Winning Space Race with Data Science

Ashutosh Jha
24 September 2022

Outline

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

Executive Summary

- In the following report, we collect data from various sources, using different techniques. We clean and format the data to prepare for modelling. We perform exploratory analysis to gauge correlated features and relationship with target variable. We have created interactive data visualization dashboards and other plots which help us identify relevancy order for different feature variables for prediction task. At the end we model the data with different classification algorithms and select the best performing based on evaluation metrics like accuracy score and confusion matrix.
- We find that decision Tree algorithm best predicts the target variable. We are able to see the rules used which helps with explanation of model prediction in such a high cost value question.

Introduction

- 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.
- With Past Launch Data from different sources, We are trying to come up with a reliable model to predict whether a launch will land or not. This can help an alternate company to bid against SpaceX for rocket launches.

Section 1

Methodology

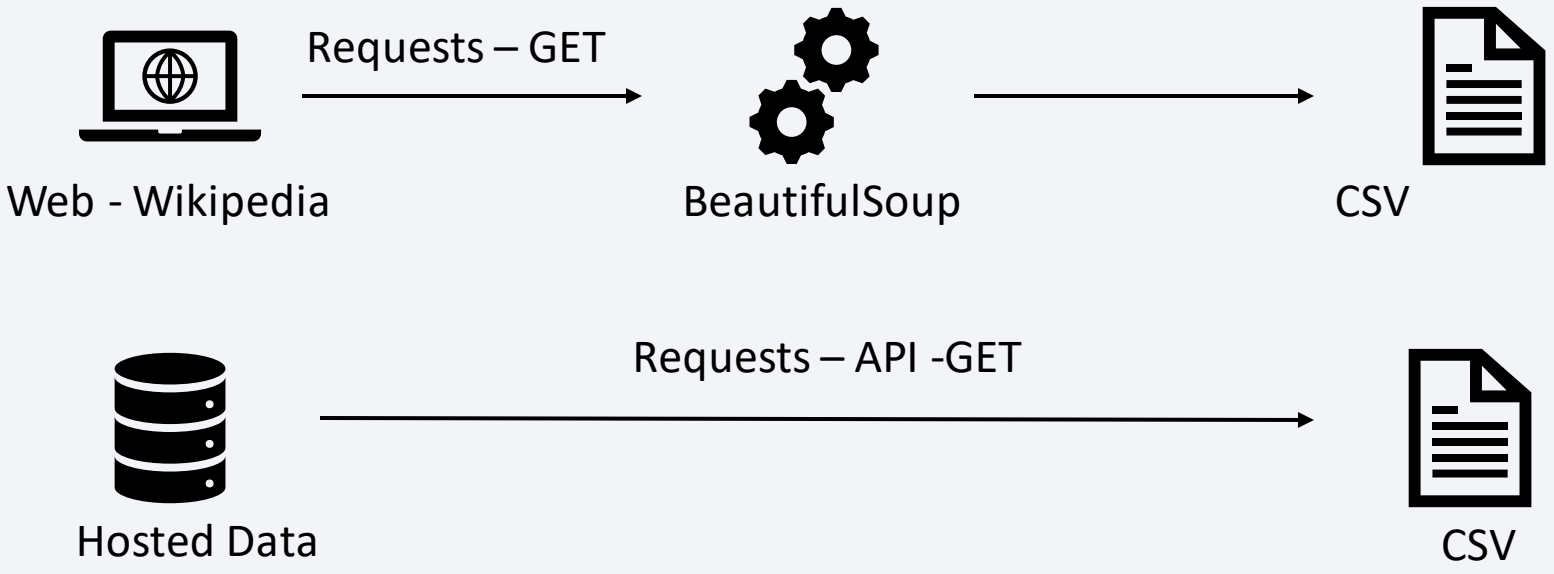
Methodology

Executive Summary

- Data collection methodology:
  - The Data was collected from SpaceX public API and with web scraping Wikipedia tables regarding Falcon9/Heavy Launch Records.
- Perform data wrangling
  - Target Column converted to Categorical Values, data cleaning (dealing with Null values) and analyzing launch distribution among feature column unique values.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Apply different Classification models, logistic regression, k-nearest neighbors, support vector machines and decision trees, find best parameters with GridSearch and best model.

Data Collection

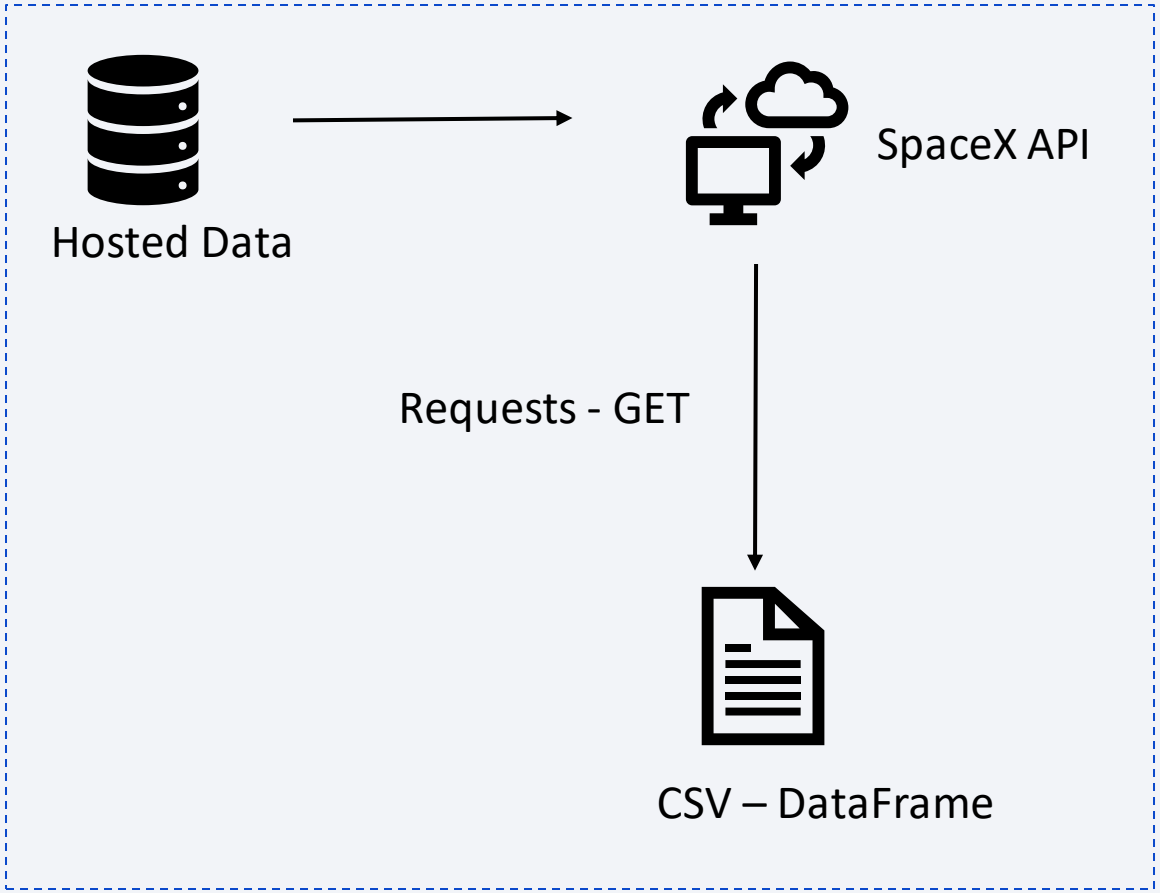
- The Data was collected with SpaceX public APIs and with web scraping HTML tables on Wikipedia page "Falcon 9 and Heavy Launches". The python library used for APIs GET request was "requests" and for web scraping it was "BeautifulSoup" and "requests".



Data Collection – SpaceX API

- REST GET calls were implemented using python library requests to fetch data from the SpaceX API.

