IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Muhammad Awais
1/11/2024

Outline

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

Executive Summary

- I’ve to analyze SpaceX Falcon 9 data and predict if it can successfully land the first stage or not. We get this using data analysis, visualization and using machine learning techniques.
- I’ve find the failure rate and compare them based on the launch sites and different other parameters. We also train machine learning models on data for further predictions.
- I also compare different machine learning models to check which one is best for our data and which of the parameters are best for the performance of model.

Introduction

- SpaceX 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 SpaceX 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 SpaceX for a rocket launch.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - I collect the data using SpaceX API and by scraping Wikipedia pages.
- Perform data wrangling
  - O clean the data and remove anomalies from it
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - I use built-in libraries to build the model and use Grid Search to get the best parameters for model to get maximum accuracy

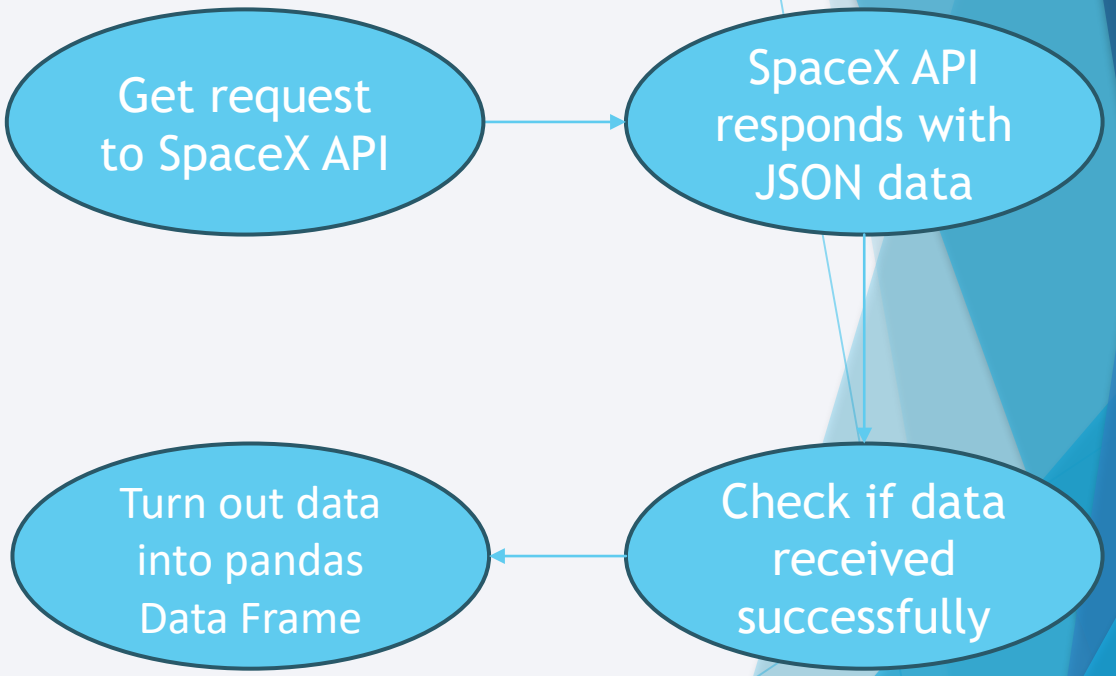
Data Collection

- SpaceX provides it's API to get the data for the processing and analysis. I use this API to get the data and also fetch some data from the Wikipedia pages using web scraping.
- I use requests library for sending Get request to the API and BeautifulSoup library for web scraping of the Wikipedia pages to get the data.

Data Collection – SpaceX API

- I show the process to collect the data using the SpaceX API.

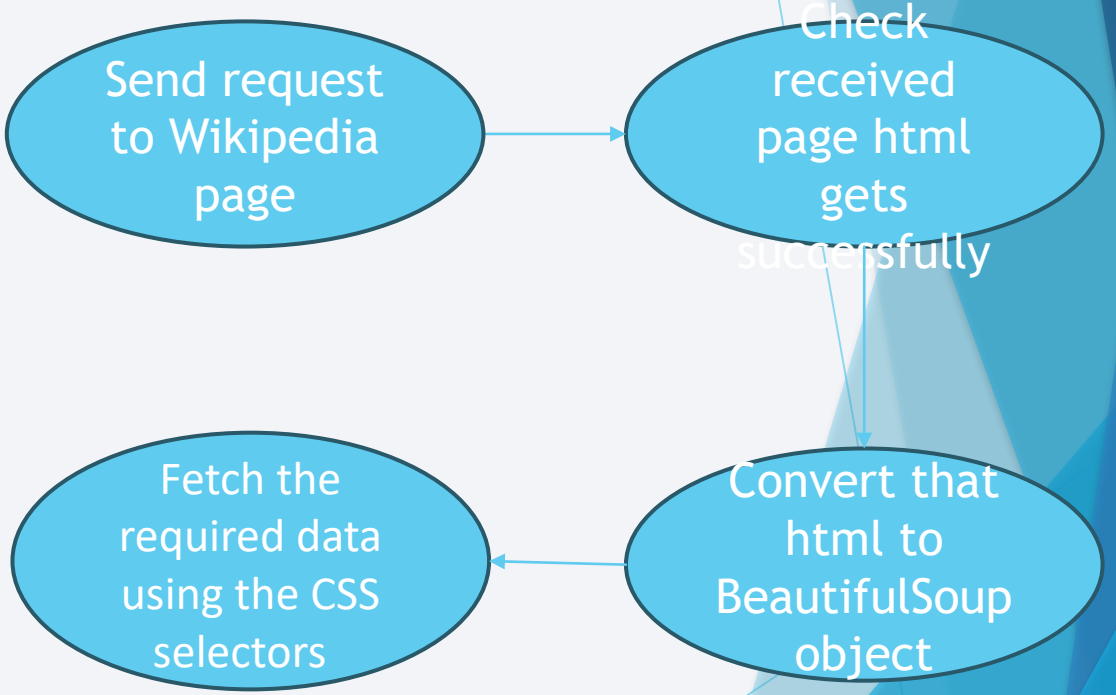
- I upload the notebook on GitHub related to data collection using SpaceX API. (https://github.com/digitalfivestar/IBM-DS-FINAL-PROJECT/blob/main/jupyter-labs-spacex-data-collection-api.ipynb).



Data Collection - Scraping

- I show the process to collect data using scraping Wikipedia pages.

- I upload the notebook on GitHub related to data collection using Web Scraping Wikipedia pages. (https://github.com/digitalfivestar/IBM-DS-FINAL-PROJECT/blob/main/jupyter-labs-webscraping.ipynb).



Data Wrangling

- I check the missing values in the data and fill them using appropriate methods.
- I also make a column with class which shows the success or failure of successful landing of first stage.
- The notebook to check the full procedure, you can visit the link (https://github.com/digitalfivestar/IBM-DS-FINAL-PROJECT/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb).

