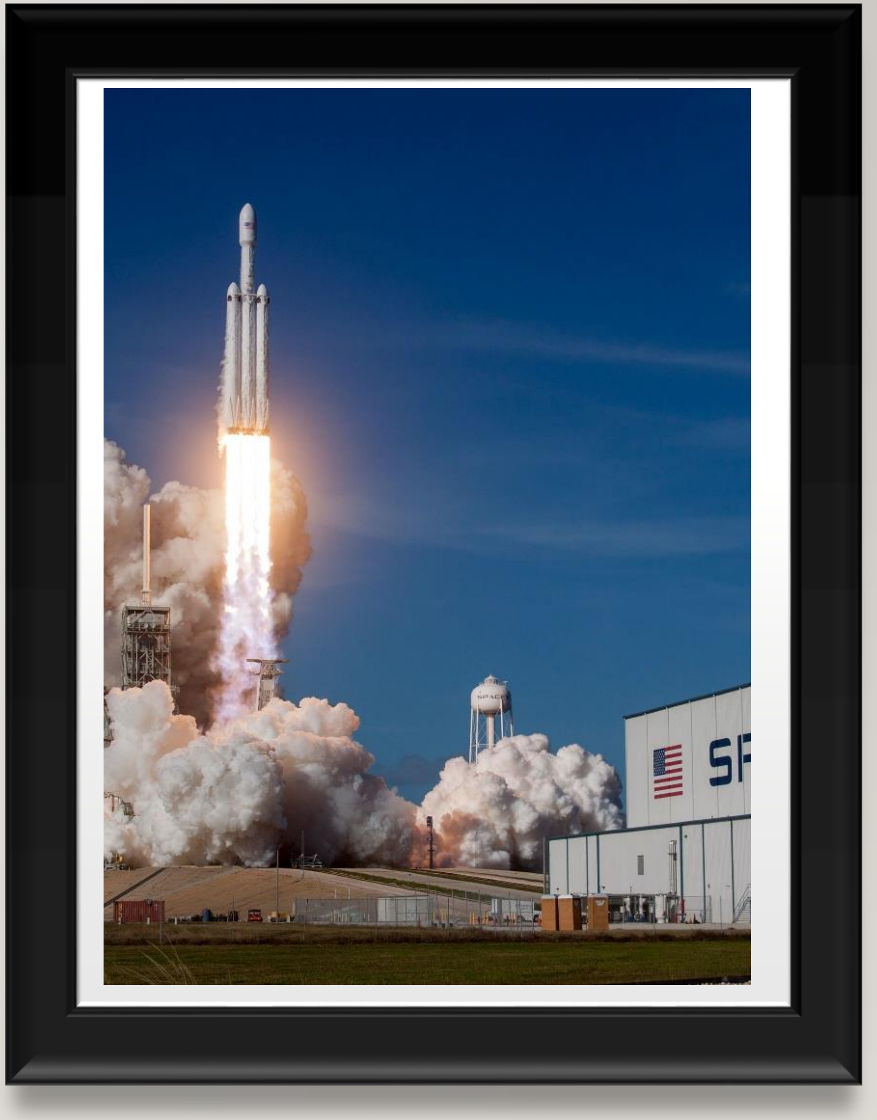
DATA SCIENCE CAPSTONE PROJECT

Mohammaderfan Maleki



OUTLINE

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

EXECUTIVE SUMMARY

- Summary of methodologies
  - Data Collecting
  - Data Wrangling
  - Exploratory Data Analysis and Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis results
  - Interactive analytics results
  - Best models for Predictive Analysis

INTRODUCTION

- Project background and context:

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 we are going to address:
  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

METHODOLOGY

- Data Collection methodology:
  - SpaceX API
  - Web Scrapping (Wikipedia)
- Data wrangling

  Performing one hot encoding and dropping irrelevant columns
- Exploratory data analysis (EDA)

  Using visualization and SQL
- Interactive visual analytics

  Using Dash visualization and Folium
- Predictive analysis

  Using classification models

DATA COLLECTION

- The data was collected using 2 methods
  - First we performed data collection using SpaceX API.

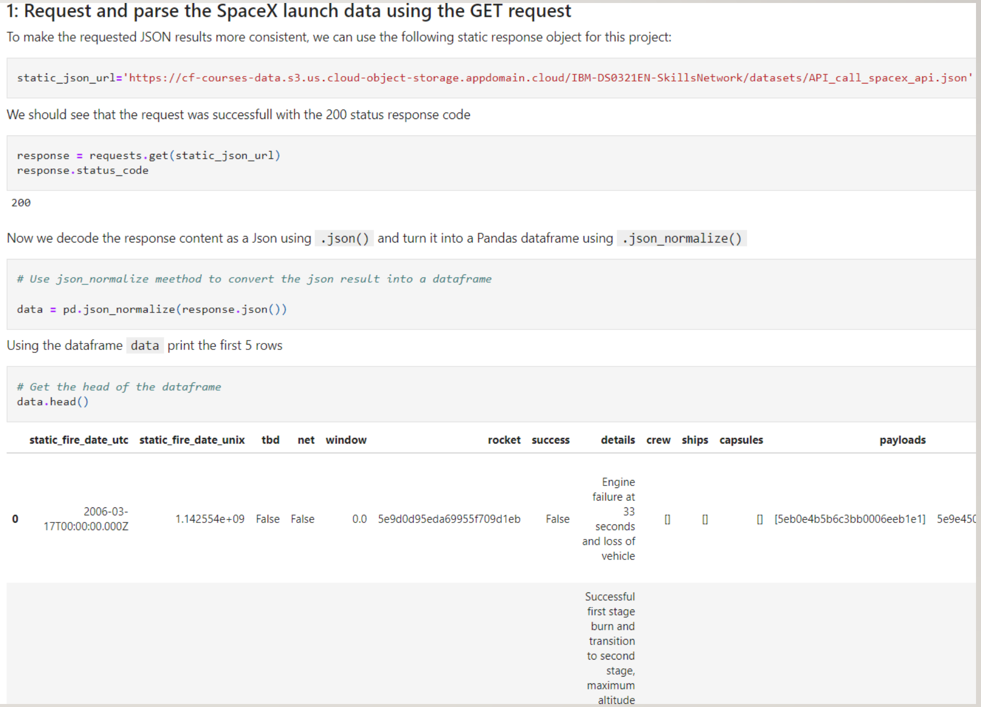
    we decoded the response as a JSON . Then we converted the JSON into a pandas DataFrame using.

    then cleaned the data, checked for missing values and fill in missing values where necessary.
  - In addition, we performed web scraping from Wikipedia for Falcon 9 launch records using BeautifulSoup package.

    The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas DataFrame for future analysis.

DATA COLLECTION – SPACEX API

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.
- This is the link to the notebook
    - https://github.com/erfanma/SpaceX-Data-Science-Project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we can use the following static response object for this project:

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

We should see that the request was successfull with the 200 status response code

```
response = requests.get(static_json_url)
response.status_code
```

```
200
```

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())
```

Using the dataframe `data` print the first 5 rows

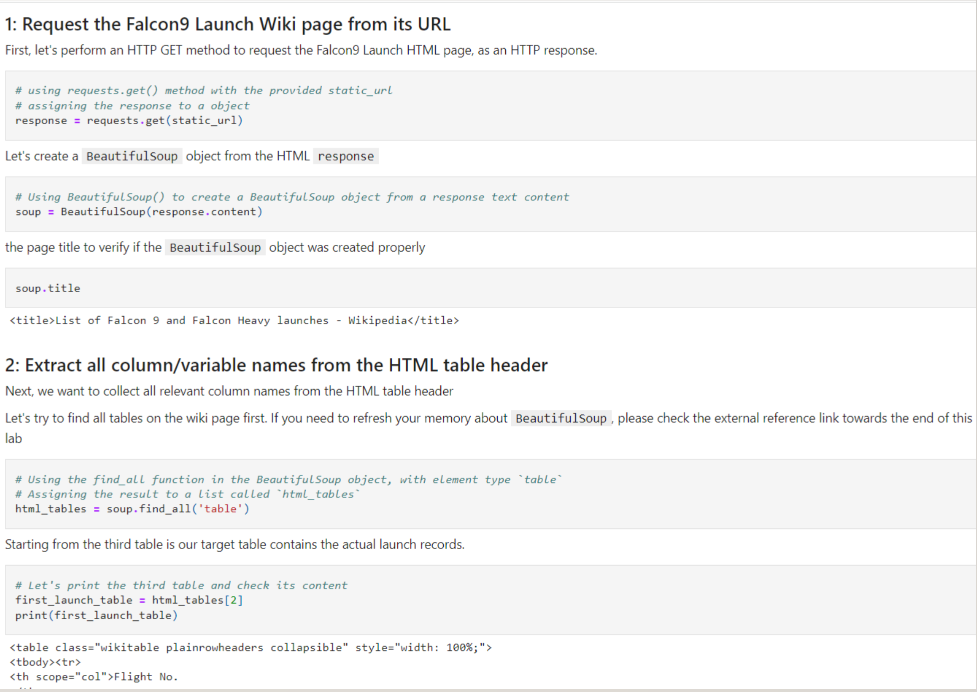
```
# Get the head of the dataframe
data.head()
```

	static_fire_date_utc	static_fire_date_unix	tbd	net	window	rocket	success	details	crew	ships	capsules	payloads	
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	False	0.0	5e9d0d95eda69955f709d1eb	False	Engine failure at 33 seconds and loss of vehicle	[]	[]	[]	[5eb0e4b5b6c3bb0006eeb1e1]	5e9e450
								Successful first stage burn and transition to second stage, maximum altitude					

DATA COLLECTION – WEB SCRAPPING

- We performed web scrapping to collect data of Falcon 9 launch records with BeautifulSoup package.
- We parsed the table and converted it into a pandas dataframe.
- Data was collected from Wikipedia
- Notebook url:

https://github.com/erfanma/SpaceX-Data-Science-Project/blob/main/jupyter-labs-webscraping.ipynb



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.

