IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Martyn Ben Ami
29th October 2022

Outline

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

Executive Summary

- Data collection using calls from the publicly available data through the SpaceX API
- Web scraping using the SpaceX website for Launch information, putting useful collumns into a Pandas data frame.
- Data wrangling by means of filtering data based on mission parameters and outcomes
- EDA with SQL using IBM db2 API to store and query data to identify which rockets had more successful missions and which carried the biggest payloads.
- EDA with interactive visual analysis using Folium to display in context of the earth where missions were successful and what parameters are requirements for a launchsite i.e proximity to the public and logistical resources
- EDA with interactive visual analysis using Dash to be abe toquicklyaccess different information about the launch sites and payloads without having to make a new graph for every paraeter, one can select the reas of interest and get an impression using appropriate visaul tools
- Machine learning using scikit-learn to develop, compare select the best predictive model trained and teted on the cleaned and collected data from SpaceX launches
- In summary our model can predict launch outcome based on launch site location, payload and rocket used to an accuracy of more than 0.85

Introduction

- SpaceY wants to join the spacerace in competition with SpaceX to do that, we have to determine the state of the art and other factors involved to best use the technology.
- Goal of this project is to find, sort and use data to determine What factors, when combined resulted in the most successful launch missions for spacex to not have to go through the same trial and error and get a competitive advantage from their efforts. We want to avoid reinventing the wheel.
- To do this we will use publicly available historical data and machine learning to build a model that determines which factors are most important for successful launches.

Section 1

Methodology

Methodology

Executive Summary

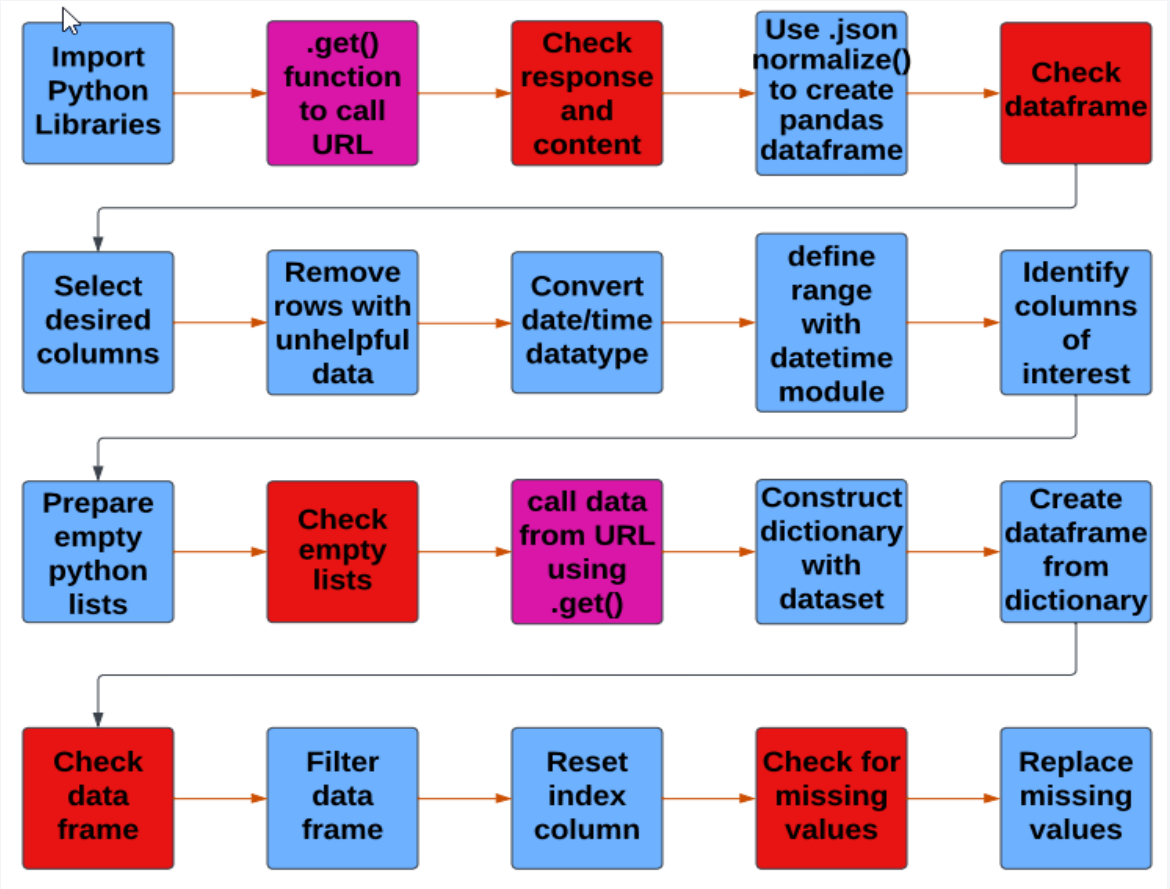
- Data collection methodology:
  - Data was collected from the SpaceX website using python, wget and requests libraries.
- Perform data wrangling
  - Data processed using python and put into pandas dataframes then cleaned and organised
- Perform exploratory data analysis (EDA) using visualization and 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

- Data was collected from the SpaceX website using python, wget and requests libraries
- Data collection process followed princple step laid out in flowcharts in the next slide

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

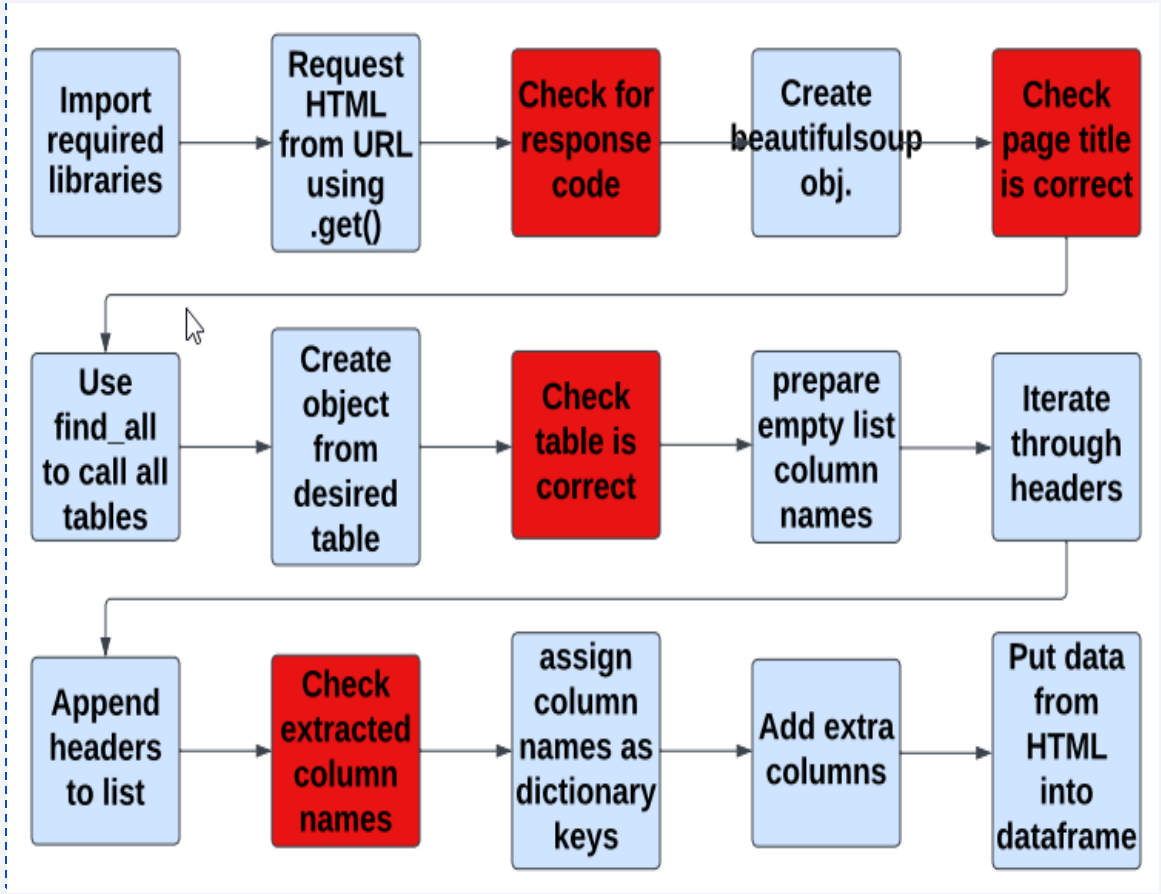


- https://github.com/Tex6298/SpaceY/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- Extract a Falcon 9 records table from wikipedia.
- Parse the table and convert it into a Pandas data frame.

