 1

Methodology

Methodology

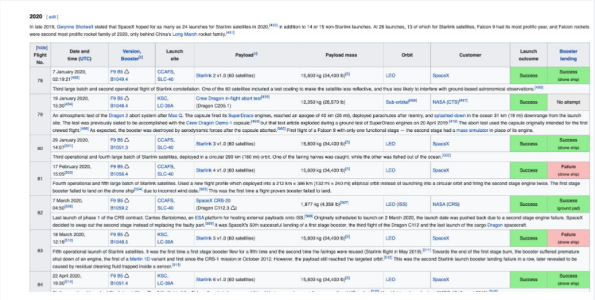
Executive Summary

- Data collection methodology:
  - Both Web Scraping with “requests” and “BeautifulSoup” packages and the use of the SpaceX API. For consistency on the results, an URL to a JSON file was provided *
- Perform data wrangling
  - Data was selected for FALCON 9 only, missing values were dealt with.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Several models were applied on the standardized one hot encoded Dataframe using the package scikit-learn from Python. The best performance was informed.

*https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json

Data Collection

- Data collection methodology:
  - On the first Notebook to be completed : WEEK 1 - Data collection with web scraping, the data was obtained from the website below: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launchess
  - The data was obtained using the Python packages ‘requests’ and ‘Beautiful Soup’



  - On the second Notebook to be completed: WEEK 1 - Hands-on Lab -Complete the Data Collection API Lab, the data was collected using the SpaceX API and a JSON provided.

Data Collection – SpaceX API I

- GitHub URL of the completed SpaceX API calls notebook
- https://github.com/IbraTebas/Capstone-Watson/blob/master/WEEK%201%20-%20Hands-on%20Lab%20-Complete%20the%20Data%20Collection%20API%20Lab.ipynb

Please paste above url in https://nbviewer.org/ to be able to see all the results.

Request for data:

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

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

Normalize Json:

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

Use of several functions to grab the appropriate data:

Now, let's apply `getBoosterVersion` function method to get the booster version

```
# Call getBoosterVersion
getBoosterVersion(data)
```

the list has now been update

```
BoosterVersion[0:5]
```

we can apply the rest of the functions here:

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

Data Collection – SpaceX API II

- GitHub URL of the completed SpaceX API calls notebook
- https://github.com/IbraTebas/Capstone-Watson/blob/master/WEEK%201%20-%20Hands-on%20Lab%20-Complete%20the%20Data%20Collection%20API%20Lab.ipynb

Please paste above url in https://nbviewer.org/ to be able to see all the results.

Filtering to keep only Falcon 9 launches:

```
# Hint data['BoosterVersion']!='Falcon 1'
df['BoosterVersion'] = BoosterVersion
data_falcon9 = df[df['BoosterVersion']!="Falcon 1"]
data_falcon9.head()
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False
5	8	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False
6	10	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False
7	11	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False
8	12	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False

Reset of the FlightNumber column:

```
In [41]: data_falcon9.reset_index(drop = True)
```

Out[41]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite
0	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40
1	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40
2	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40

Data Collection - Scraping

- GitHub URL of the completed web scraping notebook: https://github.com/IbraTebas/Capstone-Watson/blob/master/WEEK%201%20-%20Data%20collection%20with%20web%20scrapping.ipynb

Please paste above url in https://nbviewer.org/ to be able to see all the results.

Request for data:

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

In [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
data  = requests.get(static_url).text
```

Creation of the BeautifulSoup object:

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

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data, 'html5lib')
```

Location and triage of the tables on the Soup object:

In [9]:
```
# 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')
```

Starting from the third table is our target table contains the actual launch records.

In [10]:
```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

Creation of the columns of the future DataFrame and parsing of first_launch_table . Dataframe construction from dictionary.

After you have fill in the parsed launch record values into `launch_dict`, you can create a dataframe from it.

In [79]:
```
df=pd.DataFrame(launch_dict)
```

In [81]:
```
df.tail()
```

Out[81]:

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booste
116	117	CCSFS	Starlink	15,600 kg	LEO	SpaceX	Success	F9 B5B1051.10	
117	118	KSC	Starlink	~14,000 kg	LEO	SpaceX	Success	F9 B5B1058.8	
118	119	CCSFS	Starlink	15,600 kg	LEO	SpaceX	Success	F9 B5B1063.2	
119	120	KSC	SpaceX CRS-22	3,328 kg	LEO	NASA	Success	F9 B5B1067.1	
120	121	CCSFS	SXM-8	7,000 kg	GTO	Sirius XM	Success	F9 B5	

Data Wrangling

- GitHub URL of the completed data wrangling notebook: https://github.com/IbraTebas/Capstone-Watson/blob/master/WEEK%201%20-%20Data%20collection%20with%20web%20scrapping.ipynb

Please paste above url in https://nbviewer.org/ to be able to see all the results.

NaN values on the LandingPad column were kept as it represents the times the landing pads were not used.

NaN values on the PayloadMass column were filled in with the average of the column

```
# Calculate the mean value of PayloadMass column
PayloadMass_mean = data_falcon9['PayloadMass'].mean()

# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].fillna(value=PayloadMass_mean, inplace = True)
```

Checked the list of NaN or null values after this transformation, only column showing NaN values would be Landing Pad:

```
In [96]: data_falcon9.isnull().sum()

Out[96]: FlightNumber       0
         Date               0
         BoosterVersion     0
         PayloadMass        0
         Orbit              0
         LaunchSite         0
         Outcome            0
         Flights            0
         GridFins           0
         Reused             0
         Legs               0
         LandingPad        26
         Block              0
         ReusedCount        0
         Serial             0
         Longitude          0
         Latitude           0
         dtype: int64
```

EDA with Data Visualization

- Scatter point charts featuring payload mass and orbits as features were represented with the class variable as hue for the dots represented.
- A bar plot regarding the Orbit selected and the Success Rate was processed to understand the correlation between this two variables.
- A line plot show was made to show the evolution of the success absolute value through the years.
- Notebook: https://github.com/IbraTebas/Capstone-Watson/blob/master/WEEK%202%20-%20Hands%20on%20Lab%20-%20Complete%20EDA%20with%20Data%20Visualization.ipynb

Please paste above url in https://nbviewer.org/ to be able to see all the results.

EDA with SQL

- Several queries were processed to the database hosted on IBM Cloud.
- This queries include:
  - Name of launching sites
  - Booster versions against payload mass
  - Average payload mass handled
  - First occurrence of a successful Ground Pad landing
  - Total number of missions with Success / Failure
  - Some results by year depending on the outcome.
- https://github.com/IbraTebas/Capstone-Watson/blob/master/WEEK%202%20-%20Hands-on%20Lab%20-%20%20Complete%20the%20EDA%20with%20SQL.ipynb completed notebook.

Please paste above url in https://nbviewer.org/ to be able to see all the results.

Build an Interactive Map with Folium

- Several activities were processed on Folium maps:
  - Markers for the landing sites analysed
  - Markers for Success/ Failed launches for each site.
  - Distance and lines showing the proximity of one of the launch sites (CCAFS SLC-40) to several points of interest as nearest city, railway, highway and coastline.
- https://github.com/IbraTebas/Capstone-Watson/blob/master/WEEK%203%20-%20Hands-on%20Lab%20-%20Complete%20the%20Data%20visualization%20with%20Folium.ipynb GitHub with all activities completed.

Please paste above url in https://nbviewer.org/ to be able to see all the results.

Build a Dashboard with Plotly Dash

- 5 different types of piecharts were added to the Dashboard, showing the success rate of each launch site in particular and also in comparison all of them together. The selection of each site or the option "All Sites" is actioned by the user with a drop down menu.
- Scatter plots were added taking into considerations the variables of payload mass, the success or failure (class) and also the Booster Version used. The user has the ability to adjust the payload mass interval variable using a slider.
- Refer to https://github.com/IbraTebas/Capstone-Watson/blob/master/WEEK%203%20-%20Hands-on%20Lab%20-%20Build%20an%20Interactive%20Dashboard%20with%20Ploty%20Dash.ipynb complete notebook reporting the above.

Predictive Analysis (Classification)

- The data is loaded from two CSV files provided to make the results similar to all students.(*)(**)
- "Class" column in one of the two datasets is converted to a numpy array to be used as train and test Y set.
- Data for the train and test X set is Normalized.
- A test/test partition of the data is created.
- Several models are fit using GridSearchCV with cv = 10
- Accuracy and score is measure and confusion matrix provided.
- Best predictor classifier is stated as Tree Classifier based on the results.
- https://github.com/IbraTebas/Capstone-Watson/blob/master/Week%204%20-%20%20Hands-on%20Lab%20-%20Machine%20Learning%20Prediction%20lab.ipynb GitHub with all the activities completed.

Please paste above url in https://nbviewer.org/ to be able to see all the results.

*https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv

**https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_3.csv'

TASK 1

Create a NumPy array from the column `Class` in `data`, by applying the method `to_numpy()` (only one bracket df['name of column']).

```
Y = data["Class"].to_numpy()
```

TASK 2

Standardize the data in `X` then reassign it to the variable `X` using the transform provided below.

```
# students get this
transform = preprocessing.StandardScaler()
Delta = pd.DataFrame(transform.fit(X).transform(X))
Delta.columns=X.columns
X = Delta
X.head()
```

TASK 3

Use the function train_test_split to split the data X and Y into training and test data. Set the parameter test_size to 0.2 and random_state to 2. should be assigned to the following labels.

