IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Gordienko Aleksei
27.07.2023

Outline

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

Executive Summary

Summary of methodologies

Data Collection through SpaceX API

Data Collection through Web Scraping

Data Wrangling

Exploratory Data Analysis with SQL

Exploratory Data Analysis with Data Visualization

Interactive Visual Analytics with Folium and Plotly Dash

Machine Learning Predictions

Summary of all results

EDA Results

Interactive Analytics in screenshots

Predictive Analysis Results

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 you want to find answers

- Which factors determine if the first stage will land successfully?
- How different factors influence the success rate?
- What condition should be in place to maximize the success rate?

Section 1

Methodology

Methodology

- Executive Summary
- Data collection methodology:
  - Data was collected through SpaceX API and Web Scraping from Wikipedia
- Perform data wrangling
  - Most of the data were converted to numeric one using OneHot Encoding.
- 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

- Most of the data was collected through get request to SpaceX API
- The response was collected and decoded as Json file, using .json() method, and converted to pandas DataFrame, using .json_normalize()
- Then, the data was filtered and processed to contain only information necessary
- In addition, web scraping from Wikipedia was done, using BeatifulSoup library, to obtain Falcon 9 launch records
- HTML table was collected and converted to pandas DataFrame for further analysis

Data Collection – SpaceX API

- We used get request to SpaceX API to collect data, then we cleaned and filtered the data, and did some basic data wrangling.

- Link to Jupyter notebook: [SpaceX API Data Collection]()

1. Get request for launch data using SpaceX API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

2. Convertion of the Json file to pandas DataFrame

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

3. Additional calls to SpaceX API

```
# Obtains Booster Version name from SpaceX API using rocket serial number in 'data'
getBoosterVersion(data)
# Obtains Launch Site name and Location from SpaceX API using launchpad serial number in 'data'
getLaunchSite(data)
# Obtains Payload Data from SpacesX API using 'data'
getPayloadData(data)
# Obtains core information about launches using SpaceX API
getCoreData(data)
```

4. Dealing with missing values after data filtering

```
# Calculate the mean value of PayloadMass column
mean_mass = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9.loc[:,'PayloadMass'].replace(np.nan, mean_mass, inplace = True)
data_falcon9.isnull().sum()
```

Data Collection - Scraping

- We exported HTML content of the Wikipedia page containing Falcon 9 launch record table.
- Using BeatifulSoup library we converted this table to pandas DataFrame.
- Link to Jupyter Notebook: [Web Scraping for Falcon 9 launches]()

1. Obtaining HTML content of the Wikipedia page

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
response = requests.get(static_url)
soup = BeautifulSoup(response.text)
```

2. Table heads extraction

