IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Daniel Lopez
December 21, 2024

# Outline

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

# Executive Summary

- Summary of methodologies

The question we want to answer was "Instead of using rocket science, could we predict if a SpaceX Falcon 9 Booster would land successfully using a data science approach?" The answer is, yes. To achieve this end we decided to take a machine learning classification approach, were we would feed our model several features and our model would then classify the mission as "1" a success or "0" not a success. A John Rollins approach of creating a feedback loop of exploring the data and transforming our feature set was then carried out to choose our sample data. We then deployed several ML models on the sample data and evaluated each model to achieve our final optimum result.


- Summary of all results, we transformed features such as 'Launch Site', 'Booster Version', and 'Payload Mass' into 83 numerical features, using 90 data records, and a ML Decision Tree Classification Model, we were able to predict a successful landing on average 83.3% of the time with training accuracy as high as 89%.

# Introduction

SpaceX became the first company to successfully land a rocket upright on 12/21/2015. Rocket reusability has now become a staple of the SpaceX brand. Given that rockets weigh on the order of 500,000 KGs to take payloads into space on the order of 10,000 kg, we can see that there is a tremendous amount of room for profit if the rocket is reusable. SpaceX launches cost roughly $67M while there competitors can range anywhere from $100M to $350M per launch. So, therefore a critical business question would be is it possible to predict a successful launch outcome?

In this Data Science report we seek to answer this question, given a set of features can we predict a successful landing for reusability? Additionally, can we do this with publicly available data? Can we use machine learning to accomplish this?

Note: In this report we will use launch success and landing success interchangeably to both mean a successful booster landing, since we are interested in Falcon 9 Launch & Lands. When referring to a successful payload launch into space, we will use the term successful mission outcome.

Section 1

# Methodology

# Methodology

## Executive Summary

- Data collection methodology:
  - ML Supervised Learning Classification Approach
- Perform data wrangling
  - Data was structured and cleaned using pandas dataframe manipulations
  - Data was further augmented with ML techniques in preprocessing
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Built, tuned, and evaluated several ML classification models

# Data Collection

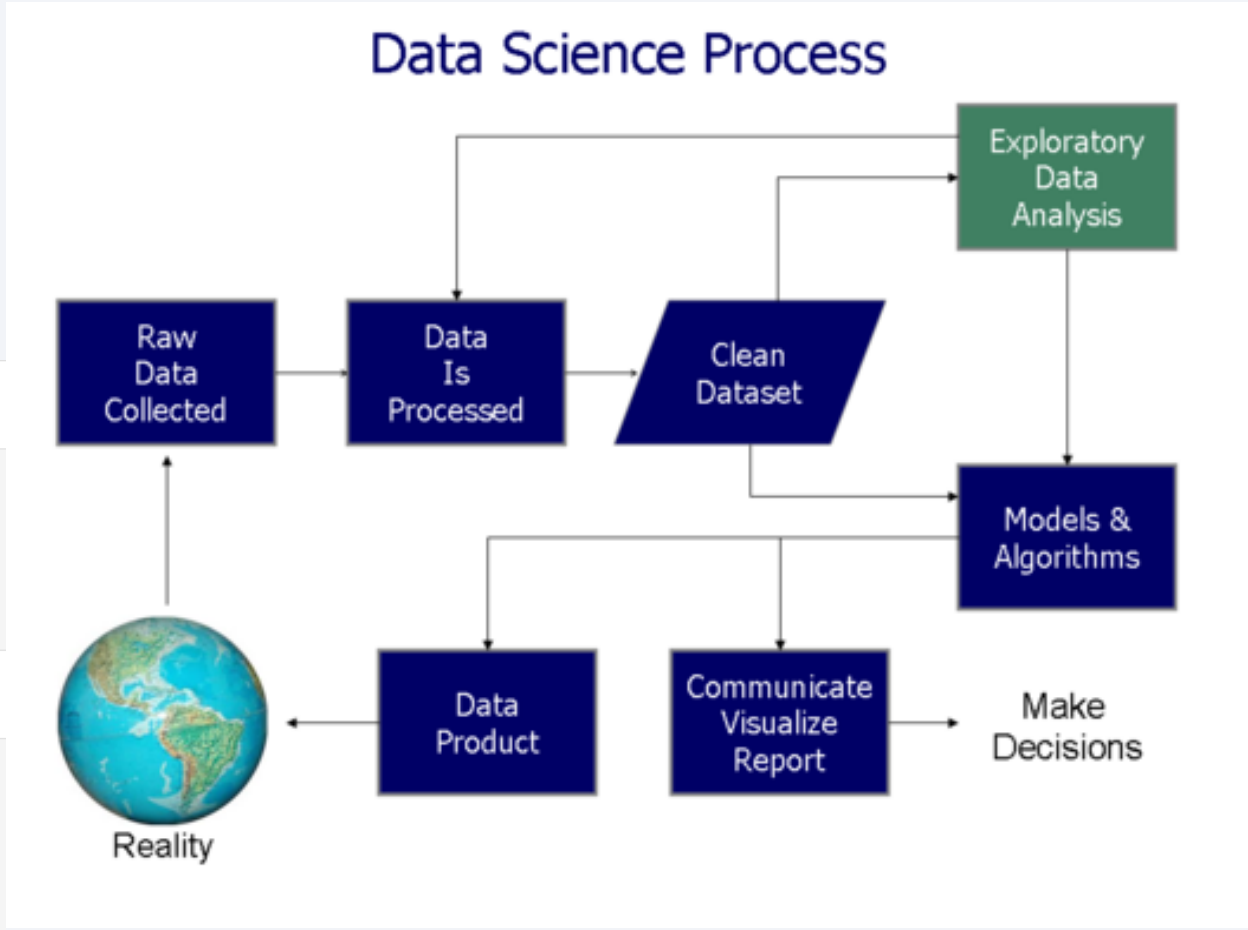
- Datasets were web scraped from public SpaceX data sources. Data was then collected into workable format using a pandas data frame. We then used ML techniques to further preprocess the data for modeling. A collection example is shown below.

Simple 1 data grab

```
[16]: # Takes the dataset and uses the rocket column to call the API and append the data to the list
def getBoosterVersion(data):
    for x in data['rocket']:
       if x:
        response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
        BoosterVersion.append(response['name'])
```

Multiple data grab

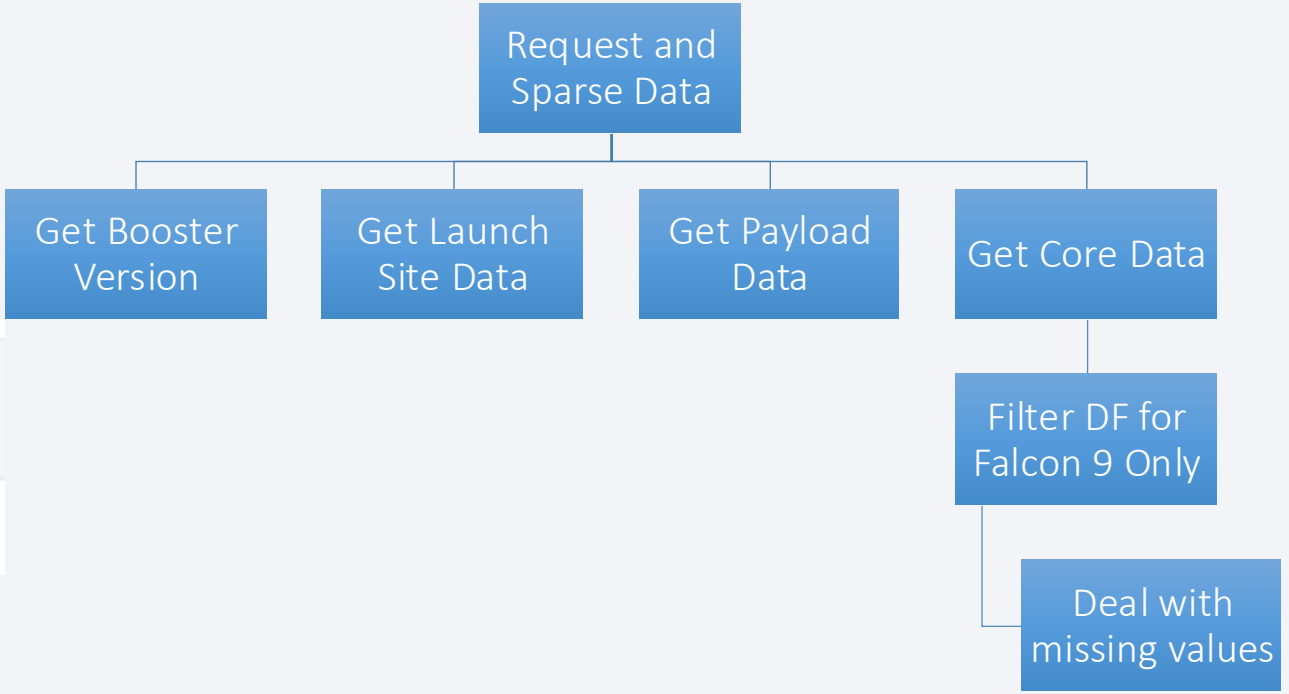
```
[18]: # Takes the dataset and uses the launchpad column to call the API and append the data to the list
def getLaunchSite(data):
    for x in data['launchpad']:
       if x:
         response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).json()
         Longitude.append(response['longitude'])
         Latitude.append(response['latitude'])
         LaunchSite.append(response['name'])
```



# Data Collection – SpaceX API

- Data collection was done with SpaceX REST calls. Utilizing Get Request then normalizing the json objects collected and converting them into a Pandas raw dataframe. Furthermore functions utilizing API REST calls were used to append useful data into lists, that were transformed into a dictionary, which was then converted into our dataframe ready for preprocessing.
- Once collected we could further refine our data, filtering, and cleaning up missing values. An example of a calculation for filling missing payload values with the mean is shown below.

