IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Peter Willcocks
6th February 2025

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction

- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- Project background and context

  Space X 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 Space X 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 space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Problems you want to find answers

  - What factors determine if the rocket will land successfully?
  - The relationship between various features that determine the success rate of landings?
  - What operating conditions need to be in place to ensure a successful landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia
- Perform data wrangling
  - Missing values replaced with mean value for features
  - One-hot encoding was applied to categorical features
- 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

SpaceX API

Get request from SpaceX API → Decoded Json file .json() → Converted to Pandas dataframe → Data cleaned-checking and replacing missing values

Web scraping from Wikipedia

Webscraping falcon 9 launch records from wikipedia → Parsed table data with Beautiful Soup → Converted to Pandas dataframe

Data Collection – SpaceX API

https://github.com/pjwillcocks/Data-Science-Capstone-Project-/blob/9acdcea36d7045edb90a573b68ba1ed42a075b9f/1%20jupyter-labs-spacex-data-collection-api.ipynb

Get requests from SpaceX API for data using static json object

response = requests.get(static_json_url)

Decode response and convert to pandas dataframe

data = pd.json_normalize(response.json())

Used SapceX API calls to obtain additional data for Booster Version, Launch Site, Payload and Core

Populated dictionary with additional SpaceX API data and wanted from features from first dataframe

Converted dictionary into pandas dataframe

Data Collection - Scraping

- https://github.com/pjwillcocks/Data-Science-Capstone-Project-/blob/9acdcea36d7045edb90a573b68ba1ed42a075b9f/2%20jupyter-labs-webscraping.ipynb

Get request from Falcon 9 Launch Wikipedia

response = requests.get(static_url).text

↓

Parsed with Beautiful Soup

soup = BeautifulSoup(response, ‘htmal.parser’)

↓

Extract of tables

↓

Extract all column names

↓

Create empty dictionary of column names

↓

Extract and spend data into dictionary

↓

Convert dictionary into pandas dataframe

Data Wrangling

First Stage – Deal with missing values in SpaceX API data

Identify null values → Calculate mean value of feature with missing data → Replace missing data with mean value

Second Stage – Assign categorical variable of success and failure to each flight

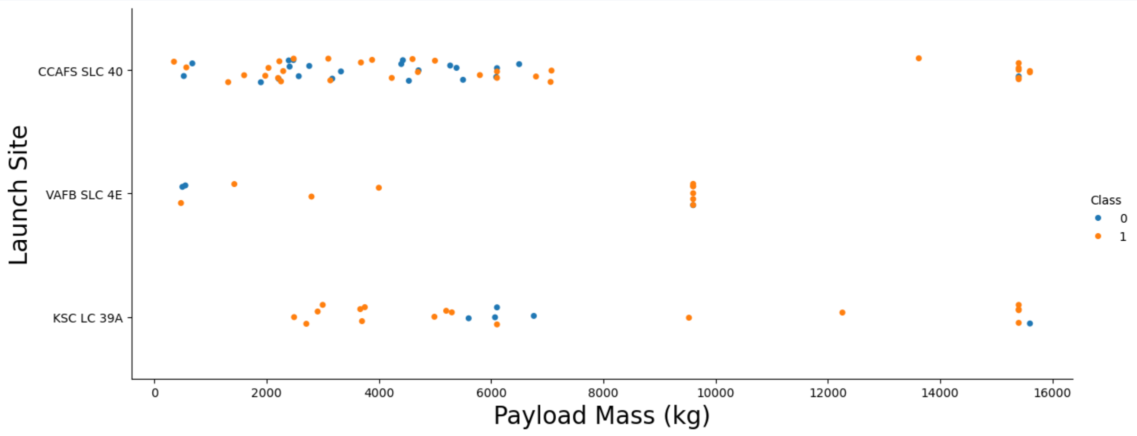
Calculate number launches each site → Calculate number and occurrence of each orbit → Calculate mission outcome from each orbit → Create column and label for landing outcome, good or bad → Assign categorical variable for success and failure for each landing outcome

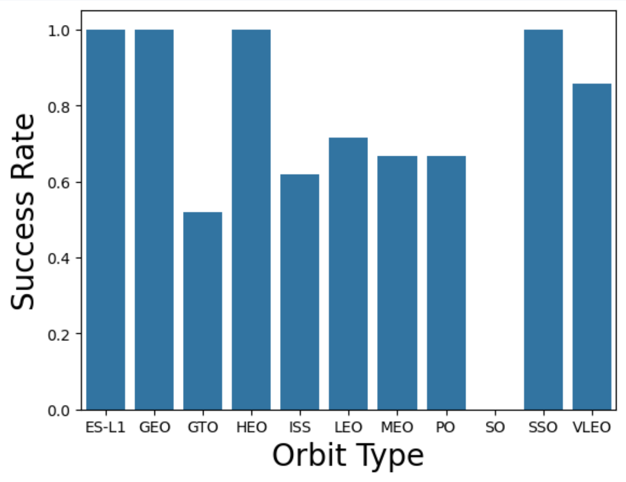
https://github.com/pjwillcocks/Data-Science-Capstone-Project-/blob/9acdcea36d7045edb90a573b68ba1ed42a075b9f/3%20labs-jupyter-spacex-Data%20wrangling.ipynb

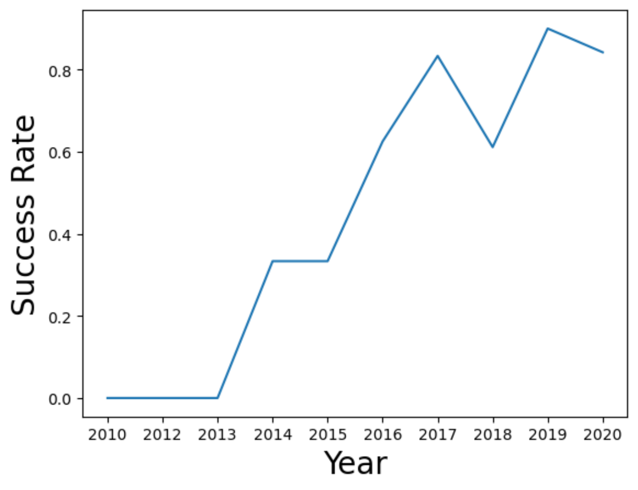
EDA with Data Visualization

We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend

https://github.com/pjwillcocks/Data-Science-Capstone-Project-/blob/9acdcea36d7045edb90a573b68ba1ed42a075b9f/5%20edadataviz.ipynb







EDA with SQL

- We applied EDA with SQL to get insight from the data. We wrote queries to establish elements such as the following:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.

- https://github.com/pjwillcocks/Data-Science-Capstone-Project-/blob/48851813f85aa047d3ae0a337874041cd807562a/4%20jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distances between a launch site to its proximities. We answered some question for instance:
  - Are launch sites near railways, highways and coastlines.
  - Do launch sites keep certain distance away from cities.

