IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Yuliya Yunayeva>
<05/19/23>

Outline

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

Executive Summary

Summary of methodologies:

- Data collection through API and Web Scraping
- Data wrangling
- EDA (exploratory data analysis) with SQL Lite and Visualization
- Visual Analytics with Folium
- Machine Learning prediction

Summary of all results

- Exploratory Data analysis, visual analytics and machine learning prediction results

Introduction

- Space X 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 Space X 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 space X for a rocket launch. In this lab, you will create a machine learning pipeline to predict if the first stage will land given the data from the preceding labs

- Falcon 9 is a reusable, two-stage rocket designed and manufactured by SpaceX for the reliable and safe transport of people and payloads into Earth orbit and beyond. Falcon 9 is the world's first orbital class reusable rocket. Reusability allows SpaceX to refly the most expensive parts of the rocket, which in turn drives down the cost of space access.

- EDA was perform on the SpaceX data to find following:

- What attributes should be taken into account to understand whether the rocket land successfully or not?

- What machine learning machine should be used?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - The data was collected via SpaceX public API and also through Wikipedia (web scrappingg)
- Perform data wrangling
    - Data was pre-processed to be trained on the models
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - How to build, tune, evaluate classification models

Data Collection

Data sets were collected by following means:

- Using SPaceX public API;
- Wikipedia Space X page via web scrapping.
- Then data was cleaned and checked for any missing values.
- Then Beautiful Soup was used to perform web scrapping from Wikipedia

Data Collection – SpaceX API

- GitHub URL is here:

https://github.com/wooljemper/yy_repo/blob/main/1st_notebook.ipynb

- Data flow:

1) Get the data via Space X API
2) Clean the data
3) Data wrangling
4) Format the data
5) Export the data as a dataset

Now let's start requesting rocket launch data from SpaceX API with the following URL:

```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
In [7]: response = requests.get(spacex_url)
```

Check the content of the response

```
In [ ]: print(response.content)
```

Request and parse the SpaceX launch data using the GET request

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

```
In [9]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/da
```

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

```
In [10]: response.status_code
Out[10]: 200
```

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

```
In [11]: # Use json_normalize meethod to convert the json result into a dataframe
         response = requests.get(static_json_url).json()
         data = pd.json_normalize(response)
```

Using the dataframe `data` print the first 5 rows

```
In [12]: # Get the head of the dataframe
         data.head()
```

Data Collection - Scraping

- Request the data from Wikipedia
- Create a Beautiful Soup object
- Extract all tables, then name of columns
- Create a dictionary
- Create a Panda dataframe
- Export the dataset
- GitHub:
- https://github.com/wooljemper/yy_repo/blob/main/2nd_notebook_Webscrapping.ipynb

Request the Falcon9 Launch Wiki page from its URL

In [5]:
```
html_data = requests.get(static_url)
html_data.status_code
```

Out[5]:
```
200
```

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

In [6]:
```
# create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html_data.text, 'html.parser')
```

We'll print the page title to verify if the `BeautifulSoup` object was created properly

