IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

KANISHK JOSHI
31-March 2025

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- Project background and context
  - Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.
- Problems you want to find answers
  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Using SpaceX REST API
  - Using Web Scarping from Wikipedia
- Perform data wrangling
  - Data Filtering was done
  - Missing values were handled
  - One Hot encoding was used to prepare the data for classification.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Classification models were built and enhanced to get the best results

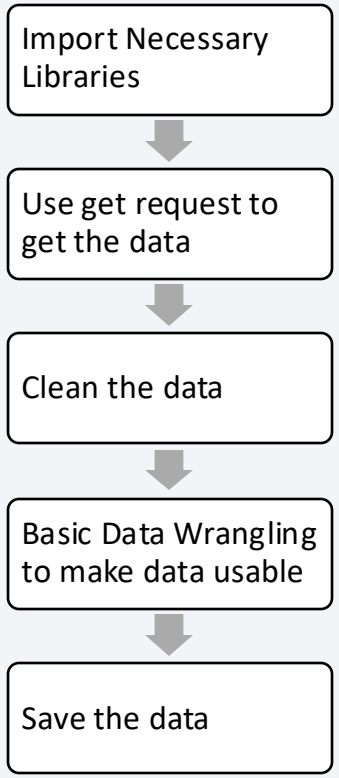
Data Collection

- Data collection process involved a combination of API requests from SpaceX REST API and Web Scraping data from a table in SpaceX's Wikipedia entry. Two sources had to be used in order to get complete information about the launches for a more detailed analysis.
- Data Columns obtained by using SpaceX REST API: FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude
- Data Columns obtained by using Wikipedia Web Scraping: Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time

Data Collection – SpaceX API

- SpaceX API was used to collect data, clean the requested data and do some basic data wrangling and formatting.

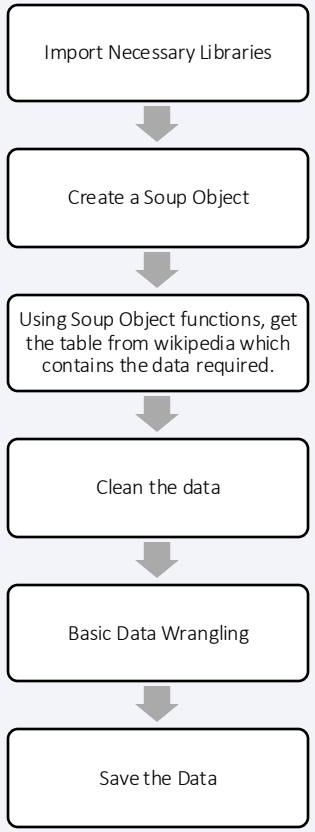
- Github Link - ibm_data_science/jupyter-labs-spacex-data-collection-api.ipynb at main · SuperKJ/ibm_data_science



Data Collection - Scraping

- WebScraping was done using Beautiful to extract Falcon 9 Data from Wikipedia

- Github URL - ibm_data_science/jupyter-labs-webscraping (1).ipynb at main · SuperKJ/ibm_data_science

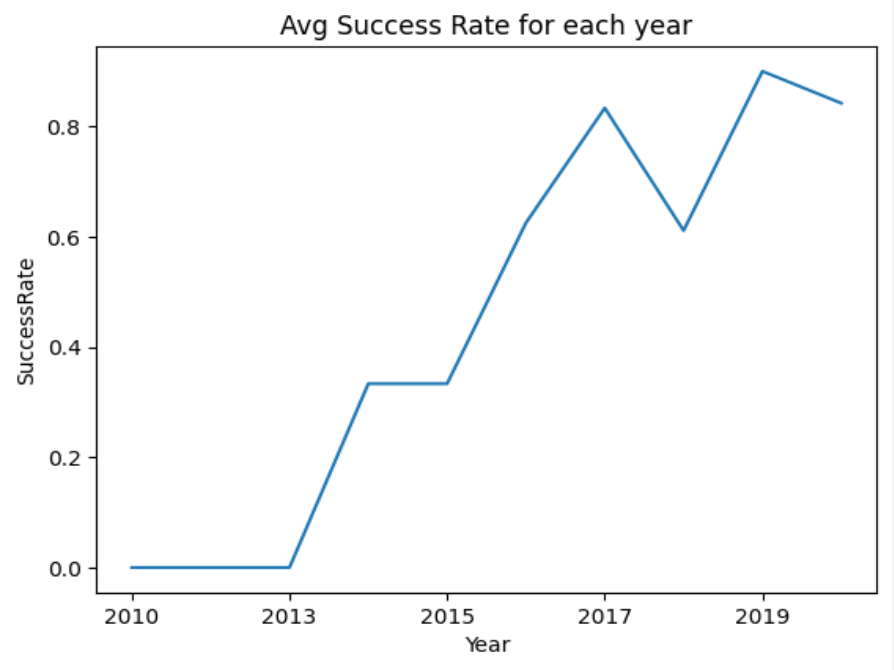


Data Wrangling

- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- Github URL - ibm_data_science/labs-jupyter-spacex-Data wrangling-v2.ipynb at main · SuperKJ/ibm_data_science

EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.
- Github URI - ibm_data_science/jupyter-labs-eda-dataviz-v2.ipynb at main · SuperKJ/ibm_data_science



EDA with SQL

- We loaded the SpaceX dataset into a SQLLite database without leaving the jupyter notebook.
- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:

  -The names of unique launch sites in the space mission.

  -The total payload mass carried by boosters launched by NASA (CRS)

  -The average payload mass carried by booster version F9 v1.1

  -The total number of successful and failure mission outcomes

  -The failed landing outcomes in drone ship, their booster version and launch site names.

- **Github URL -** ibm_data_science/jupyter-labs-eda-sql-coursera_sqllite (1).ipynb at main · SuperKJ/ibm_data_science

Build an Interactive Map with Folium

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

  -Are launch sites near railways, highways and coastlines.