```
# using requests.get() method with the provided static_url
# assigning the response to a object
response = requests.get(static_url)
```

Let's create a `BeautifulSoup` object from the HTML `response`

```
# Using BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.content)
```

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

```
soup.title
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

```
# Using the find_all function in the BeautifulSoup object, with element type `table`
# Assigning 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.

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

```
<table class="wikitable plainrowheaders collapsible" style="width: 100%;">
<tbody><tr>
<th scope="col">Flight No.
```

DATA WRANGLING

- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- This is the notebook url:

https://github.com/erfanma/SpaceX-Data-Science-Project/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA WITH DATA VISUALIZATION

- The payload mass looked so important so the first chart that we created was FlightNumber vs. PayloadMass chart.
- We see that different launch sites have different success rates. So the next chart was created to Visualize the relationship between Payload and Launch Site.
- As we know they are many different orbit types. Next charts were created to observe the success rate of each of these orbit types and the relationship between FlightNumber and Orbit type.
- Payload and orbit type may be related too. So the next chart was created to visualize the relationship between Payload and Orbit type.
- In the end we visualized the Launch success yearly trend
- This is the URL to the notebook:

https://github.com/erfanma/SpaceX-Data-Science-Project/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA WITH SQL

- We loaded the SpaceX dataset into IBM database without leaving the jupyter notebook.
- We applied EDA with SQL to get insight from the data. Our queries were made to address following questions:
  - Names of the unique launch sites in the space mission
  - Total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - the date when the first successful landing outcome in ground pad was achieved
  - names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.
- The link to the notebook is:

https://github.com/erfanma/SpaceX-Data-Science-Project/blob/main/EDA%20with%20SQL.ipynb

BUILD AN INTERACTIVE MAP WITH FOLIUM

- Coordinates are just plain numbers that can not give you any intuitive insights about where are those launch sites. So we decided to mark all launch sites on a map. We added map objects such as markers, circles, lines and clusters to mark the success or failure of launches for each site on the folium map with Folium objects we could have a better insight about the coordinates of launch sites.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distances between a launch site to its proximities. We did this to answer the following questions:
  - Are launch sites in close proximity to railways?
  - Are launch sites in close proximity to highways?