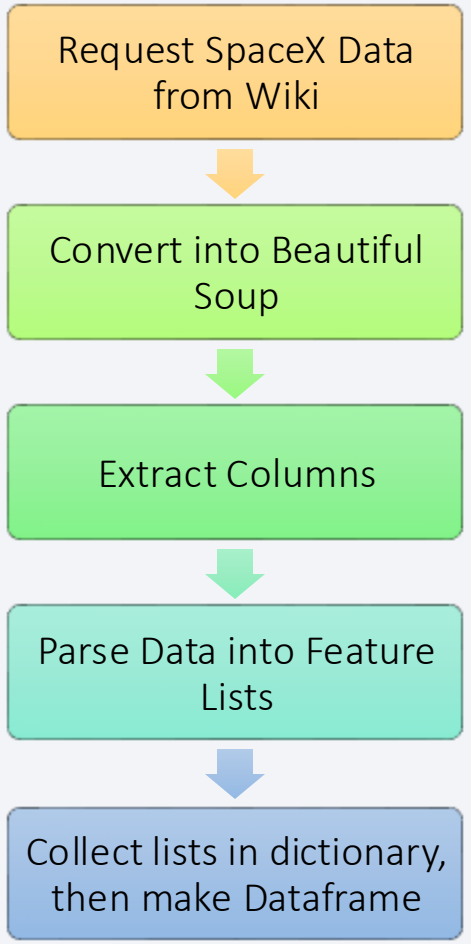
```
[28]: # Calculate the mean value of PayloadMass column
      Payload_mean = data_falcon9['PayloadMass'].astype('float').mean(axis=0)
      Payload_mean

[28]: 6123.547647058824
```




- GitHub: SpaceX Data Collection
- https://github.com/DLPhysics/Data-Science/blob/main/PyLab-M1L1_SpaceX-Data-Collection.ipynb

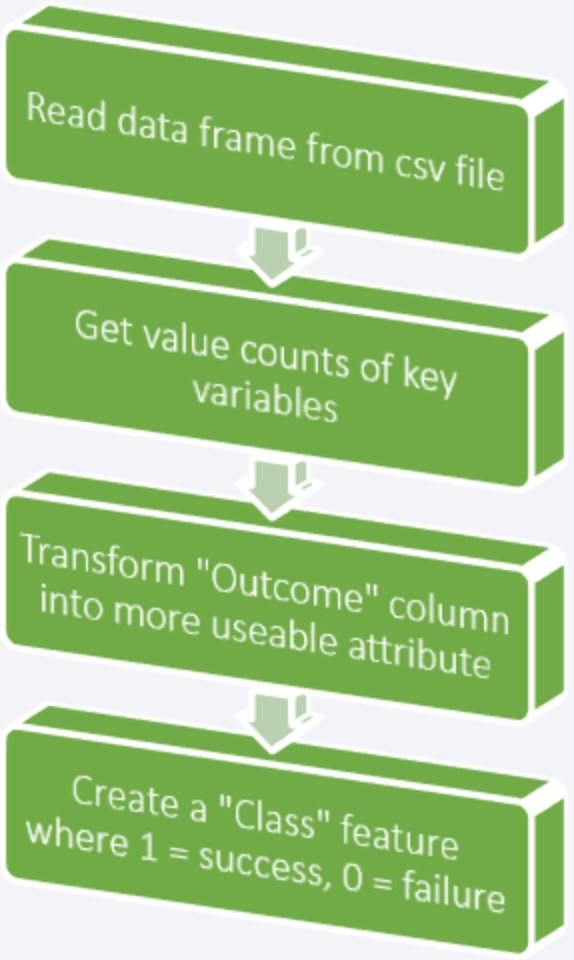
# Data Collection - Scraping

- Data was scraped from the publicly available Falcon launch wiki page (specifically the 3rd table). Functions were created to assist in extracting table row and cell information. Following the flowchart presented here, a workable starting data frame was created.
- GitHub URL: SpaceX Webscraping
- https://github.com/DLPhysics/Data-Science/blob/main/PyLab-M1L2_SpaceX-Webscraping.ipynb



# Data Wrangling

- Once our csv file was converted to a pandas data frame we could easily manipulate our data for statistical purposes.
- We would like successful landing data, so to that end its critical that we include all key features such as launch site and orbit.
- Finally, we created the "Class" feature to serve the fastest way of verifying that a launch had a successful landing
- GitHub: SpaceX DataWrangling
- https://github.com/DLPhysics/Data-Science/blob/main/PyLab-M1L3_SpaceX-Data-Wrangling.ipynb



# EDA with Data Visualization

- Data Visualization is a powerful tool for quickly gaining insight into variable relationships. The target of the report is to predict launch outcomes, so one way to rapidly get an idea of feature influence is to use scatter plots against launch outcome. Another tool used was bar charts for median success percentage for launch orbits. The yearly trend for launch success, also illustrates what the near future predictions should reflect.
- GitHub URL: SpaceX Data Visualization
- https://github.com/DLPhysics/Data-Science/blob/main/PyLab-M2L2_SpaceX-EDA_and_DataVis.ipynb

# EDA with SQL

- Using the SQL queries we inquired the dataset to understand landing outcomes, the effect of payload, date of launch, orbit, and booster. A non-exhaustive list of queries is as follows:
  - Names of unique launch sites
  - 5 Records were site begins with string "..."
  - Total payload launched for NASA
  - Average payload mass carried by Booster v1.1
  - Date of 1st successful landing
  - Names of Booster that successfully landed on a dropship, given a payload range
  - Total number of Mission Success and Failures
  - List of all Boosters that have carried the maximum payload
  - Failures on drone ship in 2015
  - Rank of landing outcomes, with multiple conditions
- GitHub URL: SpaceX SQL
- https://github.com/DLPhysics/Data-Science/blob/main/PyLab-M2L1_SpaceX-SQL_EDA.ipynb

# Build an Interactive Map with Folium

- The items added to the interactive map include Marker Icons with all the Launch Sites names, Circles to assist in locating launch site area. Marker clusters for further insights into each launch site. These clusters reveal color coding launch success/failures at each site as well as a popup with the Falcon 9 flight number. Finally, we include an image of a polyline drawn to the coastline with the distance calculation.
- These features all help quickly locate sites, evaluate launch success at each site, and aid in site planning. Marking distances to landmarks such as coastlines, major cities, railways, highways can all aid in planning launch/land abort sequences and aid in improving development and production at the site.
- GitHub URL: SpaceX Folium Map
- https://github.com/DLPhysics/Data-Science/blob/main/PyLab-M3L1_SpaceX-Folium-Mapping.ipynb

# Build a Dashboard with Plotly Dash

- Using Python Plotly and Dash API we built an interactive Dashboard to quickly look through current launch success data. The 3 visuals we will be presenting that showcase are as follows:
    1) A pie chart of All Launch Sites to show the relative percentages of successful launches by site.
    2) A pie chart of a specific site to show the ratio of successful/unsuccessful launches.
    3) A scatter plot of payload (selected by payload range) against launch success.
- These were chosen because of their importance in reviewing launch success data and planning for future successful missions based on payload.
- GitHub URL SpaceX Plotly Dashboard
- https://github.com/DLPhysics/Data-Science/blob/main/PyLab-M3L2_SpaceX-Dashboarding.ipynb

# Predictive Analysis (Classification)


- We used a supervised learning ML classification approach to determining if a future launch will have a successful landing. To achieve this we prepped the data, deployed several models, and tested over several parameters to achieve a model with a prediction accuracy of average of 83% and as high as 89%. This was achieved with less than 100 records, and should only get better with more data.
- To achieve this features that may influence landing were chosen such as flight number, reused, orbit, payload mass, etc... Then all non-numerical features were hot-coded into numerical quantities, the data was then standardize using a standard scalar preprocess. The dataset was split 80/20 for training/testing purposes. The training data had a cross-fold validation of 10 for model testing. The models deployed were logistic regression, support vector machines, K-nearest neighbors, and decision tree models. GridSearchCV was then used with these models to test over several hyperparameters and choose the optimal parameters. The Model was then evaluated using a confusion matrix.
- GitHub URL to complete lab SpaceX Machine Learning Lab
- https://github.com/DLPhysics/Data-Science/blob/main/PyLab-M4L1_SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

# Results

- Exploratory data analysis results

  These results will be displayed using seaborn data visualization techniques.

  As well as with sqlite queries.

  Flight number appears one of the heaviest weighted factors in determining success.  As SpaceX gained experience launches became more likely to succeed.

- Interactive analytics demo in screenshots

  Using folium maps to explore launch success.

  Using python DASH API to create an interactive browser based web application to view landing success.