```
html_tables = soup.find_all('table')
first_launch_table = html_tables[2]
column_names = [ extract_column_from_header(column) for column in first_launch_table.find_all('th')
                    if extract_column_from_header(column) is not None and len(extract_column_from_header(column)) > 0]
```

3. Table content extraction

4. Convertion to pandas DataFrame

Data Wrangling

We performed Exploratory Data Analysis and determined training labels

We calculated number of launches at each launch site, and, number and occurrence of each orbit type

We created outcome label to distinguish successes and failures

Link to Jupyter notebook: [SpaceX Data Wrangling]

EDA with Data Visualization

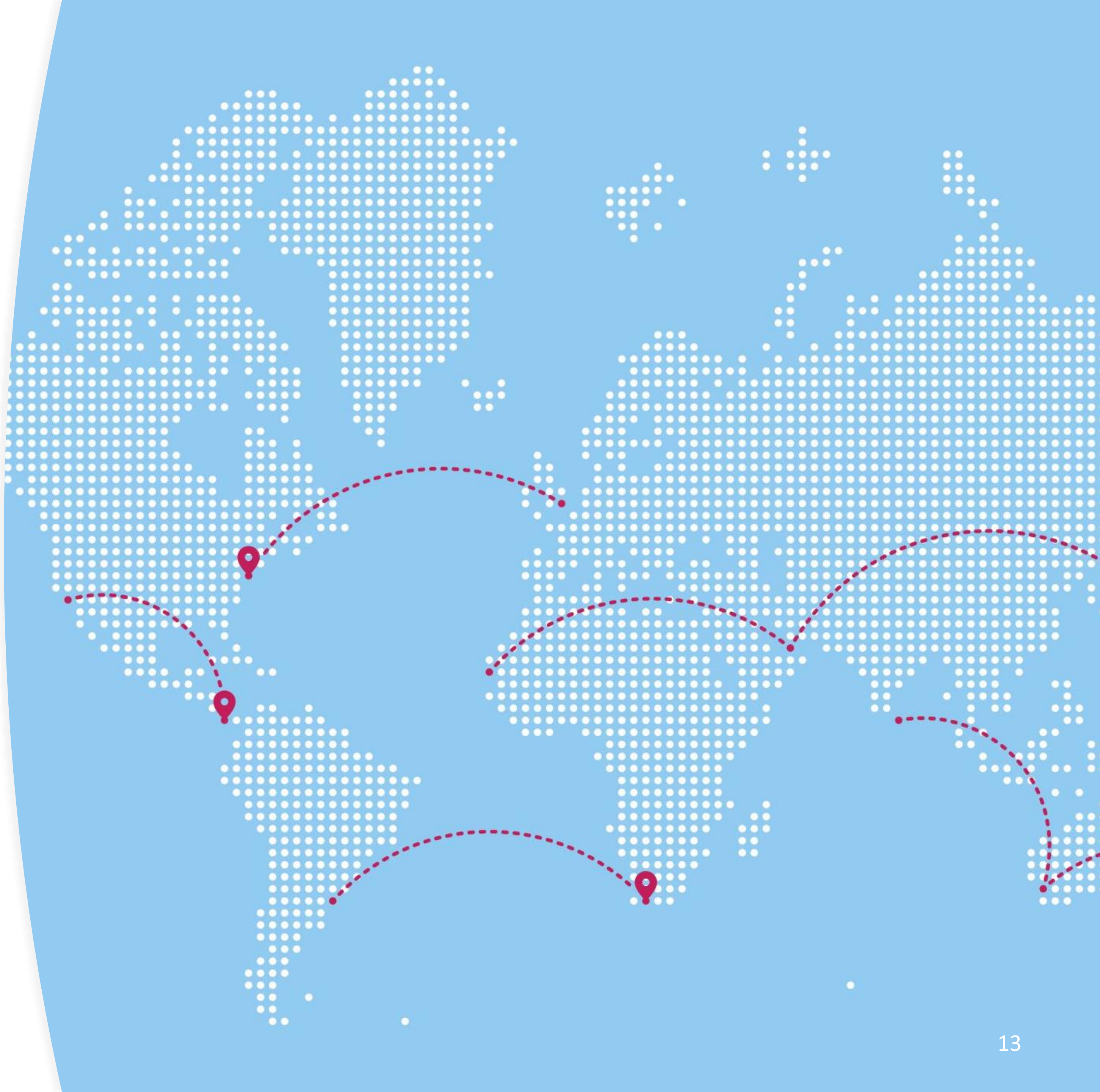
- We explored the data by visualizing the relationships between:
  - Flight number and Launch Site, to see how many successes and failures were at each site;
  - Payload Mass and Launch Site, to see how payload mass influence the outcome at each launch site;
  - Success rate and Orbit type, to see how destination influence the outcome;
  - Flight number and Orbit type, for the same reason;
  - Payload Mass and Orbit type;
- We also explored success rate yearly dynamic to see the yearly progress of the first stage landing.
- Link to Jupyter notebook: SpaceX EDA with Data Visualization

EDA with SQL

- We performed SQL queries to get the insights of the data:
  - The names of the unique launch sites in space missions;
  - The total payload mass carried by boosters launched by NASA (CRS);
  - The average payload mass carried by booster version F9 v1.1;