In [7]:
```
# Use soup.title attribute
soup.title
```

Out[7]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

Extract all column/variable names from the HTML table header

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

In [8]:
```
html_tables = soup.find_all('table')
```

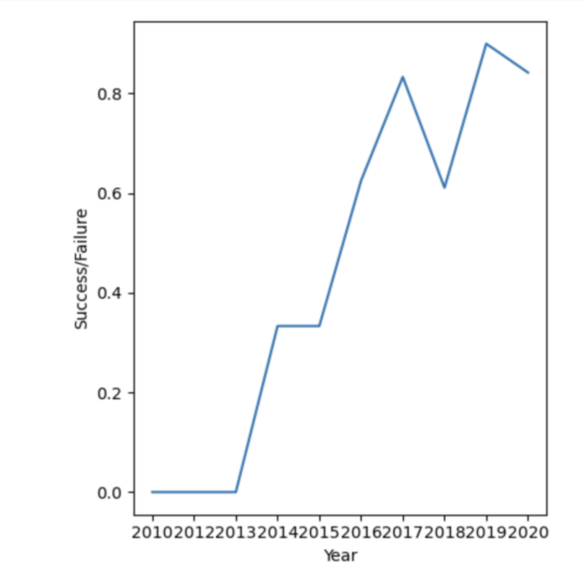
In [9]:
```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

Data Wrangling

- Data process was following:
- Check the data/null values
- Calculate the number of launches on each site
- Calculate the number of occurrence on each orbit
- Create a landing outcome label
- Export results as a new dataset into csv file
- GitHub link: link

EDA with Data Visualization

- Data was explored and visualized via following means:
- Flight number and launch site
- Flight number and payload
- Flight number vs Orbit
- Payload mass vs orbit
- Success rate of each orbit type
- GitHub: [link]



EDA with SQL

- Following queries were performed:
- the names of the unique launch sites
- the total payload mass carried by boosters launched by NASA (CRS)
- average payload mass carried by booster version F9 v1.1
- the date when the first succesful landing outcome in ground pad was acheived.
- the 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 names of the booster_versions which have carried the maximum payload mass
- the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
- the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order

- GitHub URL link

Build an Interactive Map with Folium

- All launch sites were marled on the map
- The success/failed launches for each site were marled on the map
- The distance between a launch site to its proximities was calculated
- Launch sites were close to railways, coastline and highways.
- GitHub URL link

Build a Dashboard with Plotly Dash

- On our dashboard we use pie chart and scatter plot
- For pie chart shows success rate for each launch site
