IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science


<Xin He>
<16-06-2022>

# Outline

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

# Executive Summary

- **Summary of methodologies and results:**

1) collected data using an API and using web scraping;
2) data wrangling (transformed data format, dealt with missing data, and etc.);
3) exploratory analysis and data visualization using SQL and using Pandas and Matplotlib;
4) built a dashboard and a map for interactive visual analysis
5) Built and optimized different machine learning models (KNN, SVM, logistic regression, and decision tree). Trained and tested the models, and found the model which performs best using the test data.

# Introduction

- **Project background and context:**

Rocket launches are generally expensive and risky. Recently SpaceX successfully developed the techniques to reuse the first stage, which can reduce the launch cost to 62 million dollars, compared to upward of 165 million dollars from other providers. Therefore, if we can predict whether the first stage will land with what probability, we can determine the cost of a launch. This information can be useful to other companies, which want to bid against SpaceX.

- **Problems you want to find answers**

Predict whether the first stage of Falcon 9 can land or not. Furthermore, what is the landing success rate? How does the success rate depend on landing sites, booster versions, payload mass, and orbits?

Section 1

# Methodology

# Methodology

## Executive Summary

- Data collection methodology:
  - Data collections can be independently done using two approaches: REST API, and web scraping.
- Perform data wrangling
  - Briefly browse the data to find some patterns and determine the mission outcome to be the label for machine learning. And convert the outcome to numeric.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Using sklearn library, we built five machine learning models (KNN, SVM, logistic regression, and decision tree)
  - We split the data into train and test sets, and tune the model parameters using GridSearchCV to find the best one.
  - We use the test data to evaluate the optimised model.

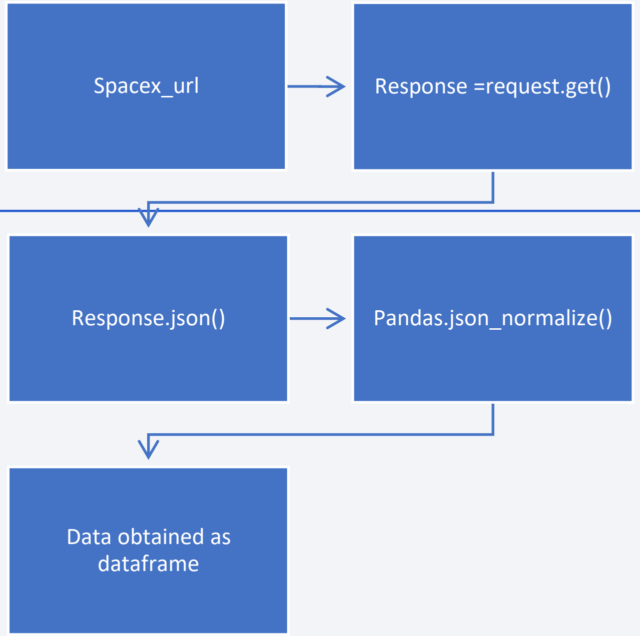
# Data Collection

- Describe how data sets were collected independently using two approaches:

- Used get request to the SpaceX REST API; decoded the response content using .json() function and convert the json format data into a pandas dataframe using pandas.json_normalize(); then cleaned data and replaced missing values; converted categorical values into numerical values for the labels of machine learning.

- Web scraped Falcon 9 launch records from Wikipedia using BeautifulSoup; created a data frame by parsing the launch HTML tables

# Data Collection – SpaceX API




- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- Add the GitHub URL of the completed SpaceX API calls notebook (https://github.com/EricHexin/capstone_project_data_science/blob/master/jupyter-labs-spacex-data-collection-api.ipynb), as an external reference and peer-review purpose

To make the requested JSON results more consistent, we will 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/
```

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