  - The total number of successful and failure mission outcomes;
  - Ranked total landing outcomes between 2010-06-04 and 2017-03-20.
- Link to Jupyter notebook: SpaceX EDA with SQL

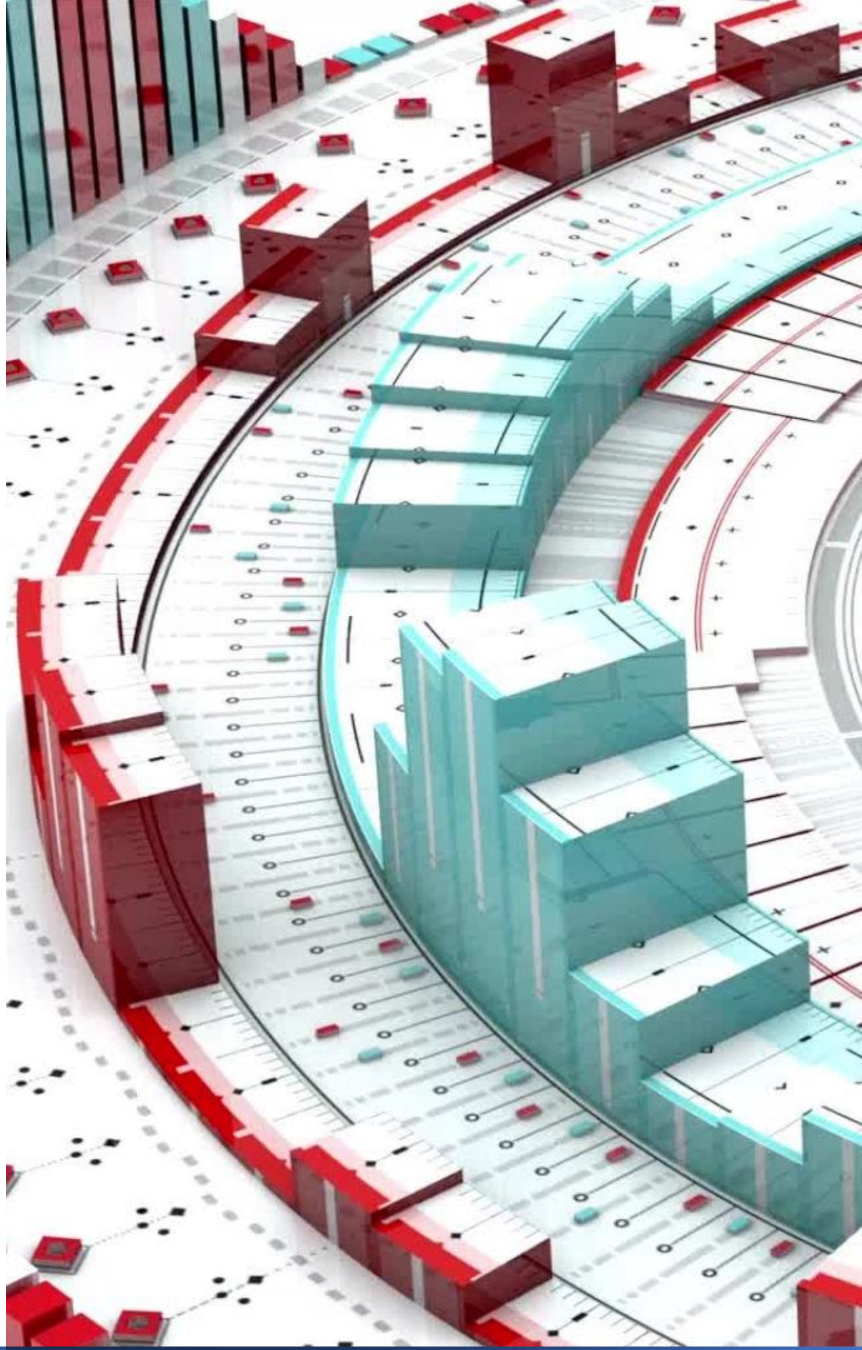
Build an Interactive Map with Folium

- We created a world map where we marked launch sites locations to visualize how close they are to the equator;
- We also added markers to display all launches from each site with corresponding color-label to show if it was successful or not;
- Finally, we added line with distance from Kennedy Space Center to nearest airport.
- Link to Jupyter notebook: SpaceX Launch Sites map



Build a Dashboard with Plotly Dash

- We created interactive pie chart:
    - It shows successful launches for all launch sites;
    - It can also show successes and failures for specific launch site.
- We also created a scatter plot that shows successful and failed landings depending on payload mass:
    - The specific payload mass range can be chosen as well as launch site;
    - It also shows what booster version was used in a launch.
- Link to Python code: [SpaceX Dash App Code]



Predictive Analysis (Classification)

- We loaded and transformed the data using numpy and pandas, then splitted the data to train and test sets.
- We built different classification models and tuned their hyperparameters using GridSearchCV.
- We also found the best performing model using accuracy as a main criteria.
- Link to Jupyter notebook: SpaceX Classification Models

Results

Exploratory data analysis results

Interactive analytics demo in screenshots