  - Are launch sites in close proximity to coastline?
  - Do launch sites keep certain distance away from cities?
- This is the URL of the notebook:https://github.com/erfanma/SpaceX-Data-Science-Project/blob/main/lab_jupyter_launch_site_location.ipynb

BUILD A DASHBOARD WITH PLOTLY DASH

- We built an interactive dashboard with Plotly dash
- We plotted pie chart showing the total success launches by a certain site or all sites (you can change it by the dropdown menu)
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version (you can change the payload range).
- The link to the app is

https://github.com/erfanma/SpaceX-Data-Science-Project/blob/main/spacex_dash_app.py

PREDICTIVE ANALYSIS (CLASSIFICATION)

- We loaded the data using numpy and pandas library. Transformed the data, splited our dataset into training and testing dataset.
- We built different machine learning models and tuned different hyper parameters using GridSearchCV.
- We used accuracy as a metric for our model performance, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- The link to the notebook is:

https://github.com/erfanma/SpaceX-Data-Science-Project/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

RESULTS

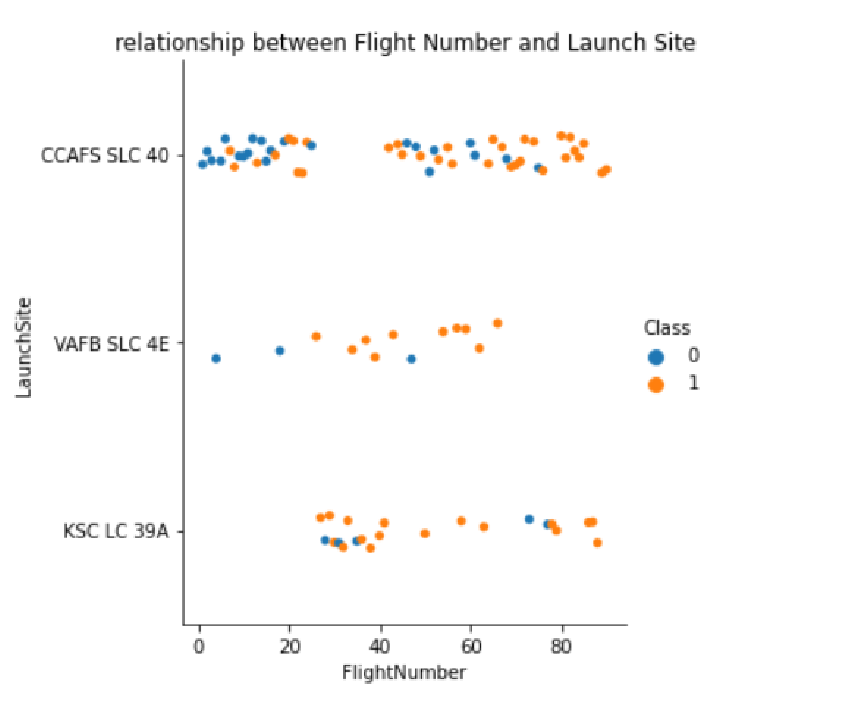
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

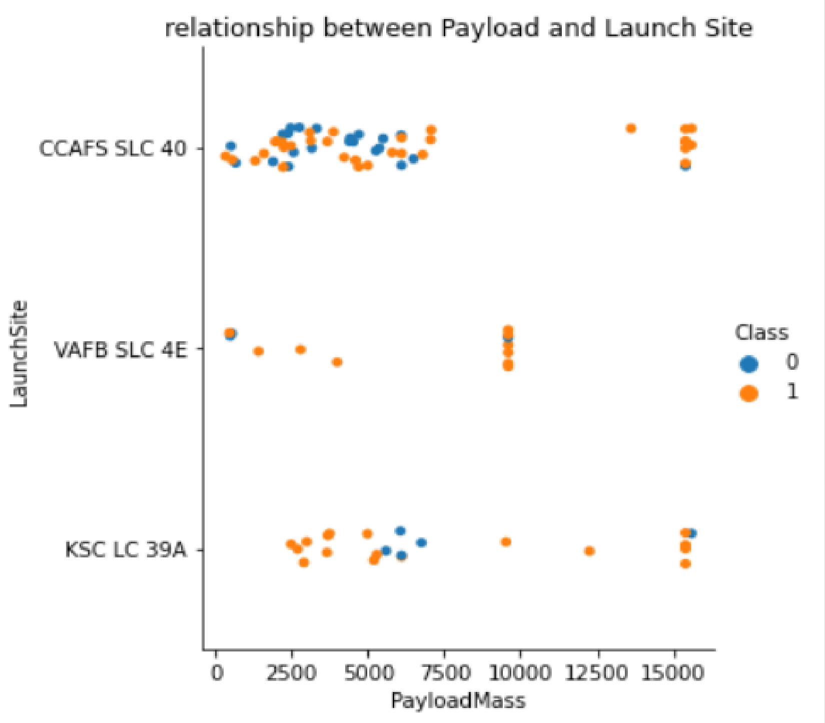
FLIGHT NUMBER VS. LAUNCH SITE

- From this chart we can understand that each launch site has a different success rate, so launch site is an important feature for us
- Another conclusion from this chart is that with more flights each site have, the success rate increases.



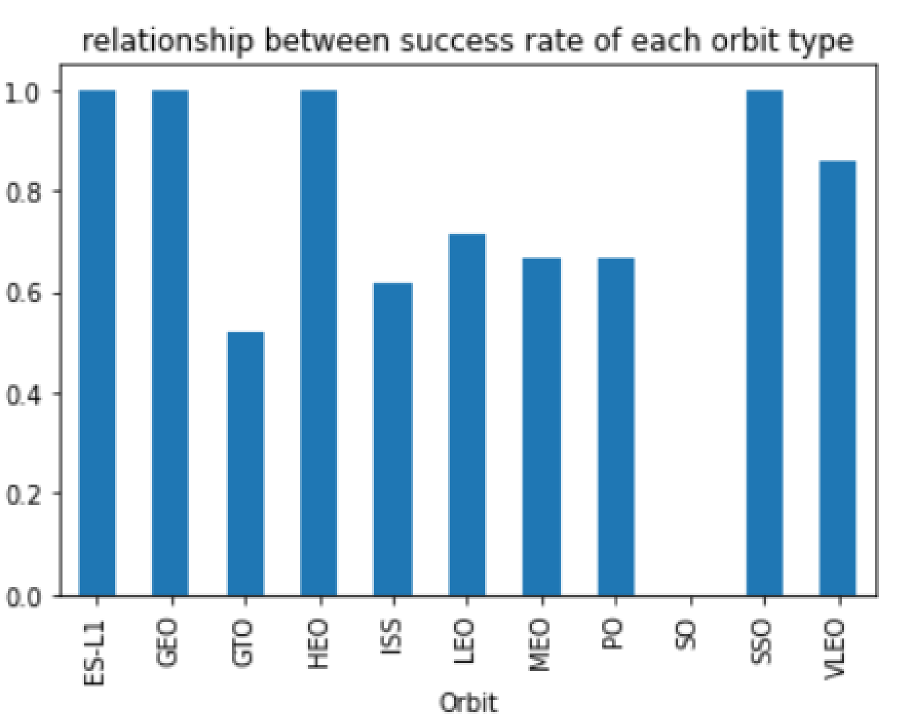
PAYLOAD VS. LAUNCH SITE

- From this chart first we can conclude that the launch site VAFB SLC 4E has no rocket launched for payload mass greater than 10000
- Second result would be that different payload mass has different result for each launch site



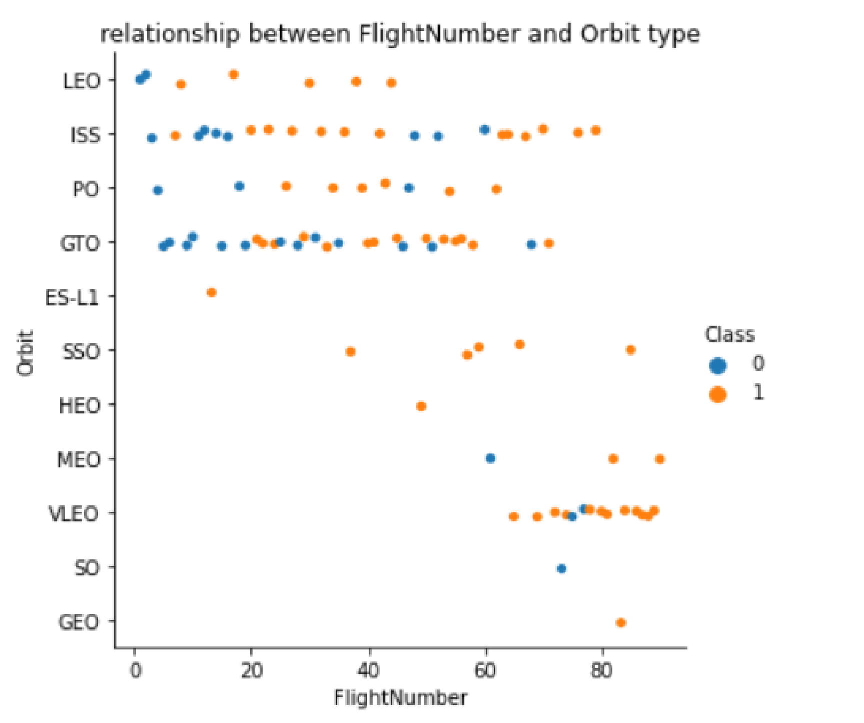
SUCCESS RATE VS. ORBIT TYPE

- From this chart we can understand some orbit types have lower success rate compared to other orbit types.
- For instance orbit types of ES-L1 , GEO, HEO, SSO and VLEO have a higher success rate compared to other orbit types



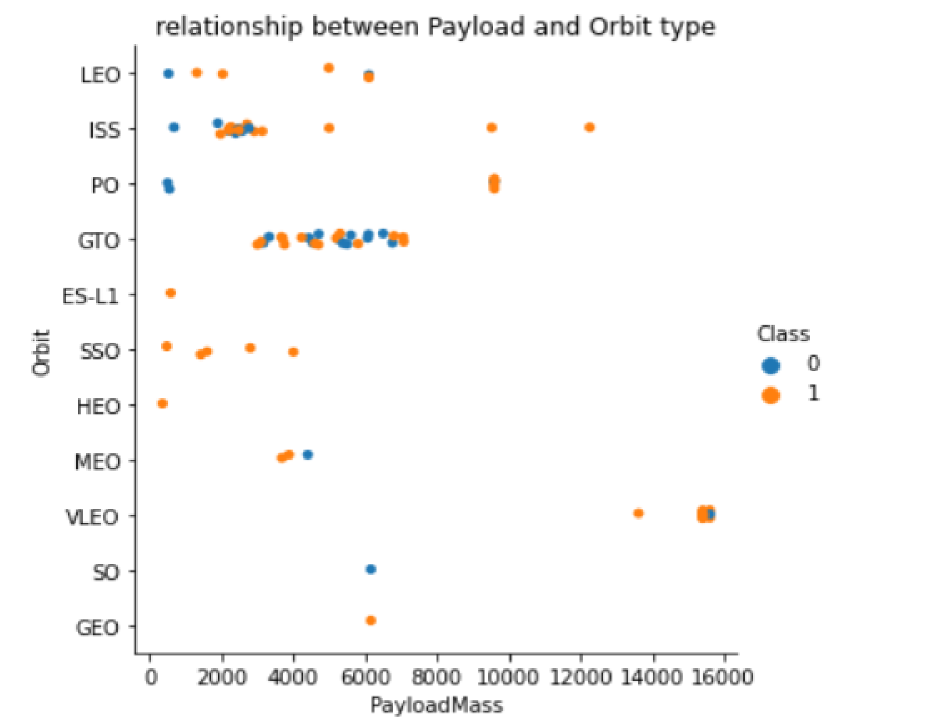
FLIGHT NUMBER VS. ORBIT TYPE

- From this chart we can see number of flights have different effect on each orbit type.
- For instance We can see that in 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.



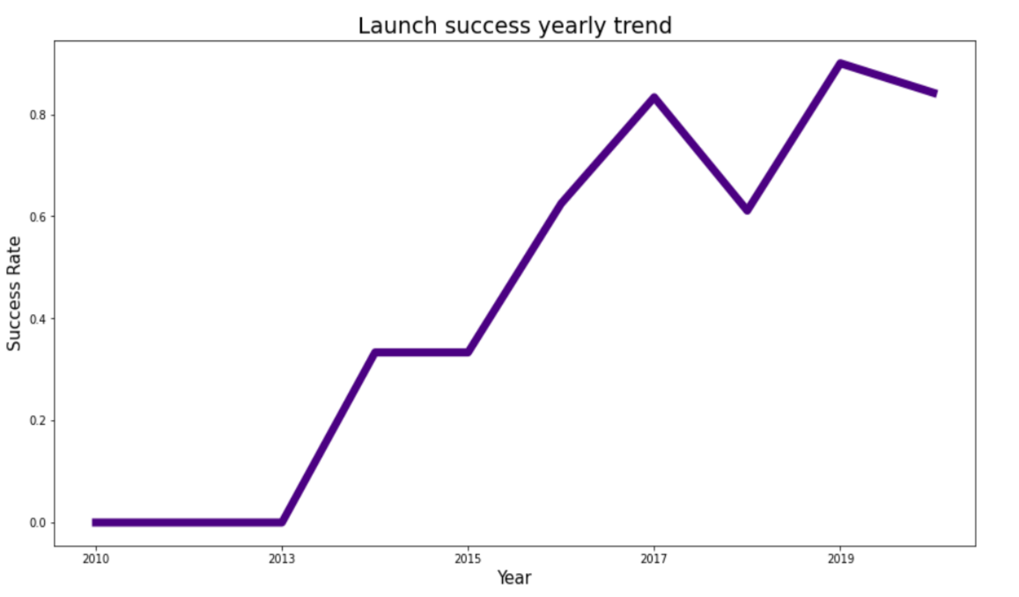
PAYLOAD VS. ORBIT TYPE

- We can see that the effect of payload mass is different for each orbit type.
- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.



LAUNCH SUCCESS YEARLY TREND

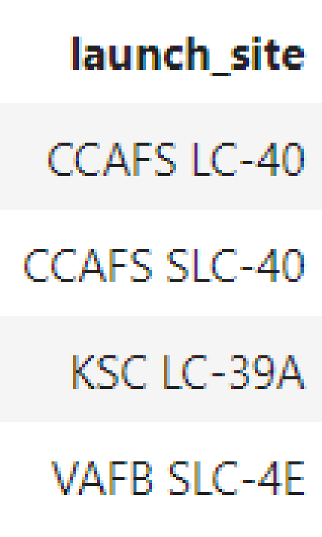
- We can easily observe that the success rate since 2013 kept increasing till 2020



ALL LAUNCH SITE NAMES

- We can get all launch site names using keyword DISTINCT.

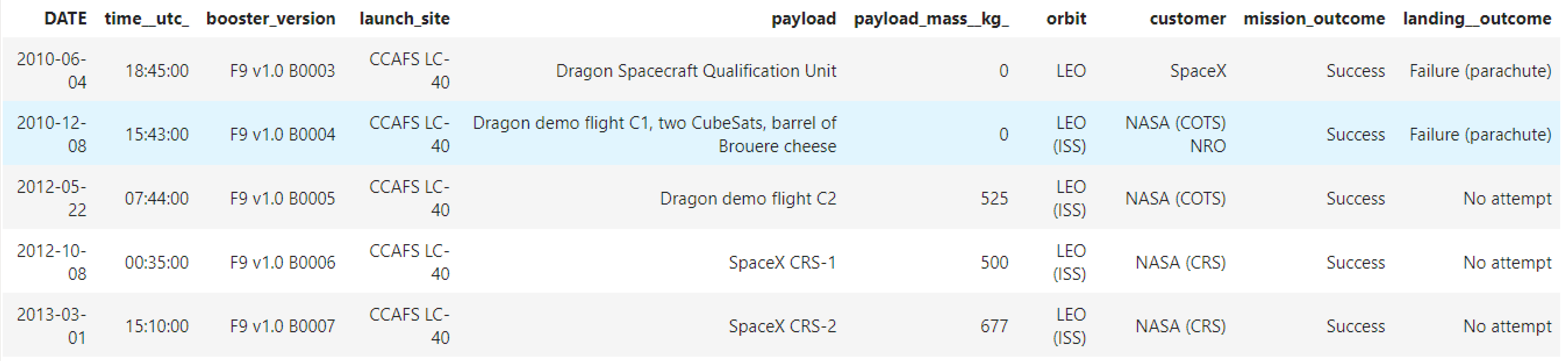
```
select DISTINCT LAUNCH_SITE from SPACEXDATASET
```



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

LAUNCH SITE NAMES BEGIN WITH 'CCA'