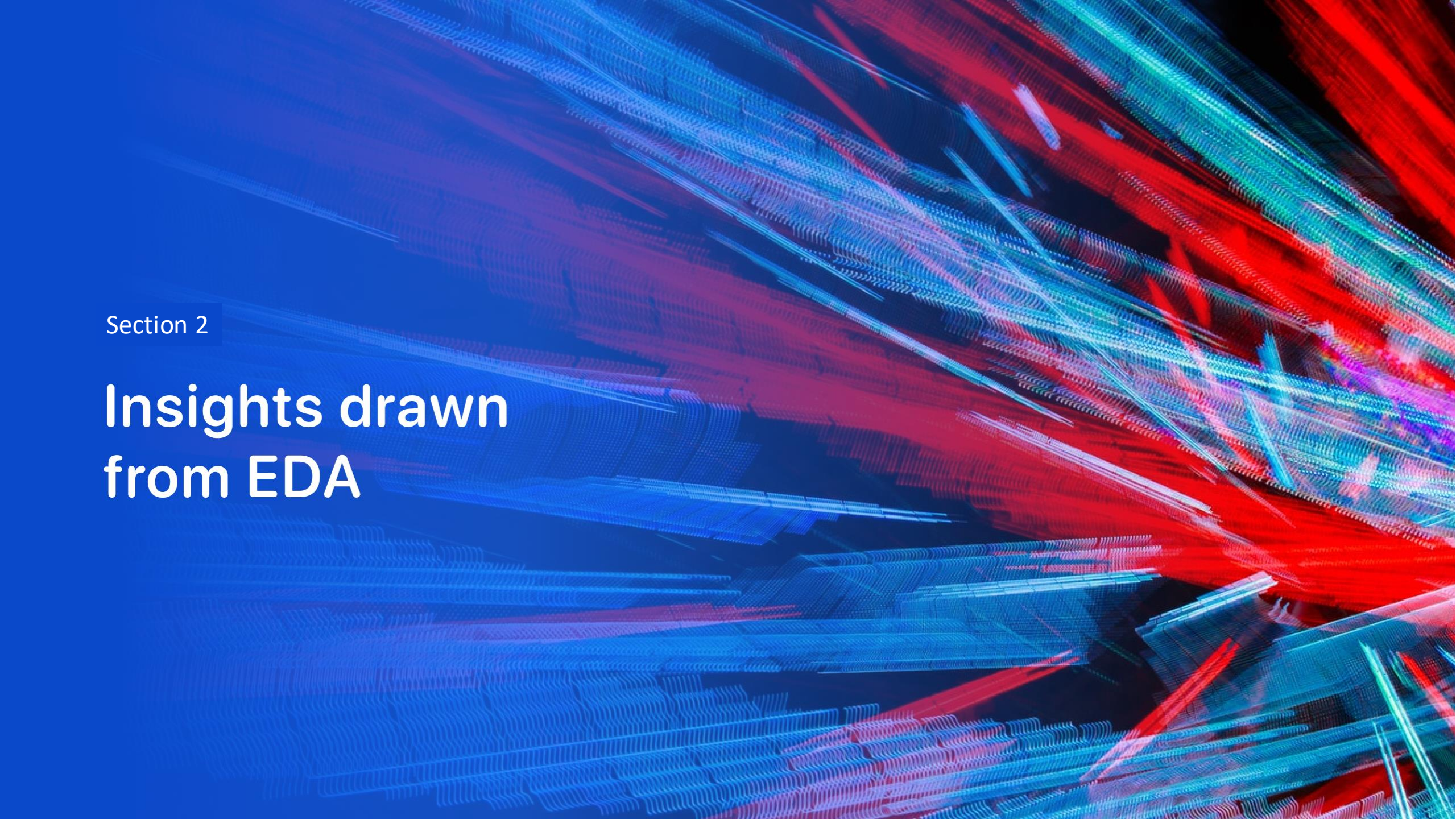
  KSC was the most successful at landing by count, but CCAFS SLC-40 was the most successful by ratio.

- Predictive analysis results

  What did are models learn from the data sets, and

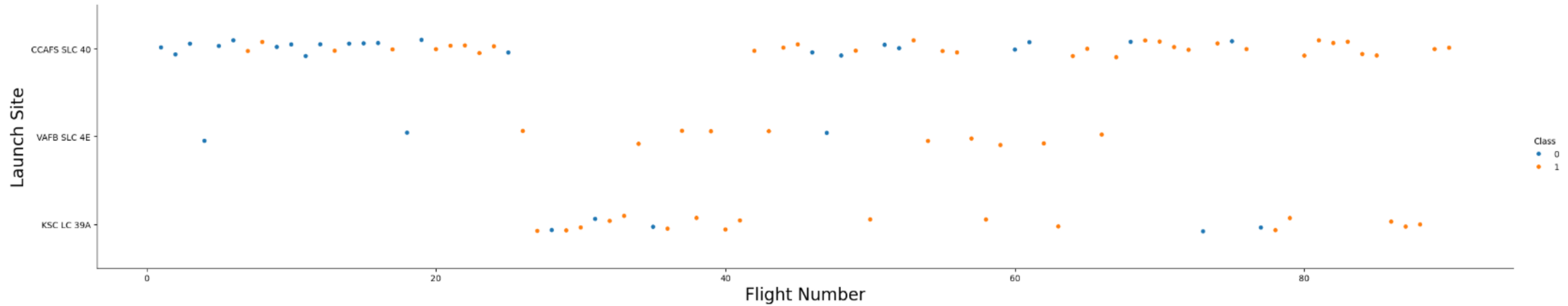
  What did we learn from our models

  Decision Tree was the most accurate. However, it was also the most volatile so maybe we should keep the other models close at hand.


Section 2
Insights drawn from EDA

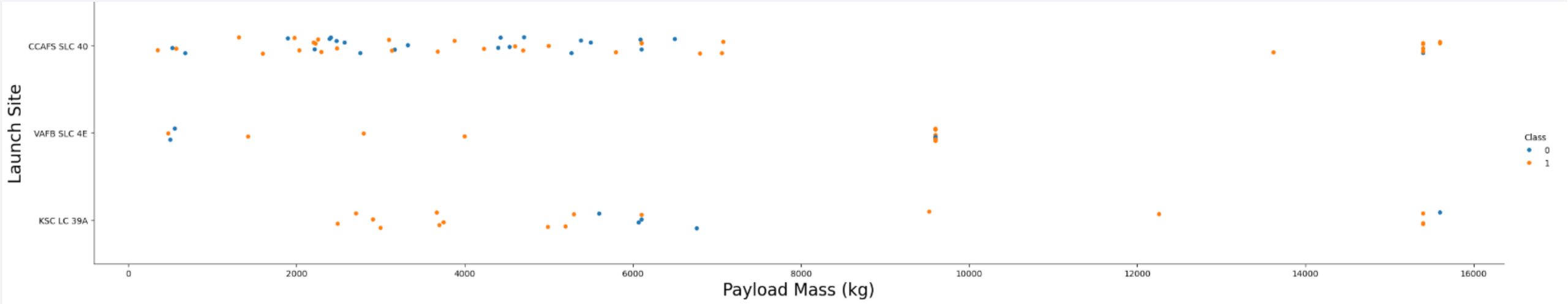
# Flight Number vs. Launch Site

- Plotting flight number against launch site we can see that flights began at CCAFS SLC-40, then the majority moved to KSC LC-39A briefly, before moving back to CCAFS SLC-40.
- The color coding (1=successful landing) also illustrates that as flight number increased, success rate at all sites tended to improve.



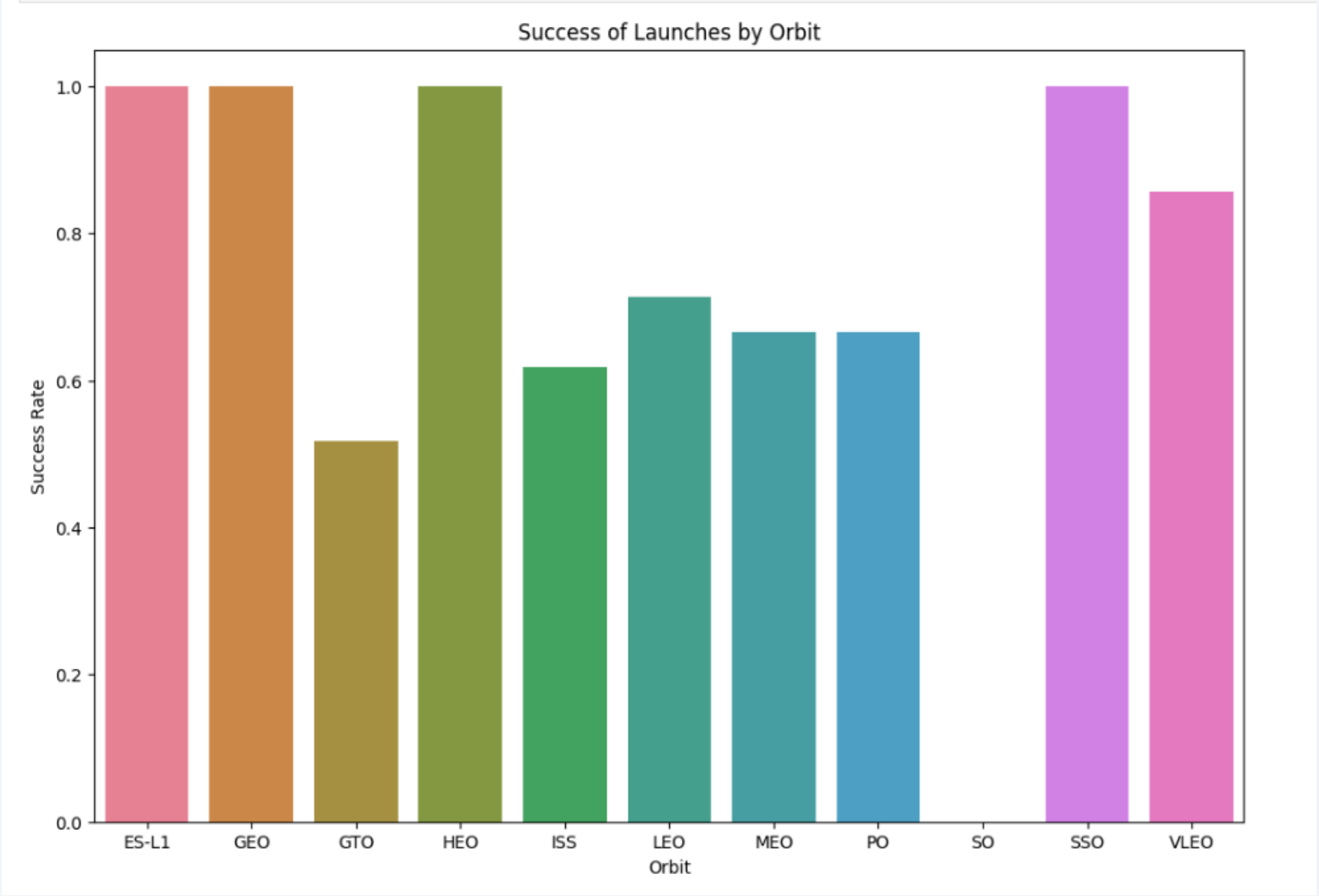
# Payload vs. Launch Site

- With this scatter plot of Payload vs. Launch Site, we can see CCAFS SLC-40 has the most experience launching the heaviest payloads and landing successfully.  Likewise VAFB SLC 4E has many successful landings after launching payloads of around ~9000 KG.
- KSC LC-39A has a 100% landing rate after launching payloads between 2000-5500 KG.



