IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Evan Cornelius
12/21/2021

Outline

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

Executive Summary

- Summary of methodologies

  Data Collection, Web Scaping, Data Wrangling, EDA – SQL & Visualization (Pandas/Matplotlib/Folium), Machine Learning Predictions

- Summary of all results

  After creating a dataset and defining a research question it was clear that using a logistic regression provides for the lowest training accuracy when determining whether a SpaceX rocket will land successfully or not. After comparing different ML models, we learned that with the amount of data collected, Decision Trees, KNN, and SVM all produce similar results to one another. The test results provided by those three models produce more precise test predictions than the logistic model, with the Decision Tree's model taking the lead as best performer. Therefor when choosing the most optimal model, you must compare multiple metrics, not just the test data accuracy of a model.

Introduction

- Project background and context

  Import data from the SpaceX API and predict whether a Falcon 9 rocket will land successfully or not in the first stage.

- Problems you want to find answers

  Which variables are we most focused on and how can we use these variables and observations to build machine learning models. These models will be able to predict and release insights not obtainable by plainly looking at the data. We can compare different models to see which one will perform best with our data.

Section 1

Methodology

Methodology

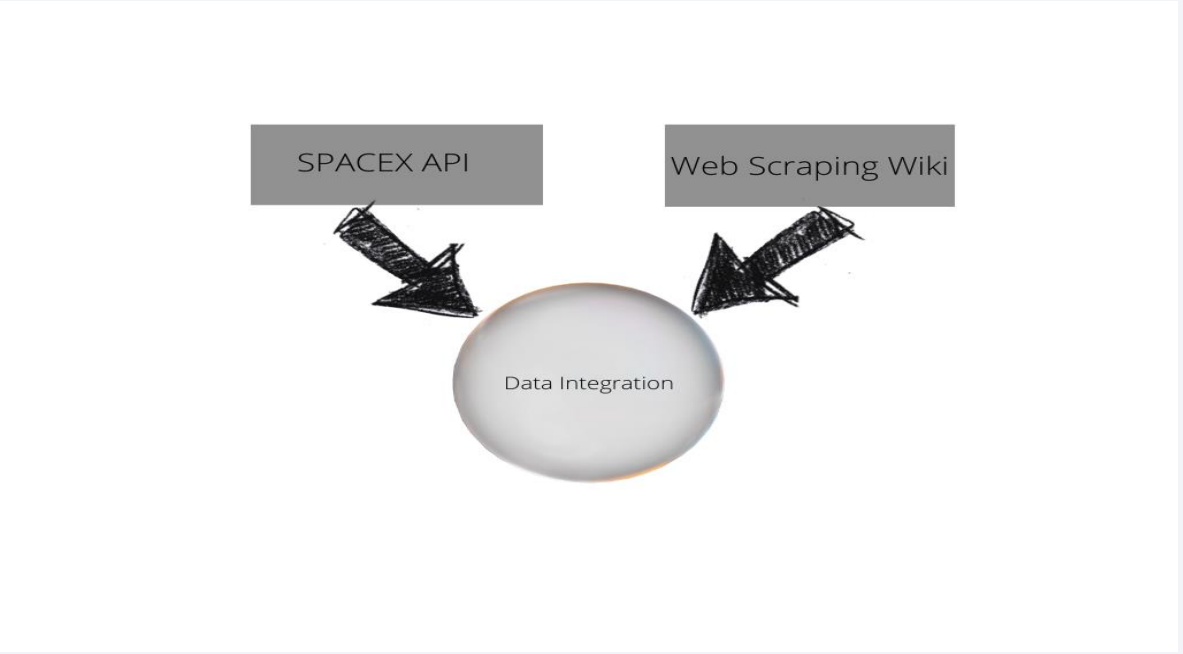
Executive Summary

- Data collection methodology:
  - Using the SpaceX API and Web Scraping from WIKI
- Perform data wrangling
  - Remove missing data entries and format collected data into cleaned dataset
- 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, and evaluate different classification models

Data Collection

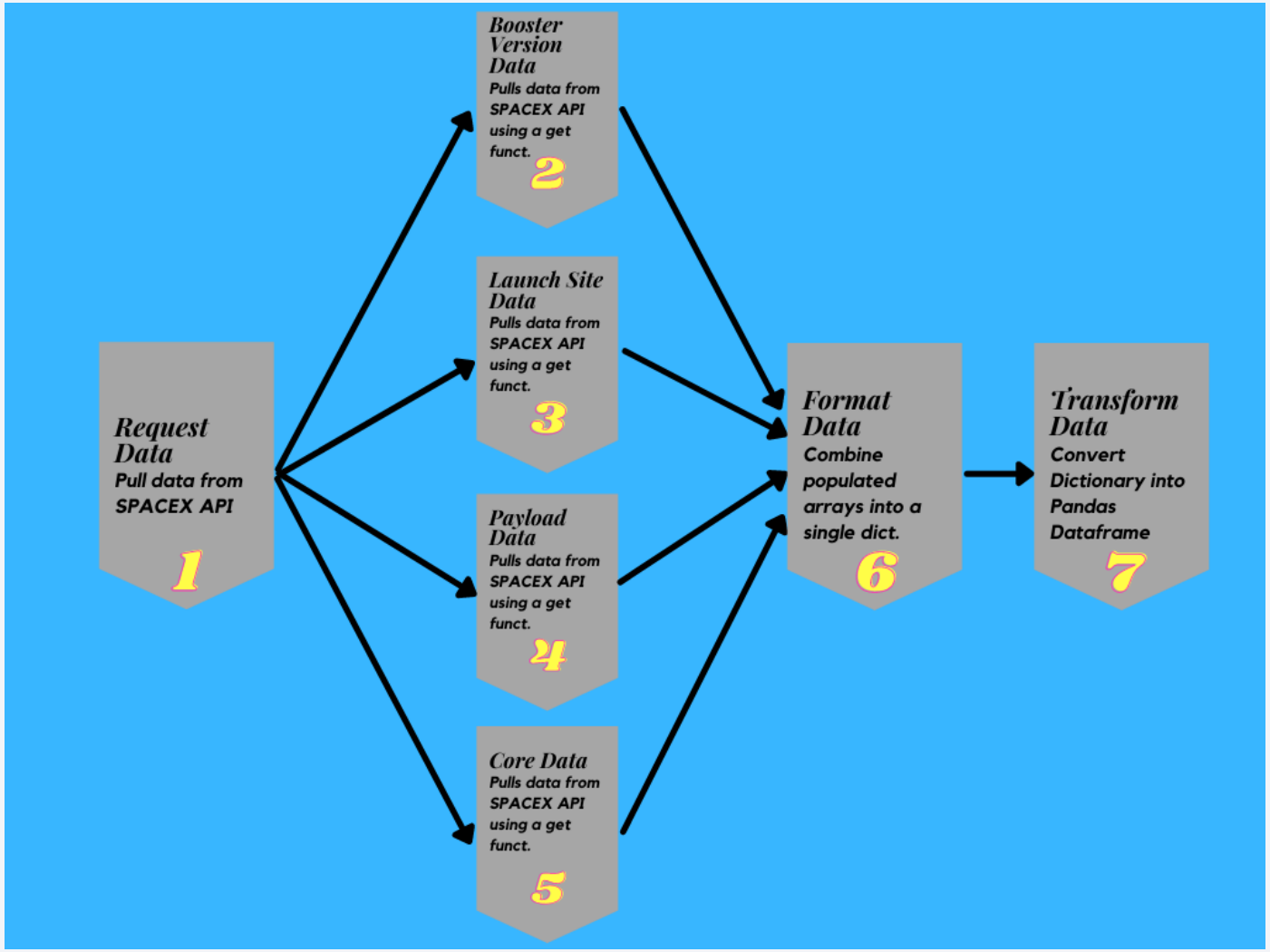
Web Scraping & API Integration

- Pulled information from the WIKI on Falcon 9 launches into a dataset
- Pulled additional information from the SpaceX API into a dataset



