Winning Space Race with Data Science

Shoaib Azam
15 Aug 2024

Outline

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

Executive Summary

- In this capstone project, we will predict if the SpaceX Falcon 9 first stage will land successfully using several machine learning classification algorithms.
- The main steps in this project include:
  - Data collection, wrangling, and formatting
  - Exploratory data analysis
  - Interactive data visualization
  - Machine learning prediction
- Our graphs show that some features of the rocket launches have a correlation with the outcome of the launches, i.e., success or failure.
- It is also concluded that decision tree may be the best machine learning algorithm to predict if the Falcon 9 first stage will land successfully.

Introduction

- In this capstone, we will predict if the Falcon 9 first stage will land successfully. 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.
- Most unsuccessful landings are planned. Sometimes, SpaceX will perform a controlled landing in the ocean.
- The main question that we are trying to answer is, for a given set of features about a Falcon 9 rocket launch which include its payload mass, orbit type, launch site, and so on, will the first stage of the rocket land successfully?

Section 1

Methodology

Methodology

- Data Collection using API and Webscraping
- Perform data wrangling
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

Data Collection – SpaceX API

- SpaceX API
  - The API used is https://api.spacexdata.com/v4/rockets/.
  - The API provides data about many types of rocket launches done by SpaceX, the data is therefore filtered to include only Falcon 9 launches.
  - Every missing value in the data is replaced the mean the column that the missing value belongs to.
  - We end up with 90 rows or instances and 17 columns or features. The picture below shows the first few rows of the data:
  - Github Link to API notebook

Data Collection - Scraping

- The data is scraped from https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
- The website contains only the data about Falcon 9 launches.
- We end up with 121 rows or instances and 11 columns or features. The picture below shows the first few rows of the data:
- Github link to Webscraping notebook

Data Wrangling

- The data is later processed so that there are no missing entries and categorical features are encoded using one-hot encoding.
- An extra column called ‘Class’ is also added to the data frame. The column ‘Class’ contains 0 if a given launch is failed and 1 if it is successful.
- In the end, we end up with 90 rows or instances and 83 columns or features.
- Github link to Data Wrangling notebook

Exploratory Data Analysis

- Pandas and NumPy
  - Functions from the Pandas and NumPy libraries are used to derive basic information about the data collected, which includes:
    - The number of launches on each launch site
    - The number of occurrence of each orbit
    - The number and occurrence of each mission outcome
- SQL
  - The data is queried using SQL to answer several questions about the data such as:
    - The names of 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
- EDA SQL Notebook Github

Data Visualization 1

- Matplotlib and Seaborn
  - Functions from the Matplotlib and Seaborn libraries are used to visualize the data through scatterplots, bar charts, and line charts.
  - The plots and charts are used to understand more about the relationships between several features, such as:
    - The relationship between flight number and launch site
    - The relationship between payload mass and launch site
    - The relationship between success rate and orbit type
  - Data Visualization Github
- Folium
  - Functions from the Folium libraries are used to visualize the data through interactive maps.
  - The Folium library is used to:
    - Mark all launch sites on a map
    - Mark the succeeded launches and failed launches for each site on the map
    - Mark the distances between a launch site to its proximities such as the nearest city, railway, or highway
- Data Visualization with Folium Github

Dashboard with Plotly Dash

- Dash

  - Functions from Dash are used to generate an interactive site where we can toggle the input using a dropdown menu and a range slider.
  - Using a pie chart and a scatterplot, the interactive site shows:
    - The total success launches from each launch site
    - The correlation between payload mass and mission outcome (success or failure) for each launch site

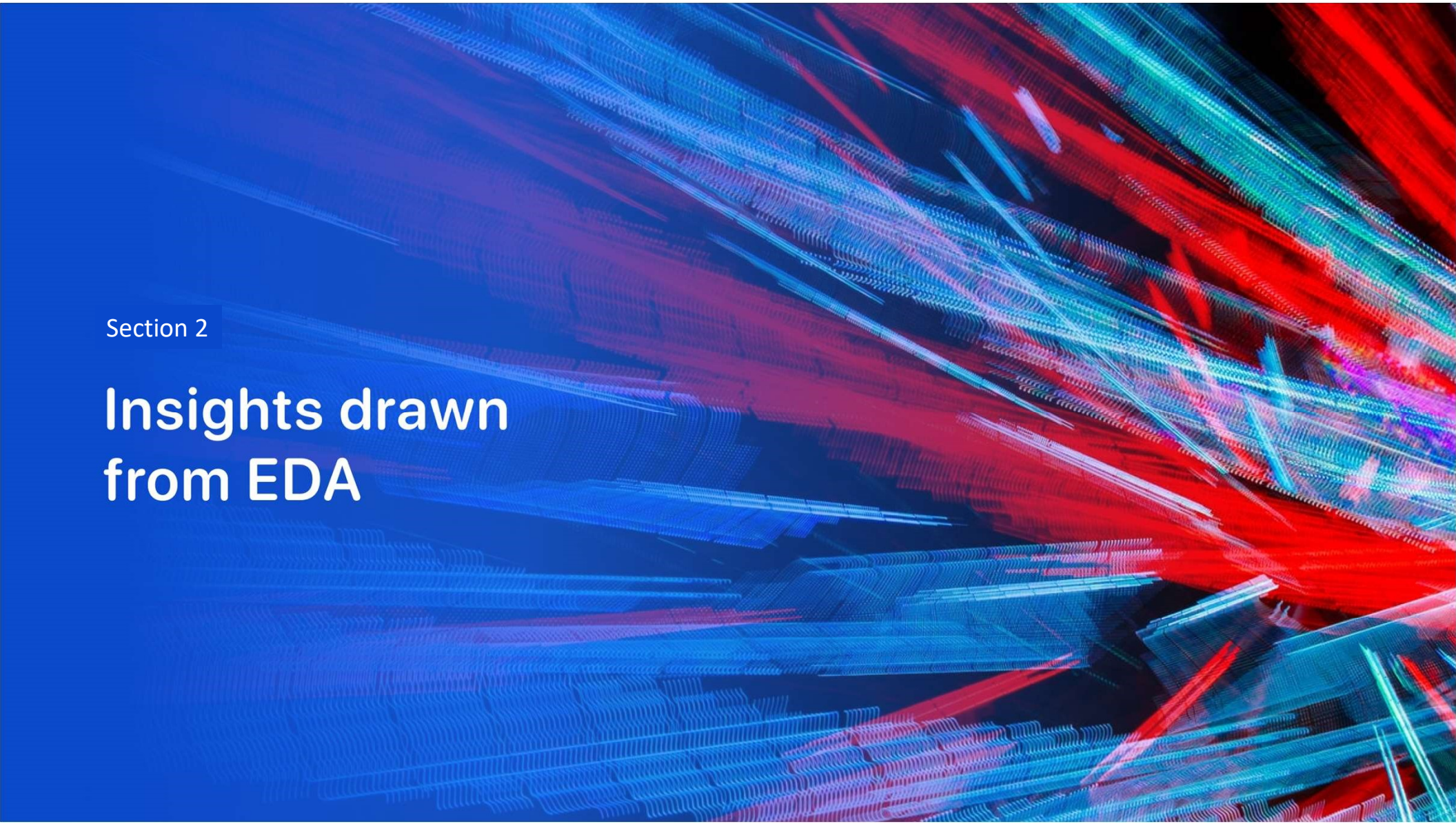
- Plotly Dash App Github

Predictive Analysis (Classification)