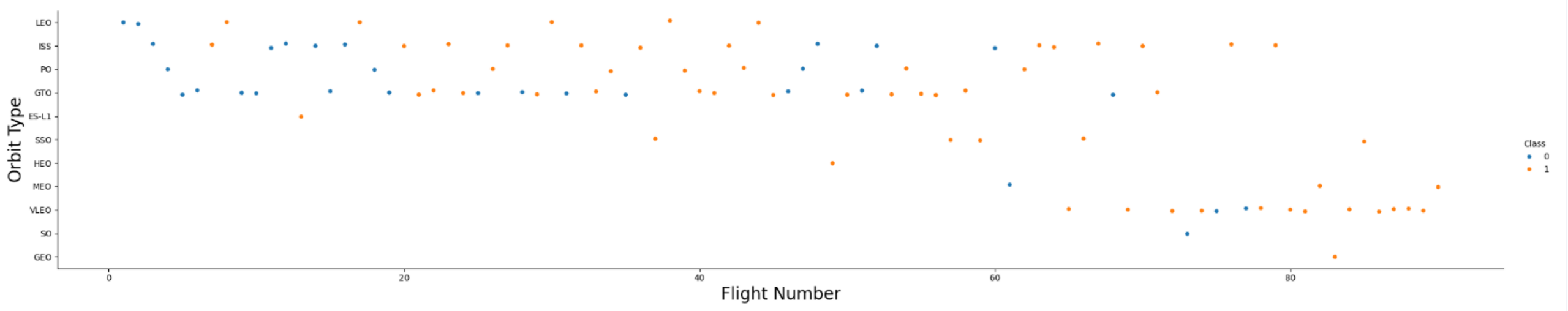
# Success Rate vs. Orbit Type

- Plotting Success Rate by Orbit we can intuit future mission success based on orbit.
- Many Orbits have a 100% success rate.
- SO and SSO Orbits are synonymous so we see are data points are all correctly classified in one group.



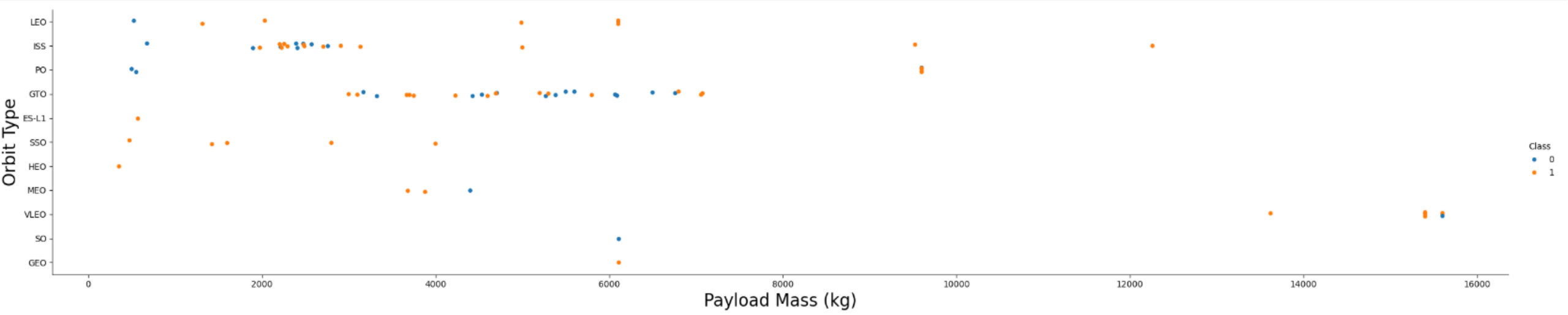
# Flight Number vs. Orbit Type

- Here we can see that as flight number increased the diversity of orbit types also increased.
- There is also an explosion of Very Low Earth Orbits around flight 75-90
- Generally, as flight number increases landing success improves for most orbits.
- Also, we can see some of the previously mentioned 100% success orbits.



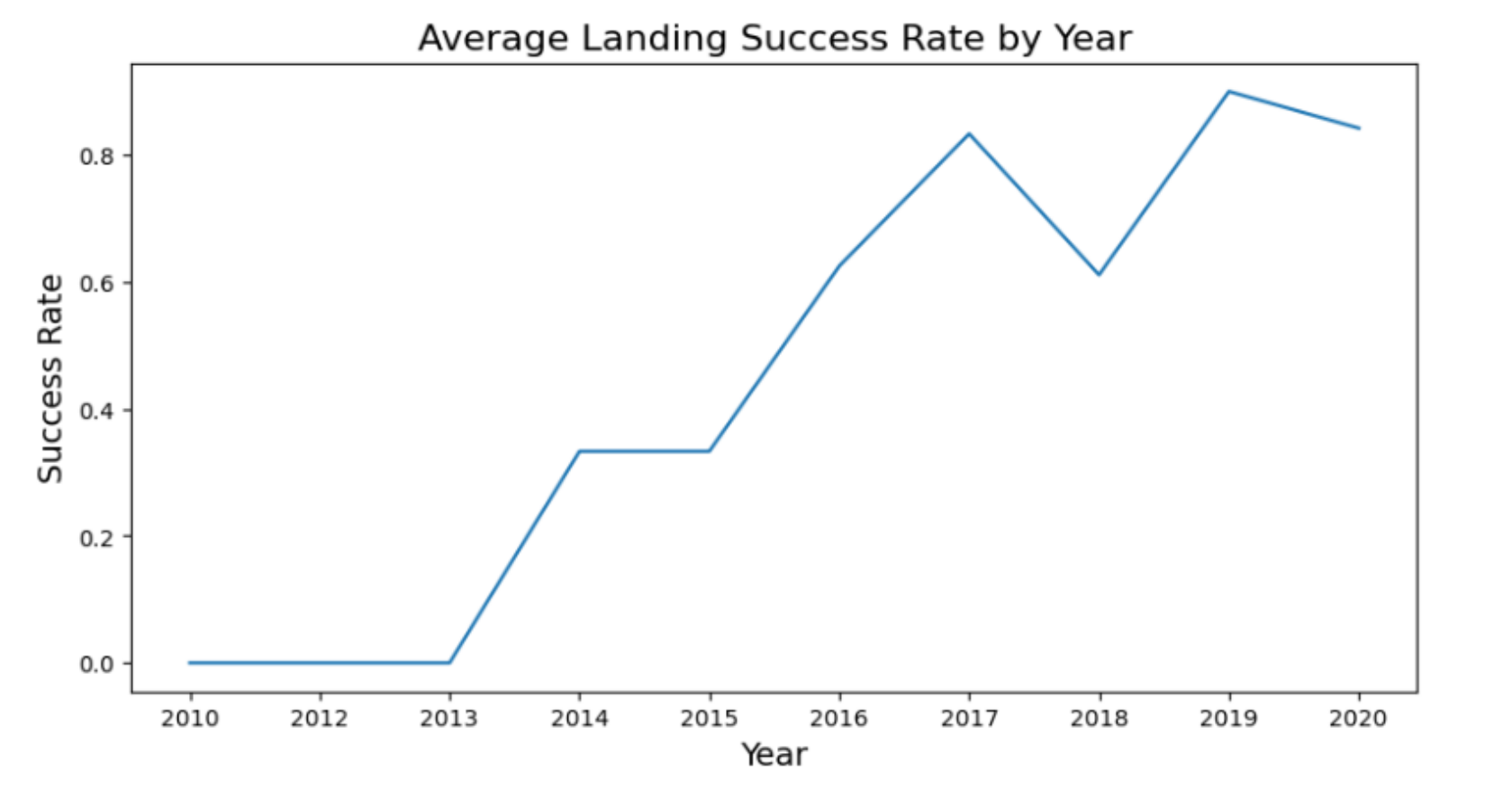
# Payload vs. Orbit Type

- This scatter plot of Payload Mass vs Orbit can be very useful for getting a rough idea about mission success based on weight and planned distance.
- SSO orbits have a 100% success at all payloads. As well as all payloads over 4000 KG to the International Space Station.



# Launch Success Yearly Trend

- Yearly average success rate, tends toward greater success.
- Peak success rate (so far) was achieved in 2019.
- Following this trend we can expect SpaceX to be even more successful in the future.



# All Launch Site Names

- Using python magic sql commands we can quickly query our SpaceX database.
- Here we can get the unique launch site names stored for each launch

```
[12]: Launch_Sites = %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTBL;
      Launch_Sites
```

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

[12]:

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

# Launch Site Names Begin with 'CCA'

- Here we find 5 records where launch sites begin with `CCA`
- This technique can be easily carried over to search through any launch feature, for information that could aid in planning a mission.