  -Do launch sites keep certain distance away from cities

- **Github URL -** ibm_data_science/lab-jupyter-launch-site-location-v2.ipynb at main · SuperKJ/ibm_data_science

Build a Dashboard with Plotly Dash

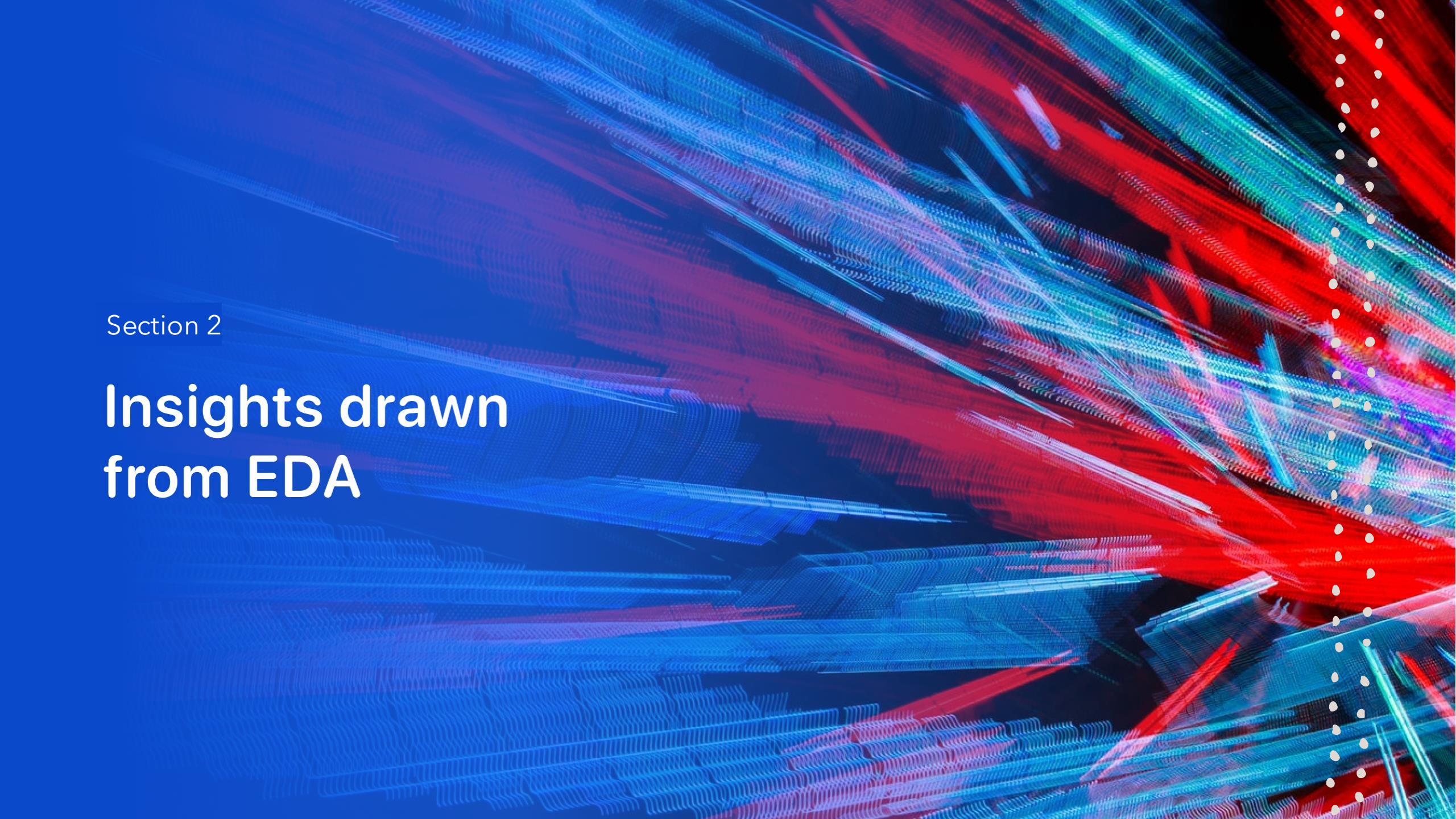
- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- **Github URL -** ibm_data_science/spacex-dash-app.py at main · SuperKJ/ibm_data_science

Predictive Analysis (Classification)

- Multiple Classification Models were used and compared to get the best results.
- The was split in to train data and test data and gridearchCV was used to get the optimum results.
- Github URL - ibm_data_science/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb at main · SuperKJ/ibm_data_science

Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

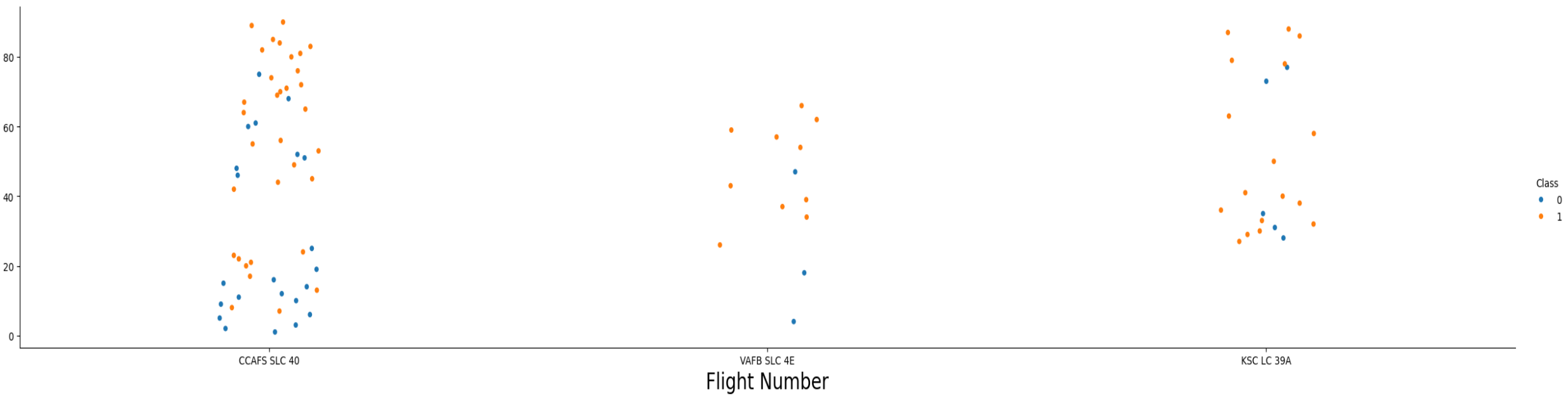


Section 2

Insights drawn from EDA

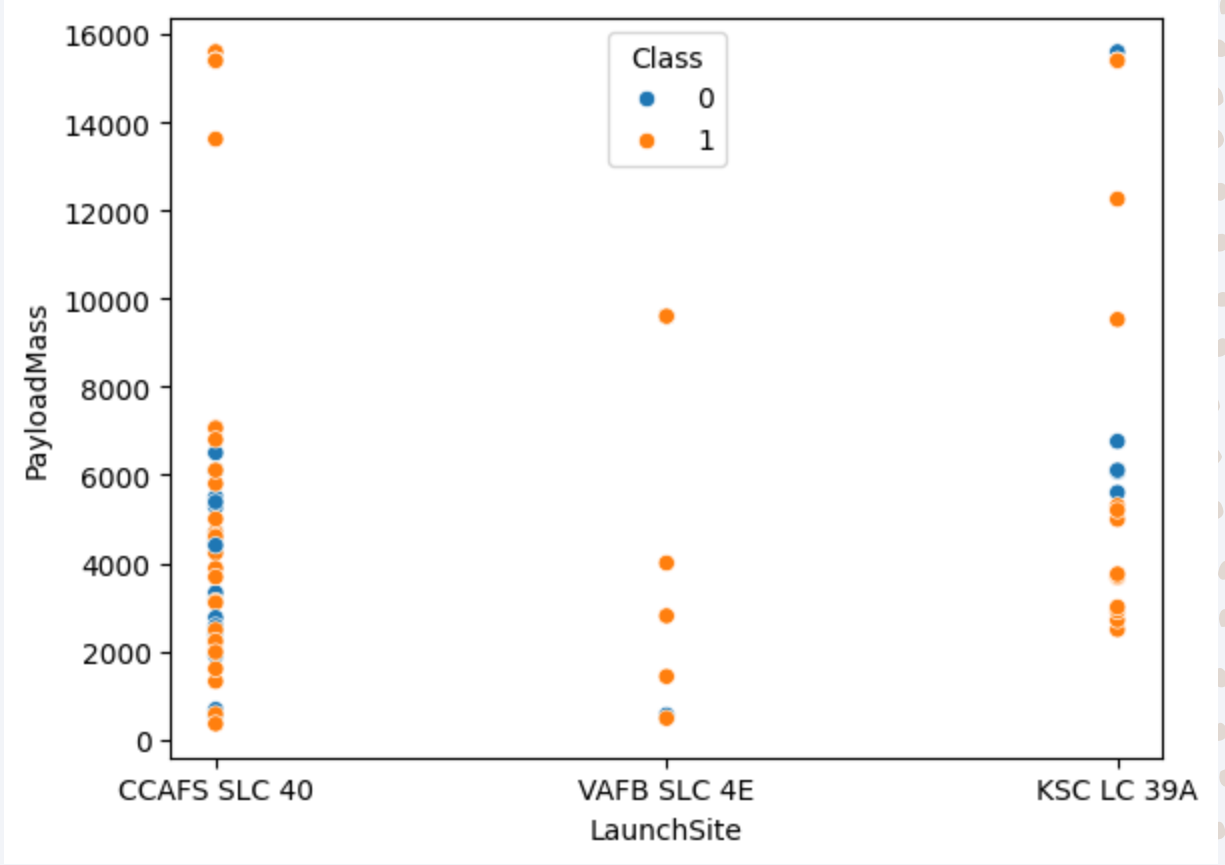
Flight Number vs. Launch Site

- It can be observed that the first Launch Site had the most number of launches.



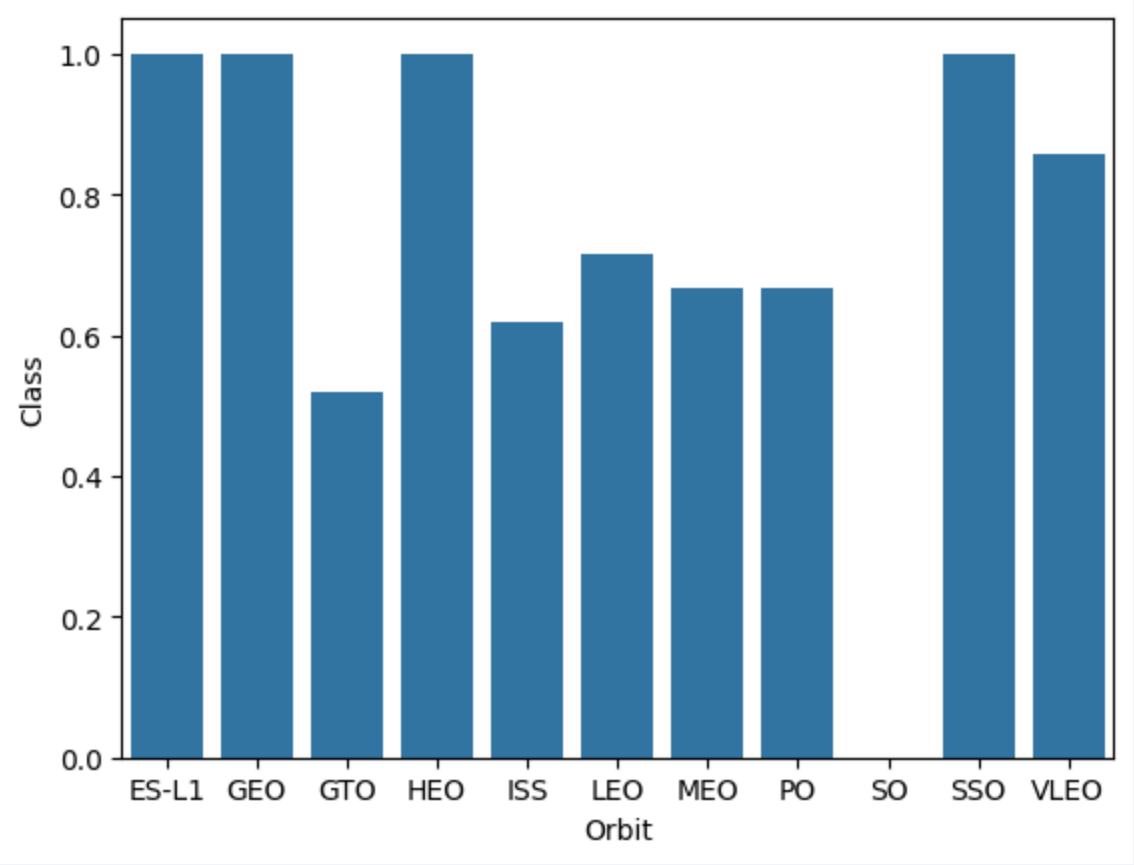
Payload vs. Launch Site

- It can be observed that most successful launches were when payload mass was below 10000.