Data Collection – SpaceX API

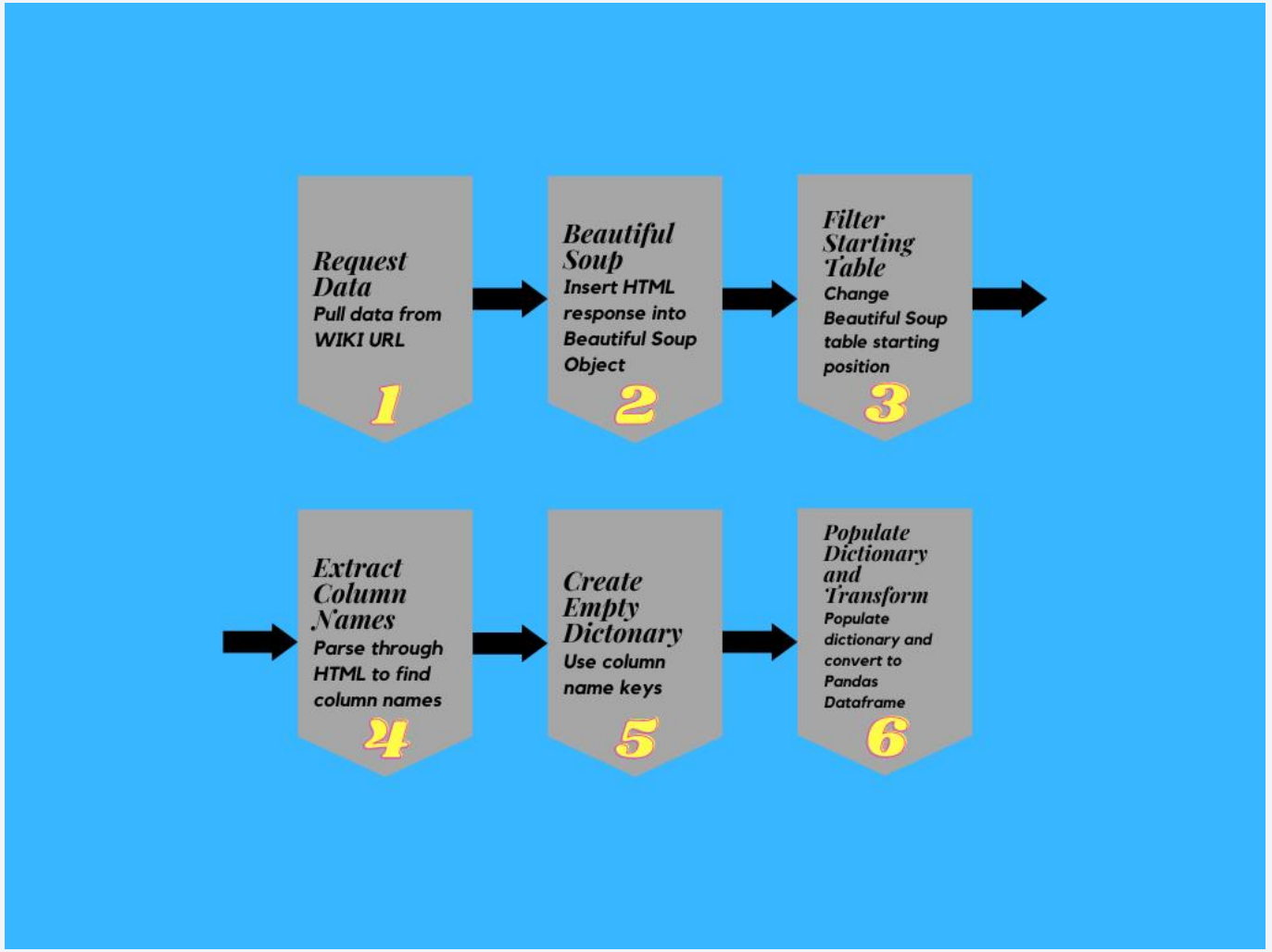
- SpaceX REST calls
  - 4 get requests to pull information into predefined functions to populate empty arrays
  - Format arrays into a Dataframe



For an example, please refer to this notebook.

Data Collection - Scraping

- Beautiful Soup
  - Parse HTML and Extract Relevant data
  - Format data into a Dataframe



For an example, please refer to this notebook.

Data Wrangling

- Remove Observations that are not conclusive to our study, for example rockets with additional boosters and multiple payloads.
- Converted the date feature to only include the date, removing the time object.
- Check missing values, removing or filling in the data where it is needed.
- Create a new column called “Class” which will contain a binary value on whether the rocket had a successful landing or not. This value is based off the landing outcomes column in the data set. To parse the results, we created a bad outcomes variable which contains false landings in the landing outcomes column.

Refer to this notebook for a reference.

EDA with Data Visualization

- Scatter plots – Helped to visualize which features were most important when trying to predict successful landings
- Line chart – Visualize yearly trends
- Bar chart - Comparing categorical vs. numerical feature analysis

Refer to this notebook for a full reference.

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
  - Displayed the names of unique launch sites & produce 5 launch records for a specific launch site
  - Displayed a payload mass carried by a specific customer (NASA CRS)
  - Produced the average payload masses carried by a certain booster (F9 v1.1)
  - Found the date of the first successful landing outcome on a ground pad
  - List of boosters which had a successful drone ship landing, with a certain range of payload mass
  - Tally of all successful and failed mission outcomes
  - Used a subquery to list off all boosters which have carried a maximum payload mass
  - Queried specific failed landing outcomes, booster versions, and launch site names for a specific year (2015)

Refer to this github link for a reference of the code.

Build an Interactive Map with Folium

- Started off by creating a map detailing the location of all the different launch sites, and then added in another layer showing the number of successful/failed launches per launch site. To wrap up the folium map I calculated distance lines to random proximities like railroads, highways, coastlines, and major cities.
- I added these proximities to discover differing insights related to the launch site's location in an interactive way.

- Refer to this github link for a reference of the code.

Build a Dashboard with Plotly Dash

- The dashboard I created includes a dropdown menu to select a specific launch site with a connected callback to render a correct pie chart depending on the specific launch site selected in the dropdown menu. The pie chart displayed the correct percentage of each type of booster launched based on the selected launch site. Additionally, I had added an interactive range slider to select specific payload mass ranges. The range slider is connected to another callback function which updates a scatter plot showing launch outcomes and their respective payload masses. Each point in the scatter plot is color coded and labeled with their unique booster version.
- This dashboard was created to visually see the success of specific booster versions based off payload mass and launch site.

Refer to this github link for a reference of the code.

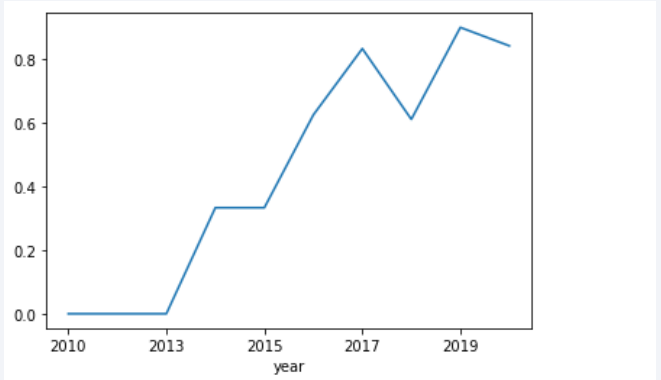
Predictive Analysis (Classification)

- Created a test and training data split to use throughout all my classification models. Next, I used GridSearchCV to implement hyperparameter tuning in each of the models. After building out each model, we can evaluate the models based off their respective test and training classification accuracy values. Furthermore, we can evaluate each model based off the results produced within their respective confusion matrix's. By comparing both training accuracies, testing accuracies and confusion matrix's we can determine the best performing classification model.