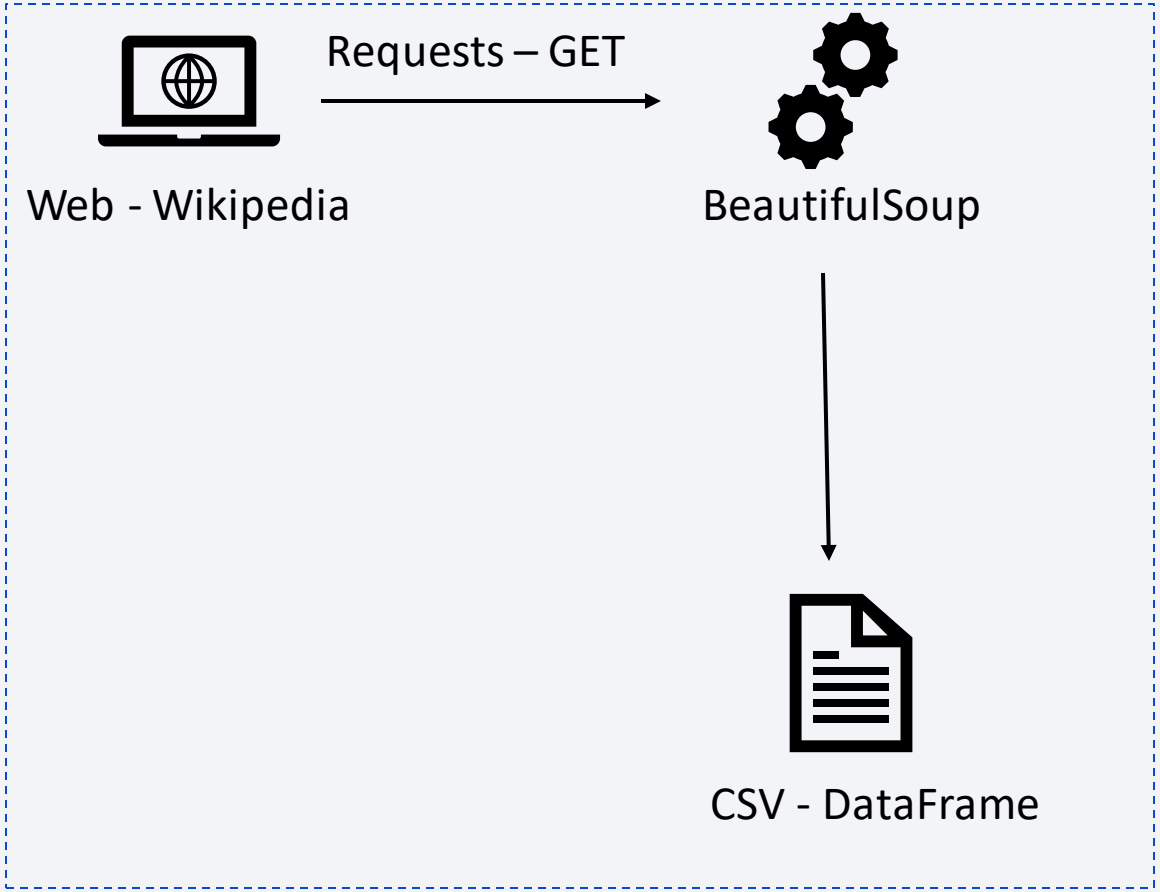
- GitHub URL for Jupyter notebook illustrating the Data Collection with SpaceX API Work - Data Collection - SpaceX API



Data Collection - Scraping

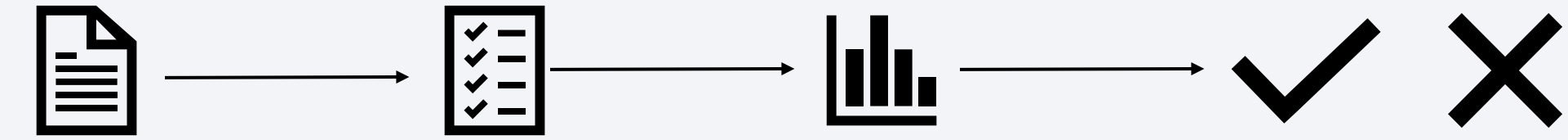
- Data was also collected with Requests GET API call to get HTML response of Wikipedia page "Falcon 9 and Heavy Launches". Then Beautiful Soup library was used to read tables and contents of the response.

- GitHub URL - Data Collection - Web Scraping



Data Wrangling

- After Loading the data, it was analyzed for NULL values and the data types of its columns. This would help clean the data where required.
- Then Categorical Columns like "Launch Site" and "Orbit" were analyzed to observe the distribution of data. Finally the target column was processed to convert its categorical values to classes.
- GitHub URL - Data Wrangling



EDA with Data Visualization

- Scatter Plots were Used for analysis of relation between feature variables like Payload and launch site, Payload mass and launch site etc. These charts helped visualize correlation between features which in turn helped identify important features for final prediction.
- Also Used Bar plots to visualize success rate with orbit type and line chart to view how the success rate has changed over the years.
- Finally, converted categorical variables with One-hot encoding and cast the data frame columns to float type.
- GitHub URL - EDA with Data Visualization

EDA with SQL

- Performed EDA with SQL using Jupyter magic commands.
- Extracted Unique Launch Sites, Launch sites having names starting with 'CCA'
- Payload mass where customer was NASA (CRS) or rocket version was F9 v1.1
- Earliest date with sucessful ground pad landing, and successful drone ship landing with payload mass between 4 and 5 ton.
- Group with distinct landing outcome, booster versions which have carried maximum payload. Failed outcome on drone ships in 2015 and ordered landing outcome between two dates.
- GitHub URL - EDA With SQL

Build an Interactive Map with Folium

- We added Circular markers in a marker cluster object to show different launches in our data set. We also plotted lines to show how close/far the launch site is from Coast line. We also labelled our markers.
- We added the objects to show how the launch sites geographical location and associated successes/failures. And lines to show proximity to different entities on a map.
- GitHub URL - Data Visualization with Folium

Build a Dashboard with Plotly Dash

- We added a dropdown where we could choose the landing site, a pie chart to show success rates at all and different sites based on the dropdown. A slider to choose a range of payload mass and a scatter plot to show success/failure versus payload mass.
- We added the dropdown and slider to filter data for drilling down. We added Pie chart and scatter plot to visualize how launch site location and payload mass affect success/failure of landing outcome.
- GitHub URL - Dashboard with Plotly Dash

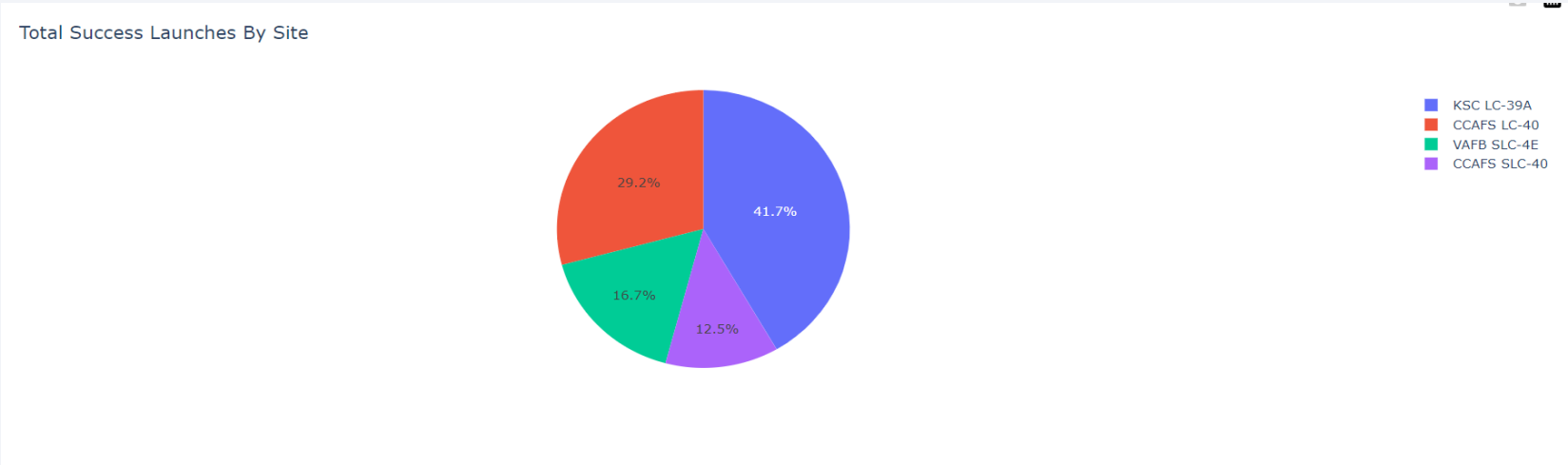
Predictive Analysis (Classification)

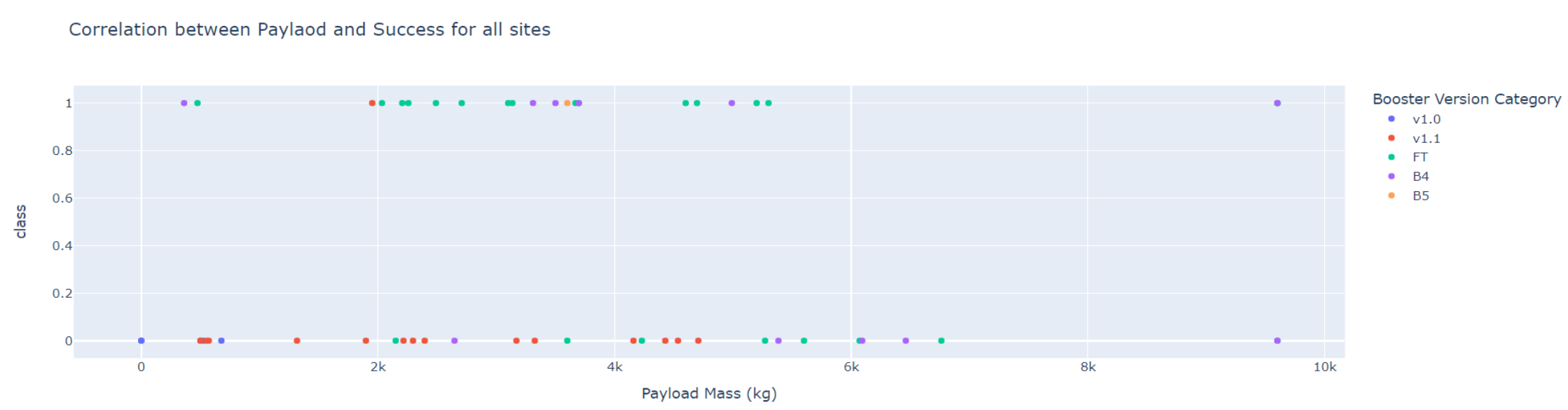
- We loaded the data into X and Y variables as feature and target variables as numpy array. Them we used standard scaler to scale individual columns to remove any weight bias. We then used several techniques like Logistic regression, K-nearest neighbors, Support Vector Machines and Decision trees with a set of parameters fed into a Grid Search object for each technique. This enabled us to find best parameters for each technique and then the best technique to use as a model.
- GitHub URL - Predictive Analysis

Results

- With EDA we observe different launch sites having different success rates and such is the case with different orbits. We see the VAFB SLC has the highest success rate, success rate increases with Flight number. Payload mass also directly affects the success rate. The GTO, ISS, LEO, MEO, PO and SO orbits have low success rates. The yearly trend suggests the success rate has improved YoY since 2013.
- We see that Decision Tree gives the highest score among the diff techniques, in our Grid Search, we get an average of 88.75% accuracy while we get 83.34% on test set alone.