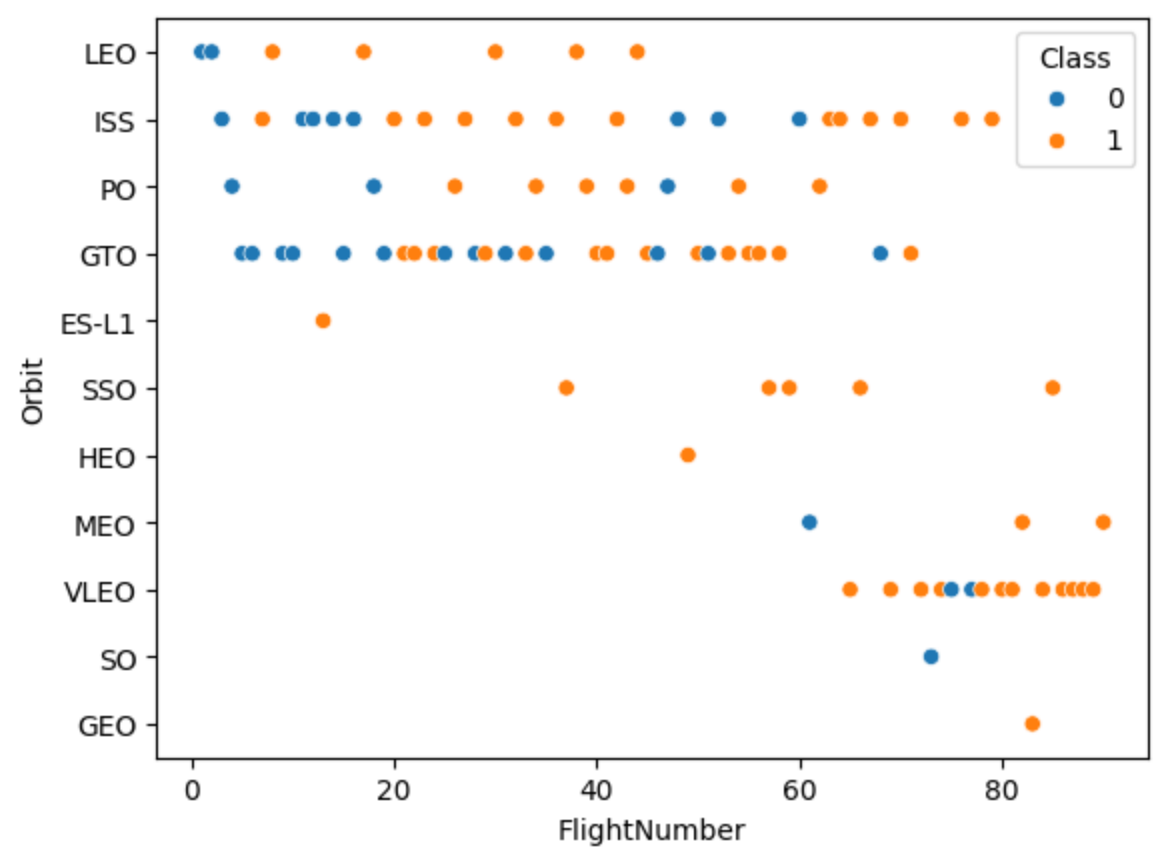
Success Rate vs. Orbit Type

- It can be observed that ES-L1,GEO,SSO have very high success rate while SO has the lowest.