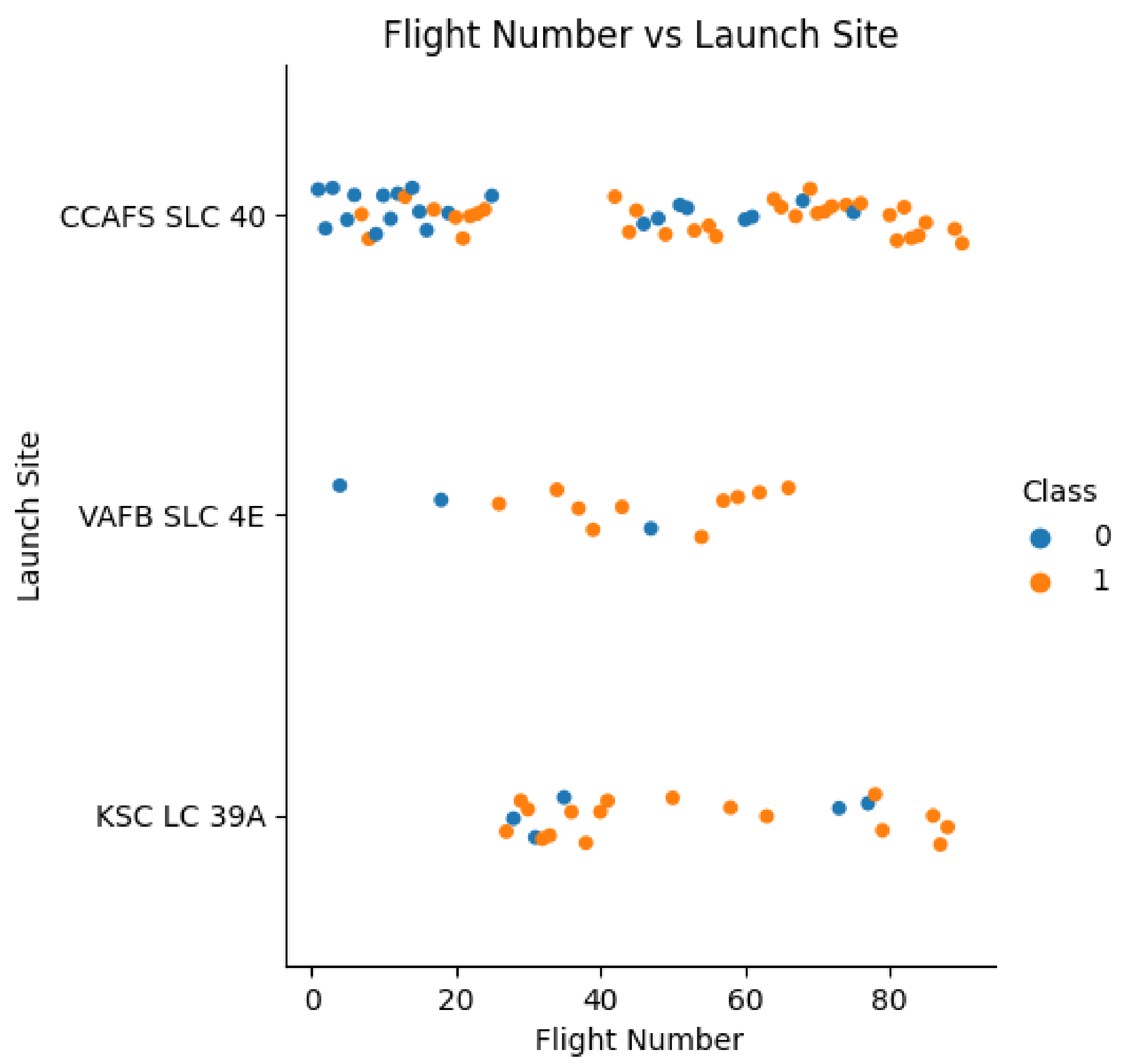
Predictive analysis results

Section 2

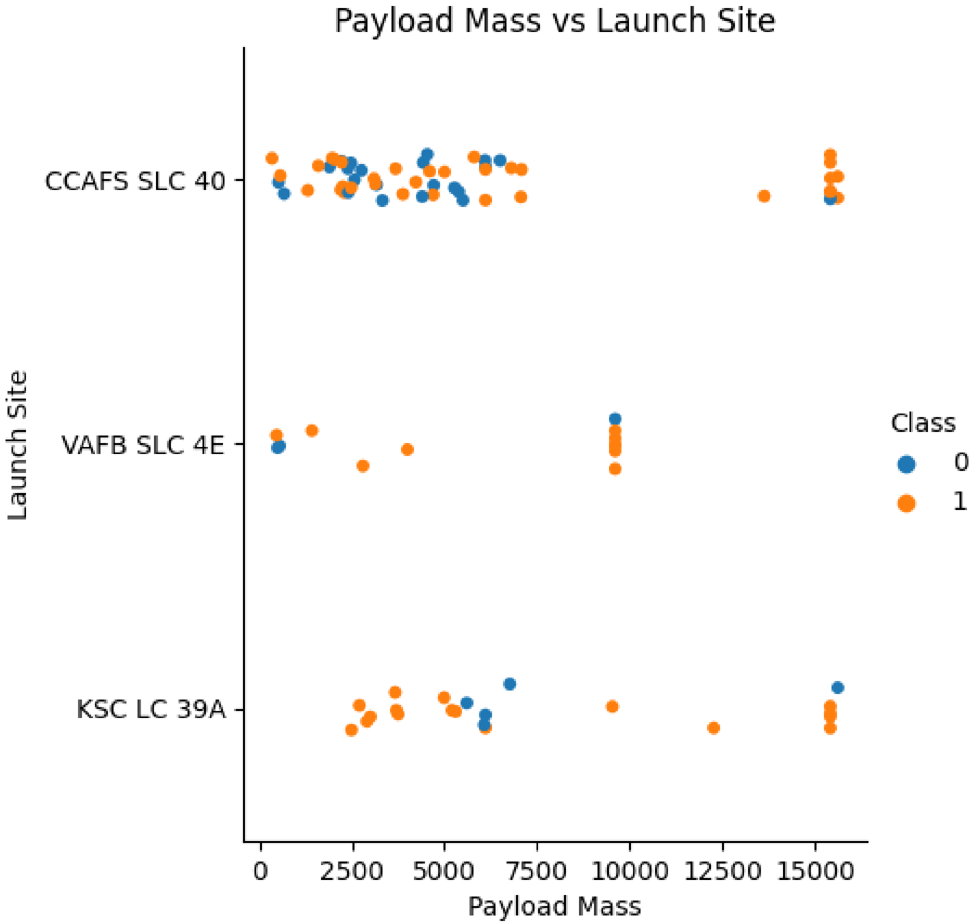
Insights drawn from EDA

Flight Number vs. Launch Site

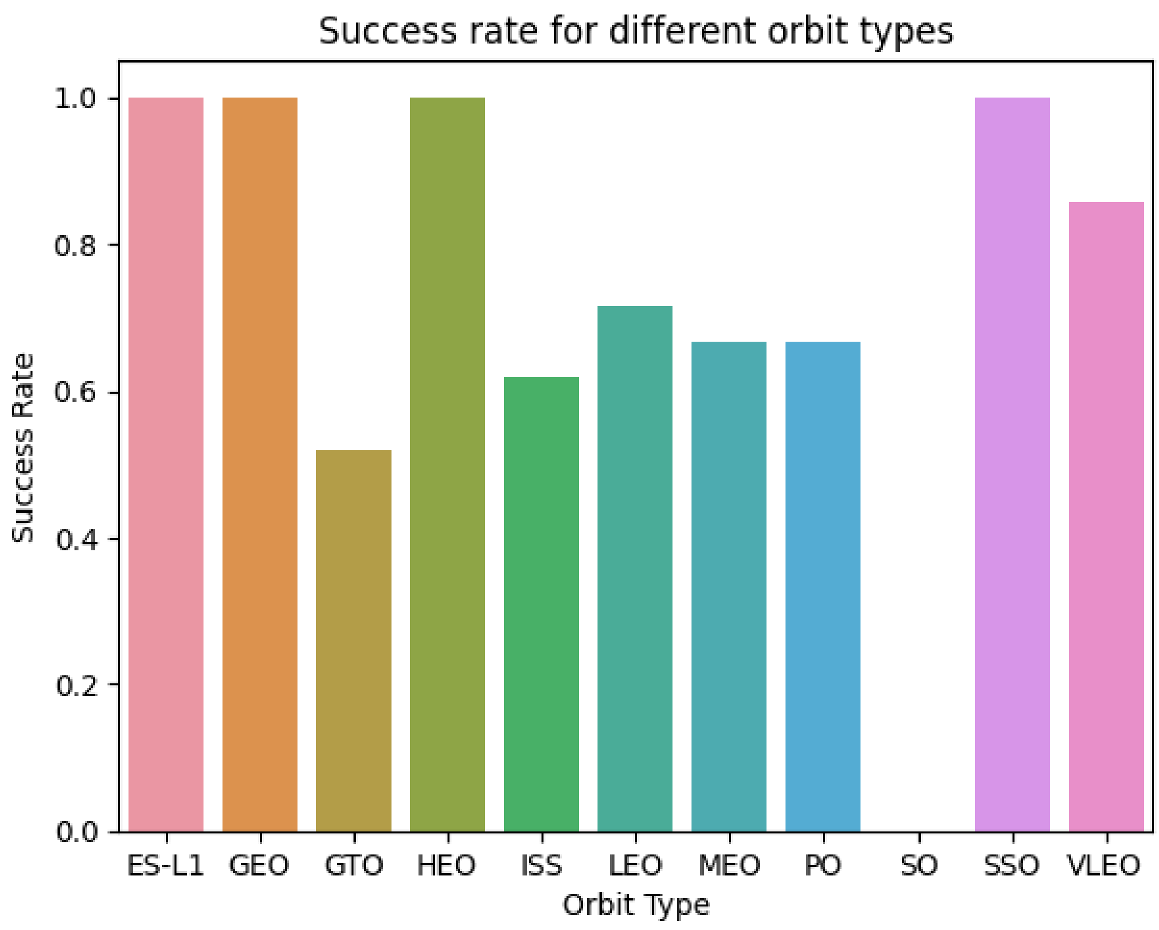
- CCAFS SLC 40 launch site has the most number of failed landings (blue dots)
- KSC LC 39A launch site has the most number of successes relative to the number of launches (orange dots)
- VAFB SLC 4E launch site has the least number of failed landings (blue dots)



Payload vs. Launch Site



- We see that flights with a greater payload mass have a greater chance of the success
- We also see that flights launched from KSC LC 39A with a relatively small payload mass have a great success rate
- But for CCAFS SLC 40 it’s the opposite

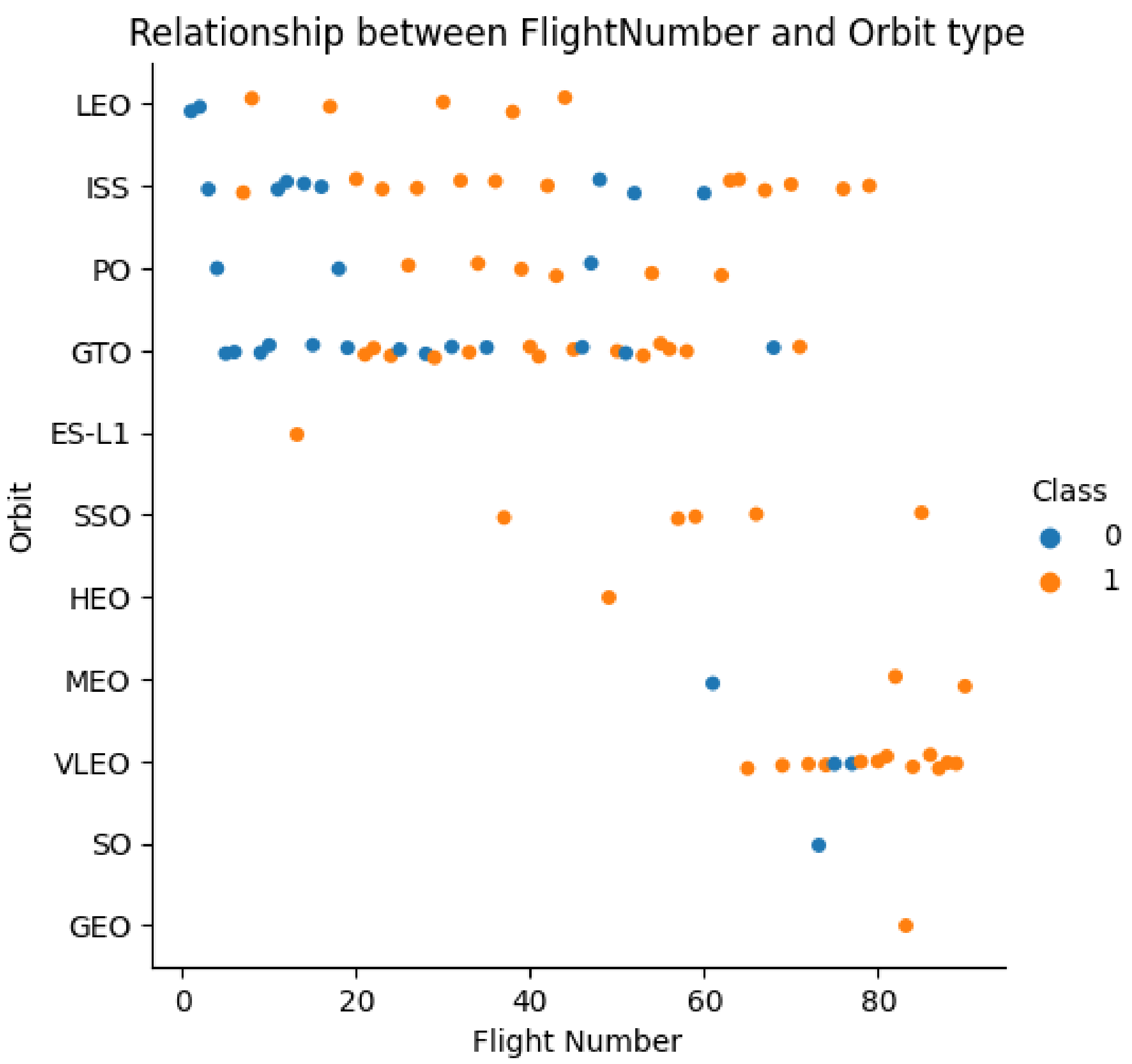


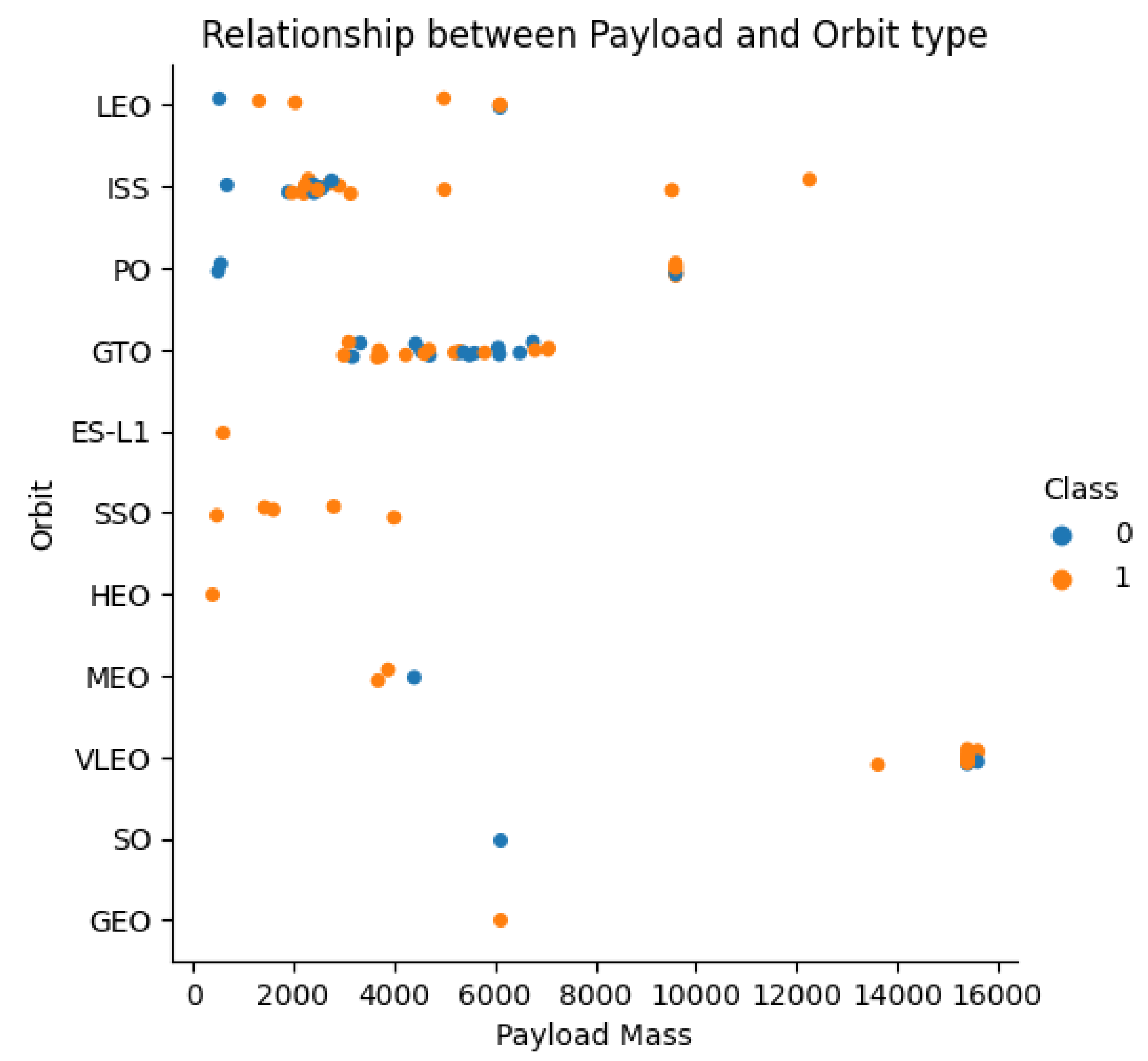
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO orbit types have the highest success rate
- GTO and ISS the lowest

Flight Number vs. Orbit Type

- In the LEO orbit we see that success is related to the number of flights
- For the GTO orbit we don’t see that relationship



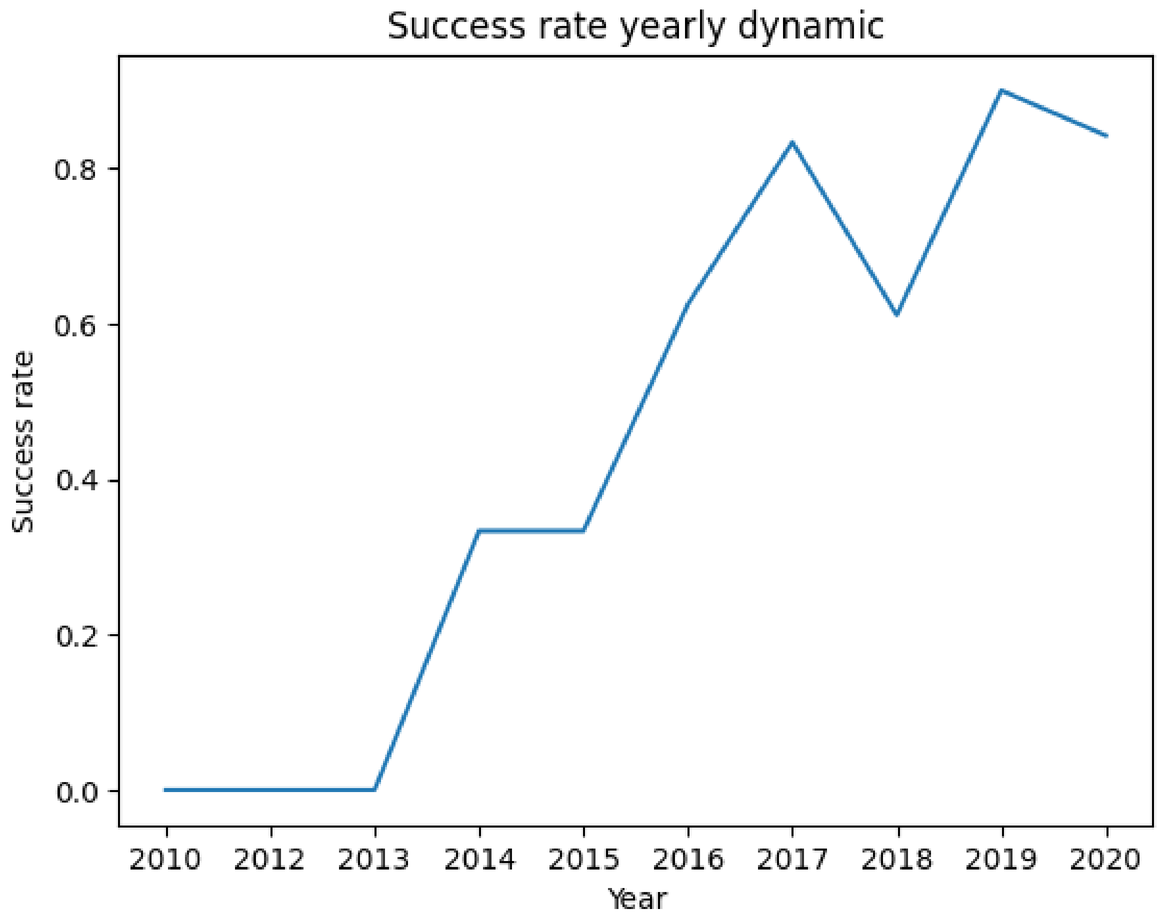


Payload vs. Orbit Type

- With heavy payloads the successful landing rate are more for Polar, LEO and ISS orbit types
- For GTO we cannot distinguish this well

Launch Success Yearly Trend

- Starting from 2013 we can see that the success rate has increased ever since
- It also had a small decline in 2018



All Launch Site Names

- We used keyword **DISTINCT** to select only unique launch site names