EDA with Data Visualization

- I use scatter plot to show relationship between variables.
- I also use bar chart for showing success rate for each orbit.
- I use the line chart to show the success rate yearly.
- To review its notebook, you can visit the link(https://github.com/digitalfivestar/IBM-DS-FINAL-PROJECT/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb).

EDA with SQL

- I use SQL to get total payload mass carried by NASA (CRS).
- I also check average payload mass carried by booster version F9 v1.1.
- I check the date of first successful landing outcome.
- I check the total number of successful and failure mission outcomes.
- I display the booster versions which have carried maximum payload mass.
- You can check the notebook at the link (https://github.com/digitalfivestar/IBM-DS-FINAL-PROJECT/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb).

Build an Interactive Map with Folium

- I add the location of NASA Johnson Space center onto the map.
- Then I mark the location of each launch site on the map.
- Then I've to add the success and failure markers on launch sites but as these are on same coordinates, I use clusters to show these markers effectively.
- I add mouse position on top of map to get the current position/location of pointer.
- Then I add the lines to connect the site to the nearest city, coastline, highway and railway using the location I get from mouse position.
- You can access the notebook at the link (https://github.com/digitalfivestar/IBM-DS-FINAL-PROJECT/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb).

Build a Dashboard with Plotly Dash

- I use the pie chart to show the success rate of the sites on selecting All sites.
- On selecting single launch site on dropdown, the ration of success and failure for that particular site will be shown.
- I use scatter plot to show correlation between payload and success rate and give the points color according to it's booster version.
- Scatter plot shows the correlation for selected site and for the given range.
- I upload the code on GitHub and you can access it at the link (https://github.com/digitalfivestar/IBM-DS-FINAL-PROJECT/blob/main/Dashboard/dashboard%20code.py).

Predictive Analysis (Classification)

- I get the data and standardize it.
- Then I split the data into testing and training data.
- Then I test different models like KNN, SVM, Decision Tree, Logistics Regression, etc. using the Grid Search by passing parameters list.
- I get that decision tree is best for this as it's giving better accuracy than others.
- I add the notebook to GitHub and you can access it using the link (https://github.com/digitalfivestar/IBM-DS-FINAL-PROJECT/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb).

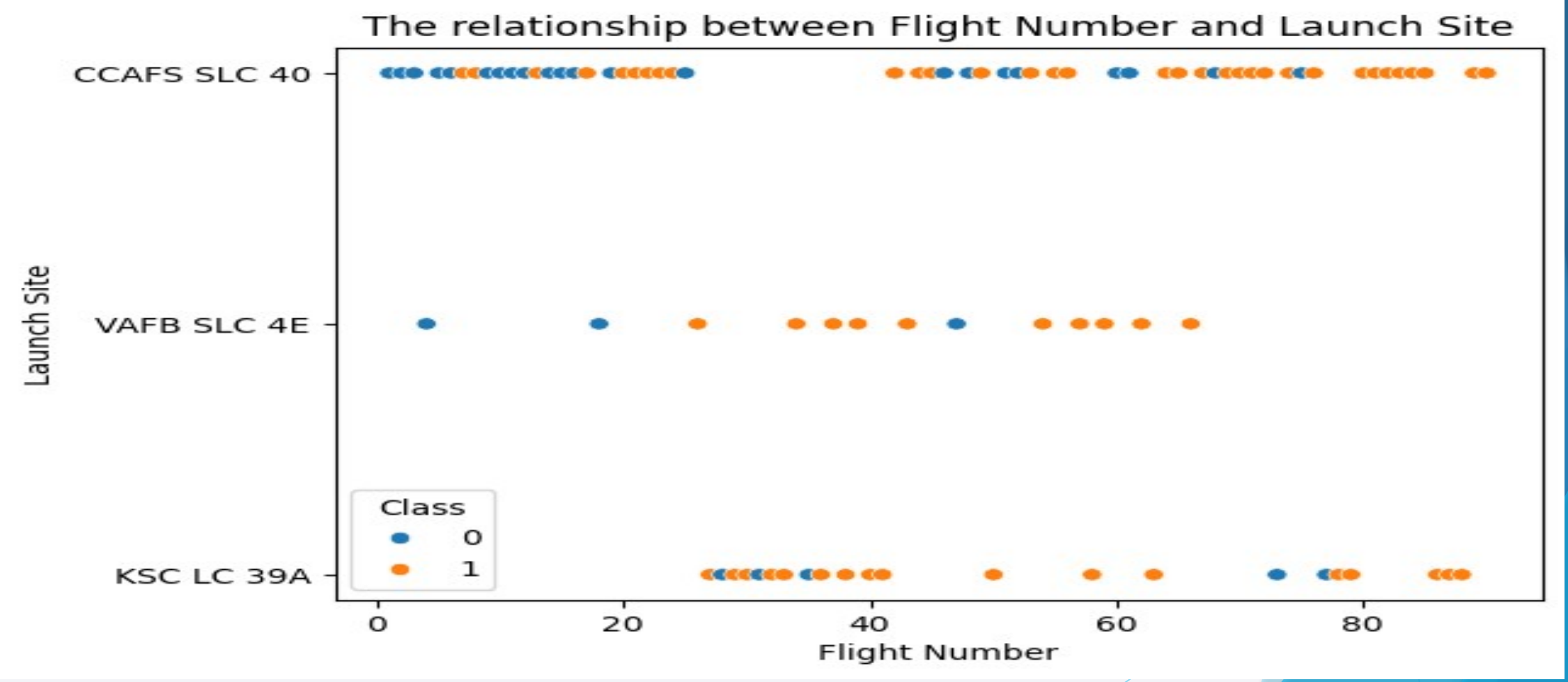
Results

- In data analysis I found a key point that success ratio is increasing year by year.
- I’ve found through dashboard that 6K-8K payload range have the lowest success rates.
- We can see that using Grid Search we can easily get the best parameters for model and also check the best model be using one by one and better here is Decision Tree.