- https://github.com/pjwillcocks/Data-Science-Capstone-Project-/blob/9acdcea36d7045edb90a573b68ba1ed42a075b9f/6%20lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- This combination allowed to quickly analyze the relation between payloads and launch sites, helping to identify where is best place to launch according to payloads.

https://github.com/pjwillcocks/Data-Science-Capstone-Project-/blob/9acdcea36d7045edb90a573b68ba1ed42a075b9f/7%20spacex_dash_app.py

Predictive Analysis (Classification)

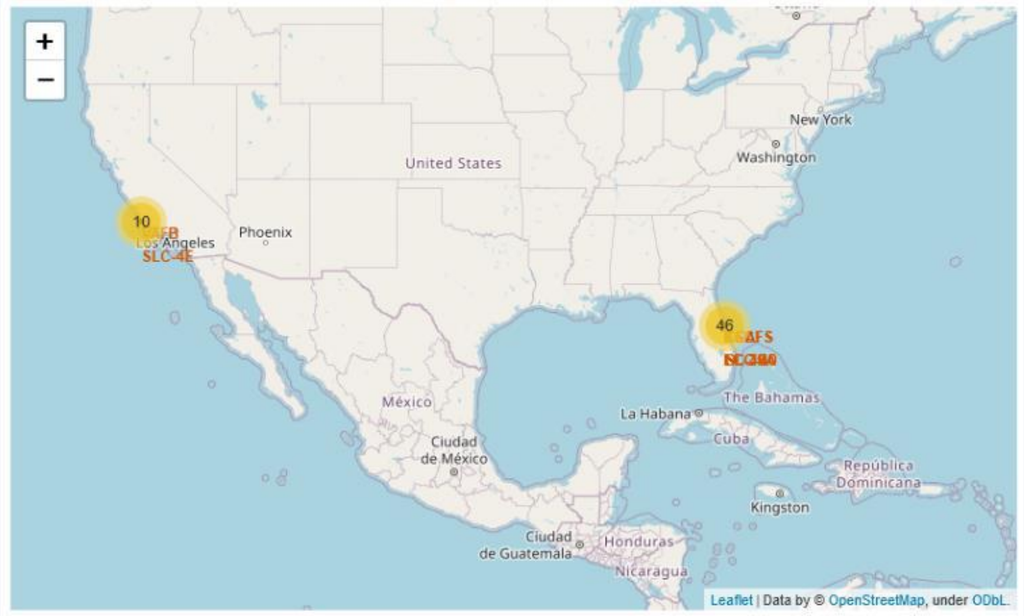
- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model
- https://github.com/pjwillcocks/Data-Science-Capstone-Project-/blob/9acdcea36d7045edb90a573b68ba1ed42a075b9f/8%20SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- Exploratory data analysis results
    - Space X uses 4 different launch sites;
    - The first launches were done to Space X itself and NASA;
    - The average payload of F9 v1.1 booster is 2,928 kg;
    - The first success landing outcome happened in 2015 fiver year after the first launch;
    - Many Falcon 9 booster versions were successful at landing in drone ships having payload above the average;
    - Almost 100% of mission outcomes were successful;
    - Two booster versions failed at landing in drone ships in 2015: F9 v1.1 B1012 and F9 v1.1 B1015;
    - The number of landing outcomes became as better as years passed.

Results

- Using interactive analytics was possible to identify that launch sites use to be in safety places, near sea, for example and have a good logistic infrastructure around.
- Most launches happens at east cost launch sites.



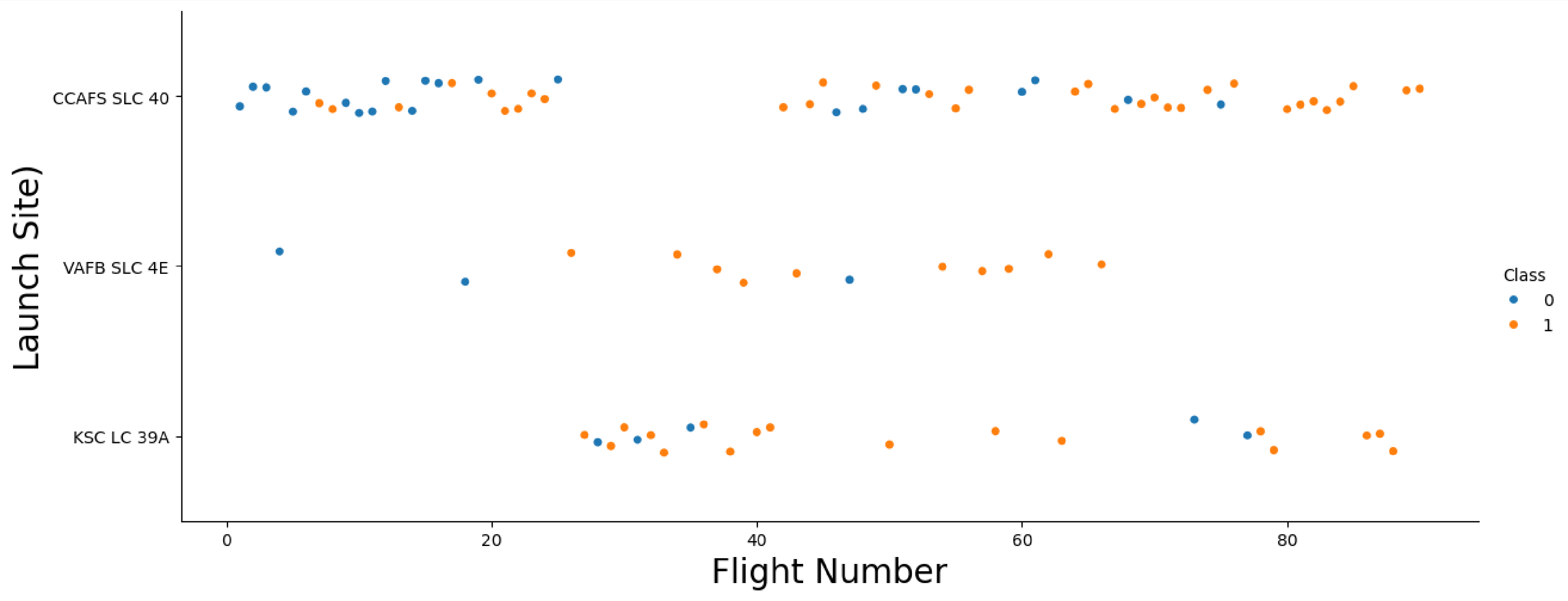
Results

- Predictive Analysis showed that Decision Tree Classifier is the best model to predict successful landings, having accuracy over 87% and accuracy for test data over 94%.