```
%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTBL
```

✓ 0.0s

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

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

Launch Site Names Begin with 'CCA'

- We used keyword **LIKE** and LIMIT to display 5 rows with launch site name beginning with 'CCA'

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

✓ 0.0s Python

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
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	Failure (parachute)
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No attempt
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attempt
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Using the query below we calculated that the total NASA payload was 45,596 kg

```
%%sql SELECT SUM("PAYLOAD_MASS__KG_") AS 'Total Payload by NASA'
        FROM SPACEXTBL WHERE "Customer" = 'NASA (CRS)' GROUP BY "Customer"
```

✓ 0.0s

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

Total Payload by NASA
45596.0

Average Payload Mass by F9 v1.1

- We calculated, in a query below, that the average payload mass for booster F9 v1.1 is equal to 2,928.4

```
%%sql SELECT AVG("PAYLOAD_MASS__KG_") AS 'Average Payload mass by F9 v1.1'
            FROM SPACEXTBL WHERE "Booster_Version"='F9 v1.1'
```

✓ 0.0s

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

Average Payload mass by F9 v1.1
2928.4

First Successful Ground Landing Date

- We found that the first successful ground landing happened on 22nd of December 2015

```
%sql SELECT MAX("Date") FROM SPACEXTBL WHERE "Landing_Outcome" = 'Success (ground pad)'
```

✓ 0.0s

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

MAX(Date)
22/12/2015

Successful Drone Ship Landing with Payload between 4000 and 6000

- We used the **WHERE** clause to filter for boosters which have successfully landed on drone ship and applied the **AND** and **BETWEEN** conditions to filter the payload mass

```
%%sql SELECT "Booster_Version" FROM SPACEXTBL
        WHERE "Landing_Outcome" = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000
```

✓ 0.0s

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

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

Total Number of Successful and Failure Mission Outcomes

- We can see that there was only one failed mission with a 100 successful ones

```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") FROM SPACEXTBL GROUP BY "Mission_Outcome"
```

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

Mission_Outcome	COUNT(Mission_Outcome)
None	0
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- To obtain results we used subquery and **MAX** function

```
%%sql SELECT "Booster_Version" FROM SPACEXTBL WHERE
    "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

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

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

- We used **WHERE** clause to filter the results as well as **AND** and **LIKE** conditions. We also used **SUBSTR** function to display months

```
%%sql SELECT substr("Date", 4,2) as "Month", "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTBL
                WHERE substr("Date",7,4)='2015' AND "Landing_Outcome" LIKE '%drone ship%'
```

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

Month	Landing_Outcome	Booster_Version	Launch_Site
10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
06	Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40

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

- We selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter the dates.
- We applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.

```
%%sql SELECT "Landing_Outcome", COUNT("Landing_Outcome") AS "Total_Outcomes" FROM  SPACEXTBL
            WHERE (CAST(substr("Date", 7,4) AS INT)*100+CAST(substr("Date", 4,2) AS INT)) BETWEEN 201006 AND 201703
            GROUP BY "Landing_Outcome" ORDER BY "Total_Outcomes" DESC
```