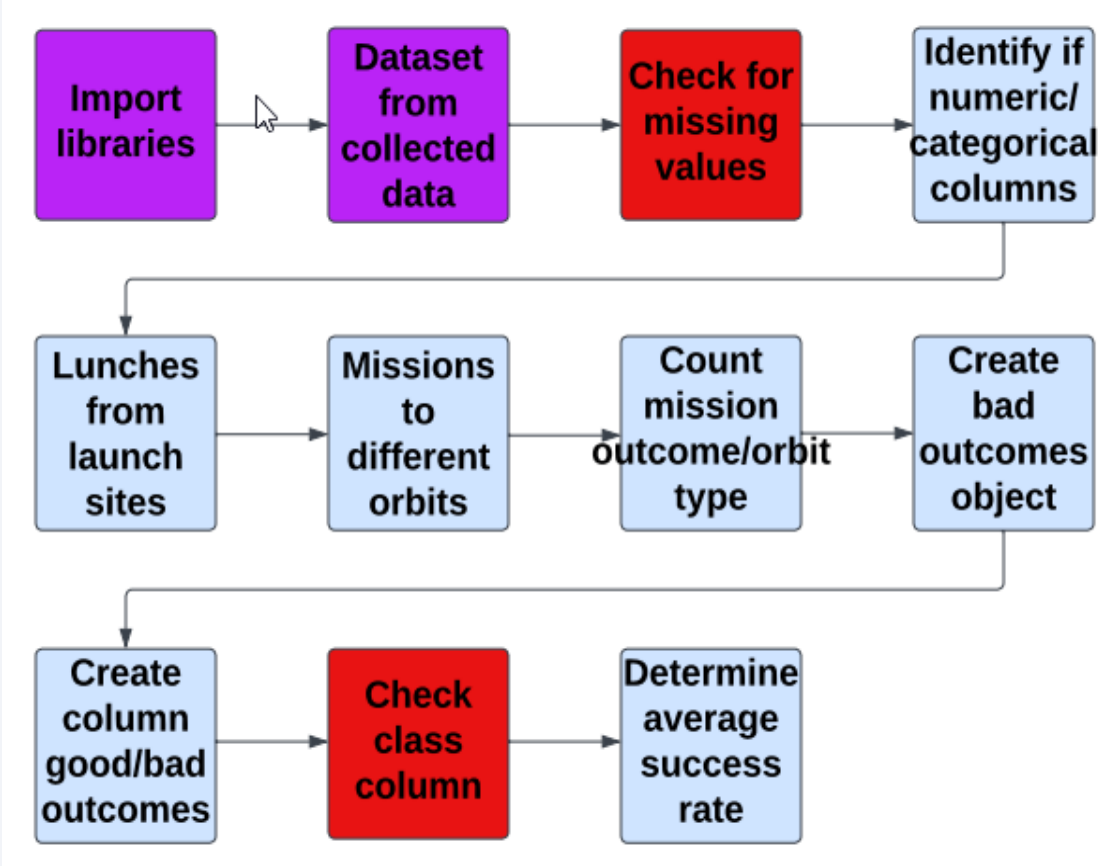
- https://github.com/Tex6298/SpaceY/blob/main/Lab_1b_jupyter-labs-webscraping.ipynb



Data Wrangling

- Data processed put into pandas dataframes then cleaned and organised for exploratory data analysis and training labels were determined.

- https://github.com/Tex6298/SpaceY/blob/main/Lab_2_Data_Wrangling.ipynb



EDA with Data Visualization

- Exploratory Data Analysis and Feature Engineering using matplotlib and seaborn.
- Produced category graphs and bar graphs demonstrate relationships between launch attempt, launch site, payload, orbit type and mission success.
- Category graphs are used to compare two variables directly while simultaneously showing a third categorical variable as colour. This makes it obvious if a combination of variables frequently resulted in a succesful mission or not.
- Bar graphs are good for comparing how often variables resulted in successful mission outcomes
- https://github.com/Tex6298/SpaceY/blob/main/Lab_4_EDA_with_Data_Visualisation.ipynb

EDA with SQL

- load the SQL extension and establish a connection with SpaceX data on IBM db2 database.
- Display the names of the unique launch sites.
- Display records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- Date when the first successful landing outcome in ground pad was acheived.
- List the boosters landing at sea and payload mass between 4000 and 6000

EDA with SQL

- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Rank 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

- https://github.com/Tex6298/SpaceY/blob/main/Lab_3_EDA_with_SQL.ipynb

Build an Interactive Map with Folium

- Created and added objects to a folium map to get visual cues to see which geographic and infrastructure features.
- All launch sites were marked with dot and circles to mark the surrounding area.
- Failed and successful launches were marked at each launch site using cluster.
- Distances to coastlines, railways, highways and cities were noted and marked with lines to give a sense of the distances.

- https://github.com/Tex6298/SpaceY/blob/main/Lab_5_Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Build a Dashboard with Plotly Dash

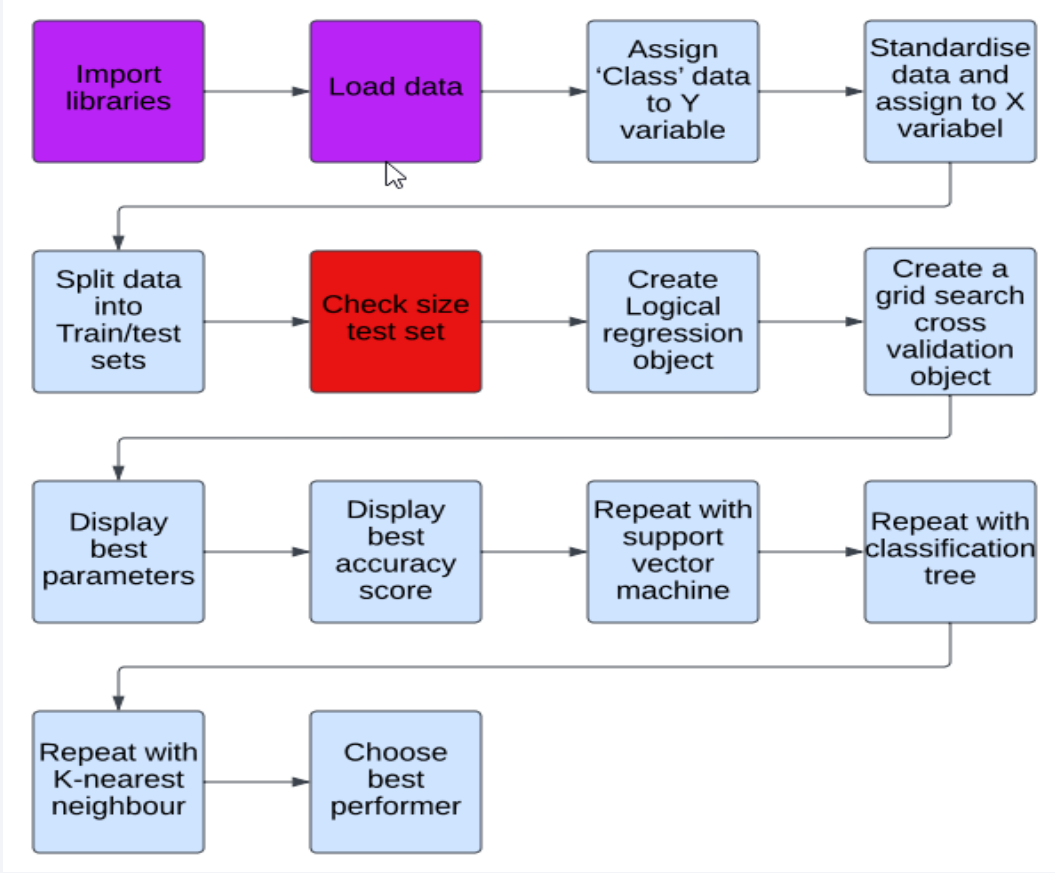
- Launch Site Drop-down Input Component with complementary callback function to render `success-pie-chart` based dropdown selection for dynamic viewing of pie charts of different launch sites.
- Range Slider to Select Payload with complementart callback function to render the `success-payload-scatter-chart` scatter plot for dynamic viewing of success rates over custom payloads.