Section 2

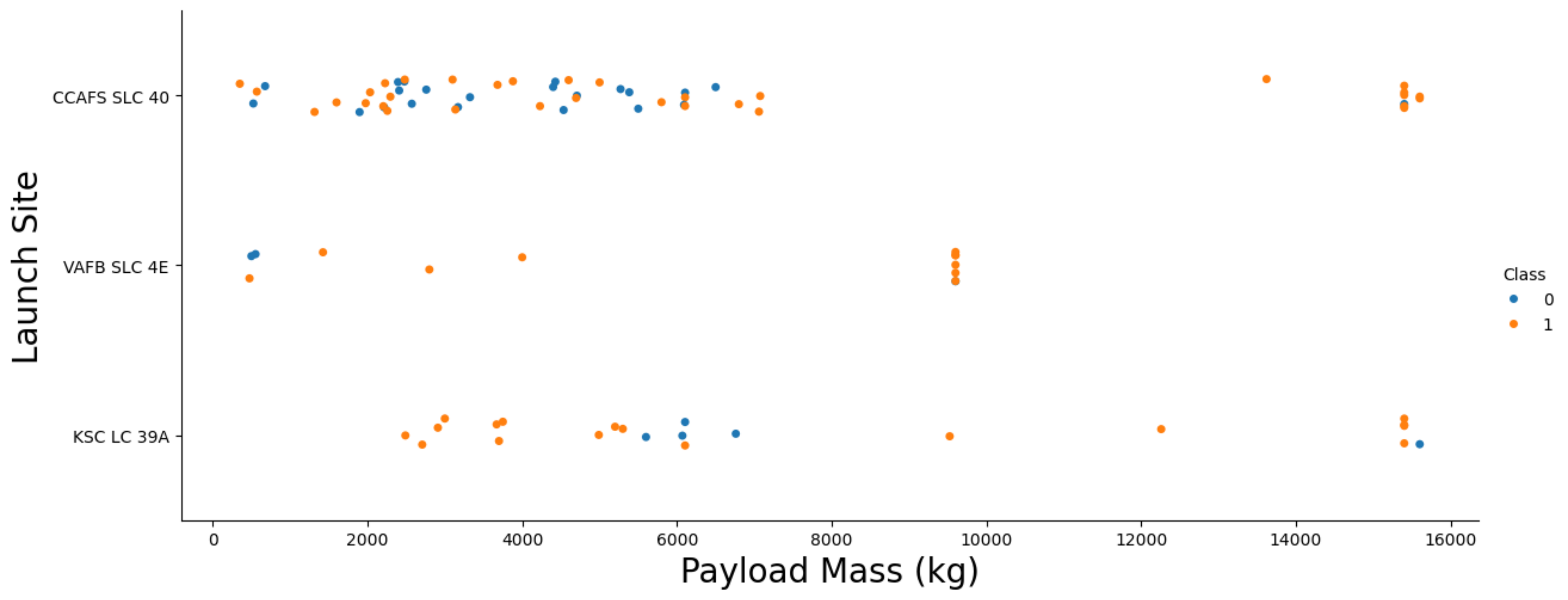
Insights drawn from EDA

Flight Number vs. Launch Site



- According to the plot above, it's possible to determine that the most frequently used launch site currently is CCAF5 SLC 40, and most of recent launches were successful;
- In second place VAFB SLC 4E and third place KSC LC 39A;
- It's also possible to see that the general success rate improved over time.

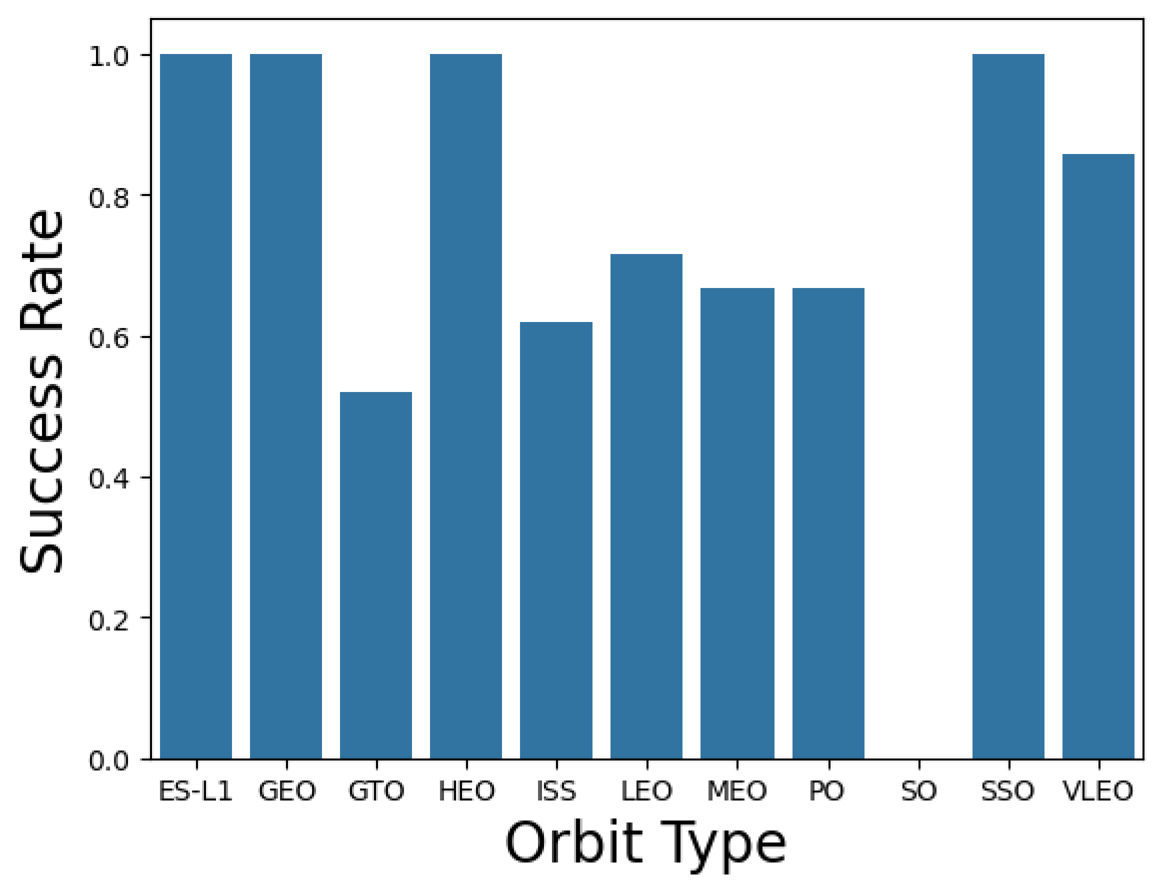
Payload vs. Launch Site



- Majority of payloads are under 8000kg
- Payloads over 9,000kg though have high success rate;
- Payloads over 10,000kg seems only to be possible from CCAFS SLC 40 and KSC LC 39A launch sites.