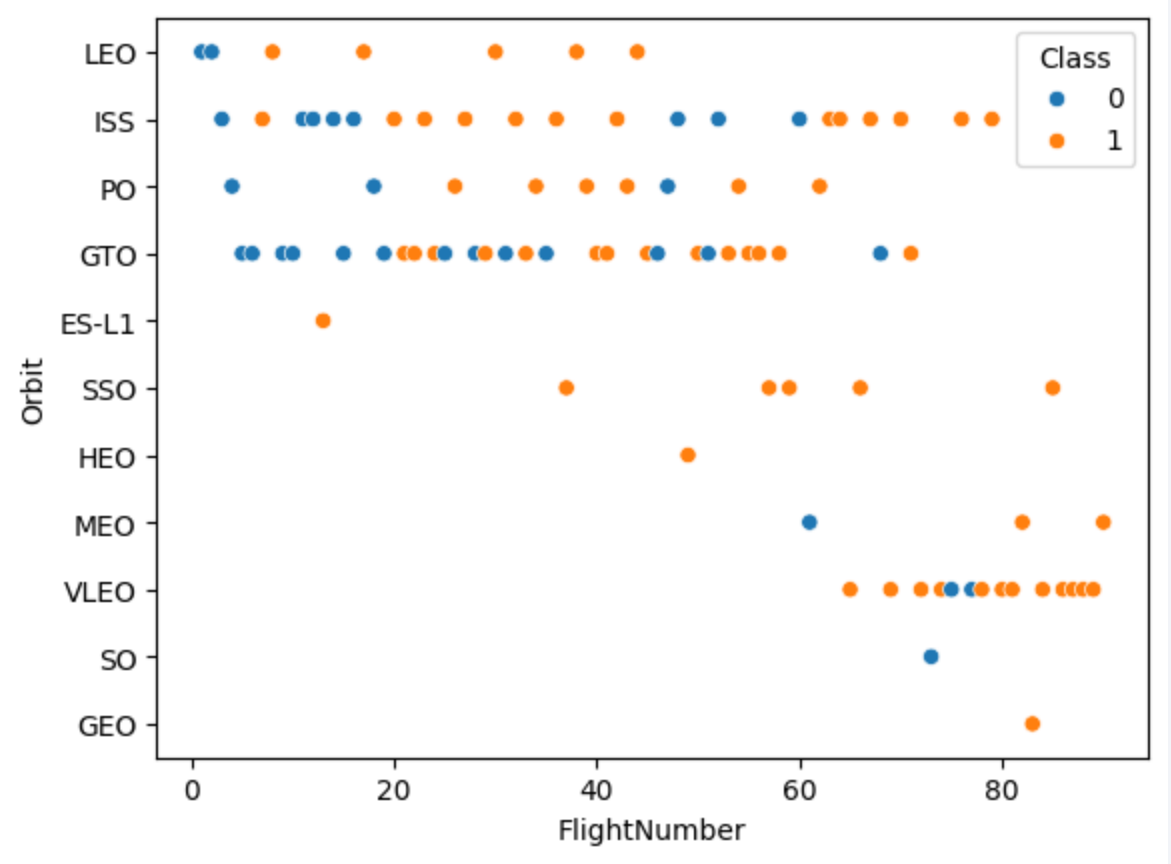
Flight Number vs. Orbit Type

- GTO had failures when flight number was below 40. Almost all the flights for VLEO after flight number 60 were successful.



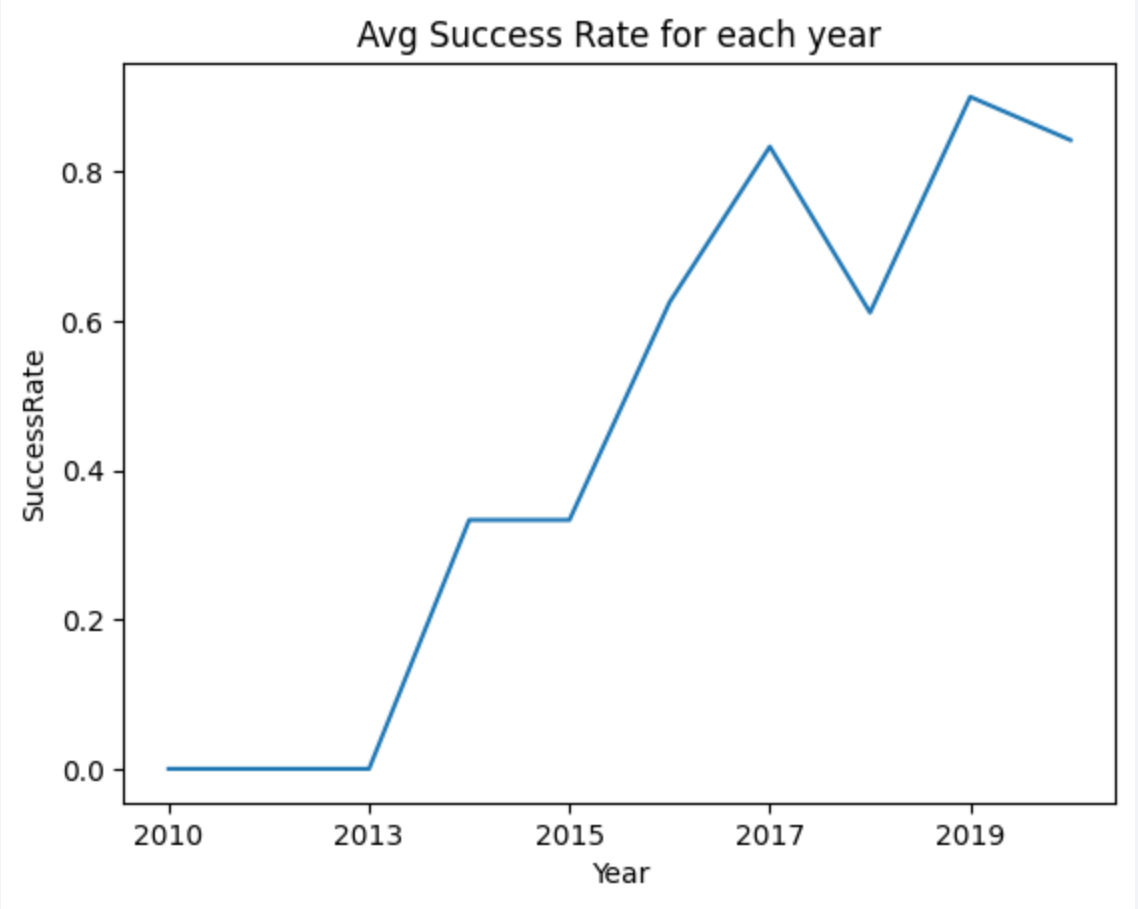
Payload vs. Orbit Type

- It can be observed that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



Launch Success Yearly Trend

- Success rate gradually increased yearly except a dip between 2017 and 2019.



All Launch Site Names

- There are 4 launch sites.

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

Launch Site Names Begin with 'CCA'

We used the query below to display 5 records where launch sites begin with `CCA`

```
[22]: %sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE "CCA%"
```

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

[22]:

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
2013-12-03	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES	Success	No attempt

Total Payload Mass

- We calculated the total payload carried by boosters from NASA as 45596 using the query below

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

```
[28]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE "Customer"="NASA (CRS)"
```

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

[28]:

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

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

```
[29]: %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE "Booster_Version"="F9 v1.1"
```

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

[29]:

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015

```
[32]: %sql SELECT * FROM SPACEXTABLE WHERE "Date"=(SELECT MIN(Date) FROM SPACEXTABLE WHERE "Landing_Outcome"="Success (ground pad)")
```

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

[32]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2015-12-22	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

- We used the **WHERE** clause to filter for boosters which have successfully landed on drone ship and applied the **AND** condition to determine successful landing with payload mass greater than 4000 but less than 6000

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
34]: %sql SELECT * FROM SPACEXTABLE WHERE "Landing_Outcome"="Success (drone ship)" AND "PAYLOAD_MASS__KG_" BETWEEN 4000 and 6000
 * sqlite:///my_data1.db
Done.
```

34]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2016-05-06	5:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-08-14	5:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-10-11	22:53:00	F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105	5200	GTO	SES EchoStar	Success	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- Below Query was used to get success and failure outcomes.

List the total number of successful and failure mission outcomes

```
[39]: %sql SELECT COUNT(*),Mission_Outcome FROM SPACEXTABLE WHERE Mission_Outcome LIKE "%Success%" OR Mission_Outcome LIKE "%Failure%" GROUP BY Mission_Outcome
```

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

[39]:

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

Boosters Carried Maximum Payload

- Subquerying was used to get list of booster version which carried max capacity payload mass

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
[41]: %sql SELECT Booster_Version FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE)
 * sqlite:///my_data1.db
Done.
```