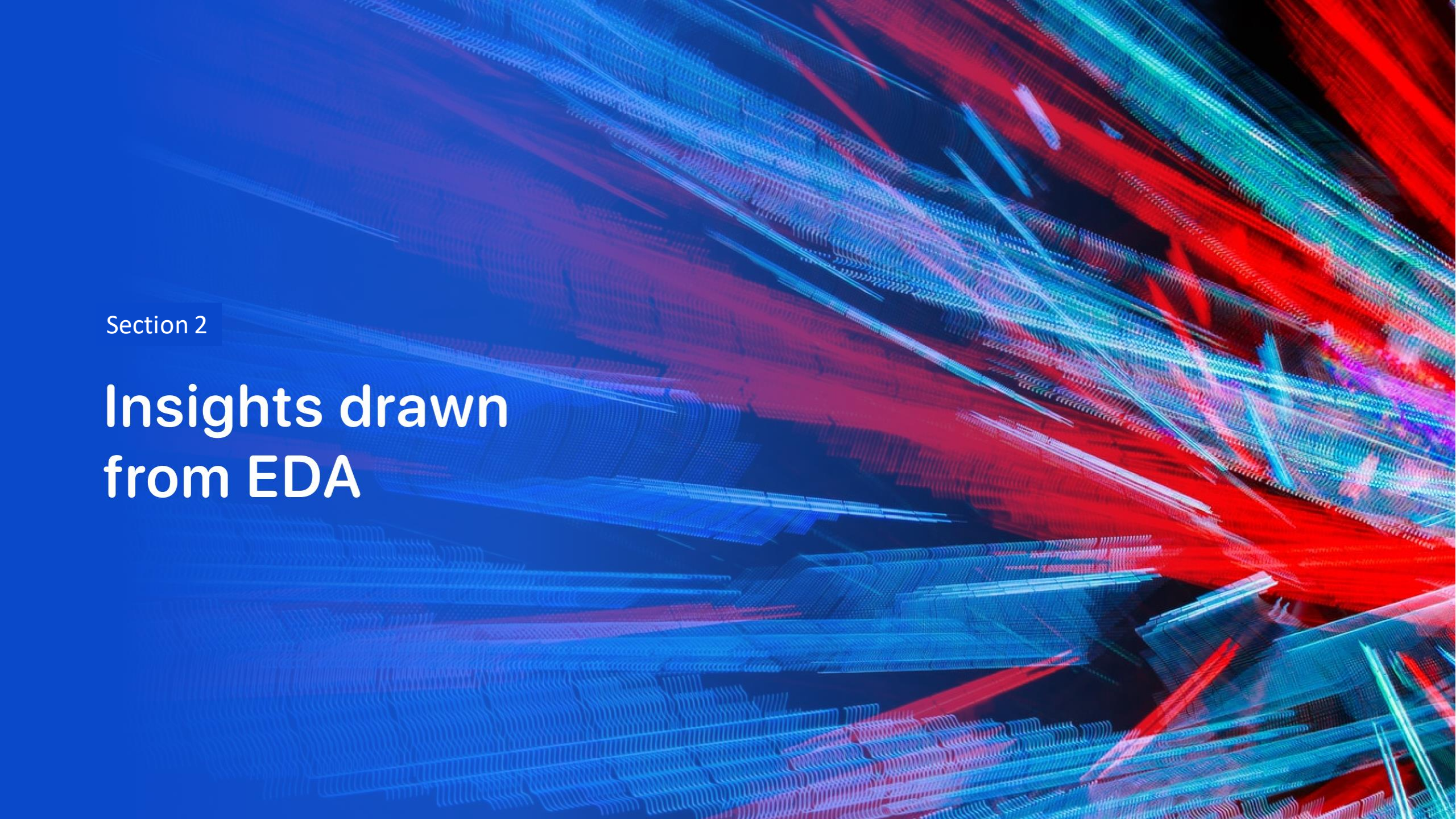
Results


Total Success Launches By Site
29.2%
41.7%
16.7%
12.5%
KSC LC-39A
CCAFS LC-40
VAFB SLC-4E
CCAFS SLC-40


Correlation between Paylaod and Success for all sites
class
1
0.8
0.6
0.4
0.2
0
0
2k
4k
6k
8k
10k
Payload Mass (kg)
Booster Version Category
v1.0
v1.1
FT
B4
B5

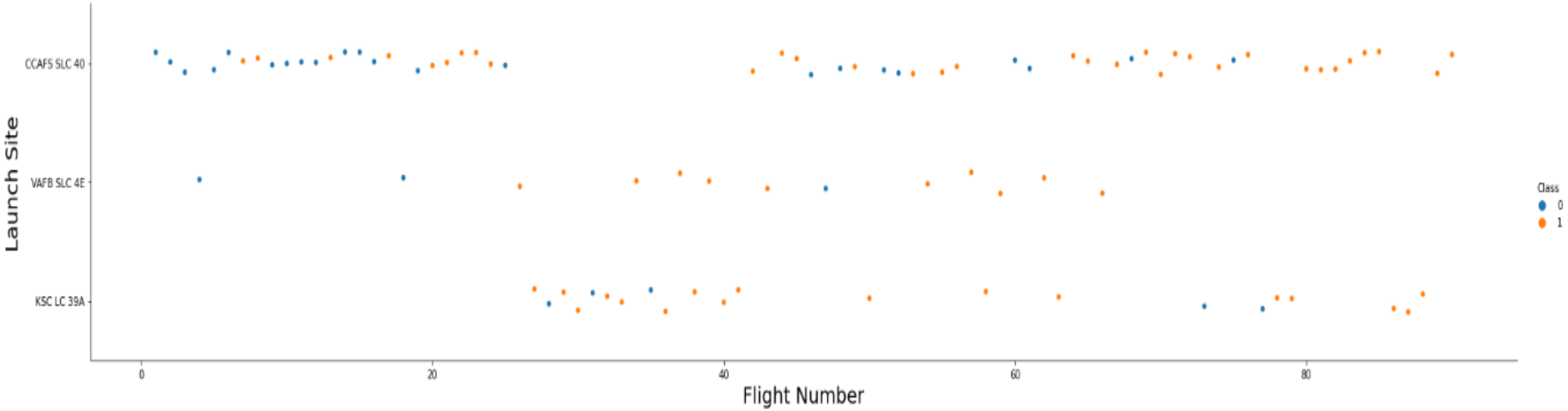
Section 2

Insights drawn from EDA



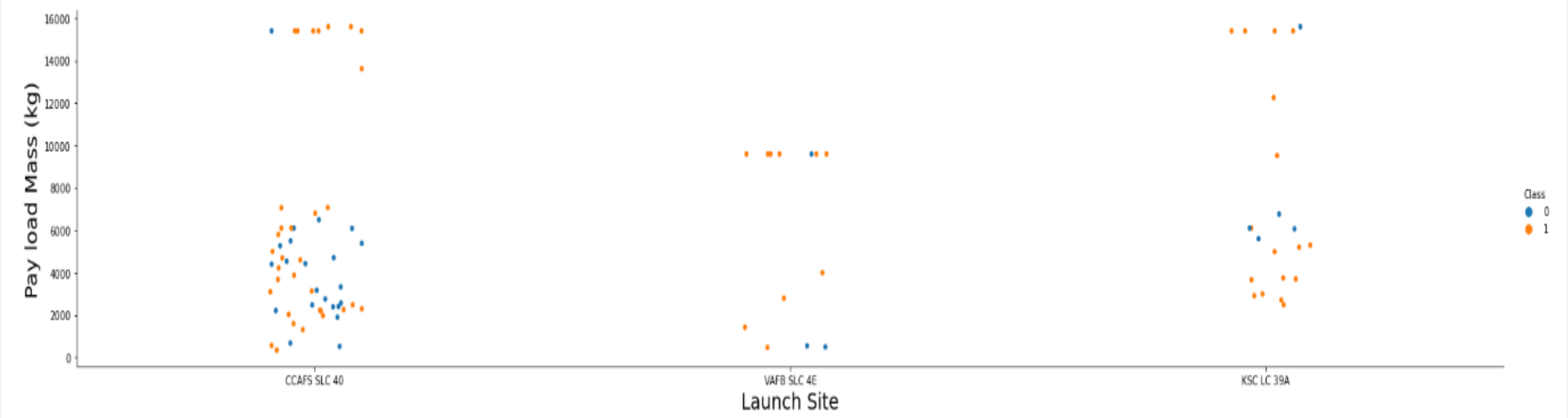
Flight Number vs. Launch Site

- For all three launch sites we see that the success rate increases as Flight number increases. There have been fewer VAFB and KSC based launces and KSC started after a while and VAFB has not been used since some time.