`X_train, X_test, Y_train, Y_test`

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.20, random_state=2)
```

TASK 9

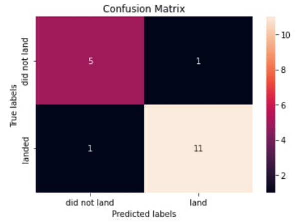
Calculate the accuracy of tree_cv on the test data using the method `score`:

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

```
0.8888888888888888
```

We can plot the confusion matrix

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



Results

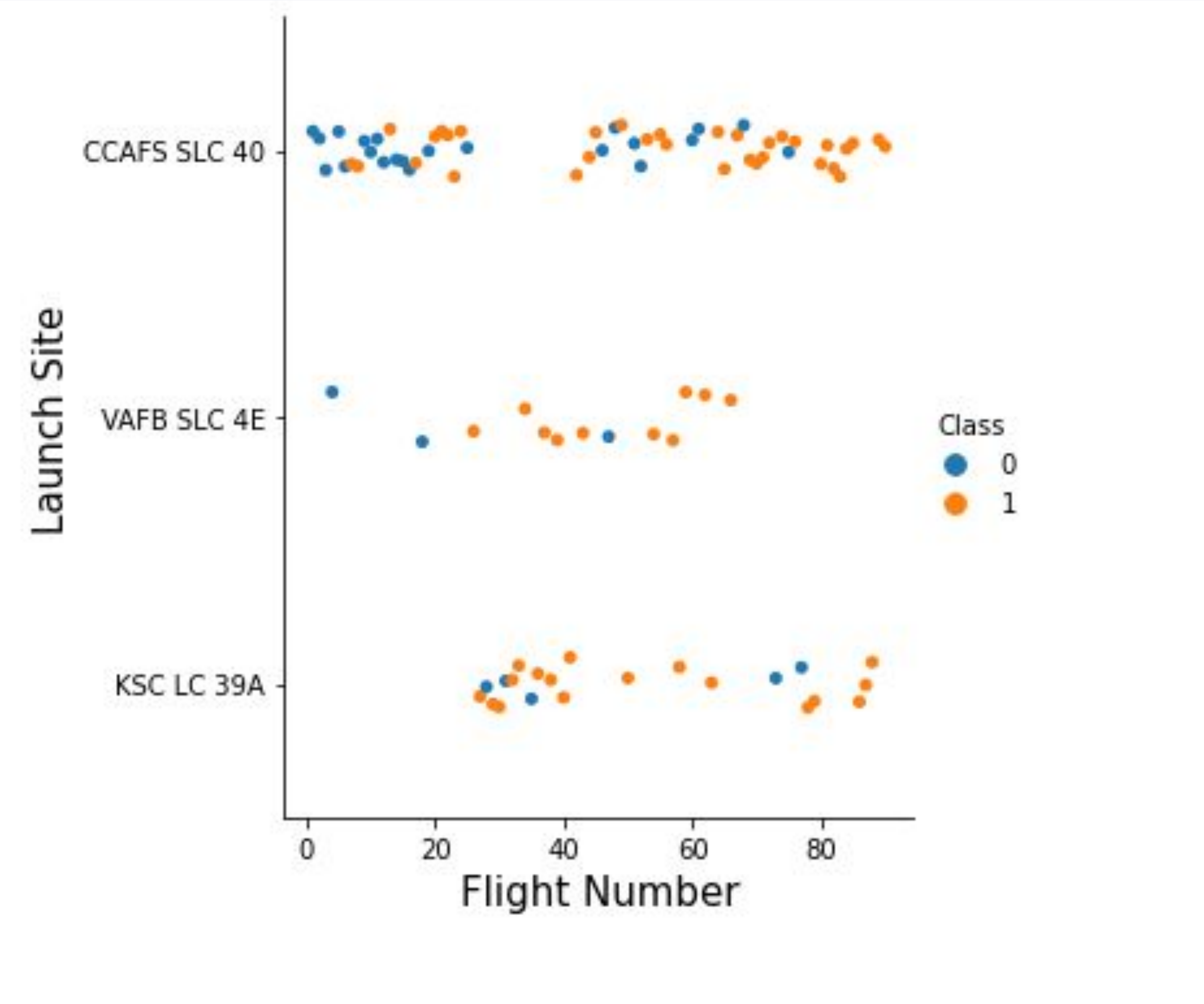
Results:

- Regarding the success rate evolution through years:
  - The launch success rate had diminished since 2020 in terms of the records analysed.
- Recommendations regarding potential orbits and payload mass with higher success rate:
  - VLEO shows as a strong candidate with 85% success rate.
  - Payload masses between 3000 kg and 4000 kg seem to have a much better performance (70% success)
- Recommendation regarding best performing launching sites.
  - KSC LC 39A (Kennedy Space Center Launch Complex 39A) has the best performance in terms of absolute amount of successful landings and in terms of success rate.

Section 2

Insights drawn from EDA

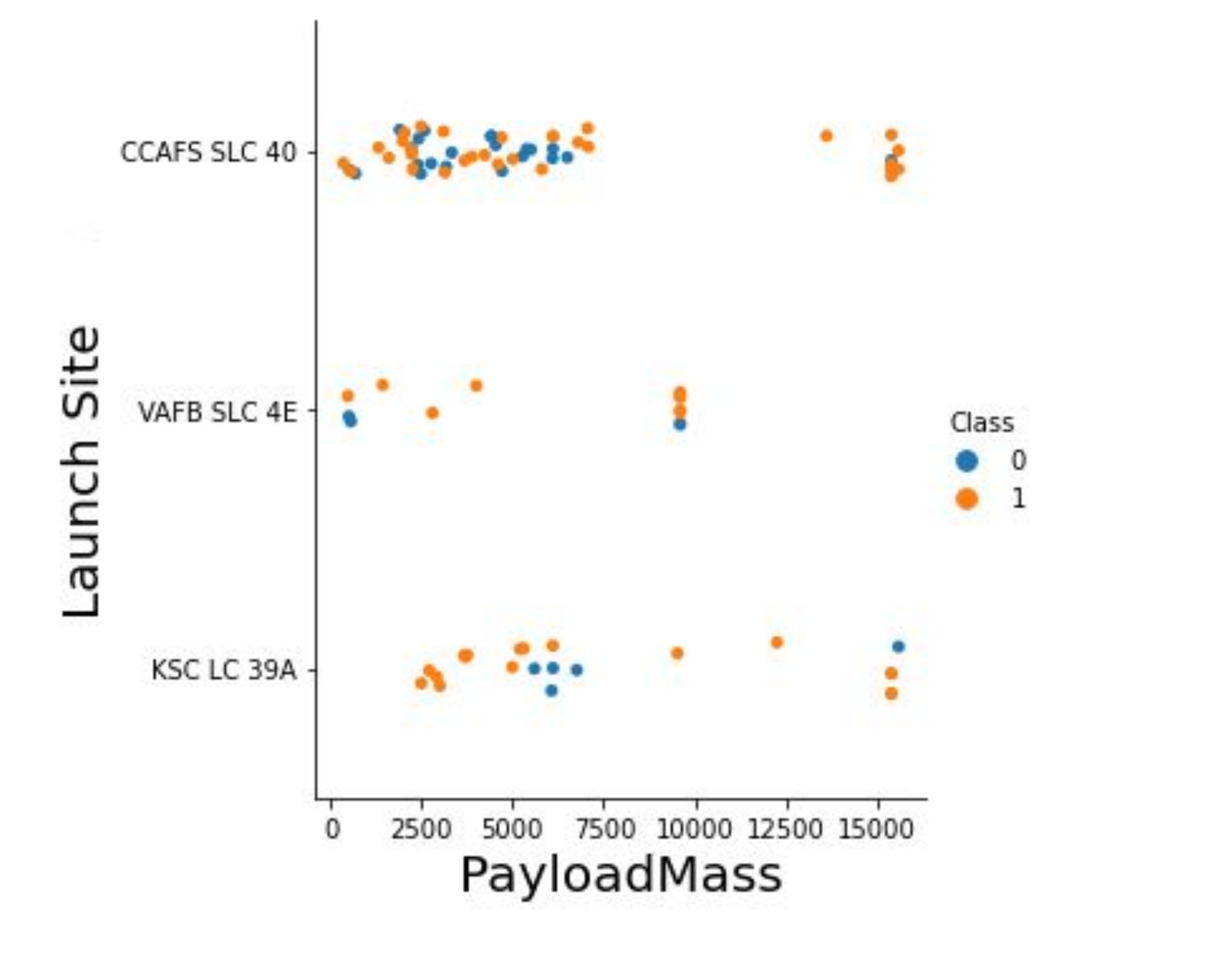
Flight Number vs. Launch Site



Insights:

- Both VAFB SLC 4E and KSC LC 39A have a better success rate than CCAFS SLC 40
- CCAFS SLC 40 improves their success rate greatly after flight number 60 and onwards.
- VABF SLC 4E has had significantly less flights than the other two sites.

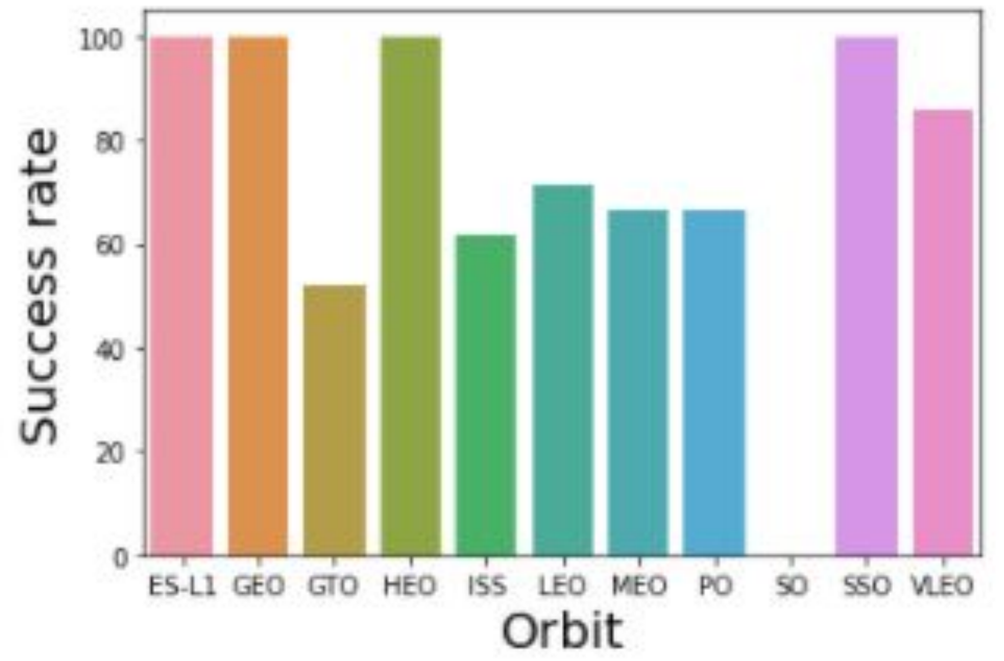
Payload vs. Launch Site



Insights:

- Barely no launches are processed for values of payload mass between 7500 and 12500 Kg, and none on site CCAFS SLC 40
- CCAFS SLC 40 has the largest amount of heavy payload mass launches successful.
- VABF SLC 4E has not processed any lunches for heavy payload mass (greater than 10,000 KG)

Success Rate vs. Orbit Type

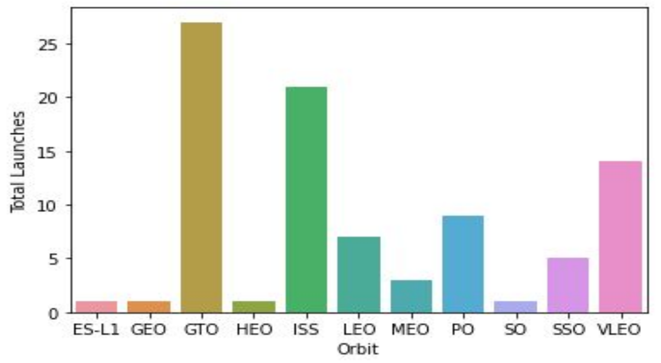


Insights:

- The success rate for orbit SO is 0% but only one launch was attempted.
- The orbit with less success rate is GTO with 51% success rate.
- ES-L1, GEO, HEO and SSO orbits shows a 100% success rate but very few launches.
- Taking into consideration that the orbits more used are GTO, ISS and VLEO in that order, VLEO shows as a strong candidate with 85% success rate.

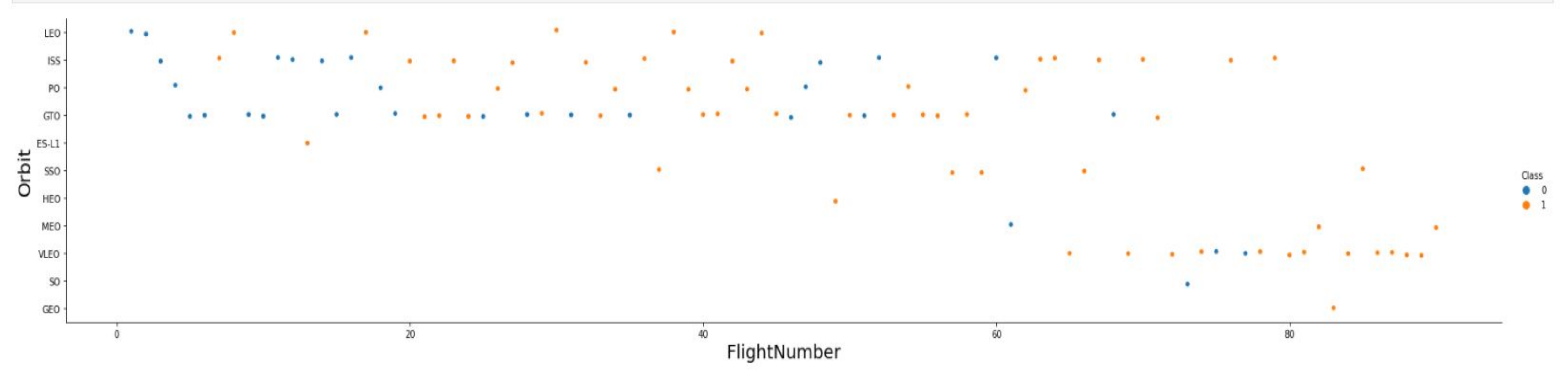
Out[29]:

	Orbit	FlightNumber	PayloadMass	Flights	GridFins	Reused	Legs	Block	ReusedCount	Longitude	Latitude	Class	Success_rate
0	ES-L1	13.000000	570.000000	1.000000	1.000000	0.000000	1.000000	1.000000	0.000000	-80.577366	28.561857	100.000000	100%
1	GEO	83.000000	6104.959412	2.000000	1.000000	1.000000	1.000000	5.000000	2.000000	-80.577366	28.561857	100.000000	100%
2	GTO	35.037037	5011.994444	1.407407	0.629630	0.333333	0.629630	3.037037	0.962963	-80.586229	28.577258	51.851852	51%
3	HEO	49.000000	350.000000	1.000000	1.000000	0.000000	1.000000	4.000000	1.000000	-80.577366	28.561857	100.000000	100%
4	ISS	39.142857	3279.938095	1.238095	0.809524	0.238095	0.857143	3.142857	1.285714	-80.583697	28.572857	61.904762	61%
5	LEO	20.000000	3882.839748	1.000000	0.571429	0.000000	0.714286	2.142857	0.428571	-80.584963	28.575058	71.428571	71%
6	MEO	77.666667	3987.000000	1.000000	0.666667	0.000000	0.666667	5.000000	0.666667	-80.577366	28.561857	66.666667	66%
