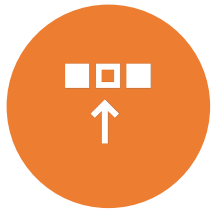
IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Anuj Phogat
May 20, 2024

Outline



EXECUTIVE SUMMARY

INTRODUCTION

METHODOLOGY

RESULTS

CONCLUSION

APPENDIX

Executive Summary

- Objective of the project is **to predict whether SpaceX will be landing the first stage.**
- Two data sources are used:
  - SpaceX API
  - Web scrapping (from Wikipedia)
- Initial data analysis showed the probability of successful landing is dependent on the launch site, the orbit and the payload mass.
- After deploying different ML models for prediction, Decision Tree model has been selected as the best predictor.
- The Decision Tree model is successfully able to predict the landing outcome with 88.89% accuracy score.

Introduction

- SpaceX has been able to reduce the average cost their Falcon 9 rocket launches down to 62 million USD while the industry average is 162 million USD each. Primary reason of their savings is their ability to reuse the first stage.

- As a competitor to SpaceX, it is essential for SpaceY to be able to predict the outcome of the landing of the first stage given the launch parameters. This information will assist SpaceY in bidding against SpaceX for a rocket launch.

- The objective of this project is to build a machine learning algorithm that will be able to predict the outcome of the first stage landing using different parameters of a launch.

Section 1

Methodology

Methodology

- Data collection methodology
  - Data is collected using multiple SpaceX APIs and by web scrapping a Wikipedia page
- Performed data wrangling
  - Replaced the missing values by average of that column (except for ‘Landing Pad’ field)
  - Filtered out data to include only ‘Falcon 9’ launch records
  - Converted all categorical variables into dummy variables using one-hot encoding
  - Created a column ‘Class’ with value 1 for successful landing and 0 for a failed one
- Performed exploratory data analysis (EDA) using visualization and SQL
- Performed interactive visual analytics using Folium and Plotly Dash

Methodology

- Performed predictive analysis using classification models
  - Built the following classification models – Logistic regression, SVM, Decision tree and k-Nearest neighbors
  - Used Grid Search to optimize hyper parameters
  - Model with highest accuracy score on test data is selected

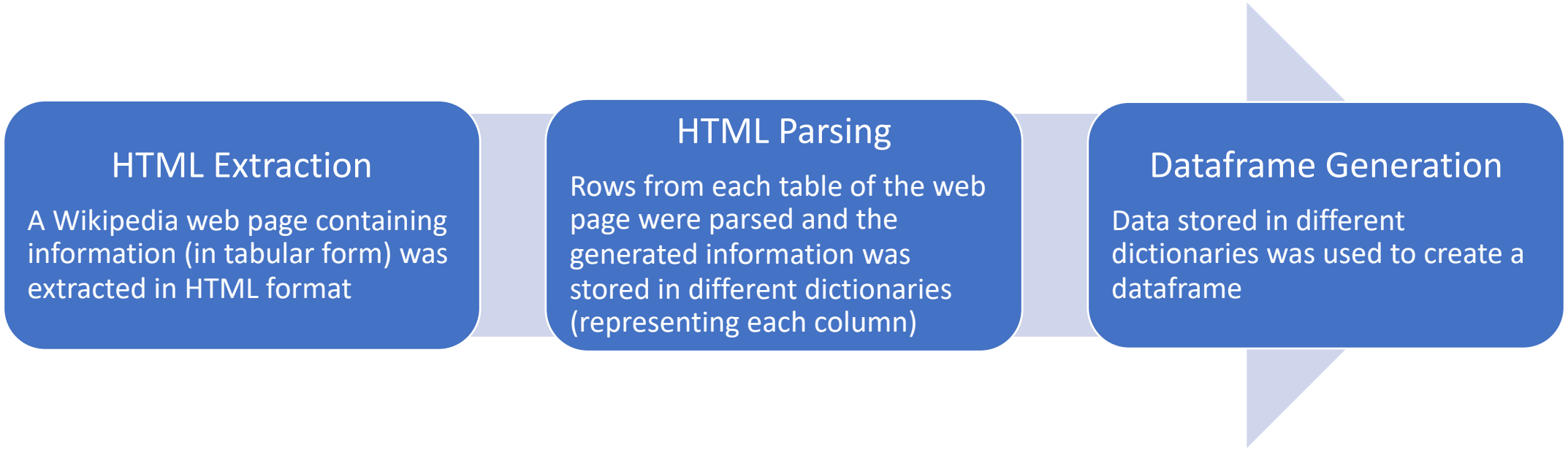
Data Collection

Data sets were collected from two sources: SpaceX APIs and Wikipedia web scrapping

1. SpaceX APIs

Data Collection

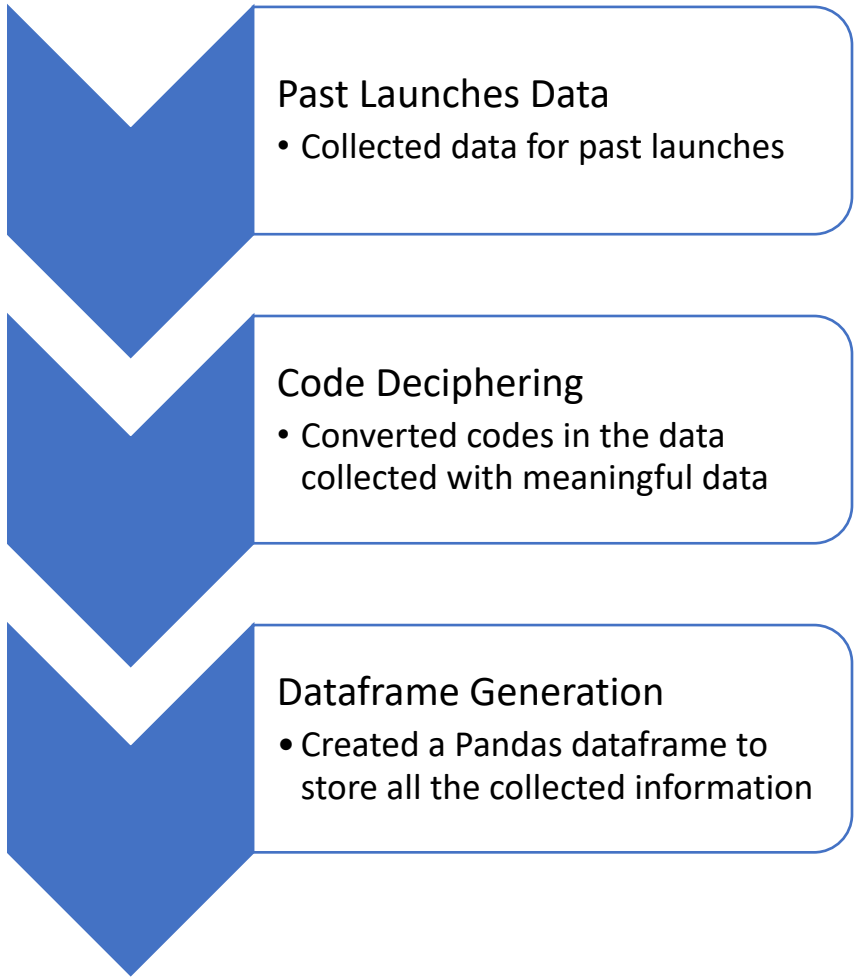
2. Wikipedia Web Scraping



Data Collection – SpaceX API

- Extracted data for past launches by sending GET request using the Request library to SpaceX APIs endpoint – launches/past
- Codes received in the booster version, payload, launch pad and core columns were then converted to meaningful data using respective API endpoints
- Output from all the APIs was stored in the lists which were then used to create a Pandas dataframe
- Filtered out data to keep only Falcon 9 launch records
- Replaced missing values with average value in the column Payload Mass

Notebook: 01_Data_Collection_API.ipynb

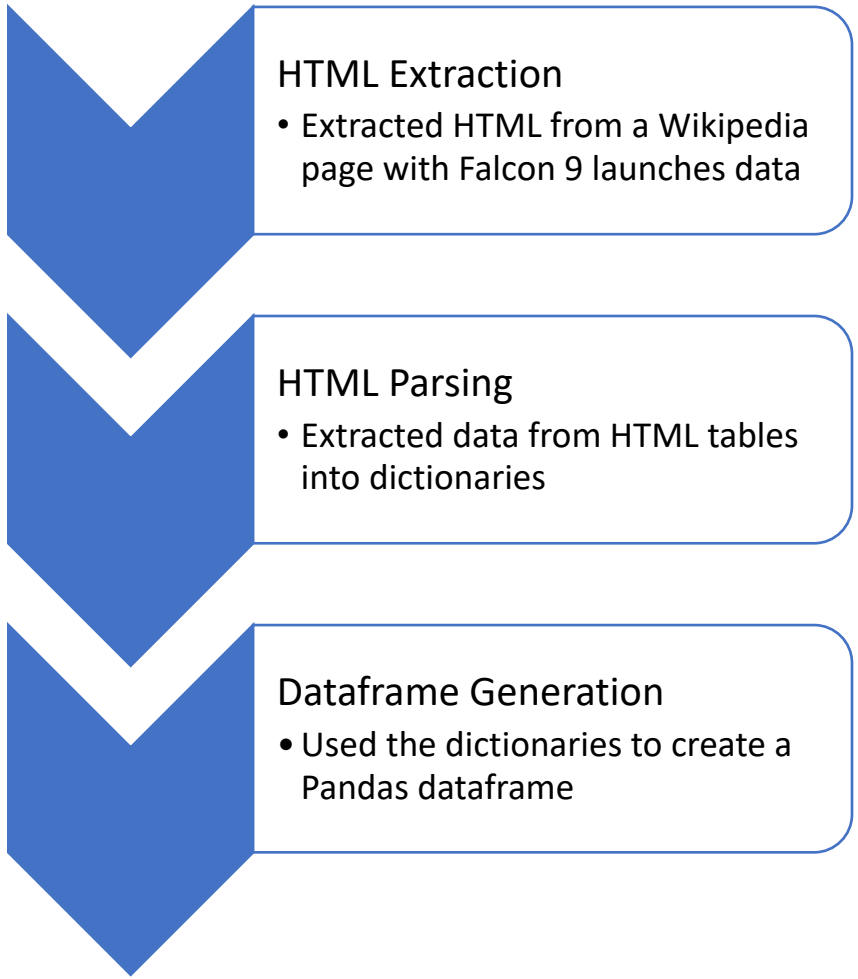


** Specific URLs for SpaceX API endpoints can be found in Appendix.*

Data Collection – Scraping

- Extracted a Wikipedia page containing information about Falcon 9 launches in HTML format by sending a GET request
- Parsed the HTML received using BeautifulSoup library
- Used *find_all()* method to extract data from all tables available in the HTML and stored it in the respective dictionary created for each column
- Dictionaries created for each column were used to generate a Pandas dataframe