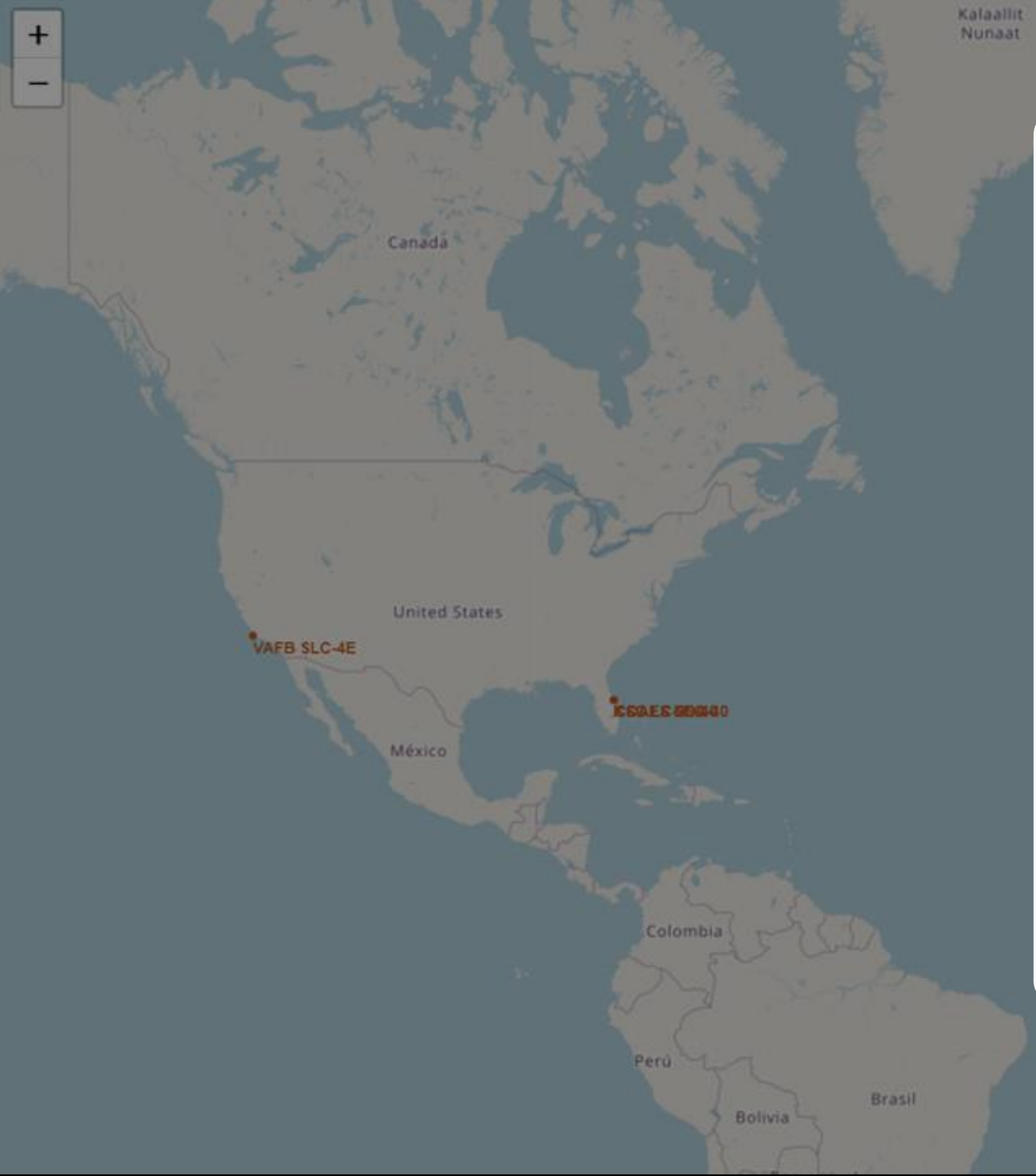
✓ 0.0s

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

Landing_Outcome	Total_Outcomes
No attempt	10
Success (drone ship)	6
Success (ground pad)	5
Failure (drone ship)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

Section 3

Launch Sites Proximities Analysis



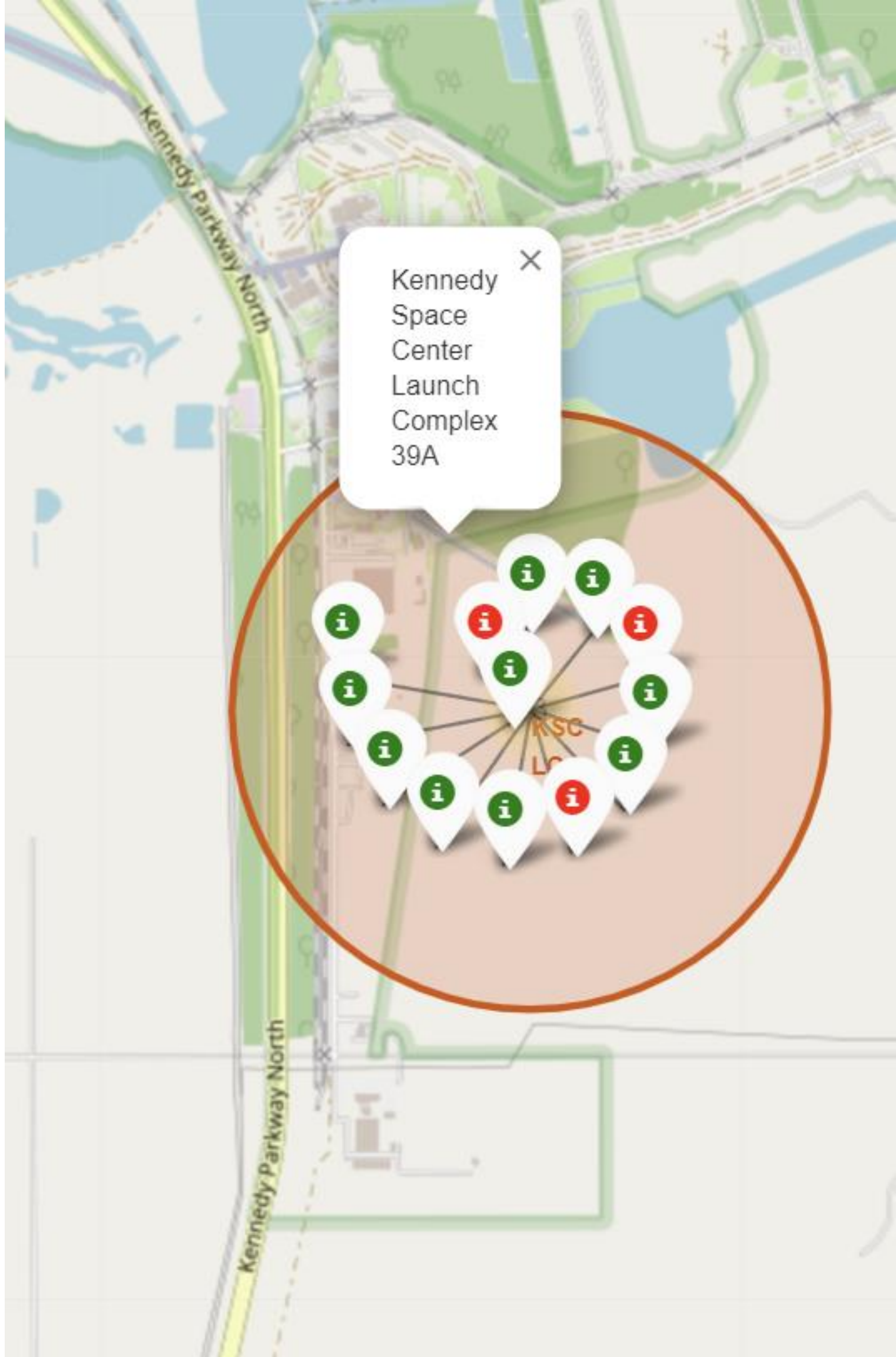
Launch Sites Location

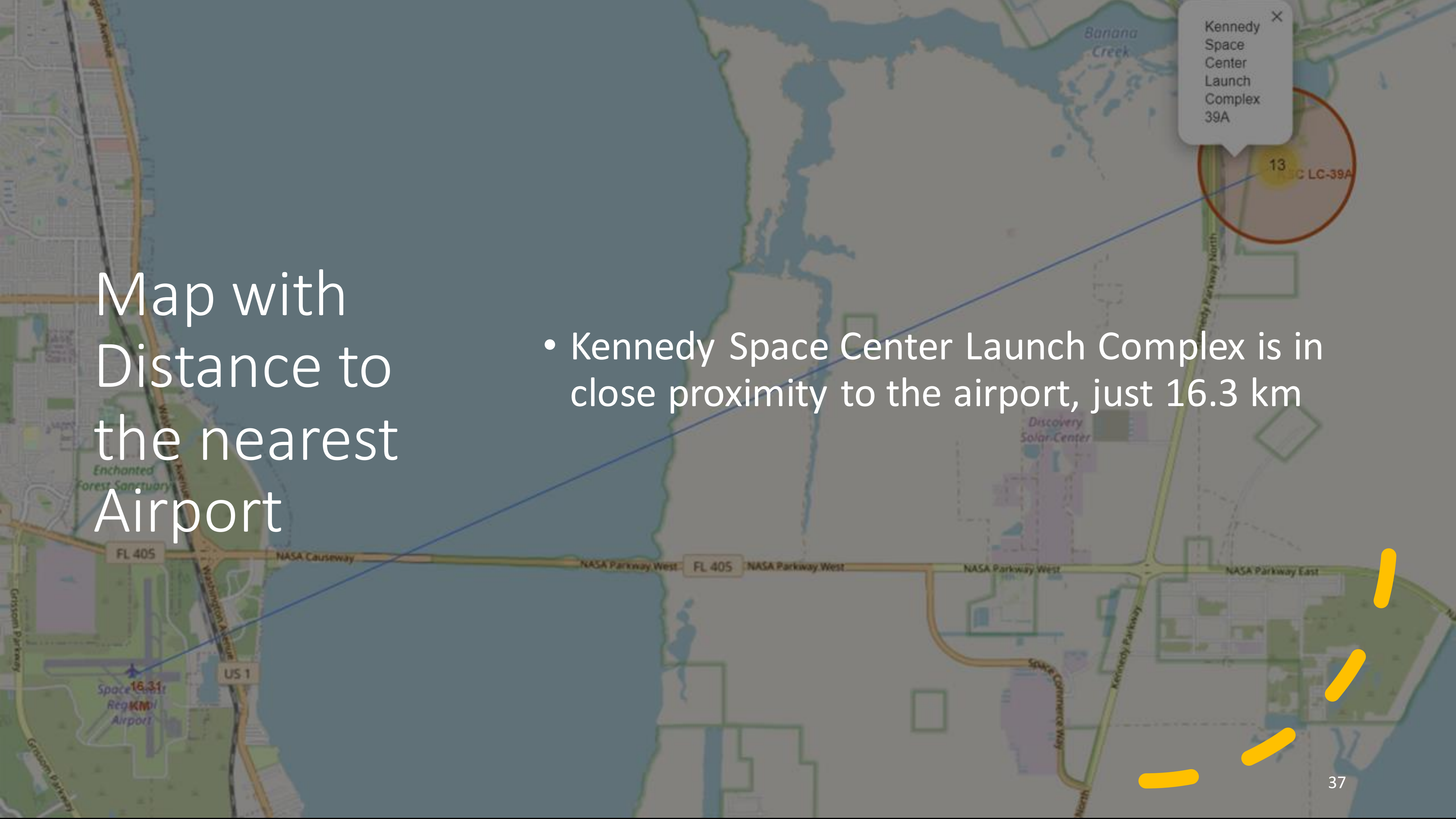
On the map we can see that all launch sites are located in the USA:

- 1 in California
- 3 in Florida

Map with Colored Markers for Each Launch

- Each launch site on the map has markers with different color: Green if landing was successful and Red if it was a failure.