7	PO	36.333333	7583.666667	1.333333	0.888889	0.333333	0.777778	3.222222	1.555556	-120.610829	34.632093	66.666667	66%
8	SO	73.000000	6104.959412	4.000000	0.000000	1.000000	0.000000	5.000000	3.000000	-80.603956	28.608058	0.000000	0%
9	SSO	60.800000	2060.000000	2.400000	1.000000	0.800000	1.000000	4.600000	3.200000	-112.604136	33.418046	100.000000	100%
10	VLEO	78.928571	15315.714286	3.928571	1.000000	1.000000	1.000000	5.000000	3.928571	-80.586862	28.578358	85.714286	85%



	Orbit	Total Launches
0	ES-L1	1
1	GEO	1
2	GTO	27
3	HEO	1
4	ISS	21
5	LEO	7
6	MEO	3
7	PO	9
8	SO	1
9	SSO	5
10	VLEO	14

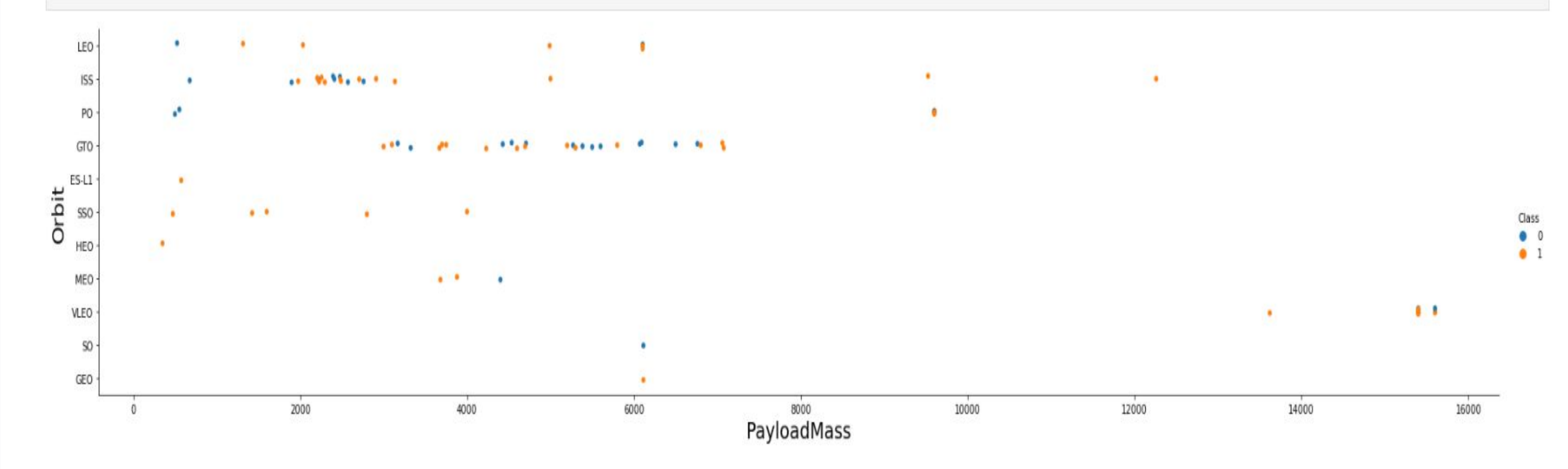
Flight Number vs. Orbit Type



Insights:

- LEO Orbit: the success seems to be correlated to the flight number.
- GTO orbit: the success does not seem to be correlated to the flight number (keeps on altering colors)

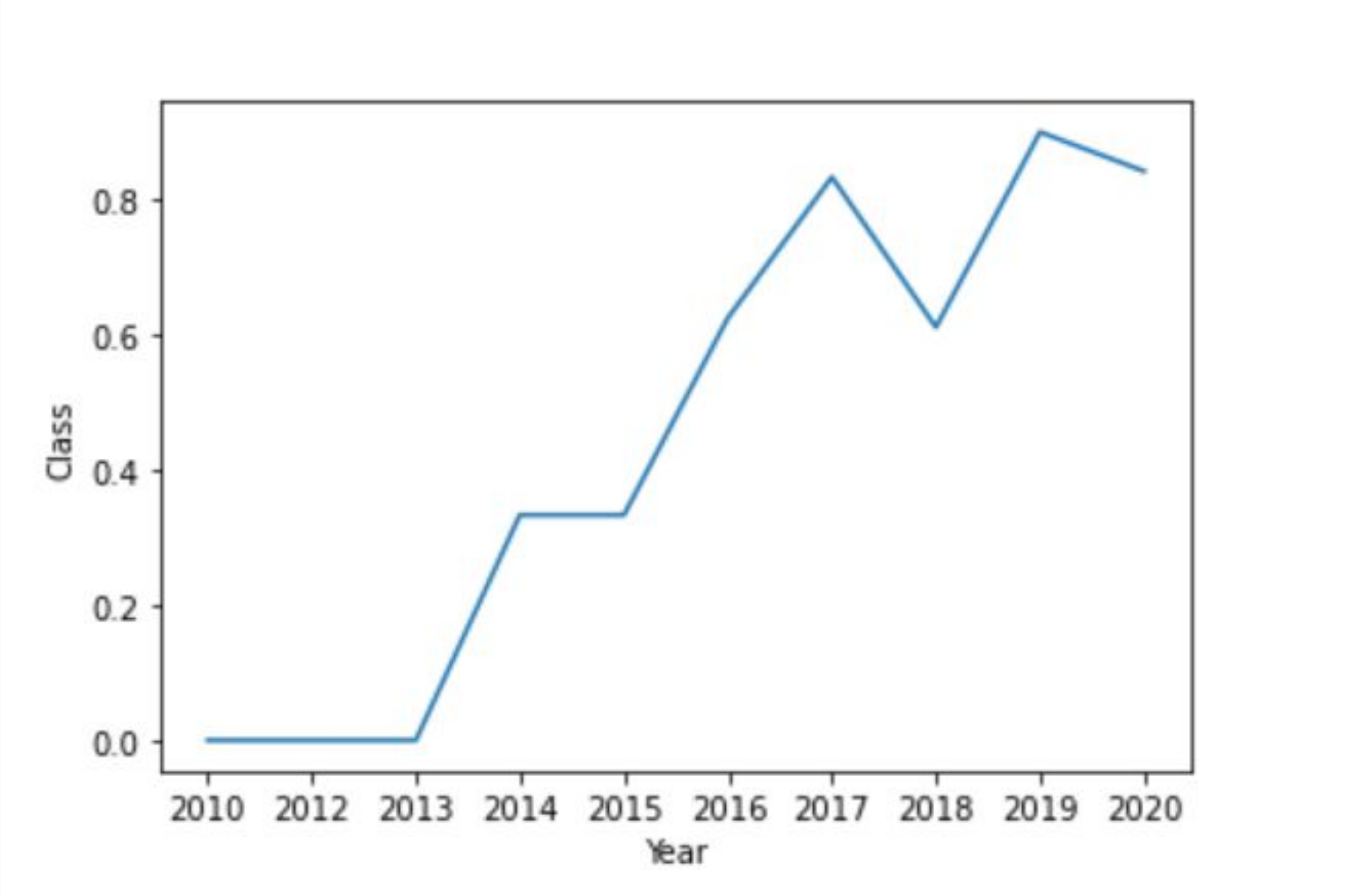
Payload vs. Orbit Type



Insights:

- POLAR, LEO and ISS orbits: heavy payloads seem to be successful more frequently.
- GTO orbit: the alternancy of the success and failure seems to not be correlated to payload mass.

Launch Success Yearly Trend



Insights:

- The success rate had mostly increased since 2013 until 2020.
- The success rate seems to diminish after 2020.

All Launch Site Names - SQL Query

Task 1

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

In [6]:
```
%sql SELECT DISTINCT launch_site FROM SPACEXTBL
```

Out[6]:

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

The "SELECT DISTINCT" part of the query provides the categories of the variable.

All Launch Site Names - Further information

Cape Canaveral Space Launch Complex 40

https://en.wikipedia.org/wiki/Cape_Canaveral_Space_Launch_Complex_40

From Wikipedia, the free encyclopedia

Space Launch Complex 40[2][3] (**SLC-40**), previously **Launch Complex 40** (**LC-40**) is a launch pad for rockets located at the north end of Cape Canaveral Space Force Station, Florida.

Kennedy Space Center Launch Complex 39A

https://en.wikipedia.org/wiki/Kennedy_Space_Center_Launch_Complex_39A

From Wikipedia, the free encyclopedia

Launch Complex 39A (**LC-39A**) is the first of Launch Complex 39's two launch pads, located at NASA's Kennedy Space Center in Merritt Island, Florida. The pad, along with Launch Complex 39B, were first designed for the Saturn V launch vehicle, which is still the United States' most powerful rocket. Typically used to launch NASA's crewed spaceflight missions since the late 1960s, the pad was leased by SpaceX and has been modified to support their launch vehicles.[2][3]

Vandenberg Space Launch Complex 4

https://en.wikipedia.org/wiki/Vandenberg_Space_Launch_Complex_4

From Wikipedia, the free encyclopedia

Space Launch Complex 4 (**SLC-4**) is a launch and landing site at Vandenberg Space Force Base, California, U.S. It has two pads, both of which are used by SpaceX for Falcon 9, one for launch operations, and other as **Landing Zone 4** (**LZ-4**) for SpaceX landings.

	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610746

Codes and coordinates (Lat = Latitude // Long = Longitude) for the launch sites analysed.

Refer to top Wikipedia article regarding the full name of each launch site.

Launch Site Names Begin with 'CCA'

Task 2

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