[14]:
```
%sql SELECT * FROM SPACEXTBL WHERE "Launch_Site" LIKE 'CCA%' limit 5;
```

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

[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 |

# Total Payload Mass

```
[15]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)';
 * sqlite:///my_data1.db
Done.
[15]:
```

| SUM(PAYLOAD_MASS__KG_) |
|---|
| 45596 |

- Above we have the total payload carried by boosters from NASA
- Extrapolating this technique, we have the total payload for ALL customers then list descending from most to least.
- We can see that SpaceX is by far the Leader when it comes to taking payloads into space.

```
[16]: %sql SELECT Customer,SUM(PAYLOAD_MASS__KG_) AS "TOTAL_PAYLOAD" \
      FROM SPACEXTBL \
      GROUP BY Customer \
      ORDER BY TOTAL_PAYLOAD DESC\
      LIMIT 5;
 * sqlite:///my_data1.db
Done.
[16]:
```

| Customer | TOTAL_PAYLOAD |
|---|---|
| SpaceX | 185220 |
| Iridium Communications | 67200 |
| NASA (CRS) | 45596 |
| SpaceX, Planet Labs | 31010 |
| SES | 23355 |

# Average Payload Mass by F9 v1.1

- Using SQL we can calculate statistics such as the average payload mass carried by Falcon 9's with booster version F9 v1.1
- For reference Falcon 9 v1.1 weighed about ~506,000 KG in total, with the fuel weighing about ~287,000 KG ...so we can see why reusability would be critical

```
[22]: %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.1%';
 * sqlite:///my_data1.db
Done.
[22]: AVG(PAYLOAD_MASS__KG_)
      2534.6666666666665
```

# First Successful Ground Landing Date

- Using SQL querying we can find the date of the first successful landing outcome on ground pad
- Here in the USA the date was actually 12-21-2015, but our data set is in Coordinated Universal Time

```
[23]: %sql SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)';
```

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

[23]:

| MIN(Date) |
|---|
| 2015-12-22 |

# Successful Drone Ship Landing with Payload between 4000 and 6000

- These boosters have all successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- Have a 5000 KG payload? Have a launch orbit where the booster can only land at sea? These boosters have you covered!

```
[26]: %sql SELECT Booster_Version FROM SPACEXTBL \
          WHERE Landing_Outcome = 'Success (drone ship)' \
          AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

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

[26]:

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

# Total Number of Successful and Failure Mission Outcomes

- Here we have the mission outcomes and see that regardless of landing outcome, SpaceX is quite adept at getting payloads into orbits.
- The unsuccessful mission was a ~2K KG payload to the ISS several years ago in 2015.

```
[28]: %sql SELECT COUNT(Mission_Outcome) as "Mission Success" FROM SPACEXTBL \
          WHERE Mission_Outcome LIKE 'Success%';

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

[28]:

| Mission Success |
|---|
| 100 |

```
[29]: %sql SELECT COUNT(Mission_Outcome) as "Mission Fail" FROM SPACEXTBL \
          WHERE Mission_Outcome LIKE 'Failure%';

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

[29]:

| Mission Fail |
|---|
| 1 |

# Boosters Carried Maximum Payload

- Using a Sub-Query we can list the names of the booster which have carried the maximum payload mass.
- We can see that SpaceX has been successful several times with several boosters for launching max payload.

```
[19]: %sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTBL \
          WHERE PAYLOAD_MASS__KG_ = ( SELECT max(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
 * sqlite:///my_data1.db
Done.
```

[19]:

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

# 2015 Launch Records

- Here we present a query of a list of the failed landing outcomes on a drone ship, their booster versions, and launch site names for the year 2015
- These 2 results only proceeded the first successful land-site landing by less than 1yr. Recall 1st Falcon 9 successful landing was 12-21-2015.

[36]:
```
%sql SELECT substr(Date,0,5) as Year, substr(Date,6,2) as Month, Landing_Outcome, Booster_Version, Launch_Site \
    FROM SPACEXTBL \
    WHERE substr(Date,0,5)='2015' \
    AND Landing_Outcome = 'Failure (drone ship)';
```

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

[36]:

| Year | Month | Landing_Outcome | Booster_Version | Launch_Site |
|---|---|---|---|---|
| 2015 | 01 | Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| 2015 | 04 | Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 |

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

- A Query representing the ranked 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