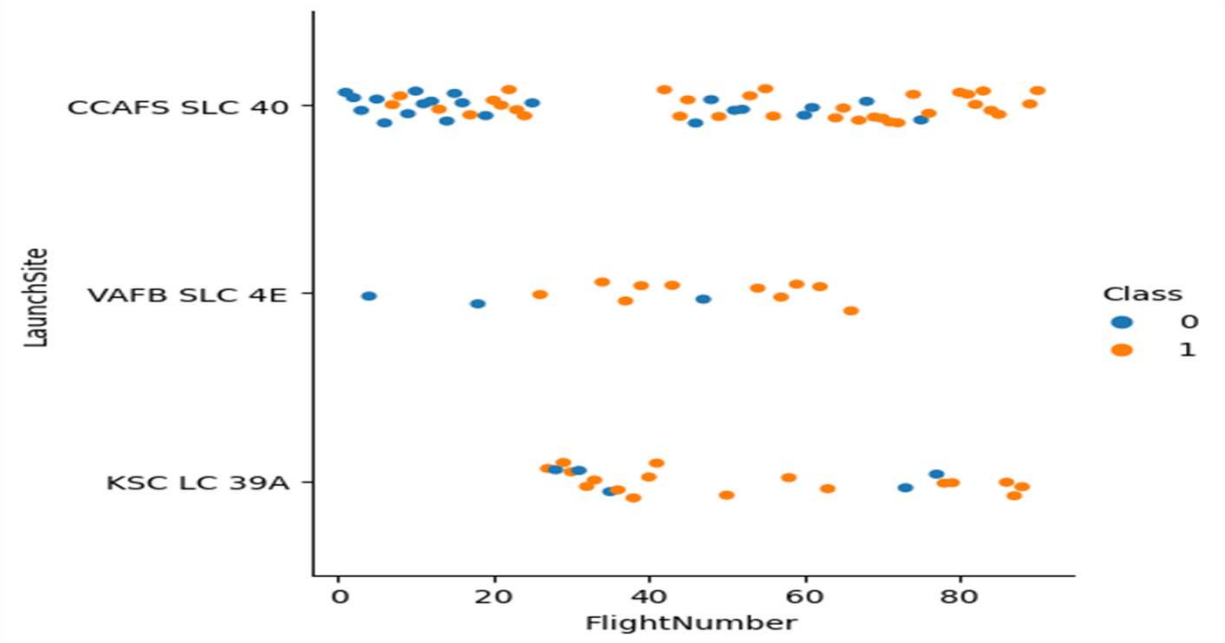
- Functions from the Scikit-learn library are used to create our machine learning models.
- The machine learning prediction phase include the following steps:
  - Standardizing the data
  - Splitting the data into training and test data
  - Creating machine learning models, which include:
    - Logistic regression
    - Support vector machine (SVM)
    - Decision tree
    - K nearest neighbors (KNN)
  - Fit the models on the training set
  - Find the best combination of hyperparameters for each model
  - Evaluate the models based on their accuracy scores and confusion matrix
- Predictive Analysis Github link

Results

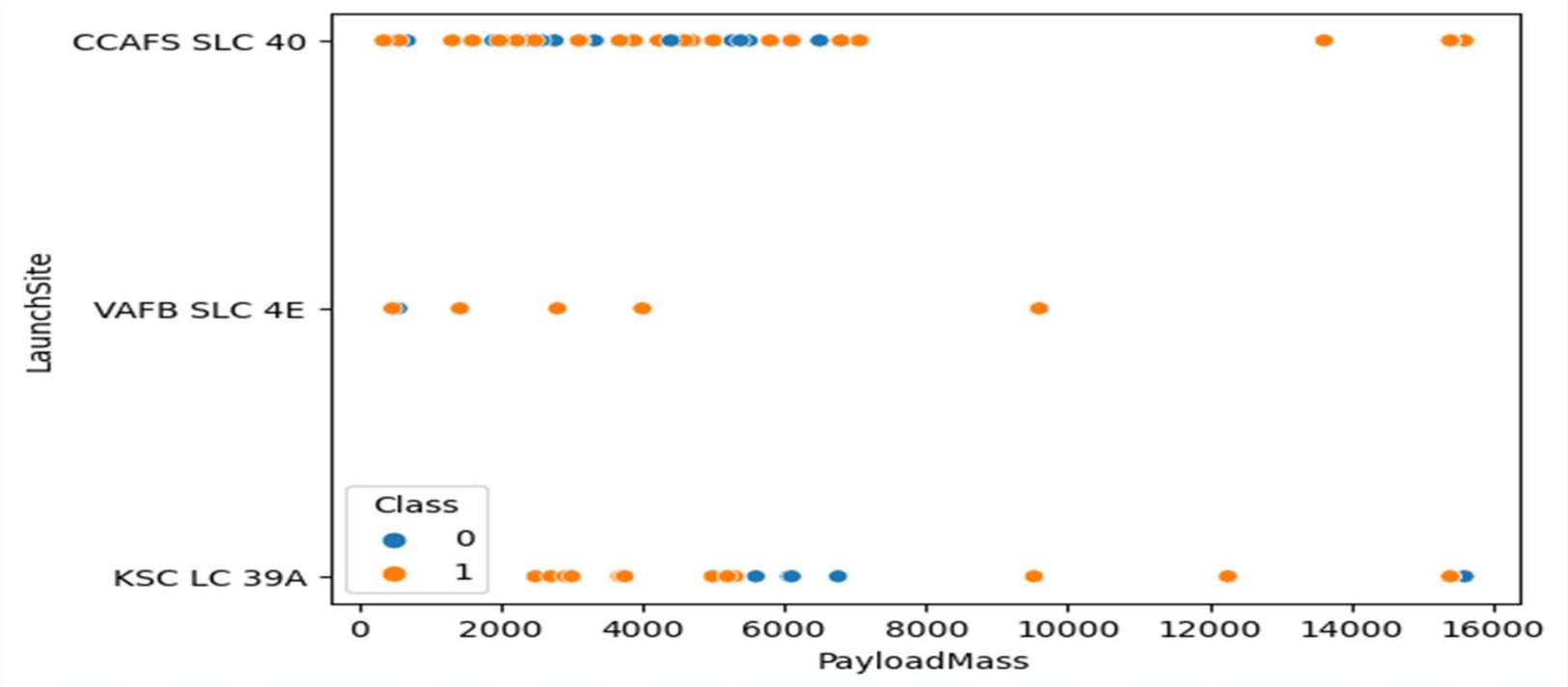
- The results are split into 5 sections:
  - Matplotlib and Seaborn (EDA with Visualization)
  - SQL (EDA with SQL)
  - Folium
  - Dash
  - Predictive Analysis
- In all of the graphs that follow, class 0 represents a failed launch outcome while class 1 represents a successful launch outcome.