```
%sql SELECT * FROM SPACEXTBL WHERE ( launch_site LIKE 'CCA%') LIMIT 5
```

DATE	time_utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	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)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

The “LIKE ‘CCA%” sets the condition and the “LIMIT” the amount to display on this query.

Total Payload Mass

Task 3

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

In [8]:
```
%sql SELECT SUM(payload_mass__kg_) as total_payload_mass__kg_ FROM SPACEXTBL
```

Out[8]:

total_payload_mass__kg_
619967

"SUM" process the query as an addition of the variable indicated.

Average Payload Mass by F9 v1.1

Task 4

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

```
%sql SELECT  Booster_Version, AVG(payload_mass__kg_) AVG FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.1' GROUP BY Booster_Version
```

booster_version	AVG
F9 v1.1	2928

"AVG" process the average, the "LIKE" part filters the Booster Version

First Successful Ground Landing Date

Task 5

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

Hint:Use min function

```
%sql SELECT MIN(DATE) FIRST_LANDING_GORUND_PAD FROM SPACEXTBL WHERE landing__outcome LIKE 'Success (ground pad)'
```

first_landing_gorund_pad
2015-12-22

Wow! SpaceX Lands Orbital Rocket Successfully in Historic First

By Mike Wall published December 22, 2015

Article header extracted from "https://www.space.com/31420-spacex-rocket-landing-success.html"

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

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 payload_mass__kg_ > 4000 AND payload_mass__kg_ < 6000 AND landing__outcome LIKE 'Success (drone%'
```

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

In this query two “AND” were needed, as the condition implies to be lesser than a number and greater than other and also with a specific outcome (three conditions to impose on the WHERE)

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

```
%sql SELECT COUNT(*) total_number_successful_and_failure_missions FROM SPACEXTBL WHERE landing__outcome like 'Failure%' OR landing__outcome like 'Succ
```

total_number_successful_and_failure_missions
71

‘COUNT’ is used here to provide the total required, the ‘WHERE’ is applied to the landing__outcome category with both options of Success or Failure with the use of the “%”

Boosters Carried Maximum Payload

Task 8

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_) max_payload FROM SPACEXTBL)
```

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

The subquery resolves with the MAX obtains the maximum payload to then filter the Booster Versions were the maximum payload was attempted.

2015 Launch Records

Task 9

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%' AND DATE LIKE '2015%'
```

landing__outcome	booster_version	launch_site	DATE
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015-01-10
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015-04-14

Two records are displayed when the year is filtered to 2015 using the "%" wildcard, and the condition of a Failure (drone ship) scenario is imposed.