```
select * from SPACEXDATASET 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

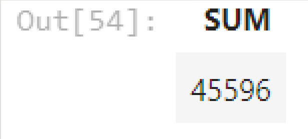
We used the above query to display 5 records where launch sites names begin with `CCA`

TOTAL PAYLOAD MASS

The quert to get total payload mass:

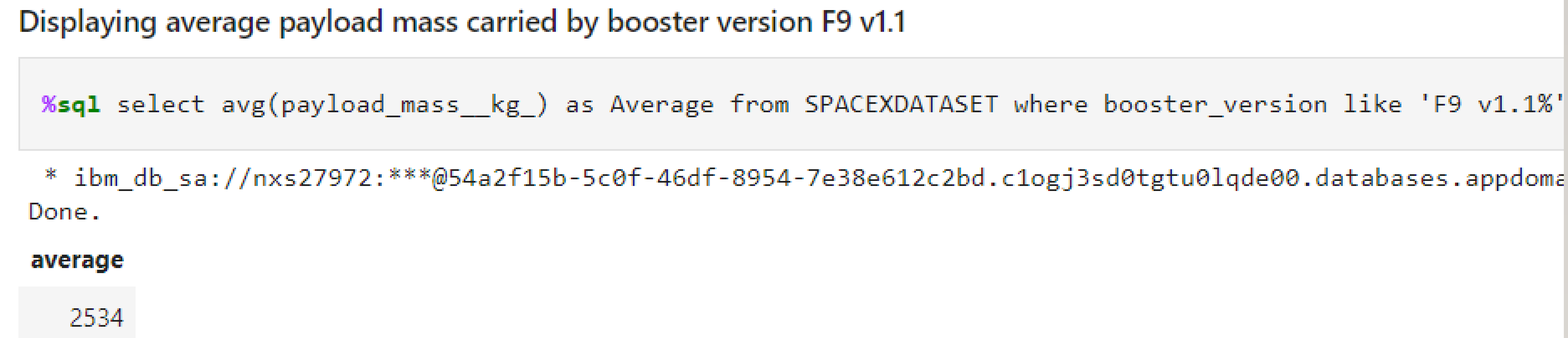
```
select sum(payload_mass__kg_) as sum from SPACEXDATASET where customer like 'NASA (CRS)'
```

Output of the query:



```
Out[54]:  SUM
        45596
```

AVERAGE PAYLOAD MASS BY F9 V1.1



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

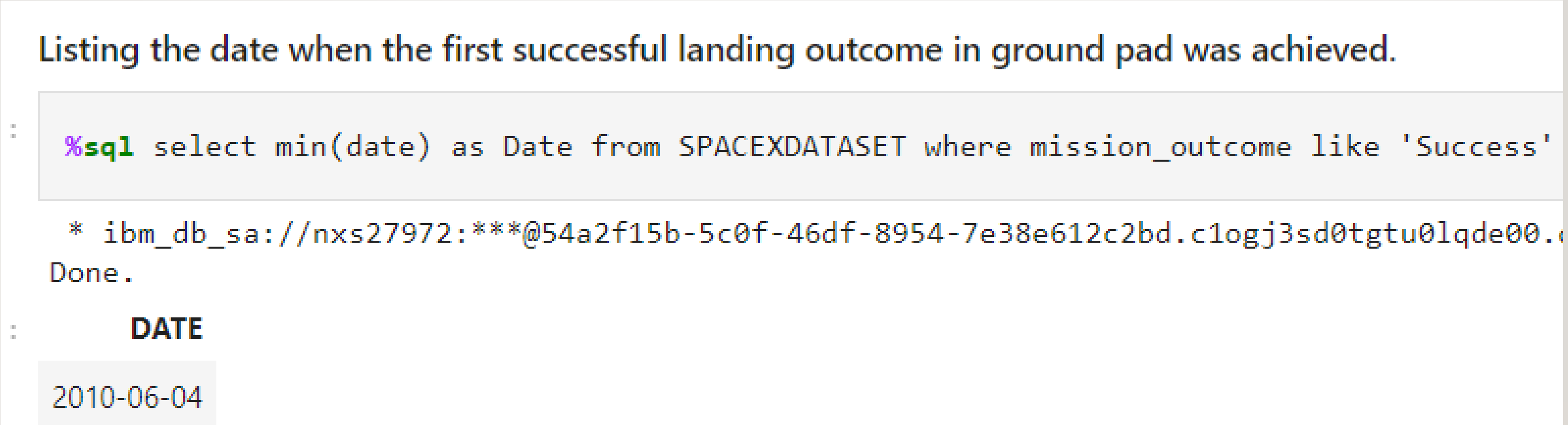
```
%sql select avg(payload_mass__kg_) as Average from SPACEXDATASET where booster_version like 'F9 v1.1%'
```

```
 * ibm_db_sa://nxs27972:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdoma
Done.
```

average
2534

We used keyword avg to get the average payload mass.

FIRST SUCCESSFUL GROUND LANDING DATE



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

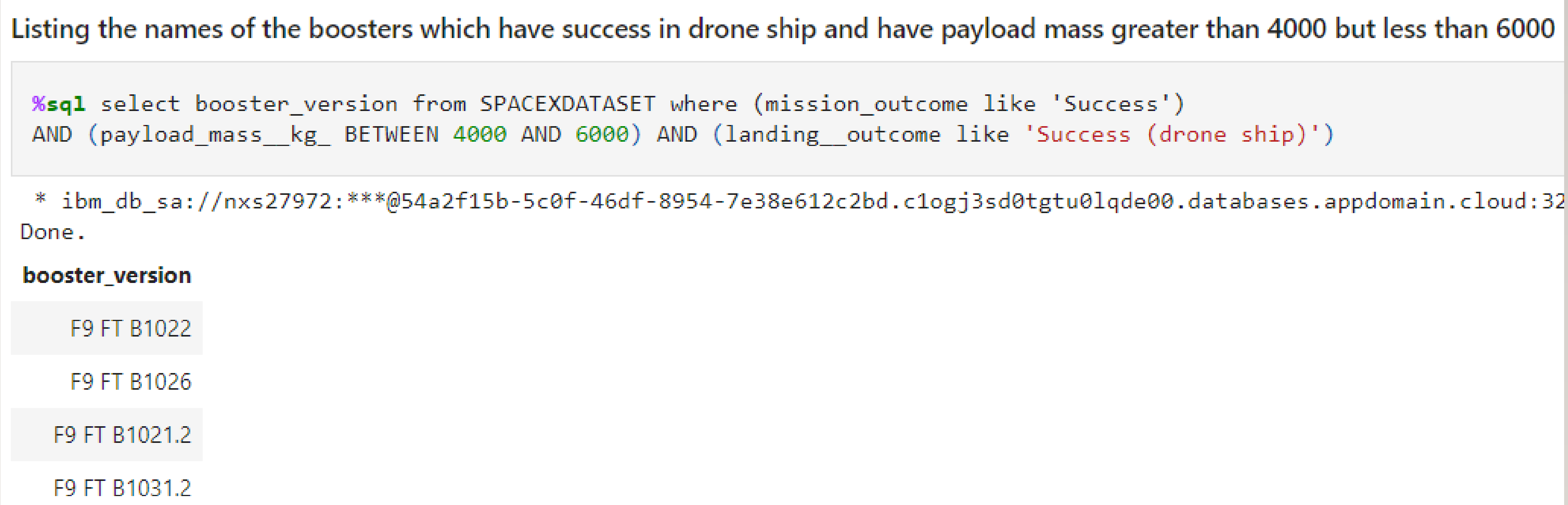
```
%sql select min(date) as Date from SPACEXDATASET where mission_outcome like 'Success'
```

```
 * ibm_db_sa://nxs27972:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.
Done.
```

DATE
2010-06-04

For getting the first successful ground landing date we used keyword min on date field.