[41]:

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

- We used a combinations of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions to filter for failed landing outcomes in drone ship names for year 2015

```
[47]: %sql SELECT SUBSTR(Date,6,2) "MonthName",Landing_Outcome FROM SPACEXTABLE WHERE Landing_Outcome LIKE "%Failure%" AND SUBSTR(Date,0,5)="2015";
 * sqlite:///my_data1.db
Done.
```

[47]:

MonthName	Landing_Outcome
01	Failure (drone ship)
04	Failure (drone ship)

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

- 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

- Pres

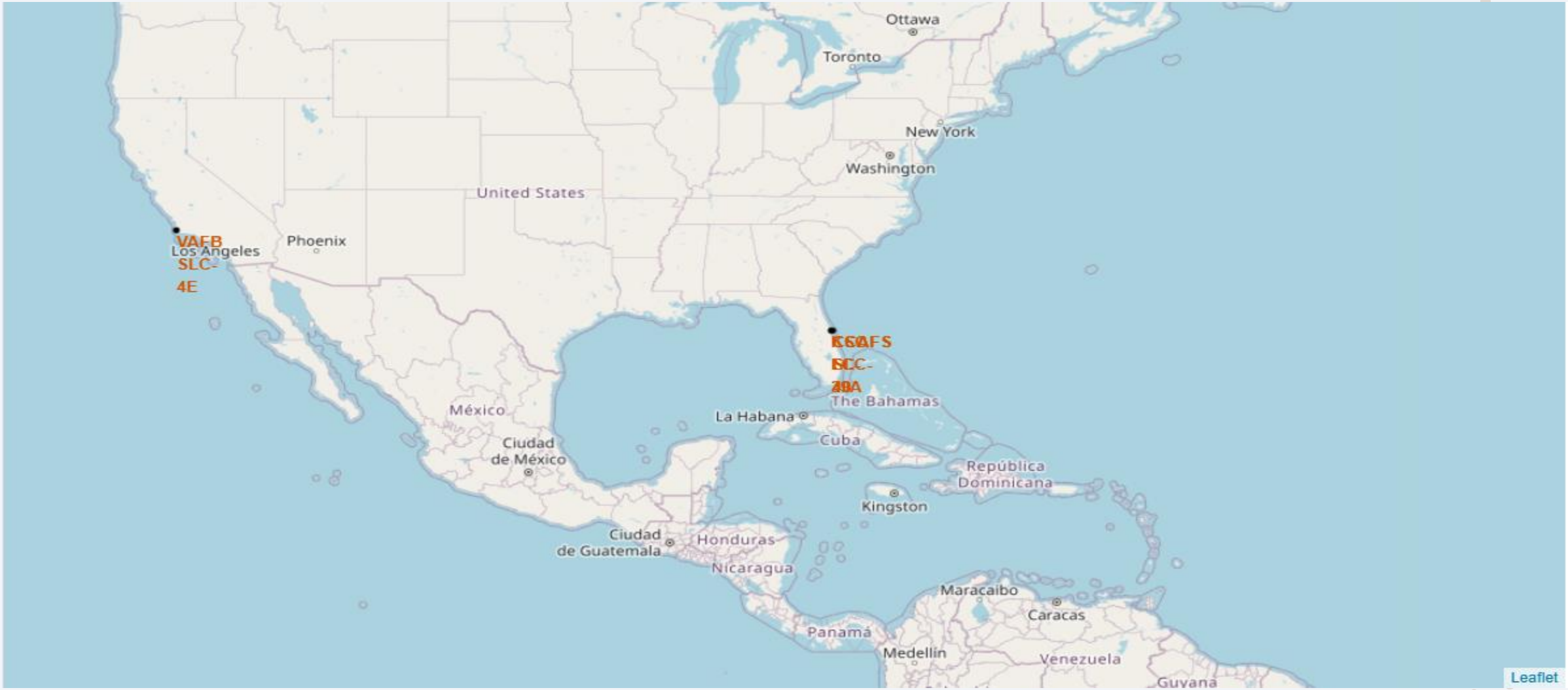
```
[51]: %sql SELECT COUNT(*), Landing_Outcome FROM SPACEXTABLE GROUP BY Landing_Outcome HAVING DATE BETWEEN "2010-06-04" AND "2017-03-20" ORDER BY Count(*) DESC;
 * sqlite:///my_data1.db
Done.
```

[51]:

COUNT(*)	Landing_Outcome
21	No attempt
14	Success (drone ship)
9	Success (ground pad)
5	Failure (drone ship)
5	Controlled (ocean)
2	Uncontrolled (ocean)
2	Failure (parachute)
1	Precluded (drone ship)