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

Task 10

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 landing__outcome, COUNT(*) COUNT FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY landing__outcome ORDER BY COUNT
```

landing__outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

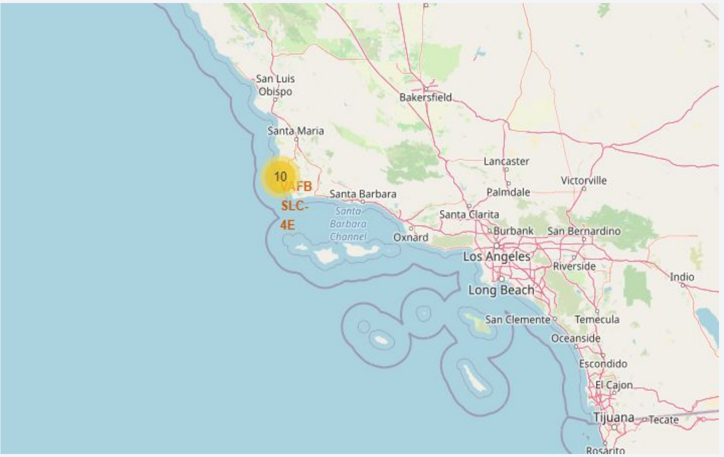
Based on the fact that in a previous part was established that the first successful landing in Ground Pad was the 22nd December 2015, it means that this feat was repeated another two times between the 22nd December, 2015 and the 20th March, 2017

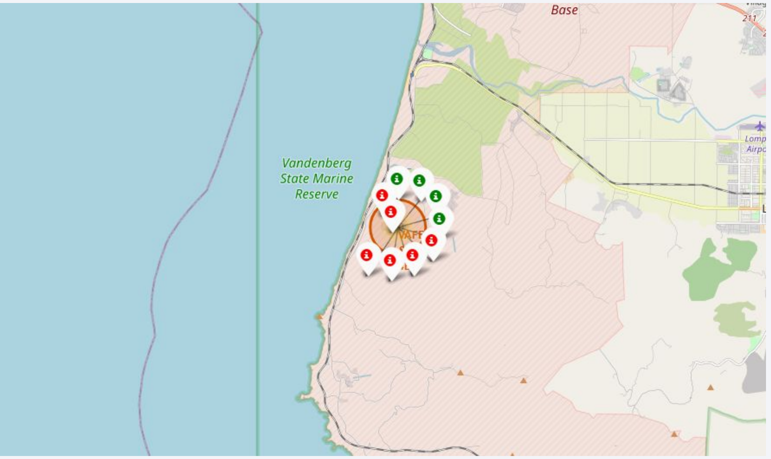
Section 3

Launch Sites Proximities Analysis

Location of the launching sites - Map I

CCAFS SLC-40 is the name update to the launch site named before CCAFS LC-40, therefore this references are mostly in the same position on the map, very near of the KSC LC-39A launch site as well.



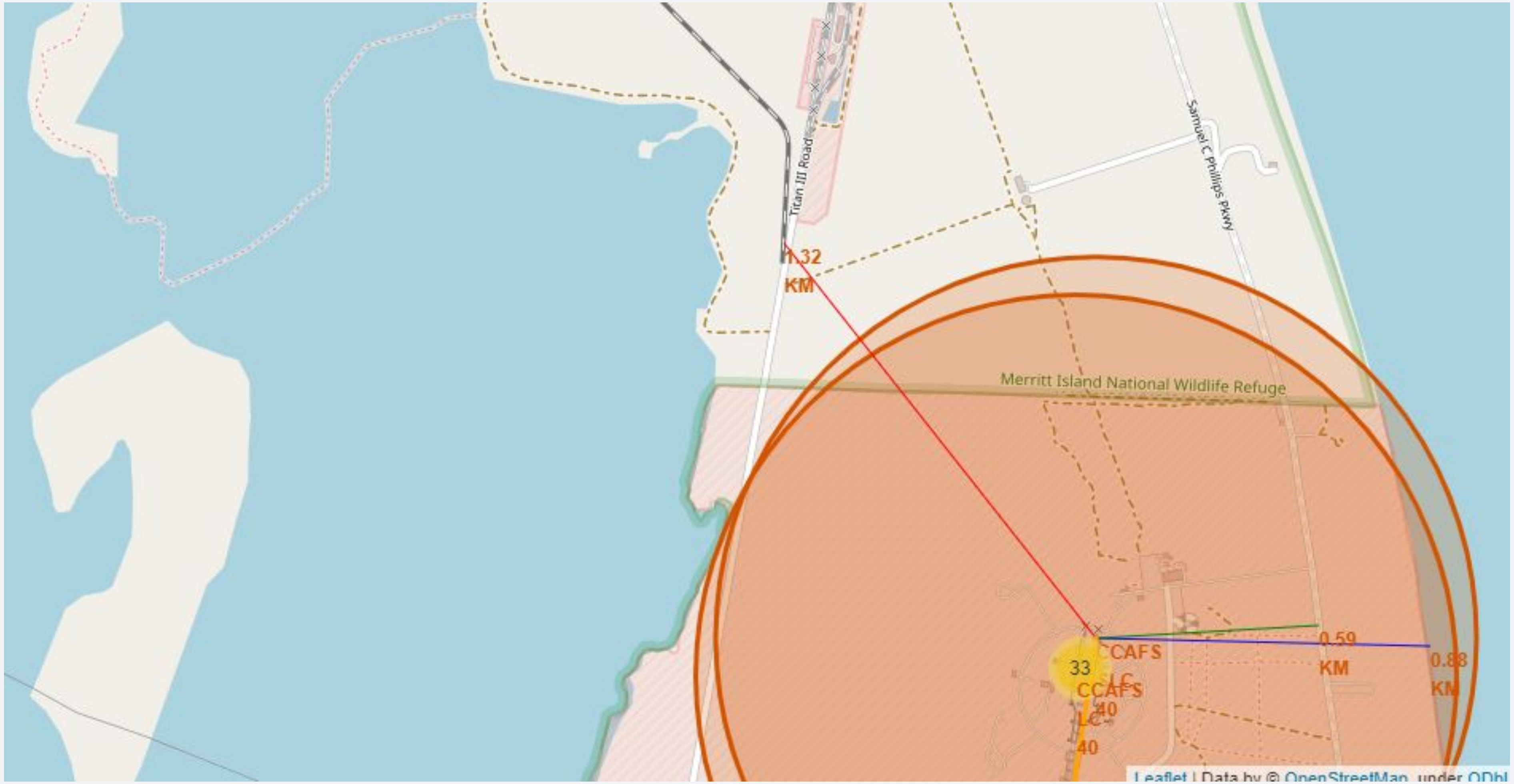


The launch site VAFB SLC-4E is located on the West Coast and only 10 records from that launching site are analysed on this report.
It is visually clear there is slightly more Failures than successes on the 10 records analysed.

Distance from CCAFC SLC-40 to the coast and highway

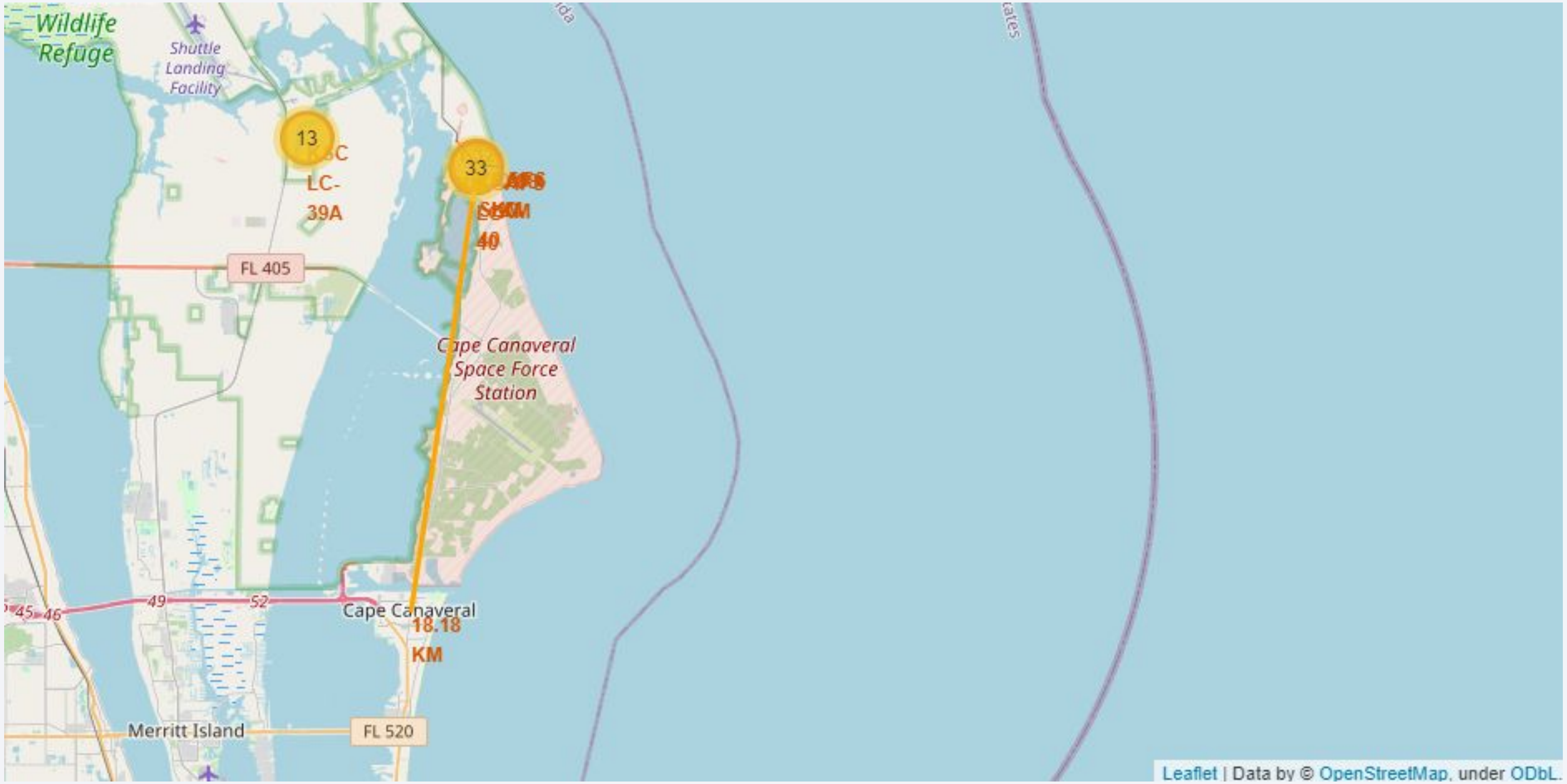
Both the distance to the highway (in green) and to the coast from CCAFC SLC-40 is lesser than 1 km.

Distance from CCAFS SLC-40 to the railway



In terms of access to railway, the access is located 1.8 km North-West from the CCAFS SLC-40 launch site.

Distance from CCAFS SLC-40 to the nearest city / town.



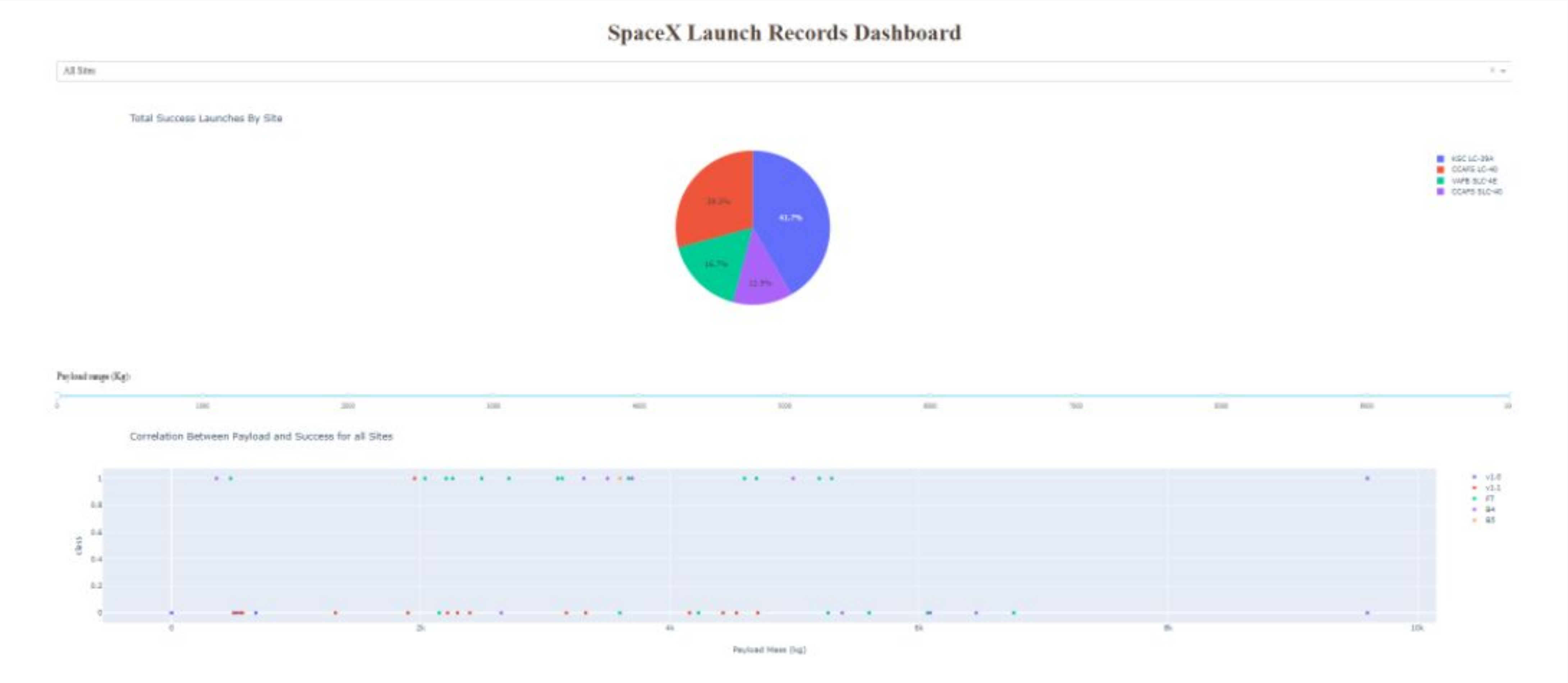
The nearest city is located South from the launching site CCAFS SLC-40, about 18.18 km