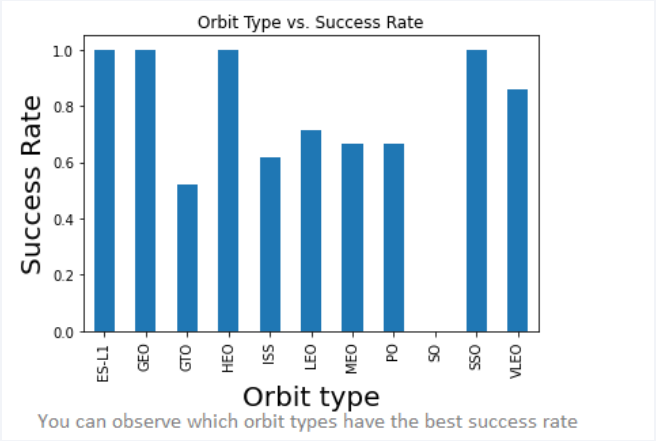
Refer to this github link for a reference of the code.

Results

- The results from EDA show us many different relevant insights

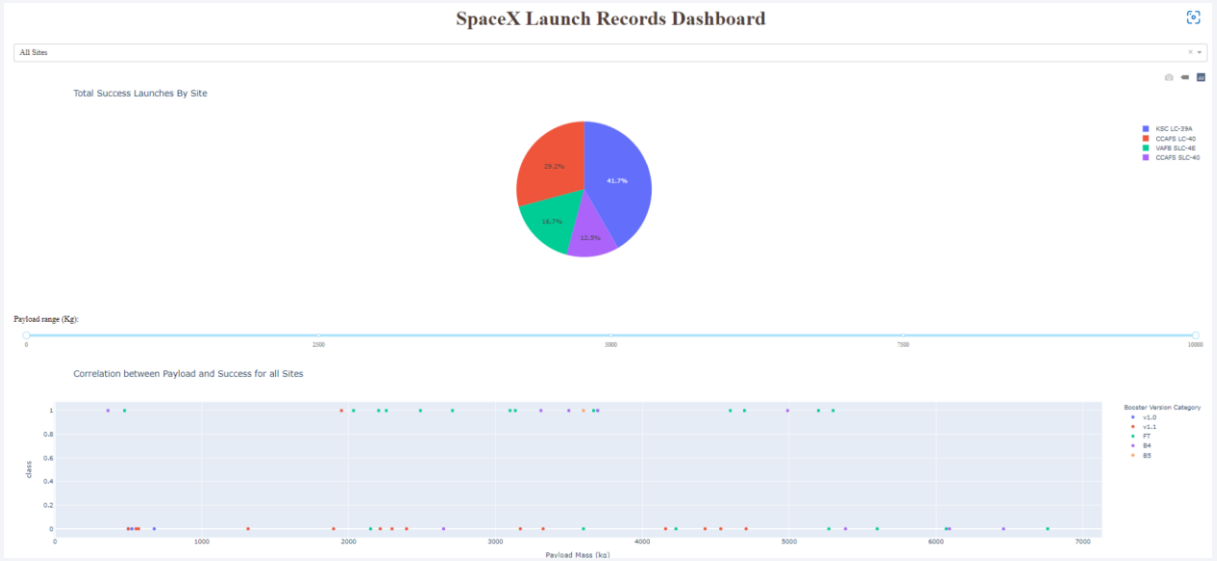


you can observe that the sucess rate since 2013 kept increasing till 2020



You can observe which orbit types have the best success rate

- Interactive analytics using plotly dash



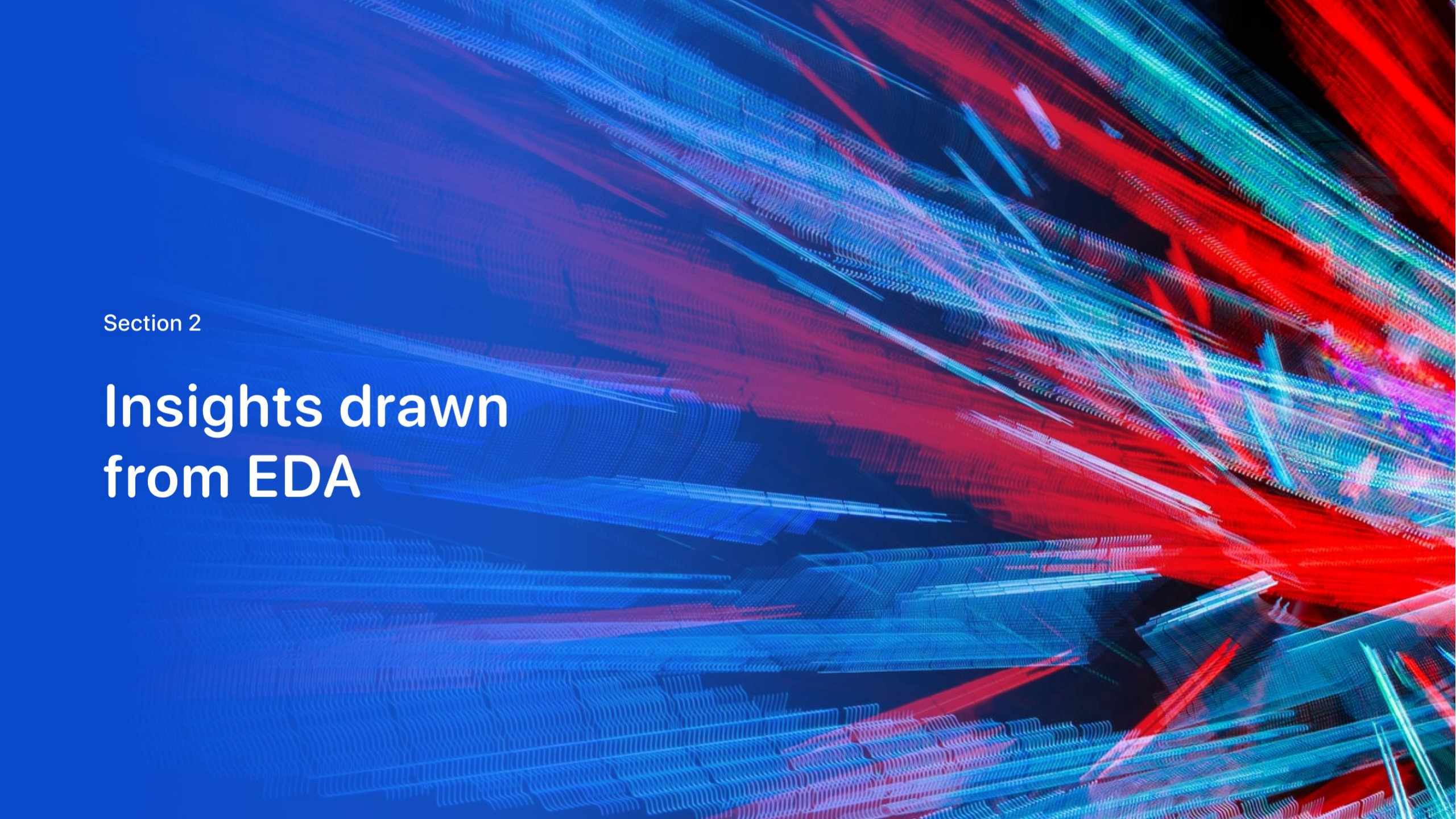
- The data for this project was sourced from the SPACEX API and WIKI for Falcon 9 launches. It was cleaned and organized into a singular dataset where analysis on different variables could take place.