Notebook: 02_Data_Collection_with_Web_Scrapping.ipynb



** Specific URL for Wikipedia webpage used for scraping can be found in Appendix.*

Data Wrangling

- First level of exploratory data analysis was performed:
  - Calculated the number of launches for each site
  - Calculated the number and occurrence of each orbit
  - Created a dictionary for all the possible launch outcomes and their occurrence in the data
- Created a landing outcome label with value 0 for failed landings and 1 for successful landings

- Notebook: 03_Data_Wrangling.ipynb

EDA with Data Visualization

- Scatter plots were created to visualize the combined effect of below variables on landing outcome:
  - Flight Number vs Launch Site (to analyze whether the rate of successful landing is going up with more launches for any of the launch site)
  - Payload Mass vs Launch Site (to analyze whether using any specific launch site for a particular payload mass range leads to successful landings)
  - Flight Number vs Orbit (to analyze whether the rate of successful landing is going up with more launches to any of the orbits)
  - Payload Mass vs Orbit (to analyze whether launches sent to a specific orbit type for a particular payload mass range leads to successful landings)
- Scatter plots were chosen as there is at least one categorical variable between the variables being analyzed
- Different color markers were used to specify successful vs failed landings in each scatter plot
- A bar chart was created to visualize the success rate for each orbit type
- A line chart was created to visualize the rate of successful landings from 2013 to 2020

- Notebook: 05_EDA_with_Data_Visualization.ipynb

EDA with SQL

- Loaded the Pandas dataframe into a SQL table
- Found out the unique launch sites in the dataset by using DISTINCT in the SELECT statement
- Displayed five records from the dataset where launch site began with *CCA* using WHERE and LIMIT clause in the SELECT statement
- Calculated the total and average payload mass for boosters launched by *NASA (CRS)* and booster version *F9 v1.1* using SUM and AVG functions, respectively
- Found out total number of successful and failed landings by using LIKE with WHERE clause
- Found out the booster versions which carried the maximum payload using a sub-query
- Ranked the count of landing outcomes between a certain date range using WHERE clause with GROUP BY and ORDER BY

- Notebook: 04_Exploratory_Data_Analysis_Using_SQL.ipynb

Build an Interactive Map with Folium

- A map object was created using Folium
- Circles were added at each launch site along with site name to observe the proximity of each launch site from equator and the coast
- Markers with marker clusters were added for each launch record at the corresponding launch site
- Each marker was color-coded to distinguish successful and failed landings to identify if any launch site has a higher success percentage
- Lines were added in the map to check the distance between the launch site and the nearest coastline and railway track

- Notebook: 06_Interactive_Visual_Analytics_with_Folium.ipynb

Build a Dashboard with Plotly Dash

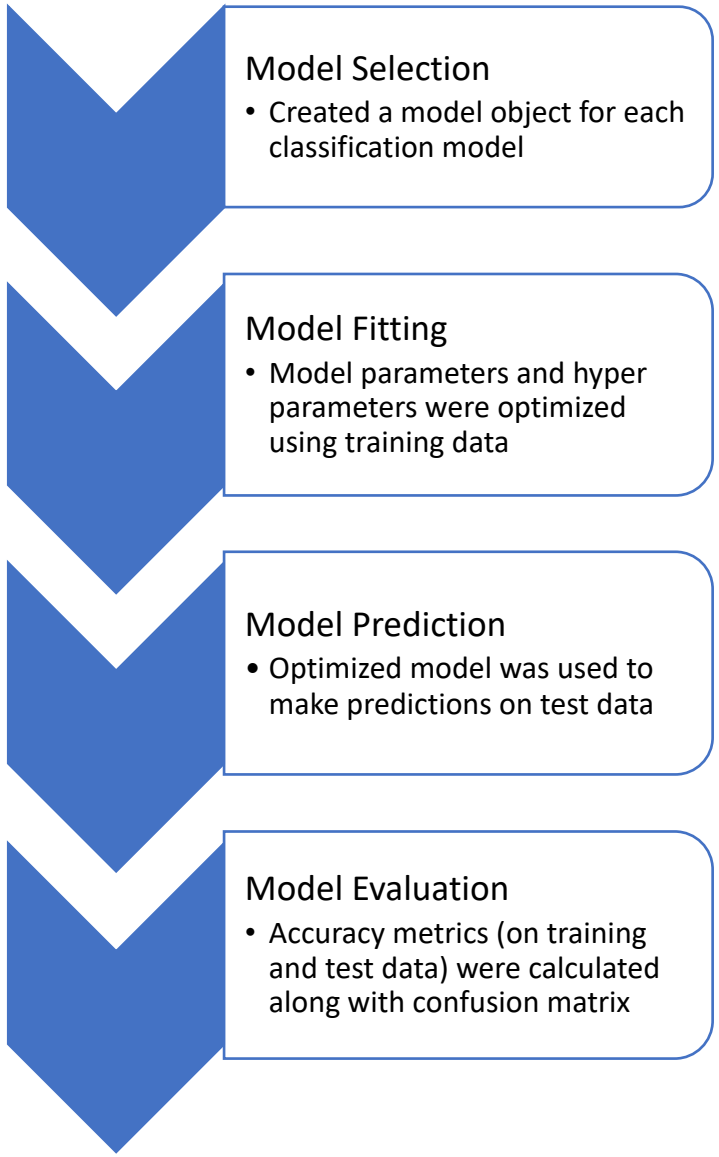
- A dashboard with title *SpaceX Launch Records Dashboard* was created
- A dropdown and a range slider was added to allow for launch site and payload mass selection, respectively
- A pie chart was added:
  - Showing the total successful launches by site when no specific launch site is selected from dropdown
  - Showing successful vs failed launches when a specific launch site is selected from dropdown
- A scatter chart was added:
  - Showing the correlation between payload mass (as defined by the range slider) and launch outcome for all sites when no specific launch site is selected from dropdown
  - Showing the correlation between payload mass (as defined by the range slider) and launch outcome for the selected launch site when a specific launch site is selected from dropdown

- Notebook: 07_Interactive_Dashboard_with_Plotly_Dash.ipynb

Predictive Analysis

- Feature set and target set were split into training and test set
- Following classifier models were created and evaluated - Logistic regression, SVM, Decision tree and k-Nearest neighbors
- Model object was first created and then GridSearchCV was used to tune the hyper parameters
- Model was fitted on training data
- Out-of-sample accuracy score for each model was calculated using test data
- A prediction array was generated to make predictions on test data and to generate a confusion matrix

- Notebook: 08_Machine_Learning_Prediction.ipynb



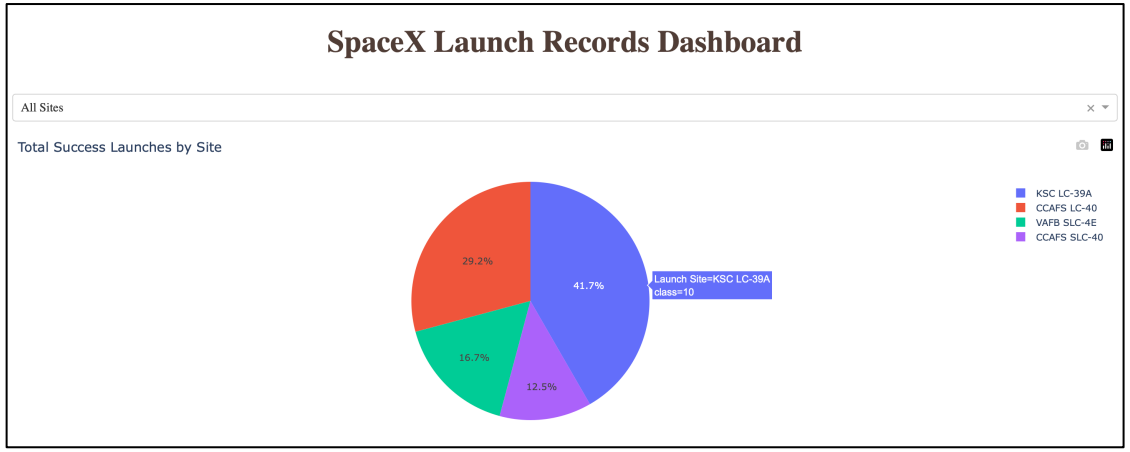
Results

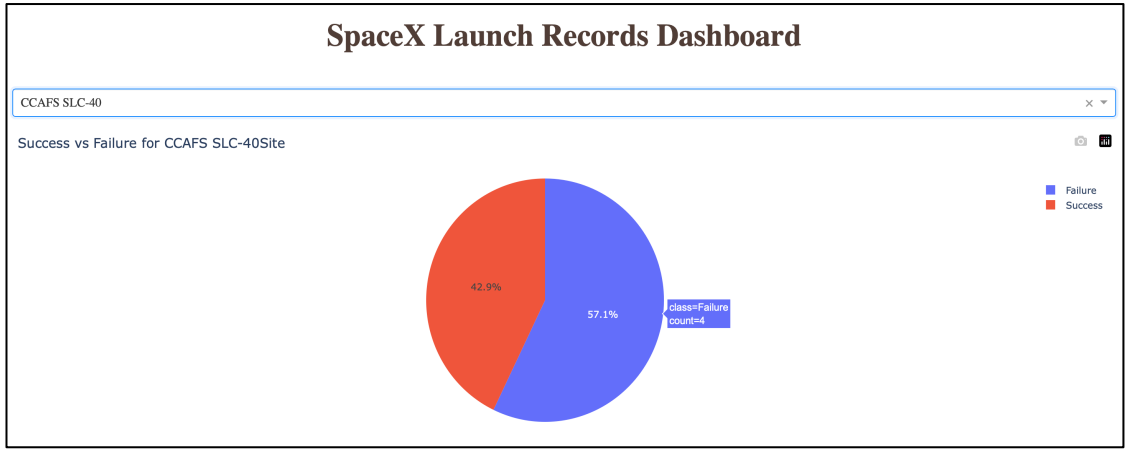
Exploratory Data Analysis:

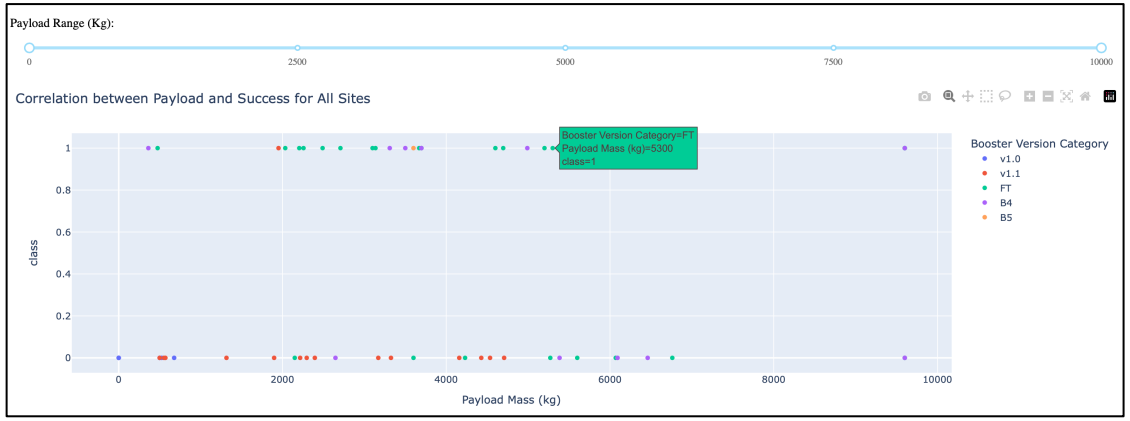
- Dataset has launch records from four unique launch sites in which “CCAFS SLC 40” has highest launch count with increasing success rate
- Average payload mass of all launches is 2928.4 kgs
- Number of successful landings is 61 and failed landings is 10
- The success rate is 100% for “ES-L1”, “GEO”, “HEO” and “SSO” orbits
- For “ISS” and “LEO” orbits, success rate goes up with higher payload mass
- The success rate is trending upwards over the years

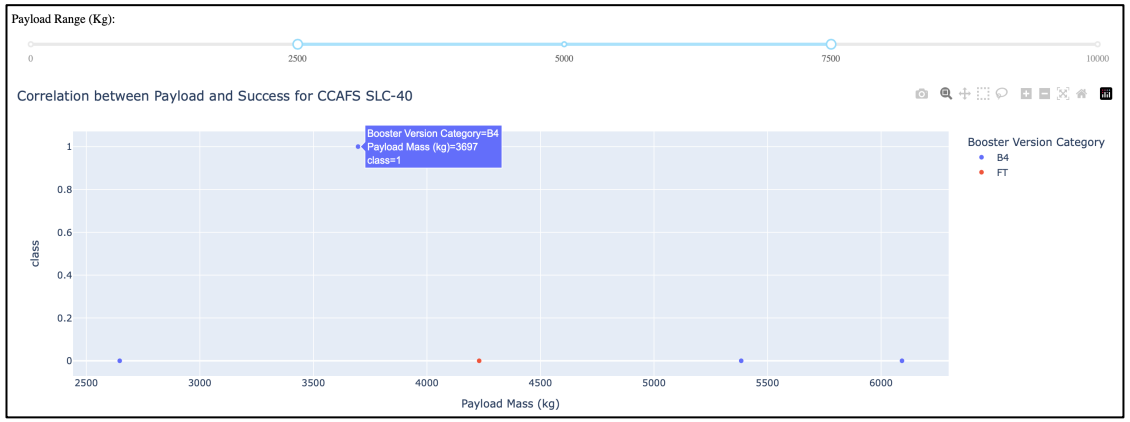
Results

Interactive Analysis:

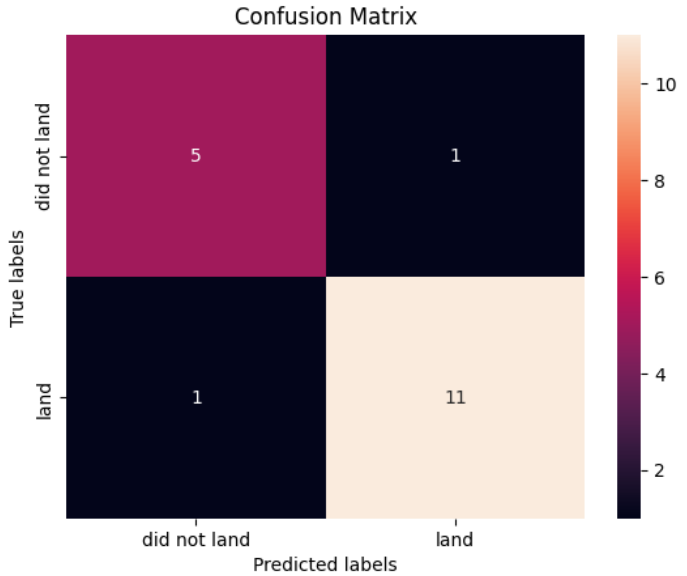






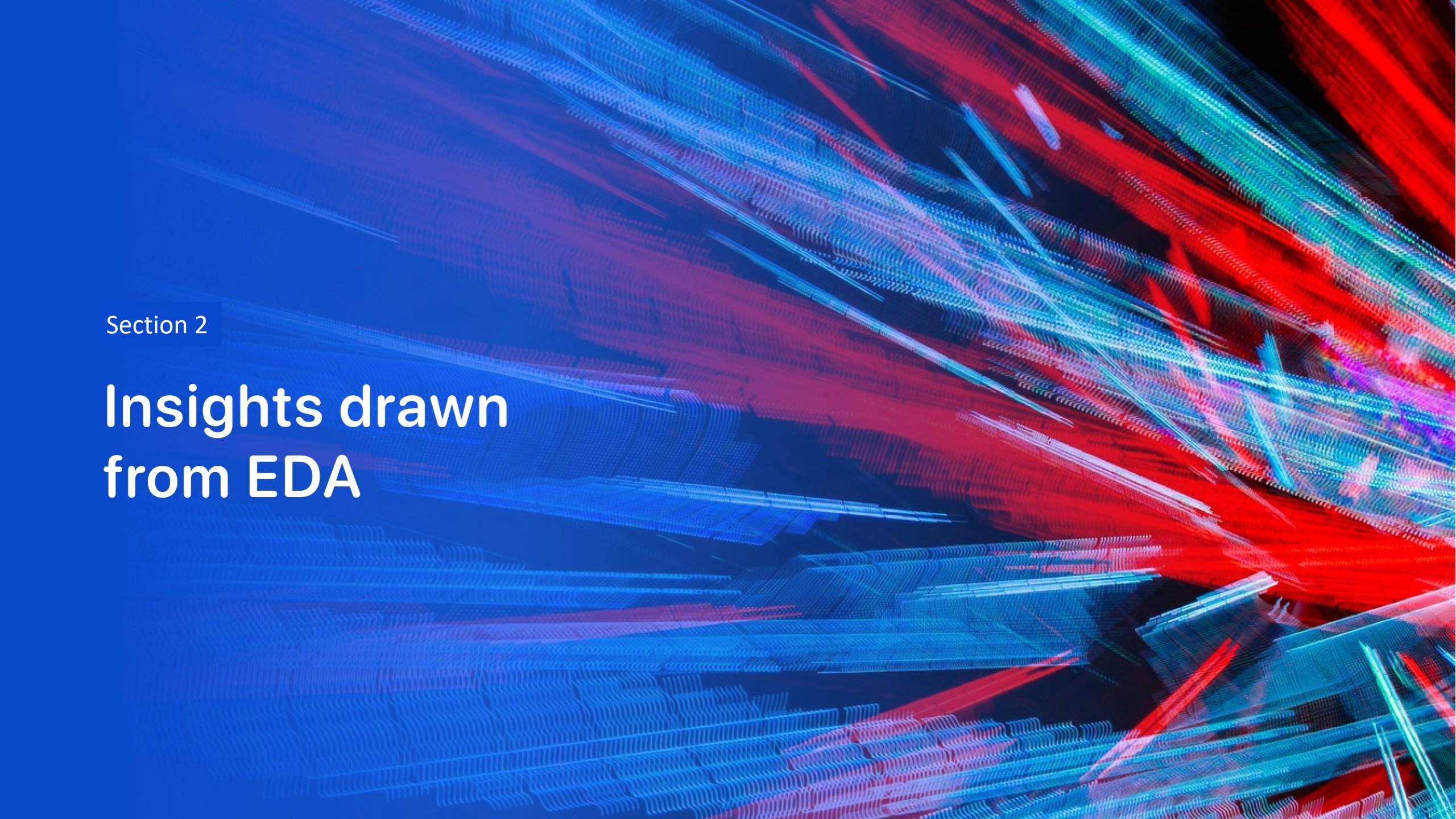


Results



Predictive Analysis:

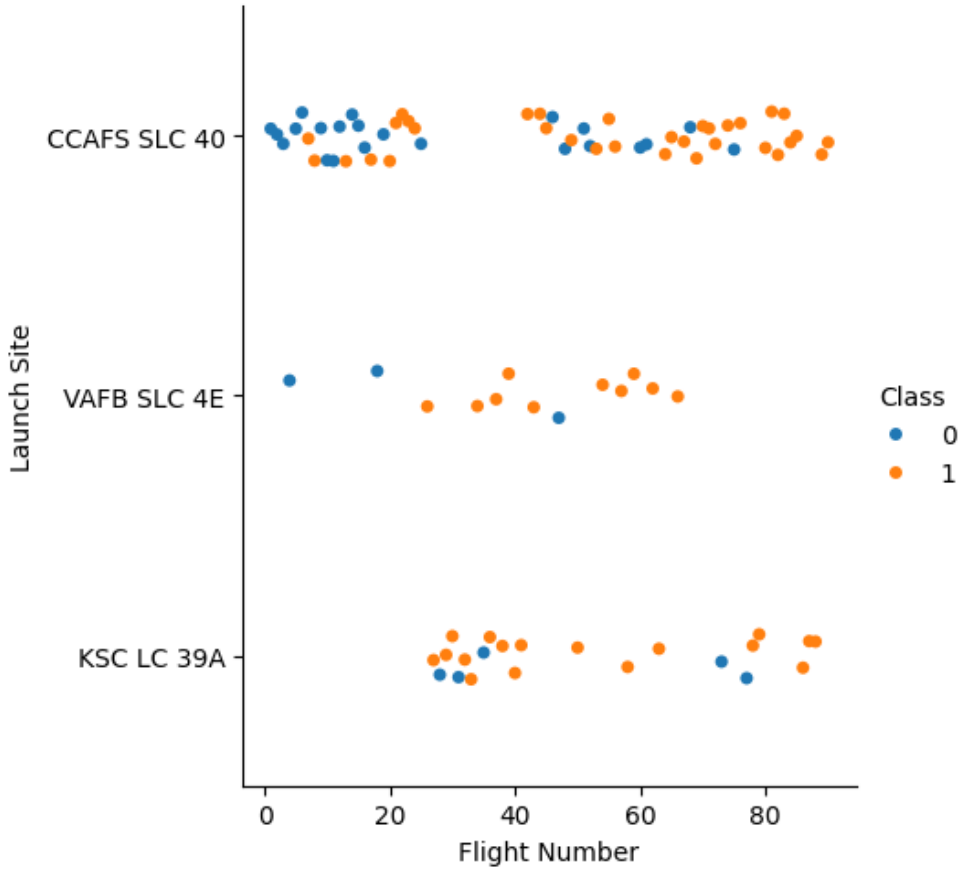
- In-sample accuracy score of each classifier:
  - Logistic Regression – 84.64%
  - Support Vector Machine – 84.82%
  - Decision Tree – 88.93%
  - K-Nearest Neighbors – 84.82%
- Decision tree model had the out-of-sample score of 88.89%, rest of the models had the same out-of-sample accuracy score of 83.33% and the same confusion matrix
- Decision tree was considered as the best classifier as it has the highest out-of-sample accuracy score


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

Observations:

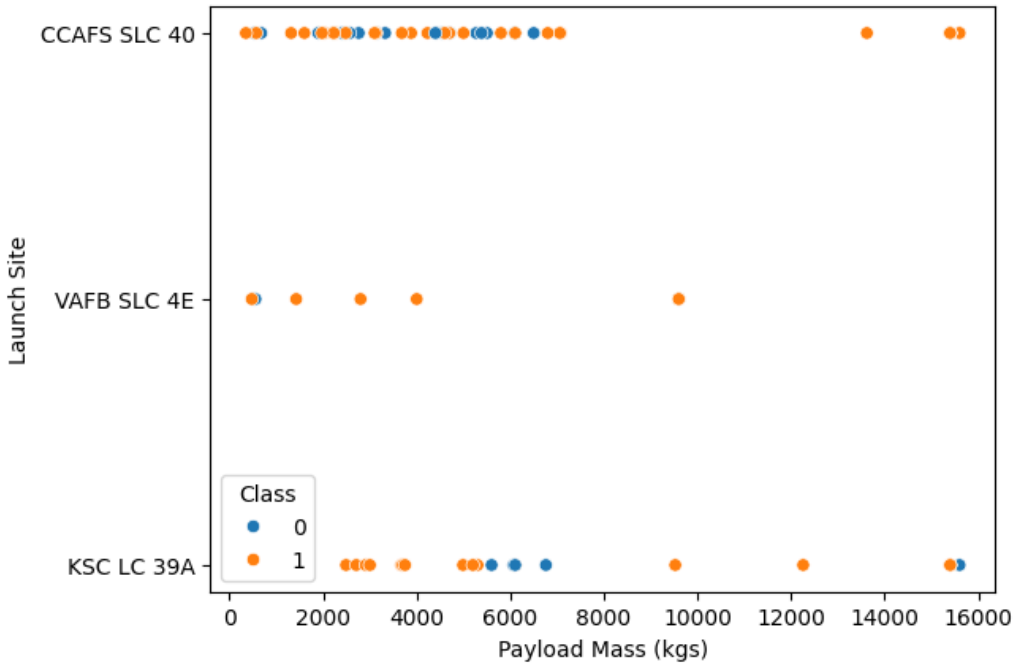
- CCAFS SLC 40 has the highest number of launches
- VAFB SLC 4E has the least number of launches and does not seem to be used for later launches
- KSC LC 39A was not used for launches initially and is only being used after about first 25 launches
- All three launch sites have success rates going up with the number of launches



Payload vs. Launch Site

Observations:

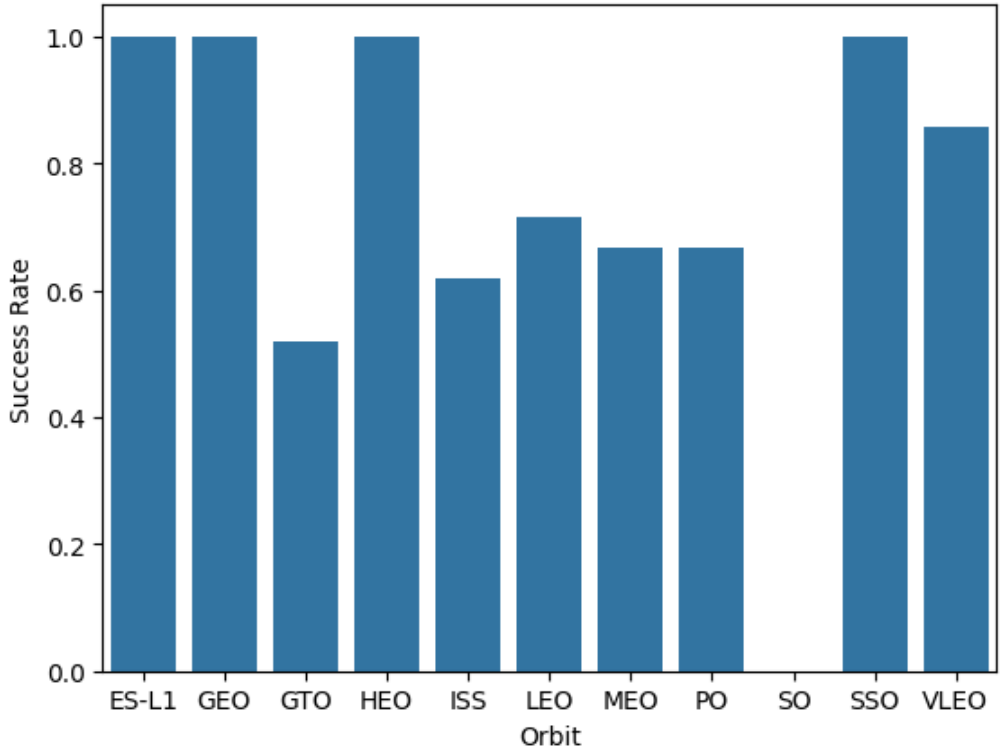
- CCAFS SLC 40 and KSC LC 39A are the only launch sites being used for launches with heavy payload mass
- VAFB SLC 4E is being used for launches with light-weight and medium-weight payload mass
- Success rate is high for launches with light-weight and heavy-weight payloads than with medium-weight payloads for all three sites
- The launch distribution is skewed towards lower spectrum of payload mass



Success Rate vs. Orbit Type

Observations:

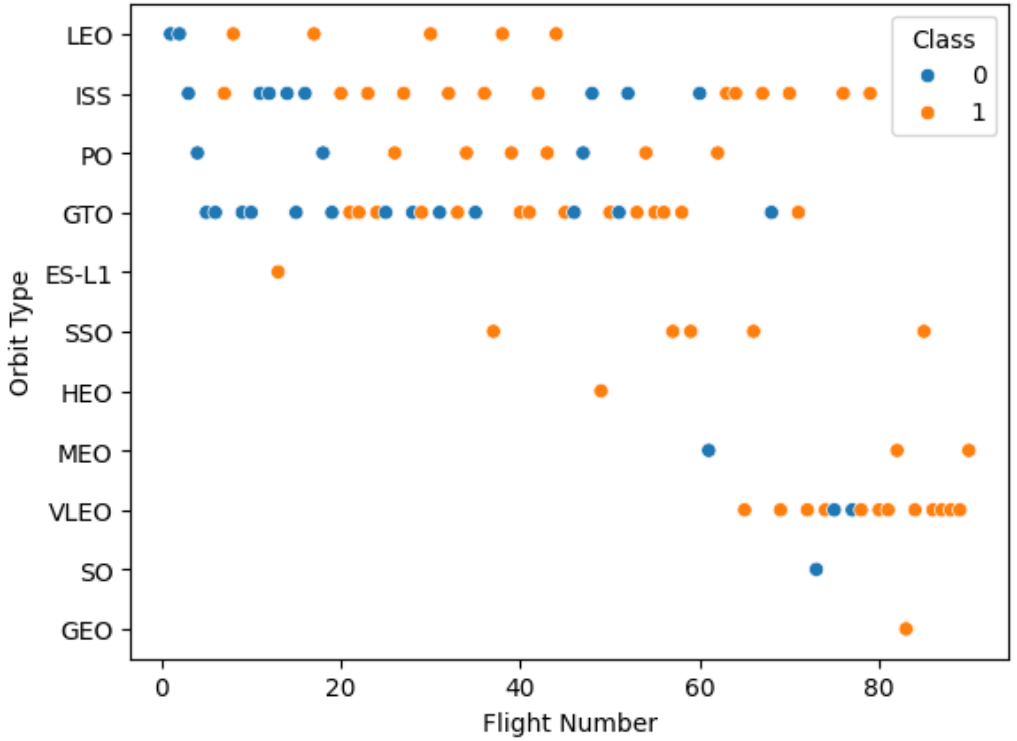
- The success rate is 100% for the ES-L1, GEO, HEO and SSO orbits
- The success rate for SO orbit is 0% although there is only one launch record for this orbit
- The success rate appears to be more than 60% on average for all the orbits



Flight Number vs. Orbit Type

Observations:

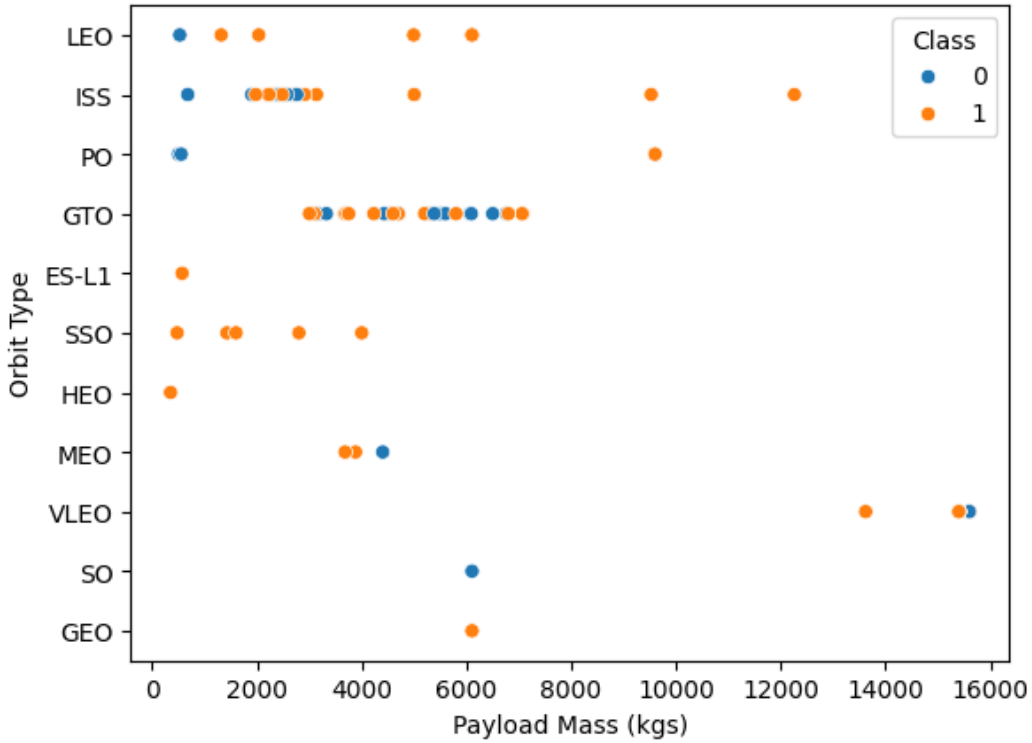
- For LEO orbit, as flight number is increasing, success rate is going up
- No visible relationship between flight number and success rate for GTO orbit
- The number of launches to ES-L1, HEO, SO, GEO and PO orbits are too low to make any observations



Payload vs. Orbit Type

Observations:

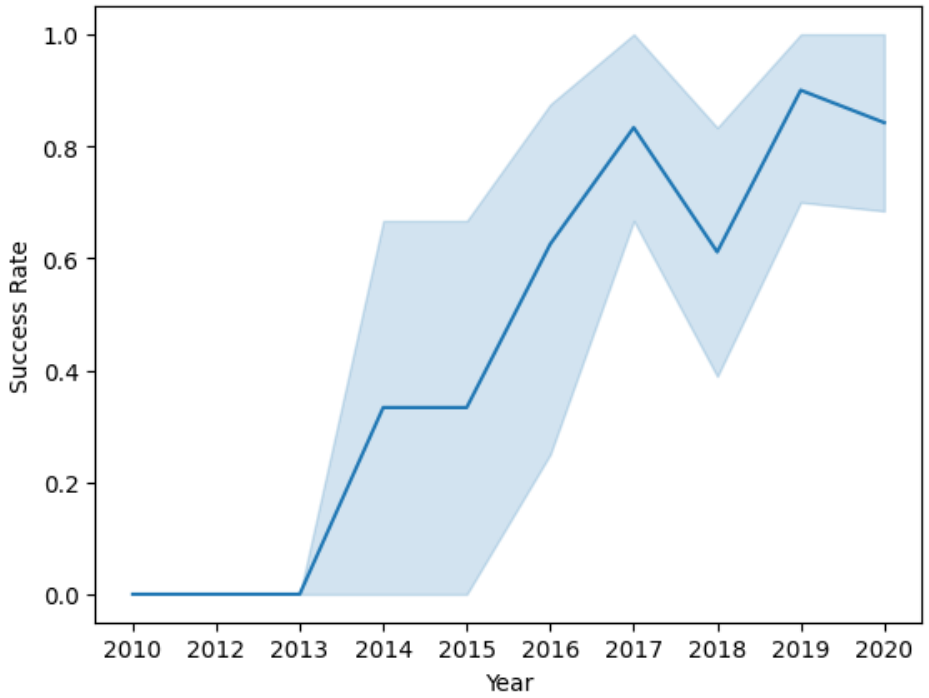
- The success rate for ISS and LEO orbits is going up with increase in payload mass
- No visible relationship between payload mass and success rate for GTO orbit
- SSO orbit has 100% success rate for light-weight payload launches
- The number of launches to ES-L1, HEO, SO, GEO and PO orbits are too low to make any observations



Launch Success Yearly Trend

Observations:

- The success rate has consistently gone up since 2013, except for a decline (of about 20%) in 2018



All Launch Site Names

- Used the keyword DISTINCT in the SELECT statement to get the name of unique launch sites
- Have 4 unique launch sites – CCAFS LC-40, VAFB SLC-4E, KSC LC-39A and CCAFS SLC-40

```
# Unique values in the Launch site column
%sql select distinct "Launch_Site" from SPACEXTABLE;
```

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

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

Launch Site Names Begin with 'CCA'

- Used the keyword LIMIT in the SELECT statement to restrict the output to five records
- Used the WHERE clause in the SELECT statement to filter on launch site column
- Used the keyword LIKE in the WHERE clause to search for the names that begin with CCA

```
%sql select * from SPACEXTABLE where "Launch_Site" like 'CCA%' limit 5;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
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

- Used the WHERE clause in the SELECT statement to filter on customer column
- Used the function SUM in the SELECT statement to sum up the payload mass
- Used the keyword AS in the SELECT statement to rename the column returned by the query
- Total payload mass carried by boosters launched by NASA (CRS) = 45,596 kgs

```
%sql select sum("PAYLOAD_MASS__KG_") as 'Total Payload Mass by NASA (CRS)' from SPACEXTABLE where Customer = 'NASA (CRS)';
 * sqlite:///my_data1.db
Done.
Total Payload Mass by NASA (CRS)
45596
```

Average Payload Mass by F9 v1.1

- Used the WHERE clause in the SELECT statement to filter on booster version column
- Used the function AVG in the SELECT statement to calculate the average of the payload mass
- Average payload mass carried by booster version F9 v1.1 = 2,928.4 kgs

```
%sql select avg("PAYLOAD_MASS__KG_") from SPACEXTABLE where "Booster_Version" = 'F9 v1.1';

 * sqlite:///my_data1.db
Done.
avg("PAYLOAD_MASS__KG_")
2928.4
```

First Successful Ground Landing Date

- Used the WHERE clause in the SELECT statement to filter on landing outcome column
- Used the function MIN in the SELECT statement to find out the oldest date
- First successful ground landing date is Dec 22nd, 2015

```
%sql select min("Date") from SPACEXTABLE where "Landing_Outcome" = 'Success (ground pad)';
```

```
 * sqlite:///my_data1.db
Done.
min("Date")
2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- Used the WHERE clause in the SELECT statement to filter on landing outcome and payload mass columns
- Used the keyword AND in WHERE clause to apply simultaneous filters on two columns
- Used the keyword DISTINCT in the SELECT statement to get the name of unique booster versions
- Have 4 booster versions that meet the specified criteria – F9 FT B1022, F9 FT B1026, F9 FT B1021.2 and F9 FT B1031.2

```
%sql select distinct "Booster_Version" from SPACEXTABLE where "Landing_Outcome" = 'Success (drone ship)' and "PAYLOAD_MASS__KG_" between 4000 and 6000;
```

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

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

Total Number of Successful and Failure Mission Outcomes

- Used the WHERE clause in the SELECT statement to filter on landing outcome column
- Used the keyword LIKE in the WHERE clause to search for the outcomes that begin with Success (Failure)
- Used the function COUNT in the SELECT statement to calculate the number of launches that meet each criteria
- Used the keyword AS in the SELECT statement to rename the column returned by the query
- Number of successful (failure) mission outcomes = 61 (10)

```
# Success outcomes
%sql select count("Landing_Outcome") as 'No of successes' from SPACEXTABLE where "Landing_Outcome" like 'Success%';

 * sqlite:///my_data1.db
Done.
No of successes
61
```

```
# Failure outcomes
%sql select count("Landing_Outcome") as 'No of failures' from SPACEXTABLE where "Landing_Outcome" like 'Failure%';

 * sqlite:///my_data1.db
Done.
No of failures
10
```

Boosters Carried Maximum Payload

- Used the WHERE clause in the SELECT statement to filter on payload mass column
- Used a subquery to get the maximum payload mass in the dataset
- Used the function MAX in the SELECT statement (subquery) to find out the maximum payload mass
- Used the keyword DISTINCT in the SELECT statement to get the name of unique booster versions
- Booster versions = 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 and F9 B5 B1049.7

```
%sql select distinct "Booster_Version" from SPACEXTABLE where "PAYLOAD_MASS__KG_" = (select max("PAYLOAD_MASS__KG_") from SPACEXTABLE);

 * sqlite:///my_data1.db
Done.
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

- Used the WHERE clause in the SELECT statement to filter on (month from) date and landing outcome columns
- Used the keyword AND in WHERE clause to apply simultaneous filters on two columns
- Used the function SUBSTR in the SELECT statement to extract month from the date column
- There are two launch records that had failed drone ship landing in the year 2015