```
[47]: %sql with Dated_Data AS (SELECT * FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' AND '2017-03-20') \
          SELECT Landing_Outcome, COUNT(Landing_Outcome) as "Count", RANK() OVER(ORDER BY COUNT(Landing_Outcome) DESC) as "Rank" \
          FROM Dated_Data GROUP BY Landing_Outcome ORDER BY Count DESC;
```

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

[47]:

| Landing_Outcome | Count | Rank |
|---|---|---|
| No attempt | 10 | 1 |
| Success (drone ship) | 5 | 2 |
| Failure (drone ship) | 5 | 2 |
| Success (ground pad) | 3 | 4 |
| Controlled (ocean) | 3 | 4 |
| Uncontrolled (ocean) | 2 | 6 |
| Failure (parachute) | 2 | 6 |
| Precluded (drone ship) | 1 | 8 |

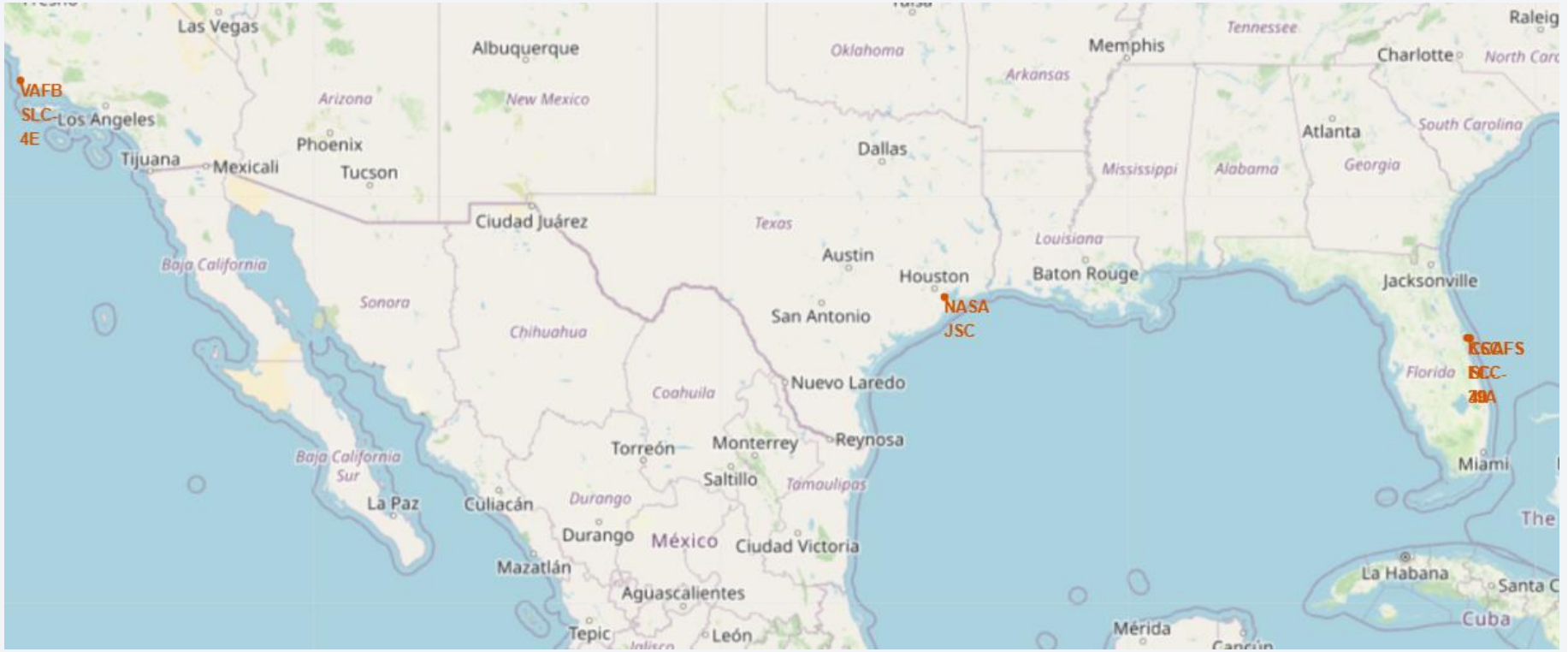
- Considering the first successful landing was in 2015, it is quite impressive that 2 successful categories made it in the top 4, in this date range.

Section 3

# Launch Sites Proximities Analysis

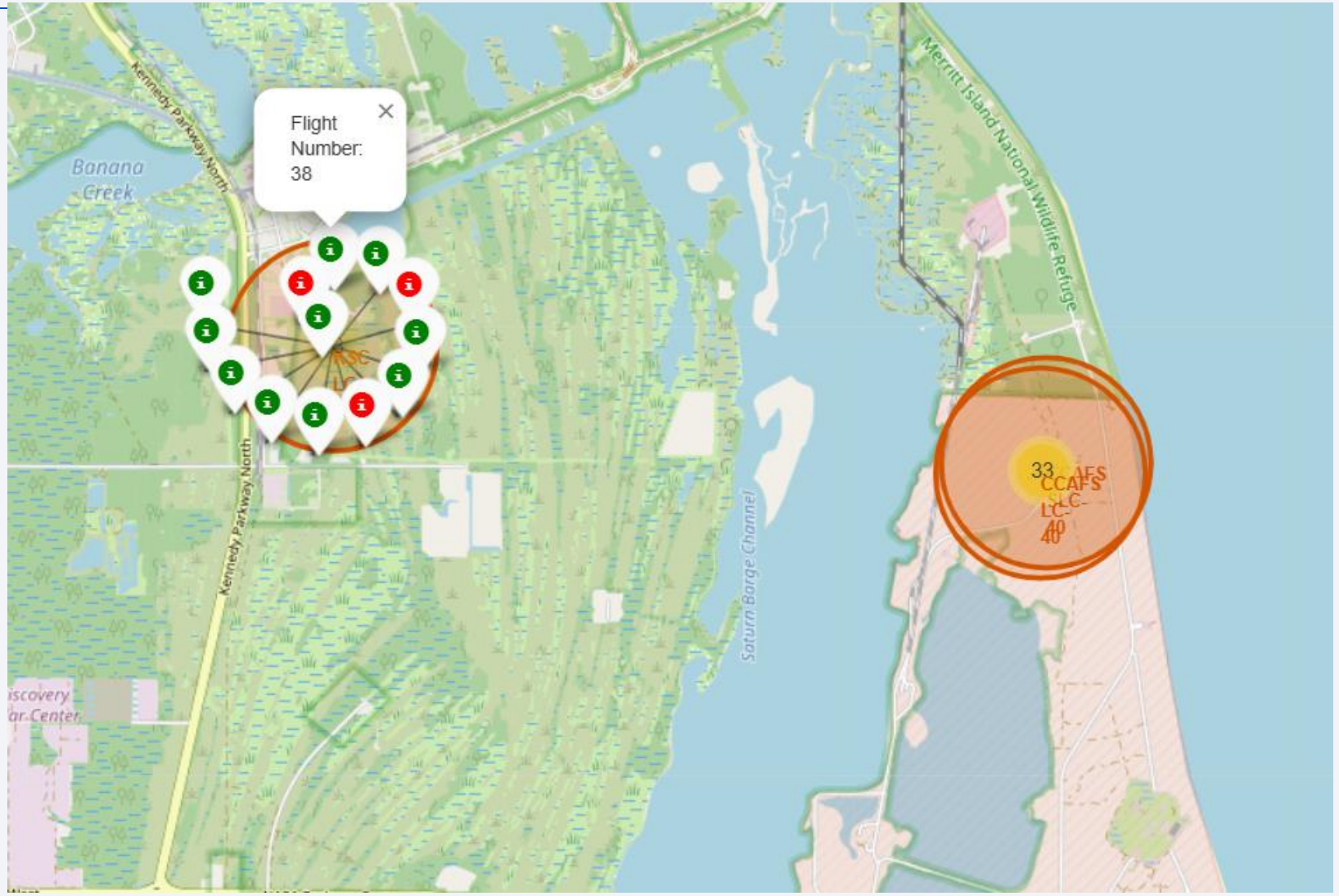
# Folium Map of Falcon 9 Launch Sites

- Using Folium we can create a map with the names of all SpaceX Falcon 9 Launch Site locations from the dataset.
- Additionally NASA's Johnson Space Center is also labeled.
- As we can see the sites are located in the US, near the equator, with several in Florida.



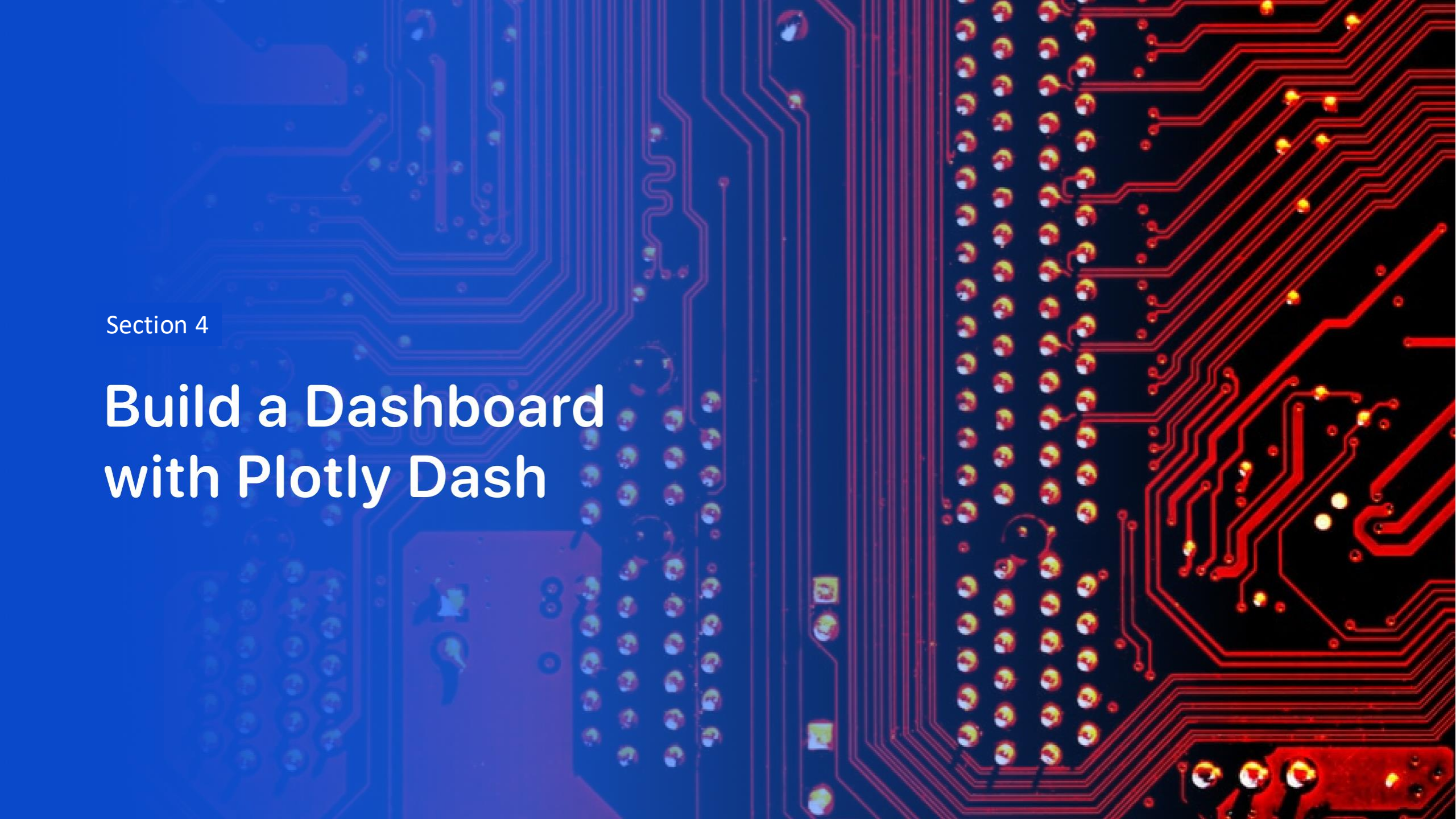
# Folium Marker Clusters Reveal Landing Success

- Using a Folium's Marker Cluster Feature we can create an interactive map for viewing the landing successes and failures at each launch site.
- With a tool as powerful as this, one can quickly view the color coded success rate at each site.
- KSC-LC39A seems to be in pretty good shape



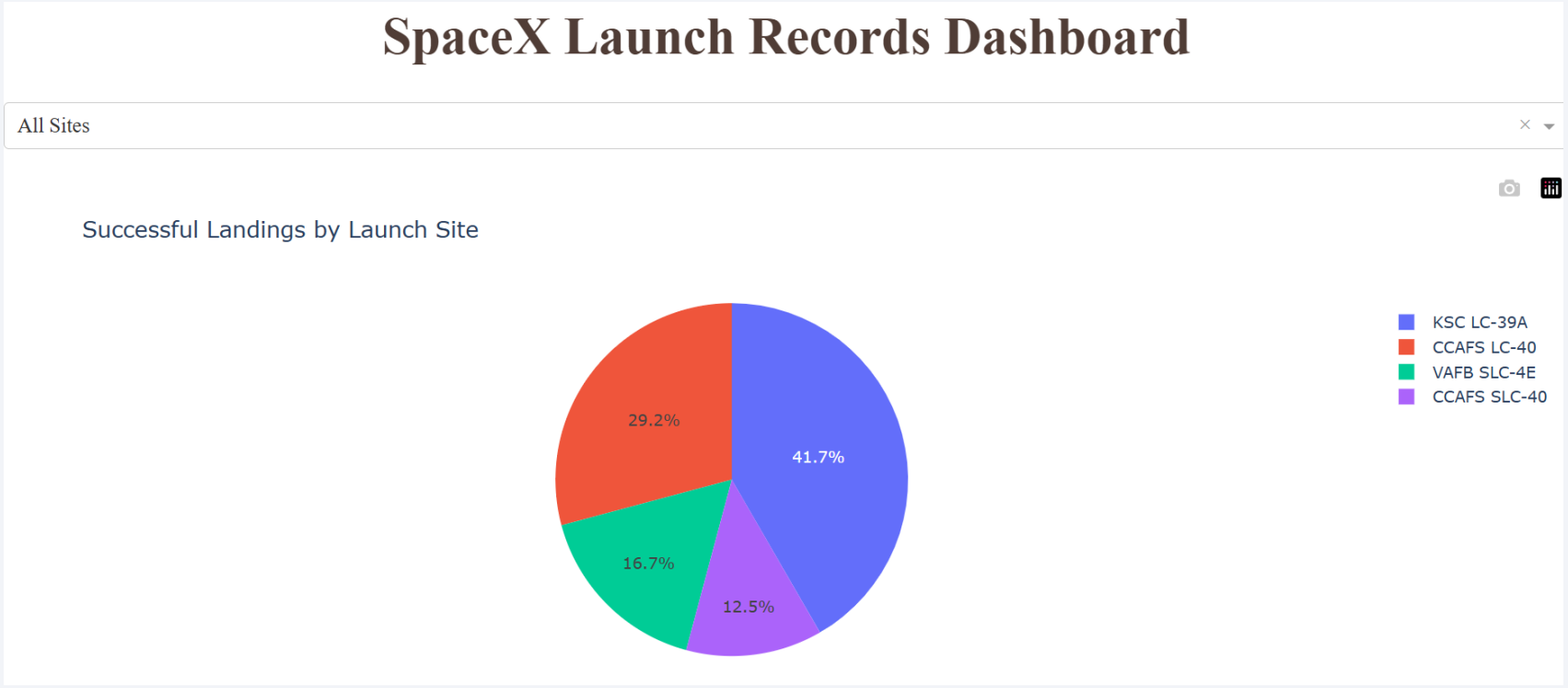
# Folium Distance Mapping to Key Landmarks

- Here we have calculated the distance to the coast from Vandenberg Space Launch Complex (1.37 KM).