- https://github.com/Tex6298/SpaceY/blob/main/Lab_5b_%20spacex_dash_app.py

Predictive Analysis (Classification)

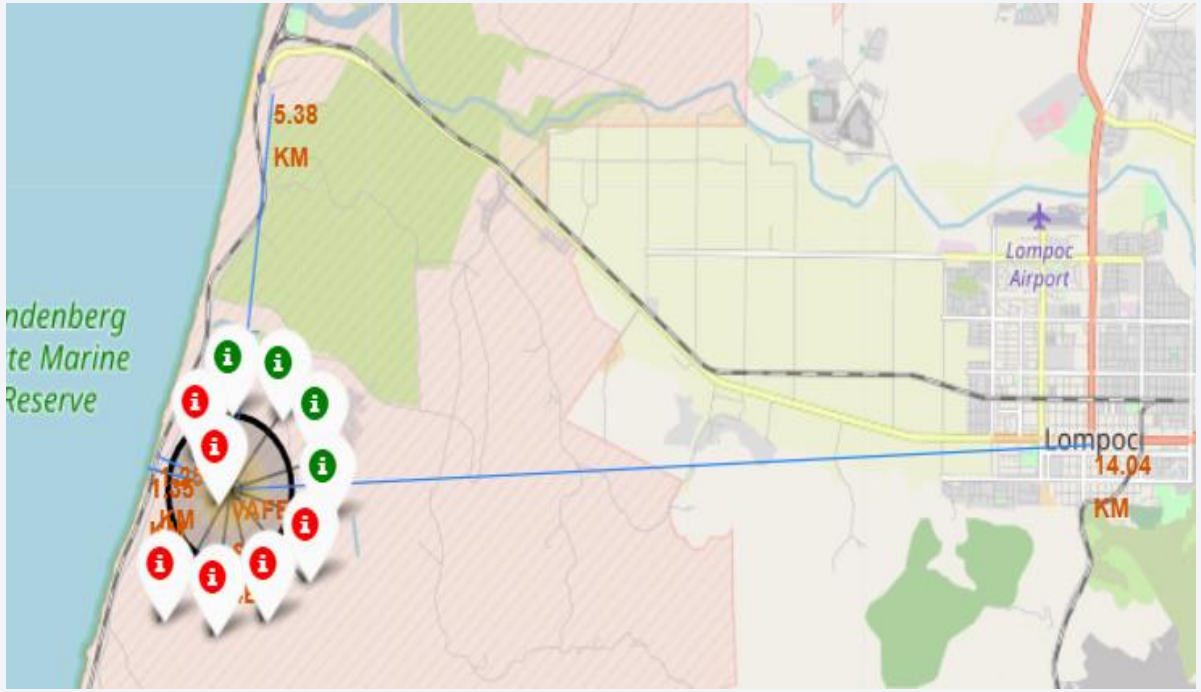
- Mission Data were prepared as input and output variables and split into training and test data.
- Data was fit to different ML strategies
- Best parameters were found using cross-validation grid search

- https://github.com/Tex6298/SpaceY/blob/main/Lab_6_Machine_Learning_Predicitон_Lab.ipynb



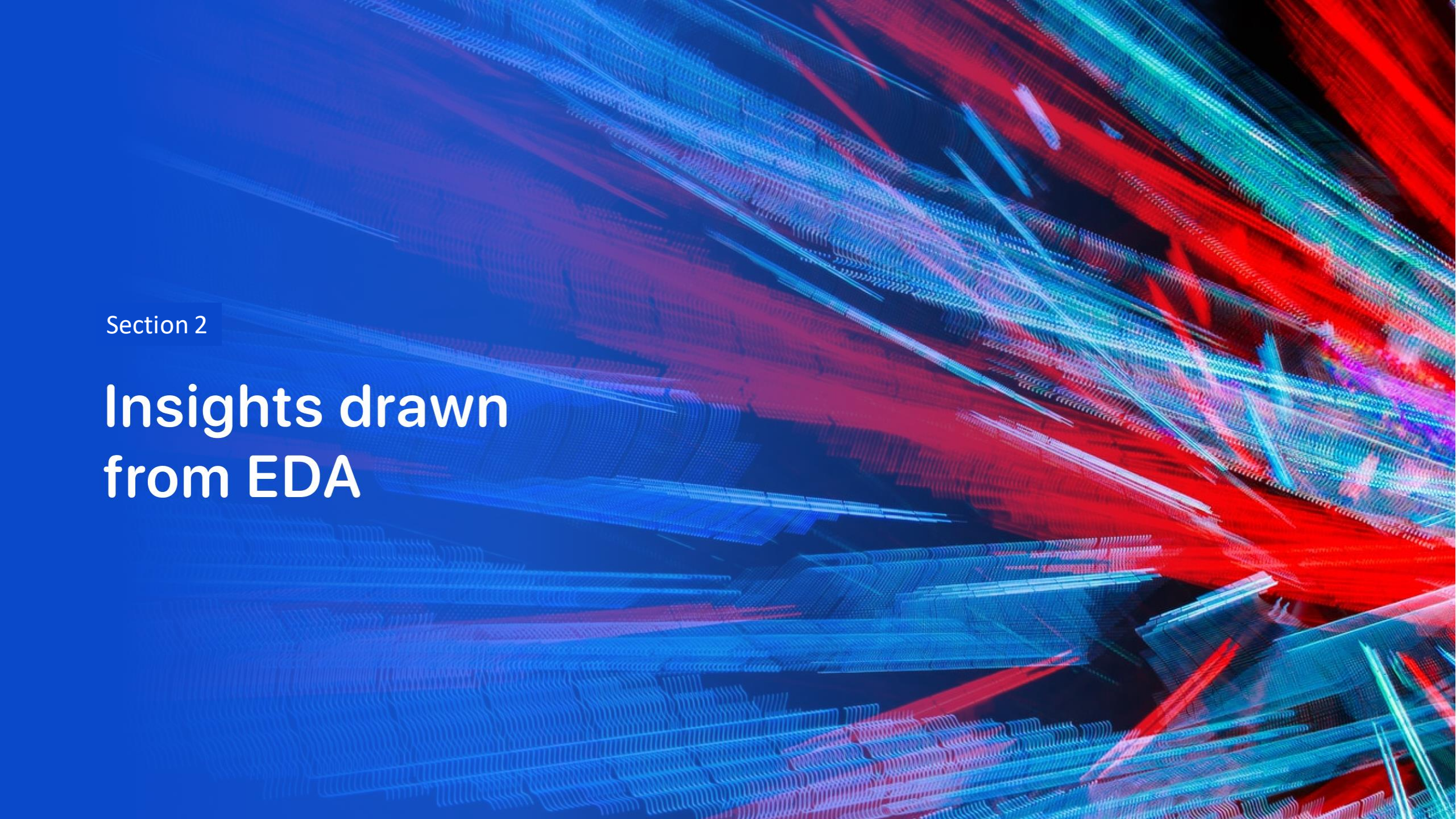
Results

- Average success rate 0.66, Success rate improved year on year,
- ES-L1, GEO, HEO & SSO missions had 100% success rate
- LEO success rate improved with time while there is no evidence of improvements for GTO
- Heavy payloads was correlated with higher success rate but not for GTO
- No heavy payload missions were launched from VAFB-SLC launch
- All predictive models were very good with an accuracy between 0.85 and 0.9
- Best predictive analysis results were from classification tree method with an accuracy of 0.889