SUCCESSFUL DRONE SHIP LANDING WITH PAYLOAD BETWEEN 4000 AND 6000



Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

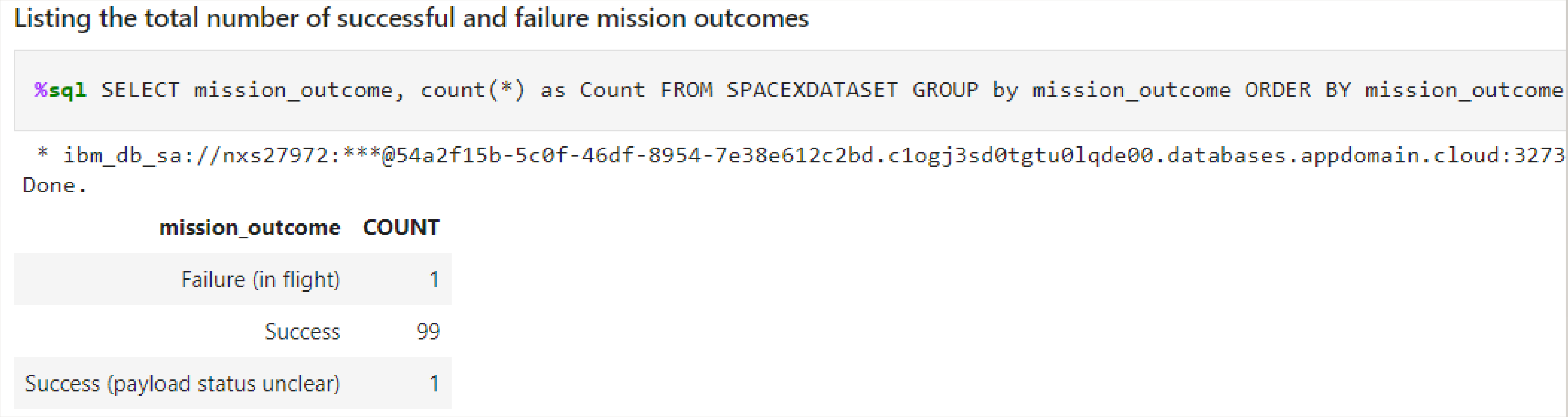
```
%sql select booster_version from SPACEXDATASET where (mission_outcome like 'Success')
AND (payload_mass__kg_ BETWEEN 4000 AND 6000) AND (landing__outcome like 'Success (drone ship)')
```

```
 * ibm_db_sa://nxs27972:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32
Done.
```

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

As we can see there are 4 booster version for above query

TOTAL NUMBER OF SUCCESSFUL AND FAILURE MISSION OUTCOMES



Listing the total number of successful and failure mission outcomes

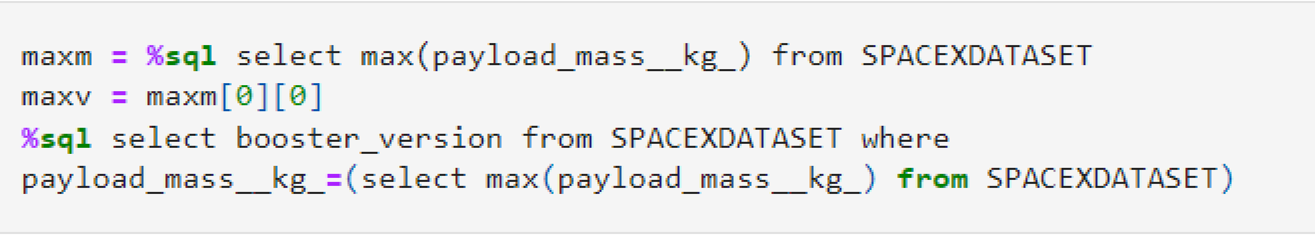
```
%sql SELECT mission_outcome, count(*) as Count FROM SPACEXDATASET GROUP by mission_outcome ORDER BY mission_outcome
```

```
 * ibm_db_sa://nxs27972:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3273
Done.
```

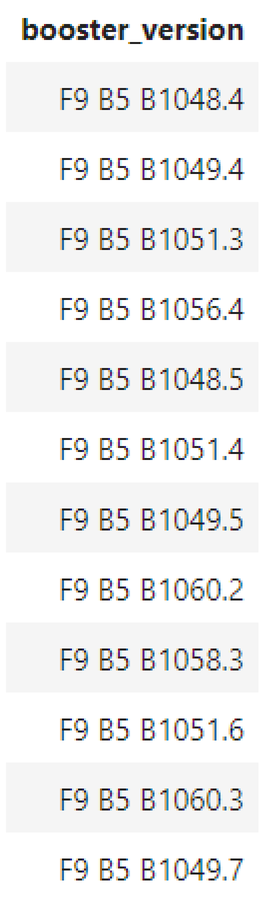
mission_outcome	COUNT
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

BOOSTERS CARRIED MAXIMUM PAYLOAD

For this part we actually used 2 query. 1 to get maximum payload mass
And used the result for the final quert

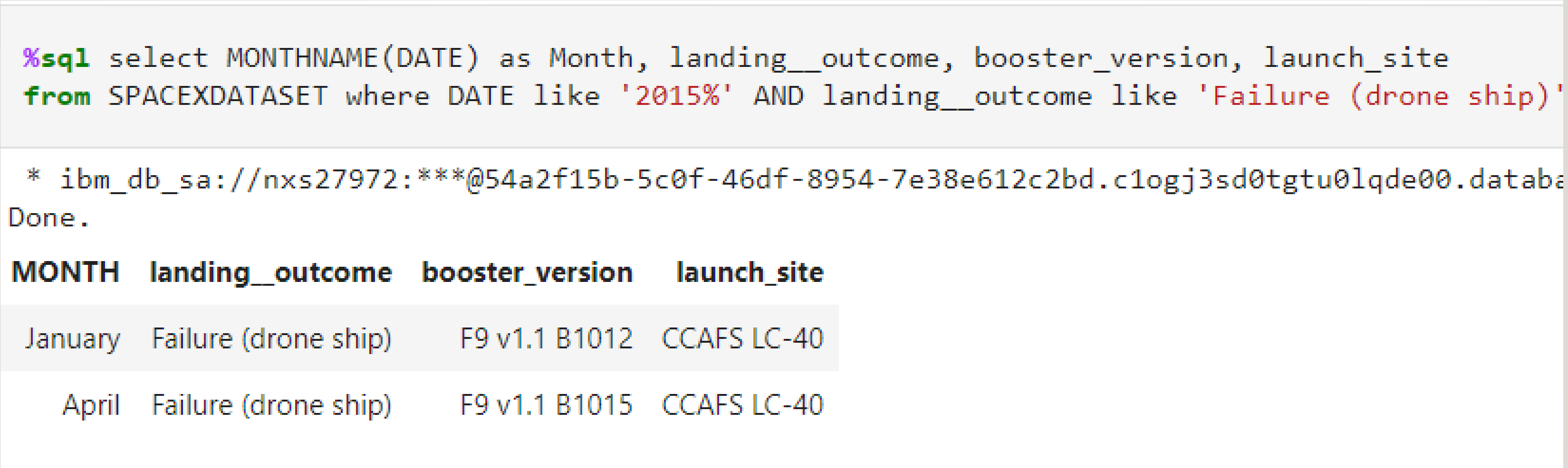


```
maxm = %sql select max(payload_mass__kg_) from SPACEXDATASET
maxv = maxm[0][0]
%sql select booster_version from SPACEXDATASET where
payload_mass__kg_=(select max(payload_mass__kg_) from SPACEXDATASET)
```



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



```
%sql select MONTHNAME(DATE) as Month, landing__outcome, booster_version, launch_site
from SPACEXDATASET where DATE like '2015%' AND landing__outcome like 'Failure (drone ship)'
```

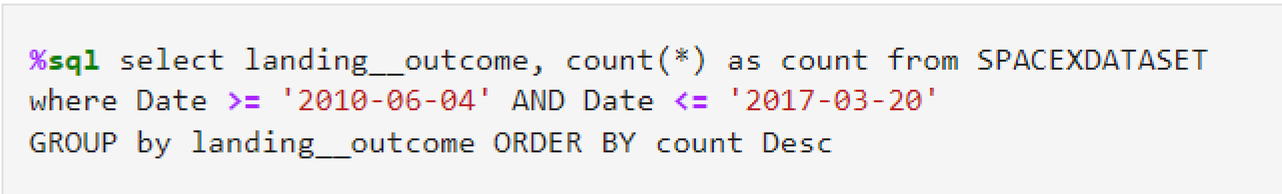
```
 * ibm_db_sa://nxs27972:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databa
Done.
```

MONTH	landing__outcome	booster_version	launch_site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

As we can see they were 2 failure (drone ship) in 2015. both were accrued in launch site CCAFS LC-40. one was accrued in January and the other one in april.

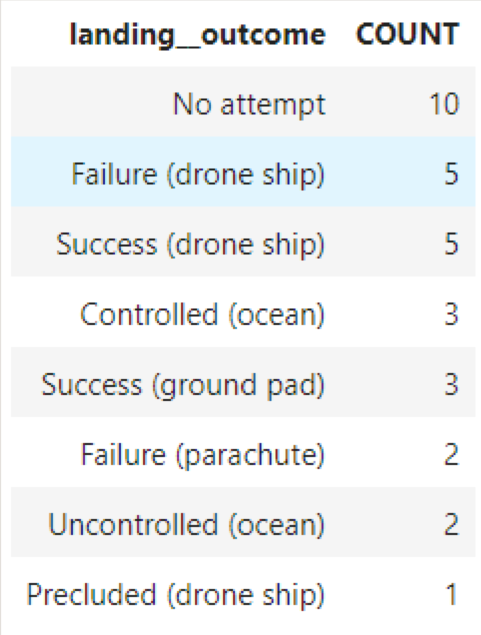
RANK LANDING OUTCOMES BETWEEN 2010-06-04 AND 2017-03-20

The query:



```
%sql select landing__outcome, count(*) as count from SPACEXDATASET
where Date >= '2010-06-04' AND Date <= '2017-03-20'
GROUP by landing__outcome ORDER BY count Desc
```

The result:



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

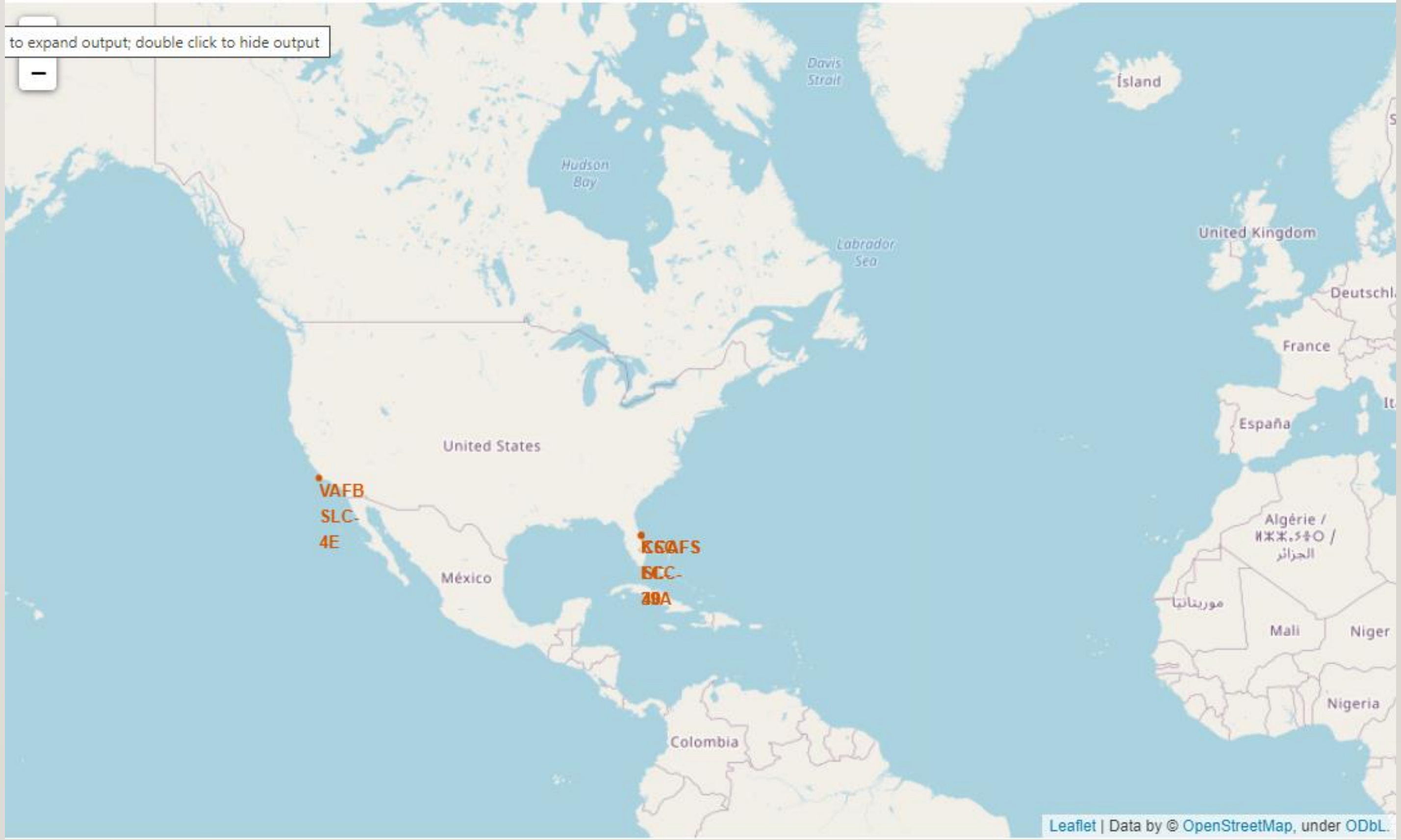
Section 3

Launch Sites Proximities Analysis

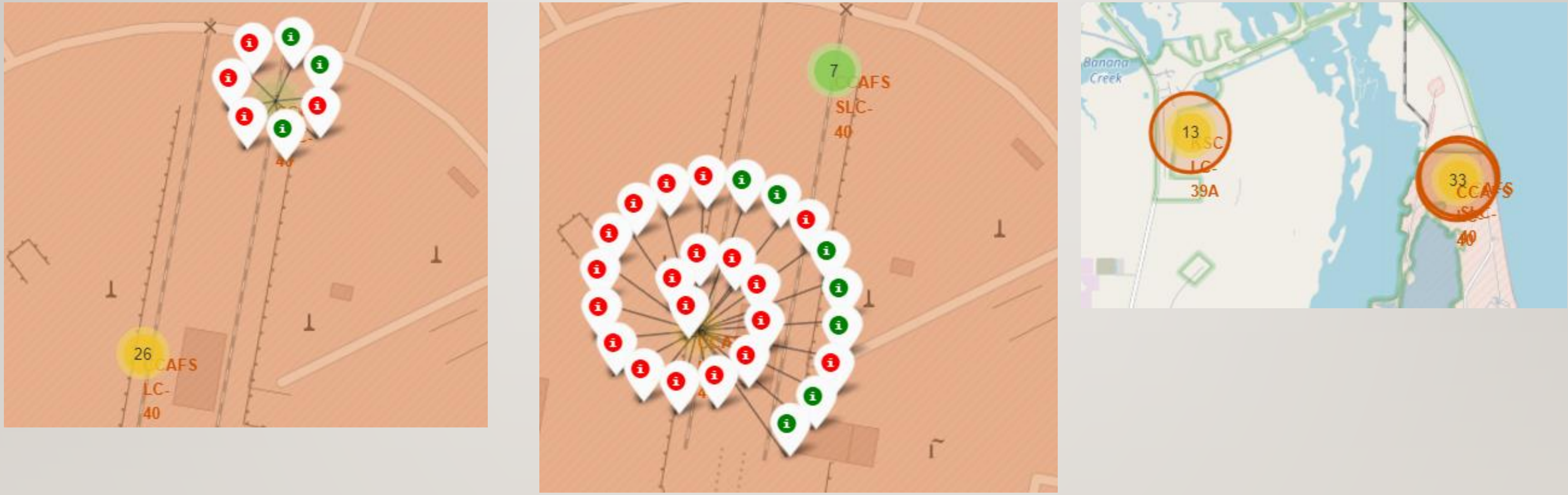
LAUNCH SITES GLOBAL MAP MARKERS

We can see that the SpaceX launch sites are in USA.

The launch sites are all near The coast in California and Florida.