- The scatter plot was used to demonstrate the relationship between Payload (mass kg) and the Class for different booster versions.

Predictive Analysis (Classification)

- For predictive analysis we used the pre-processed data, split it to training and testing sets.
- Following models were trained:
- SVM model
- Logistic Regression
- KNN model
- Decision tree model
- GitHub link

Results

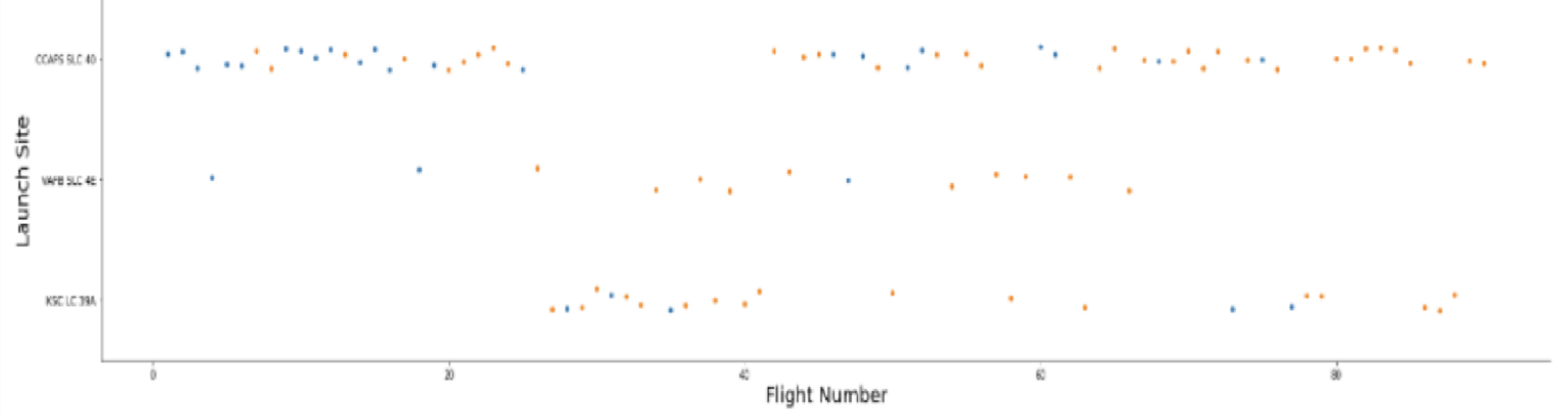
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

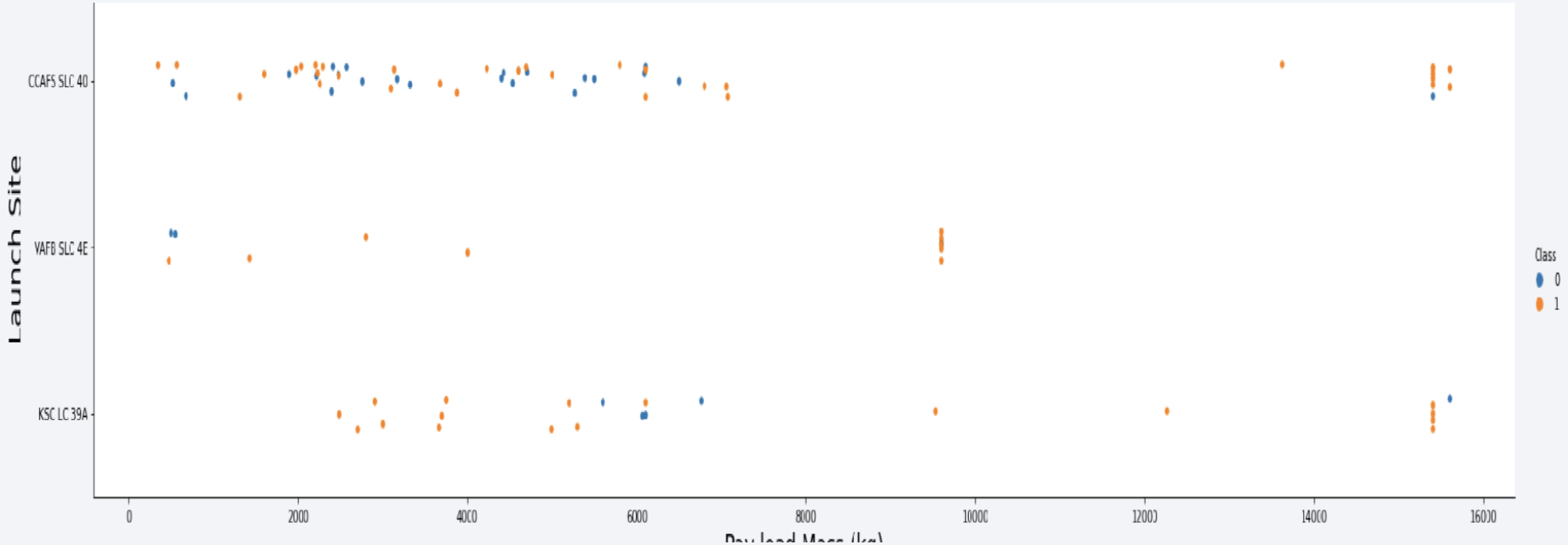
- A scatter plot of Flight Number vs. Launch Site



- The scatter plot above is demonstrating successful landing as orange dots and failed as blue ones.

Payload vs. Launch Site

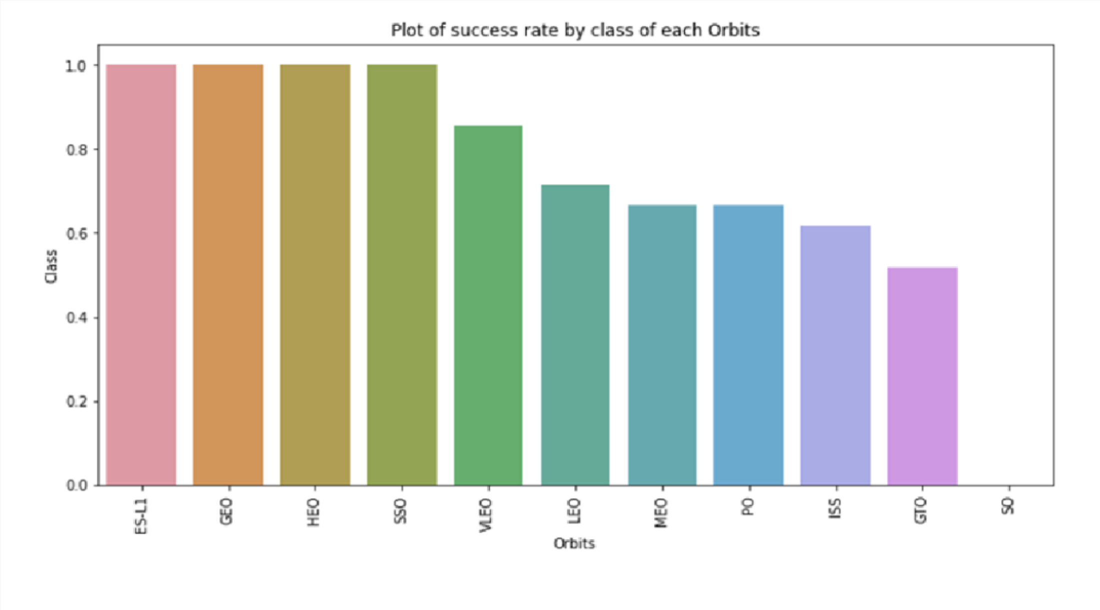
- Scatter plot of Payload vs. Launch Site



- The scatter plot above is demonstrating successful landing as orange dots and failed as blue ones.

Success Rate vs. Orbit Type

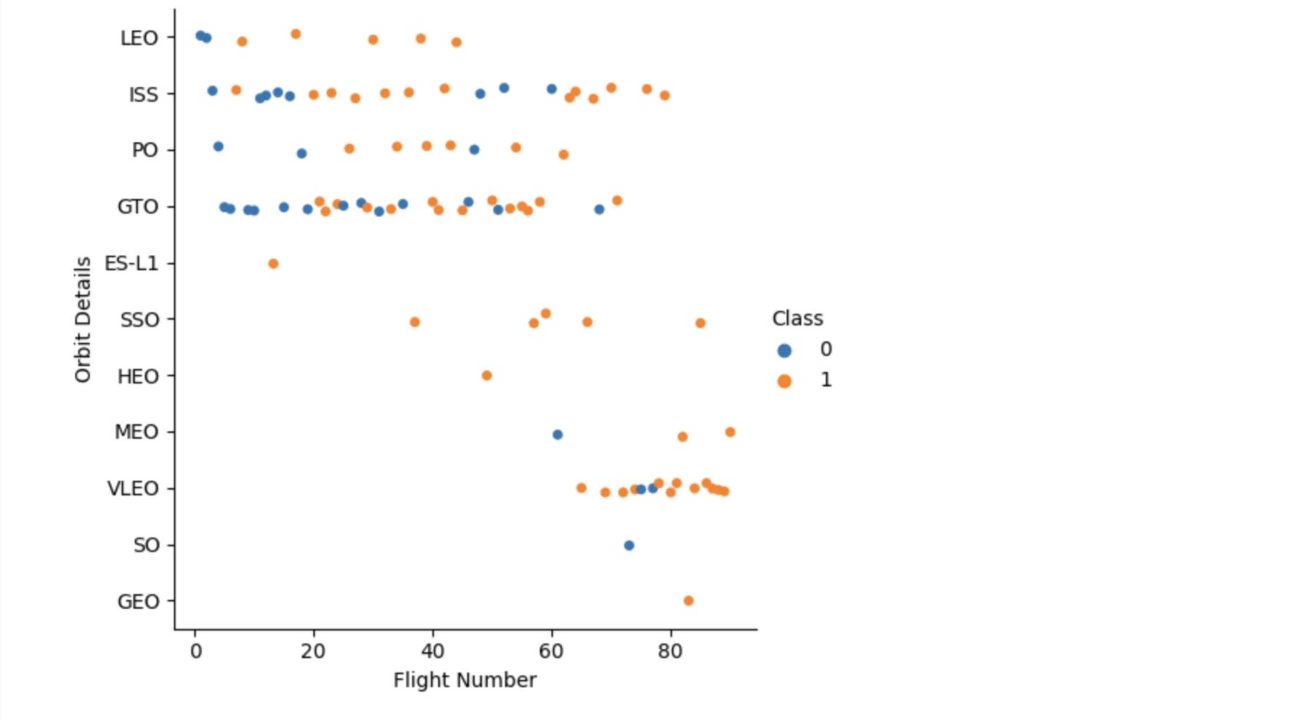
- Bar chart for the success rate of each orbit type



- From the chart above we can come to conclusions that the most successful rate have GEO, SSO, HEO and ES-L1.

Flight Number vs. Orbit Type

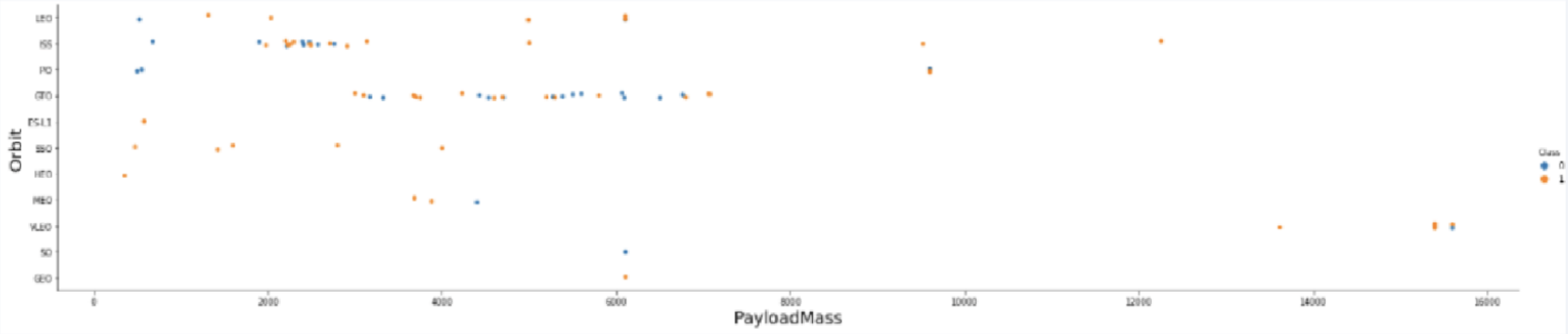
- Scatter point of Flight number vs. Orbit type



- The scatter plot above is showing the relationship between orbit and flight number. Based on the plot, the GTO doesn't have any relationship between flight numbers and the orbit, while LEO does.

Payload vs. Orbit Type

- Scatter point of payload vs. orbit type

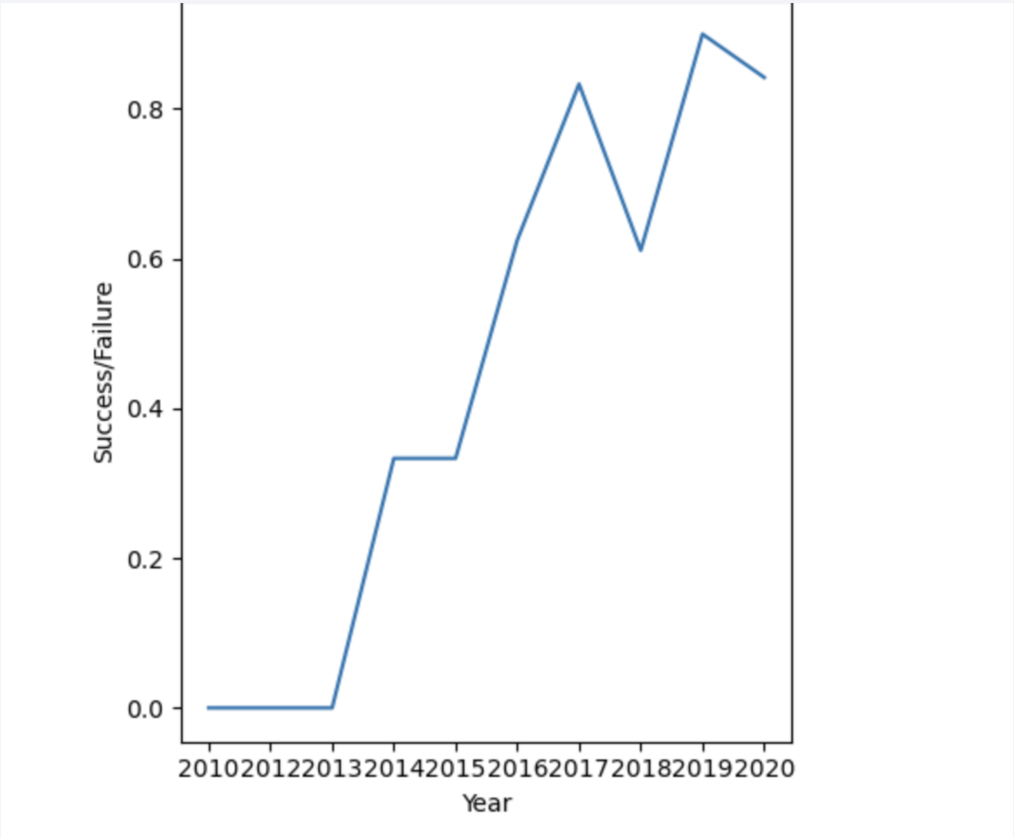


- PO, LEO and VSS seem to have heavier payload.

Launch Success Yearly Trend

Line chart of yearly average success rate

Based on the trend, we can see that with the years of testing, success rate went up and the most successful years were 2017 and 2019.



All Launch Site Names

- The names of the unique launch sites
- Using distinct, we found the unique launch names

Task 1

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

In [9]:
```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