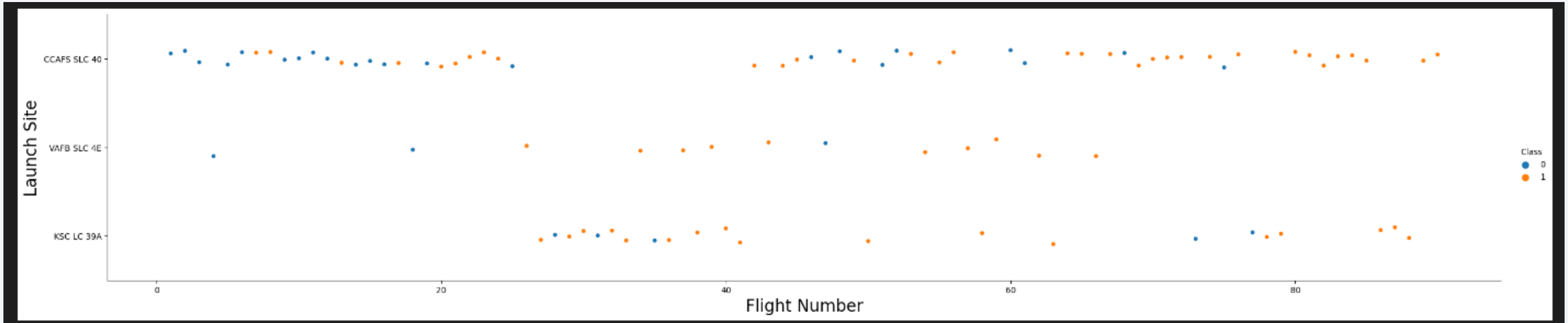
Section 2

Insights drawn from EDA



Flight Number vs. Launch Site

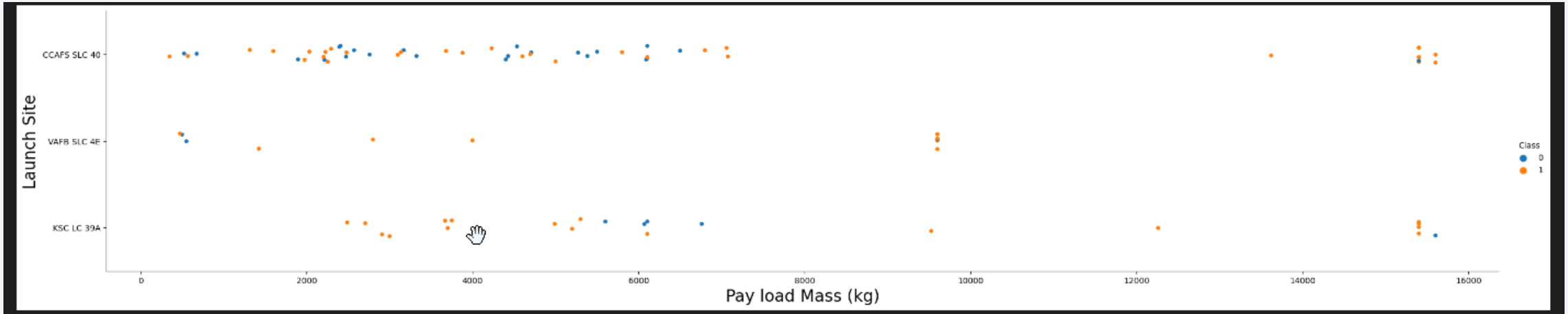
Scatter plot of Flight Number vs. Launch Site



At first Launches were almost exclusively from CCAFS SLC 40 then From KSC LC 39A for a time but then the trend was back to CCAFS SLC 40 with about a third split between the other two launch sites. In general number of successful launches improved.

Payload vs. Launch Site

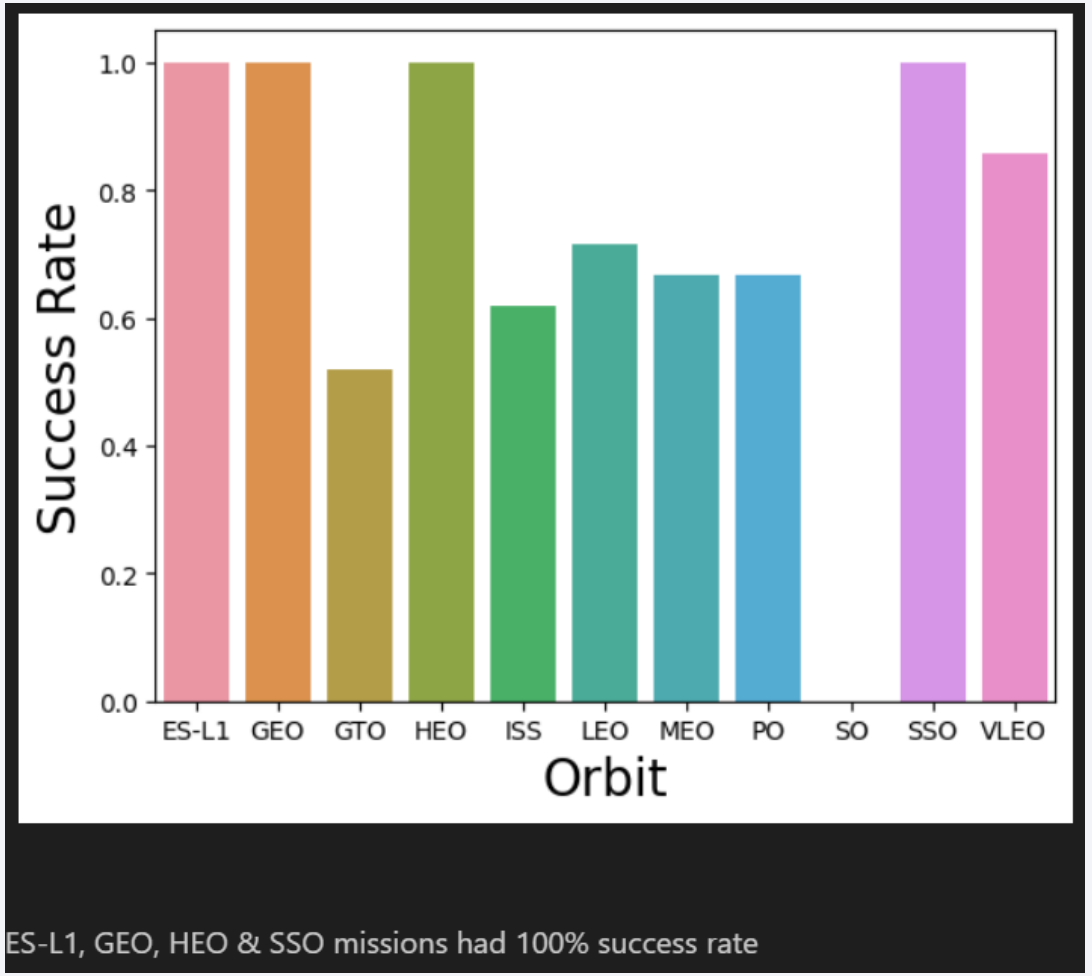
Scatter plot of Payload vs. Launch Site



Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type

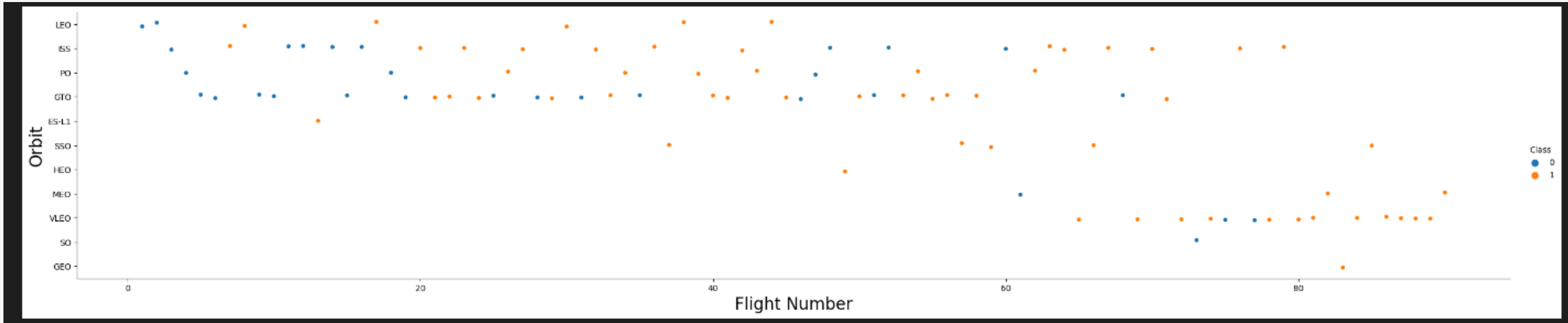
Bar chart for the success rate of each orbit type



ES-L1, GEO, HEO & SSO missions had 100% success rate

Flight Number vs. Orbit Type

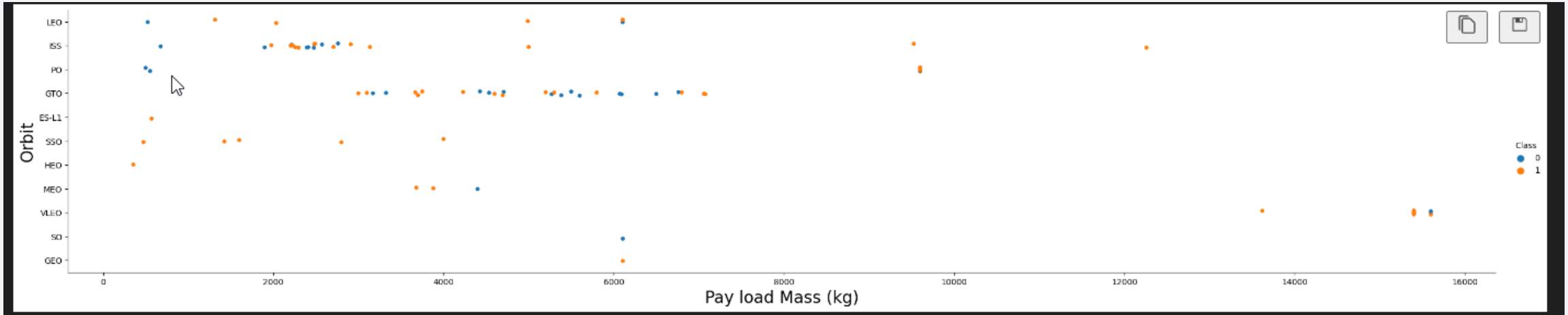
Scatter plot of Flight number vs. Orbit type