```
%%sql
select substr("Date", 6, 2) as 'Month', "Landing_Outcome", "Booster_Version", "Launch_Site"
from SPACEXTABLE where "Landing_Outcome" = 'Failure (drone ship)' and substr("Date", 0, 5) = '2015';
```

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

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

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

- Used the WHERE clause in the SELECT statement to filter on date column
- Used GROUP BY clause in the SELECT statement to group all the launch records by the launch outcomes
- Used the ORDER BY clause in the SELECT statement with keyword DESC to sort the results in descending order of count
- Used the function COUNT in the SELECT statement to calculate the number of launches for each launch outcome

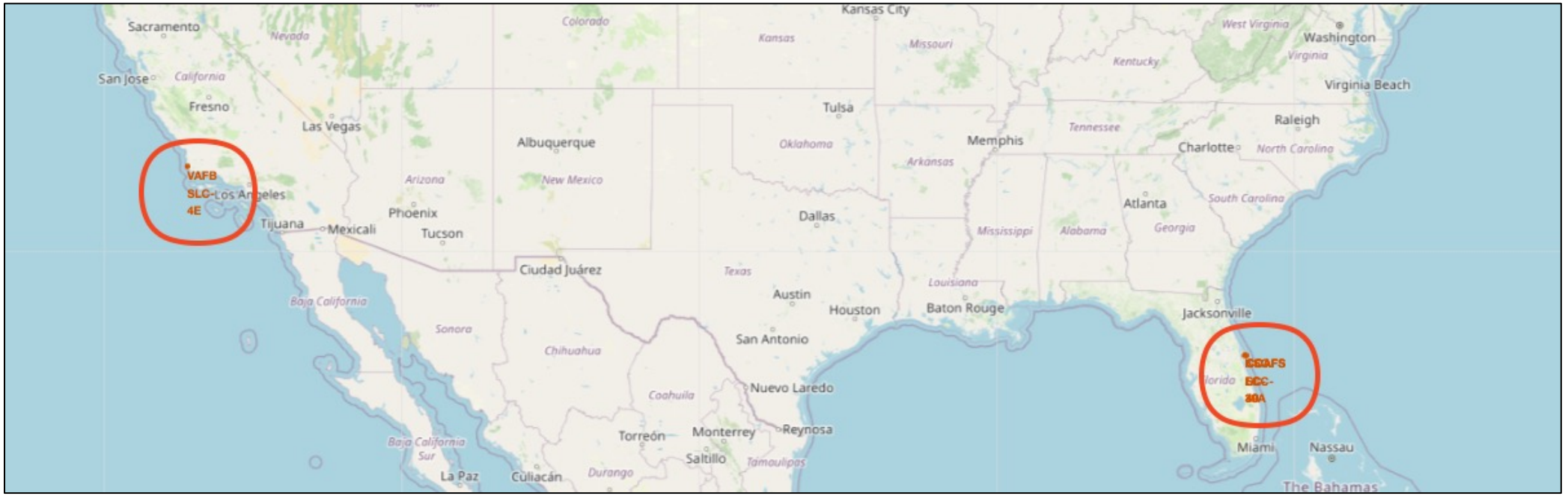
```
%%sql
select "Landing_Outcome", count("Landing_Outcome") as 'Count'
from SPACEXTABLE where "Date" between '2010-06-04' and '2017-03-20'
group by "Landing_Outcome" order by "Count" Desc;
```

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

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

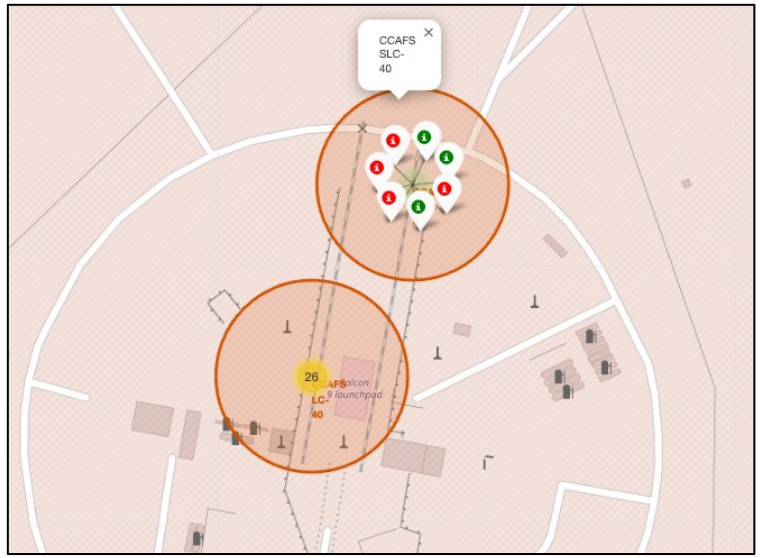

Section 3
Launch Sites
Proximities Analysis

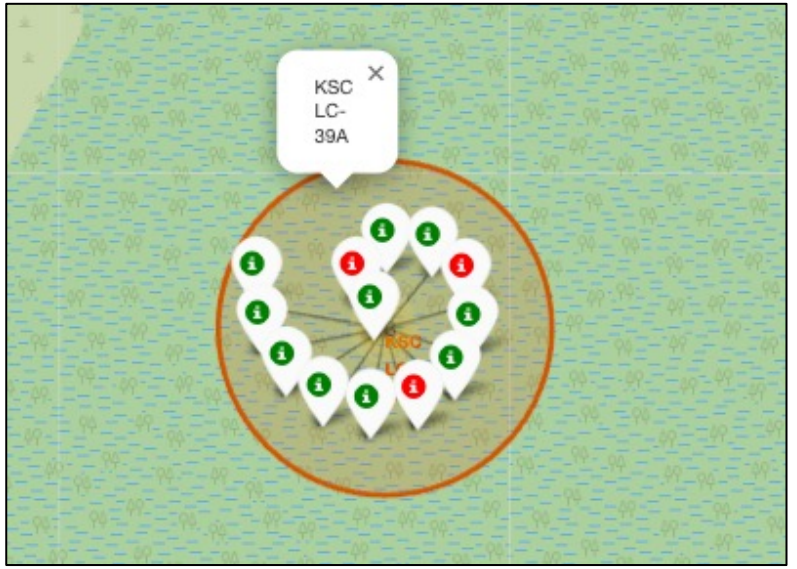
Launch Site Locations

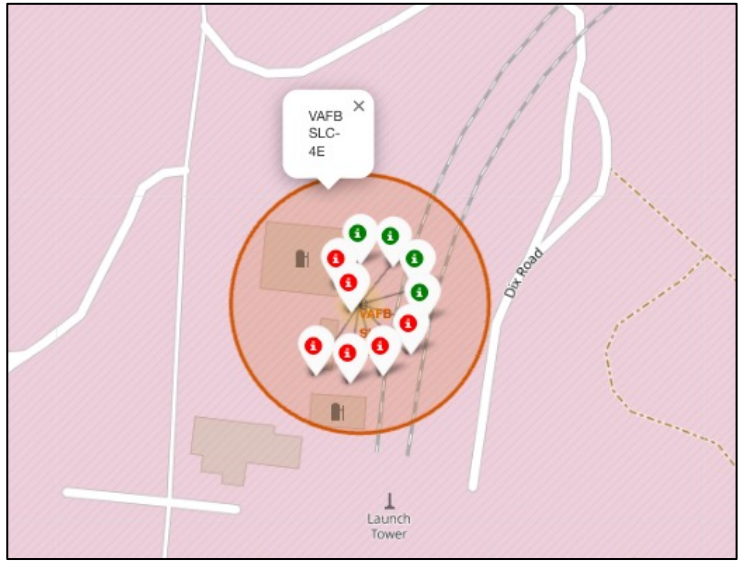


- Four launch site locations have been marked on map – CCAFS LC-40, CCAFS SLC-40, KSC LC-39A and VAFB SLC-4E