Predictive analysis results

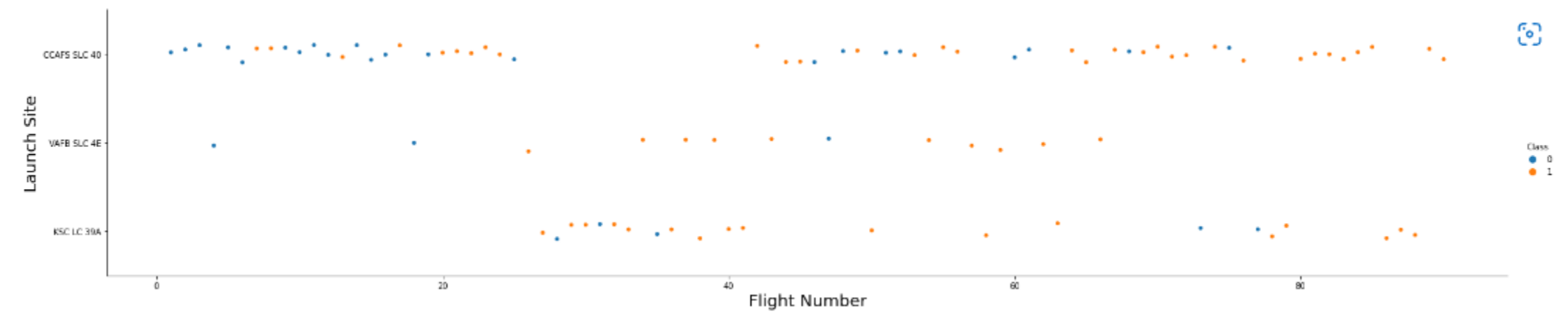
- After building a Logistic Regression, Support Vector Machine, Decision Tree, and K-Nearest Neighbors model, it was easy to tell that the logistic regression was the worst performing model. The SVM, Decision Tree, and K-Nearest models all produced the same test accuracies and almost identical confusion matrices. After looking over all the models, it was obvious that the Decision Tree model was the best performer because of its 0.9196 training accuracy and 0.7777 test accuracy.

Section 2

Insights drawn from EDA

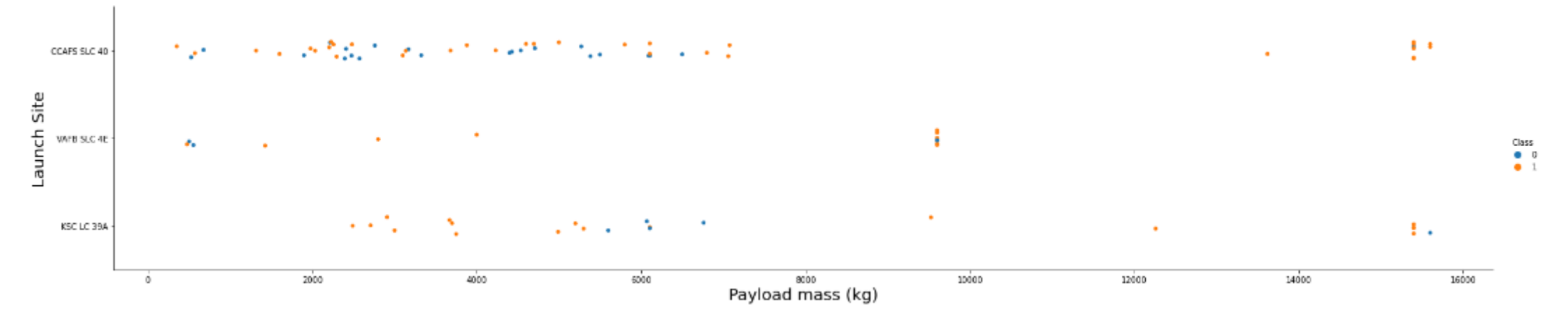


Flight Number vs. Launch Site



- The scatter plot above depicts a high failure rate for low flight numbers and rockets launched from launch site CCAFS SLC 40. Additionally, the trend seems to indicate that there were fewer landing failures as the flight number increased.

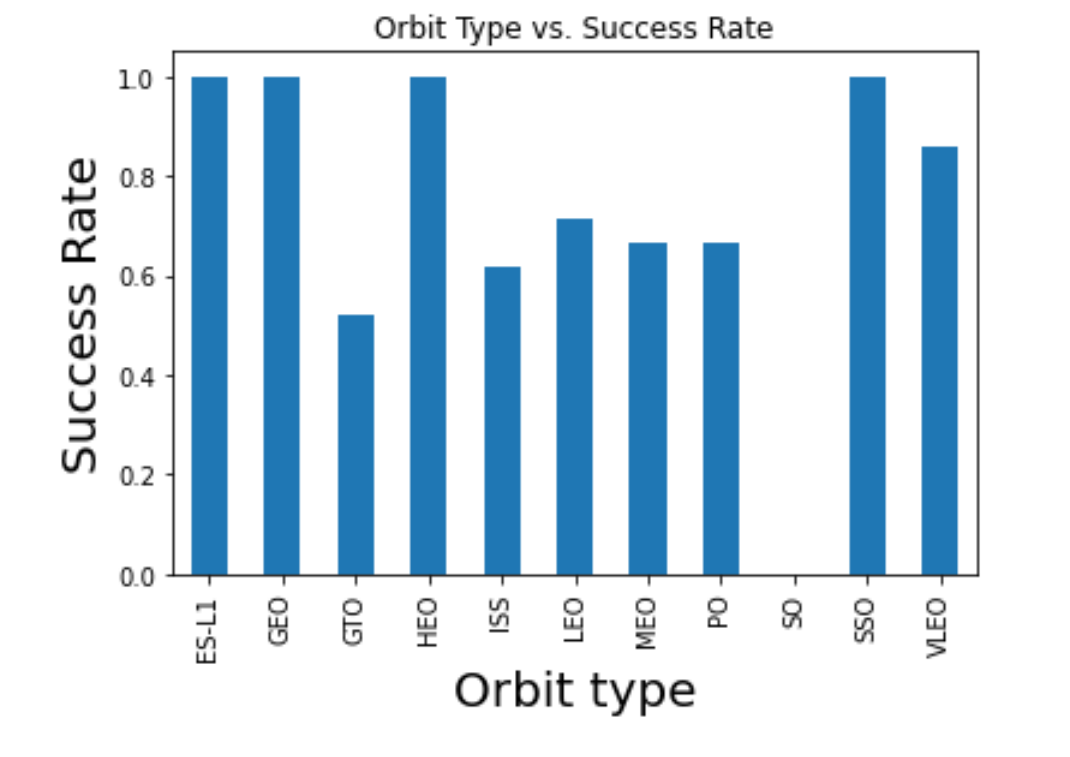
Payload vs. Launch Site



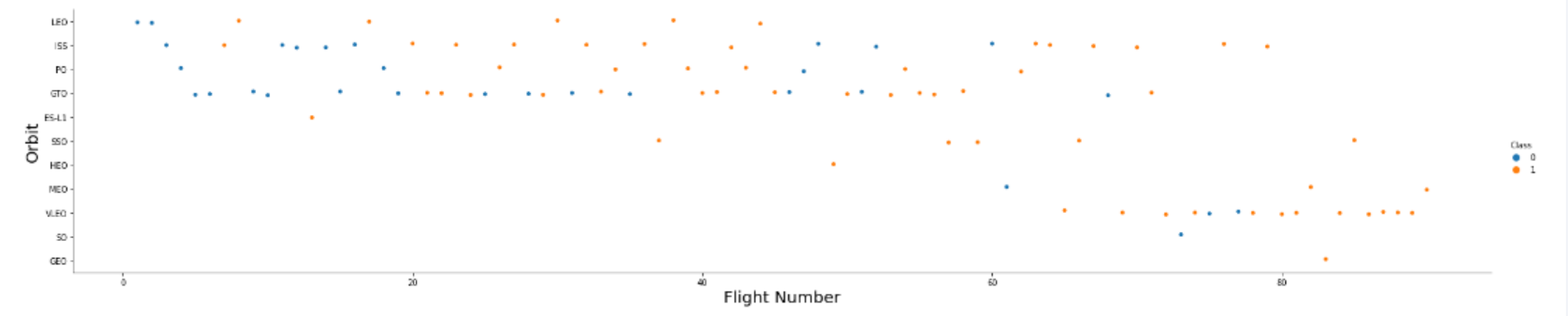
- This scatter plot shows the correlation between launch sites and payload mass of each rocket. For the VAFB-SLC launch site there were no rockets launched with a payload mass over 10,000 kg. We can infer that the VAFB-SLC launch site can only handle lighter payloads.