```
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
response = requests.get(static_json_url)
data = response.json()
data = pd.json_normalize(data)
```

Using the dataframe `data` print the first 5 rows

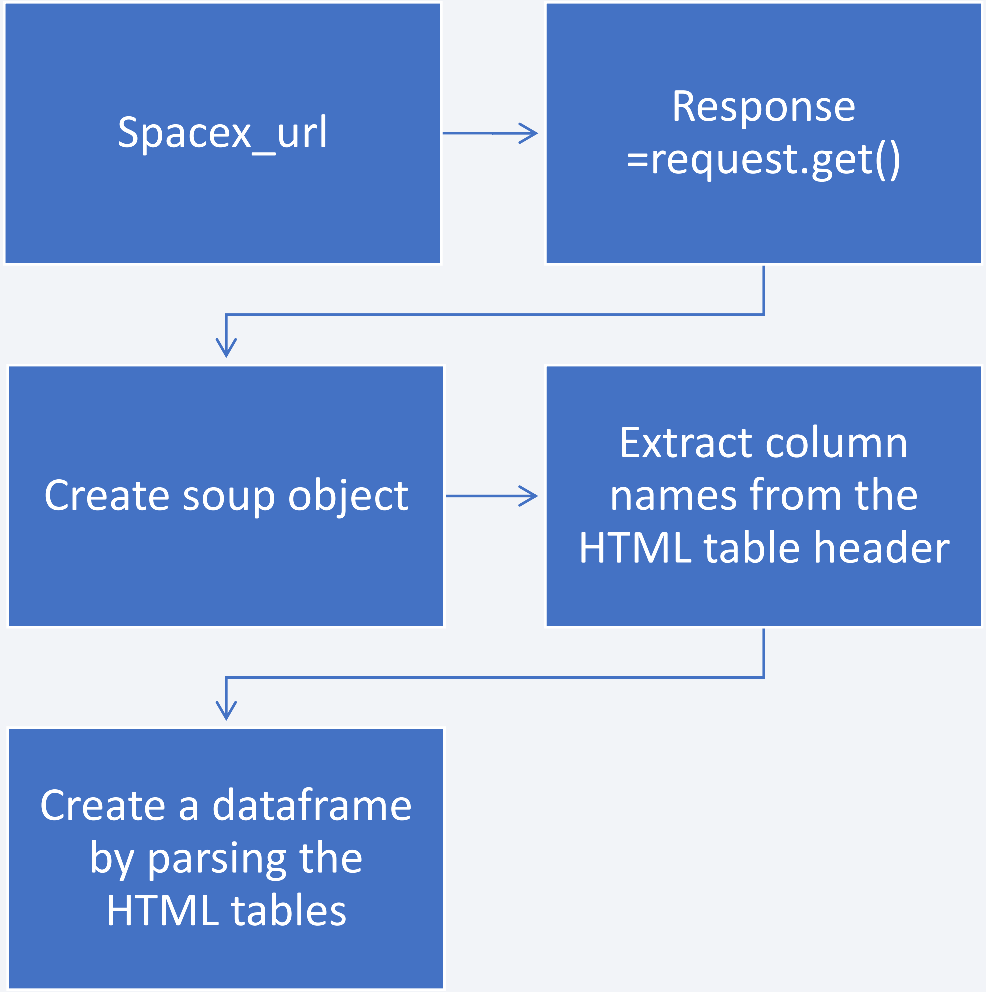
```
# Get the head of the dataframe
data.head()
```

| static_fire_date_utc | static_fire_date_unix | tbd | net | window | rocket | success | details | crew | ships | capsul |
|---|---|---|---|---|---|---|---|---|---|---|

# Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- https://github.com/EricHexin/capstone_project_data_science/blob/master/jupyter-labs-data_collection_webscraping.ipynb

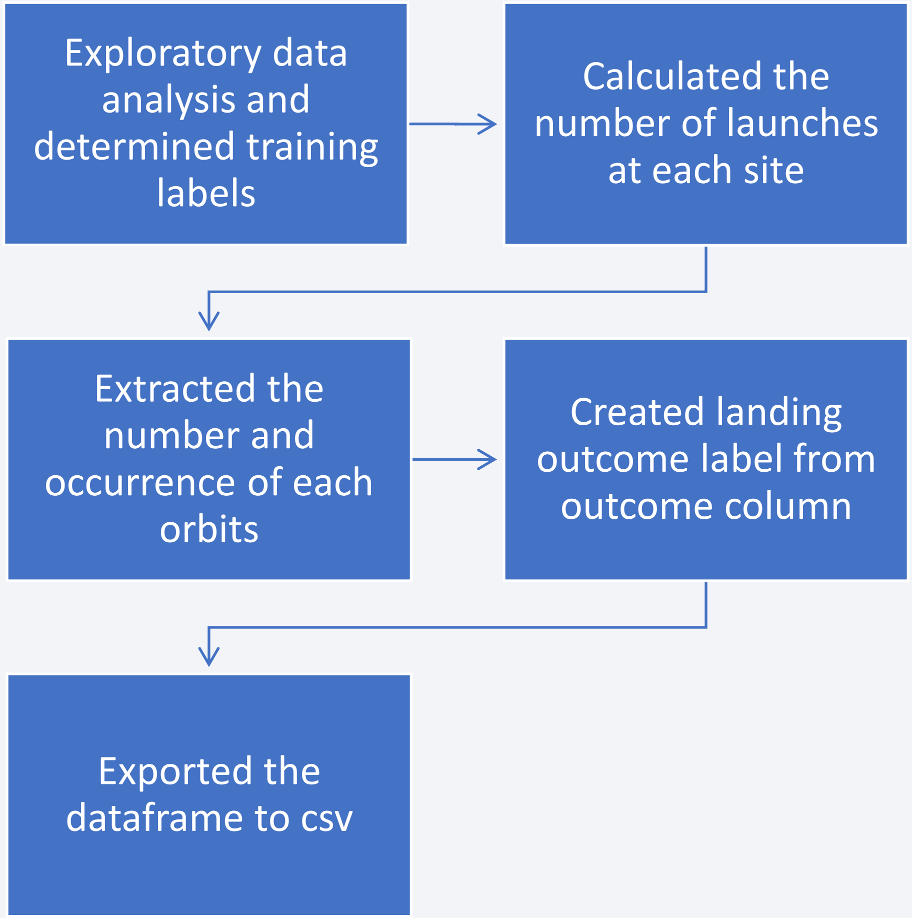


# Data Wrangling

- Describe how data were processed

Briefly browse the data to find some patterns and determine the mission outcome to be the label for machine learning. And convert the outcome to numeric.

- https://github.com/EricHexin/capstone_project_data_science/blob/master/labs-jupyter-spacex-Data%20wrangling.ipynb



# EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts

1) Payload mass vs flight number: We see that as the flight number increases, the first stage is more likely to land successfully. The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return

2) Flight number vs launch site: I found for the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).

3) Success rate vs orbit type: visually checked the relationship between success rate and orbit type, and which orbits have a high success rate

4) Flight number vs orbit type: I can visually see the number of successful flights for each orbit.

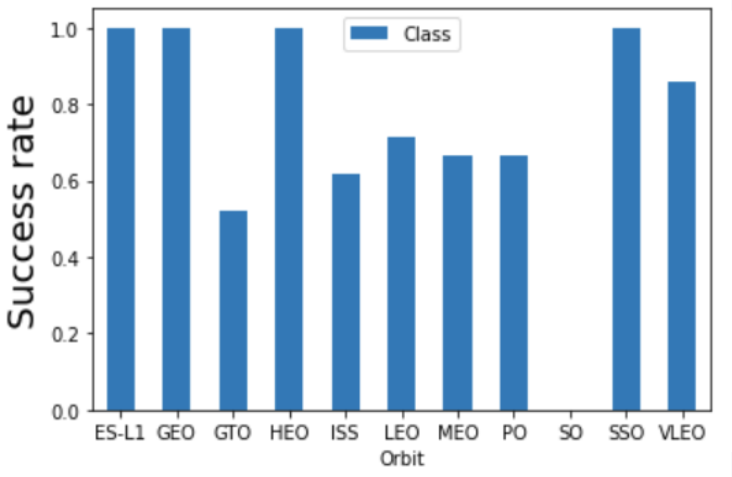
5) Payload vs orbit type: I found with heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.

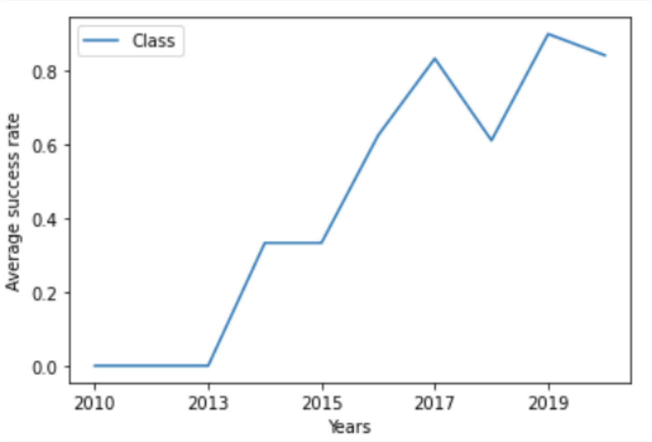
6) Chart of the yearly successful launch rate:

you can observe that the success rate since 2013 kept increasing

till 2020.

https://github.com/EricHexin/capstone_project_data_science/blob/master/jupyter-labs-eda-dataviz.ipynb



# EDA with SQL

- Using bullet point format, summarize the SQL queries you performed

    1) Names of the unique launch sites.