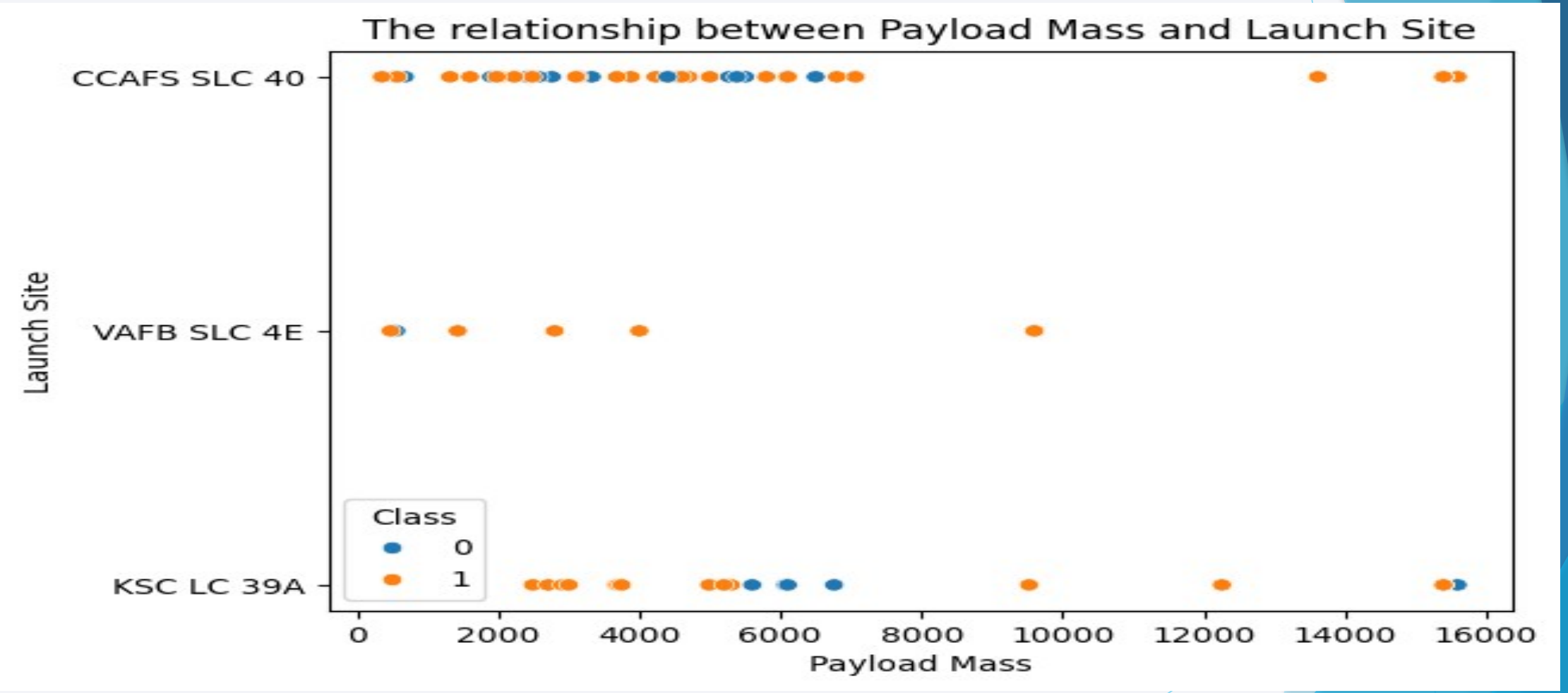
Section 2

Insights drawn from EDA

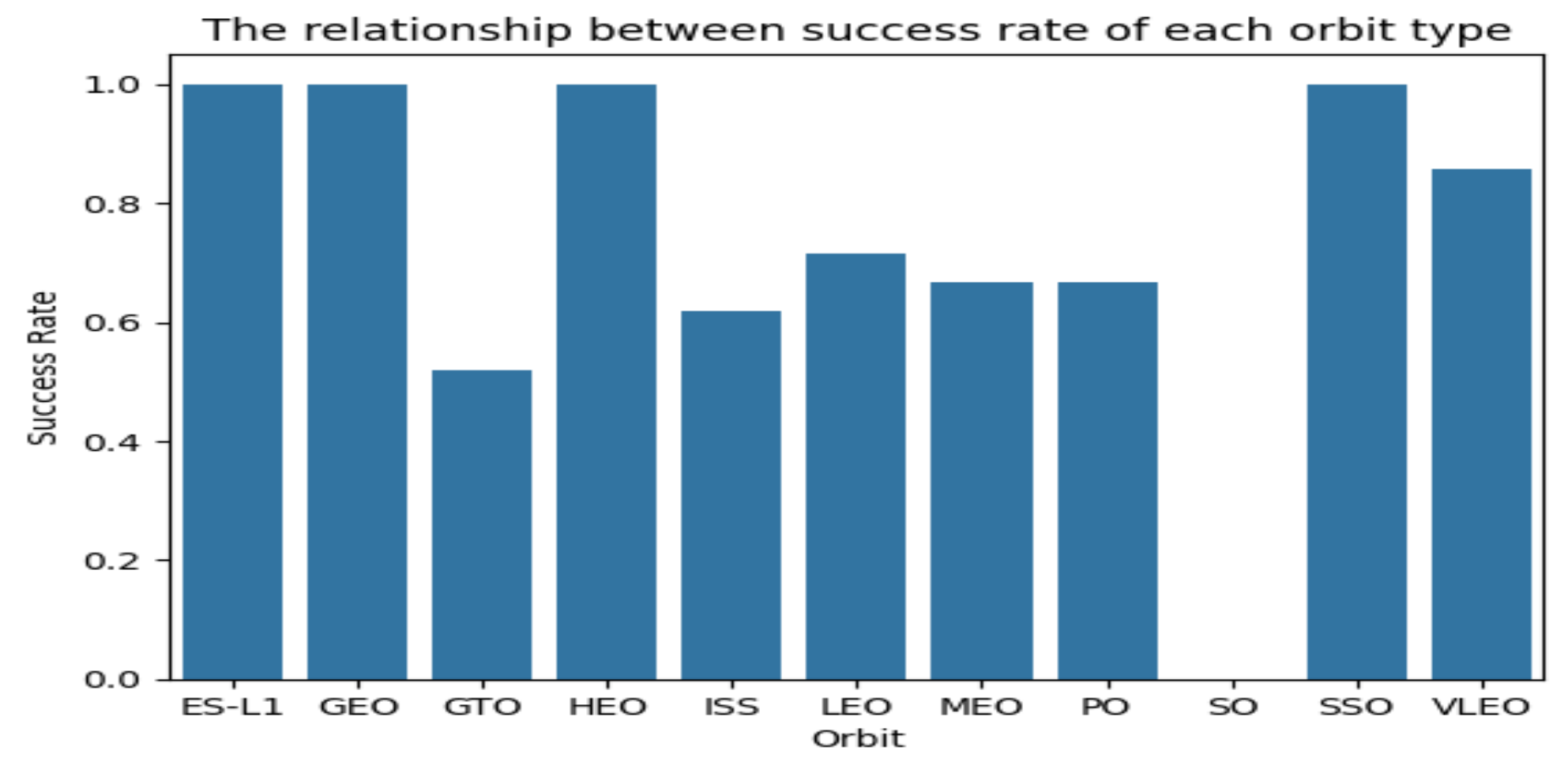
Flight Number vs. Launch Site



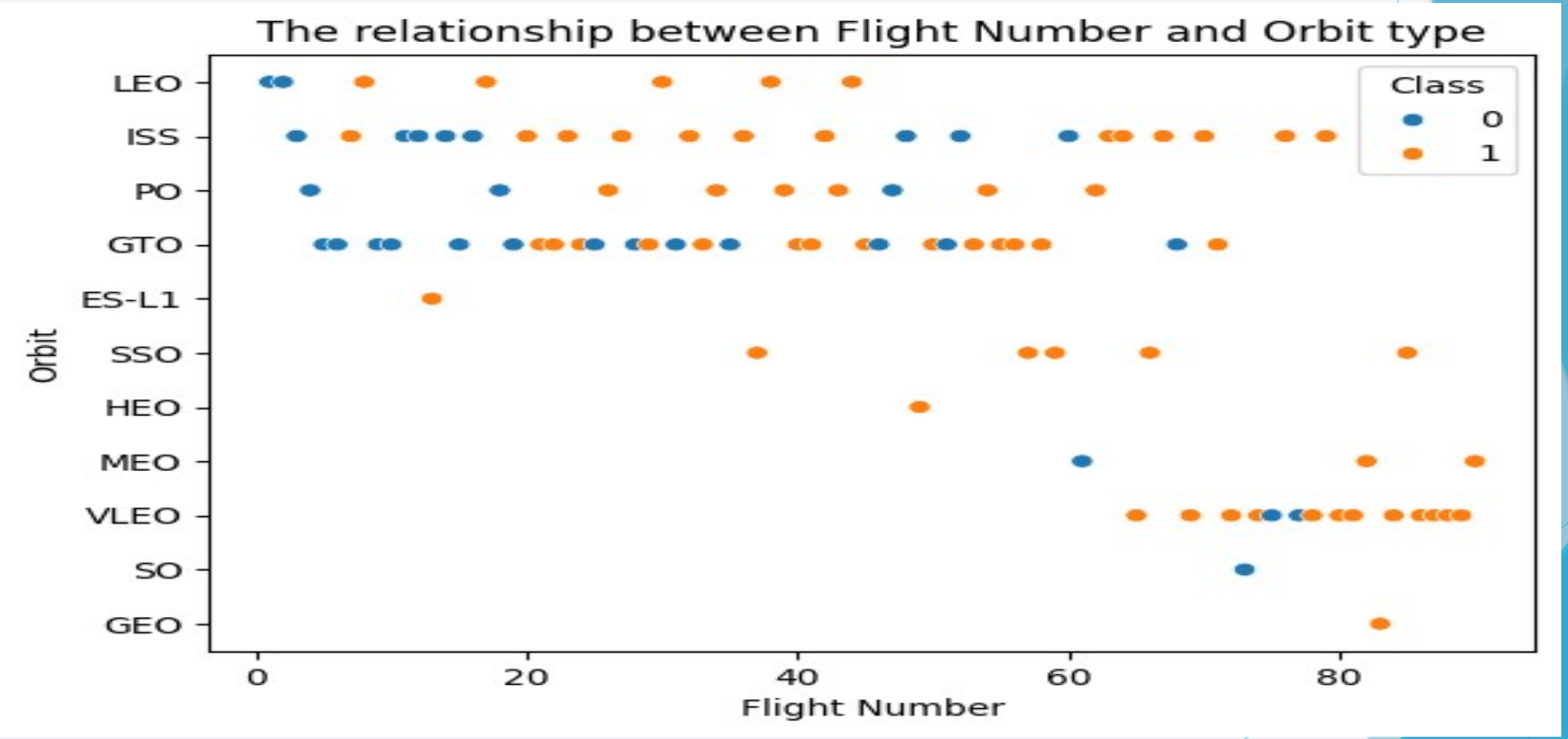
Payload vs. Launch Site



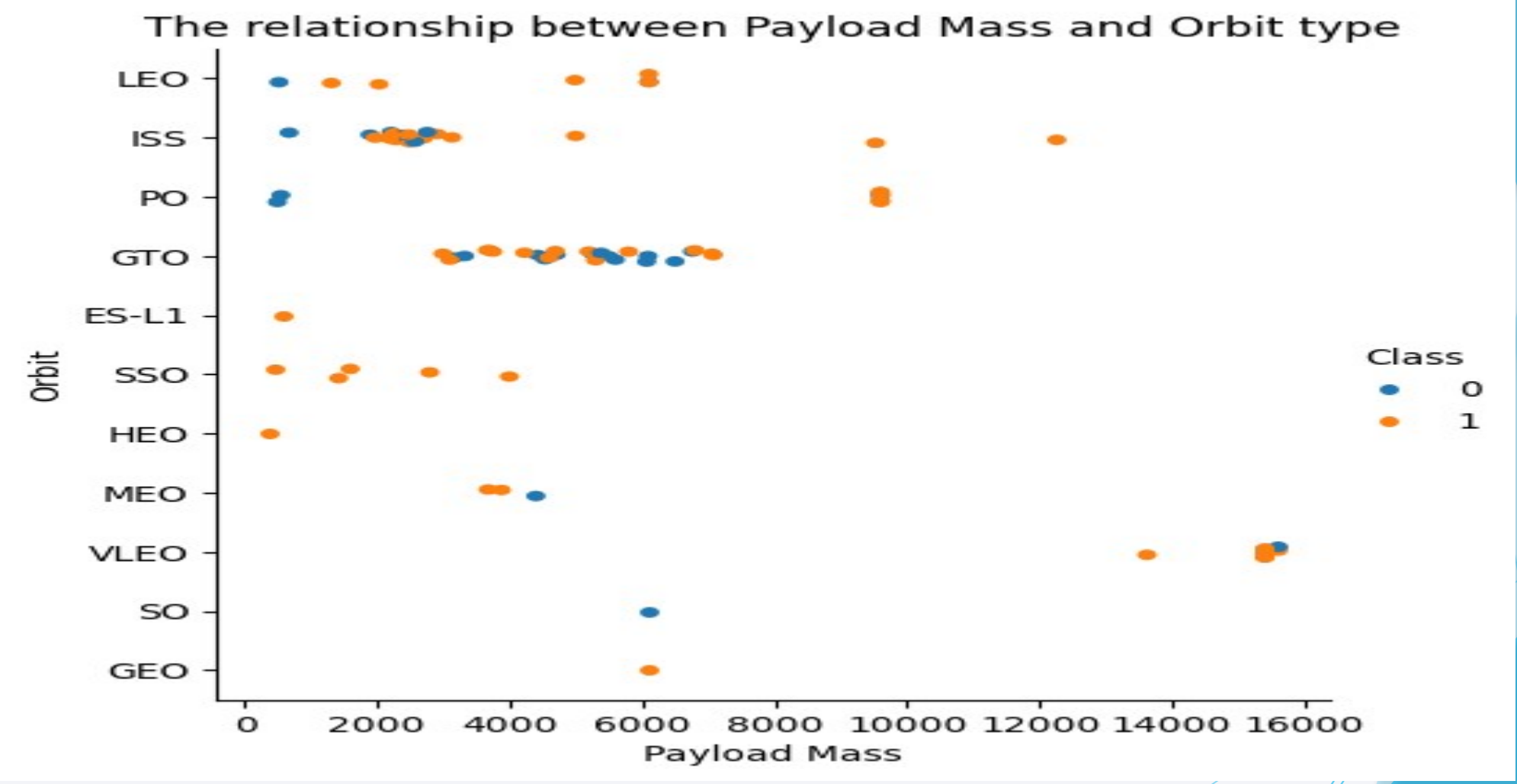
Success Rate vs. Orbit Type



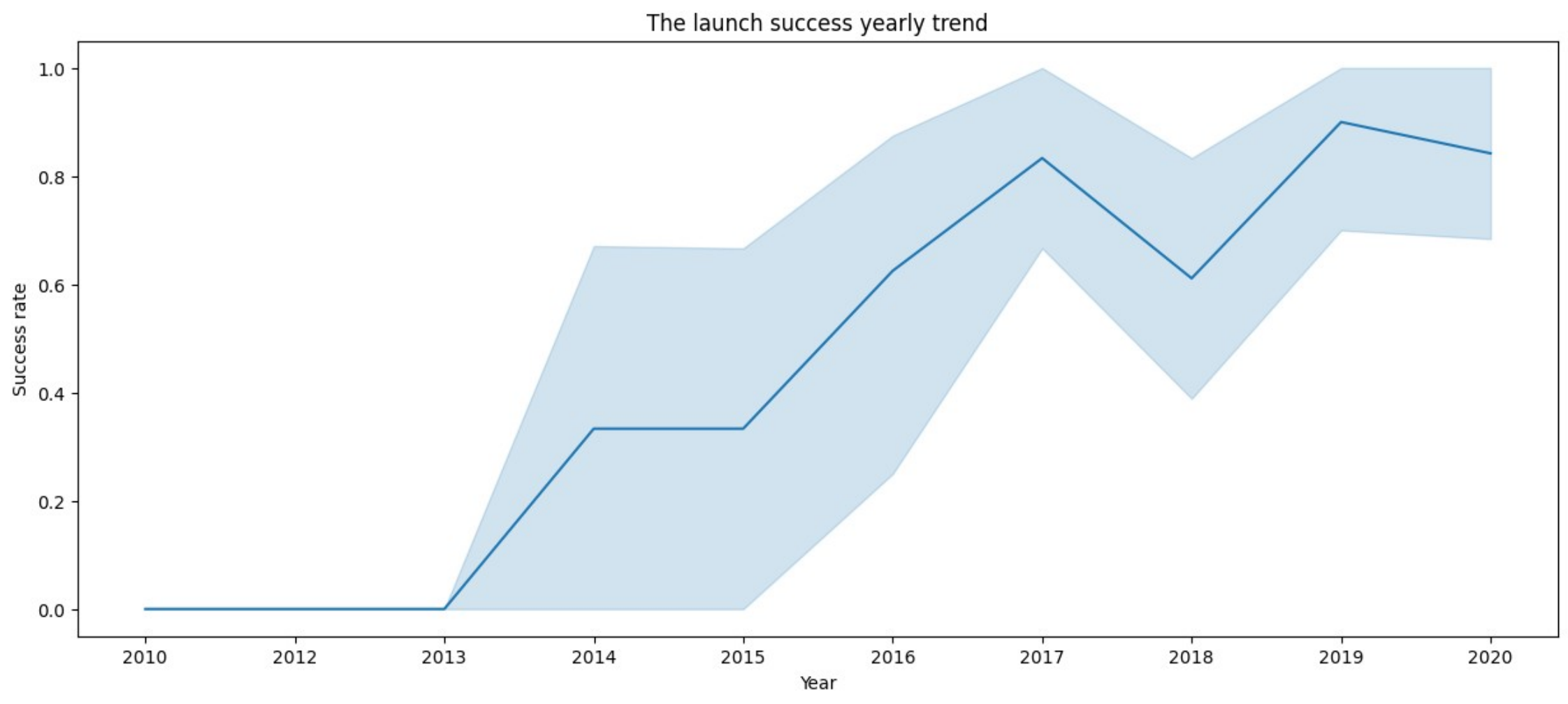
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

I use the DISTINCT function to get the unique launch site names from SPACEXTABLE.

I also use SQL magic function to run the query in notebook.

```
%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'

I find 5 records where launch sites begin with `CCA` using the like keyword.

```
%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_Outc
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parach
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 (parach
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No atte
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No atte
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No atte

Total Payload Mass

I calculate the total payload carried by boosters from NASA using the SUM aggregate function in SQL and put the condition to only NASA boosters and sum their PAYLOAD_MASS_KG_ to get the total payload.

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)'
```