You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type

- Scatter plot of payload vs. orbit type

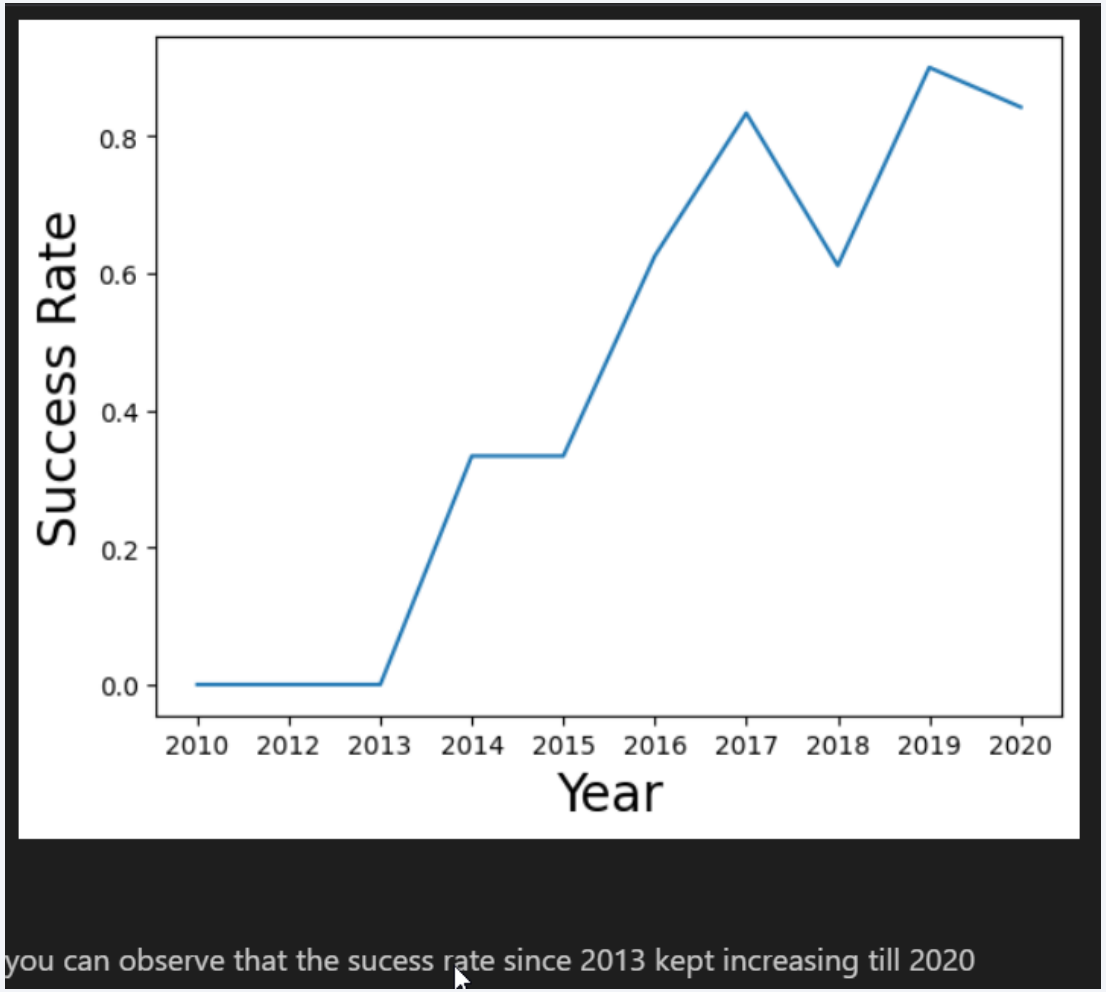


With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend

Line chart of yearly average success rate



All Launch Site Names

```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEX
```

Python

```
* ibm_db_sa://hln12668:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/BLUDB
Done.
```

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

SQL query was used to list the individual launch site names.

Launch Site Names Begin with 'CCA'

```
%sql SELECT LAUNCH_SITE FROM SPACEX WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5 ;
```

Python

```
 * ibm_db_sa://hln12668:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/BLUDB
Done.
```

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

5 results were called using the LIMIT function

Total Payload Mass

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEX WHERE CUSTOMER = 'NASA (CRS)';
```

Python

```
 * ibm_db_sa://hln12668:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/BLUDB
Done.
```

1
45596

Total payload mass was calculated using the SUM() function.

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM spacex WHERE booster_version ='F9 v1.1'
```

Python

```
 * ibm_db_sa://hln12668:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/BLUDB
Done.
```

1
2928

Average payload of F9 v1.1 booster with the AVG() function

First Successful Ground Landing Date

```
%sql SELECT MIN(date) FROM spacex where landing__outcome = 'Success (ground pad)';
```

Python

```
 * ibm_db_sa://hln12668:***@ea286ade-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/BLUDB
Done.
```

1
2015-12-22

First successful landing listed using MIN() function on the date column

Successful Drone Ship Landing with Payload between 4000 and 6000

booster_version
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1014
F9 v1.1 B1016
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1030
F9 FT B1021.2
F9 FT B1032.1
F9 B4 B1040.1
F9 FT B1031.2
F9 B4 B1043.1
F9 FT B1032.2
F9 B4 B1040.2
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5 B1046.3
F9 B5B1054
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
%sql SELECT BOOSTER_VERSION FROM spacex WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

Python

```
 * ibm_db_sa://hln12668:***@ea286ace-86c7-4d5f-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/BLUDB
Done.
```

Payload range was defined using the WHERE modifier

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT mission_outcome, count(mission_outcome) as count FROM spacex GROUP BY(mission_outcome)
```

Python

```
 * ibm_db_sa://hln12668:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/BLUDB
Done.
```

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

Total missions listed by outcome we listed using the Group By query

Boosters Carried Maximum Payload

```
%sql SELECT booster_version FROM spacex WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM spacex);
```

Python

```
 * ibm_db_sa://hln12668:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/BLUDB
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

Boosters that carried maximum payload listed using a subquery

2015 Launch Records

```
%sql SELECT landing__outcome, booster_version, launch_site FROM spacex where DATE LIKE '%2015%' AND landing__outcome = 'Failure (drone ship) ';
```

Python

```
 * ibm_db_sa://hln12668:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/BLUDB
Done.
```

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

2015 launch records were found querying a combination of LIKE and % wildcard symbol

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

```
%sql SELECT landing__outcome FROM spacex WHERE date BETWEEN '06-04-2010' AND '03-20-2017'ORDER by landing__outcome DESC
```

Python

```
* ibm_db_sa://hln12668:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/BLUDB
Done.
```

landing__outcome
Uncontrolled (ocean)
Uncontrolled (ocean)
Success (ground pad)
Success (ground pad)
Success (ground pad)
Success (drone ship)
Success (drone ship)
Success (drone ship)
Success (drone ship)
Success (drone ship)
Precluded (drone ship)
No attempt
No attempt
No attempt
No attempt
No attempt
No attempt
No attempt

No attempt
No attempt
No attempt
No attempt
Failure (parachute)
Failure (parachute)
Failure (drone ship)
Failure (drone ship)
Failure (drone ship)
Failure (drone ship)
Failure (drone ship)
Controlled (ocean)
Controlled (ocean)
Controlled (ocean)

Landing outcomes range was listed using the BETWEEN and were ranked using DESC and ORGER BY query calls

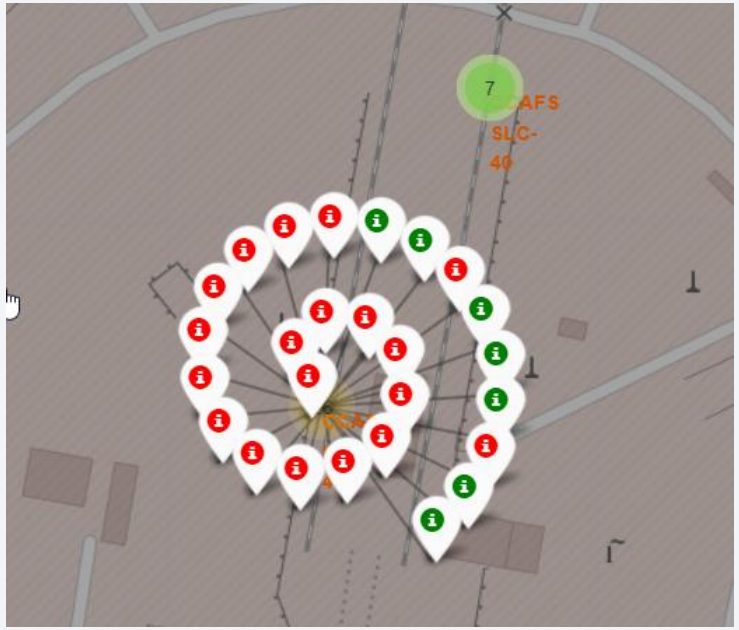

Section 3
Launch Sites
Proximities Analysis

all launch sites on a map

Launch sites are located on far coasts and to the south

Two are too close together to read in a zoomed out map

success/failed launches



Success/failed missions were colour coded and marked using a marker cluster, clicking on the launch site shows the individual missions in red for failed or green for successful

Distance to Important Features from Launch Site

Lines and distance of feature to launch site were marked on the map

Launch sites are between 0.5-1.5km from railways.