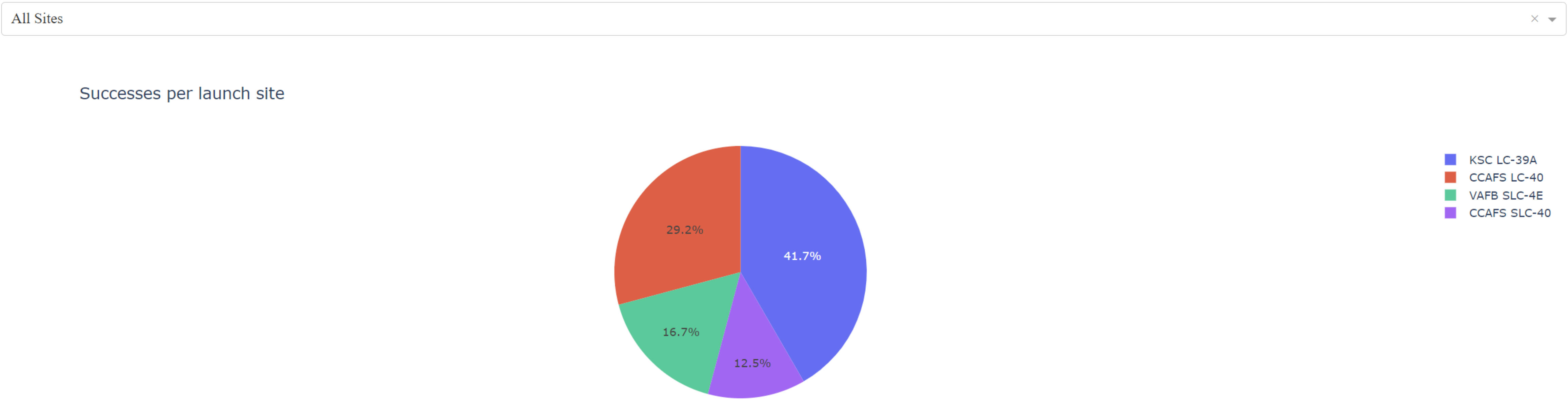


Map with Distance to the nearest Airport

- Kennedy Space Center Launch Complex is in close proximity to the airport, just 16.3 km

Section 4

Build a Dashboard with Plotly Dash

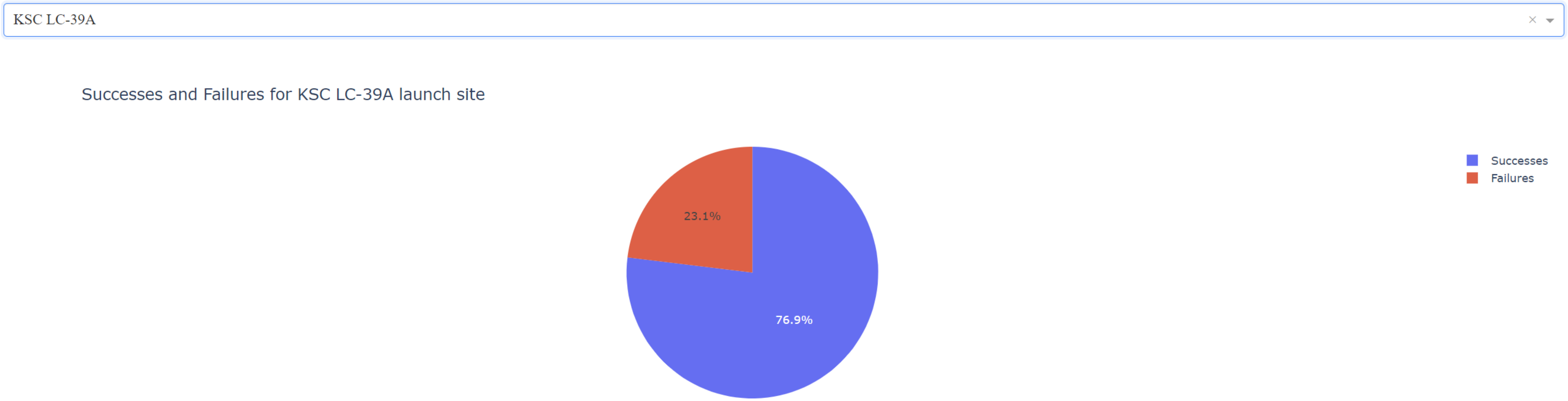


Successful Launches at each Launch Site

- KSC LC-39A has the most number of successful launches
- CCAFS LC-40 is the second most

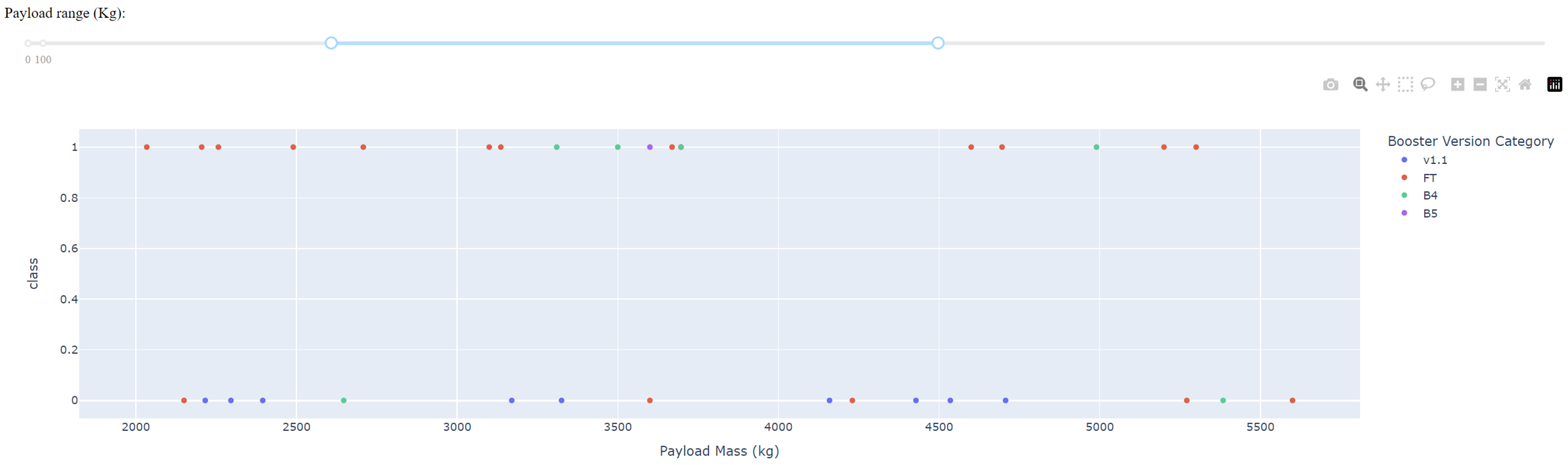
The Highest Success Ratio

- KSC LC-39A has the highest success ration
- We see that 76.9% of all launches from that site have a successful landing