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

I calculate the average payload mass carried by booster version F9 v1.1 using the AVG aggregate function in SQL which is used to calculate the average. I also apply WHERE condition to only average for F9 v1.1 booster version. I calculate the average for PAYLOAD_MASS_KG to get average

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version like 'F9 v1.1%'
```

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

I find the dates of the first successful landing outcome on ground pad using SQL by selecting only successful mission outcomes and then get the minimum of data which gives us the oldest date which we really want for successful landing outcomes.

```
%sql SELECT min(Date) FROM SPACEXTABLE WHERE Mission_Outcome = 'Success'
```

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

min(Date)
2010-06-04

Successful Drone Ship Landing with Payload between 4000 and 6000

I list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than

```
%%sql
SELECT Booster_Version
FROM SPACEXTABLE
WHERE Mission_Outcome = 'Success'
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

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

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

Total Number of Successful and Failure Mission Outcomes

I calculate the total number of successful and failure mission outcomes using COUNT and GROUP BY clauses in SQL. I also use aliases to rename the result columns.

```
%sql SELECT Mission_Outcome, COUNT(*) AS mission_outcome_count FROM SPACEXTABLE GROUP BY Mission_Outcome
```

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

Mission_Outcome	mission_outcome_count
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

I list the names of the booster which have carried the maximum payload mass using the subquery and MAX function in SQL.

```
%%sql
SELECT 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

2015 Launch Records

I list the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015. I use substr to get the months and years from date in SQL as it is sqlite and don't have other way to get month and year from date.

```
%%sql
SELECT substr(Date, 6, 2) as Month, Landing_Outcome, Booster_Version, Launch_Site
FROM SPACEXTABLE
WHERE substr(Date, 0, 5) = '2015' AND Landing_Outcome = 'Failure (drone ship)'
```

```
 * 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

I 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. I use COUNT aggregate function in it. I also use GROUP BY and ORDER BY DESC clauses to get the desired results.

On the latest notebook, it’s missing the result because I think I forget to run the cell but query is correct and I just check it before

ıg.