Out[9]:

Launch_Sites
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`.
- The query displays 5 records starting with CCA

In [11]:
```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

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

Out[11]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Lan
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Fai
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Fai
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	

Total Payload Mass

- The total payload carried by boosters from NASA
- We calculated total payload from the table which is 45596

Task 3

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

In [15]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE customer = 'NASA (CRS)';
```

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

Out[15]:

SUM(PAYLOAD_MASS__KG_)
45596.0

Average Payload Mass by F9 v1.1

- Calculated the average payload mass carried by booster version F9 v1.1
- In order to calculate the average payload AVG function which returns the average was used

Task 4

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

In [16]:
```
%sql SELECT AVG(payload_mass__kg_) FROM SPACEXTBL WHERE booster_version = 'F9 v1.1';
```

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

Out[16]:

AVG(payload_mass__kg_)
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Based on the calculation, first successful landing was on 01/08/2018

Hint:Use min function

In [17]:
```
%sql SELECT MIN(DATE) AS "First Successful Landing" FROM SPACEXTBL WHERE landing_outcome = 'Success (ground pad)'
```

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

Out[17]:

First Successful Landing
01/08/2018

Successful Drone Ship Landing with Payload between 4000 and 6000

- The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- The "WHERE" clause was used.

```
In [20]: %sql SELECT booster_version FROM SPACEXTBL WHERE landing_outcome = 'Success (drone ship)' AND payload_mass__kg_ >

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

Out[20]:

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

Total Number of Successful and Failure Mission Outcomes

- Calculates the total number of successful and failure mission outcomes
- In order to calculate successful/failed mission outcomes we used Count.

```
In [21]: %sql SELECT COUNT(mission_outcome) FROM SPACEXTBL WHERE mission_outcome LIKE 'Success%'

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

Out[21]:

COUNT(mission_outcome)
100

```
In [22]: #Failure
         %sql SELECT COUNT(mission_outcome) FROM SPACEXTBL WHERE mission_outcome LIKE 'Fail%'

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

Out[22]:

COUNT(mission_outcome)
1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- We used Distinct and Where clause.

In [25]:
```
%sql SELECT DISTINCT booster_version AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEXTBL
WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM SPACEXTBL);
```

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

Out[25]:

Booster Versions which carried the Maximum Payload Mass
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

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- The query used data from SPACEXTBL with data set to 2015 and outcome as failure

```
In [26]: %sql SELECT booster_version, launch_site FROM SPACEXTBL WHERE DATE LIKE '2015-%' AND \
         landing_outcome = 'Failure (drone ship)';

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

Out[26]:

Booster_Version	Launch_Site

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
- We used Where clause to provide the date

Interval and then Group by to group by

Landing Outcome column.