Success Rate vs. Orbit Type

- By looking at the bar plot it is easy to infer that not all orbit types have the same success rates. For example, SO has the lowest success rate at 0%. Orbits ES-L1, GEO, HEO, and SSO all have success rates of 100%.

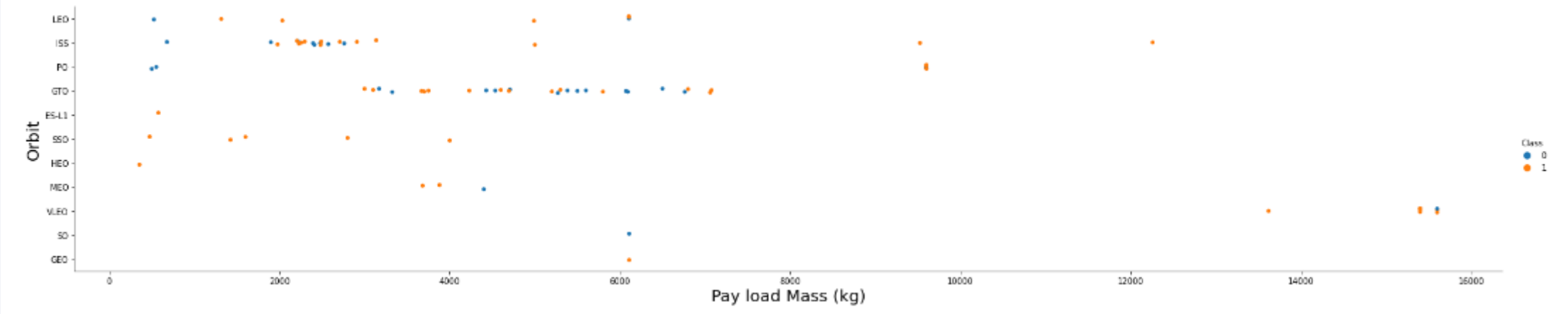


Flight Number vs. Orbit Type



- This scatter plot shows the relationship between orbit types and flight numbers. We can infer for the LEO orbit that the success rate is directly corelated to flight number, because the scatter shows increased success as the flight number increases. However, for all the other orbit types there seems to be no direct relationship with flight number.

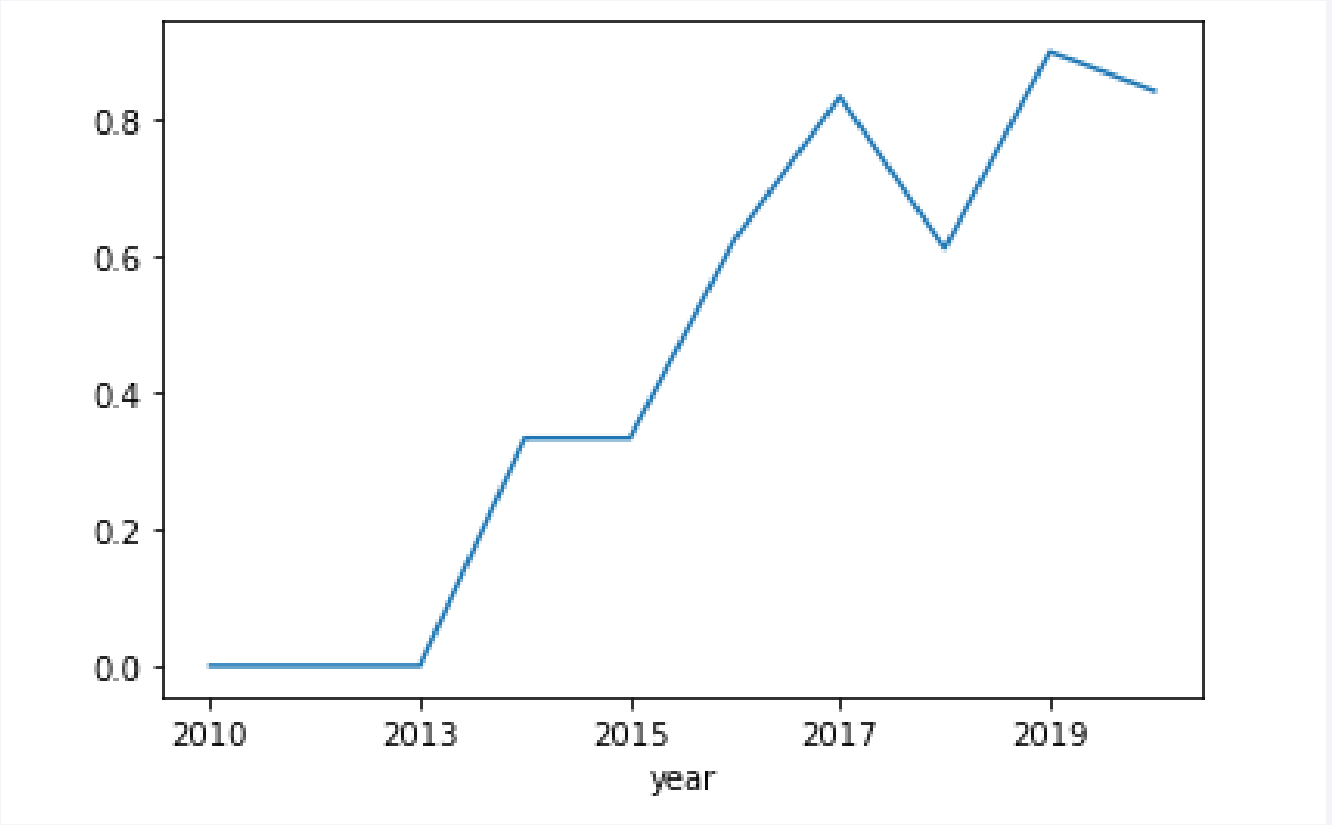
Payload vs. Orbit Type



- After analyzing the relationship between payloads and orbit type, we can conclude that certain orbit types are more successful when the payload is either heavy or light. For example, we can tell that the PO orbit is more successful when carrying heavier payloads. Additionally, we can determine that the SSO orbit prefers lighter payloads, but we cannot be definitive because there is no data regarding heavier payloads for SSO orbit launches.

Launch Success Yearly Trend

- By looking at the line chart we can definitively conclude that the success rate of rocket launches since 2013 has steadily increased until 2020. However, we can also see that 2018 was not a good year for rocket launches as we had a little downward trend in the success rate.