```
%%sql
SELECT Landing_Outcome, COUNT(Landing_Outcome) as landing_outcome_count
FROM SPACEXTABLE
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY landing_outcome_count DESC
```

Section 3

Launch Sites Proximities Analysis

Launch sites on map

You can easily check the location on the map. Launch sites are shown with red dots on map and I also add their names as labels on them

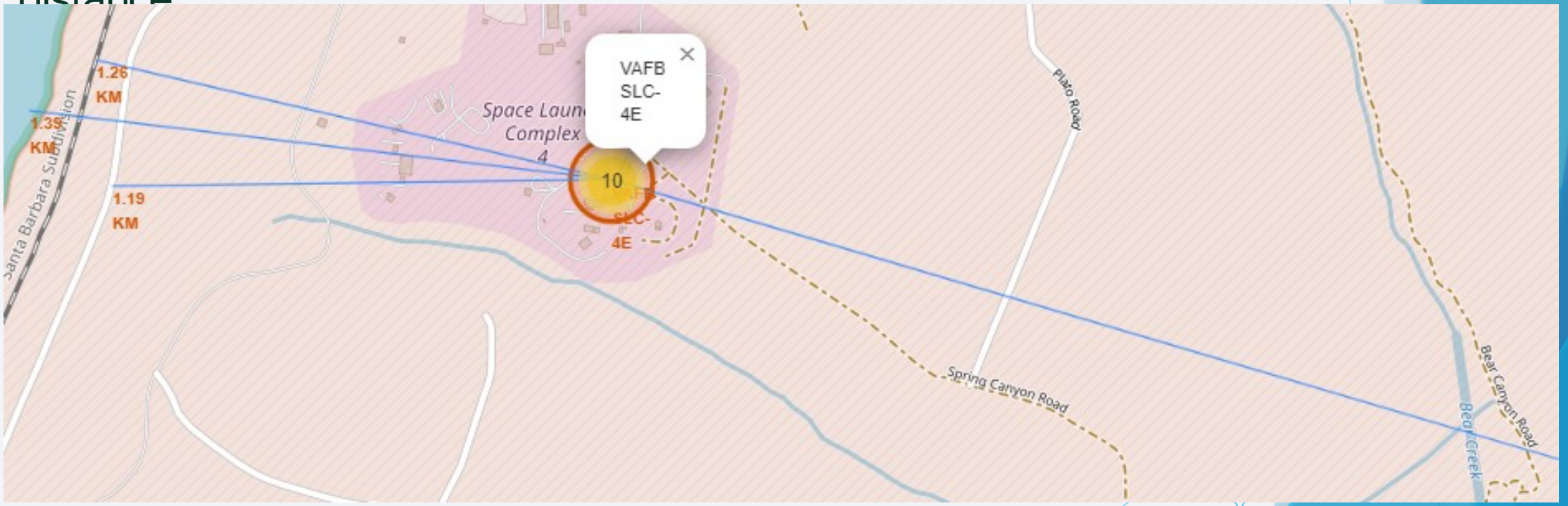
Launch outcomes show up on sites

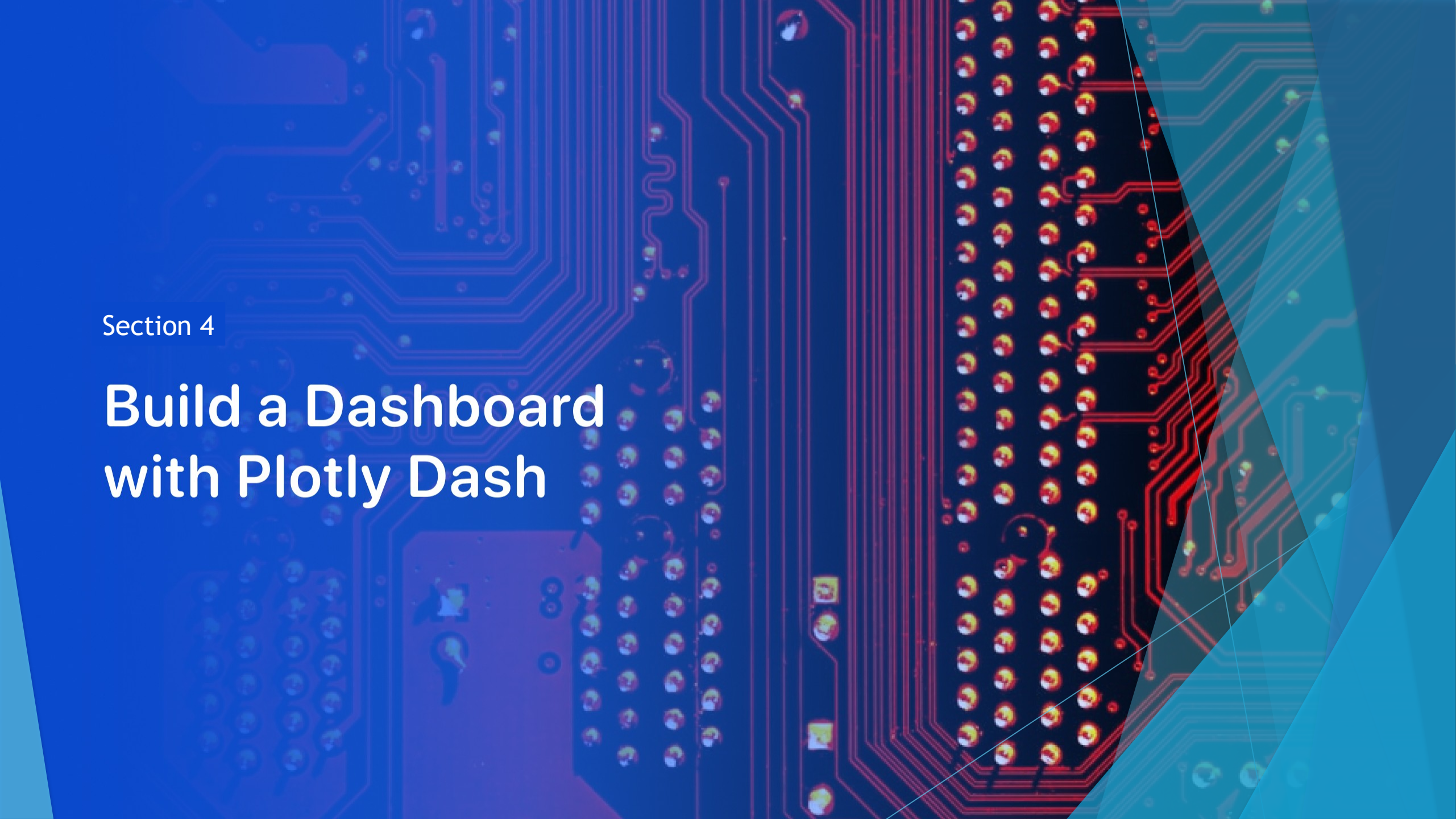
You can easily check the color popups on the launch showing success and failure popups with red as failure and green as

Launch site and it’s proximities

Here in the map we can easily see the launch site and it’s distance to it’s proximities clearly mentioned and line drawn between them.

A line going from the site is connecting it to city LA with 227.93 KM distance