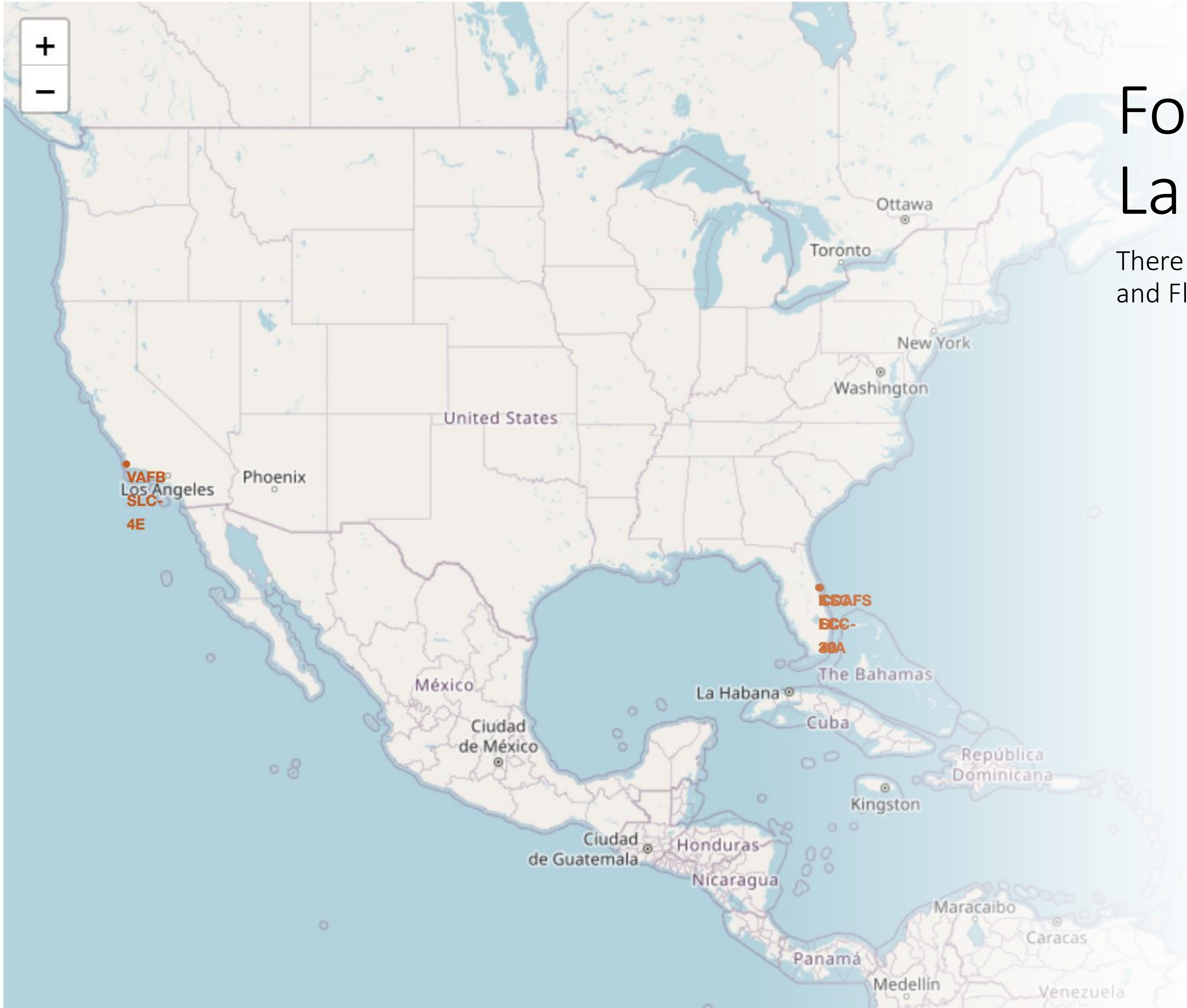
```
%%sql
Select
    "Landing _Outcome", count("Landing _Outcome") as cnt
From SPACEXTBL
Where Date Between '04-06-2010' and '20-03-2017'
Group by "Landing _Outcome"
Order by 2 desc
```