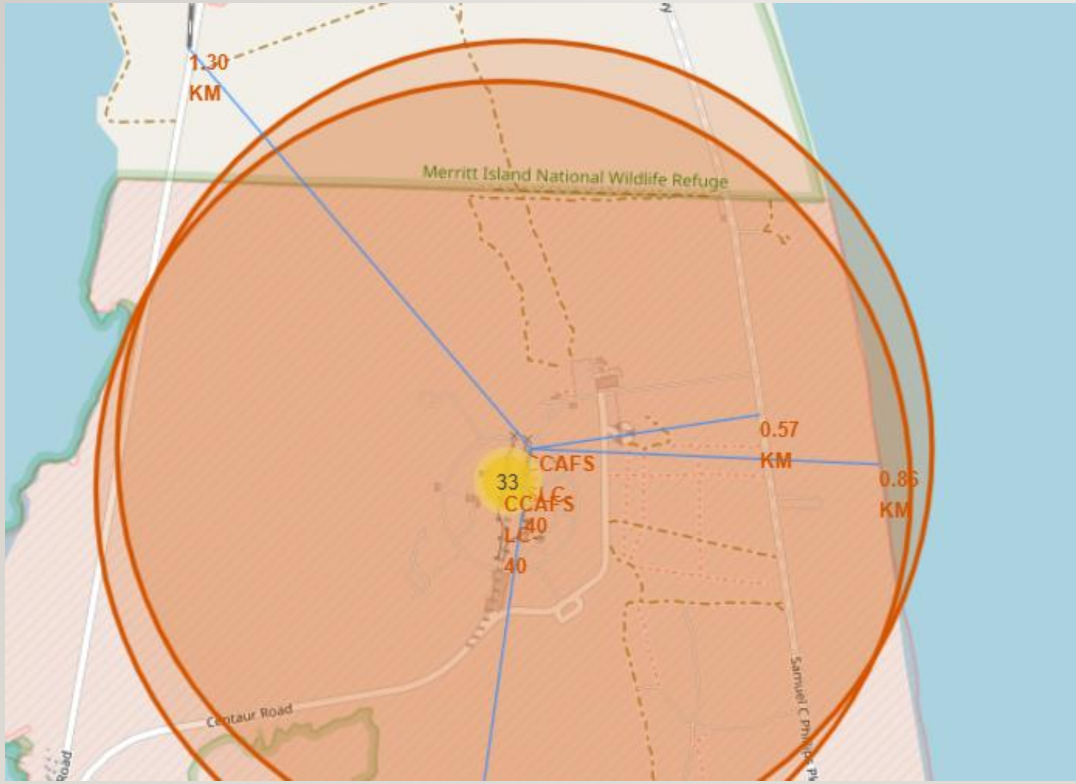


MARKERS SHOWING LAUNCH SITES WITH COLOR LABELS



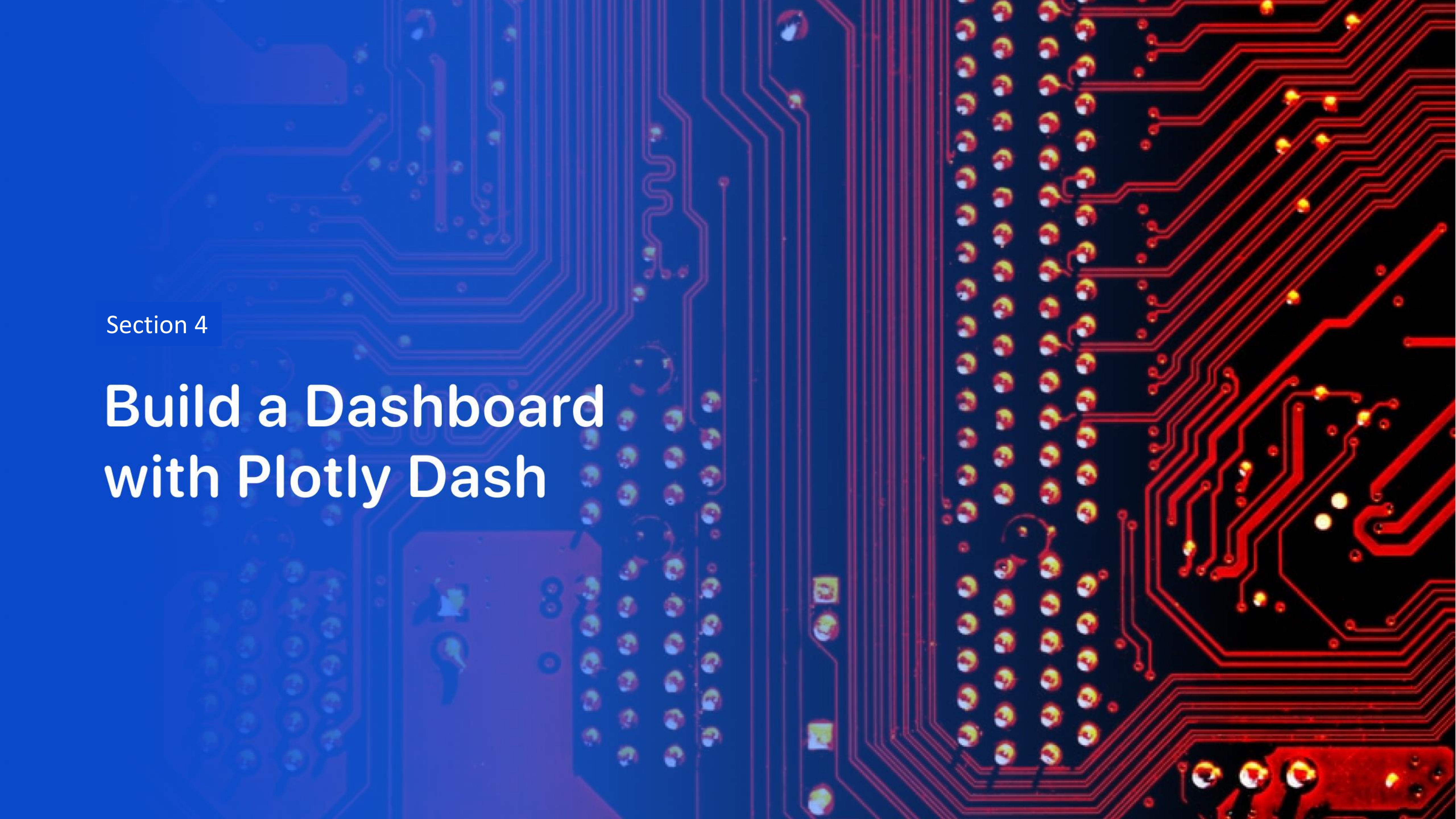
Green markers are representing successful classes and the red ones representing unsuccessful classes

LAUNCH SITE DISTANCE TO LANDMARKS



In this map we can observe the distance of launch sites to coast line, railway, Highway and a city.

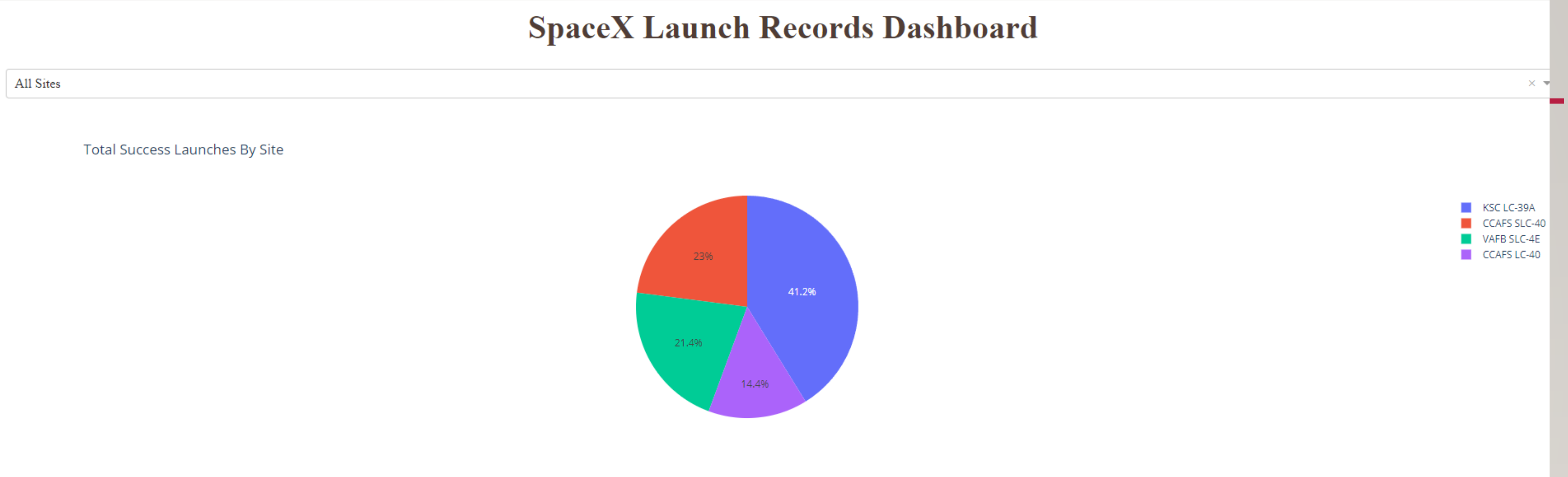
As we can observe, the launch site is very close to these landmarks.



Section 4

Build a Dashboard with Plotly Dash

PIE CHART SHOWING THE SUCCESS PERCENTAGE ACHIEVED BY EACH LAUNCH SITE



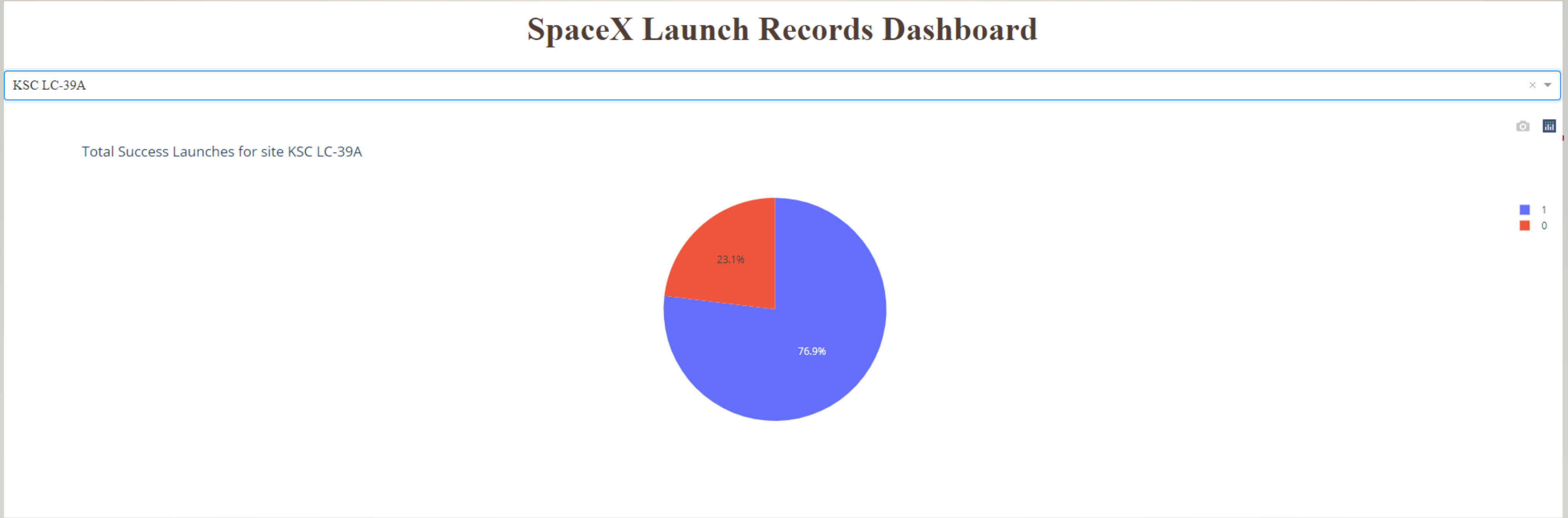
SpaceX Launch Records Dashboard

All Sites

Total Success Launches By Site

We can observe that the launch site KSC LC-39A has the highest success rate and the Launch site CCAFS SLC-40 has the lowest success rate.