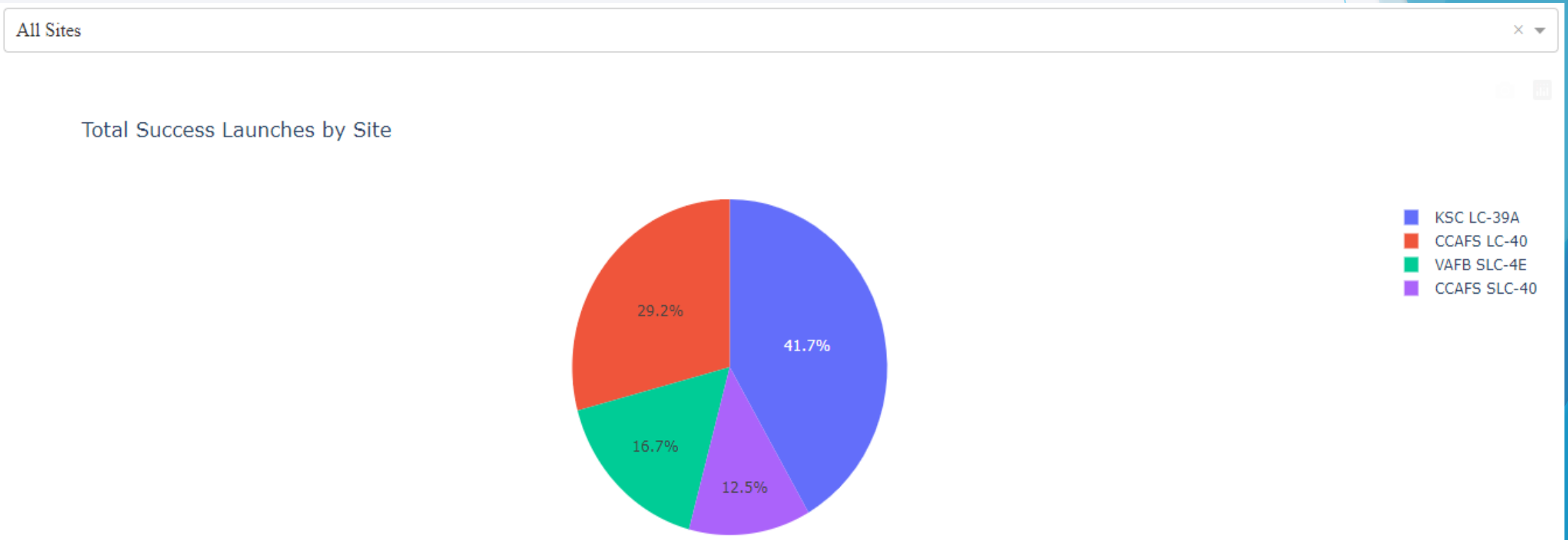


Section 4

Build a Dashboard with Plotly Dash

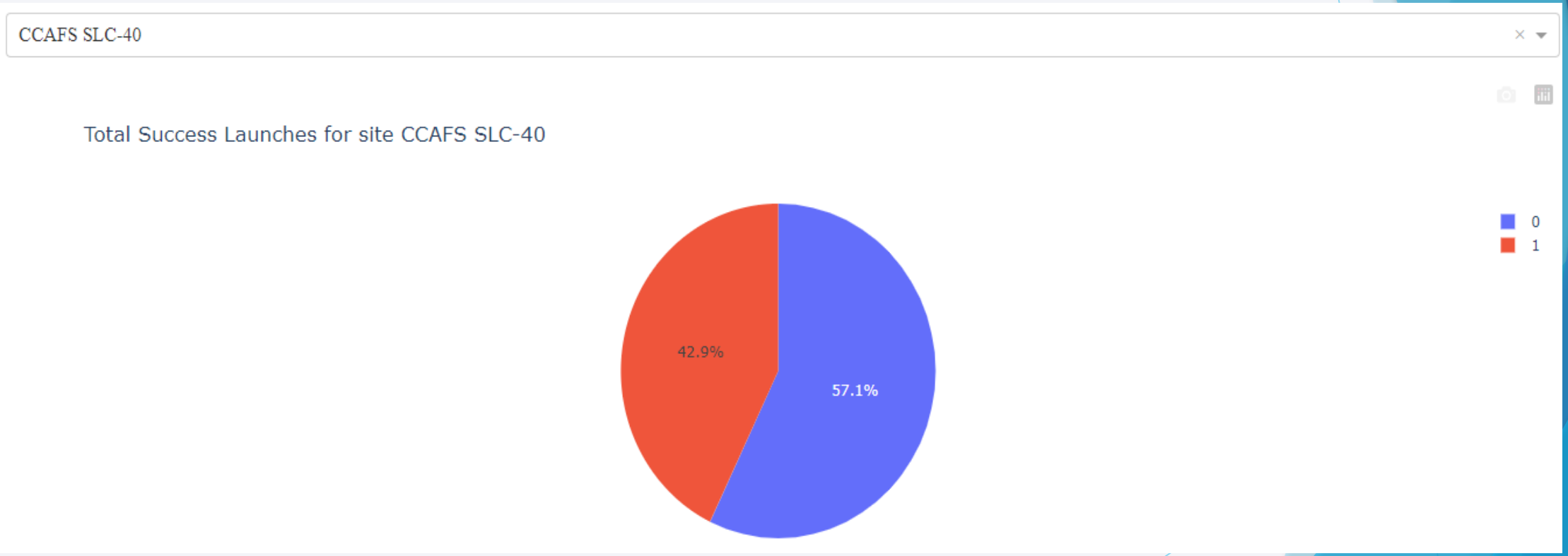
Launch success count for all sites

I get the success rate for all sites using the pie chart. From this we can check that KSC LC-39A have largest success count with 41.7% of ration



Launch site with highest launch success ratio

From the pie chart we can easily see that CCAFF SLC-40 has highest success ration with 42.9% of success ration



Success vs payload mass

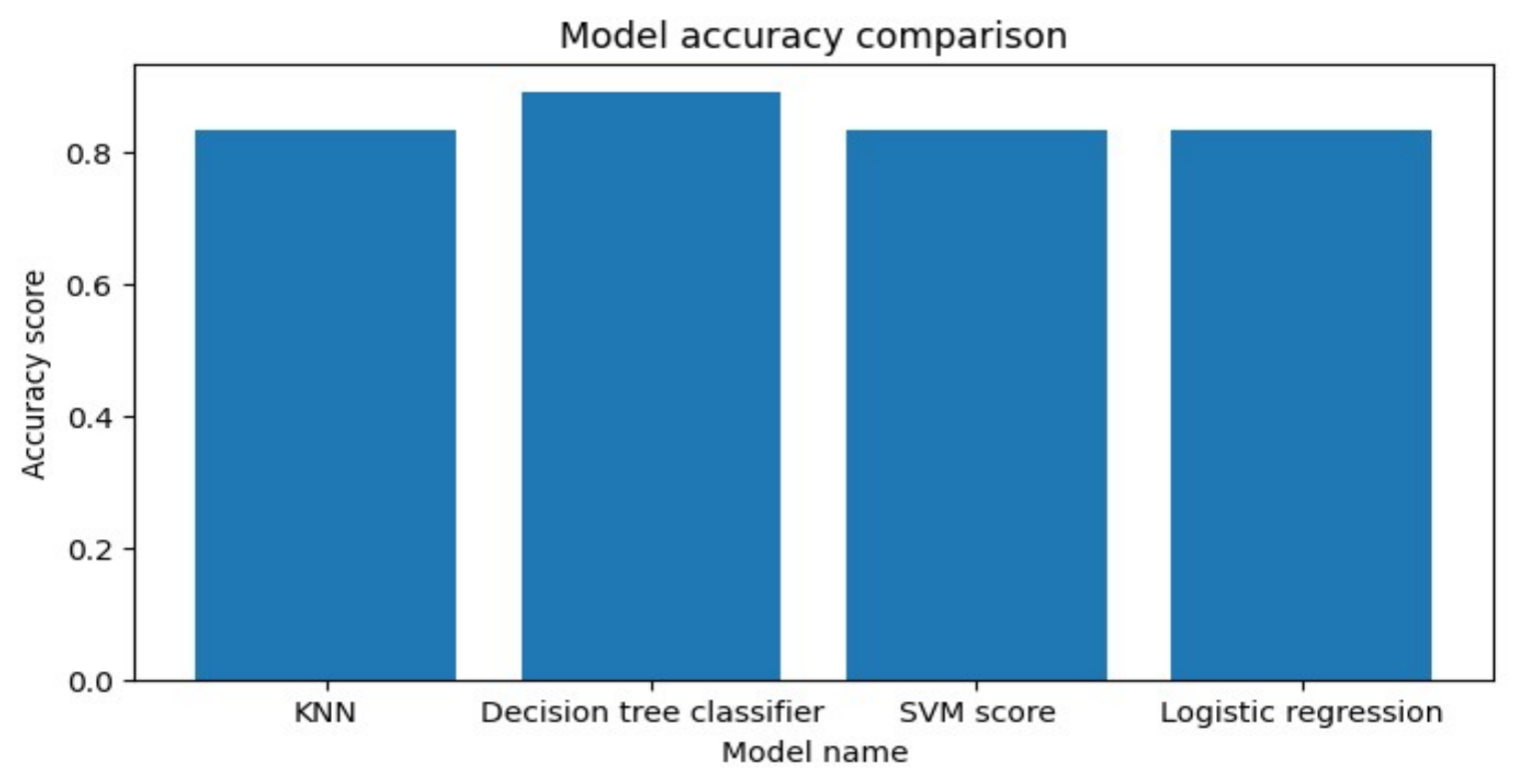
We can easily see the all payload ranges in the given scatter plot.

Section 5

Predictive Analysis (Classification)

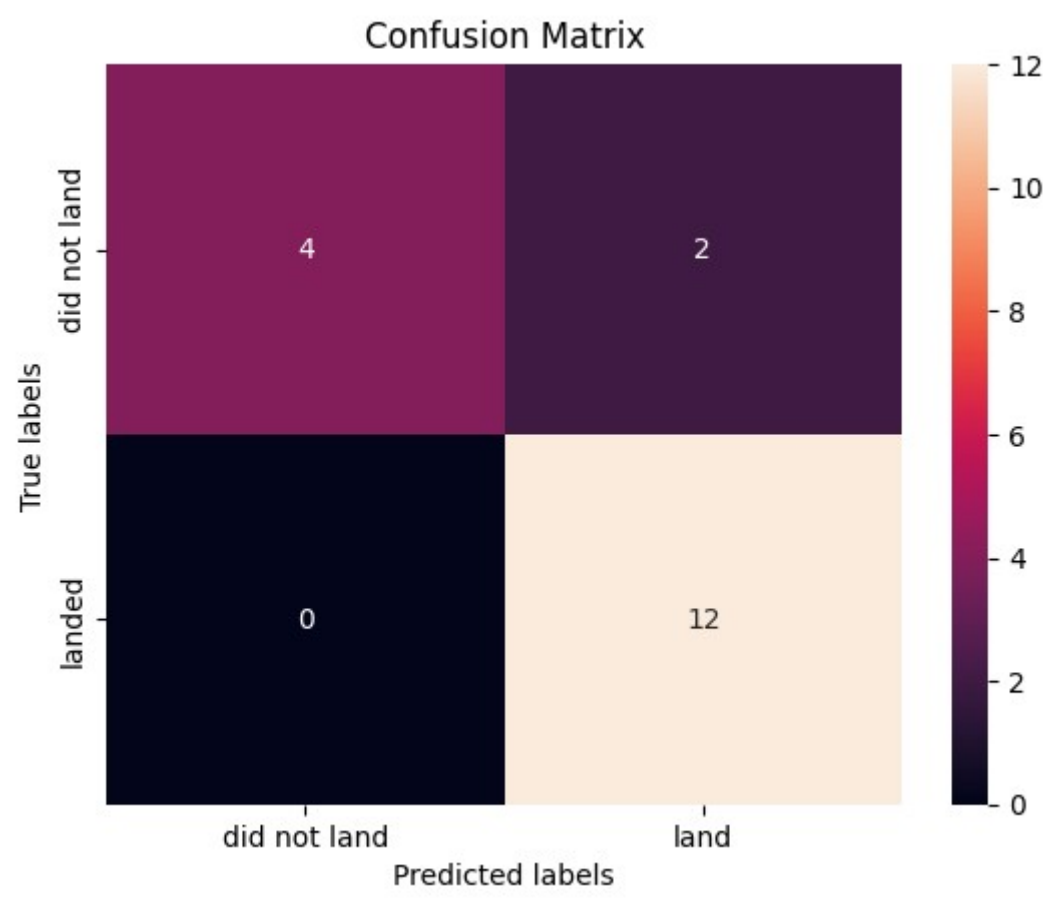
Classification Accuracy

From the bar chart we can easily see that Decision tree has maximum accuracy.



Confusion Matrix

Here in the figure, you can easily see the confusion matrix of the most accurate



Conclusions

- Success rate is increasing year by year.
- KSC LC-39A has the largest successful launches.
- CCAFS SLC-40 has the highest launch success rate.
- Payload range of 2K-4K have the highest launch success rate.
- 6K-8K have the lowest launch success rate.
- F9 booster version, FT has the highest launch success rate.
- Decision tree classifier is best to train model on our data as it is providing the highest accuracy.

Appendix

To see the project in detail and for checking the codes and snippets and also to get and download the notebooks and to check the more useful resources, you can go to the link (https://github.com/digitalfivestar/IBM-DS-FINAL-PROJECT/tree/main) where you can find each code in well-settled manner and you can also clone the whole repository to your laptop/PC for your use.

You can get help from code and also give suggestions to the code.

Thank you!