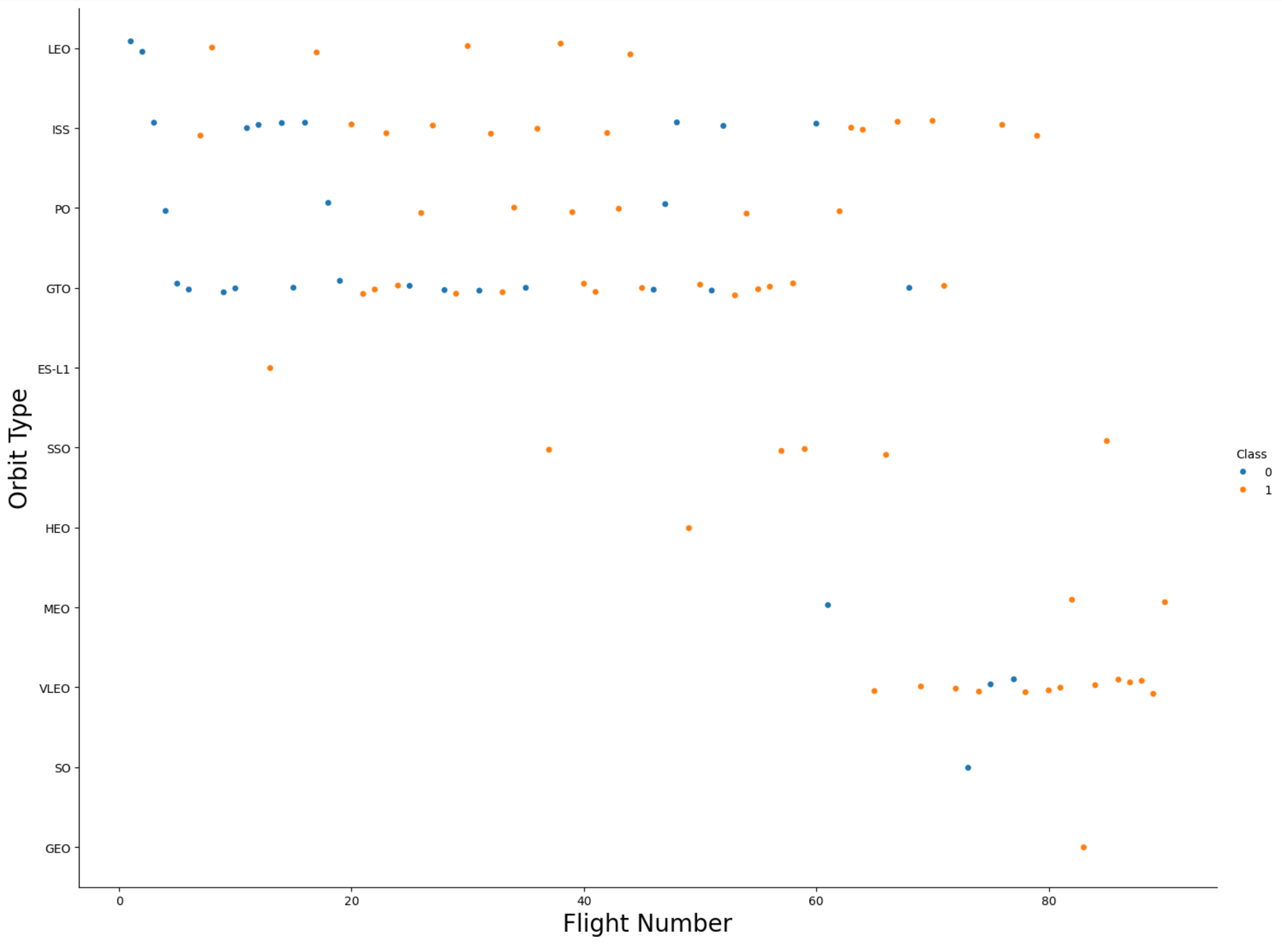
Success Rate vs. Orbit Type

- The following orbits have 100% success rate:
  - ES-L1
  - GEO
  - HEO
  - SSO
- Followed by:
  - VLEO (above 80)
  - LFO (above 70%)

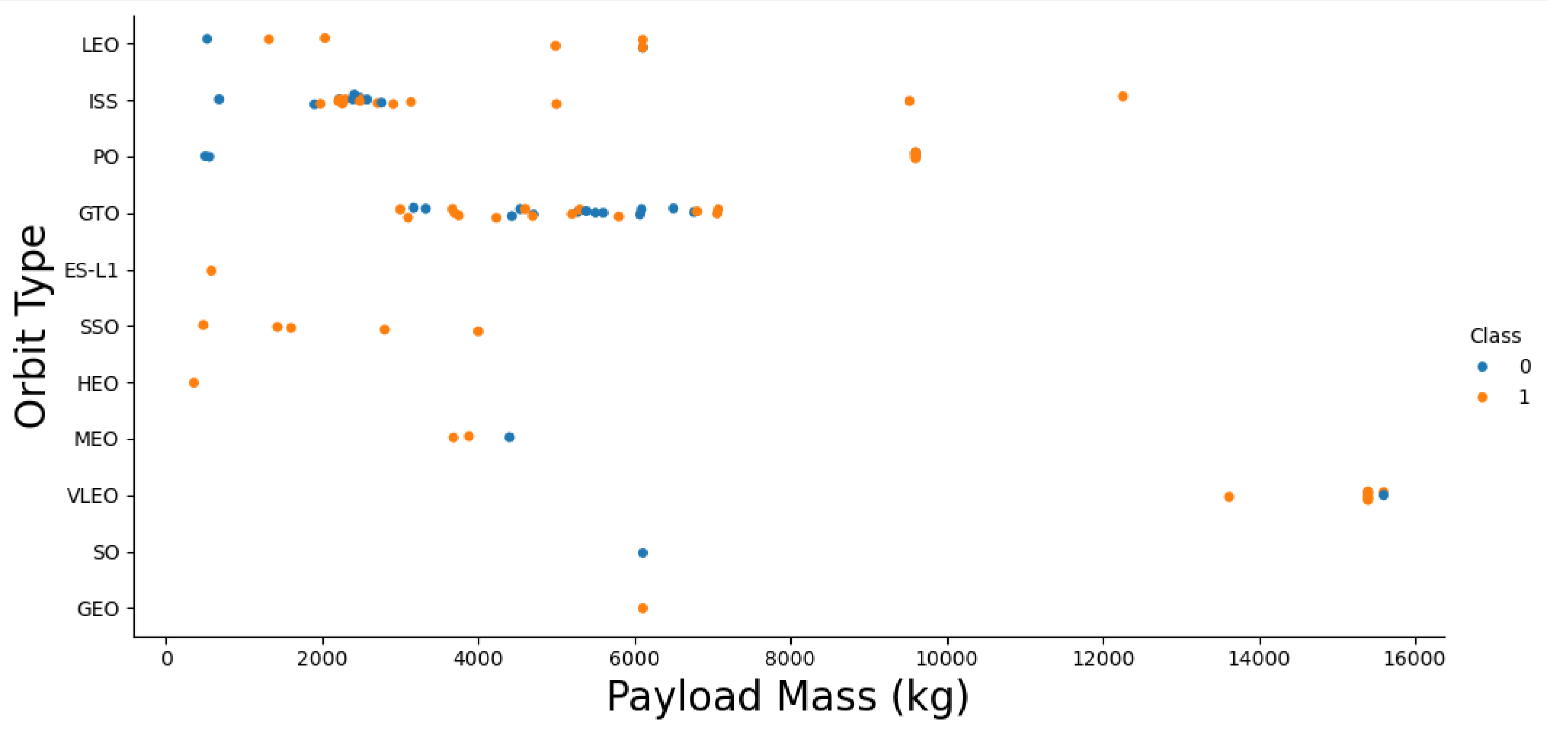


Flight Number vs. Orbit Type

- Success rate appears to have improved over time to all orbits
- The launches to different orbits changes over time with the VLEO now the most popular orbit for launch