    2) 5 records where launch sites begin with the string 'CCA'
    3) Total payload mass carried by boosters launched by NASA (CRS)
    4) Average payload mass carried by booster version F9 v1.1
    5) the date when the first successful landing outcome in ground pad was achieved
    6) names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
    7) Total number of successful and failed mission outcomes
    8) the names of the booster_versions which have carried the maximum payload mass.
    9) the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015
    10) Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

    https://github.com/EricHexin/capstone_project_data_science/blob/master/jupyter-labs-eda-sql-coursera_sqllite.ipynb



1) %sql SELECT DISTINCT Launch_Site FROM SPACEXTBL;
2) %sql SELECT Launch_Site FROM SPACEXTBL where Launch_Site like 'CCA%' LIMIT 5;

# Build an Interactive Map with Folium

- Marked all launch sites, and the map objects I created are circles, makers, marker cluster, and polyline.
- Converted the categorical outcome column into numeric labels
- Assigned success and failure numbers to each launch site using the color-labeled marker clusters.
- Calculated distances from a launch site to its proximities, to know whether there are railways, highways, or coastlines, or whether the launch site is away from cities.
- https://github.com/EricHexin/capstone_project_data_science/blob/master/lab_jupyter_launch_site_location_folium_maps.ipynb

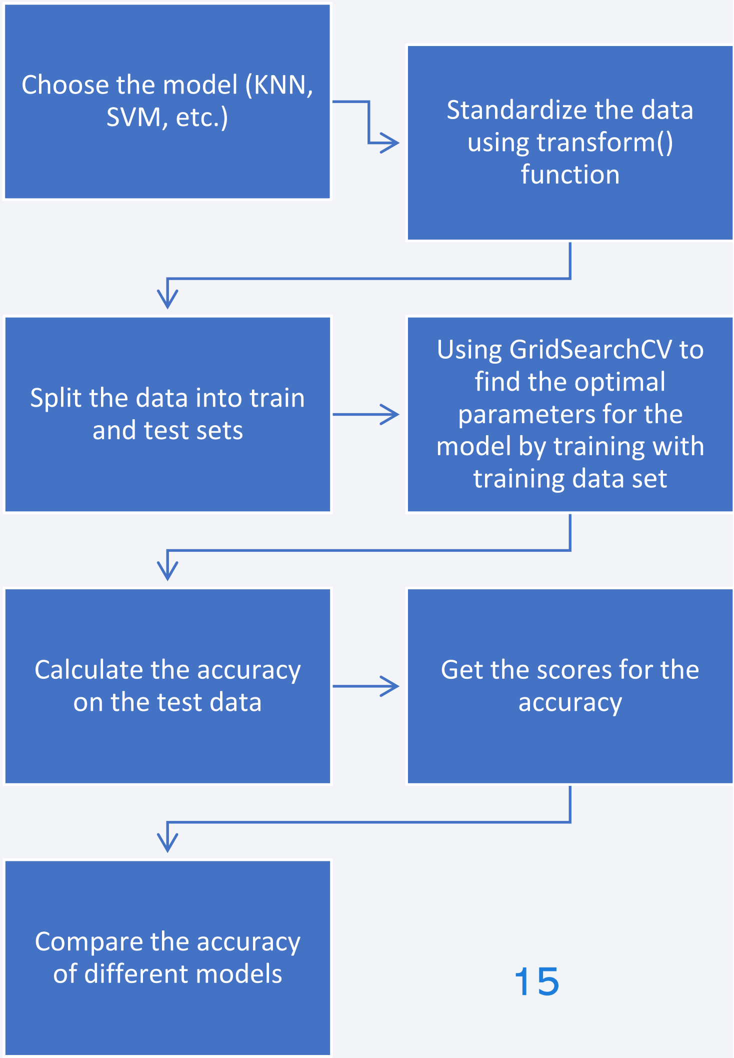
# Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Added a dropdown list to enable Launch Site selection
- Plotted a pie chart to show the total successful launches count for all sites
- Plotted a scatter chart to show the correlation between payload and launch success
- Reasons for the dashboard is to provide interactive visual analytics on SpaceX launch data in real time, so that we can get some quick insights about the data.
- https://github.com/EricHexin/capstone_project_data_science/blob/master/spacex_dash_app.py