All Launch Site Names

- We can easily select all our unique launch sites from the SpaceX table we imported into DB2 on IBM's Cloud using SQL

```
In [5]: %sql SELECT unique(LAUNCH_SITE) FROM SPACEXTBL;
```

Out[5]:

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

Launch Site Names Begin with 'CCA'

- SQL to find 5 records of launch sites that begin with the string ‘CCA’

In [6]:
```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

Out[6]:

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

- In this SQL query we can find the total payload mass carried by boosters contracted out by NASA

```
%sql SELECT customer, sum(payload_mass__kg_) as total_payload FROM SPACEXTBL WHERE customer LIKE 'NASA (CRS)%' Group By customer;
```

Out[7]:

customer	total_payload
NASA (CRS)	45596
NASA (CRS), Kacific 1	2617

Average Payload Mass by F9 v1.1

- In this query we can display the average payload mass for F9 v1.1 boosters

In [8]:
```
%sql SELECT booster_version, avg(payload_mass__kg_) as avg_payload FROM SPACEXTBL WHERE booster_version LIKE 'F9 v1.1%' Group By booster_version;
```

Out[8]:

booster_version	avg_payload
F9 v1.1	2928
F9 v1.1 B1003	500
F9 v1.1 B1010	2216
F9 v1.1 B1011	4428
F9 v1.1 B1012	2395
F9 v1.1 B1013	570
F9 v1.1 B1014	4159
F9 v1.1 B1015	1898
F9 v1.1 B1016	4707
F9 v1.1 B1017	553
F9 v1.1 B1018	1952

First Successful Ground Landing Date

- This query demonstrates how easily we can extract the date in which the first successful ground landing happened

```
In [9]: %sql SELECT min(DATE) FROM SPACEXTBL WHERE LANDING__OUTCOME LIKE 'Success (ground pad)'
```

```
Out[9]:
1
2015-12-22
```

Successful Drone Ship Landing With Payload Between 4000 and 6000

- We can find the boosters which have a successful landing on drone ships using this query. It is also filtered to only show boosters with a specific payload mass between 6000 kg and 4000 kg

```
In [10]: %sql SELECT booster_version FROM SPACEXTBL WHERE LANDING__OUTCOME LIKE 'Success (drone ship)' AND payload_mass__kg_ < 6000 AND payload_mass__kg_ > 400
```

Out[10]:

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

Total Number of Successful and Failure Mission Outcomes

- This query demonstrates the total number of successful and failed mission outcomes for all booster versions. One thing to note, this is not representation of boosters that were destroyed because the mission outcomes sometimes do not include a successful landing, but rather just a successful payload drop off. There are other metrics outside the scope of this project that would consist of a successful mission outcome based on predetermined metrics for the booster's mission.

```
In [11]: %sql SELECT mission_outcome, COUNT(*) FROM SPACEXTBL GROUP BY mission_outcome ORDER BY 2 DESC;
```

Out[11]:

mission_outcome	2
Success	99
Failure (in flight)	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- The result of this subquery shows all booster versions which carried a maximum payload. This is important because we can subjectively state that this is our heavy-duty booster

```
In [12]: %sql SELECT booster_version FROM SPACEXTBL WHERE payload_mass__kg_ = (SELECT max(payload_mass__kg_) FROM SPACEXTBL)
```

Out[12]:

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

- This query provides an example of all the failed landing outcomes for boosters landing on a drone ship. The output also provides the launch site name, booster version, and filters the date for 2015 launches only.

In [13]:
```
%sql SELECT booster_version, Launch_Site, Landing__outcome FROM SPACEXTBL WHERE LANDING__OUTCOME LIKE 'Failure (drone ship)' AND date BETWEEN '01-01-2