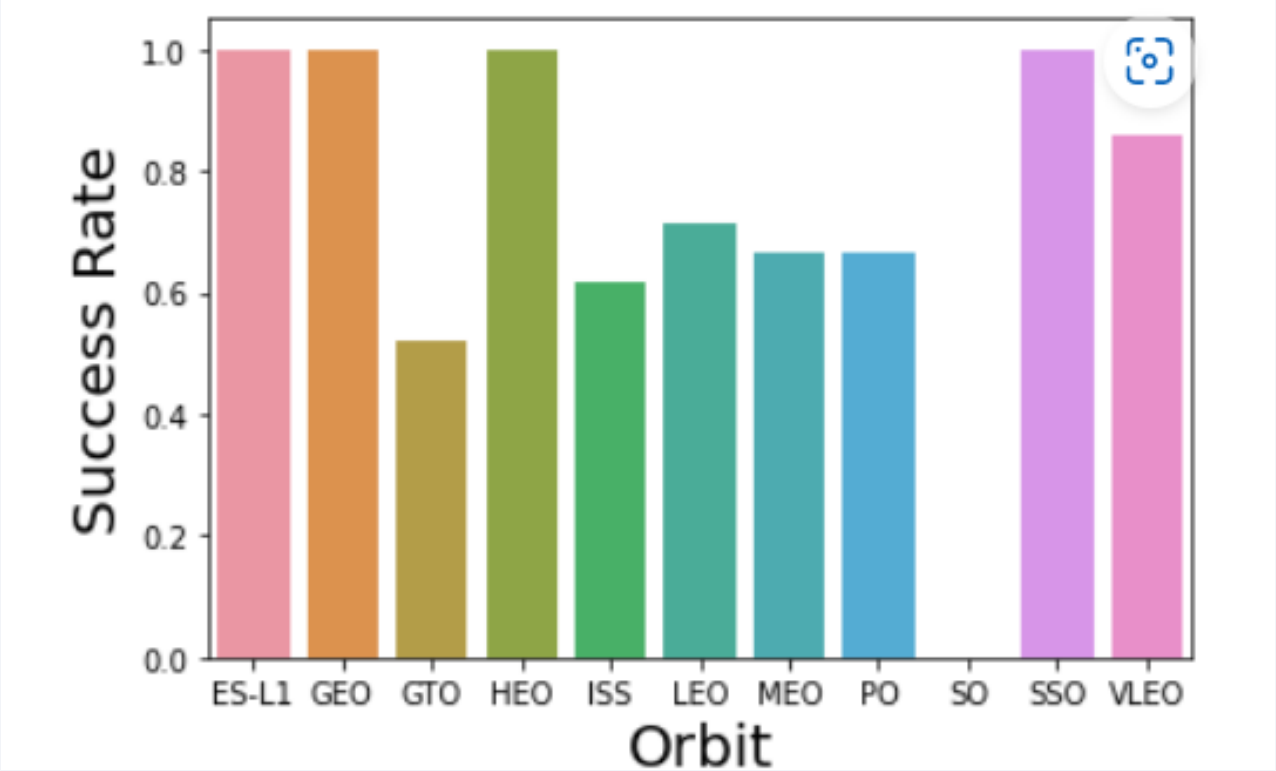
Payload vs. Launch Site

- For each launch site we see that greater payload, the success rate has been higher, VAFB does not have any payloads greater than 10,000KG. While it looks like above a certain threshold (10,000KG) the payload is identical among launches.



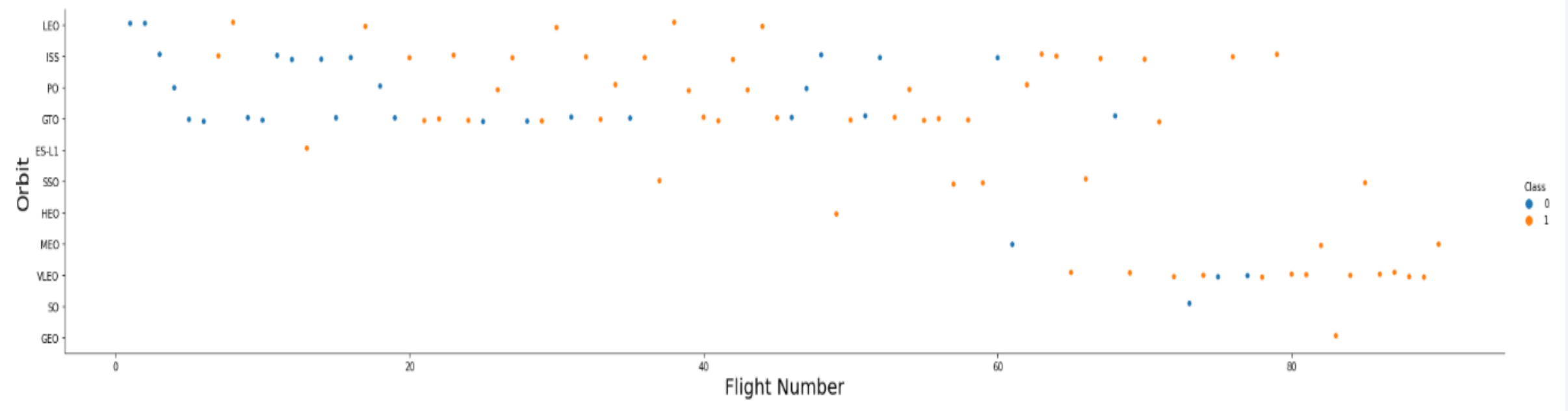
Success Rate vs. Orbit Type

- We observe that while ES-L1, GEO, HEO, SSO and VLEO have high success rate, GTO, ISS, LEO, MEO, PO have low success rate. This may be down to the sample size of each type of launch as well.



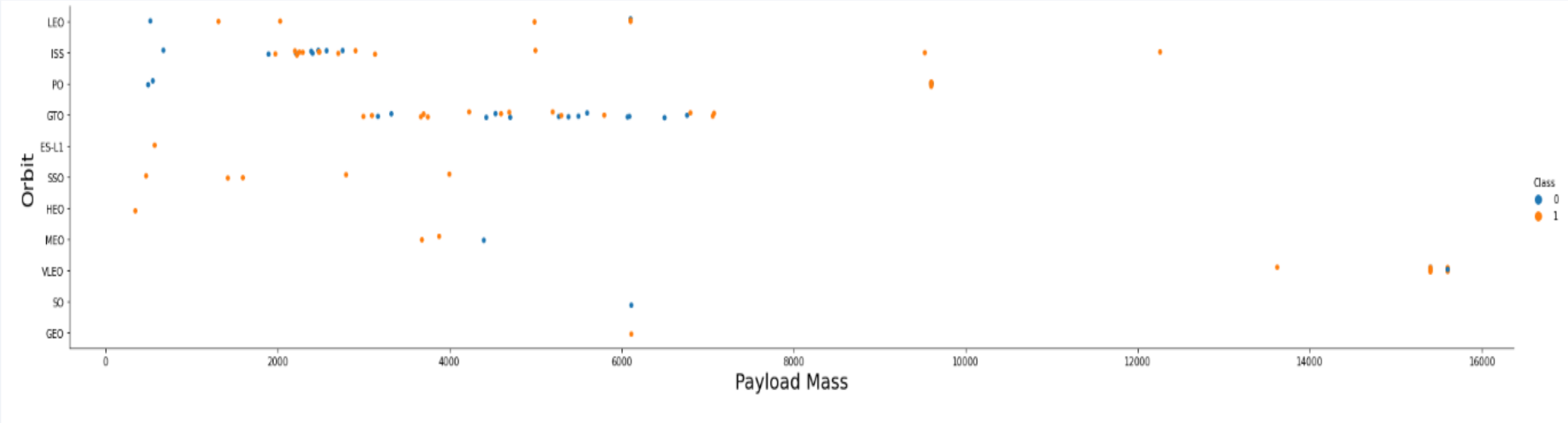
Flight Number vs. Orbit Type

- We observe that LEO, VLEO have some correlation with flight number, most orbits don't seem to have significant correlation with flight number.