# Predictive Analysis (Classification)

1) Built and optimized different machine learning models (KNN, SVM, logistic regression, and decision tree). Trained and tested the models, and found the model which performs best using the test data.

- https://github.com/EricHexin/capstone_project_data_science/blob/master/SpaceX_Machine%20Learning%20Prediction_Part_5_all_classifiers_GridSearchCV_best_param.ipynb



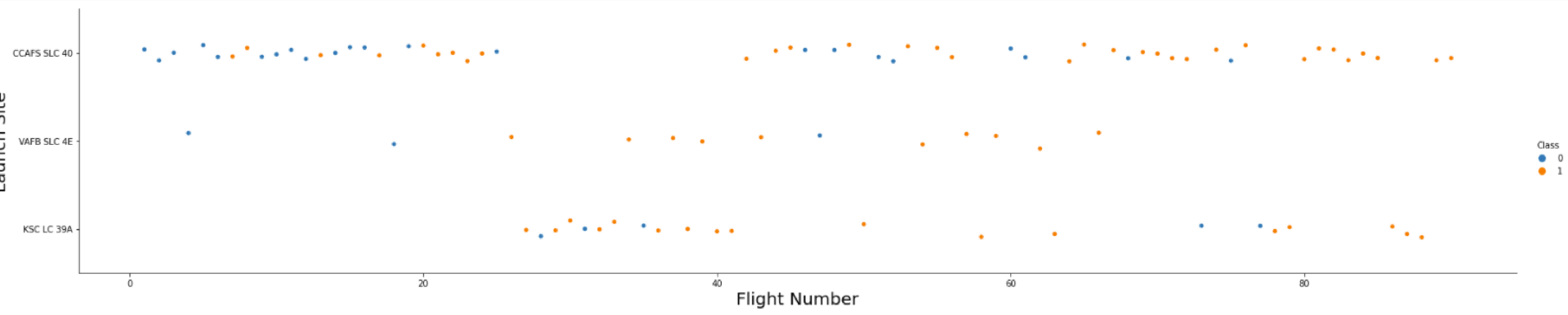
# Results

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

Section 2

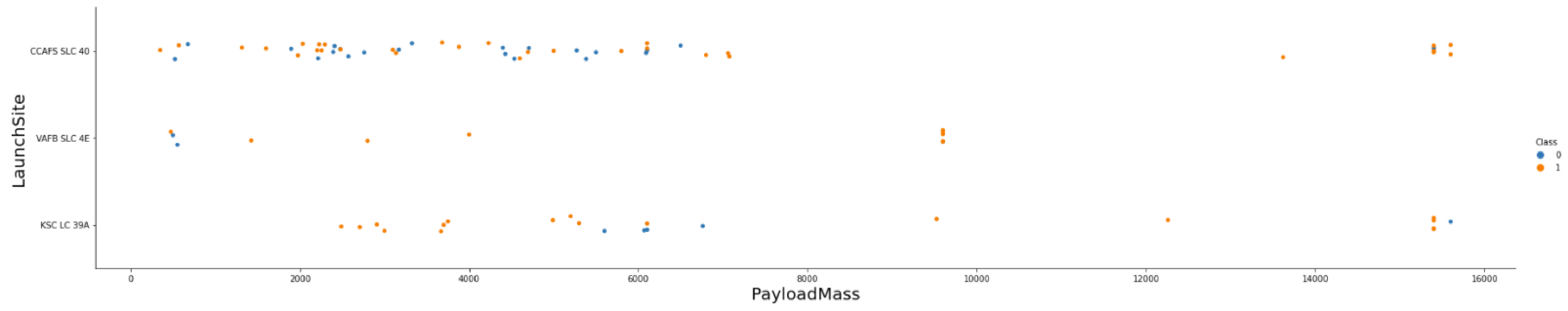
# Insights drawn from EDA

# Flight Number vs. Launch Site




1) Launch Site CCAFS SLC 40 has the most launches.
2) But VAFB SLC 4E has only 3 failed launches, although it has the least launches.
3) As the flight number increases, the failed launches appear to be rarer.

# Payload vs. Launch Site

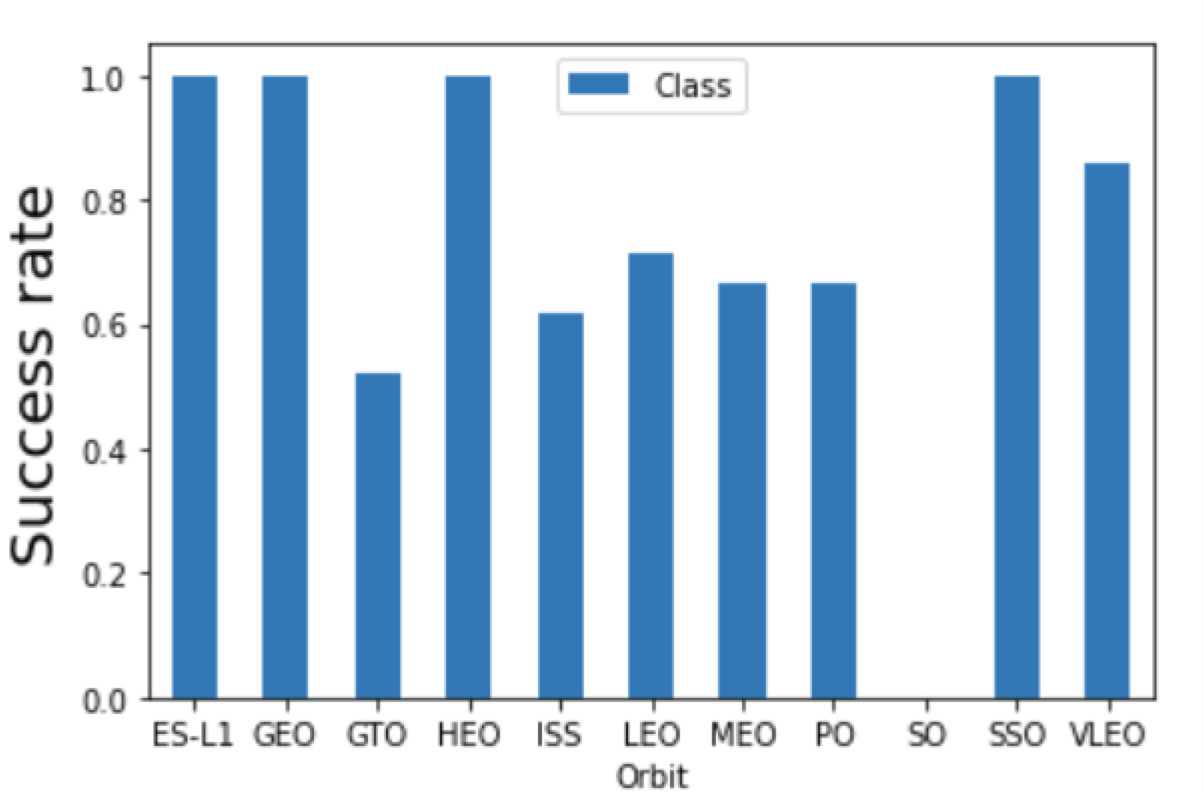



- For some reason, there are only two failed launches for payload mass above 8000 kg. Maybe the higher the payload mass, the higher the success rate.
- The launches with large payload mass at Launch Site VAFB SLC 4E are all successful.

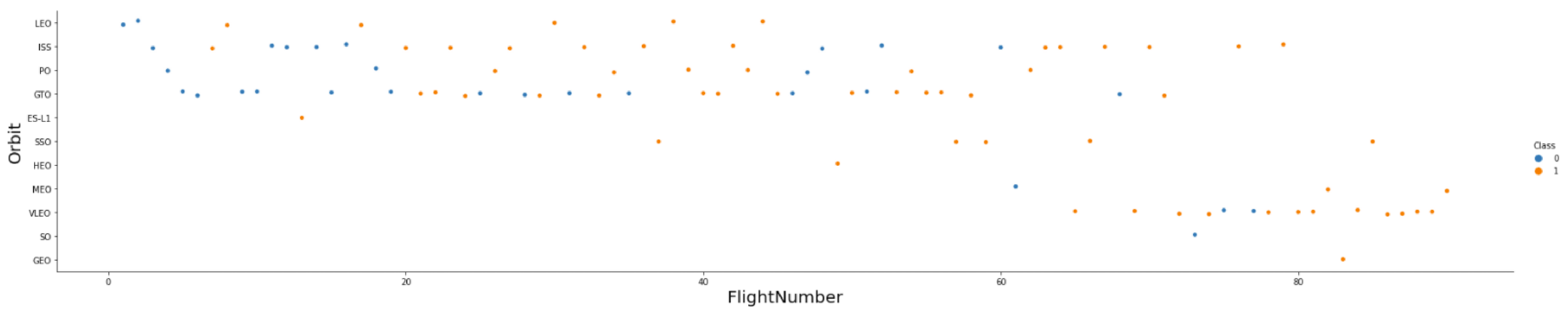
# Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type

- Show the screenshot of the scatter plot with explanations



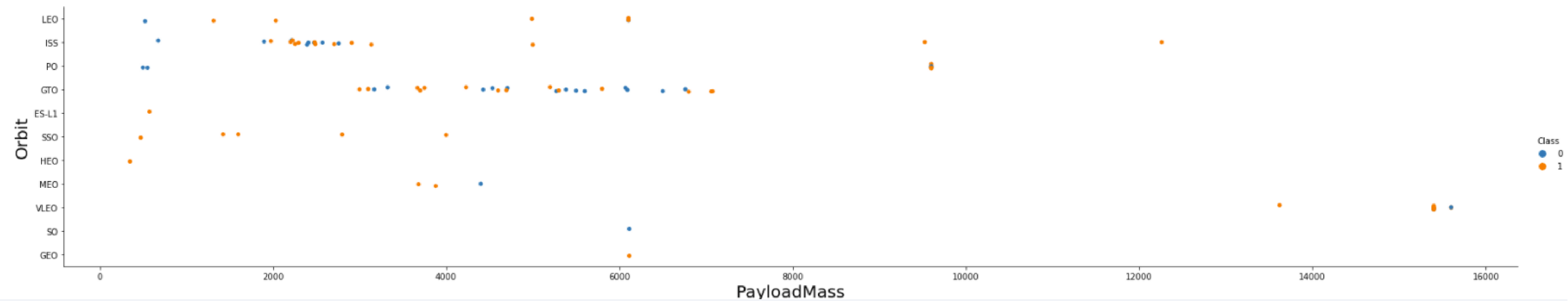
# Flight Number vs. Orbit Type




- Seems like most of the launches are for low orbit type.

- The success rate for the high orbit type is much higher.

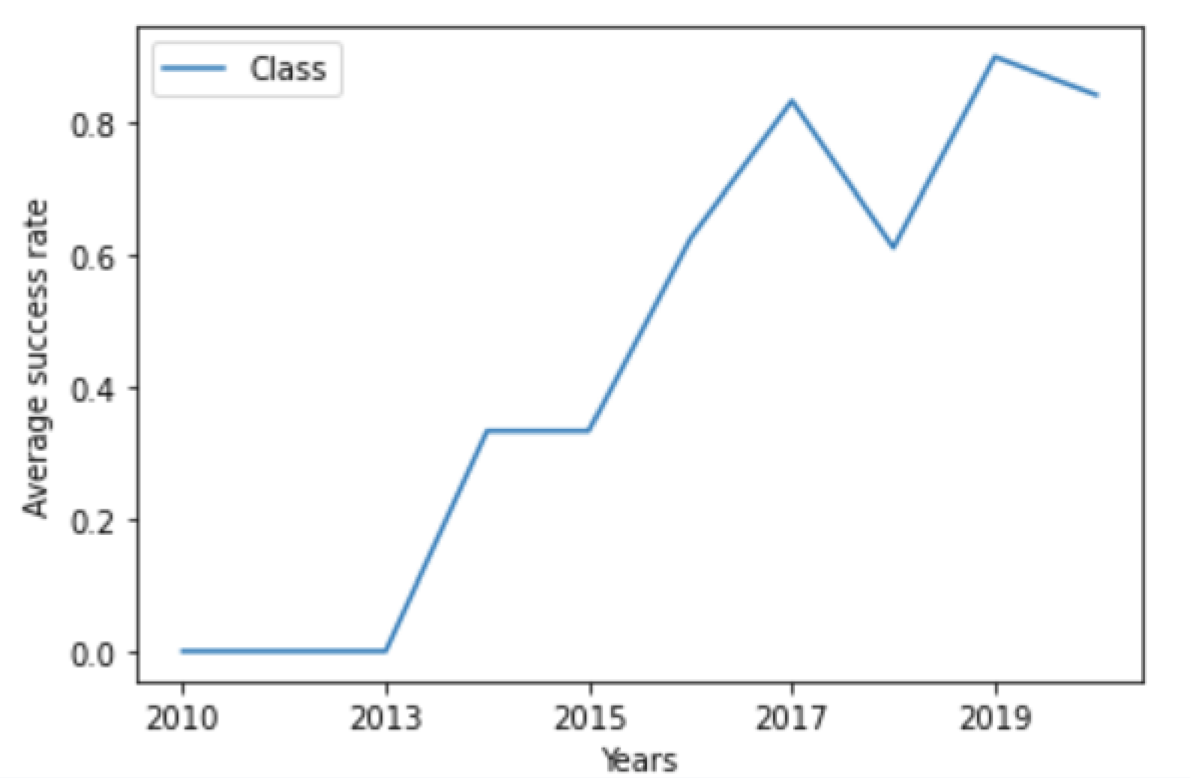
# Payload vs. Orbit Type




- The plot shows that the launches for the high orbit types usually have large payload mass.

# Launch Success Yearly Trend

- It is shown in the figure that from 2013 onwards the successful launch rate kept increasing. From 2019, the success rate starts to plateau at around 0.8 to 0.9.



# All Launch Site Names

- DISTINCT is used to show unique launch site names.

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

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

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

# Launch Site Names Begin with 'CCA'

- “LIKE” is used in the condition of the query to display records where launch sites begin with `CCA`