Section 2
Insights drawn
from EDA

Flight Number vs. Launch Site

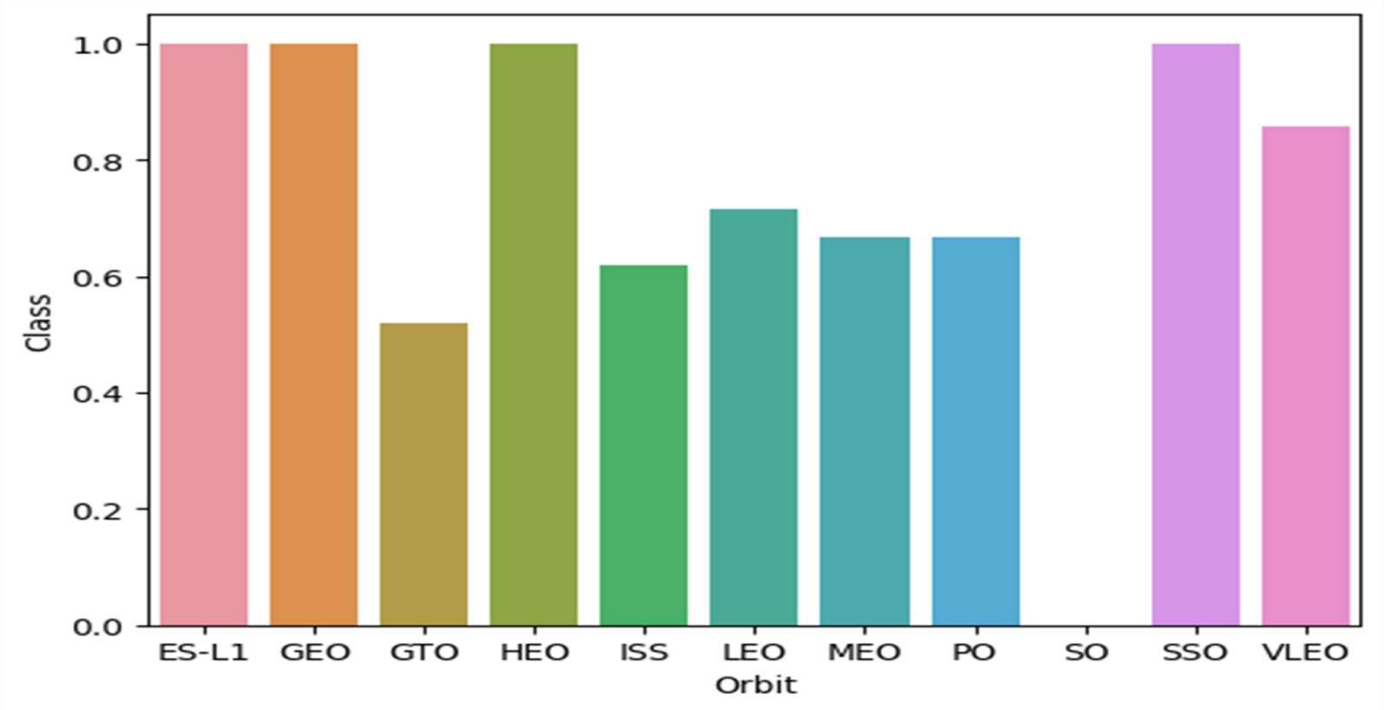


Payload vs. Launch Site



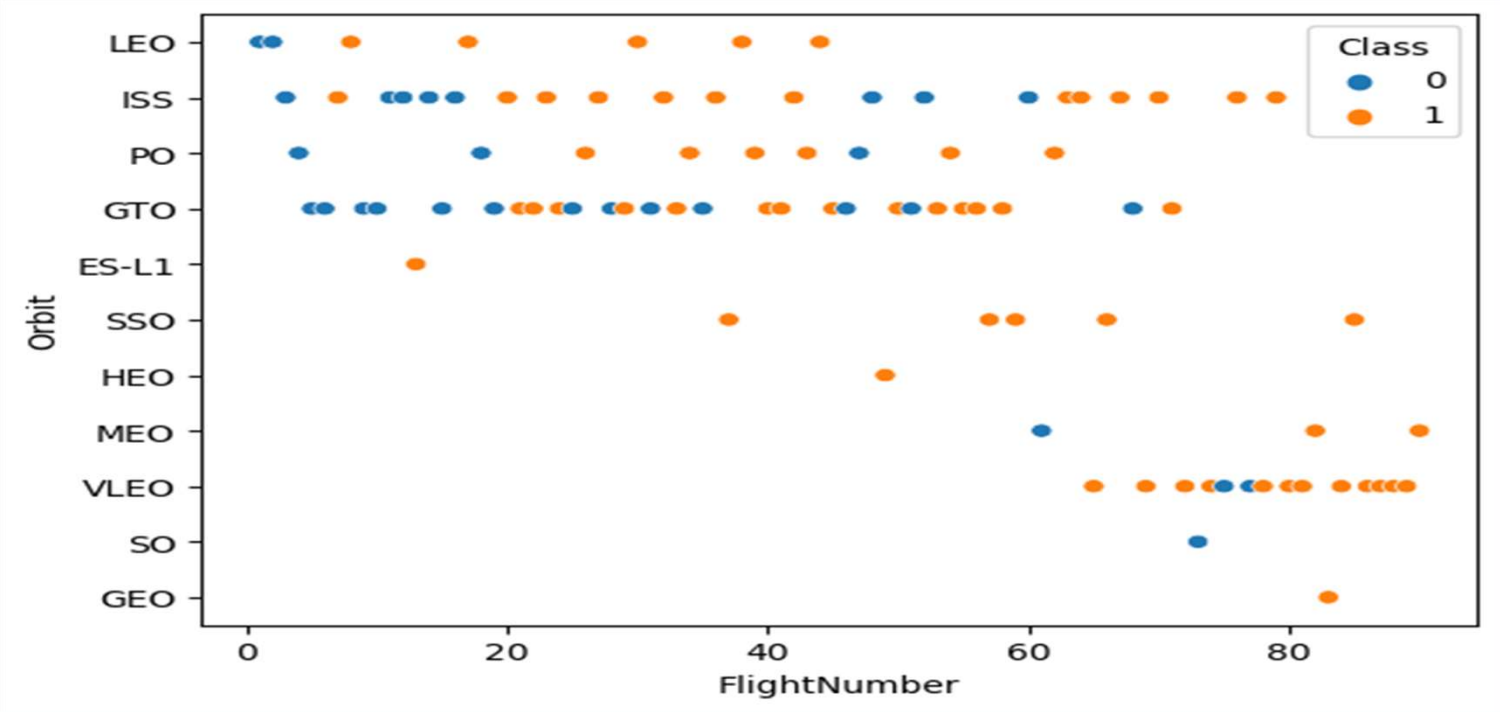
We can see that VAFB site is not used for launches with high mass payloads (>10000kg)

Success Rate vs. Orbit Type

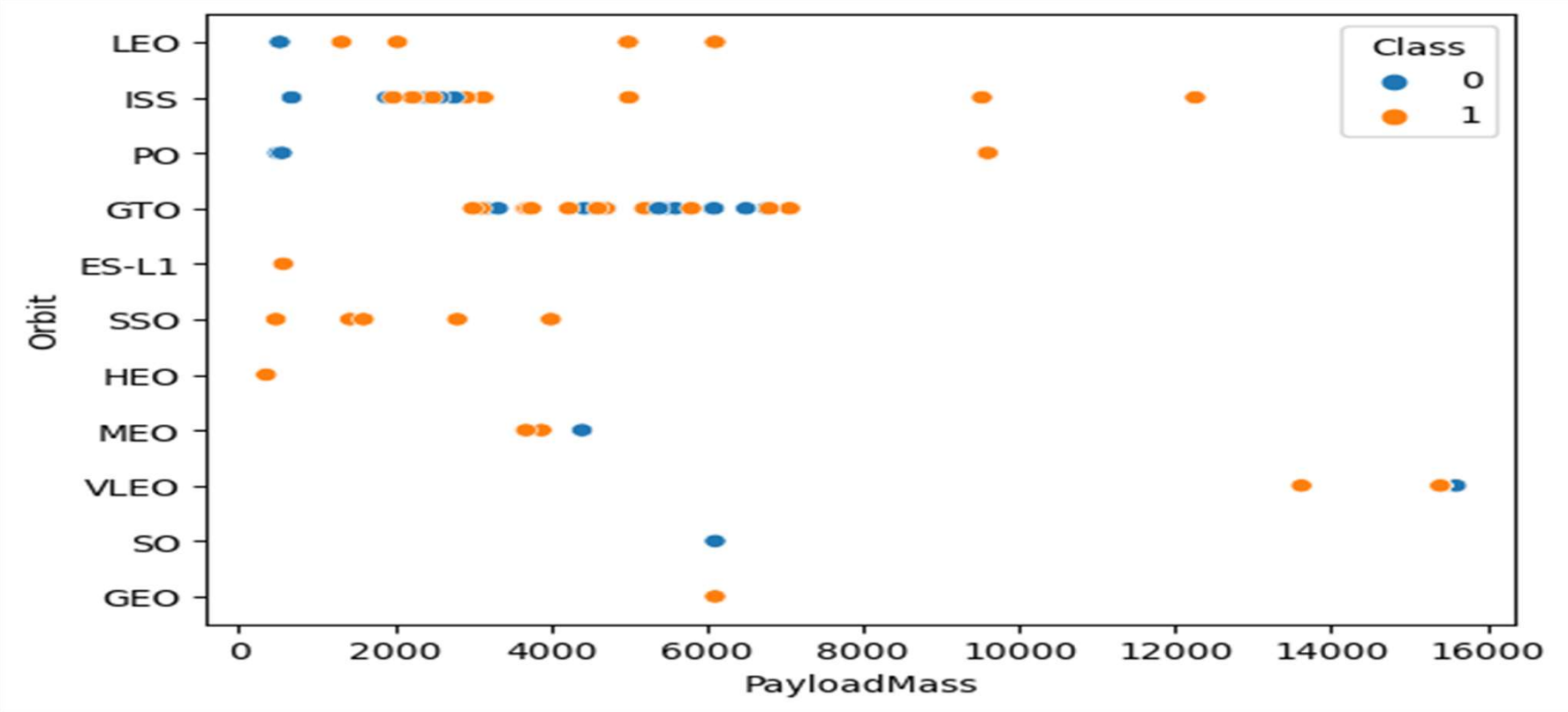


- We can see that SO Orbit launches have zero success rate. And 4 orbits have a perfect success rate.