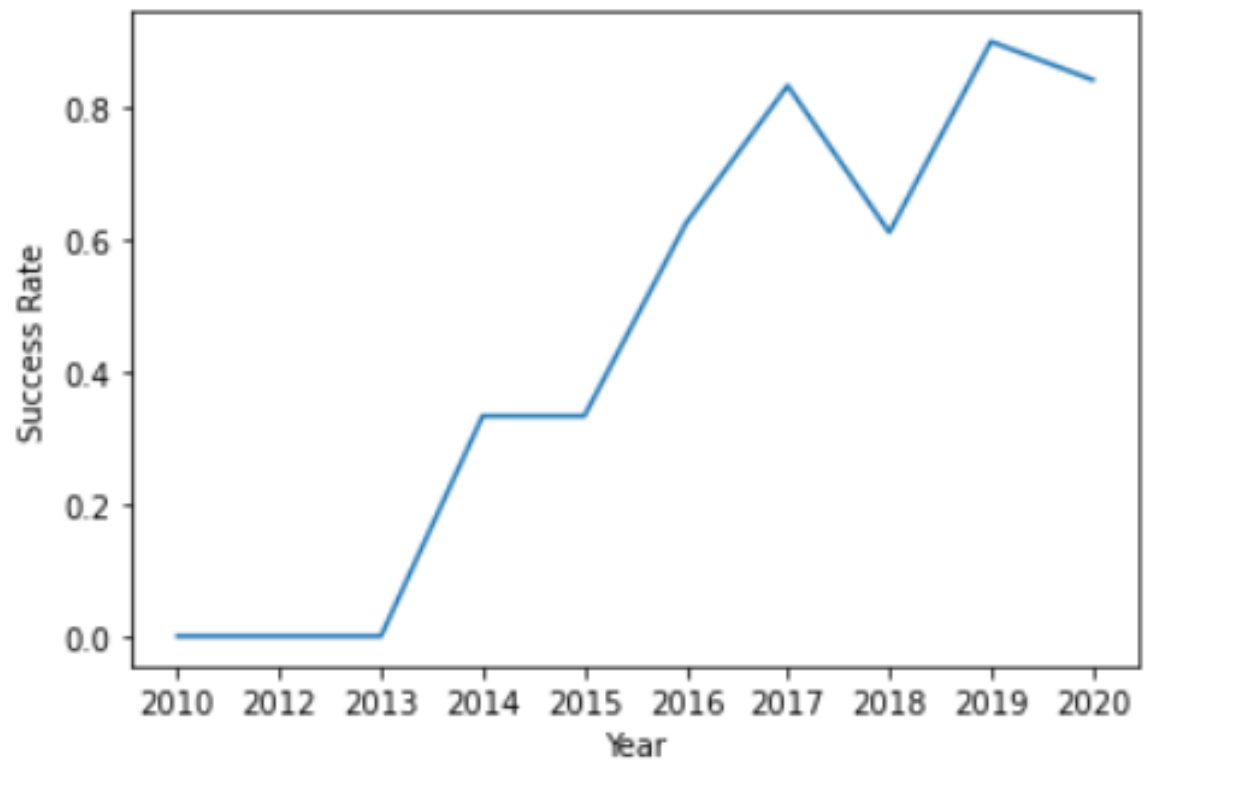
Payload vs. Orbit Type

- For PO, LEO and ISS the success rate increases with payload mass, while for GTO it seems to decrease. There seem to be only success datapoints for SSO.



Launch Success Yearly Trend

- We observe that the success rate has increased year on year since 2013, except only 2018 and 2020.



All Launch Site Names

- Distinct operator to get unique launch site values.

In [7]:
```
%%sql
SELECT DISTINCT(launch_site) FROM SPACEXDATASET;
```

```
 * ibm_db_sa://lyj27996:***@fbd88901-ebdb-4a4f-a32e-9822b9fb
Done.
```

Out[7]:

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

Launch Site Names Begin with 'CCA'

- Using wildcard '%' to get launch sites with name starting with 'CCA', limiting number of rows with limit operator.

In [10]:
```
%%sql
SELECT * FROM spacexdataset WHERE launch_site LIKE 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://lyj27996:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Out[10]:

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

- Using sum operator to get total mass carried for NASA (CRS) customer.

In [11]:
```
%%sql
SELECT sum(payload_mass__kg_) FROM spacexdataset WHERE customer='NASA (CRS)'
```

```
 * ibm_db_sa://lyj27996:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.a
Done.
```

Out[11]:

1
45596

Average Payload Mass by F9 v1.1

- Using AVG operator and filter on booster version to get avg payload mass by specified version.

In [12]:
```
%%sql
SELECT AVG(payload_mass__kg_) FROM spacexdataset WHERE booster_version LIKE 'F9 v1.1%';
```

```
 * ibm_db_sa://lyj27996:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.datab
Done.
```

Out[12]:

1
2534

First Successful Ground Landing Date

- Using Min operator and filter on landing outcome to get first date with success outcome.

In [14]:
```
%%sql
SELECT min(date) FROM spacexdataset WHERE landing__outcome = 'Success (ground pad)';
```

```
 * ibm_db_sa://lyj27996:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.da
Done.
```

Out[14]:

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Applying filters on landing outcome and range filter on payload mass to get rows between 4000 and 6000 only. Only distinct booster version values with distinct operator.

In [15]:
```
%%sql
SELECT DISTINCT(booster_version) FROM spacexdataset where landing__outcome = 'Success (drone ship)' AND payload_mass__kg_ > 4000 AND payload_mass__kg_
```

```
 * ibm_db_sa://lyj27996:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Out[15]:

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

Total Number of Successful and Failure Mission Outcomes

- Grouping by mission outcome and counting no of entries with each type.

In [22]:
```
%%sql
SELECT mission_outcome,COUNT(*) FROM spacexdataset GROUP BY mission_outcome;
```

```
 * ibm_db_sa://lyj27996:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.datab
Done.
```

Out[22]:

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

Boosters Carried Maximum Payload

- Using subquery in filter to get rows with max payload only.

In [26]:
```
%%sql
SELECT DISTINCT(booster_version) FROM spacexdataset WHERE payload_mass__kg_ = (SELECT max(payload_mass__kg_) FROM spacexdataset);
```

```
 * ibm_db_sa://lyj27996:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Out[26]:

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

- Filtering using landing outcome and date fields using wildcard to get all from 2015.

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

In [27]:
```
%%sql
SELECT landing__outcome, booster_version, launch_site FROM spacexdataset WHERE landing__outcome = 'Failure (drone ship)' AND DATE LIKE '2015%';
```

```
 * ibm_db_sa://lyj27996:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

Out[27]:

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

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

- Filter for Dates, grouping by landing outcome and ordering in descending order by count.