```
# if percentage sign is in the front, it means ends with CCA'
%sql SELECT * FROM SPACEXTBL where Launch_Site like 'CCA%' LIMIT 5;
```

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

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer | Mission_Outcome | Landing _Outcome |
|---|---|---|---|---|---|---|---|---|---|
| 04-06-2010 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success | Failure (parachute) |
| 08-12-2010 | 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) |
| 22-05-2012 | 07:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | No attempt |
| 08-10-2012 | 00:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) | Success | No attempt |
| 01-03-2013 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 | LEO (ISS) | NASA (CRS) | Success | No attempt |

# Total Payload Mass

- Sum() function is used and “Customer” column is filtered to calculate the total payload carried by boosters from NASA

```
%sql SELECT sum(PAYLOAD_MASS__KG_) FROM SPACEXTBL where Customer = 'NASA (CRS)';
```

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

| sum(PAYLOAD_MASS__KG_) |
|---|
| 45596 |

# Average Payload Mass by F9 v1.1

- Similar to the approach in the previous slide, the query below is used to calculate the average payload mass carried by booster version F9 v1.1

```
%sql SELECT avg(PAYLOAD_MASS__KG_) FROM SPACEXTBL where Booster_Version = 'F9 v1.1';
```

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

| avg(PAYLOAD_MASS__KG_) |
|---|
| 2928.4 |

# First Successful Ground Landing Date

- The first successful landing on a ground pad according to this table is on 1st May, 2017

```
# there is a space in the column name and you have to use square bracket
%sql SELECT min(Date) FROM SPACEXTBL where [Landing _Outcome] like 'Success (ground pad)';
```

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

| min(Date) |
|---|
| 01-05-2017 |

# Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT Booster_Version FROM SPACEXTBL where [Landing _Outcome] like 'Success (ground pad)' AND PAYLOAD_MASS__KG_ > 4000 AND P
```

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

| Booster_Version |
| --- |
| F9 FT B1032.1 |
| F9 B4 B1040.1 |
| F9 B4 B1043.1 |

- AND is used in the WHERE clause to filter the successful landing with payload mass between 4000 and 6000

# Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here

```
%sql SELECT count(Mission_Outcome) FROM SPACEXTBL where Mission_Outcome like 'Success%';
```

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

| count(Mission_Outcome) |
|---|
| 100 |

```
%sql SELECT count(Mission_Outcome) FROM SPACEXTBL where Mission_Outcome like 'Failure%';
```

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

| count(Mission_Outcome) |
|---|
| 1 |

# Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

```
%sql SELECT Booster_Version, PAYLOAD_MASS__KG_ From SPACEXTBL where PAYLOAD_MASS__KG_ = (select MAX(PAYLOAD_MASS__KG_) from SPACEXTBL) order by Booster_Version;
```

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

| Booster_Version | PAYLOAD_MASS__KG_ |
|---|---|
| F9 B5 B1048.4 | 15600 |
| F9 B5 B1048.5 | 15600 |
| F9 B5 B1049.4 | 15600 |
| F9 B5 B1049.5 | 15600 |
| F9 B5 B1049.7 | 15600 |
| F9 B5 B1051.3 | 15600 |
| F9 B5 B1051.4 | 15600 |
| F9 B5 B1051.6 | 15600 |
| F9 B5 B1056.4 | 15600 |
| F9 B5 B1058.3 | 15600 |
| F9 B5 B1060.2 | 15600 |
| F9 B5 B1060.3 | 15600 |

# 2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

- Present your query result with a short explanation here

```
%sql SELECT  substr(Date, 4, 2), [Landing _Outcome], Booster_Version, Launch_Site FROM SPACEXTBL where substr(Date,7,4)='2015' AND [Landing _Outcome] = 'Failure (drone ship)';
```

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

| substr(Date, 4, 2) | 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 |

# 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

```
%sql SELECT  [Landing _Outcome], count([Landing _Outcome]) FROM SPACEXTBL where (substr(Date, 1, 2)||'-'|| substr(Date, 4, 2)||'-'||substr(Date, 7, 4)) between '04-06-2010' and '20-03-2017' Group BY [Landing _Outcome] Order by count([Landing _Outcome]) desc;
```

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

| Landing _Outcome | count([Landing _Outcome]) |
|---|---|
| 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