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

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

Section 3

Launch Sites Proximities Analysis

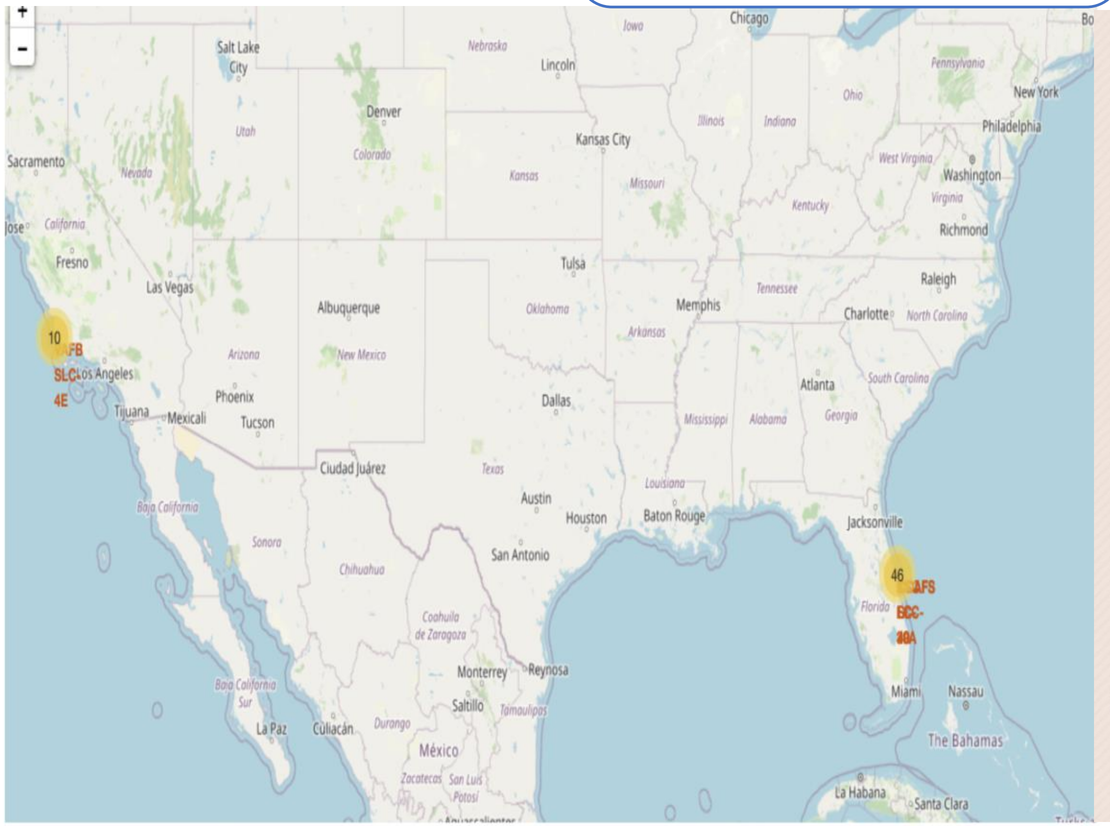


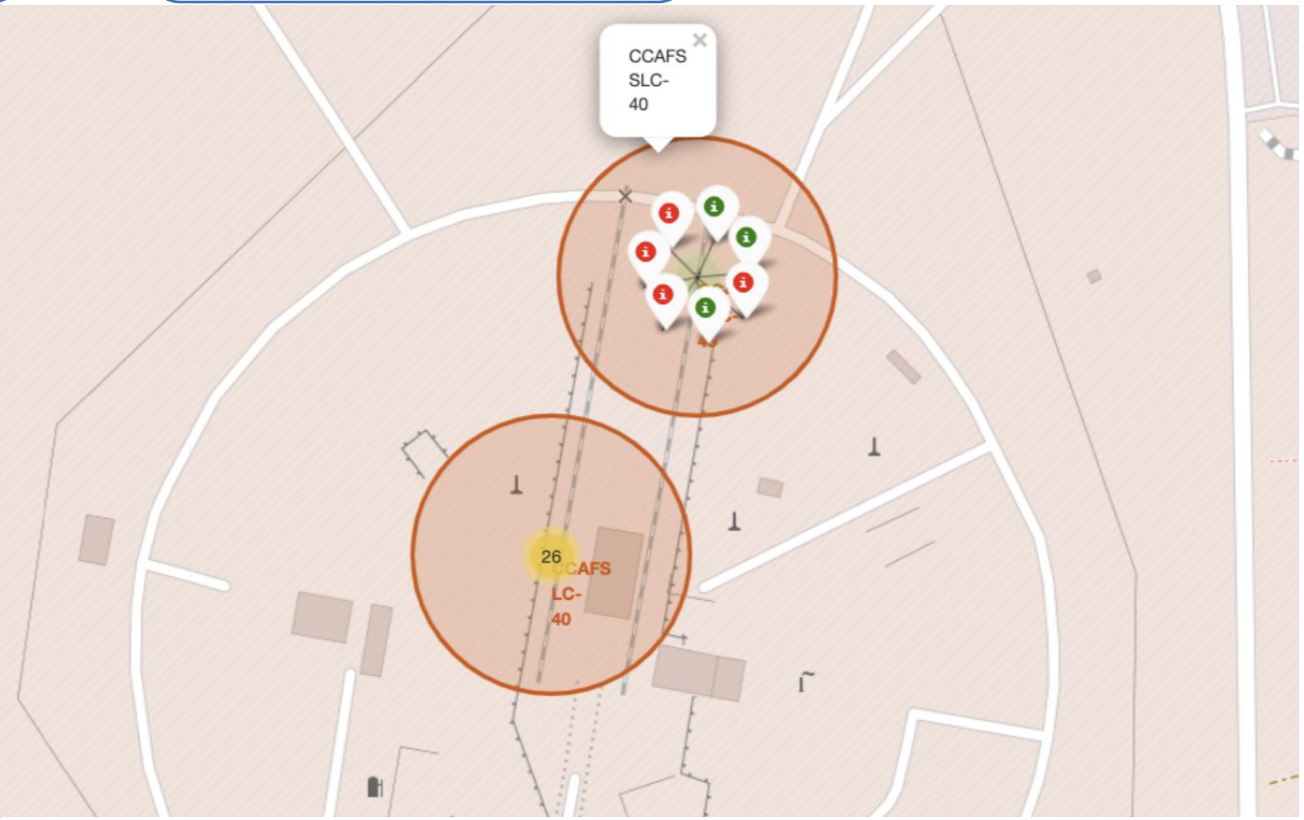
Folium Map 1. US Launch sites

There were two launch points – California and Florida

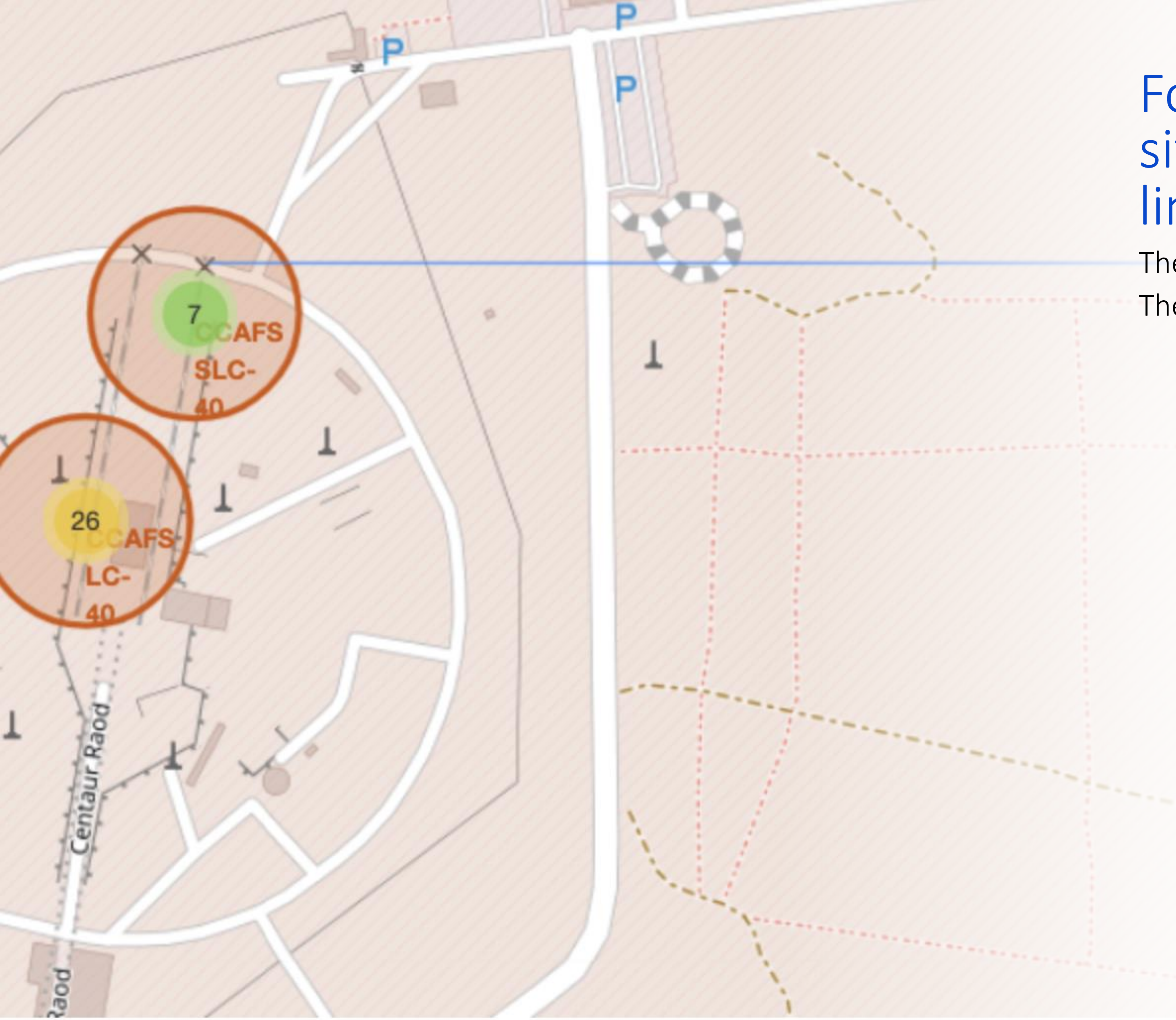
Folium Map. Number of launces and outcomes