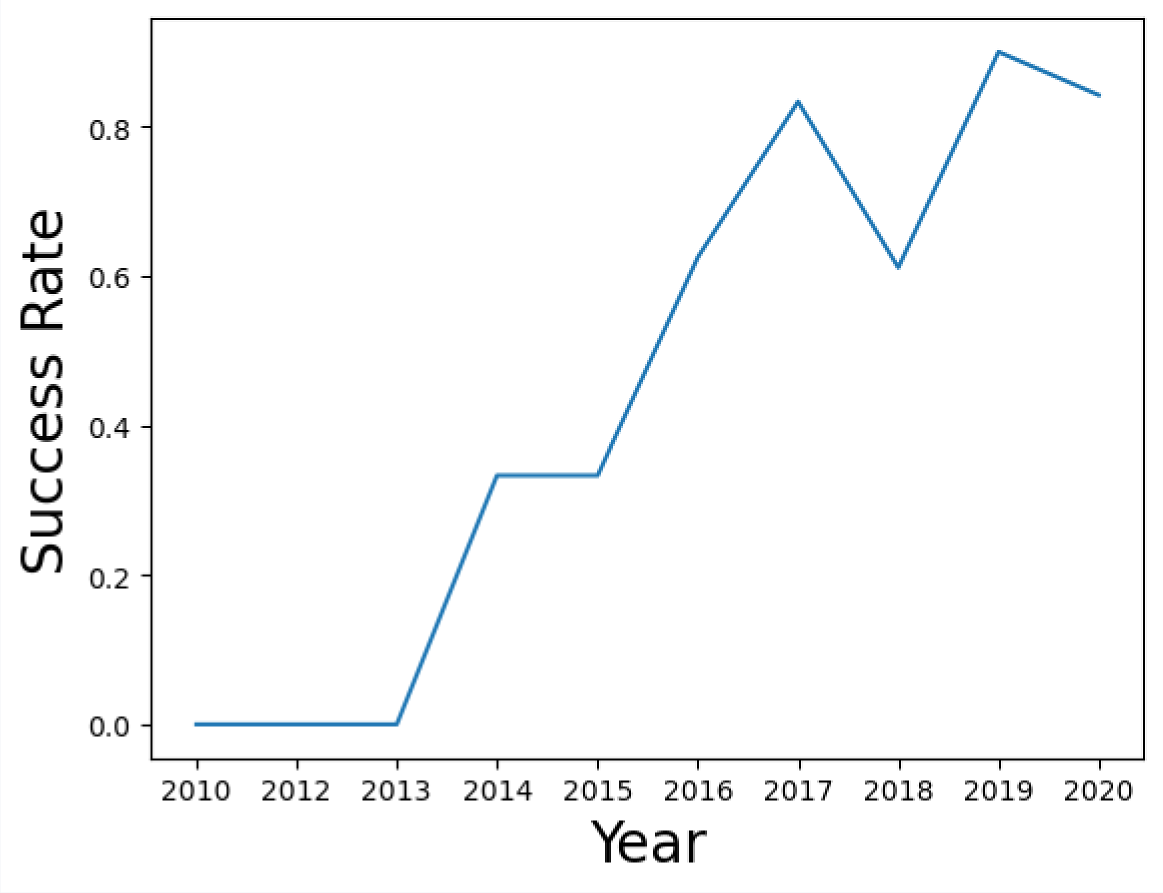
Payload vs. Orbit Type



- Seemingly there is no clear relation between payload and success rate to orbit GTO
- ISS orbit has the widest range of payload and a good rate of success
- There are few launches to the orbits SO and GEO
- The highest payloads are launched to VLEO orbit

Launch Success Yearly Trend

- Success rate has seen a general sharp increase over time
- Decreases in success rate however occurred in 2018 and 2020



All Launch Site Names

- %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE
- Using sqlite3 and line magic (%sql) an SQL query was directly written into the python notebook to extract distinct launch sites from the connection to the spaceX table

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

Launch Site Names Begin with 'CCA'

- Sql query written in Python notebook to obtain the first 5 rows of the table where launch site column contains "CCA"

In [14]:
```
%sql SELECT * from SPACEXTABLE where Launch_Site like 'CCA%' limit 5;
```

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

Out[14]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677

Total Payload Mass

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

```
In [16]: %sql SELECT sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer='NASA (CRS)';

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

Out[16]:

sum(PAYLOAD_MASS__KG_)
45596

- SQL query sum column PAYLOAD_MASS__KG_ to give the total payload launched from all boosters launched by NASA (CRS)

Average Payload Mass by F9 v1.1

Task 4

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

```
In [18]: %sql SELECT avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version='F9 v1.1';

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

Out[18]:

avg(PAYLOAD_MASS__KG_)
2928.4

- SQL query to show the average payload by mass carried by the F9 v1.1 booster

First Successful Ground Landing Date

```
In [20]: %sql SELECT min(Date) from SPACEXTABLE where Landing_Outcome='Success (ground pad)';

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

Out[20]:

min(Date)
2015-12-22

- SQL query to show the first successful landing on the ground pad

Successful Drone Ship Landing with Payload between 4000 and 6000

In [22]:

```
%sql SELECT distinct Booster_Version from SPACEXTABLE where Landing_Outcome='Success (drone ship)' and PAYLOAD_MASS__KG_ bet
```

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

Out[22]:

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

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

Total Number of Successful and Failure Mission Outcomes

In [23]:

```
%sql SELECT substr(Mission_Outcome,1,7) as Mission_Outcome, count(*) from SPACEXTABLE group by 1;
```

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

Out[23]:

Mission_Outcome	count(*)
Failure	1
Success	100

- SQL query the total number of successful and failure mission outcomes

Boosters Carried Maximum Payload

- SQL query showing the names of the booster which have carried the maximum payload mass

```
In [24]: %sql SELECT distinct Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXT
```

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

Out[24]:

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