Launch sites are between 0.6-5km from highways.

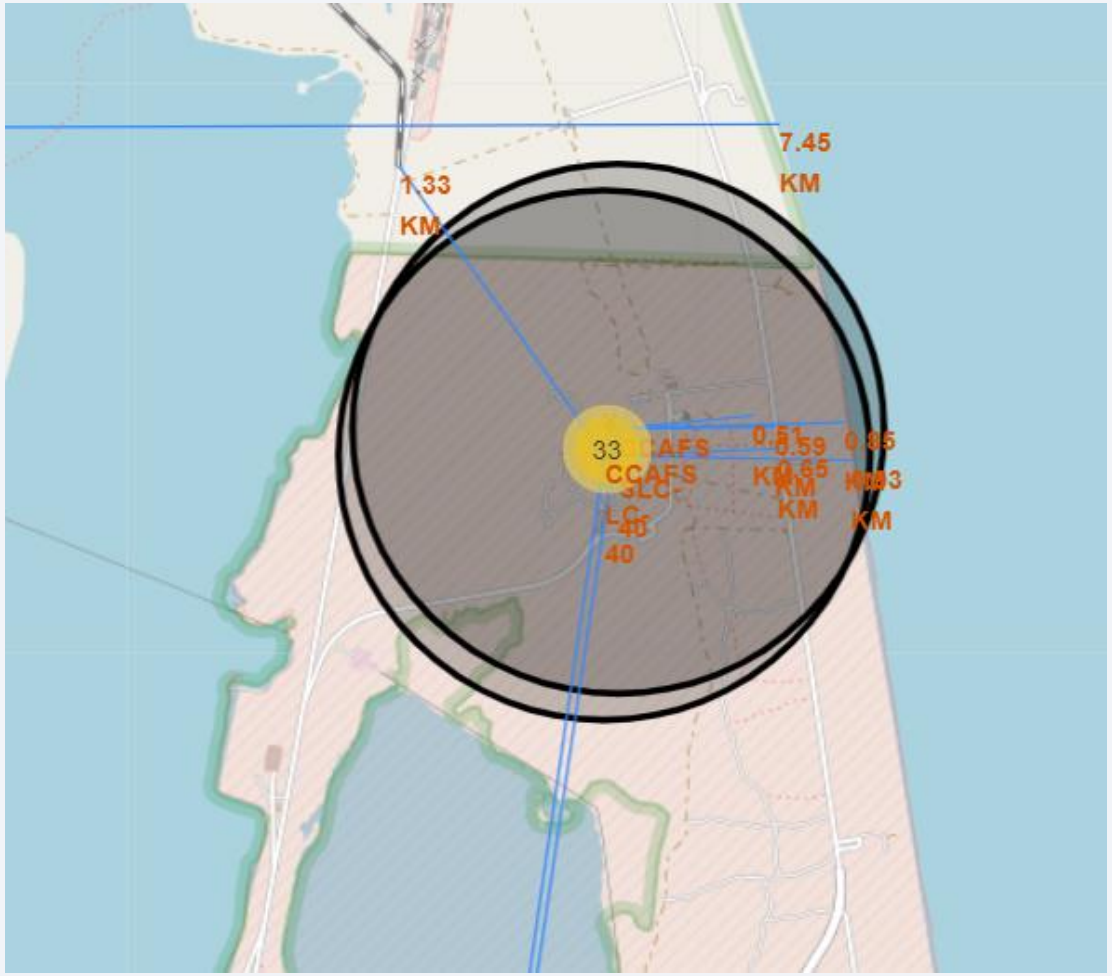
Launch sites are between 0.8-8km from coastline.

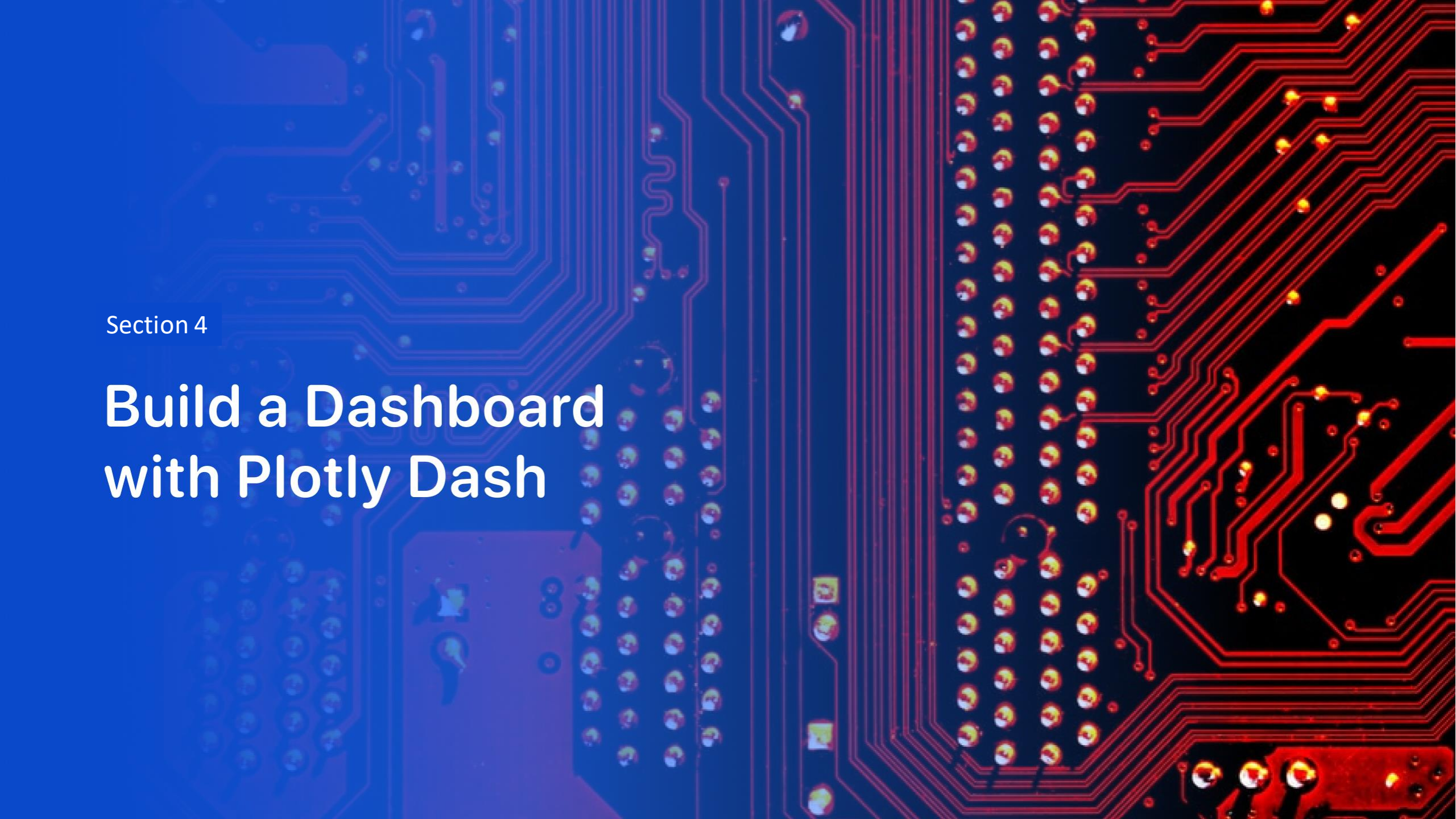
Launch sites are more than 14km from cities.

Railways and highways are probably useful to have nearby for logistics.