Based on the map, it appeared that there were as successful (marked in green) as failed launches (marked in red)



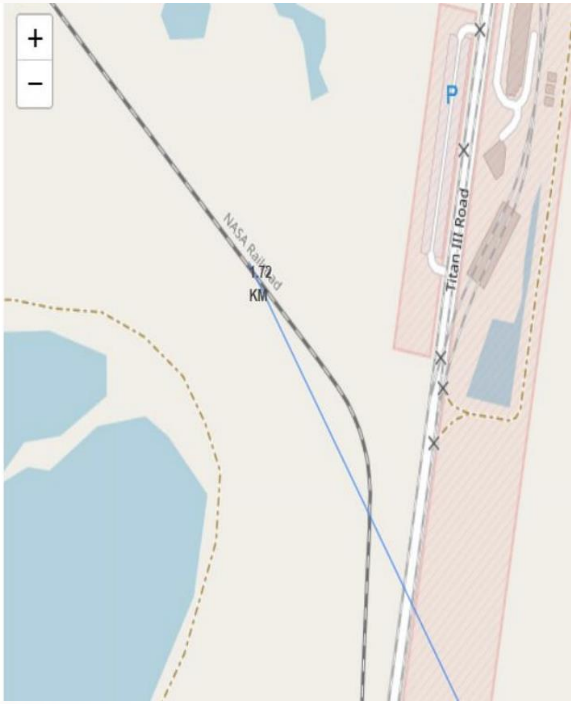


Folium Map. Launch site distance to coast line.



The distance to coast line is 0.89 km.

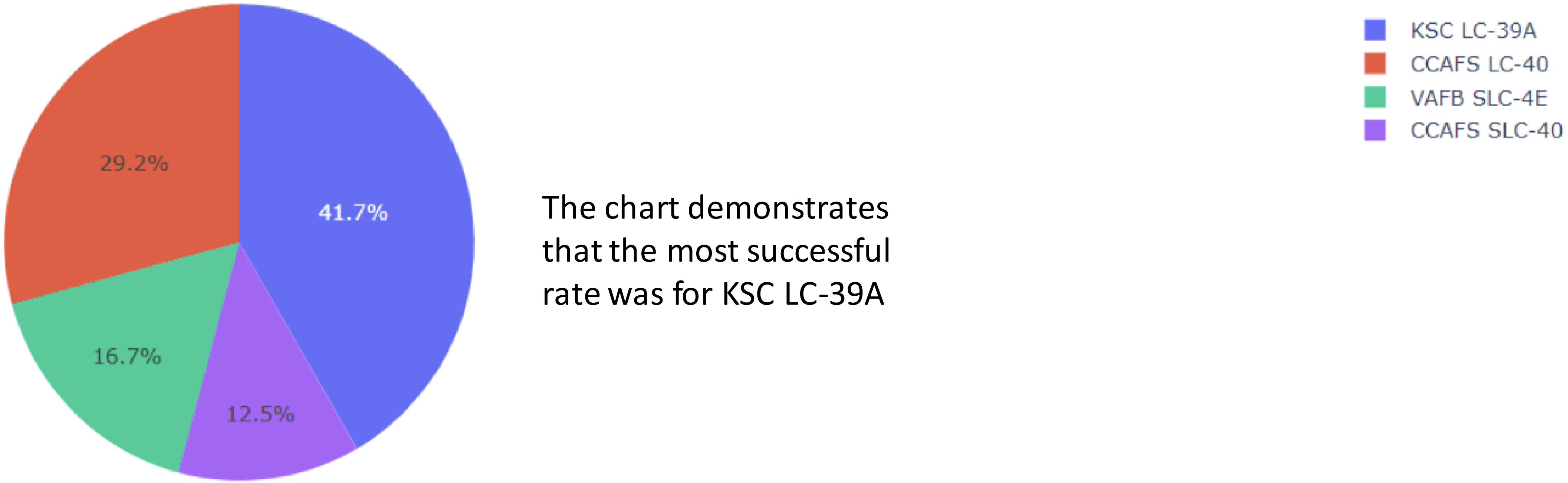
The distance to highway is 0,6 km.