```
In [37]: %sql SELECT substr(Date, 6,2) as Month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE where substr(Date, 0,
```

[No Title]

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

Out[37]:

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

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

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

In [38]:
```
%sql SELECT Landing_Outcome, count(*) from SPACEXTABLE where Date between '2011-06-04' and '2017-03-20' group by Landing_Out
```

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

Out[38]:

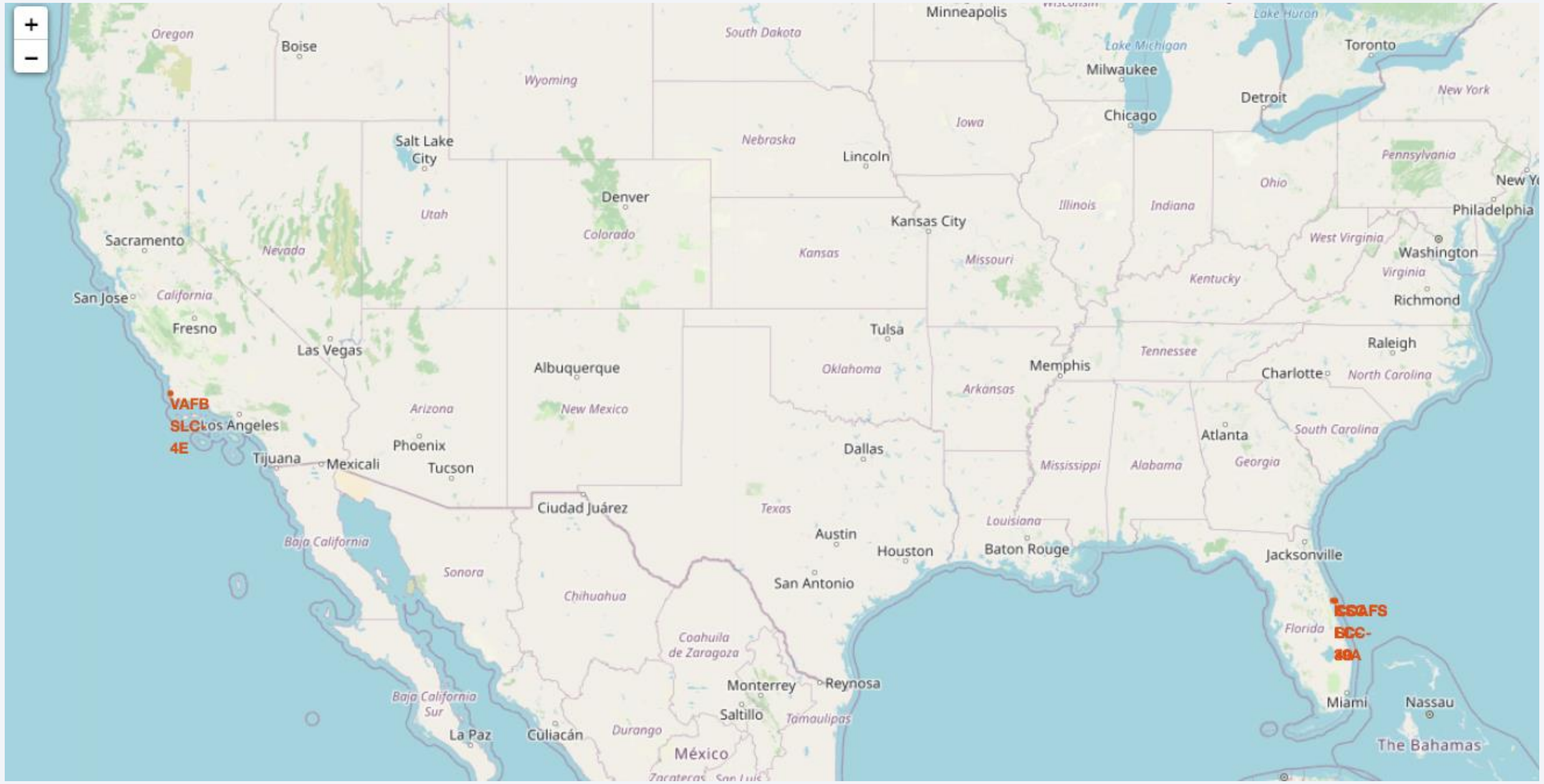
Landing_Outcome	count(*)
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1

- SQL query showing 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

Section 3

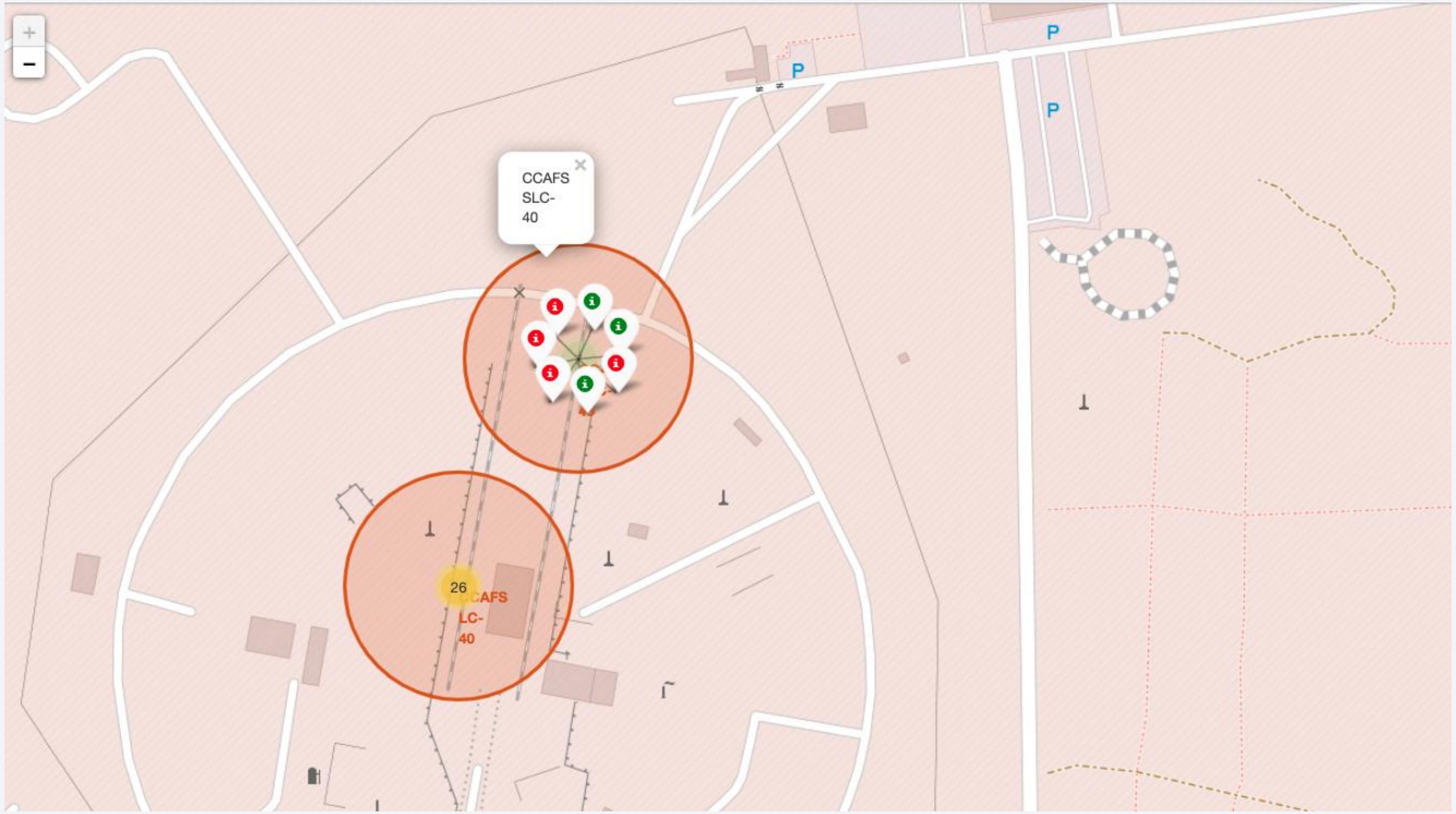
Launch Sites Proximities Analysis

All launch sites