Payload Range Scatter Plot

- Payload range is set between 2000 and 6000 kg
- Booster version v1.1 has 0 successes in that payload range
- Version FT has the most number of successful landing in that range

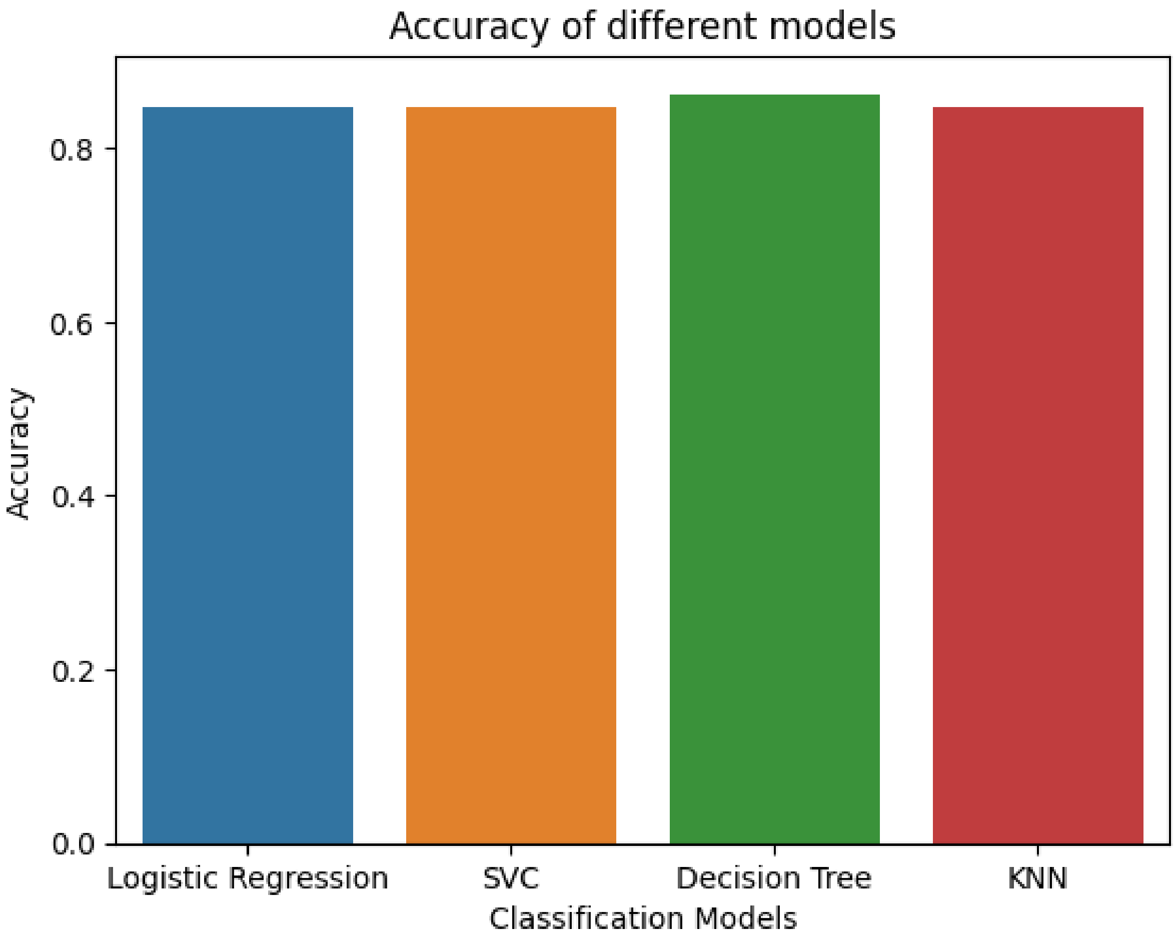


Section 5

Predictive Analysis (Classification)

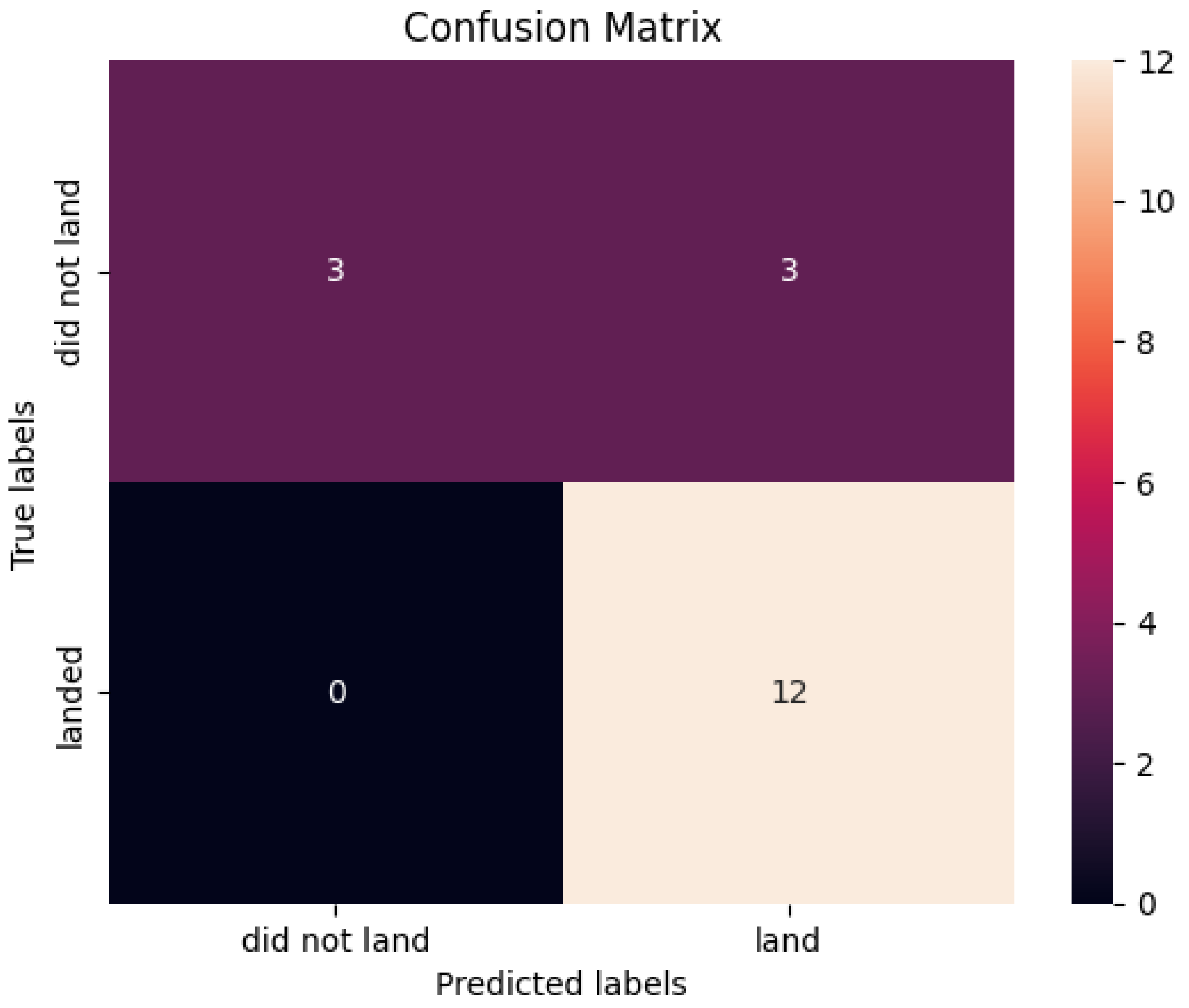
Classification Accuracy

- All models have practically the same accuracy, above 80%
- But Decision Tree has the highest accuracy, so it is more suited for our task



Confusion Matrix

- Decision Tree can distinguish between different classes and has zero False Negative
- But it has False Positive predictions, i.e. unsuccessful landings marked as successful by the classifier.



Conclusions

Launches with higher payload mass have a higher success rate.

Launch success rate started to increase in 2013 till 2020.

Orbits ES-L1, GEO, HEO, SSO had the most success rate.

KSC LC-39A had the most successful launches of any sites.

The Decision tree classifier is the best machine learning algorithm for this task.

Appendix

- All project files, images and datasets can be obtained in GitHub repository

Thank you!