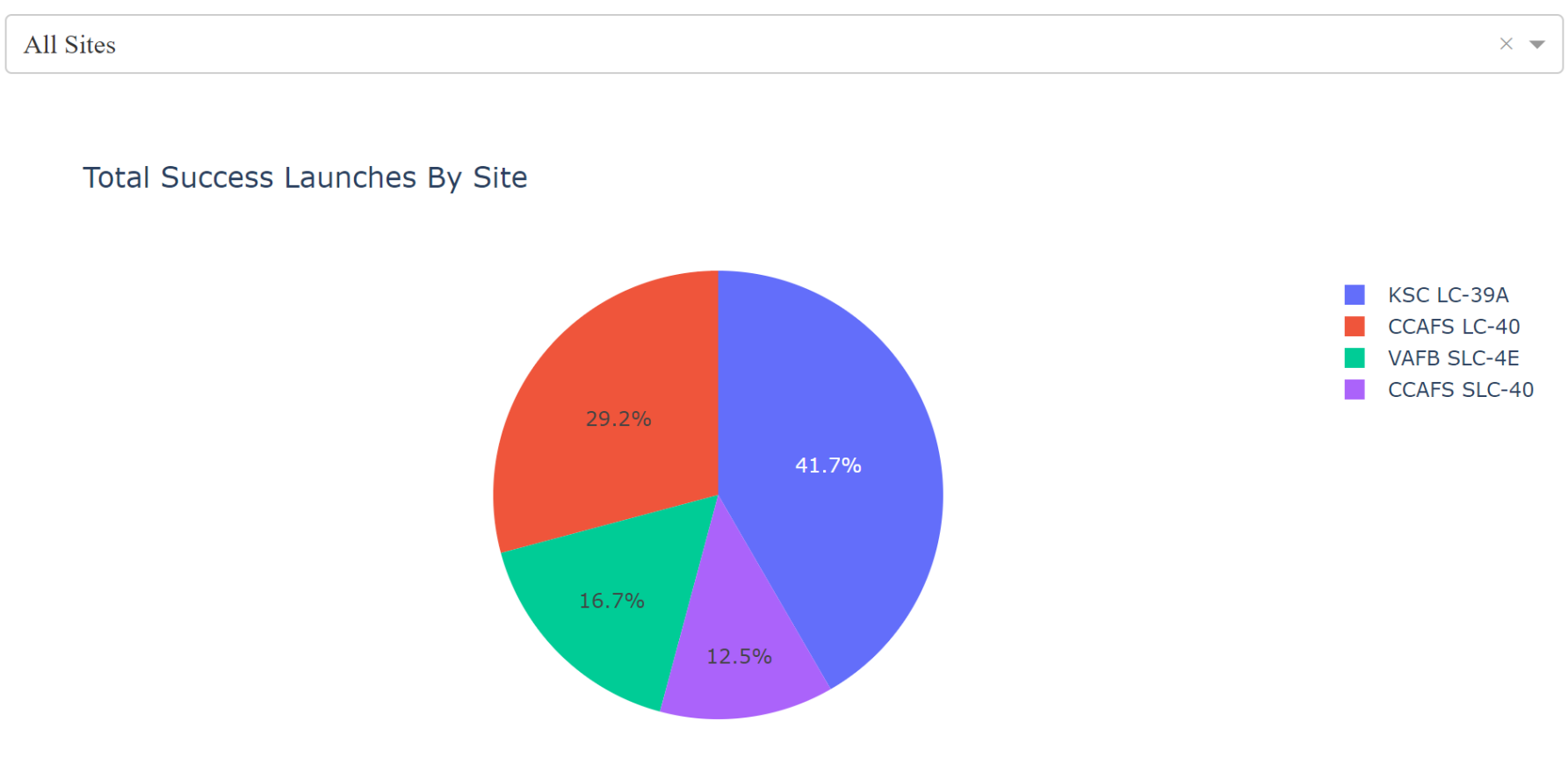
Proximity coastlines probably has to do with the weather/atmosphere and removing a direction hazards can leave or approach a launch site.

Keeping launch sites away from cities is definitely a safety and pollution consideration.



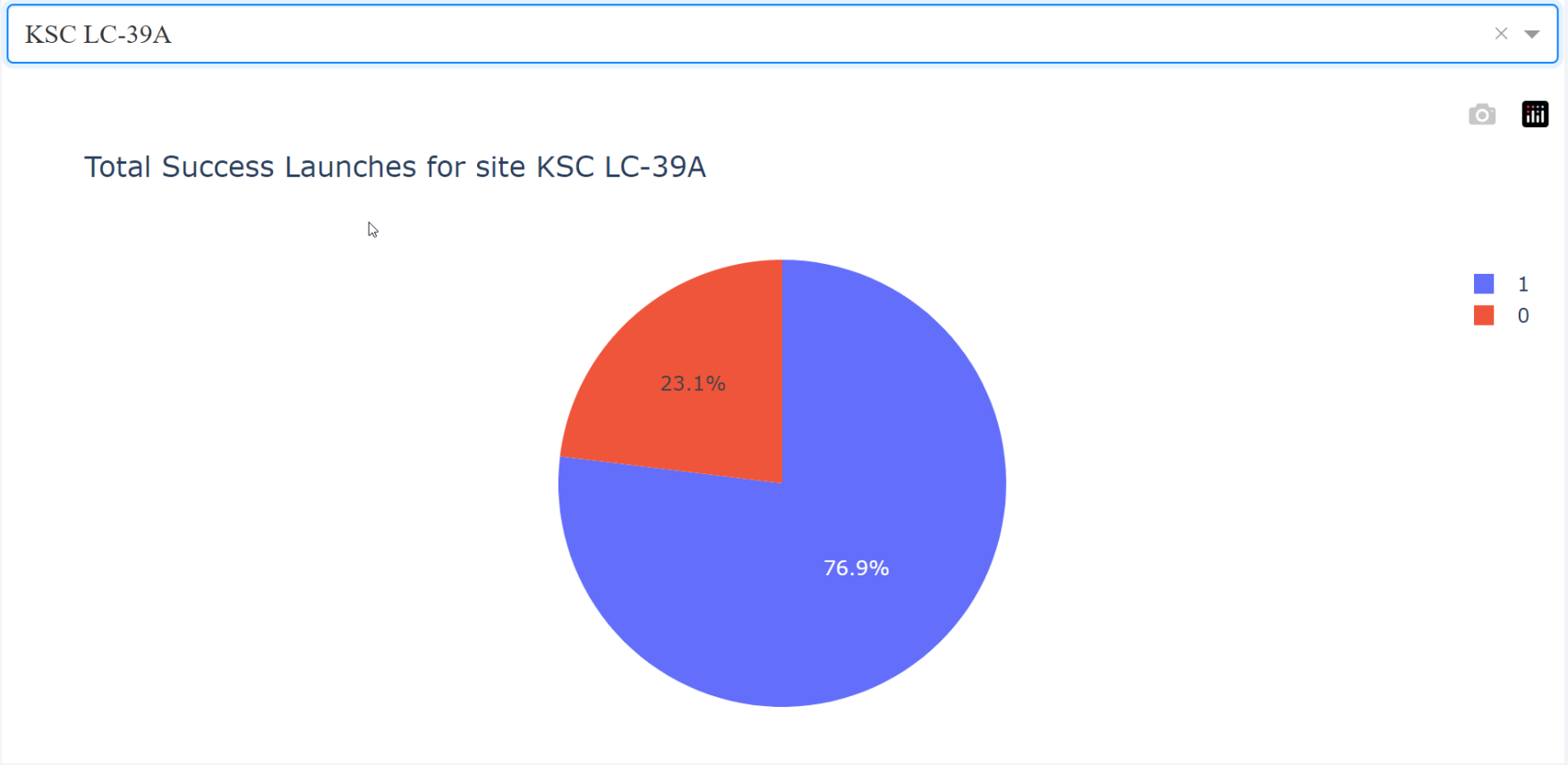

Section 4
Build a Dashboard with Plotly Dash

Percentage of Successful Launch by Launch Site



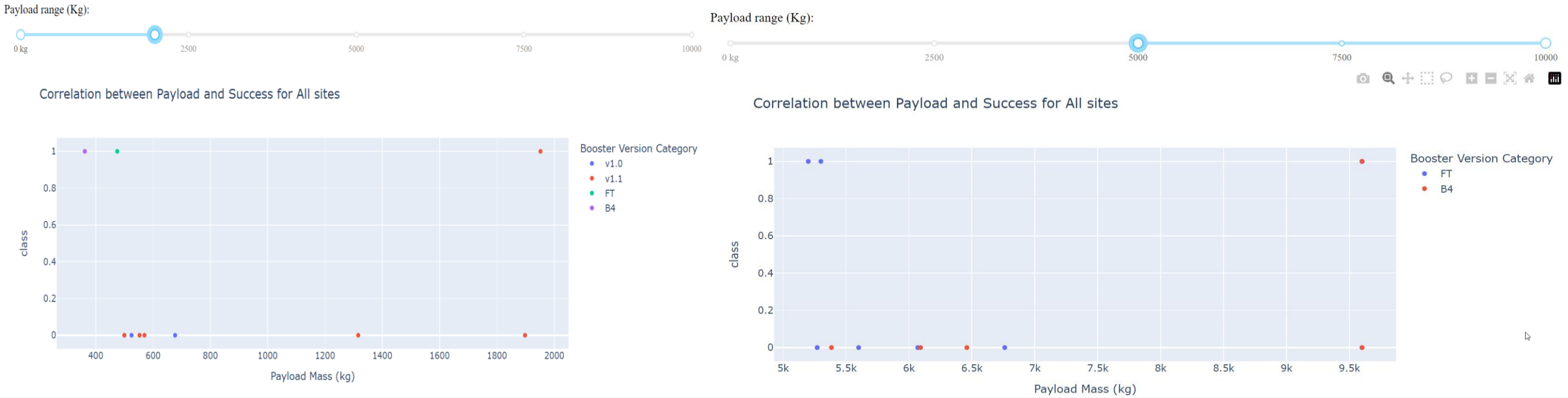
- The Pie chart shows that most of the successful launches were from KSC LC-39A