- Using polylines and map markers in this method, we can determine important location features such as proximity to railways, highways, coastlines, major cities etc...
- This information could be very useful for improving launch production conditions as well as SpaceX employee on site community living conditions.


Section 4
Build a Dashboard with Plotly Dash

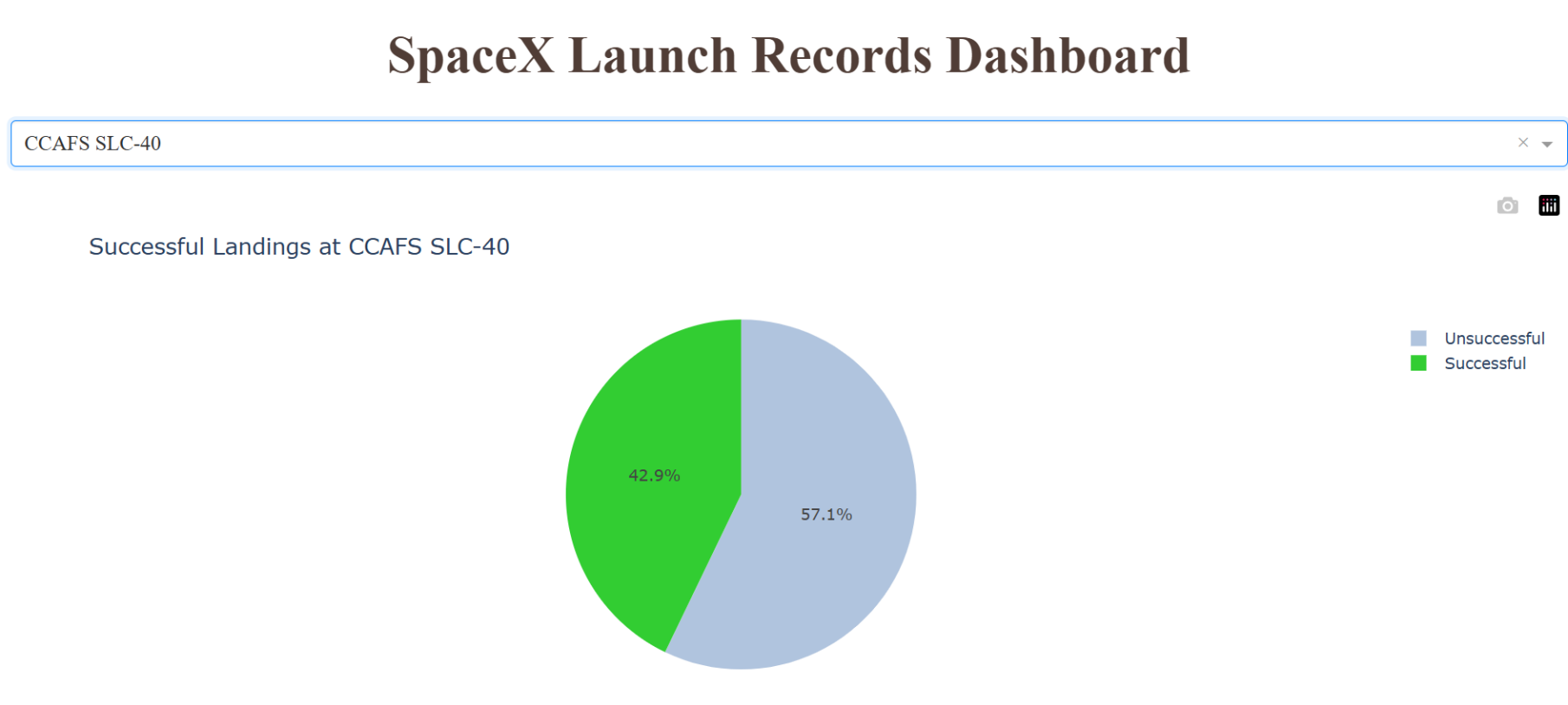
# Dashboard of Successful Launches by Launch Site

- From this diagram we can see that launch site KSC LC-39A is the most successful at Landing Falcon 9 Rockets.
- Kennedy Space Center Launch Complex 39A is located in Merritt Island, Florida.



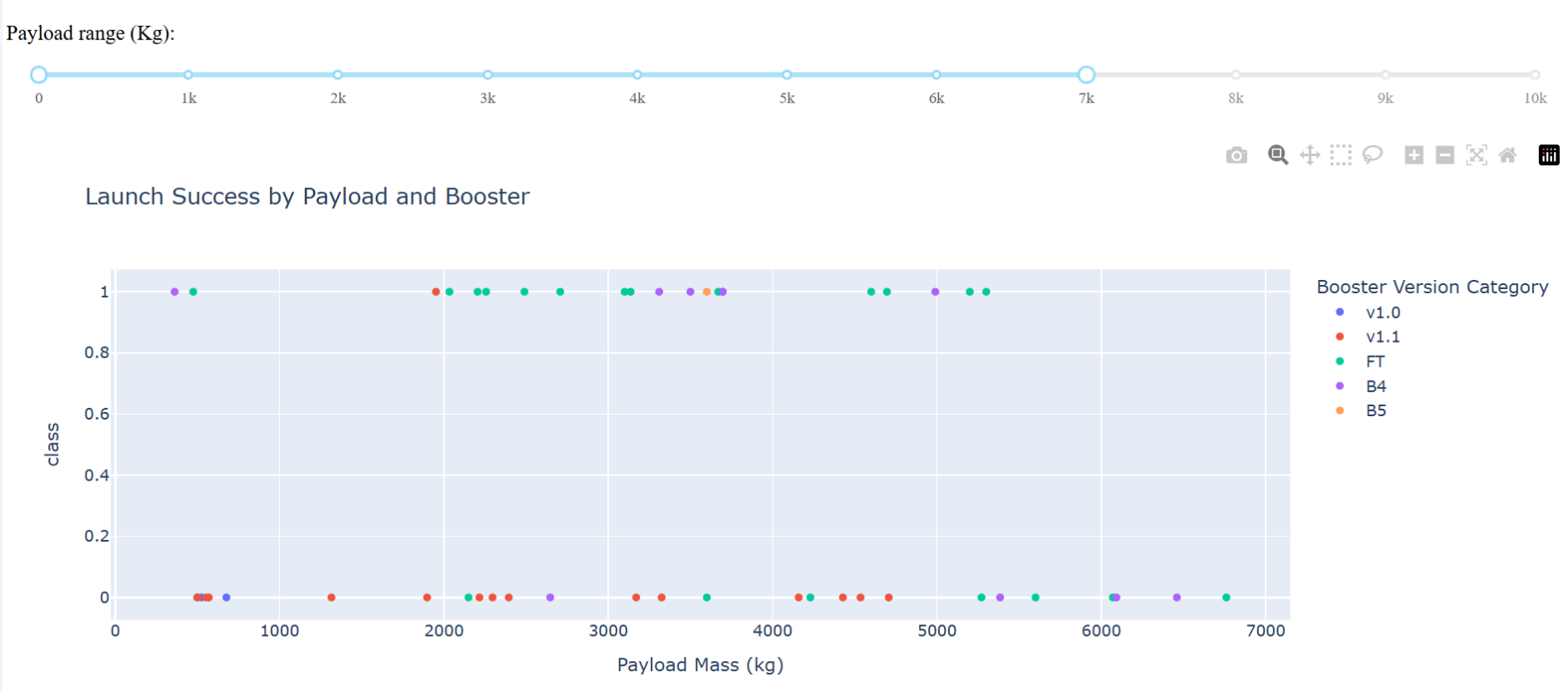
# Dashboard of Success Ratio for CCAFS SLC-40

- While KSC-LC39A is the most successful by count, CCAFS SLC-40 is the most successful by ratio.
- Cape Canaveral Space Force Station Space Launch Complex-40 is also located in Florida.



# Dashboard for Landing Success with Payload Range Slider

- Using the dashboard slider we limit the Payload to 7k Kg and compare launch landing success outcomes for all sites.
- Most of the rockets that landed successfully had a payload between 2k-5.5k kg.
- Booster Version FT had the most successful landings in this range.  While booster B5 had the best success rate being 1 for 1.



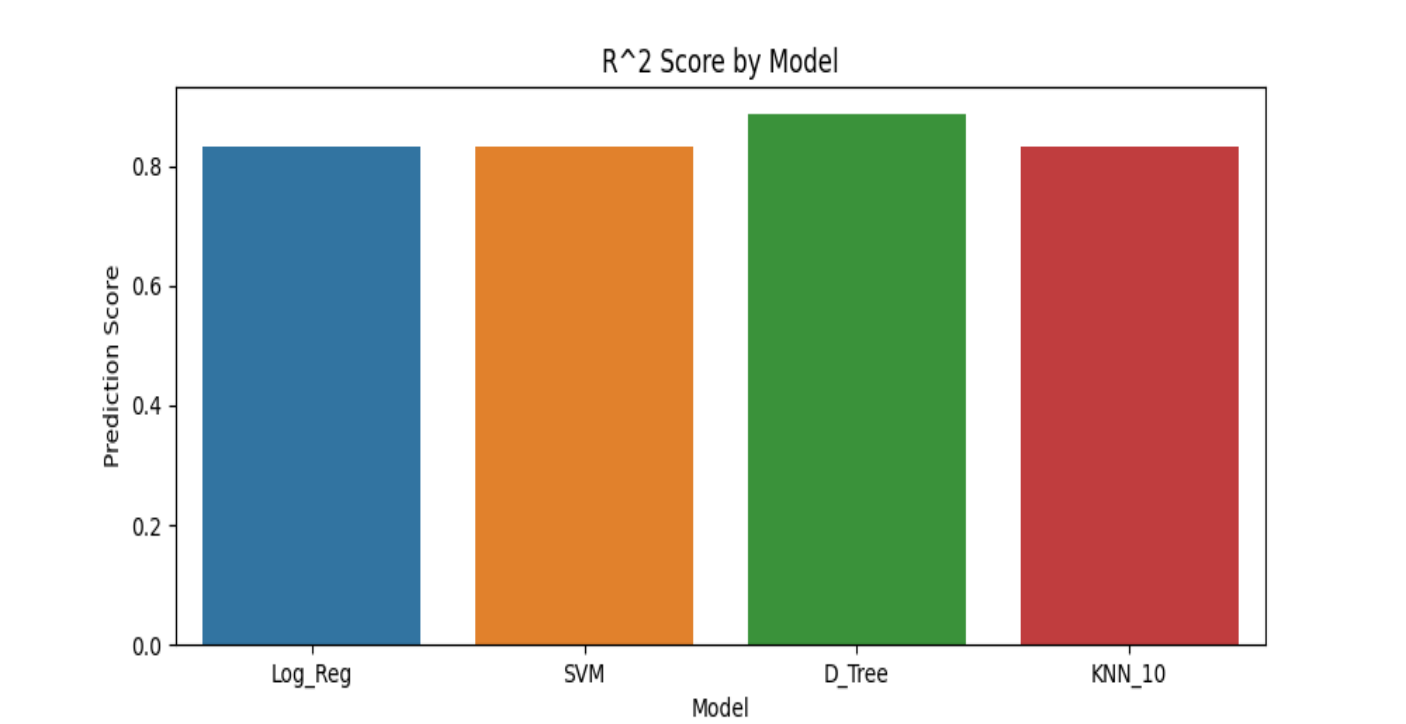
Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

- Using the Seaborn visualization library we can plot and compare the models classification accuracy.

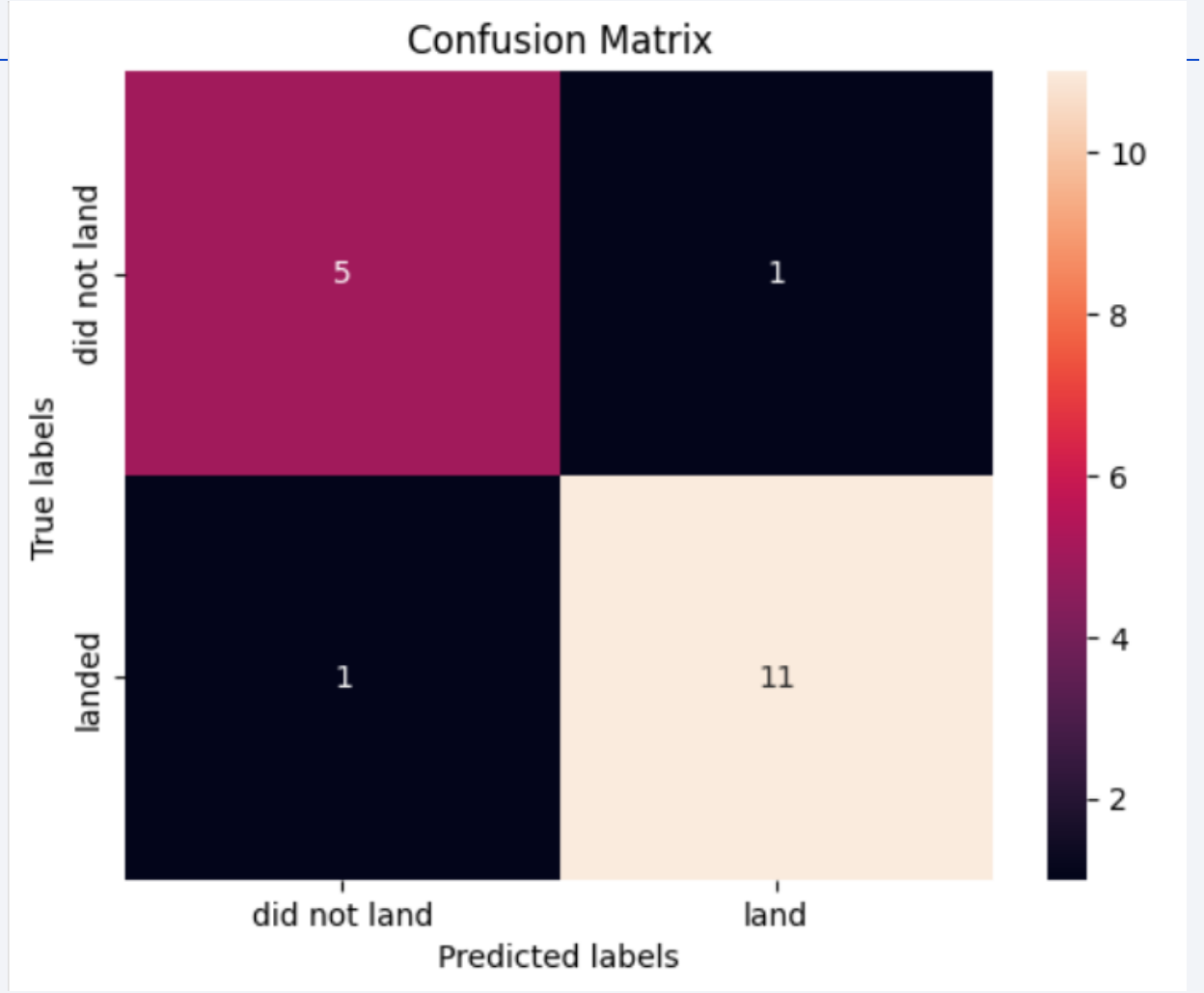
- The Decision Tree Model has highest classification accuracy



# Confusion Matrix

This is the confusion matrix for the Decision Tree model. It was the best performing model with an average of 89% prediction accuracy.