PIE CHART SHOWING THE LAUNCH SITE WITH THE HIGHEST SUCCESS RATE



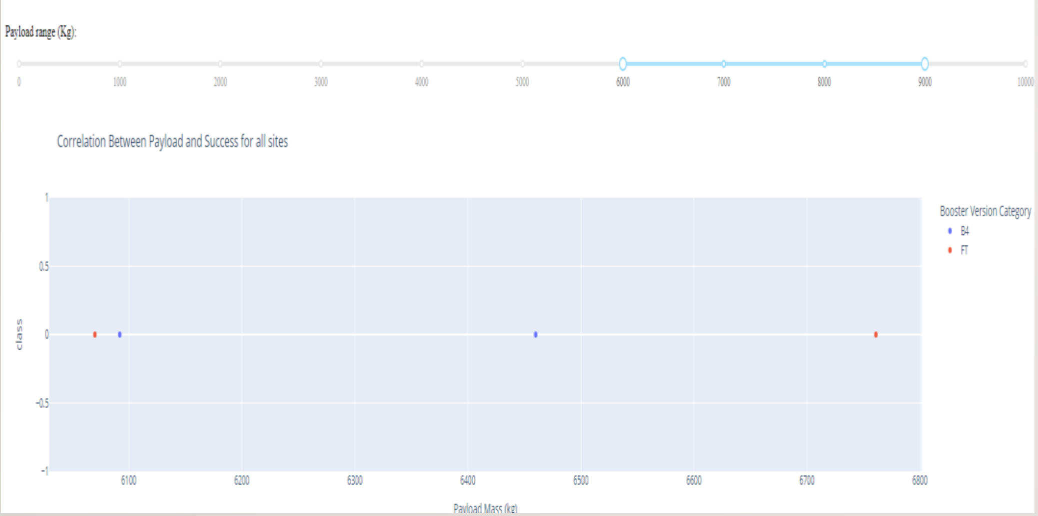
SpaceX Launch Records Dashboard

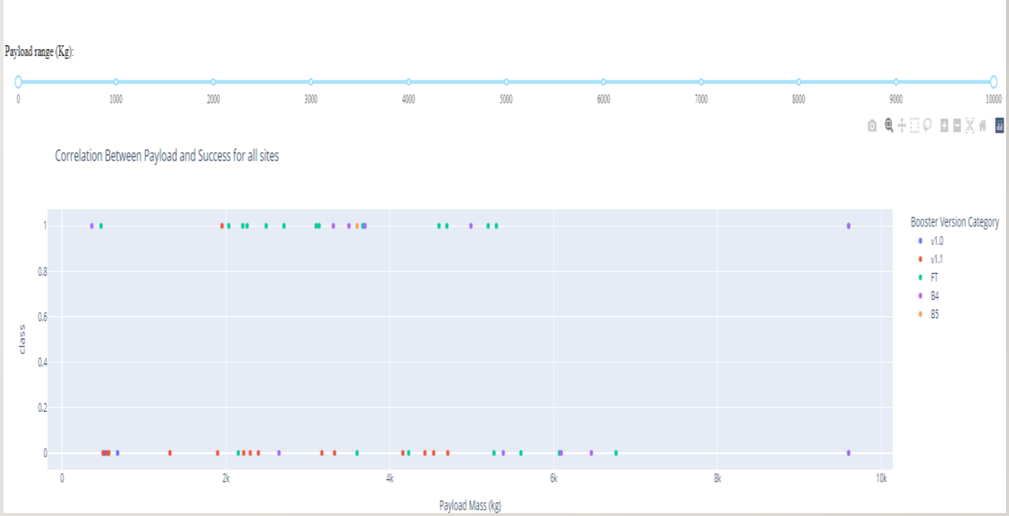
KSC LC-39A

Total Success Launches for site KSC LC-39A

We can see that the launch site KSC LC-39A has 76.9% launch success.

SCATTER PLOT OF PAYLOAD VS LAUNCH OUTCOME FOR ALL SITES WITH DIFFERENT PAYLOAD RANGE





From the above range we can
See that in the range of 6000 – 9000
We have no successful launch.

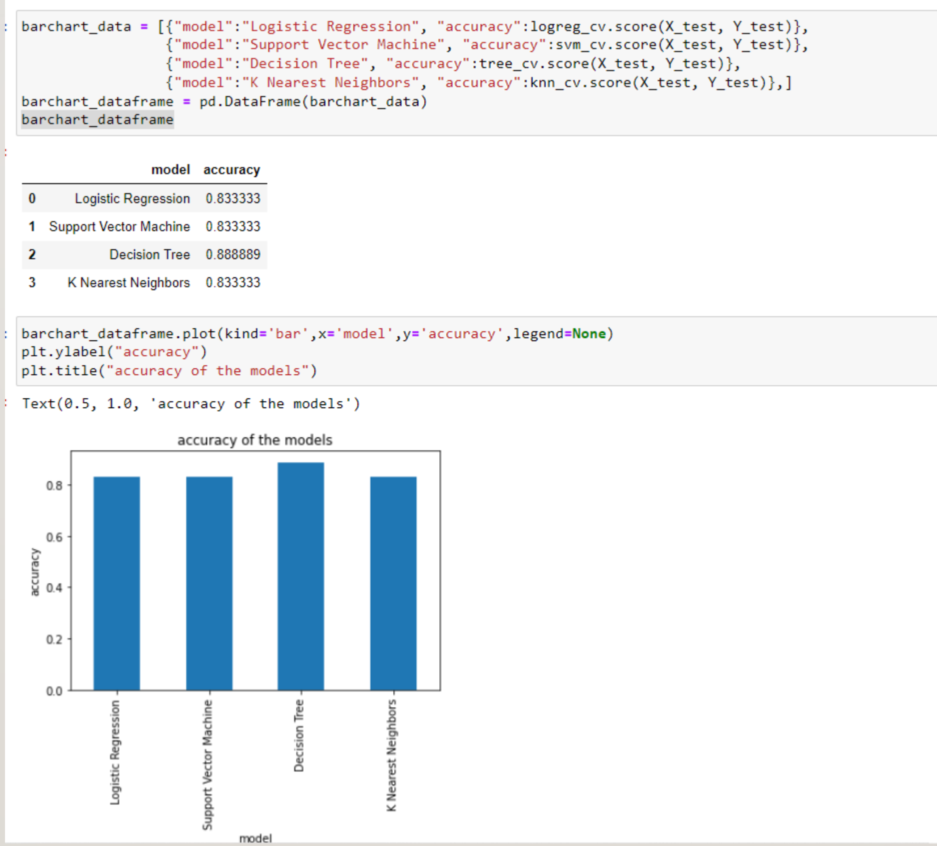
The above image is all of launches
With range of 0-10000

Section 5

Predictive Analysis (Classification)

CLASSIFICATION ACCURACY

As we can see,
The Decision Tree
Model has the highest
Accuracy among our
Classification models.



```
barchart_data = [{"model":"Logistic Regression", "accuracy":logreg_cv.score(X_test, Y_test)},
                 {"model":"Support Vector Machine", "accuracy":svm_cv.score(X_test, Y_test)},
                 {"model":"Decision Tree", "accuracy":tree_cv.score(X_test, Y_test)},
                 {"model":"K Nearest Neighbors", "accuracy":knn_cv.score(X_test, Y_test)},]
barchart_dataframe = pd.DataFrame(barchart_data)
barchart_dataframe
```

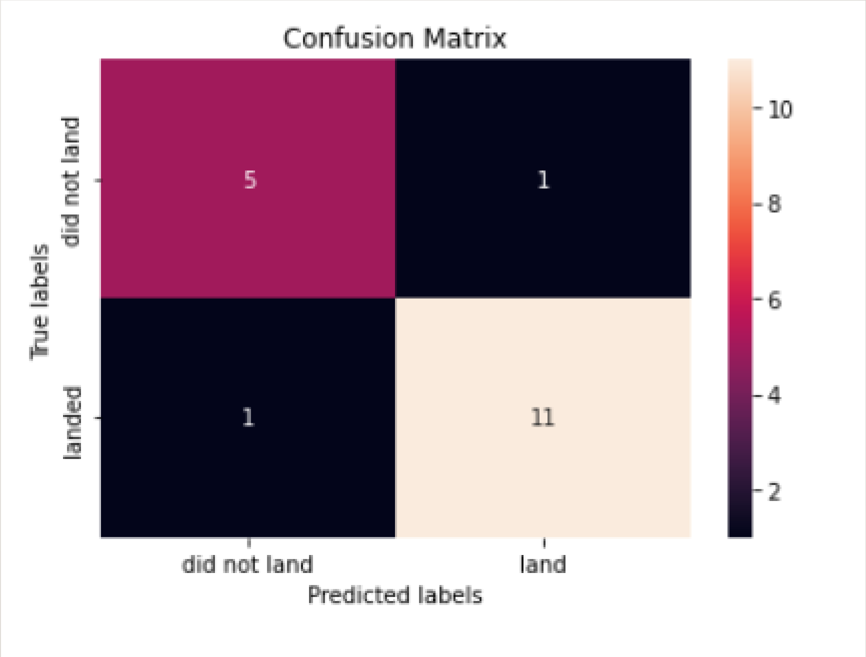
	model	accuracy
0	Logistic Regression	0.833333
1	Support Vector Machine	0.833333
2	Decision Tree	0.888889
3	K Nearest Neighbors	0.833333

```
barchart_dataframe.plot(kind='bar',x='model',y='accuracy',legend=None)
plt.ylabel("accuracy")
plt.title("accuracy of the models")
```

```
Text(0.5, 1.0, 'accuracy of the models')
```

CONFUSION MATRIX

- Confusion Matrix is s a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class.
- We can see we have 1 false negative and 1 false positive.



CONCLUSIONS

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- success rate started to increase from 2013 to 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the highest success rate.
- KSC LC-39A had the most successful rates compared to other launch sites.
- The Decision tree classifier is the best model for this task.

Thank you!