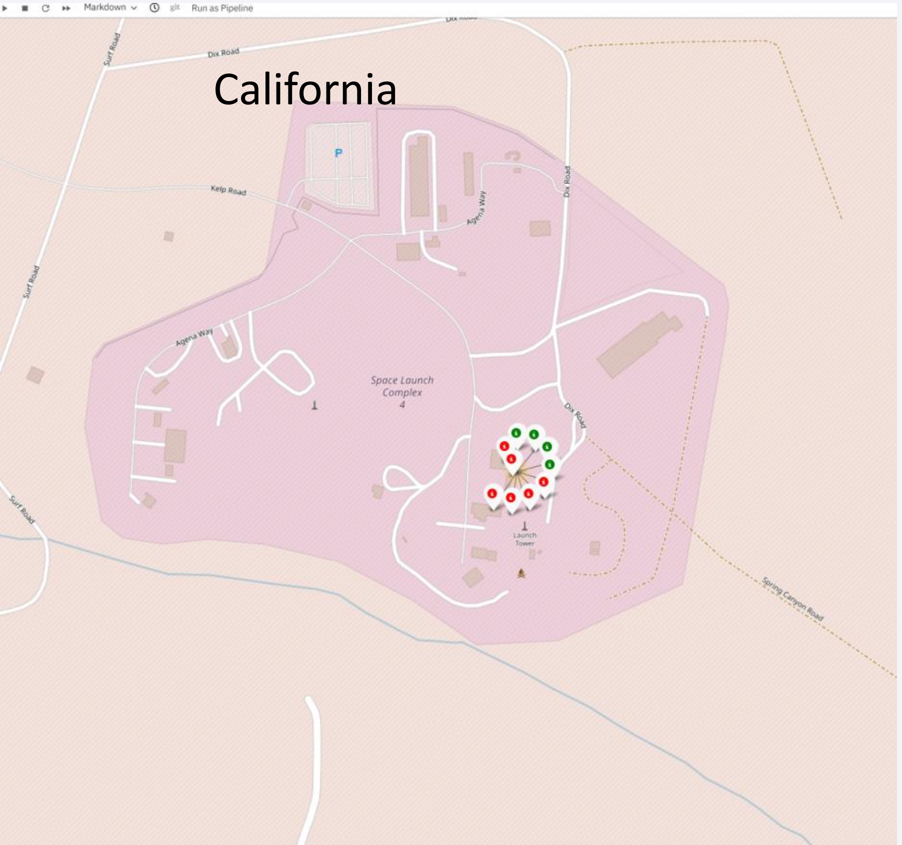
# Launch Sites Proximities Analysis

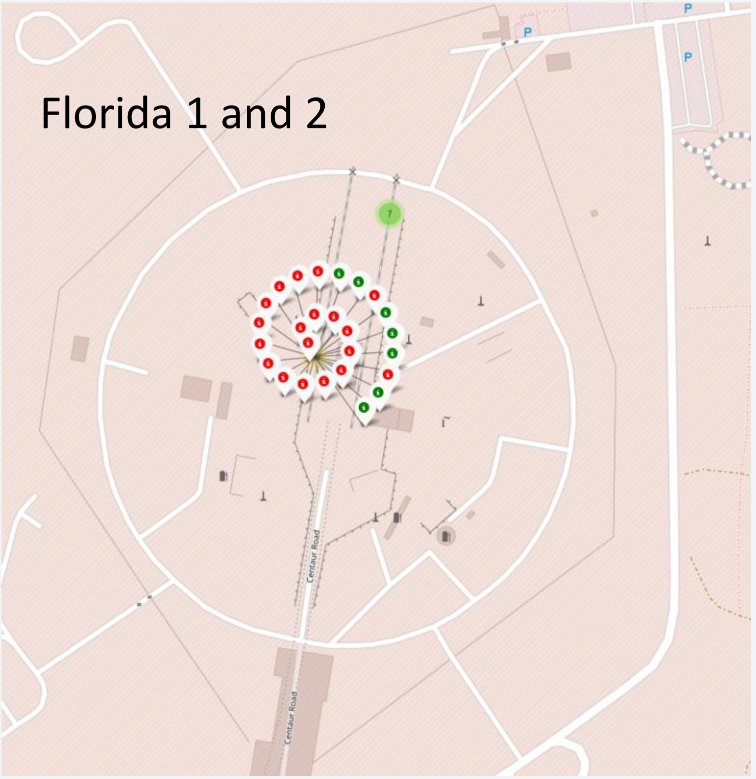
# Mark all launch sites on a map

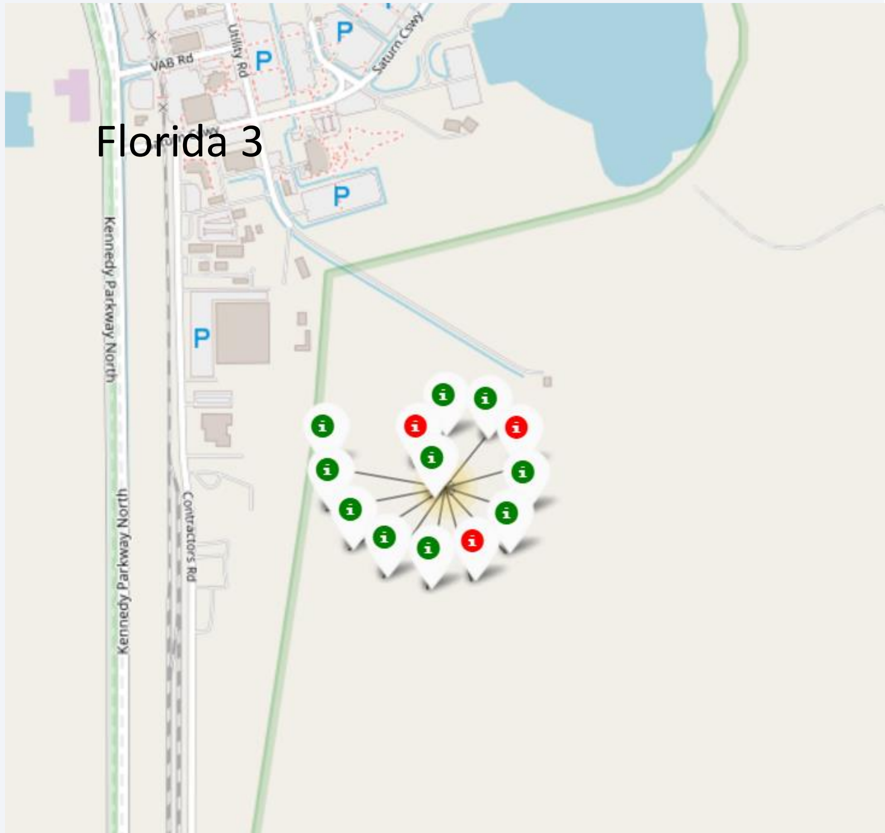
- The markers on the map show that all launch sites are near US coasts in California and Florida.
- They are all close to the equator.

# Mark the launch outcomes for each site on the map

- A green marker denotes a successful launch event, and the red stands for failed ones.

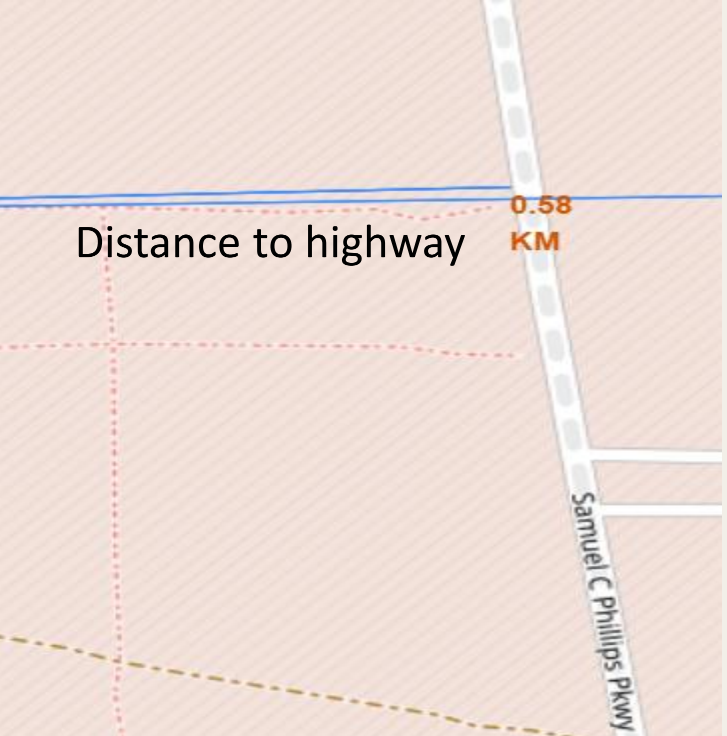


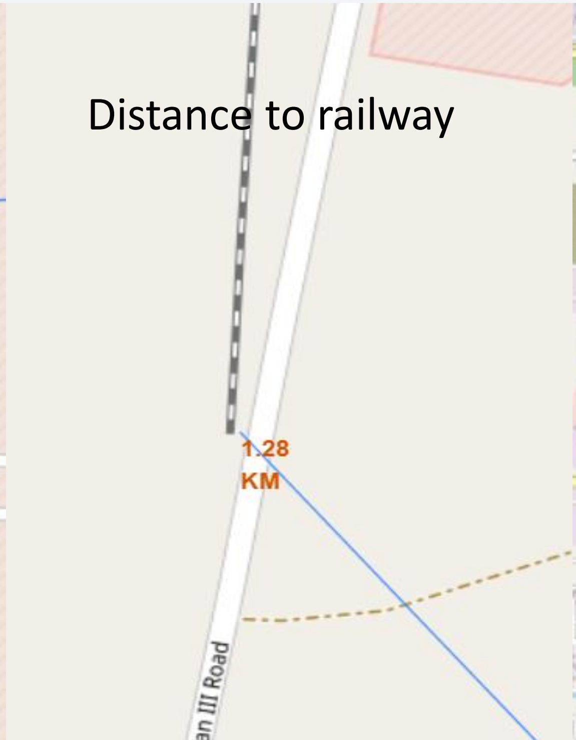



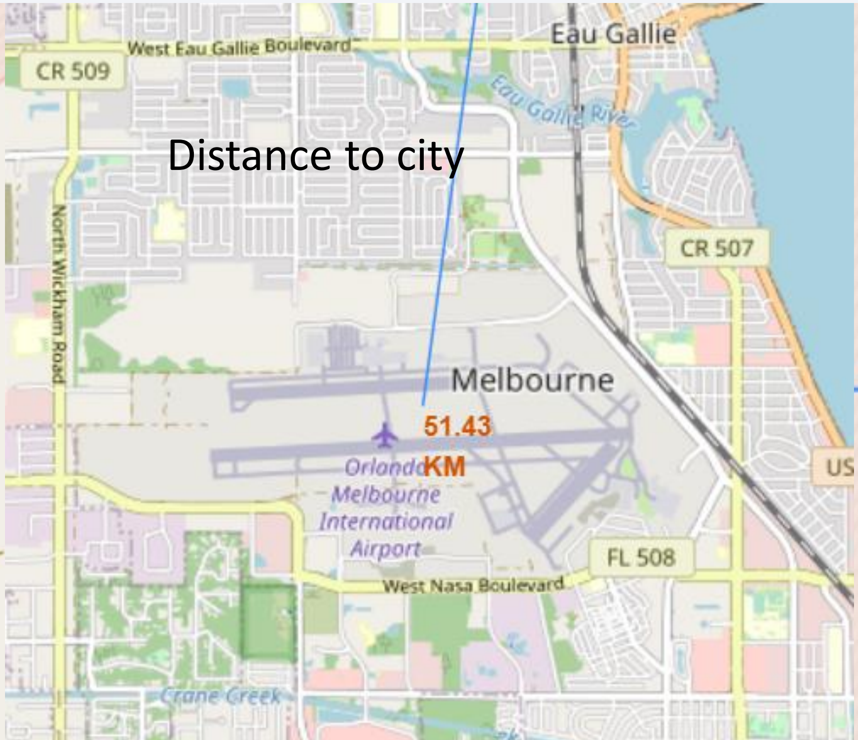



# Proximities of a launch site

- Are launch sites in close proximity to railways? Yes
- Are launch sites in close proximity to highways? Yes
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? No