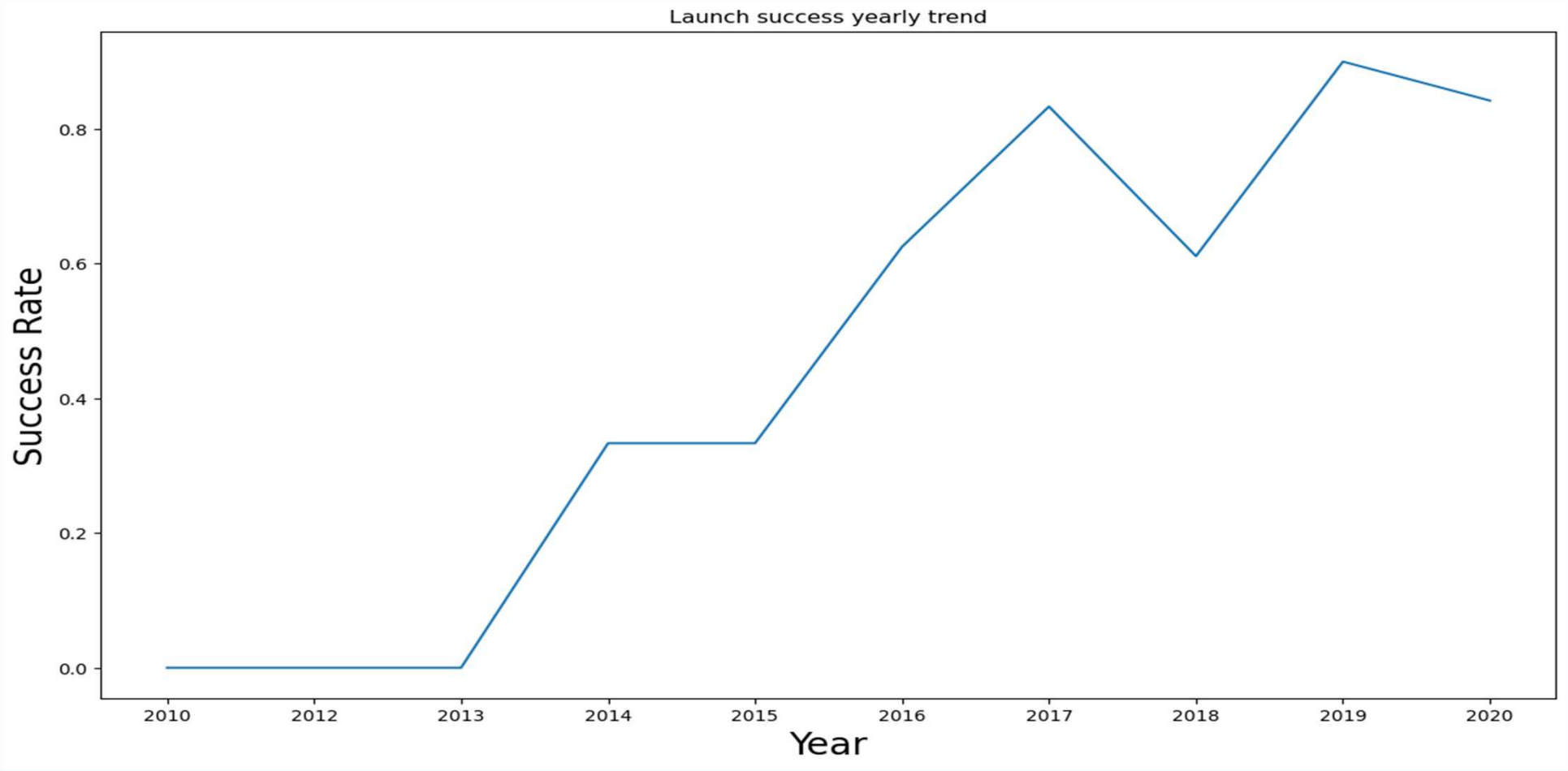
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

Task 1

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

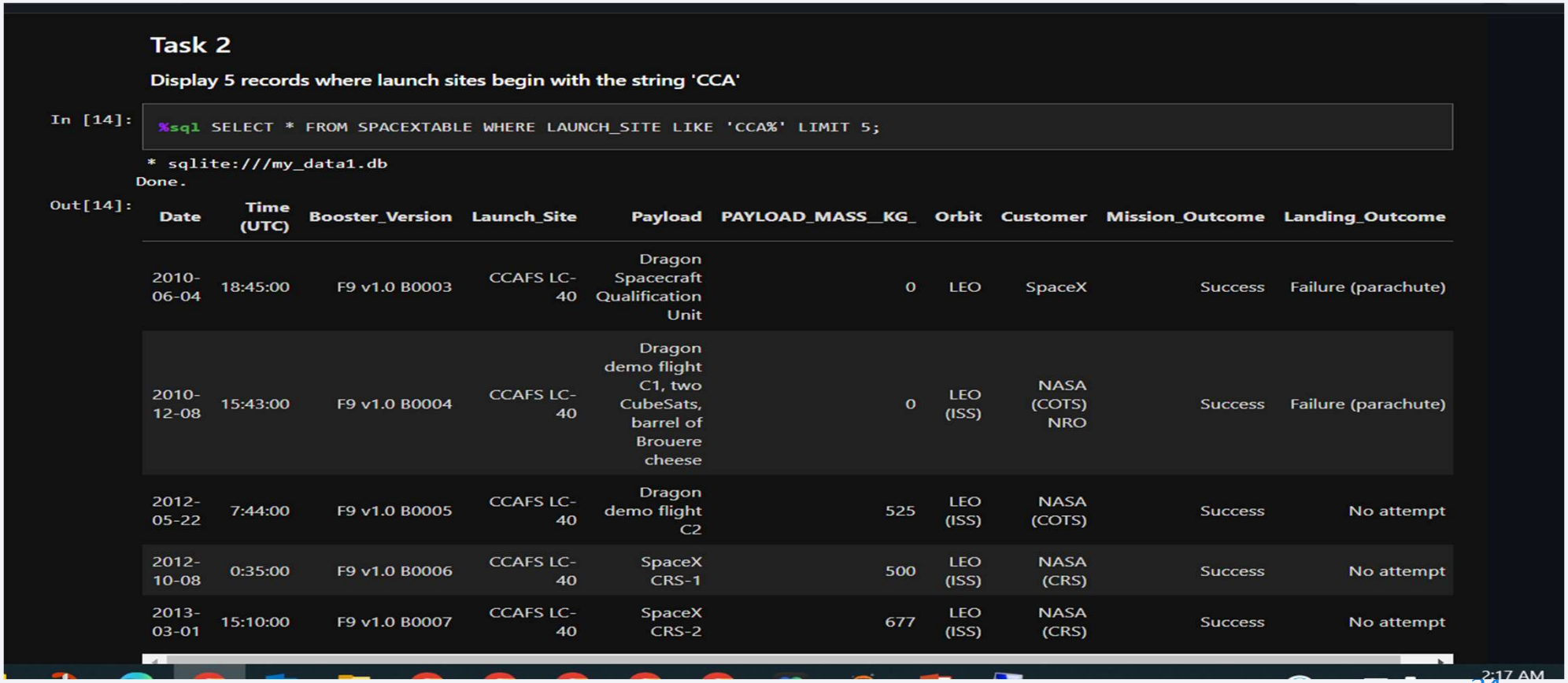
In [12]:
```
%sql Select distinct launch_site from SPACEXTABLE
```

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

Out[12]:

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

Launch Site Names Begin with 'CCA'



Task 2

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

```
In [14]: %sql SELECT * FROM SPACEXTABLE WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
 * sqlite:///my_data1.db
Done.
```

Out[14]:

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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

- Like keyword is used for launchsites starting with CCA, and limit is used to reduce output to 5 results only.