The name of the city is Cape Canaveral.

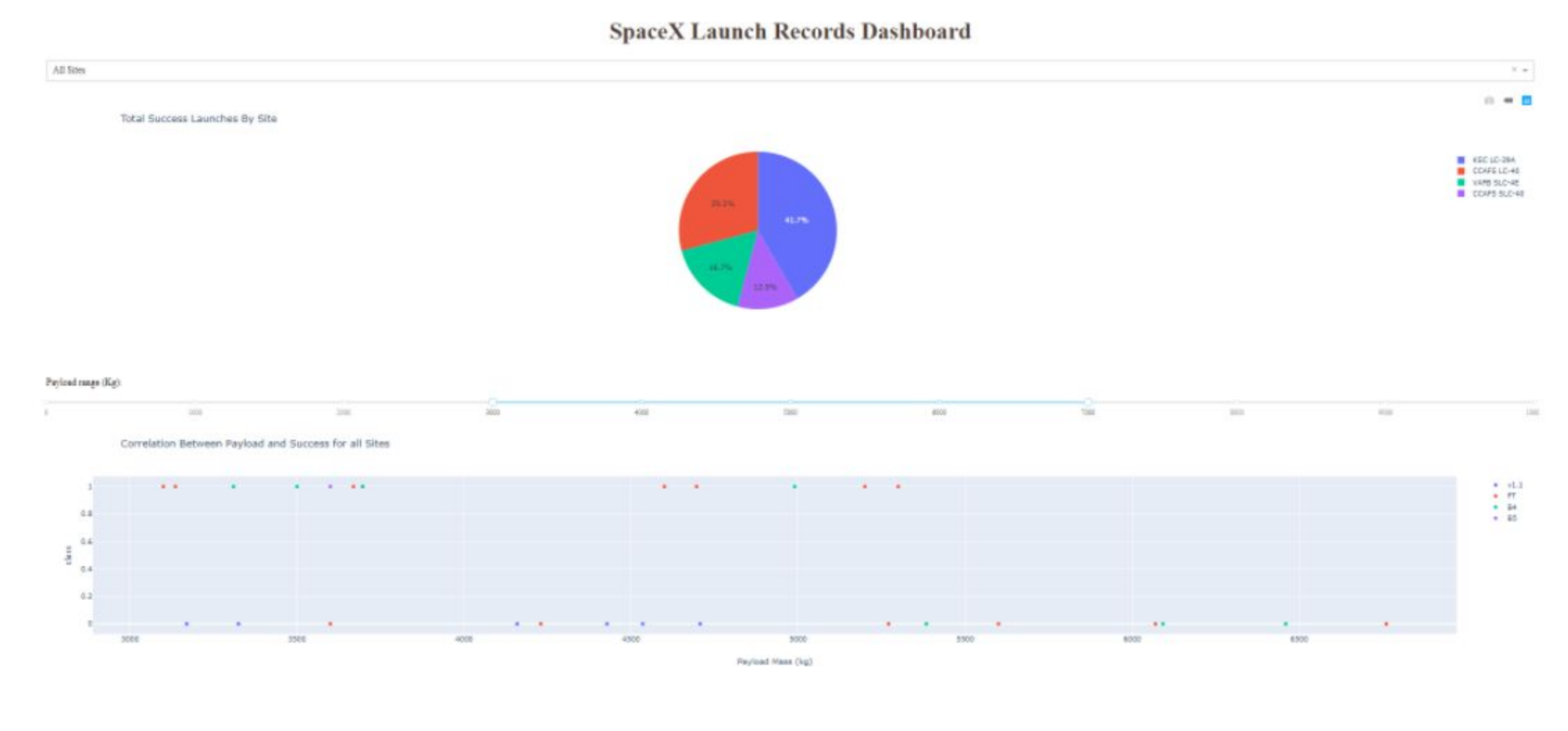
Section 4

Build a Dashboard with Plotly Dash

Selector using “All Sites option, no range filter applied with the Payload Range slider.

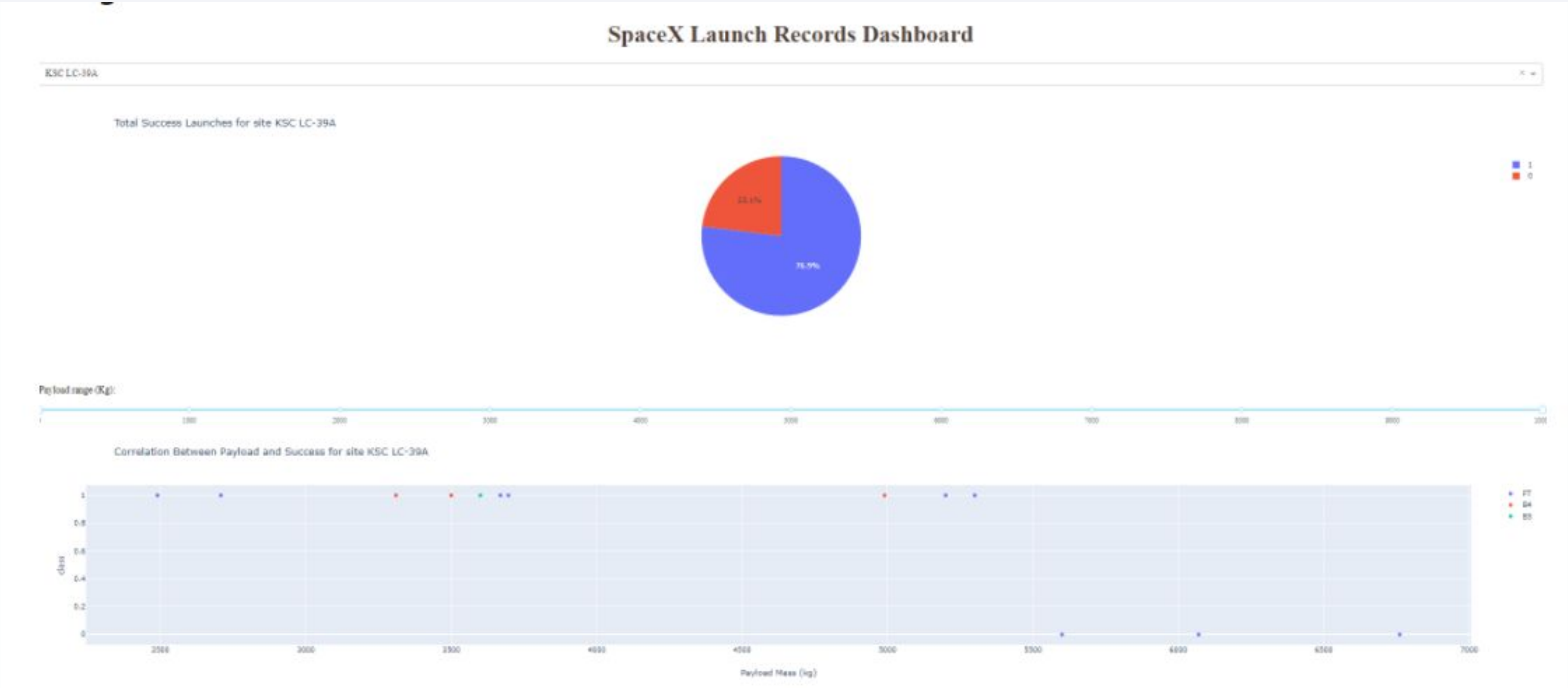


Selector using “All Sites” option, range filter applied with the Payload range slider between 3000 kg and 7000 kg



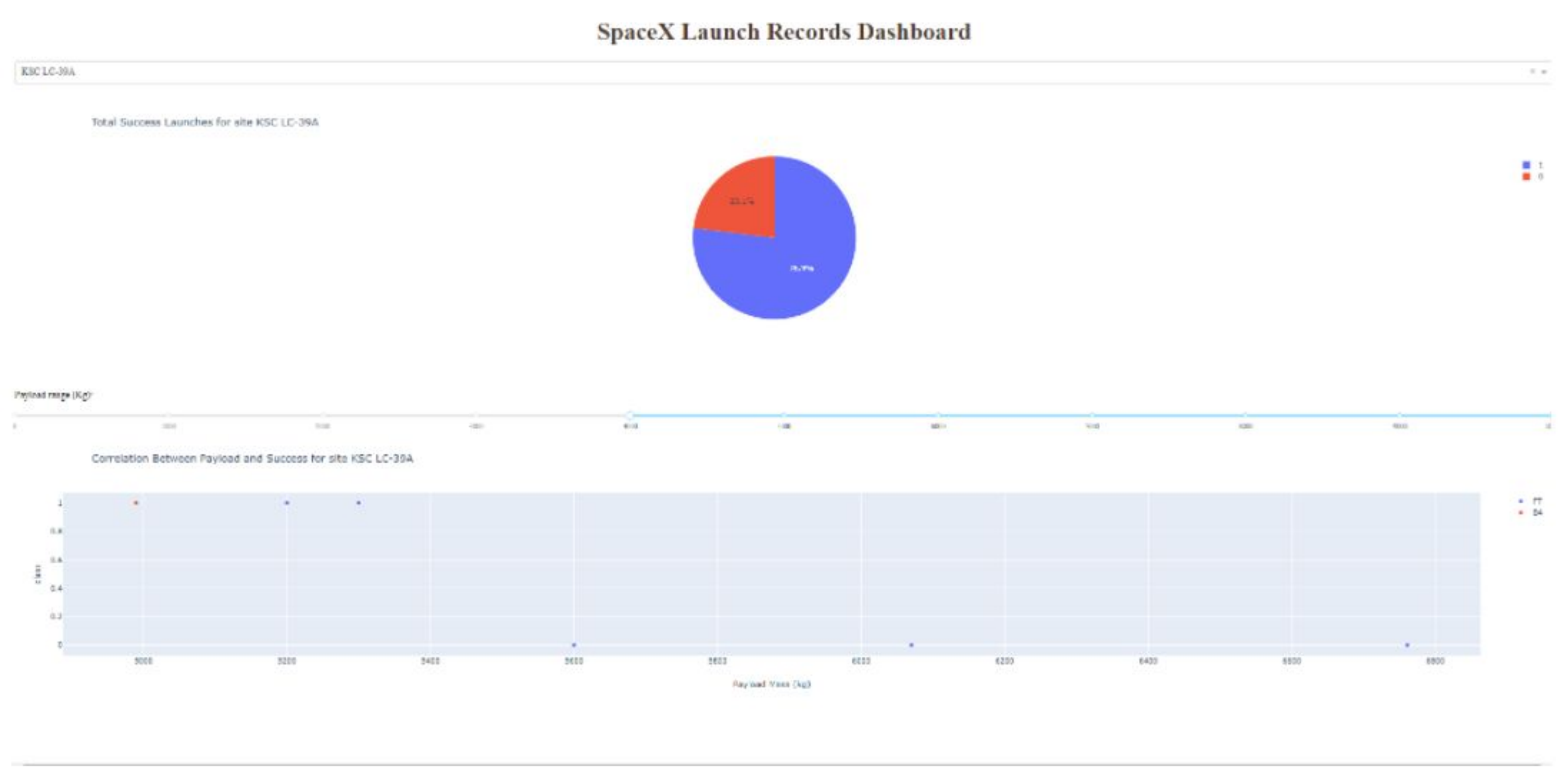
If we compare this scatter plot and the previous one, it can be observed the change on the scale of the “x” axis and also the change on the dots displayed.
It is also visible that the selected range in the slider is highlighted in light blue.

Selector using site “KSC LC-39A” option, no range filter applied with the Payload Range slider.



This is by far the site with the best success rate.

Selector using site “KSC LC-39A” option, range filter applied with the Payload Range slider between 4000 kg and 10,000 kg.



Seems like the effectiveness drop the higher the payload for this launching site.

Finding Insights Visually

Now with the dashboard completed, you should be able to use it to analyze SpaceX launch data, and answer the following questions:

1 - Which site has the largest successful launches?
2 - Which site has the highest launch success rate?
3 - Which payload range(s) has the highest launch success rate?
4 - Which payload range(s) has the lowest launch success rate?
5 - Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest launch success rate?

Responses:

1 - KSC LC-39A, this can be checked on the pie chart when "All sites" is selected