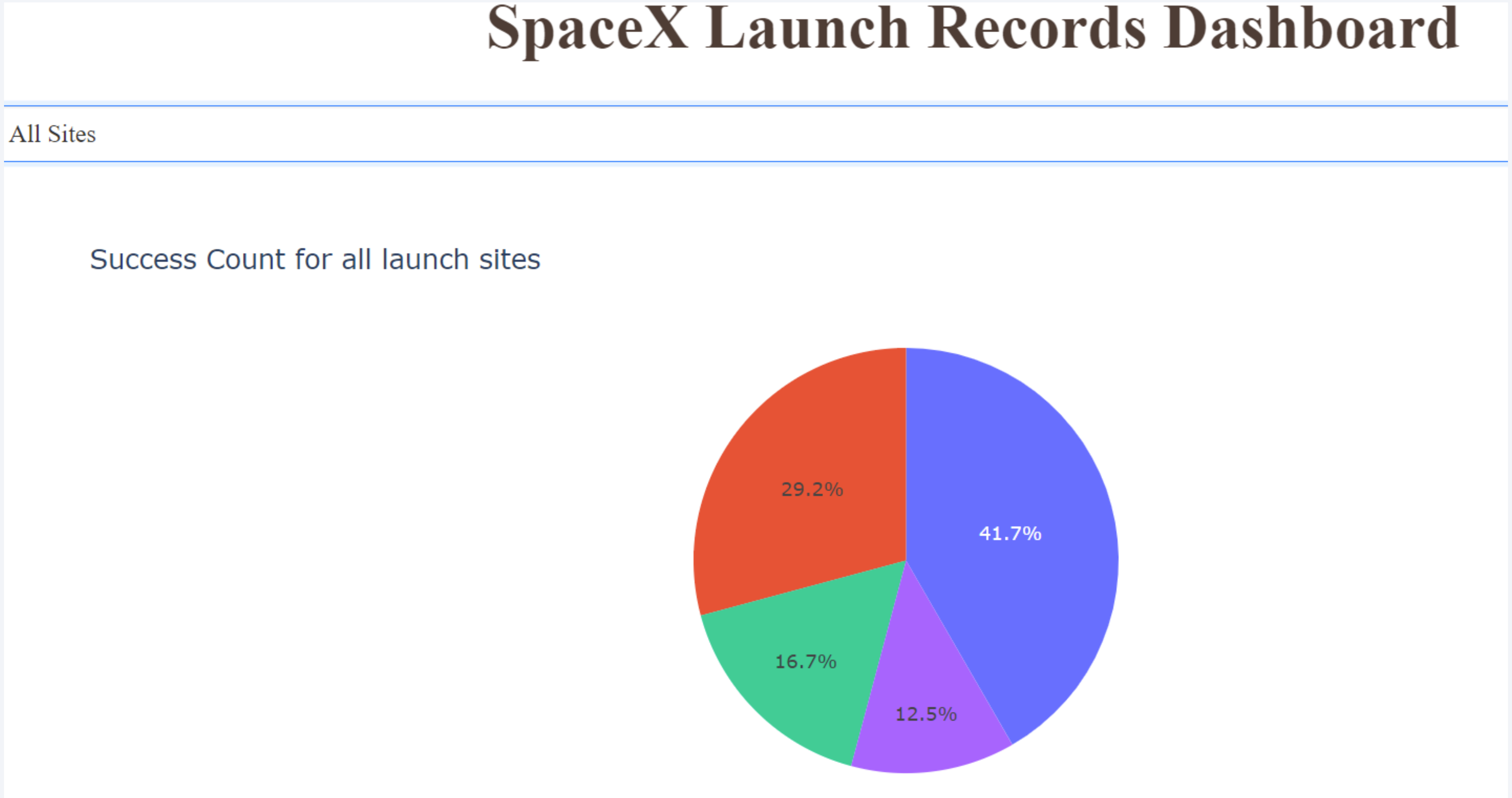




Section 4

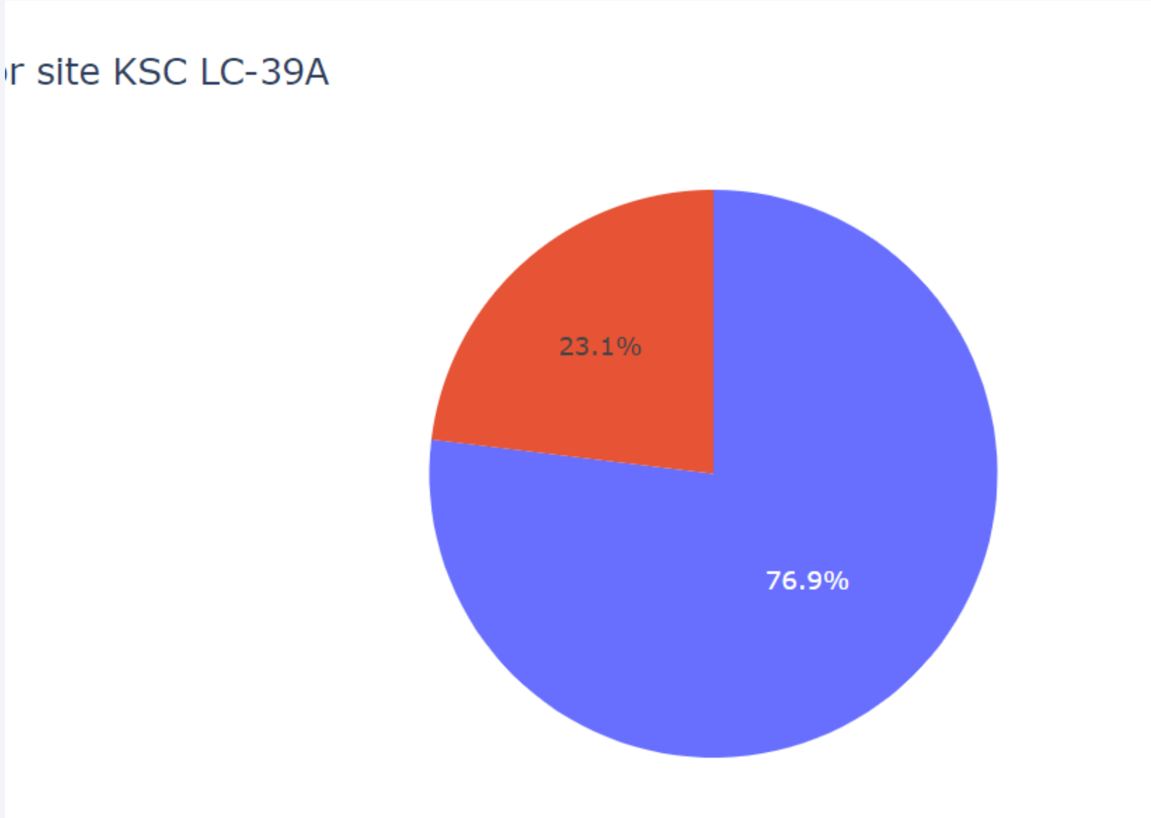
# Build a Dashboard with Plotly Dash

# Pie chart of success count for all launch sites



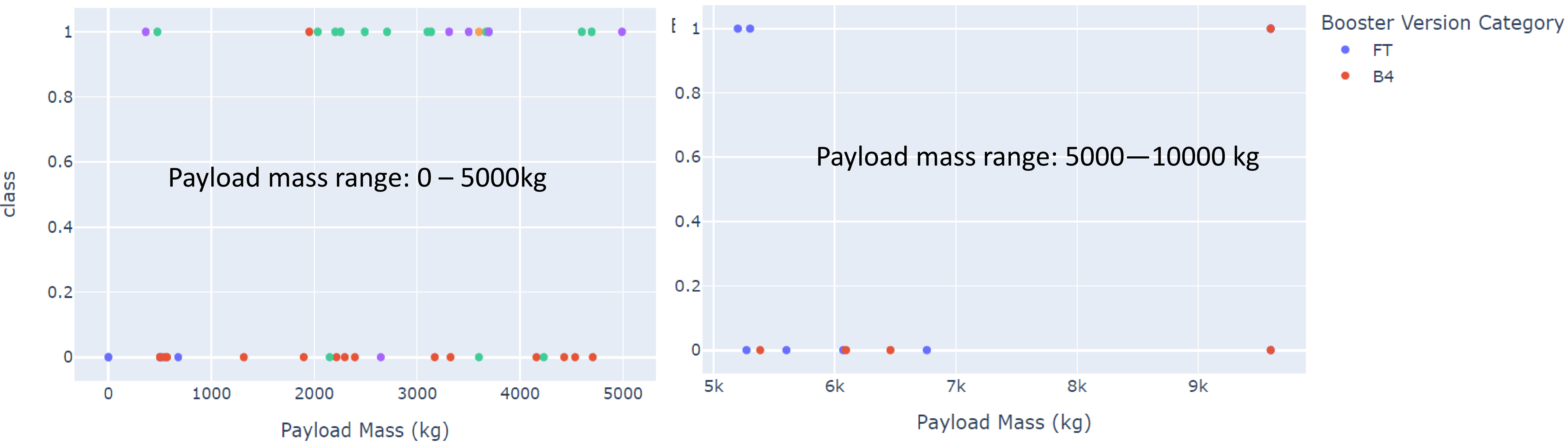

- Launch Site KSC LC-39A has the most successful launches among the four sites.

# Pie chart for the launch site with the highest launch success ratio




- Launch Site KSC LC-39A is the most successful launch site, but it still has 23.1% failure rate.

# Payload vs. Launch Outcome scatter plot for all sites



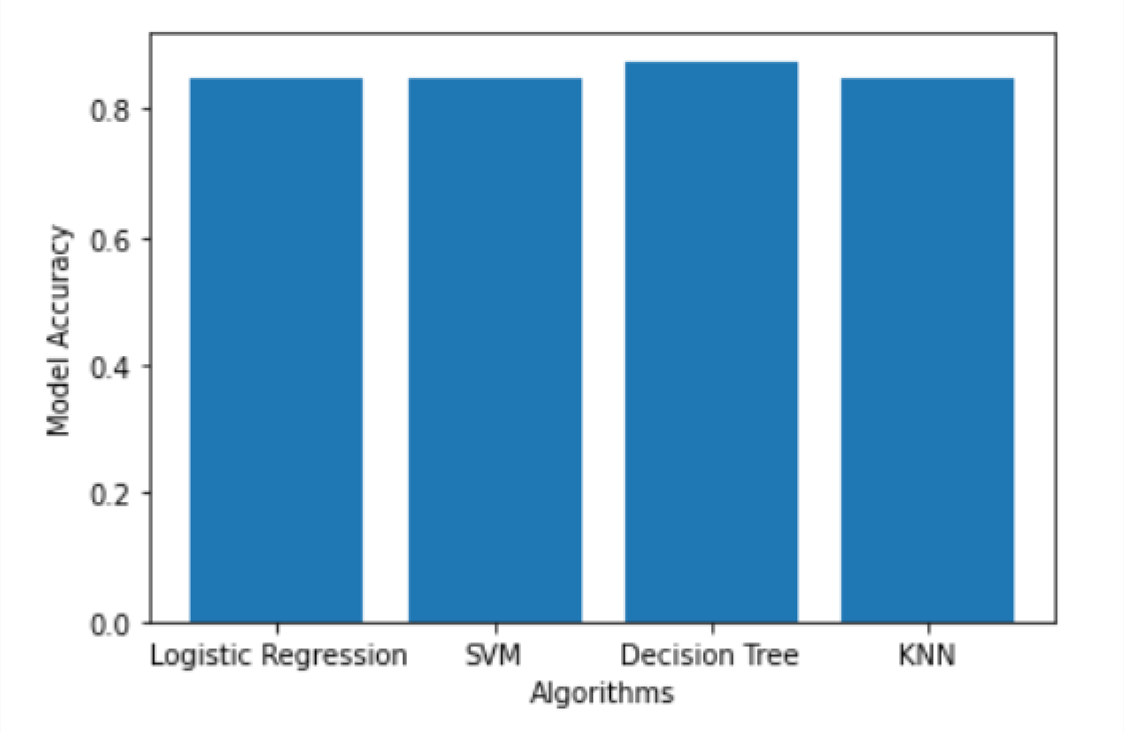

- The success rate for low payload launches is higher.

Section 5

# Predictive Analysis (Classification)

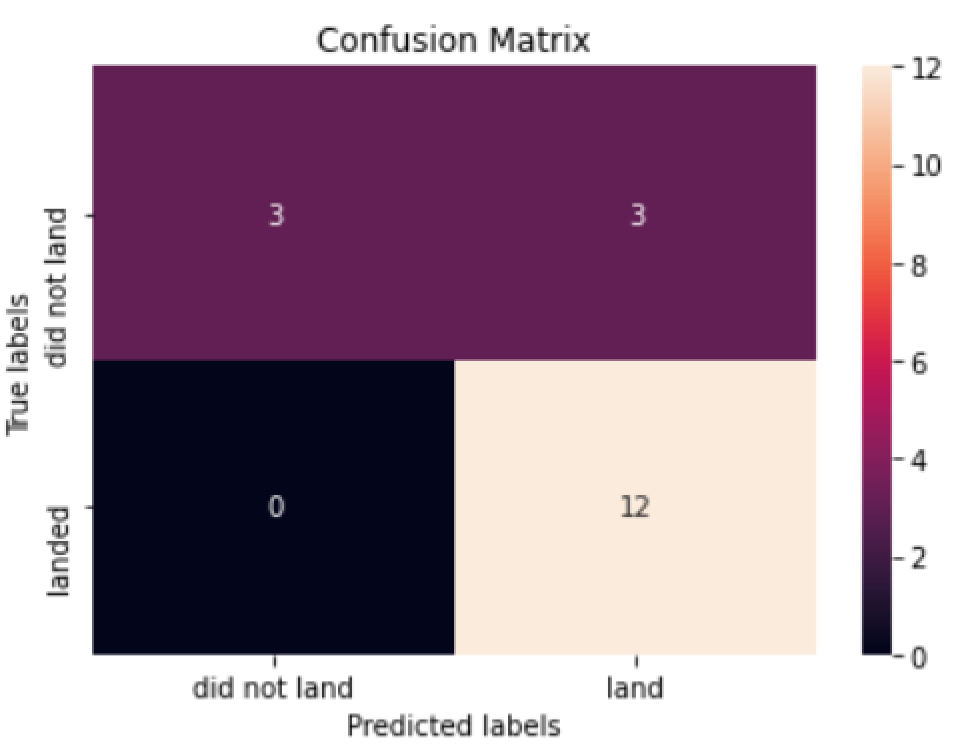
# Classification Accuracy

- From the bar chart on the right, Decision Tree has the slightly higher accuracy than others.



# Confusion Matrix

- The confusion matrix of the Decision Tree model does a good job on predicting the launch results.
- But there are 3 false-positive predictions, which are failed launches predicted as successful launches.



# Conclusions

- The successful launch rate has increased significantly since 2013.
- The success rate for low payload launches is higher.
- KSC LC-39A is the most successful launch site.
- The success rate for the high orbit type is much higher.
- The Decision tree classifier works the best for predicting SpaceX launch outcomes.

# Thank you!

# Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project