Total Payload Mass

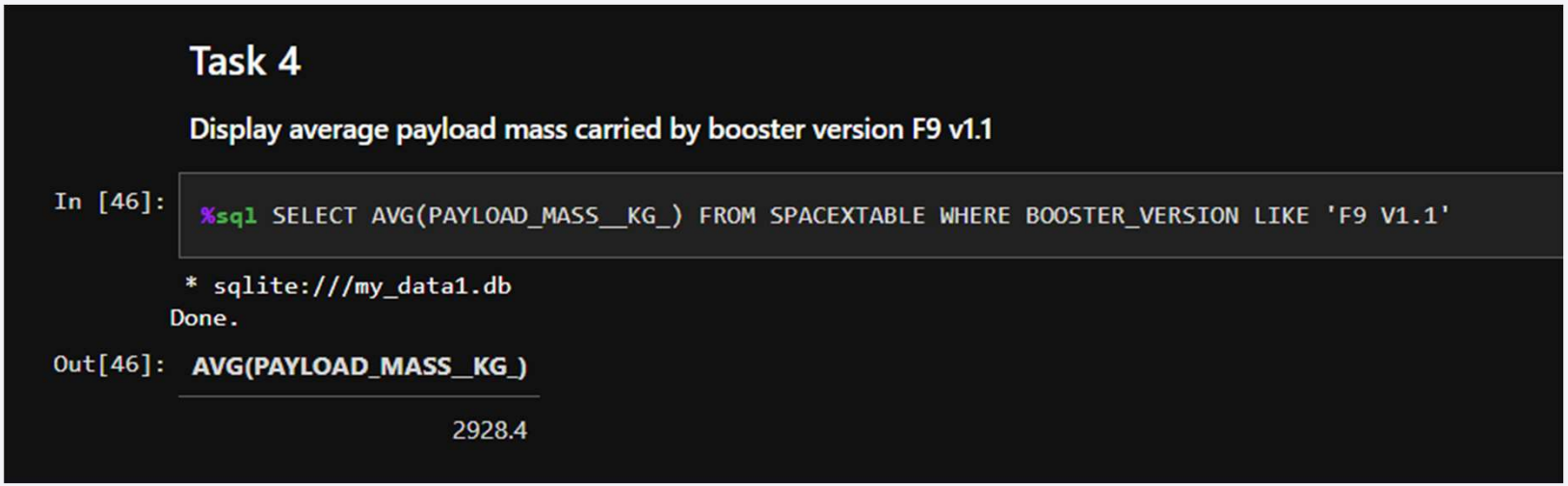
Task 3

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

```
In [15]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE CUSTOMER = 'NASA (CRS)';
 * sqlite:///my_data1.db
Done.
Out[15]: SUM(PAYLOAD_MASS__KG_)
         45596
```

- Result is filtered by selecting rows with NASA (CRS) as customer and then all the payloadmass values are added using the sum function.

Average Payload Mass by F9 v1.1



Task 4

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

In [46]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE BOOSTER_VERSION LIKE 'F9 V1.1'
```

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

Out[46]:

AVG(PAYLOAD_MASS__KG_)
2928.4

- Filtered using booster_version column and then average of payload values is found using avg function.

First Successful Ground Landing Date

Task 5

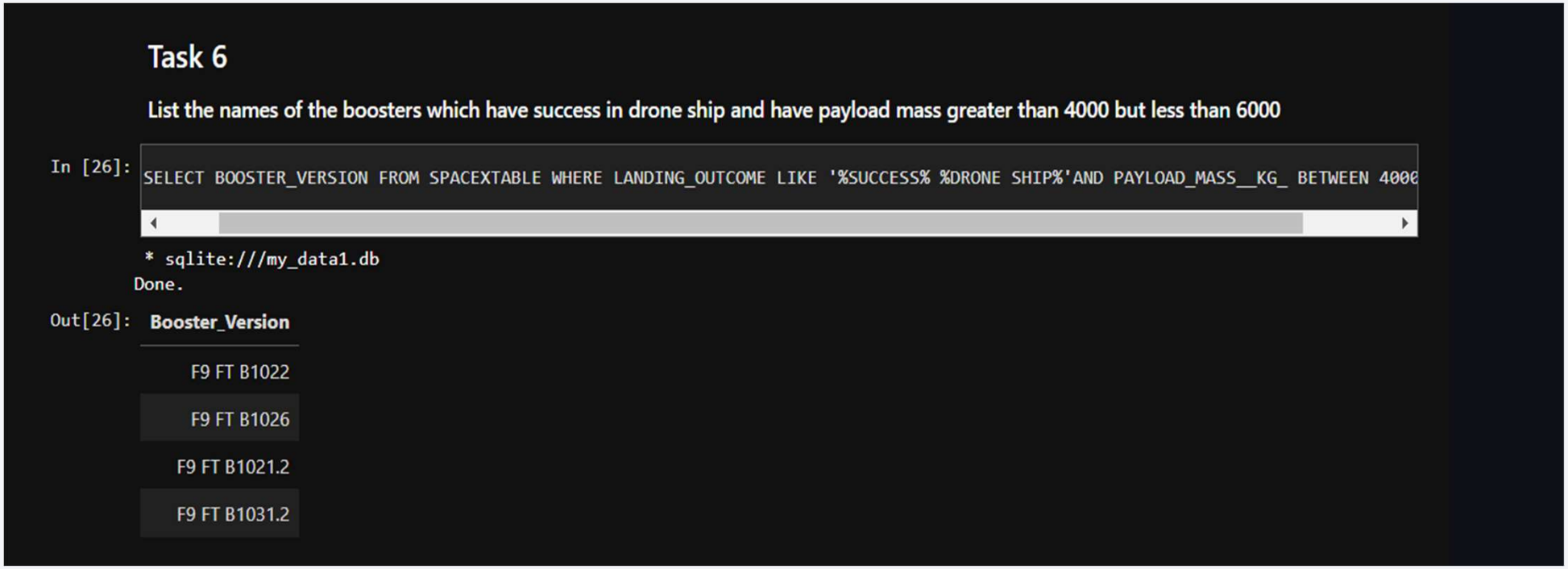
List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

```
In [23]: %sql SELECT MIN(DATE) FROM SPACEXTABLE WHERE "LANDING_OUTCOME" LIKE '%SUCCESS% %GROUND PAD%';
 * sqlite:///my_data1.db
Done.
Out[23]: MIN(DATE)
         2015-12-22
```

- Min function is used to find the oldest (numerically smallest) date for ground pad successes.

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

In [26]:
```
SELECT BOOSTER_VERSION FROM SPACEXTABLE WHERE LANDING_OUTCOME LIKE '%SUCCESS% %DRONE SHIP%'AND PAYLOAD_MASS__KG_ BETWEEN 4000
```

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

Out[26]:

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

- names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

```
MISSION_OUTCOME LIKE '%SUCCESS%' THEN 'SUCCESS' ELSE 'FAILURE' END AS OUTCOME, COUNT(*) AS COUNT FROM SPACEXTABLE GROUP BY 1;
```

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

OUTCOME	COUNT
FAILURE	1
SUCCESS	100

Query:

```
SELECT CASE WHEN MISSION_OUTCOME LIKE '%SUCCESS%' THEN 'SUCCESS' ELSE
'FAILURE' END AS OUTCOME, COUNT(*) AS COUNT FROM SPACEXTABLE GROUP BY 1;
```

Boosters Carried Maximum Payload

Task 8

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

```
[36]: BOOSTER_VERSION FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ IN (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE) ORDER BY 1 DESC
 * sqlite:///my_data1.db
Done.
```

t[36]:

Booster_Version
F9 B5 B1060.3
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1056.4
F9 B5 B1051.6
F9 B5 B1051.4
F9 B5 B1051.3
F9 B5 B1049.7
F9 B5 B1049.5
F9 B5 B1049.4
F9 B5 B1048.5
F9 B5 B1048.4

Sql:

SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ IN (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE) ORDER BY 1 DESC

2015 Launch Records

Task 9

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%sql SELECT SUBSTR(DATE, 6,2) AS MONTH, LANDING_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTABLE WHERE SUBSTR(DATE, 0
```

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

MONTH	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

sql SELECT SUBSTR(DATE, 6,2) AS MONTH, LANDING_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTABLE WHERE SUBSTR(DATE, 0,5)='2015' AND LANDING_OUTCOME LIKE 'FAILURE%DRONE SHIP%'