#AND date BETWEEN '01-01-2015' AND '12-31-2015'
```

Out[13]:

booster_version	launch_site	landing__outcome
F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- This query lists the count of landing outcomes between June 4th, 2010 and March 20th, 2017 in descending order. We can see that there were 30 successful missions and 1 failed mission.

In [18]:
```
%sql SELECT Landing__outcome, COUNT(Landing__outcome) FROM SPACEXTBL WHERE date BETWEEN '06-04-2010' AND '03-20-2017' GROUP BY Landing__outcome ORDER BY COUNT(Landing__outcome) DESC
```

Out[18]:

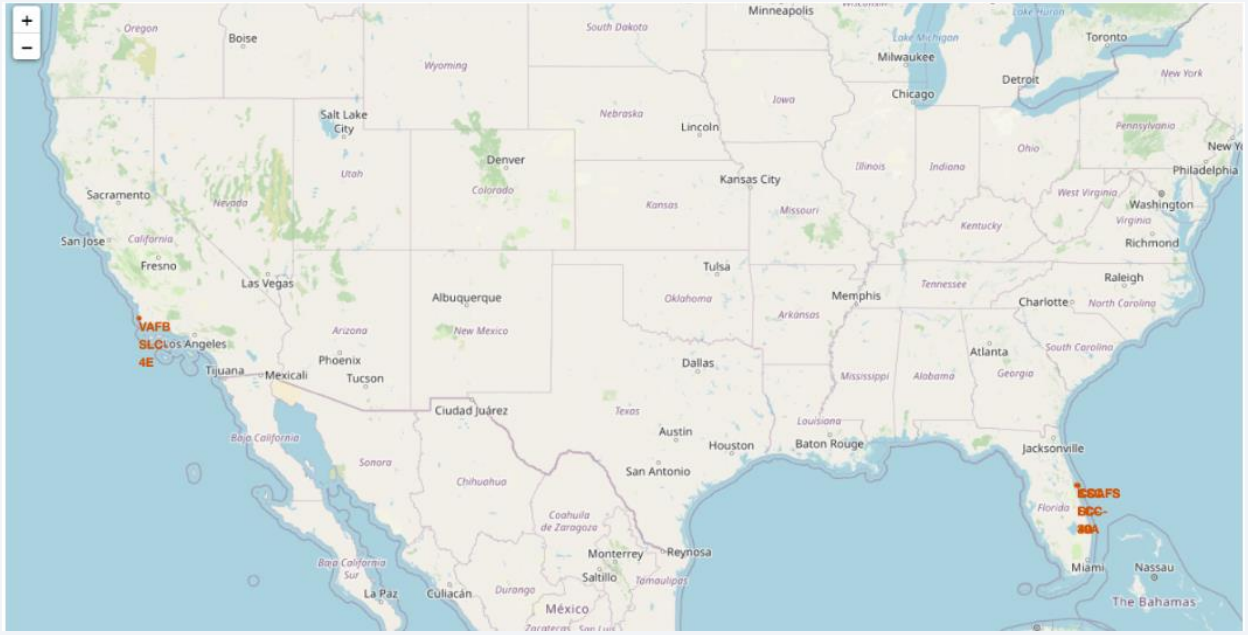
	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

Section 4

Launch Sites Proximities Analysis

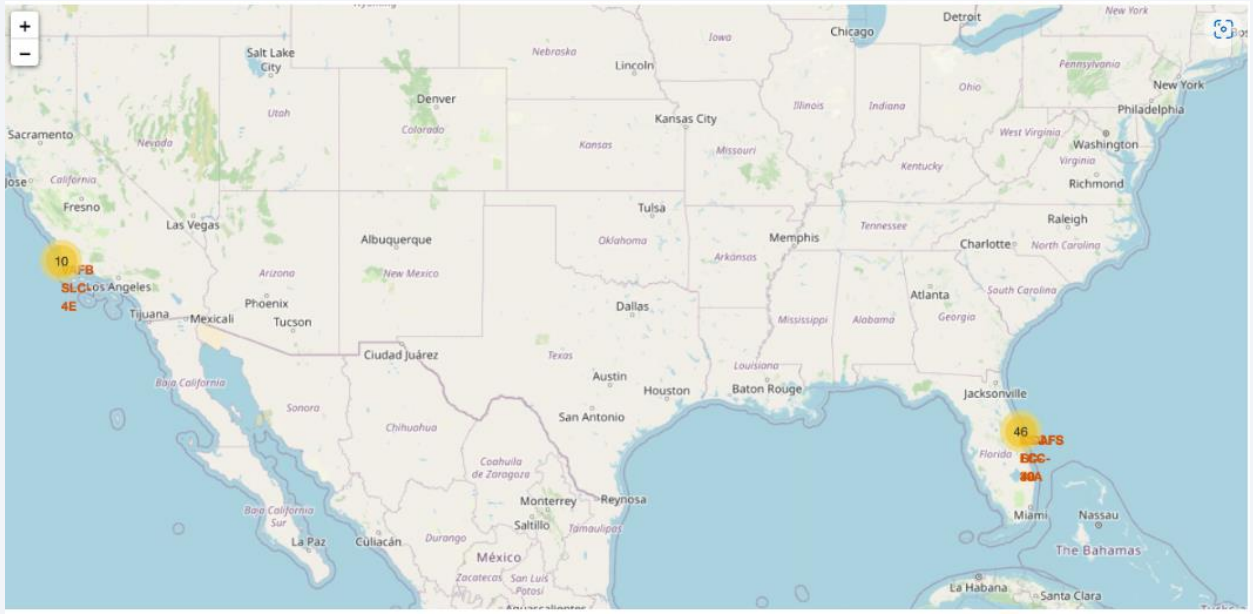
Folium Map – All Launch Sites

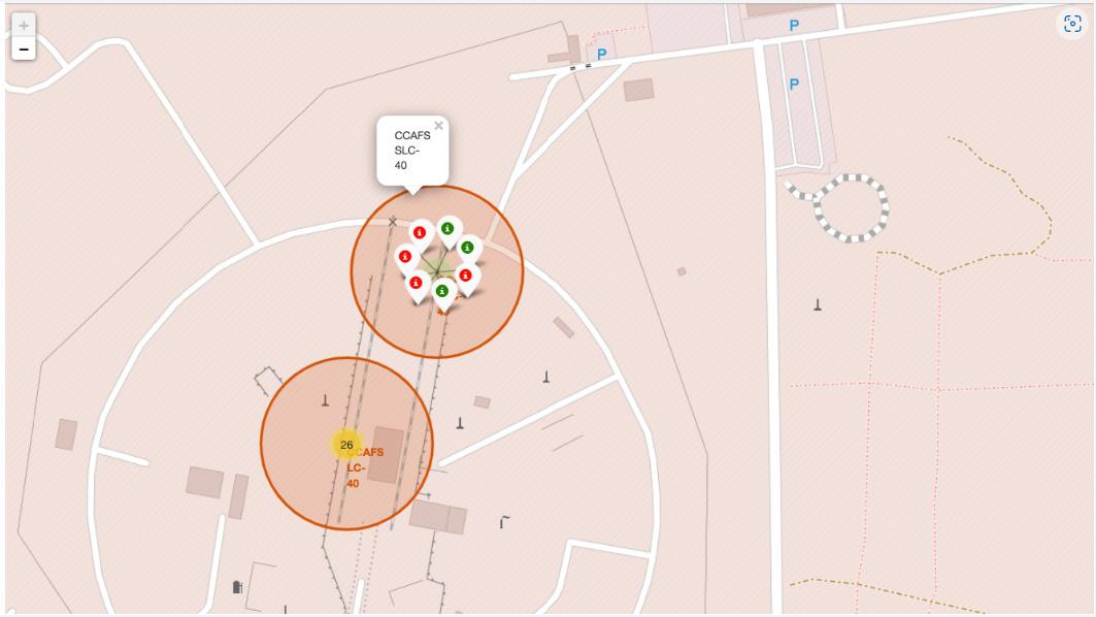
- This Folium Map shows us where all of SpaceX's launch sites are. As we can see there is one launch site in California and multiple overlapping launch sites in Florida. We can assume its safest to launch rockets on the coast, over the ocean.



Folium Map – Diving Deeper

- This portion of the interactive Folium Map allows us to dive deeper into each launch site. For each site we can see color coded successes and failures as well as the total number of launches. This is very important because it allows us to determine the most successful launch sites visually.



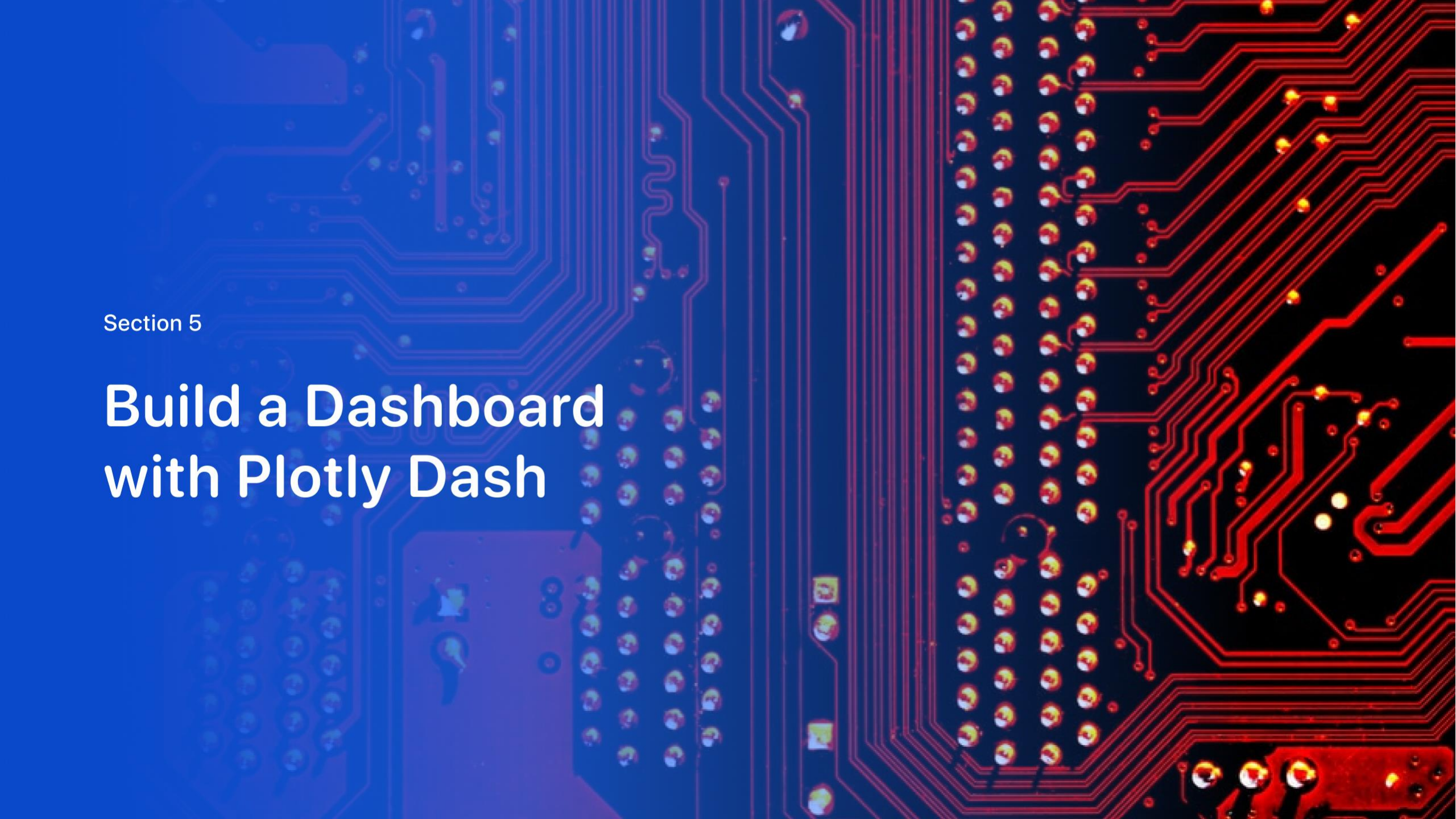


Folium Map – Proximities

- In this Folium Map we have created distance lines to differing proximities. This is very interesting because we can see all sorts of different infrastructures near our launch sites. For example, in this screenshot we can determine that the launch site is roughly 0.90KM from the coastline.

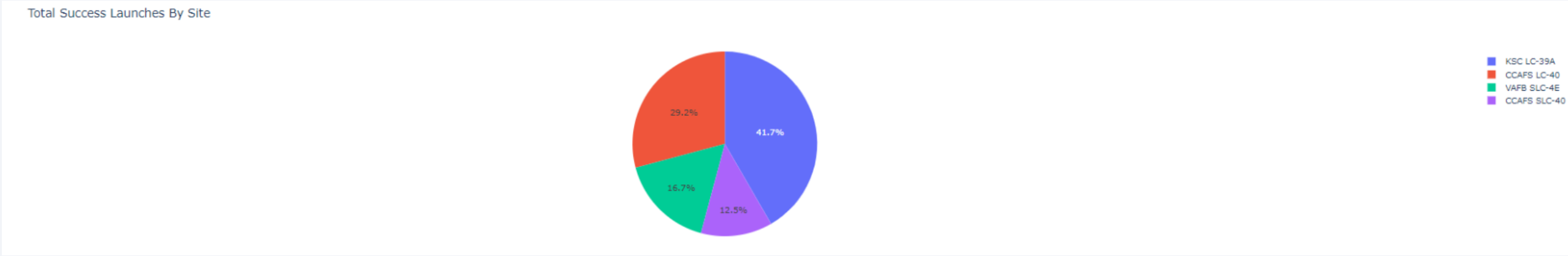
Section 5

Build a Dashboard with Plotly Dash



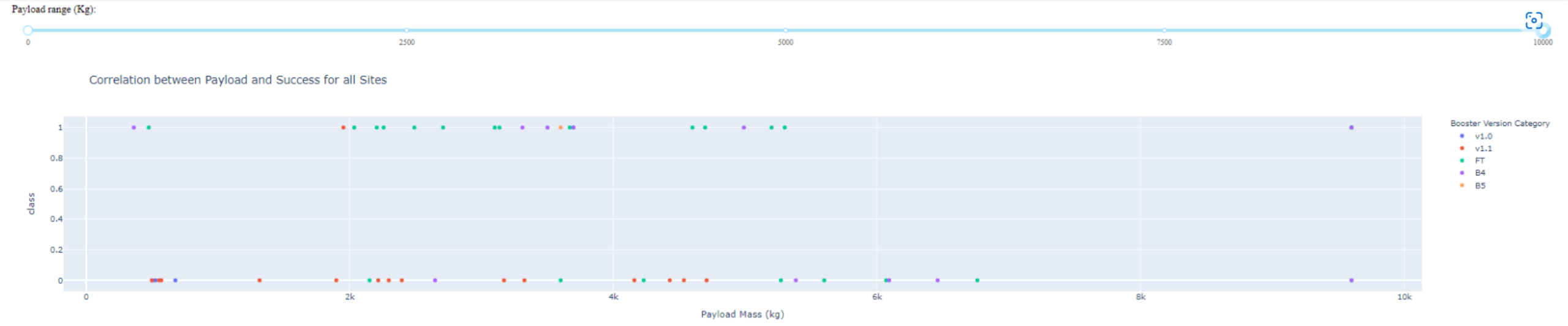