- All the launch sites are on East coast and close to equator except for VAFB SLC-4E which is on West coast and little further from the equator
- CCAFS LC-40 and CCAFS SLC-40 are in very close proximity to each other

Outcomes by Launch Site

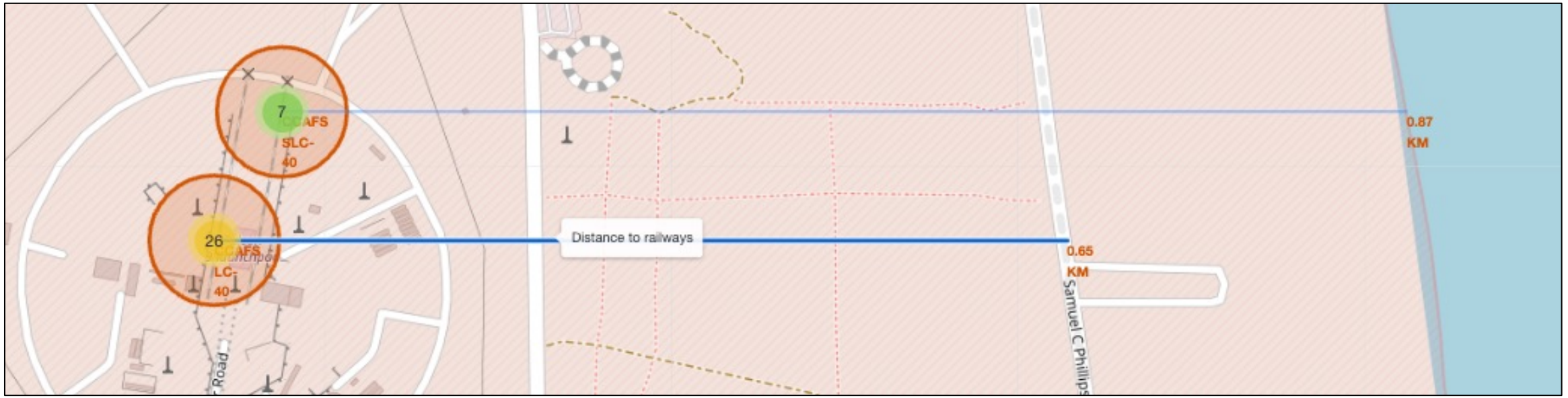




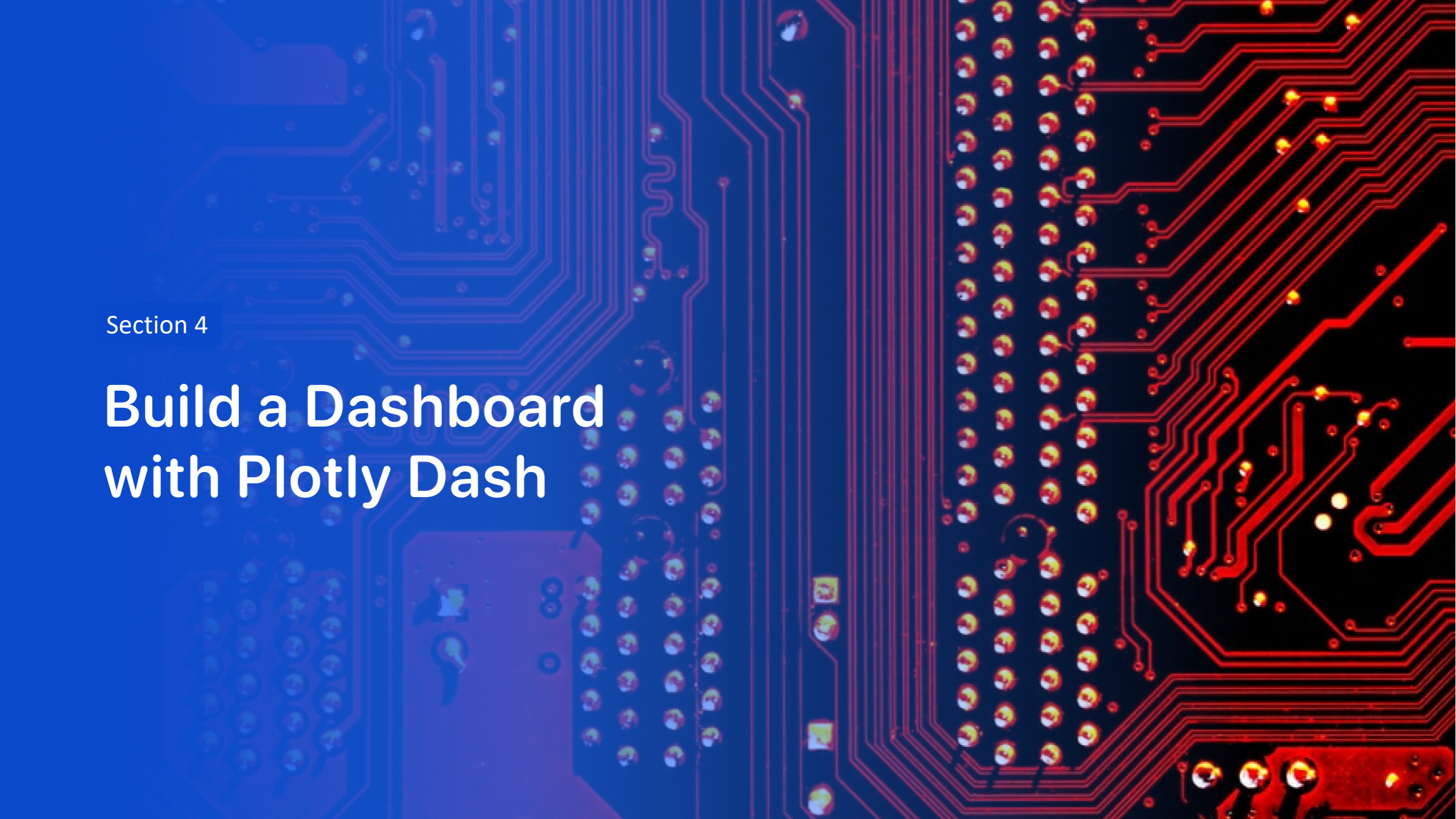


- CCAFS SLC-40 and CCAFS LC-40 are in very close proximity to each other
- CCAFS LC-40 has the highest number of launches
- KSC LC-39A has the highest success rate

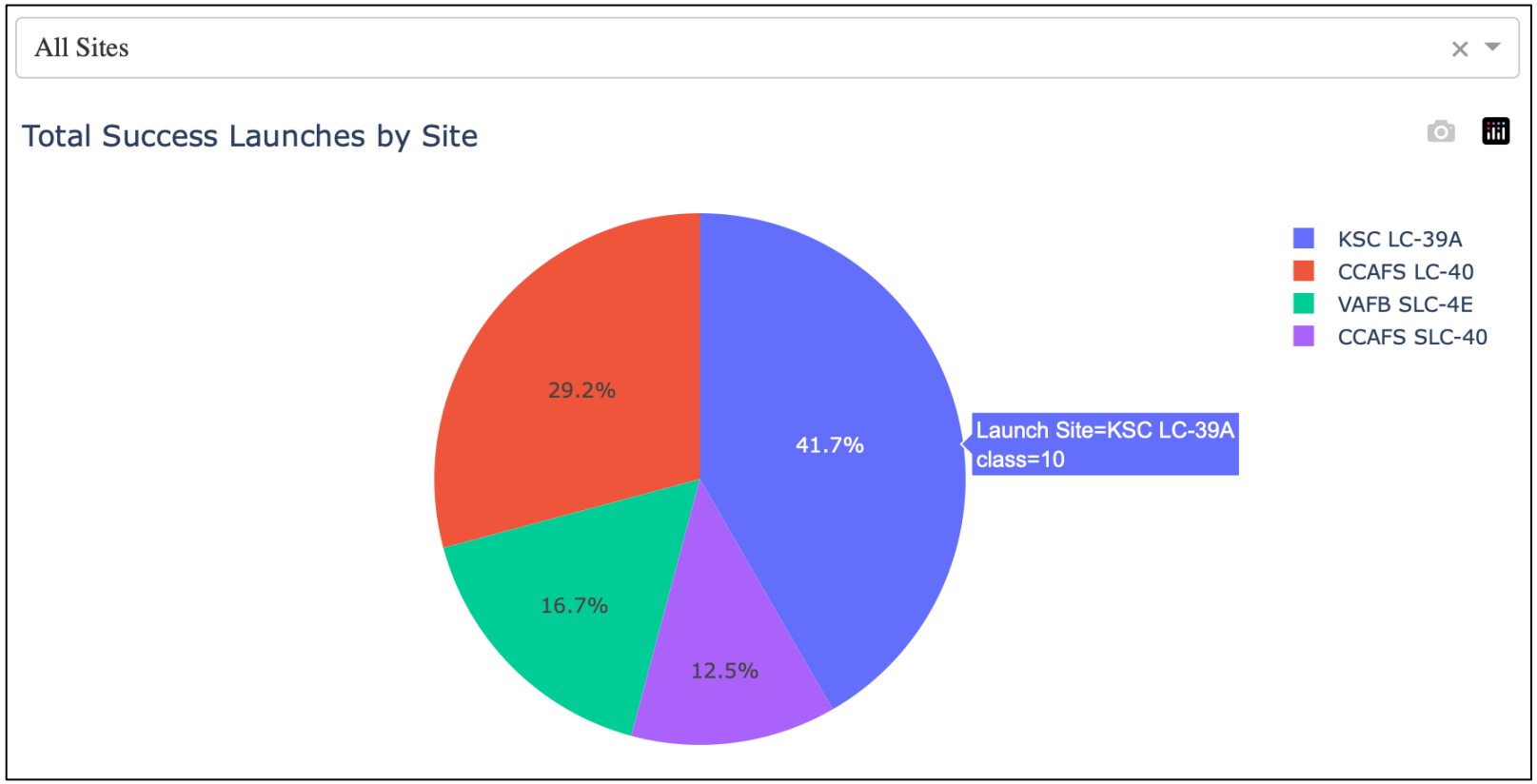
Launch Site and Proximities



- All launch sites are near a means of transport (highways or railways)
- All launch sites are close to a water body
- CCAFS SLC-40 is less than a KM away from a railway line and the coast

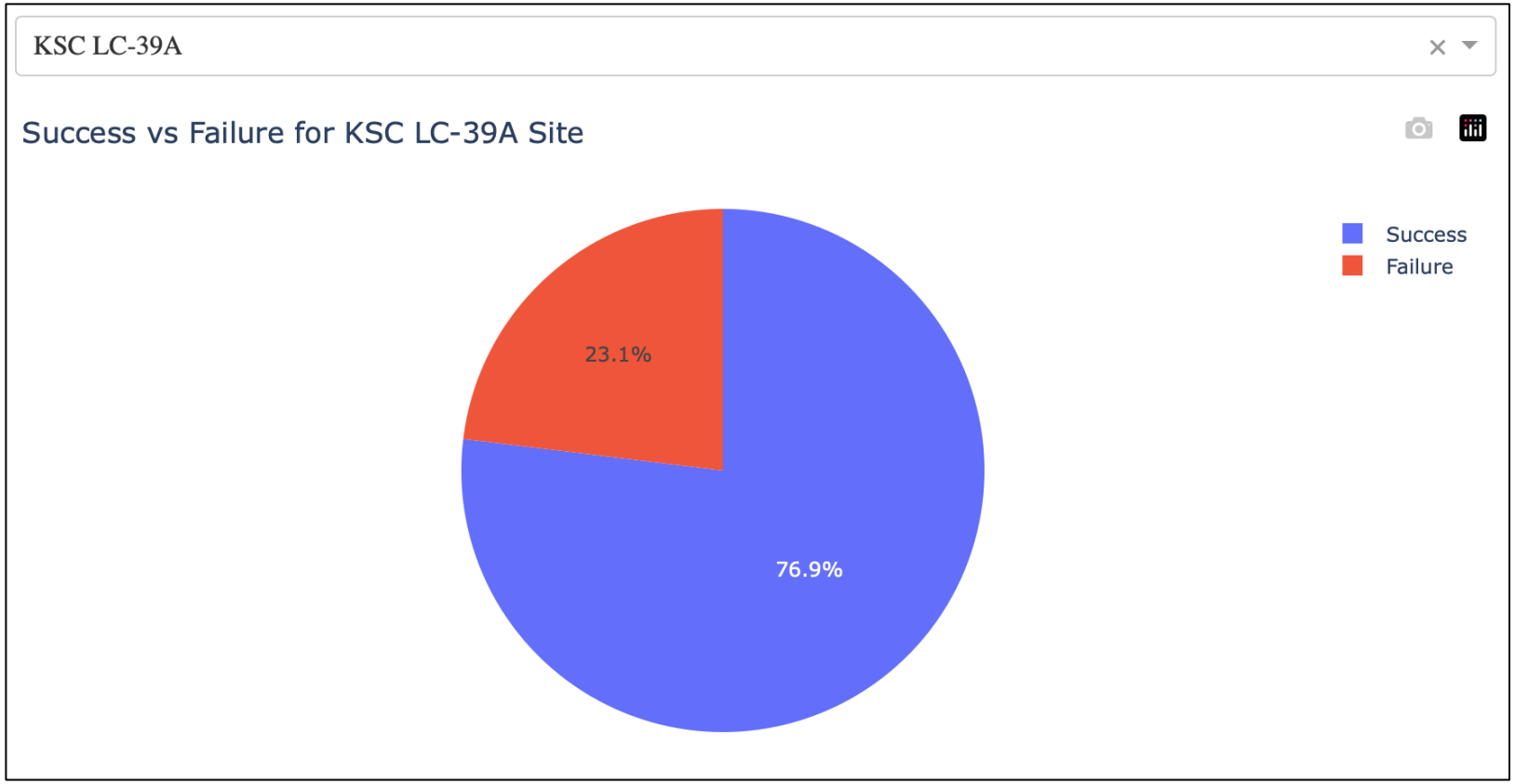

Section 4
Build a Dashboard with Plotly Dash

Launch Success by Sites



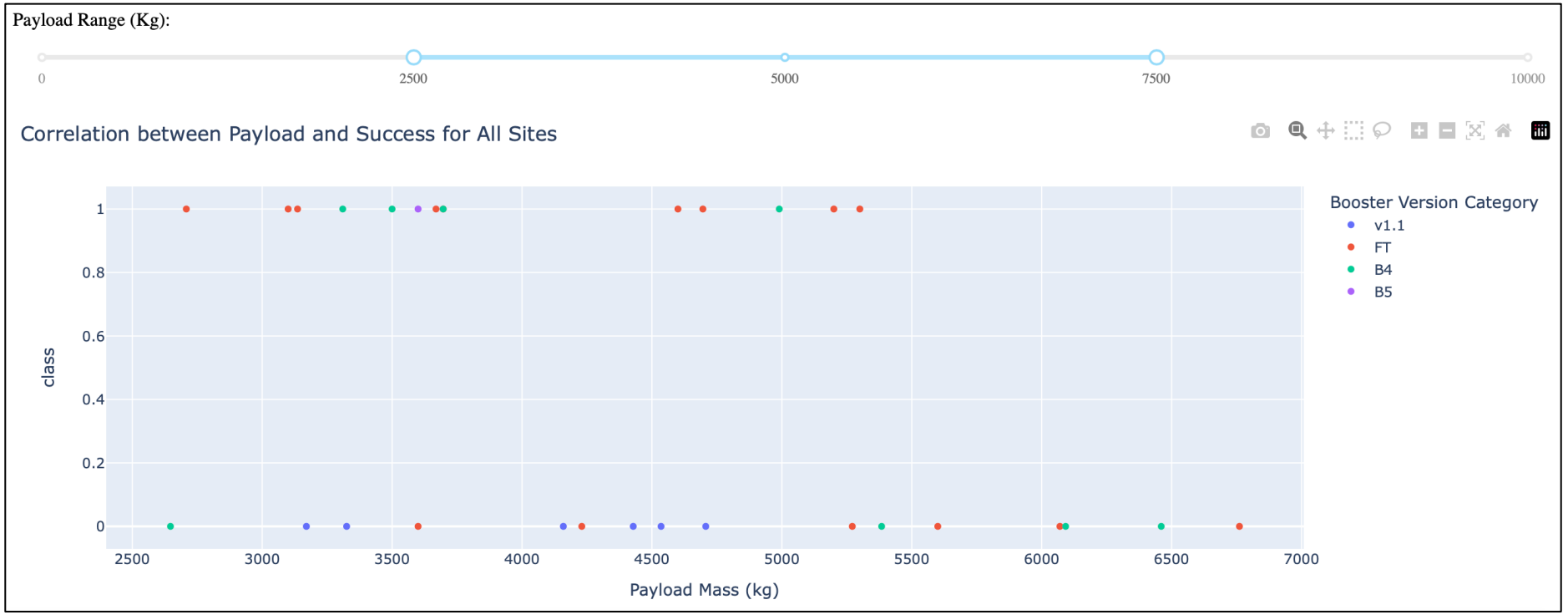
- KSC LC-39A accounts for the 41.7% of total successful landings (highest share)
- CCAFS SLC-40 accounts for the 12.5% of total successful landings (lowest share)

KSC LC-39A – Success vs. Failure



- KSC LC-39A site has been used for 13 launches
- 77% of the launches had successful landings
- 23% of the launches had failed landings

Payload vs. Outcome – All Sites



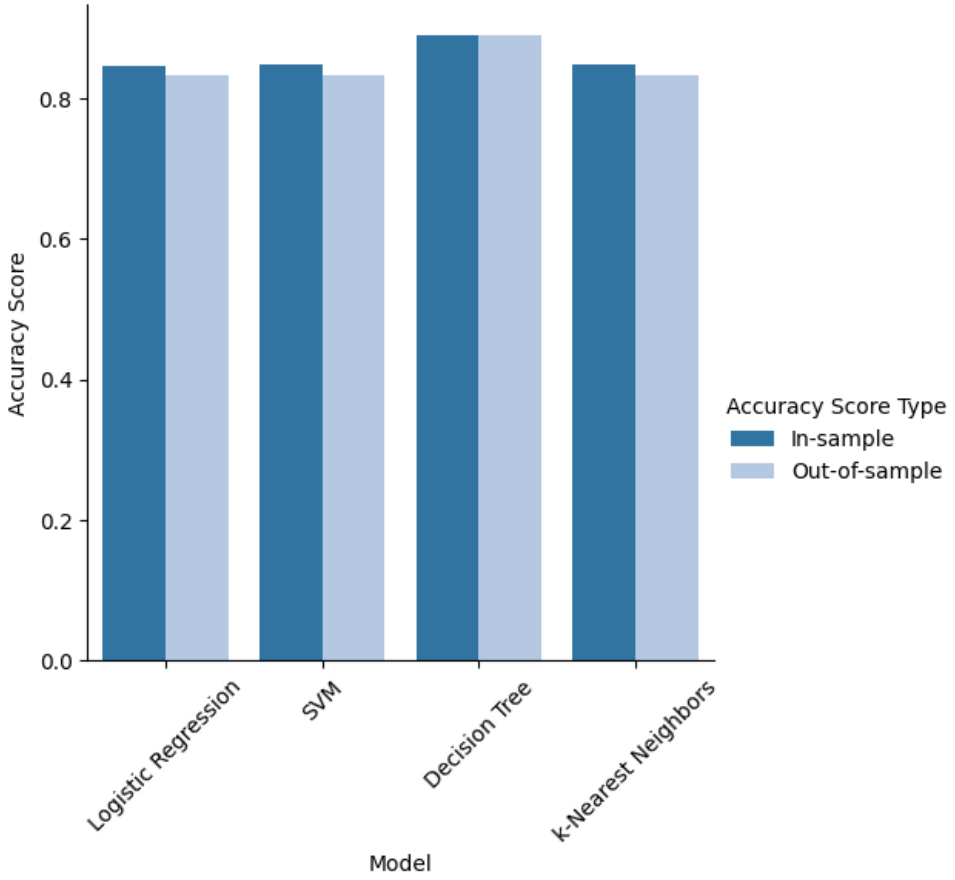
- For the medium payload range (2,500 kgs to 7,500 kgs), the number of failed landings exceed the number of successful landings
- FT booster version has the highest success rate for the medium payload range

Section 5