Section 4

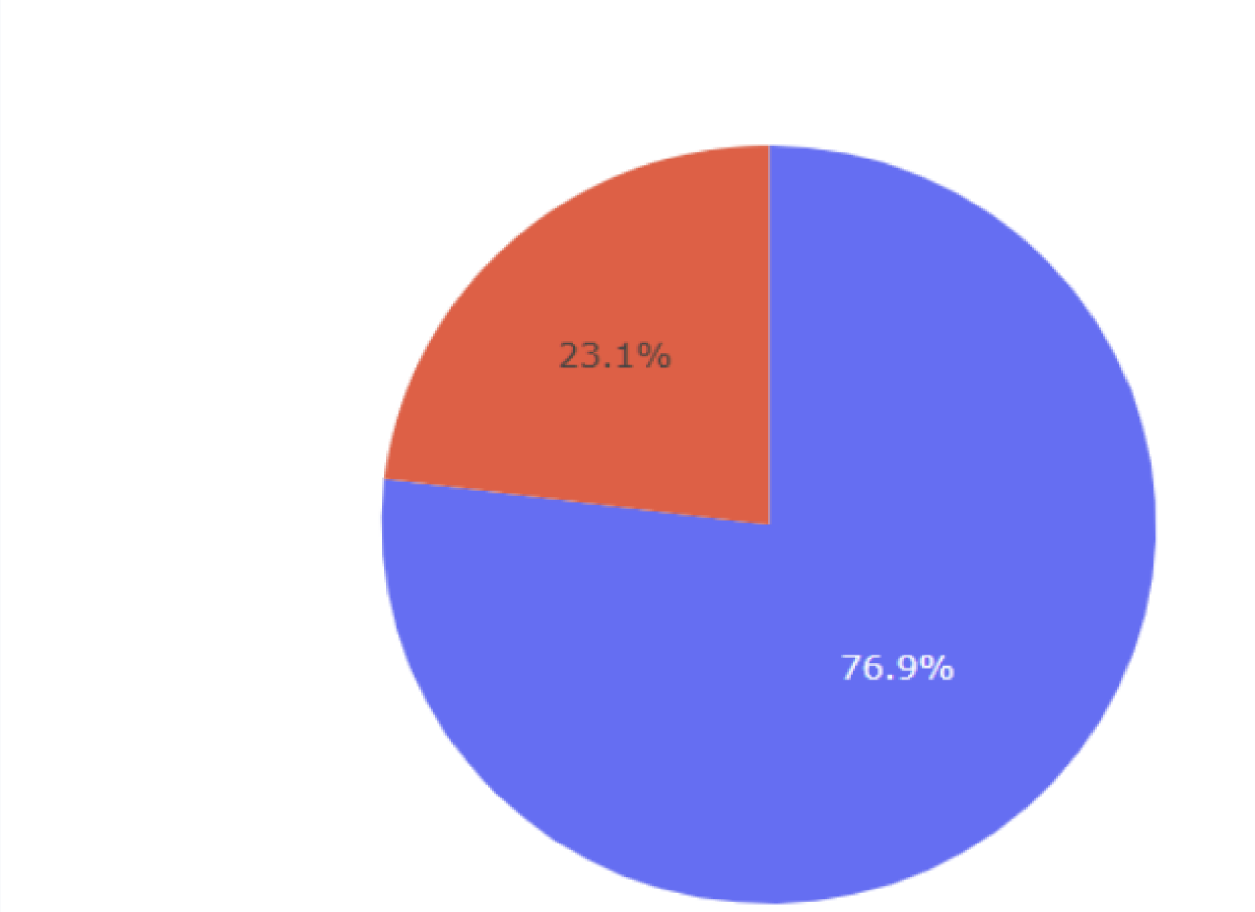
Build a Dashboard with Plotly Dash

Launches Success rate pie chart



The chart demonstrates that the most successful rate was for KSC LC-39A

The launch site with the highest success rate



Based on the previous and current charts, the most successfull rate has KLC LC-39A. It also has failure rats as 23.1%

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Based on the calculation, all the models had the highest accuracy which was equal to 83.33%

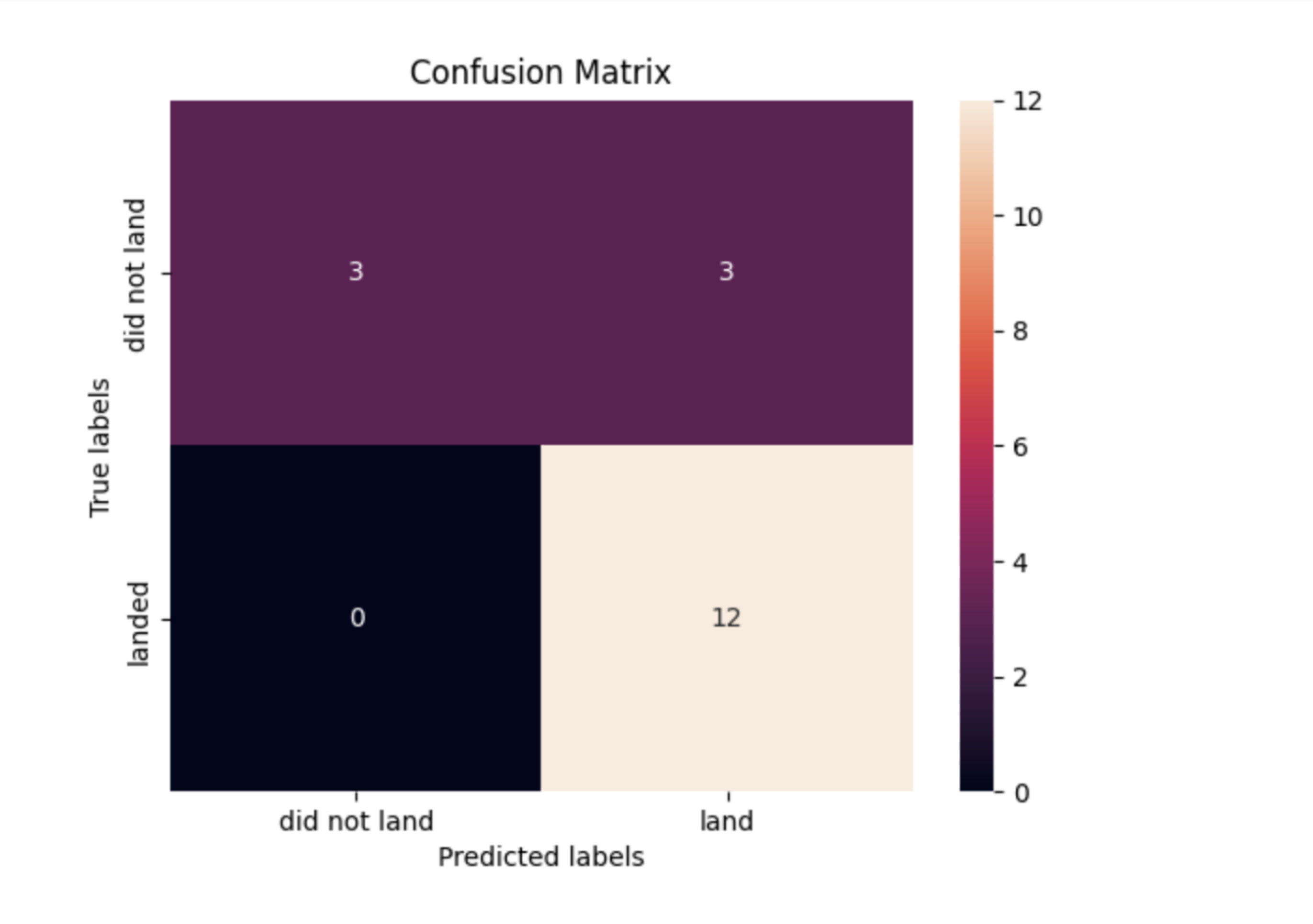
Calculate the accuracy of tree_cv on the test data using the method `score`:

```
[31]: tree_score = tree_cv.score(X_test,Y_test)
      tree_score
```

```
[31]: 0.8333333333333334
```

Confusion Matrix

- The confusion matrix demonstrates that all other methods has the same accuracy (83%) and that's why the matrix looks similar for all the models.



Conclusions

- Launch success rate was significantly improved since 2013
- The most successful Launch site is KSC LC-39A, where the Booster Version it is FT.
- There were 100 successful compared to 1 failed mission outcomes
- All models has the same score and any can be good for current testing.

Appendix

- My GitHb link link

Thank you!