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 SPACEXTABLE WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY 1 ORD
```

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

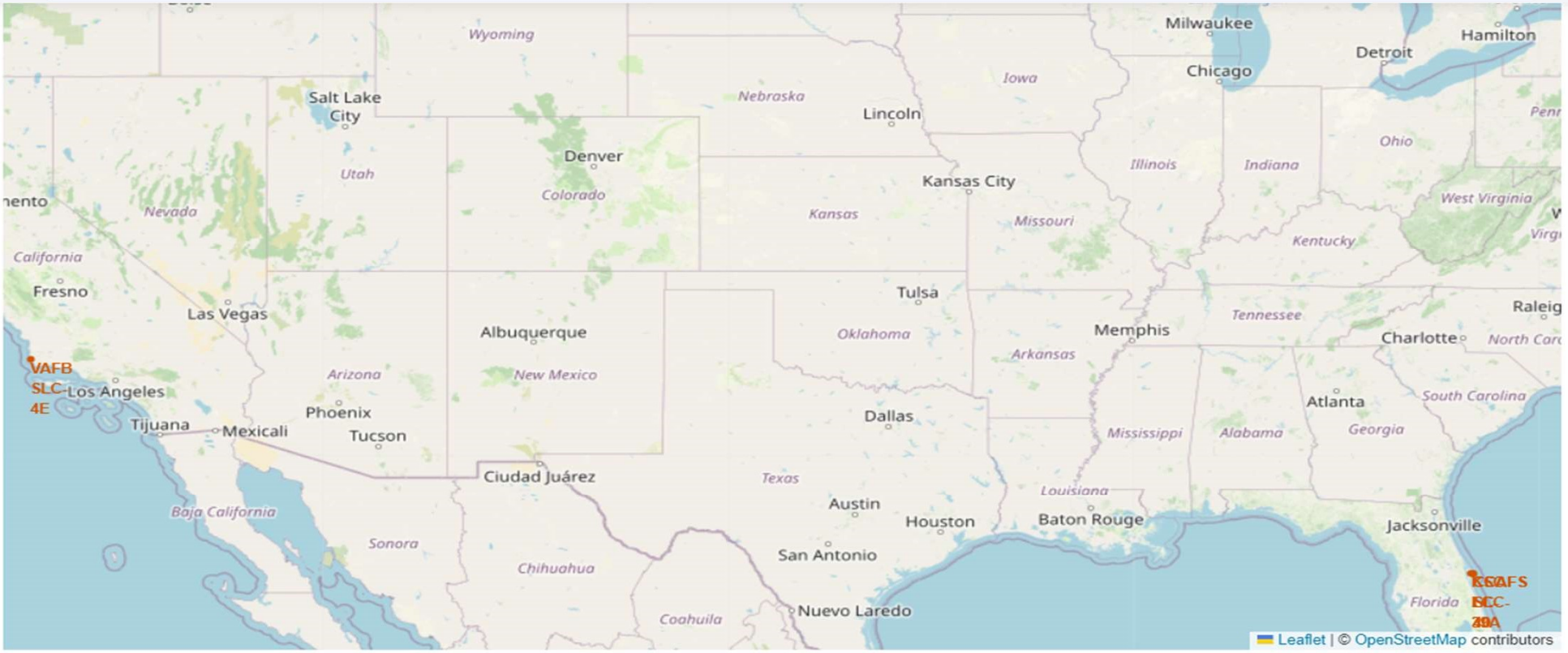
Landing_Outcome	Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

%sql SELECT LANDING_OUTCOME, COUNT(*) Count FROM SPACEXTABLE WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY 1 ORDER BY 2 DESC;

Section 3

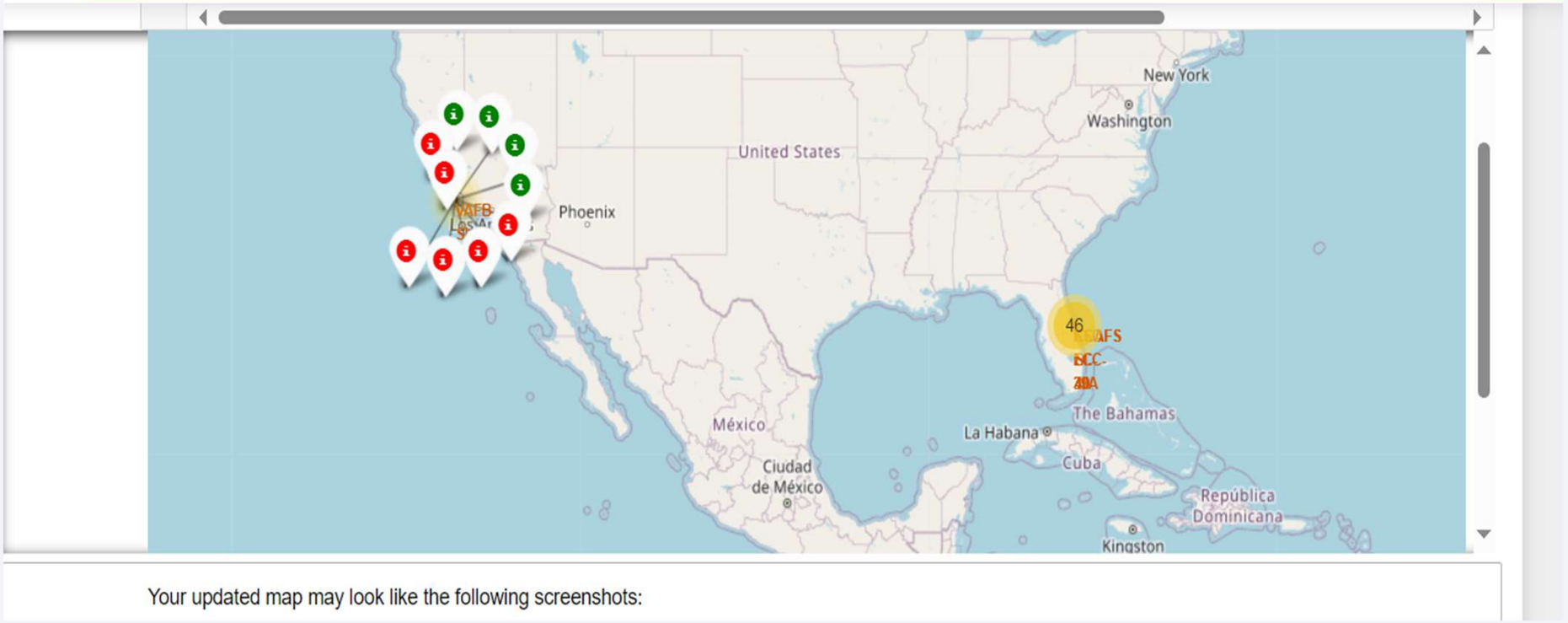
Launch Sites Proximities Analysis

SpaceX Landing Sites



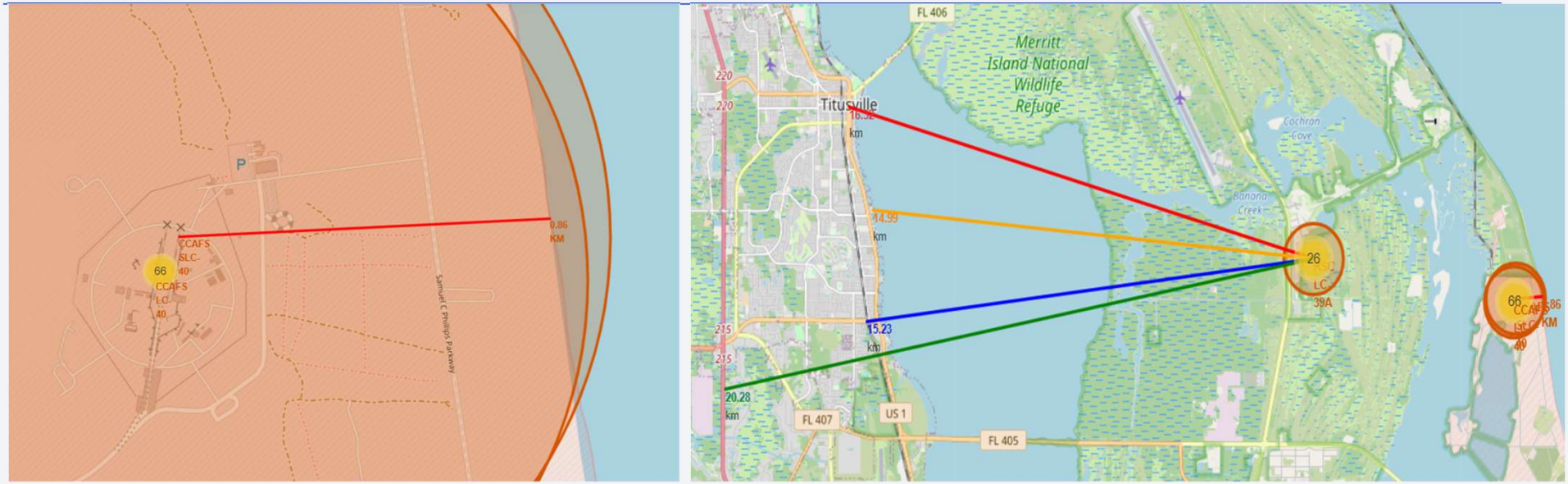
The generated map with marked launch sites should look similar to the following:

<Folium Map Screenshot 2>

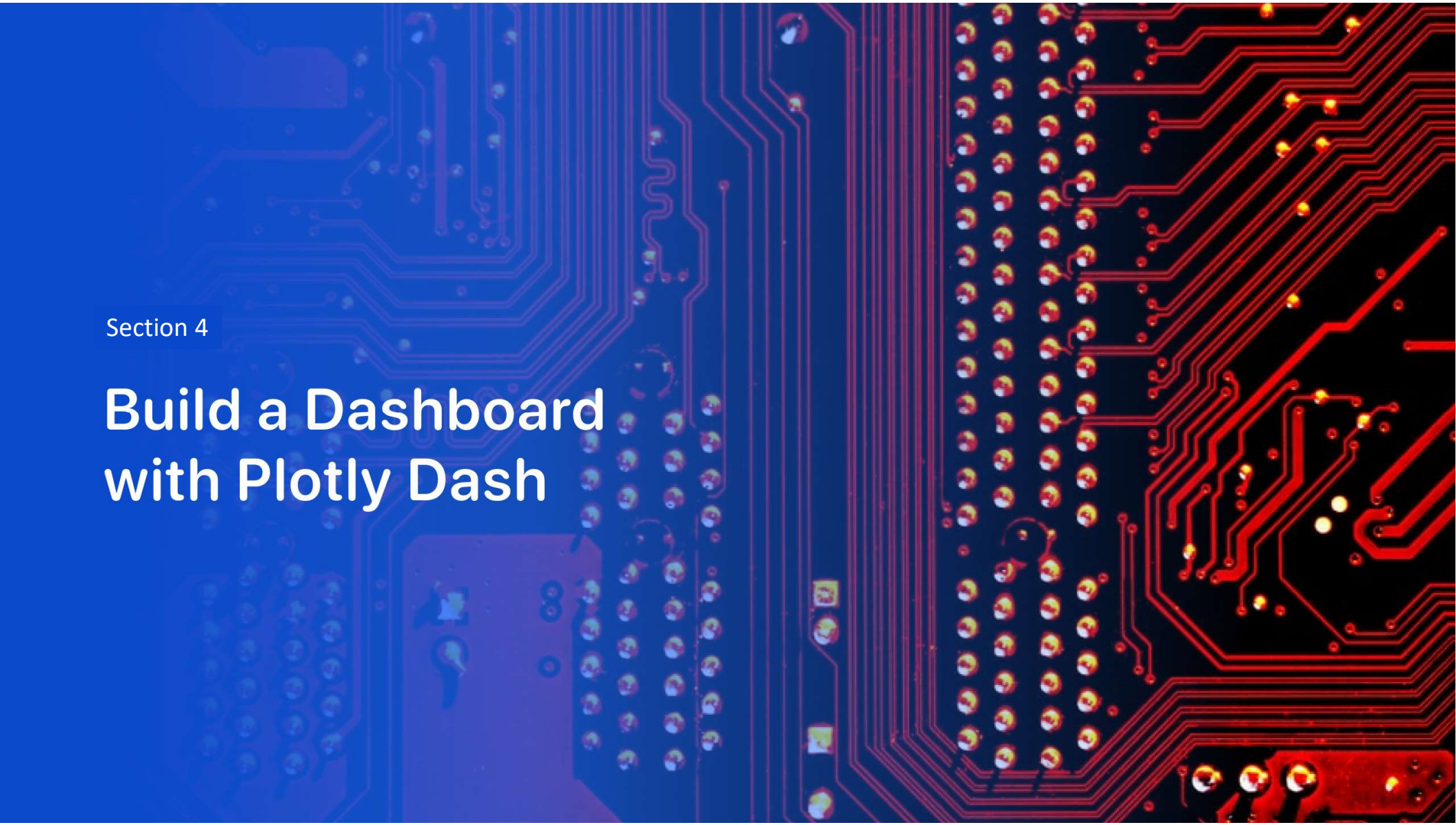


- If we zoom in on one of the launch site, we can see green and red tags. Each green tag represents a successful launch while each red tag represents a failed launch

<Folium Map Screenshot 3>



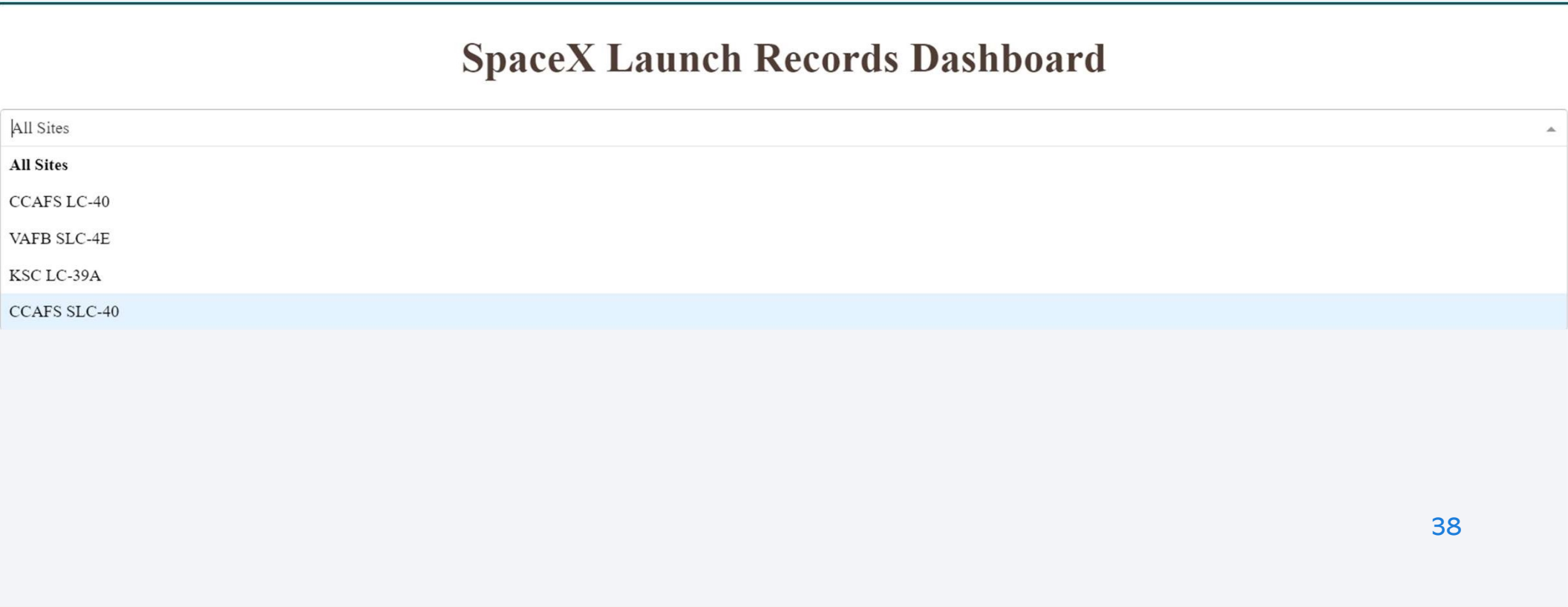