In [31]:

```
%%sql
SELECT landing__outcome, COUNT(*) FROM spacexdataset WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY landing__outcome ORDER BY 2 DESC;
```

```
 * ibm_db_sa://lyj27996:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

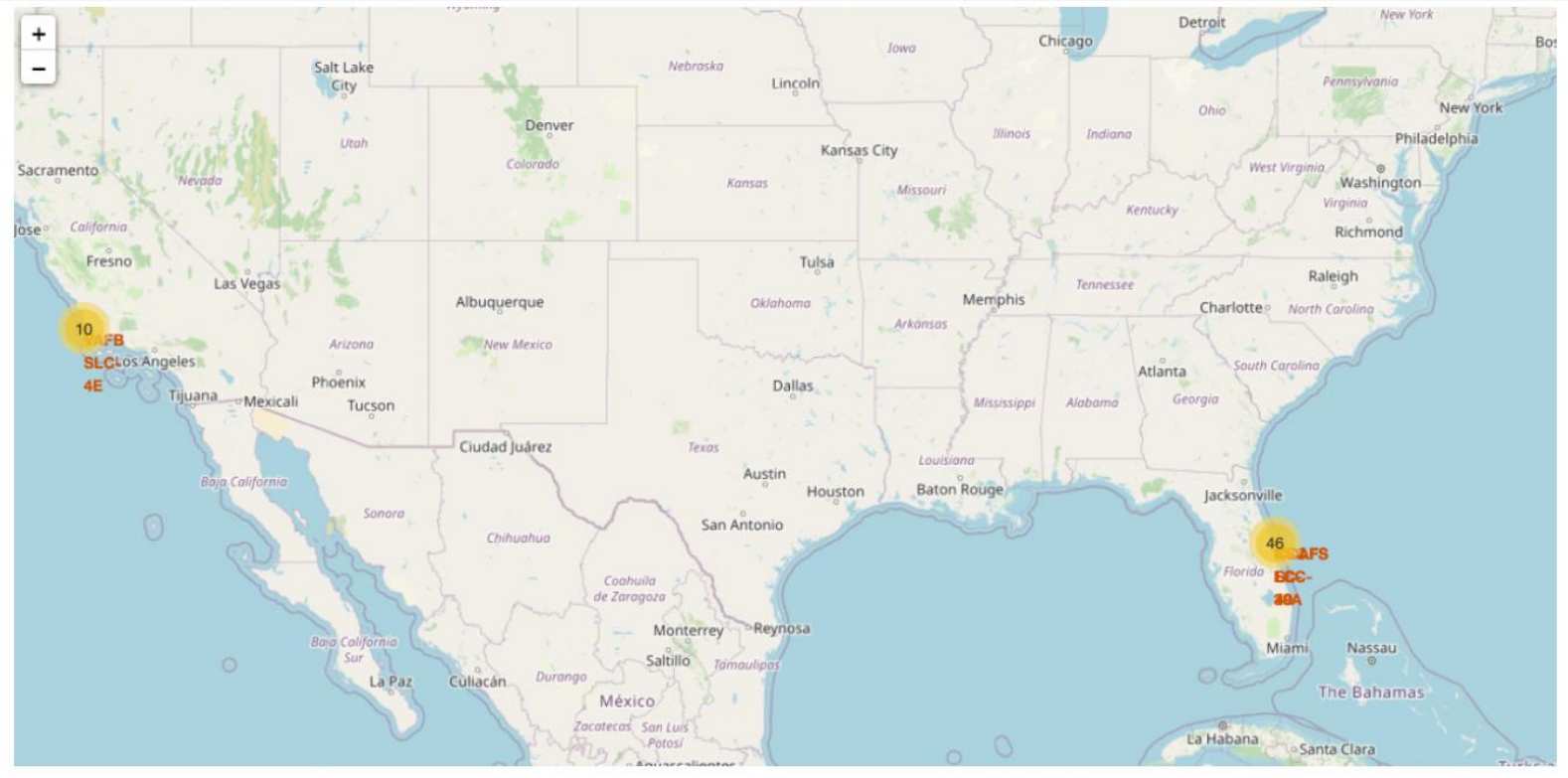
Out[31]:

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


Section 3
Launch Sites
Proximities Analysis

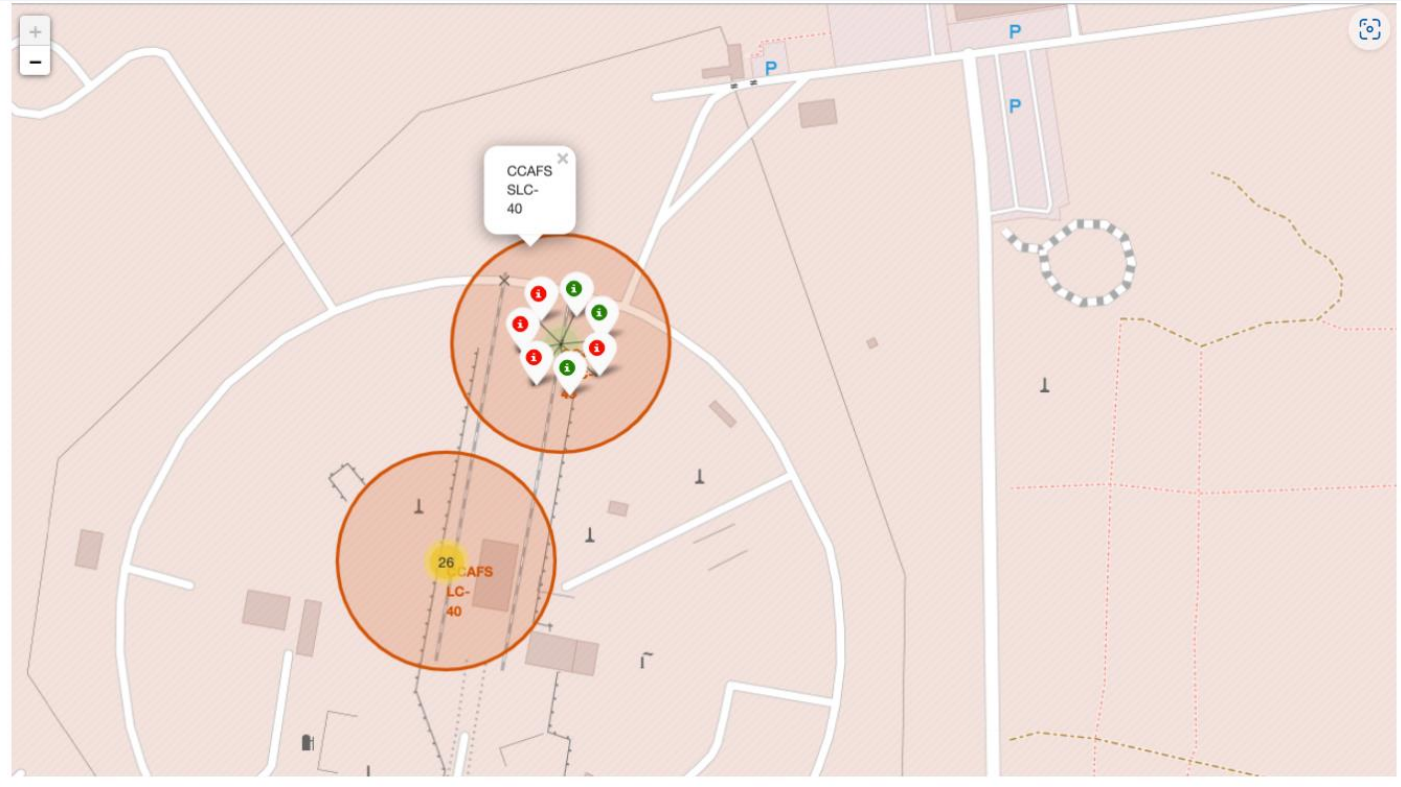
Launch Sites and No of Launches

- Folium Map With Marker Clusters representing number of launches in diff geographical launch sites.



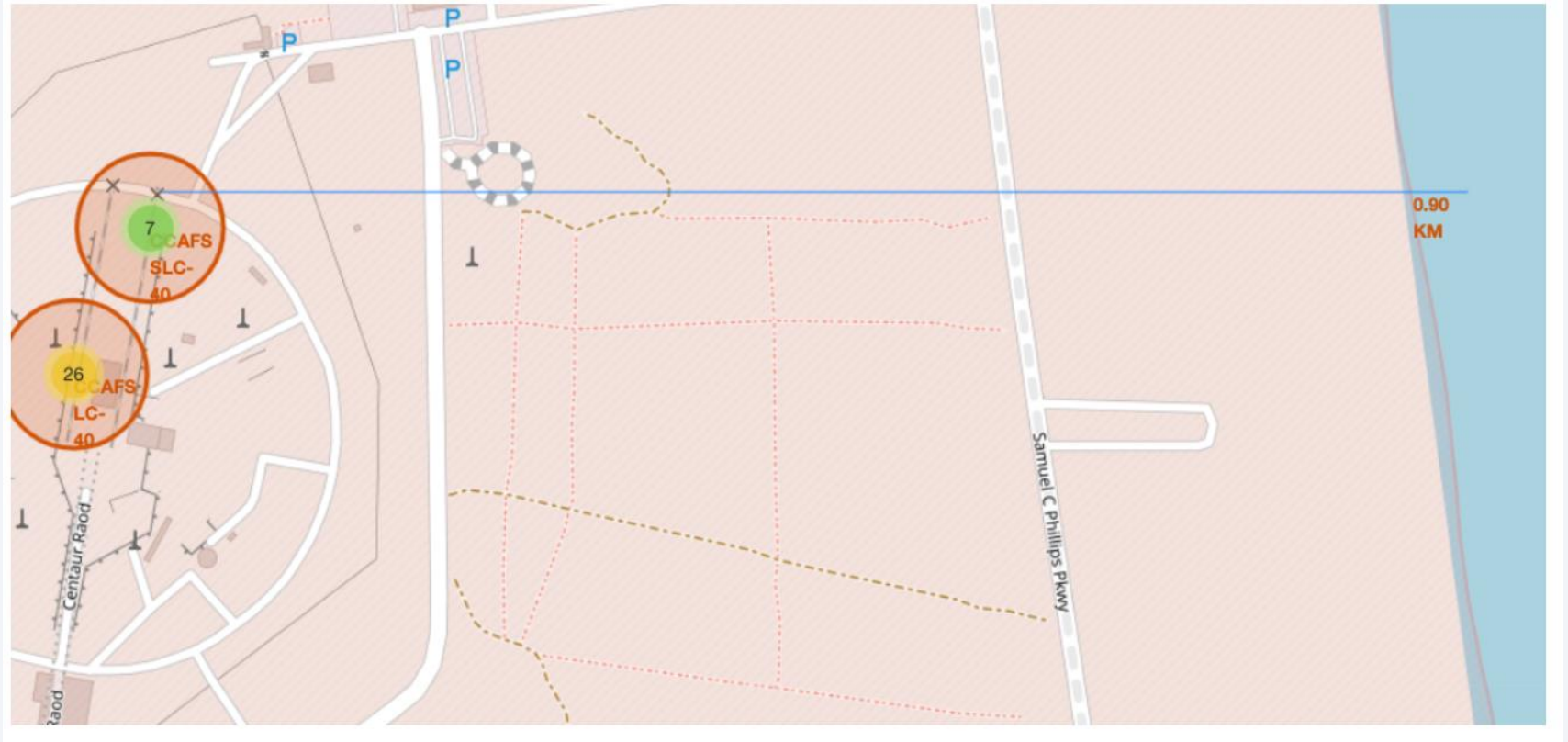
Color Labelled Launch Outcomes

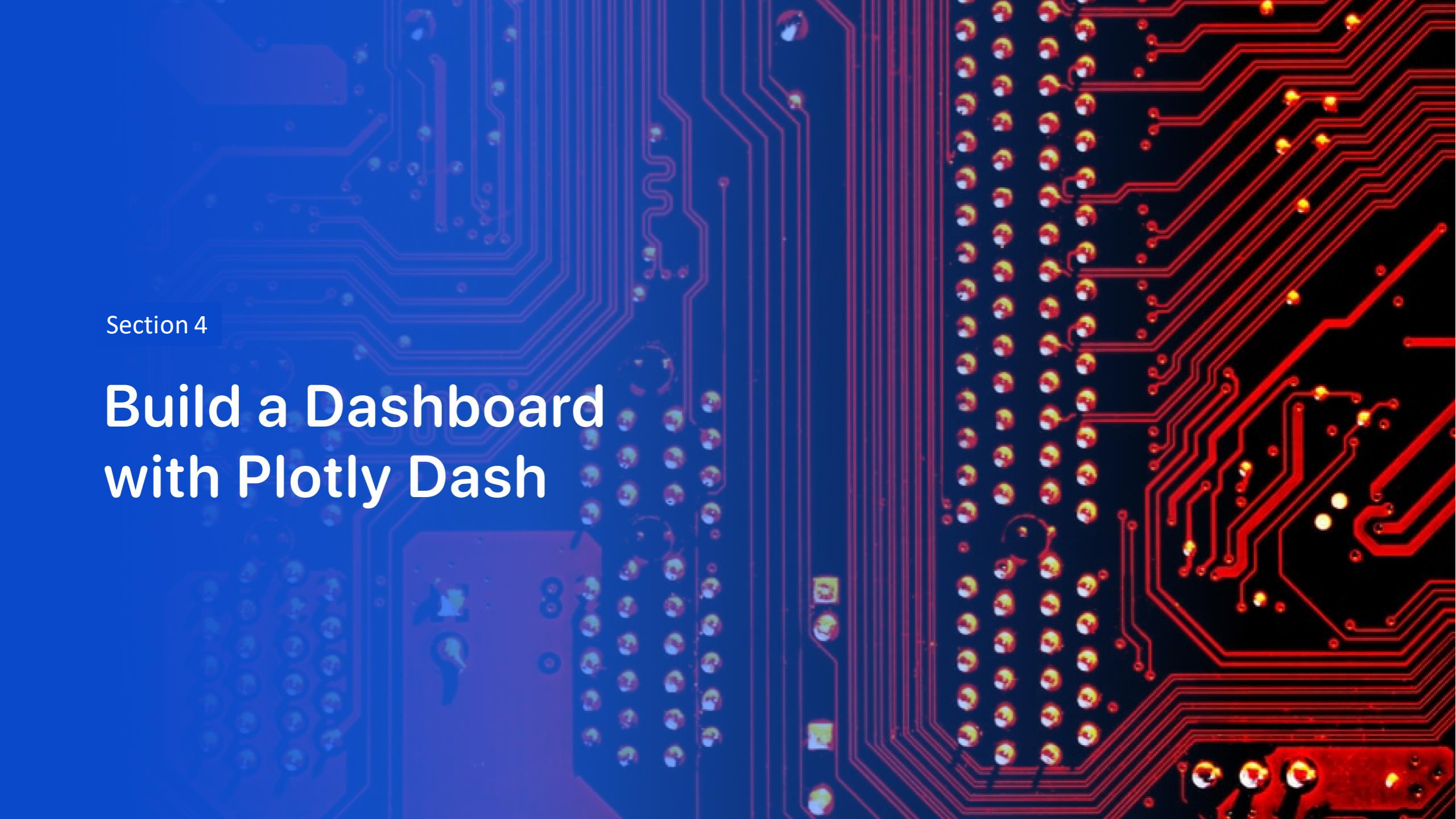
- Color Labelled Launch Outcomes for Launch sites, CCAFS SLC-40 in the below screenshot.



Launch Site DIstances

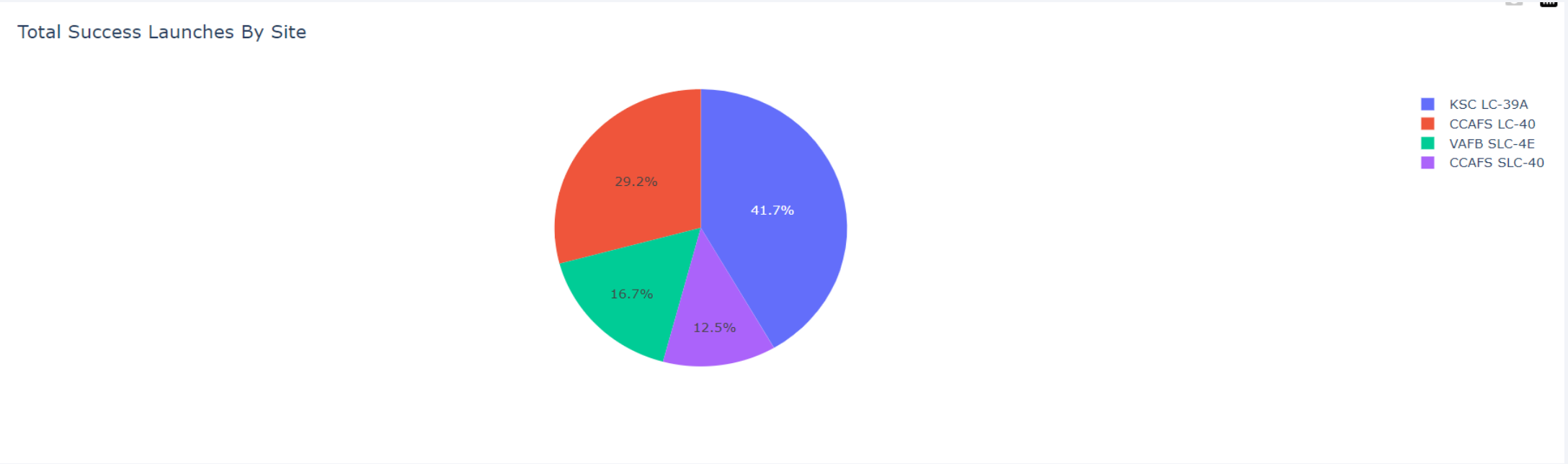