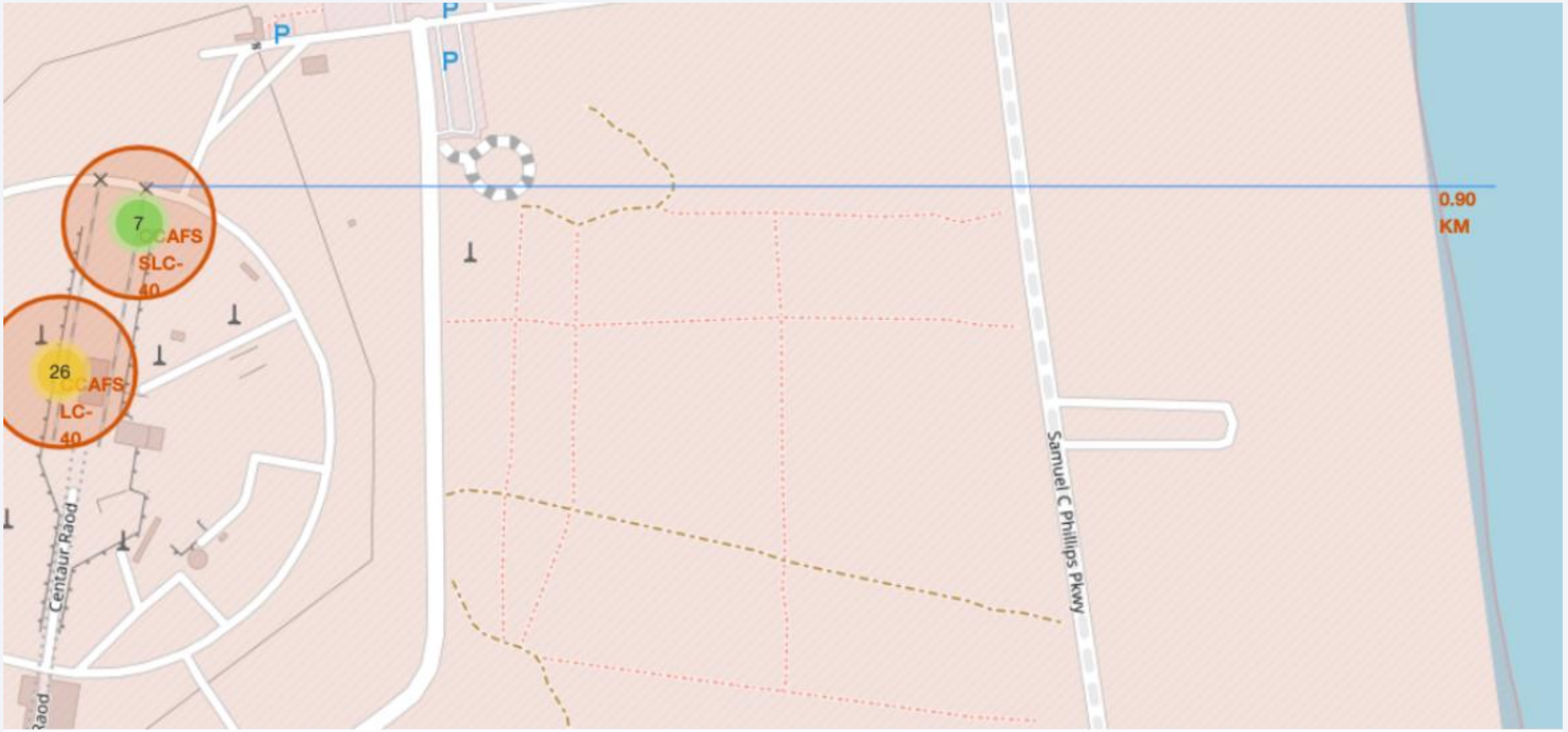

Section 3
Launch Sites
Proximities Analysis

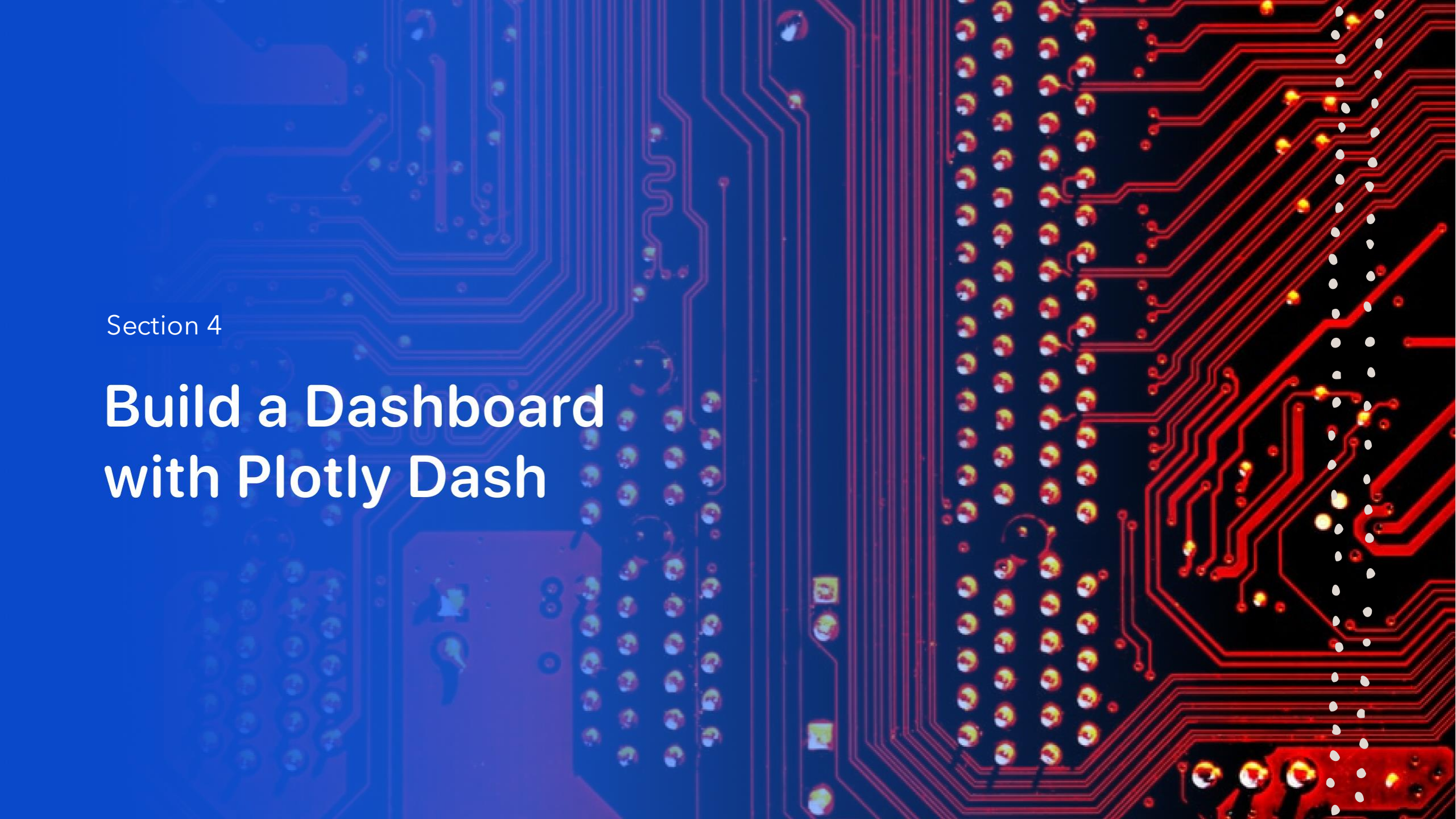
All launch sites global map markers



Markers showing launch sites with color labels