2 - The highest success rate is for KSC LC-39A, with a 76.9% success rate, this is easily obtained by going thorugh all the selector options for "Site" and grabbing the highest % with the "1" reference.

Please see below table regarding Class 1 events depending on the Pay Load Range

Payload range (kg)	Class 1 results	Total Results	Success Rate
0 - 1000	2	8	25%
1000 - 2000	1	3	33.3 %
2000 - 3000	5	10	50%
3000 - 4000	7	10	70%
4000 - 5000	3	8	37.5%
5000 - 6000	2	5	40%
6000 - 7000	0	4	0%
7000 - 8000	0	0	NA
8000 - 9000	0	0	NA
9000 - 10000	1	2	50%

The above table was constructed by using the slider for the Pay Load Range and counting the dots graphed, with "All Sites" selected.

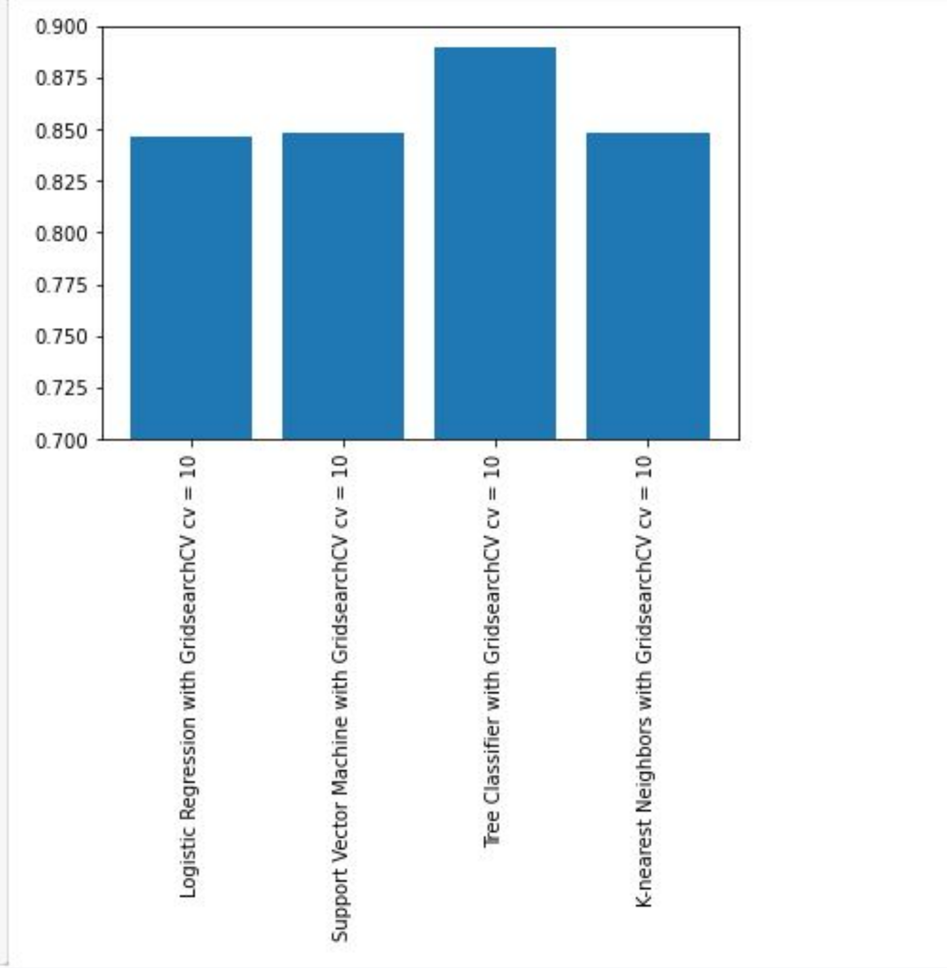
3 - The Payload range with highest launch success rate is between 3000 KG and 4000 KG, with a 70% success rate
4 - The Payload range with lowest launch success rate is between 6000 KG and 7000 KG, with a 0% success rate
5 - FT selected as a booster has the highest number of successful launches, this can be checked on the "all sites" option and no slider filtter applied, on the value 1, the reference with more dots is FT.

Some insight gained with the manipulation of the dashboard.

Section 5

Predictive Analysis (Classification)

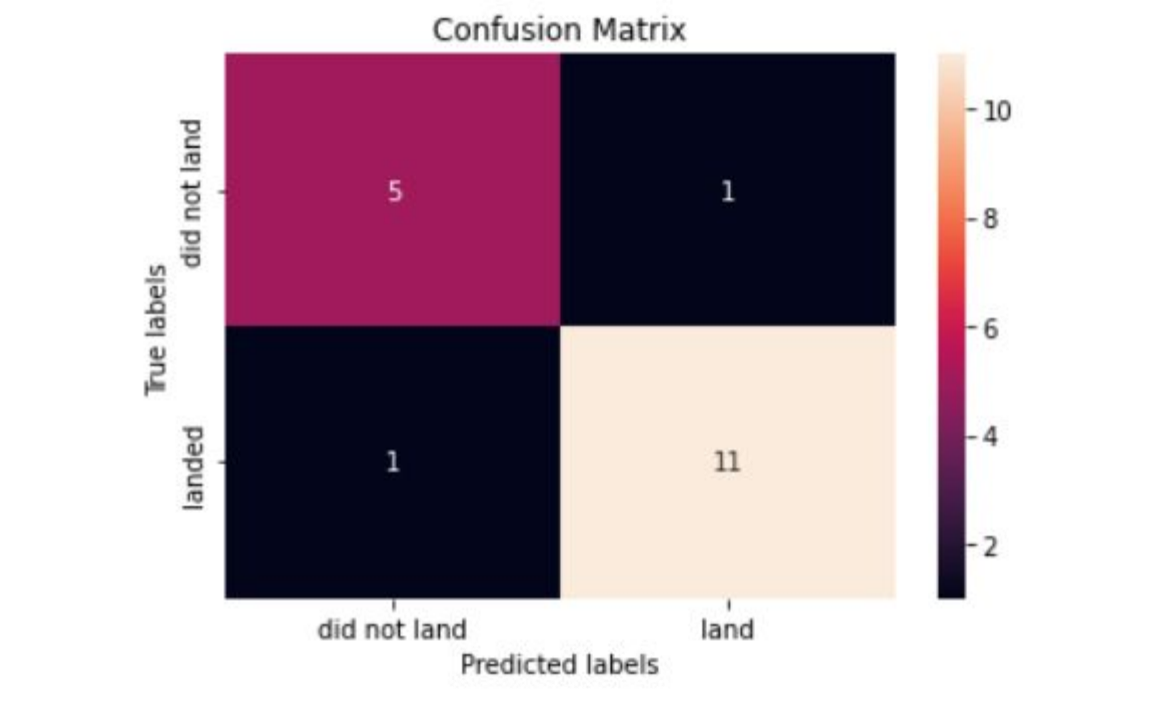
Classification Accuracy



Model	Accuracy
Logistic Regression with GridsearchCV cv = 10	0.84642857
Support Vector Machine with GridsearchCV cv = 10	0.84821429
Tree Classifier with GridsearchCV cv = 10	0.88928571
K-nearest Neighbors with GridsearchCV cv = 10	0.84821429

The model with the best accuracy is Tree Classifier when combined with GridsearchCV with cv = 10 as instructed.

Confusion Matrix - Tree Classifier