- Image on the left shows the distance between the east-most landing site in FL to the coastline.
- Image on the right shows distance of the other east coast landing site to nearby city, railway, road and coastline.



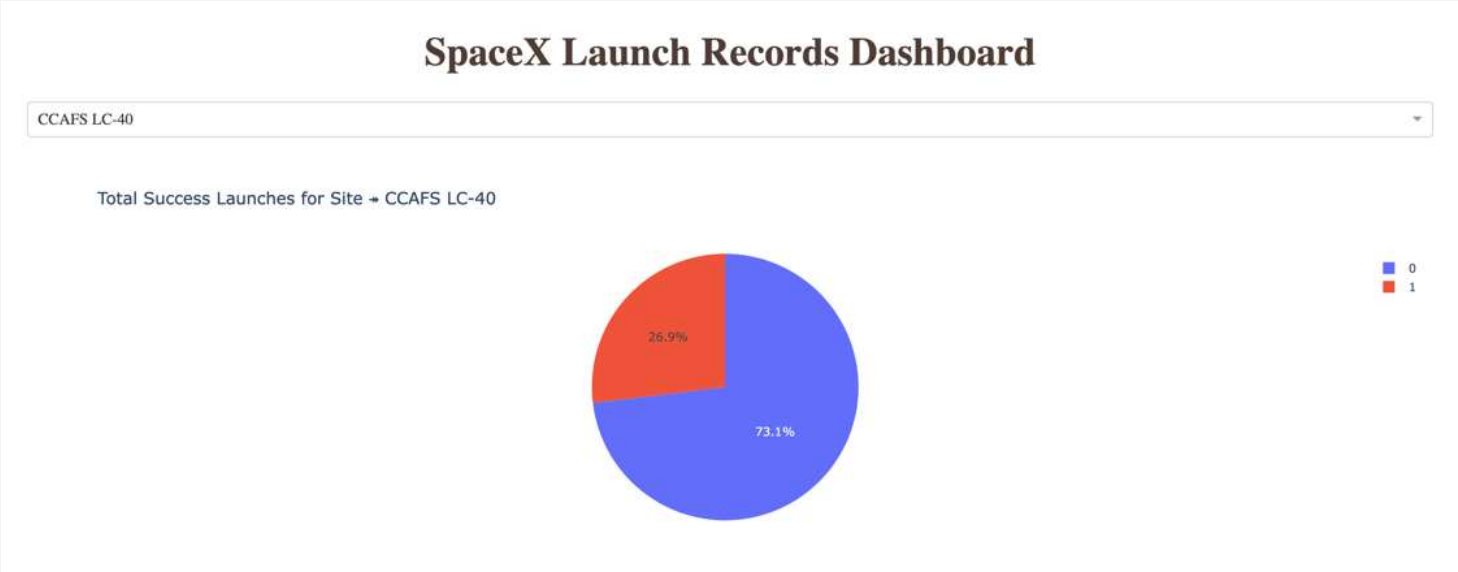
Section 4

Build a Dashboard with Plotly Dash

Dash Dropdown



<Dashboard Screenshot 2>



Shows the screenshot of the piechart for the launch site with highest launch success ratio (CCAFS LC-40) where 73% of the launch missions succeeded.

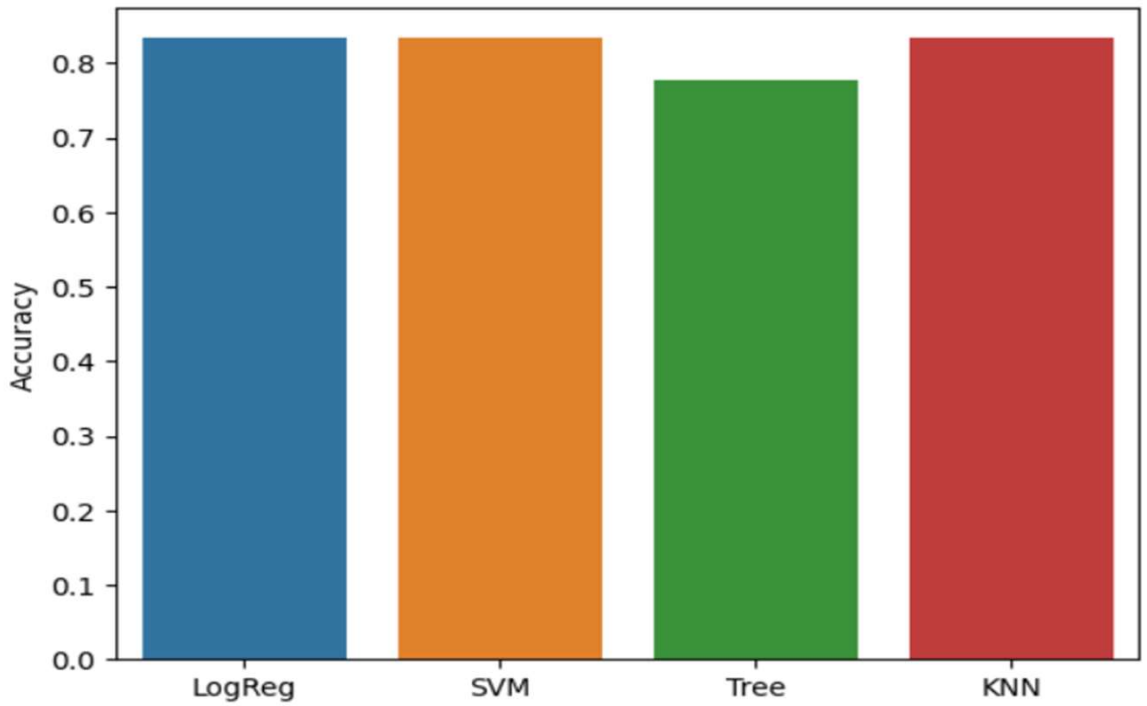
Success vs Payload

The picture below shows a scatterplot when the payload mass range is set to be from 4000 to 10000kg. Class 0 represents failed launches while class 1 represents successful launches.

Section 5

Predictive Analysis (Classification)

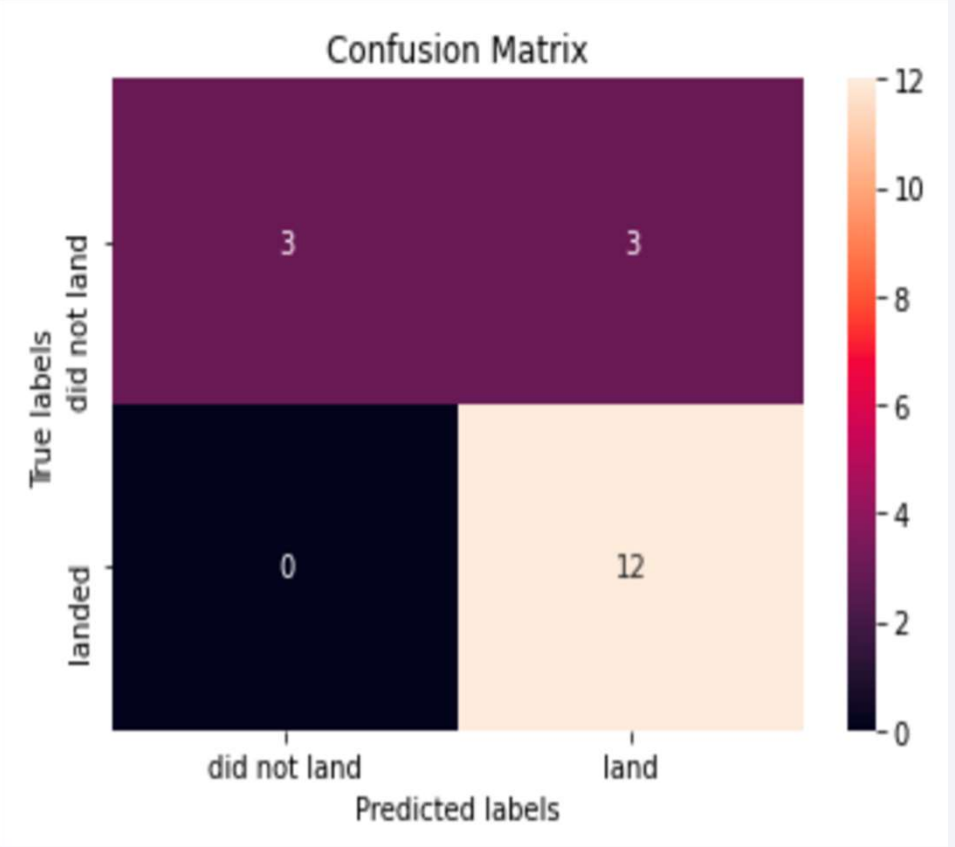
Classification Accuracy



- We can see that Decision tree model is underperforming on test data, while the rest three are equal in their accuracy scores.

Confusion Matrix

- All the three best performing models result in the same confusion matrix.
- These models have 0 False negatives and 3 False Positives (i.e. the model predicted they landed when in reality they did not)
- Since in our estimation and cost-reduction goals we don't want to be overconfident and want to err on the side of caution, having no false negatives is a great outcome.



Conclusions

- In this project, we try to predict if the first stage of a given Falcon 9 launch will land in order to determine the cost of a launch.
- Each feature of a Falcon 9 launch, such as its payload mass or orbit type, may affect the mission outcome in a certain way.
- Several machine learning algorithms are employed to learn the patterns of past Falcon 9 launch data to produce predictive models that can be used to predict the outcome of a Falcon 9 launch.
- The predictive model produced by decision tree algorithm performed the worst among the 4 machine learning algorithms employed.

Thank you!