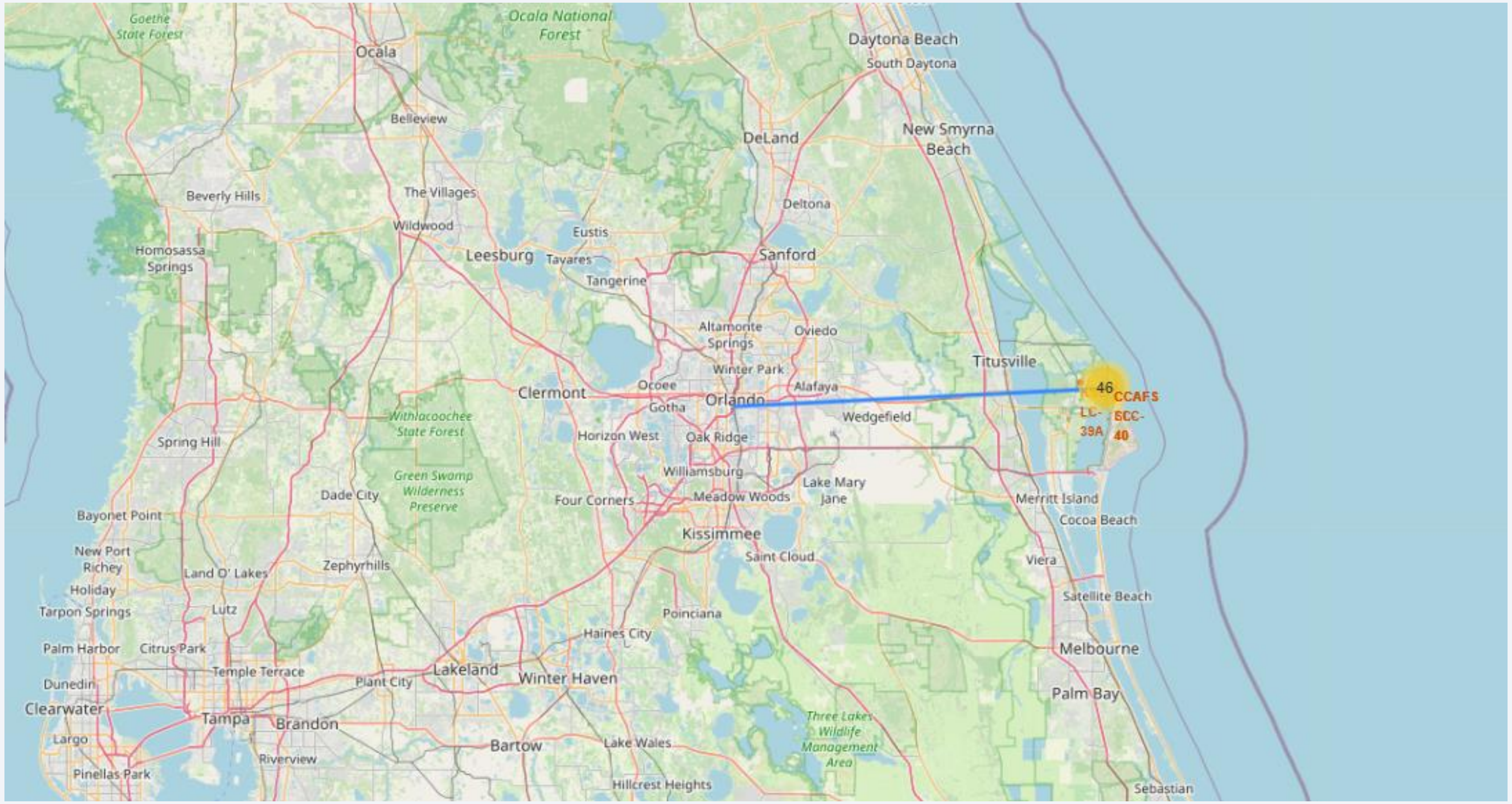
- Launch sites located on west and east coasts close to sea and nearby major conurbations

Launch Outcomes by Site



- Drilled down view of map including markers showing launch outcomes at each site

Launch site distance to major cities

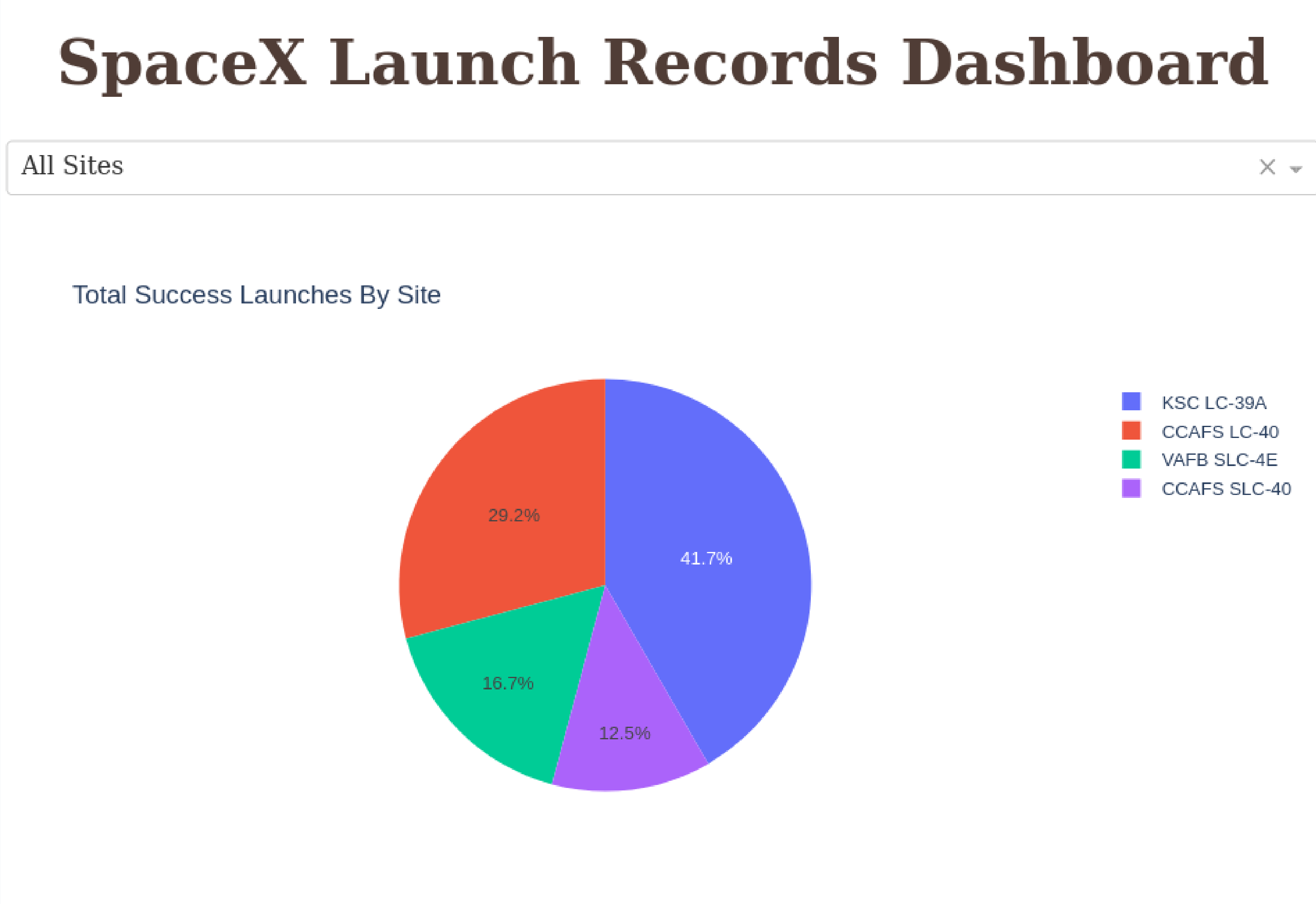


- Drilled down view showing line demonstrating distance from launch site to nearest city

Section 4

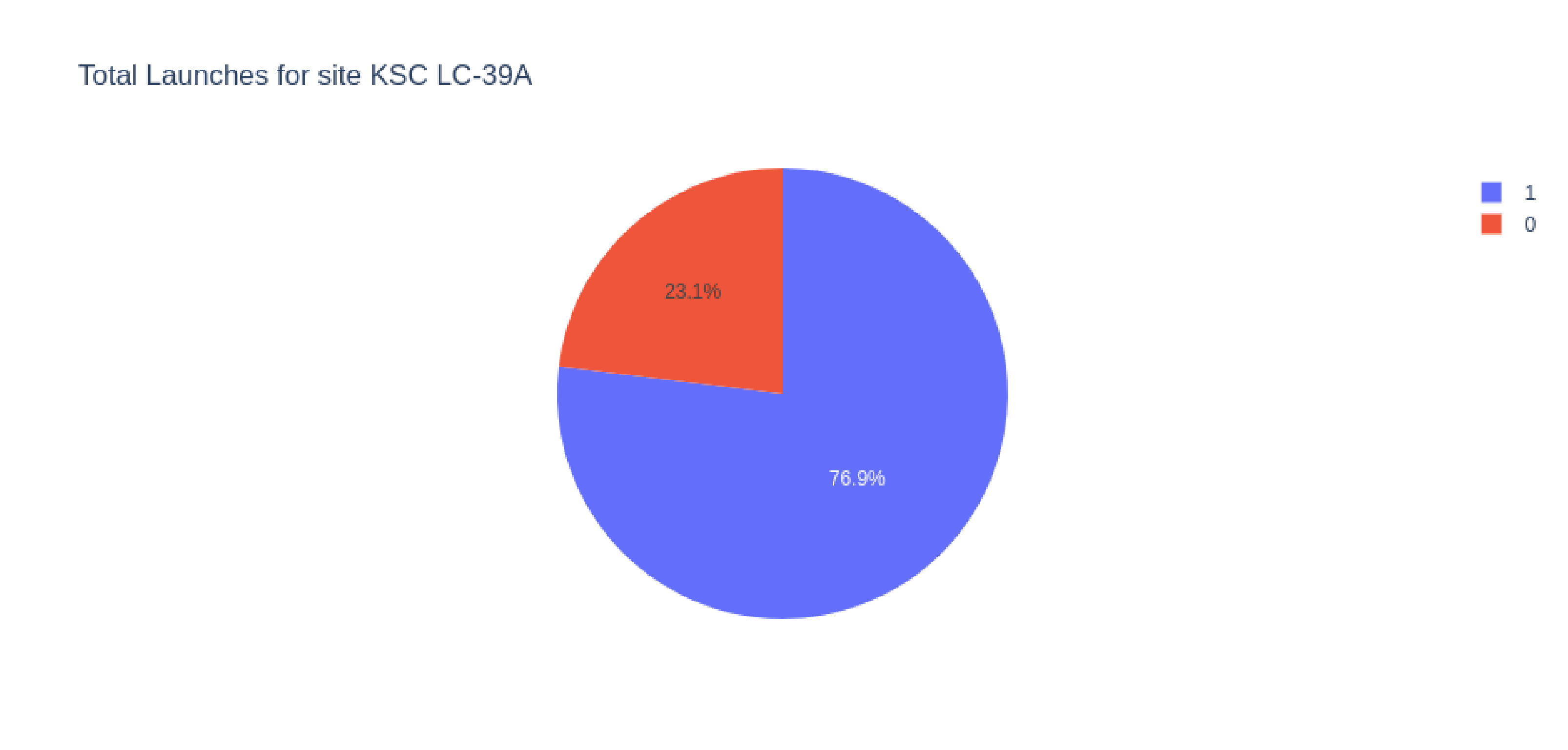
Build a Dashboard with Plotly Dash

Successful Launches by Site



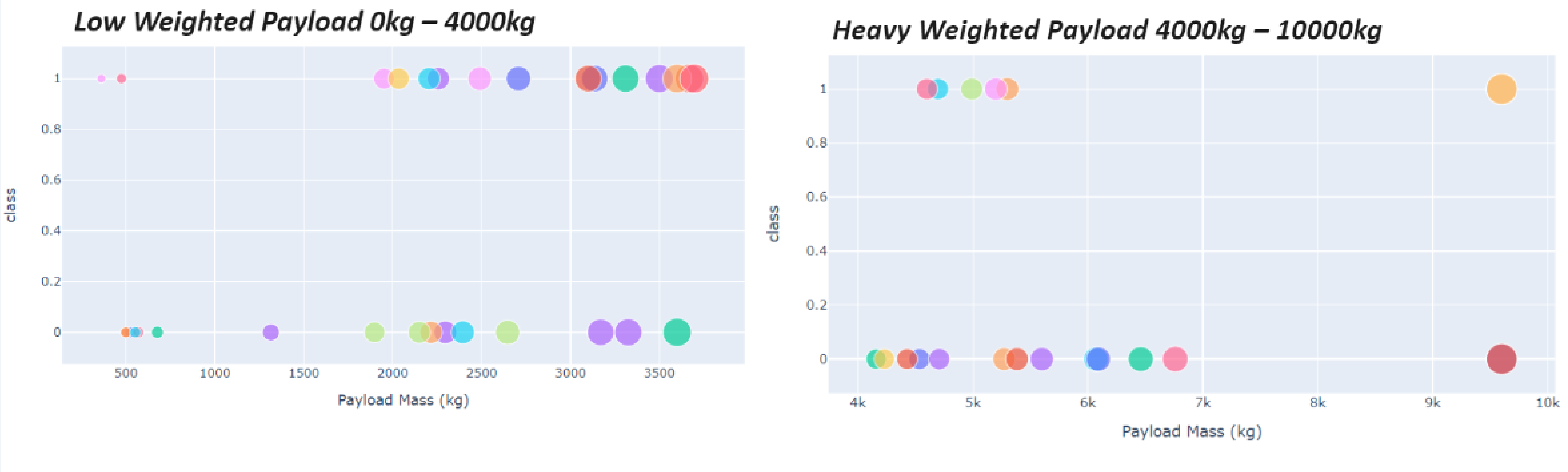
- KSC LC-39A has the highest number of successful launches with CCAFS SLC-40 the least and significantly lower

Site with Highest Launch Success Ratio



- Site KSC LC-39A had the highest success ratio of all sites at 76.9%

Success and Payload



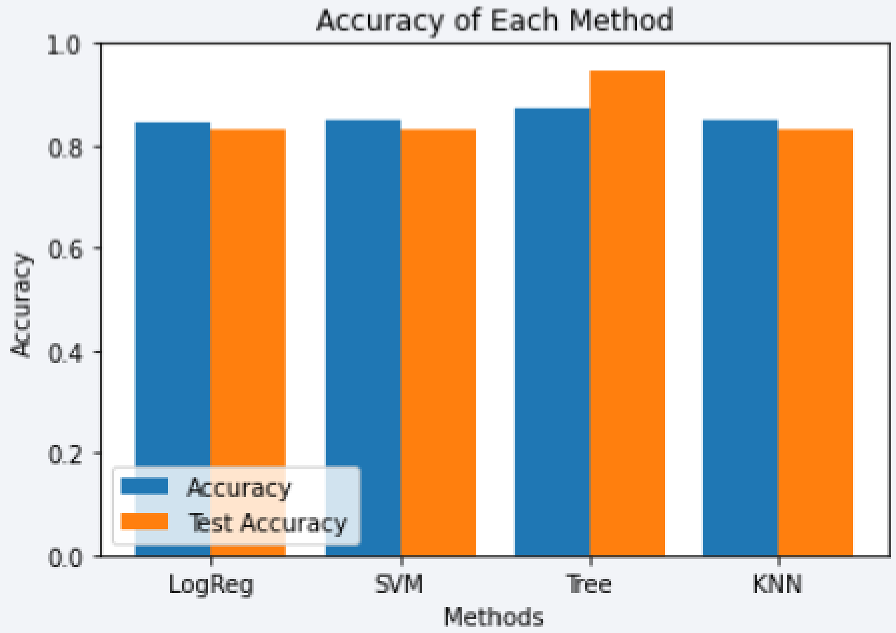
- Success rate for low weighted payloads are higher than heavy payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The model with the highest classification accuracy is Decision Tree Classifier, which has accuracies over than 87%.



Confusion Matrix

- Confusion matrix of Decision Tree Classifier proves its accuracy by showing the big numbers of true positive and true negative compared to the false ones.

Conclusions

- The best launch site is KSC LC-39A
- Launches above 7,000kg are less risky
- Although most of mission outcomes are successful, successful landing outcomes seem to have improved over time.
- Decision Tree Classifier is the best model that can be used to predict successful landings and increase profits.

Appendix

- As an improvement for model tests, it's important to set a value to `np.random.seed` variable;
- Folium didn't show maps on Github, so I took screenshots.

Thank you!