On this attempt the score obtained by the Tree Classifier was 88.89%, in 18 predictions against the test set, only two mistakes.

One mistake on the prediction against the test: it was predicted that would not land and it landed (number one on the left down black square)

Another mistake on the prediction: it was predicted that would land but it didn't (top right black square with a number 1).

Conclusions

- The launch success rate had diminished since 2020 in terms of the records analysed, a review on changes done since then would be relevant to understand the change on the tendency.

- KSC LC 39A (Kennedy Space Center Launch Complex 39A) has the best performance in terms of absolute amount of successful landings and in terms of success rate.
  An audit of the processes followed there would be beneficial to provide insight to less performers like CCAFS SLC 40 (Cape Canaveral Space Launch Complex 40)

- Taking into consideration that the orbits more used are GTO, ISS and VLEO in that order, VLEO shows as a strong candidate with 85% success rate, considering the reduced data analysed. **GTO is used the most but have the lesser success rate**, a revision in terms of VLEO orbit usage as a strong candidate should be considered if technically possible.

- Payload masses between 3000 kg and 4000 kg seem to have a much better performance (70% success)

- The Tree Classifier model trained seems to have a good performance (88.89% accuracy) but the amount of records used to trained the model was too small, so overfitting is a risk with this data size.
  In order to have a more valuable classifier predictor, more records should be added to the clean data to be able to train the model with a bigger set.

Thank you!