In this matrix we can see the model predicted most outcomes accurately, with the exception of 1 false positive and 1 false negative.



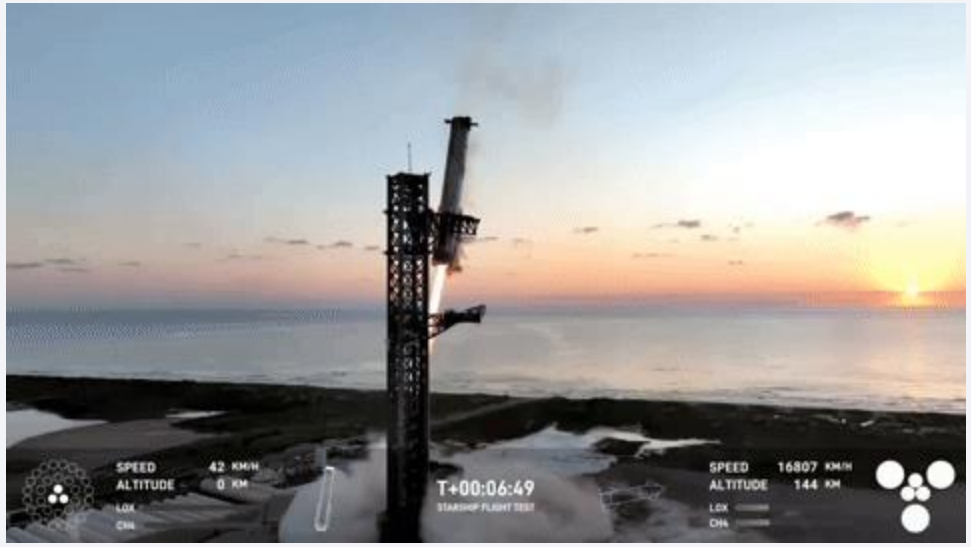
# Conclusions

- Logistic Regression
  - Testing accuracy of 83%
  - Training accuracy of 87%.
  - Errors: False Positives – predicted landing when booster did not land
- Support Vector Machine
  - Testing accuracy of 83%
  - Training accuracy of 85%
  - Errors: False Positives
- Decision Tree
  - Testing accuracy of 89%
  - Training accuracy of 89%
  - Errors: False Positives, and False Negative – predicted not landing when did land
- K-Nearest Neighbors
  - Testing accuracy of 83%
  - Training accuracy of 85%
  - Errors: False Positives

While the Decision Tree model on average performed as well or better than the other models, it was also the model with the most fluctuations. Testing accuracy went as low as 72% and as high as 89% while the other models had very little fluctuations. Also, note that the decision tree also predicted false negatives, even though that would make for a pleasant surprise in mission outcome it is not the behavior we would like the model to have for predictive analysis. It is still remarkable that we were able to achieve this accuracy given that we had less than 100 Falcon 9 launch records in our set. As launches continue and more data is acquired, we can only expect the accuracy of these predictions and SpaceX's success to continue to grow.

# A Peak into the Future

- *What will we predict next!*



# Appendix

- All relevant Python code, SQL queries, additional charts, Notebook outputs, and data sets used to build this report can be found in this GitHub repository.
- GitHub: SpaceX All Code
- https://github.com/DLPhysics/Data-Science

# Thank you!