Predictive Analysis (Classification)

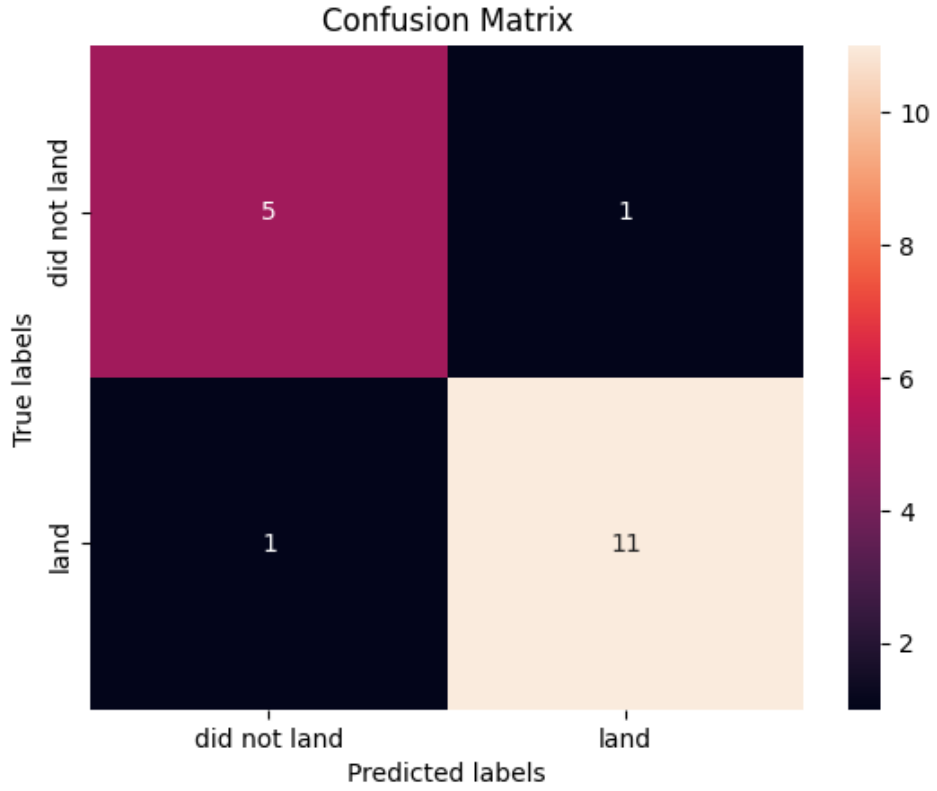
Classification Accuracy

- Decision tree classifier has the highest in-sample and out-of-sample accuracy score of 88.93% and 88.89% respectively
- Logistic regression classifier has the lowest in-sample accuracy score of 84.64%
- All the classifiers (except Decision tree) have the same out-of-sample accuracy score of 83.33%



Confusion Matrix

- **Decision Tree** classifier is selected as the best classifier as it has the highest out-of-sample accuracy score
- The model predicts the cases of successful landings with high accuracy (only 1 false negative for 12 actual positive labels)
- The model predicts the cases of failed landings with good accuracy score (1 false positive for 6 actual negative labels)



Conclusions

- To provide SpaceY an edge in bidding for future space launches, a Decision Tree classifier has been created which will be able to predict (with an accuracy of 88.89%) whether their competitor SpaceX will be able to successfully land the first stage of the launch.

Appendix

- SpaceX API endpoints:
  - Past launches: https://api.spacexdata.com/v4/launches/past/
  - Booster versions: https://api.spacexdata.com/v4/rockets/
  - Launchpad: https://api.spacexdata.com/v4/launchpads/
  - Payloads: https://api.spacexdata.com/v4/payloads/
  - Cores: https://api.spacexdata.com/v4/cores/

- Wikipedia webpage: https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

Thank you!