- Line and Marker showing distance of launch sites from Coast line.




Section 4
Build a Dashboard with Plotly Dash

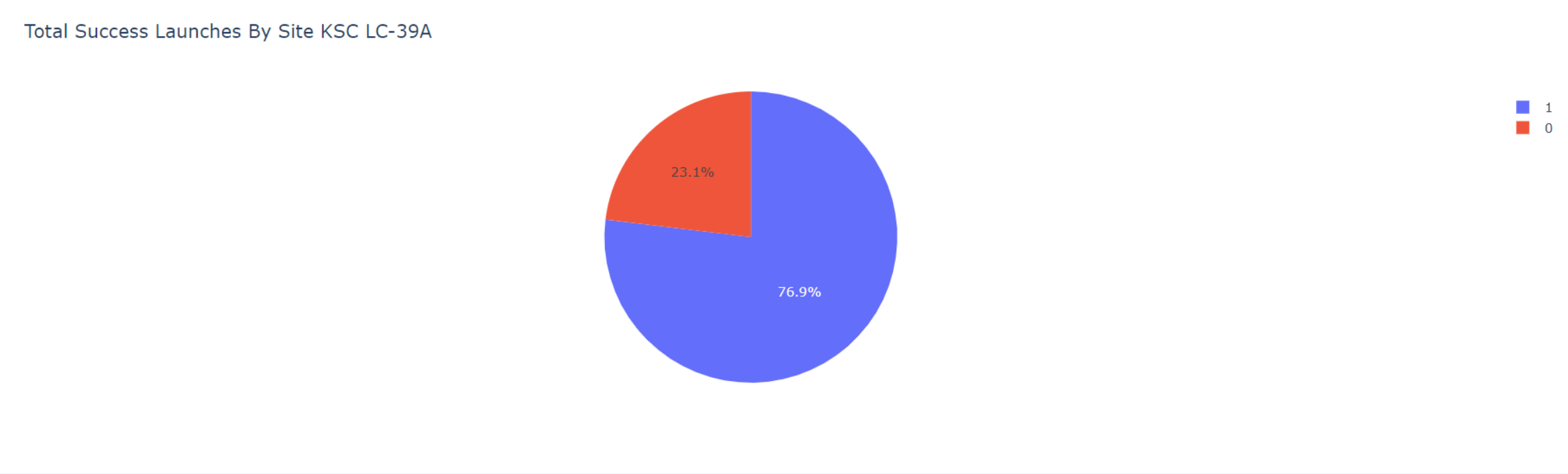
All Sites – Launch Success Counts

- Success Rates for each launch site. KSC LC-39A seems to have very high success rate.



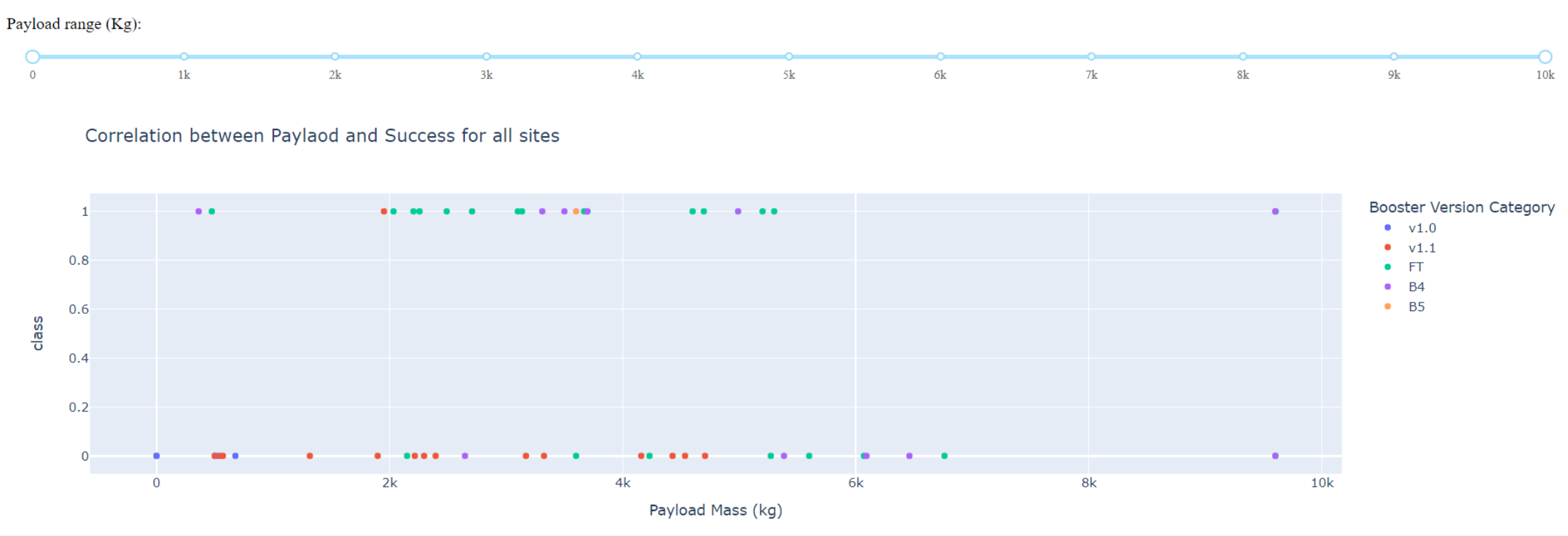
Highest Success Rate Launch Site

- KSC LC-39A, the highest success rate launch site seems to have a 77% success rate.



Payload vs Success rate: All sites

- Payload vs Success Rate: All sites. With booster version color coded.

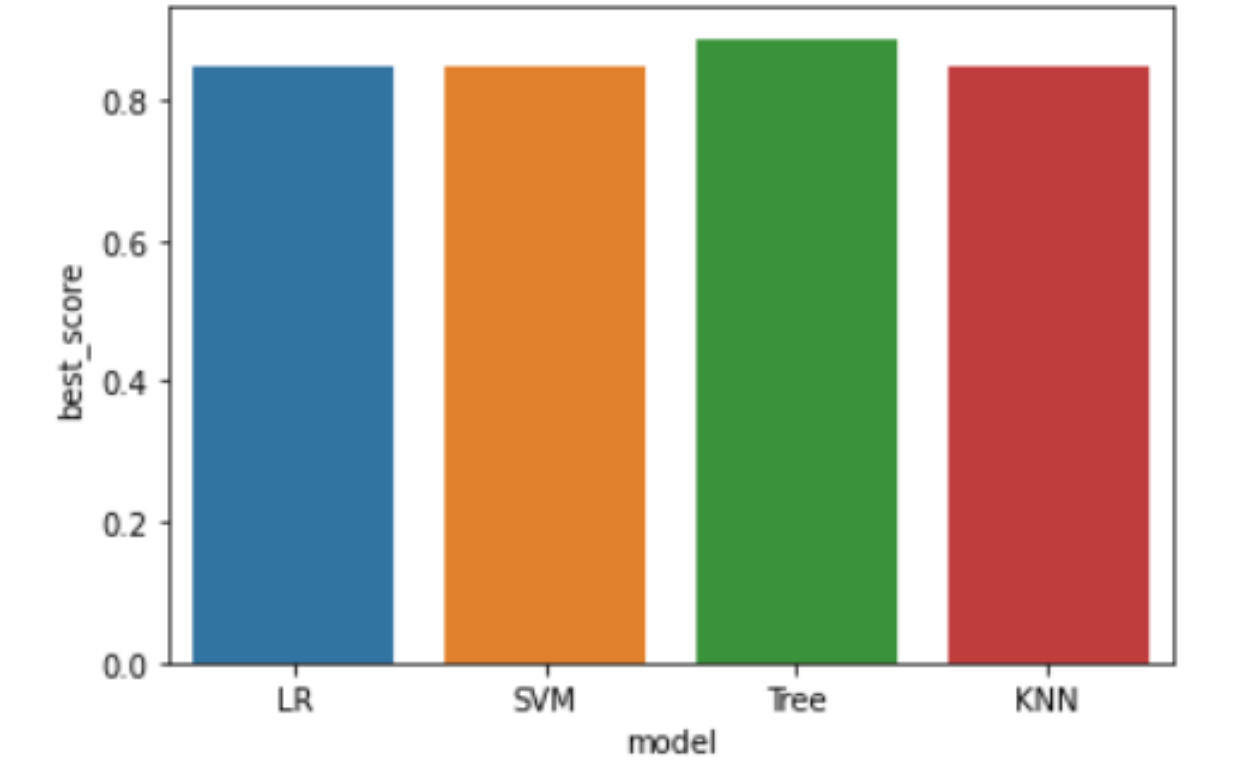


Section 5

Predictive Analysis (Classification)

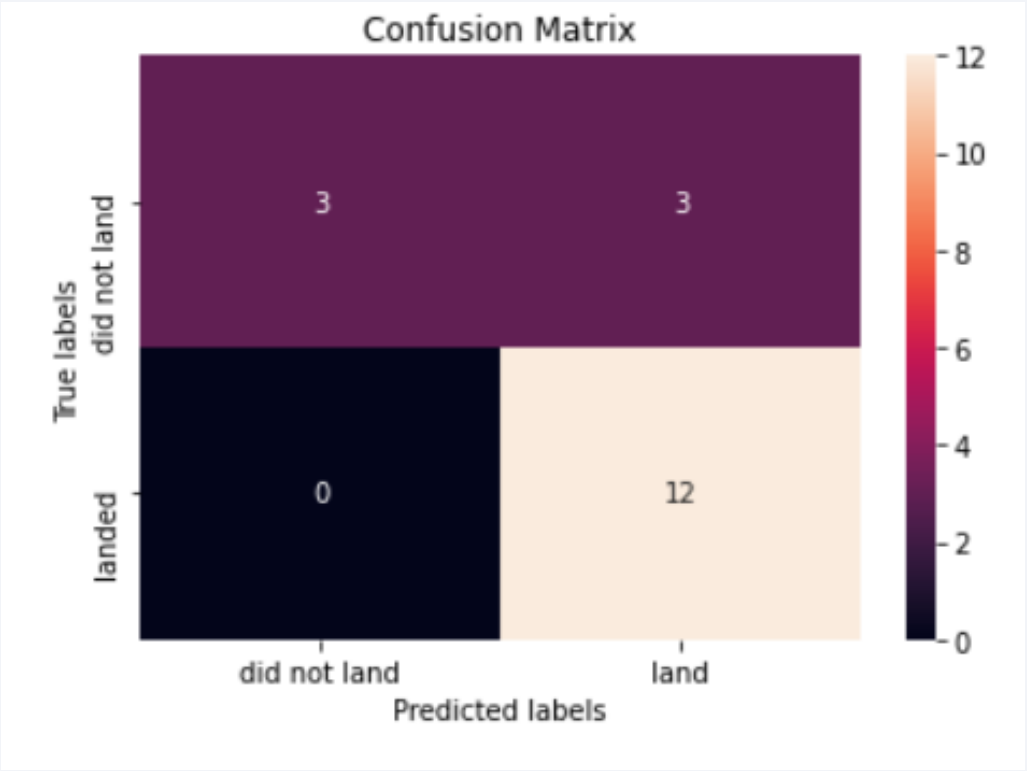
Classification Accuracy

- The bar chart indicating the best scores among the four models based on Logistic regression, support vector machines, Decision Tree and K-nearest Neighbors, in Grid Search with best parameters indicates Decision Tree has best score. So it is chosen as the model of choice.



Confusion Matrix

- Looks like our model performs well with TN, TP and FN, but we have a problem with FP.



Conclusions

- We can very confidently predict whether a launch will land or not with our Decision Tree model. By virtue of being a tree it also gives us the possibility of printing out the rules which further helps with explainability of a prediction.
- We see that the success rate of the launches has always increased since 2013 which is a sign of learning and improvement over the years. At the same time we see correlations between payload mass and success rates as well but it very heavily depends on booster version and type of orbit as well.

Thank you!