Plotly Dashboard – Successful Launches By Site

- Using the interactive Plotly Dashboard, we can filter which launch site we would like to examine, or as shown, we can see all the launch sites at once. The pie chart below shows the success rates of all our launch sites within the scope of our dataset.



Plotly Dashboard – Payload vs. Success All Sites

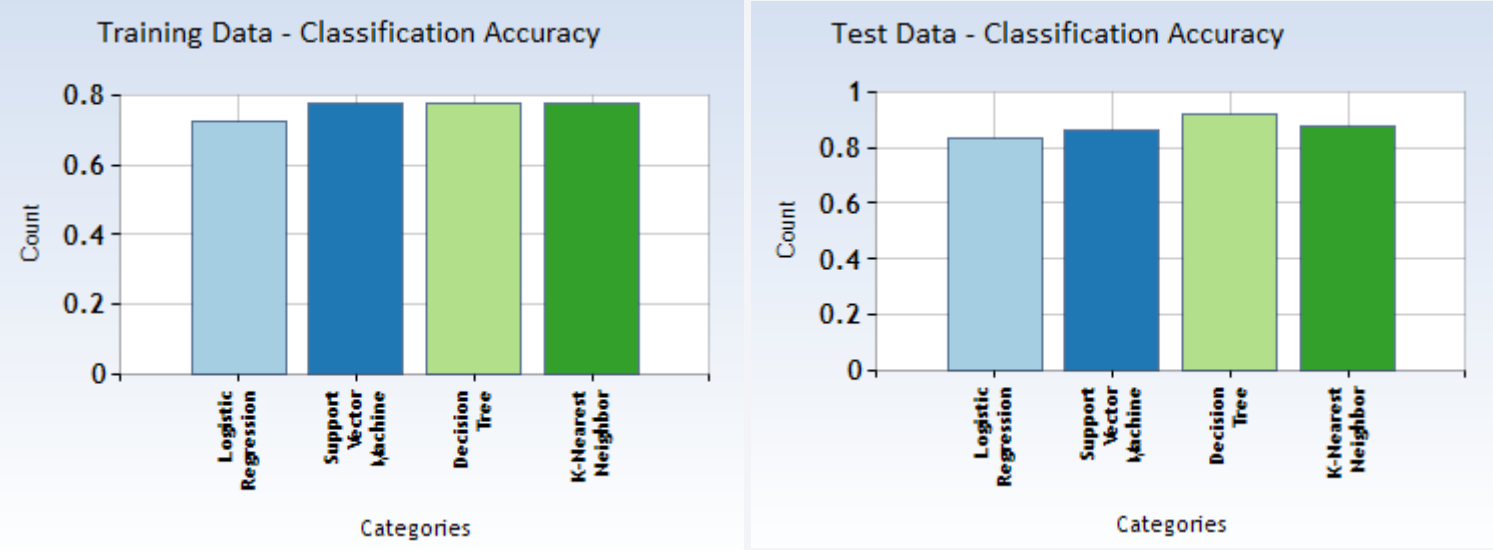
- The scatter plot below can be controlled with the range slider above it. This is all implemented in the Plotly Dashboard as well. As we can see it is apparent that the FT booster version has astonishing success when the payload is between 2,000 kg and 4,000 kg. We can also infer that the v1.1 booster version is not very successful for all payload weights.



Section 6

Predictive Analysis (Classification)

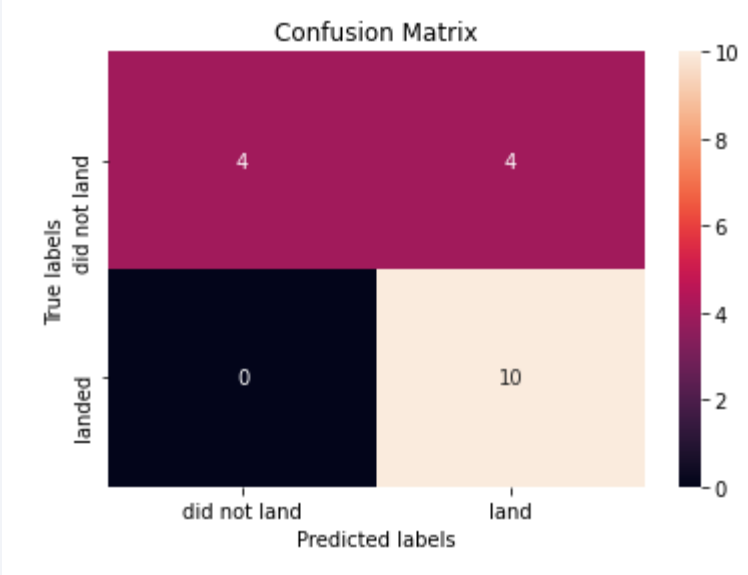
Classification Accuracy



- By looking at our model's classification accuracy for the training data it is hard to pick the best model because the accuracies are so close together. However, when we look at the models testing classification accuracy it is clear the decision tree model is the best performer coming in at roughly 91%.

Confusion Matrix

- The confusion matrix below is for the decision tree model. Without calculating the F-score, the analysis of this matrix confirms that we have 4 false positives and 0 false negatives. In this case, the 4 false positives means that our model predicted that 4 rockets would land, however they crashed instead.



Conclusions

Key takeaways from the results

- Flight success is directly correlated to flight number
- Launch success steadily increases from 2013 through 2020
- Different orbits are directly correlated to success rates
- 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 code used in this project can be found in my GitHub repository. Click here to access and view the code.

Thank you!