Percentage of Successful Launches from KSC LC-39A



The site with the most successful launched has a 76.9% success rate

<Dashboard Screenshot 3>



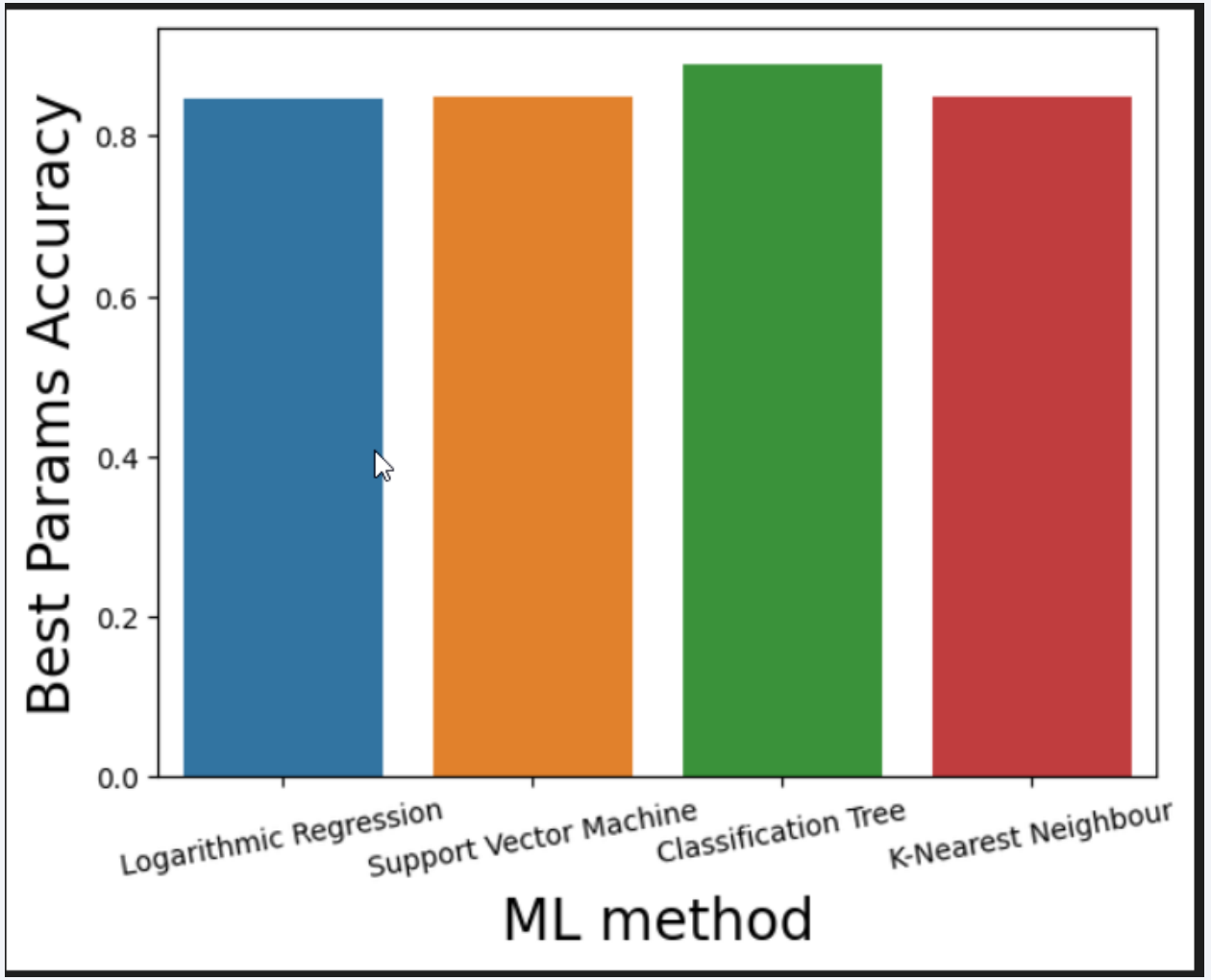
In the high payload range the result was less dependent on the type of rocket than in the low payload range where you see some rockets were only failures and other were almost entirely successful

Section 5

Predictive Analysis (Classification)

Classification Accuracy

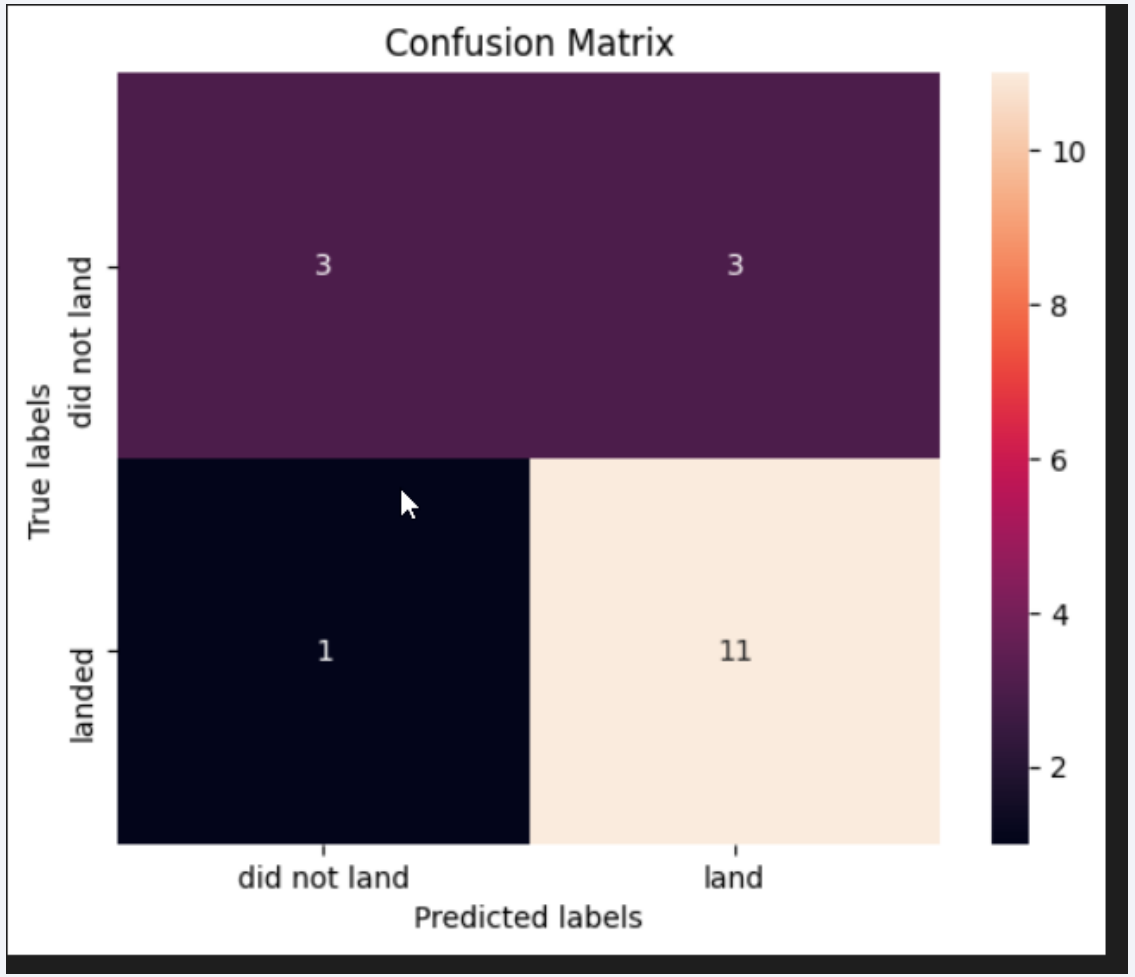
Tree classifier model has the highest classification accuracy



Confusion Matrix

Confusion matrix shows how our tree classifier model does against a test set not used in training

- 11 true positives
- 3 true negatives
- 3 false positives
- 1 false negative



Conclusions

- High payloads result in more successful missions.
- KSC LC-39A is the best launch site with most successful missions.
- Classification tree can be used to predict successful missions for maximum profits as teams get better and knowhow integrates over the work culture
- Mission success increases over time

Appendix

This snippet of code was particularly helpfu for drawing multiple features on the folium maps:

https://github.com/Tex6298/SpaceY/blob/main/shiproadrail.py

Thank you!