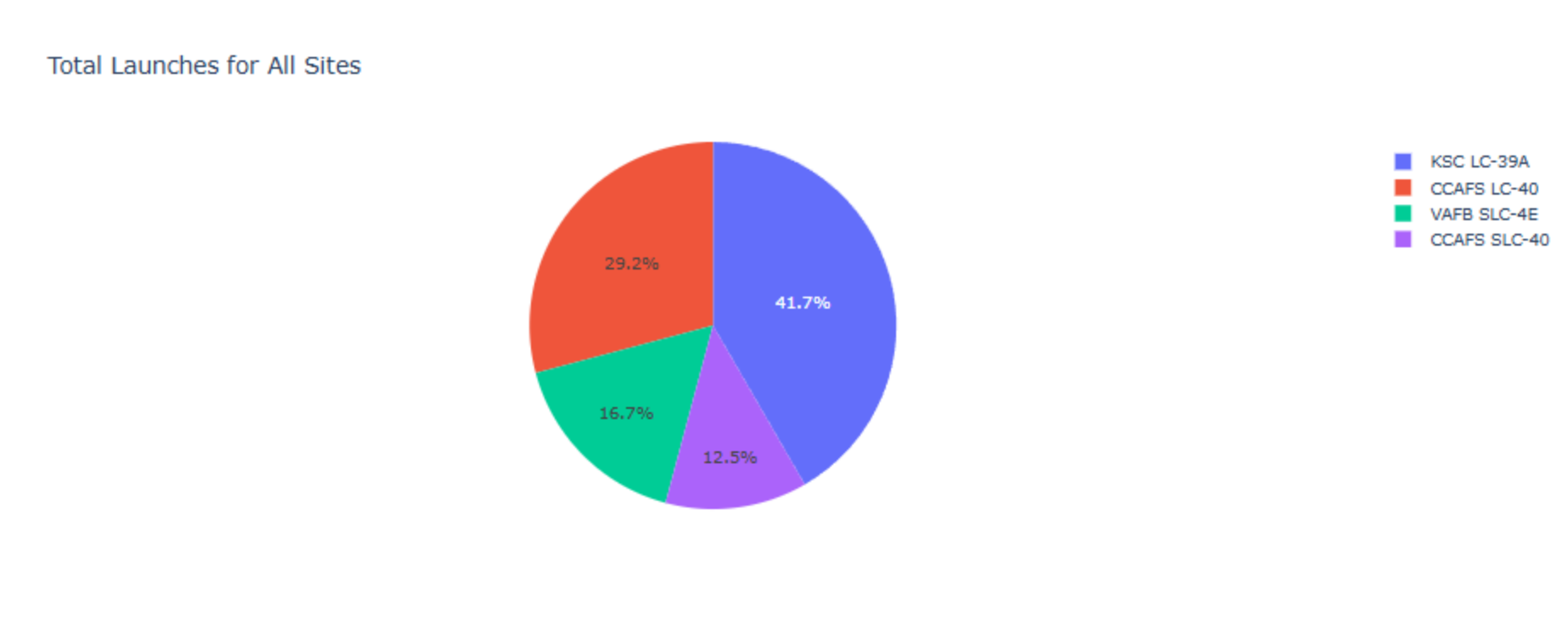
Launch Site distance to landmarks




Section 4
Build a Dashboard with Plotly Dash

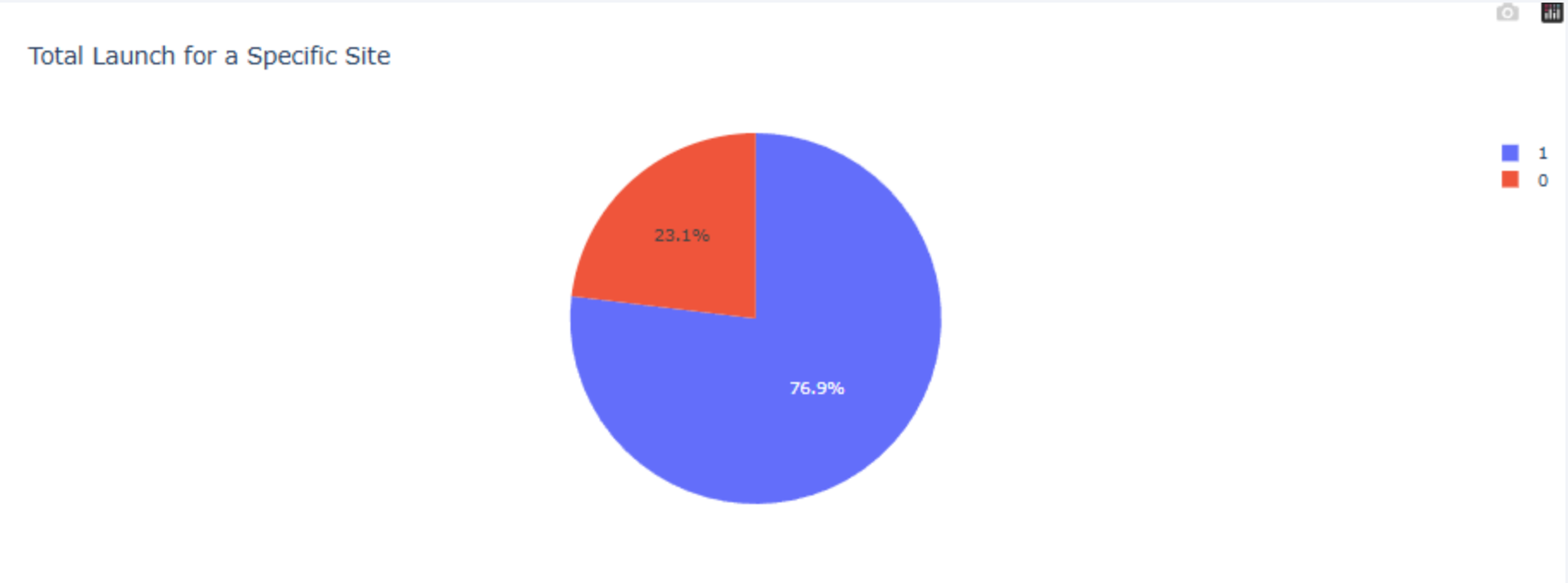
Pie chart showing the success percentage achieved by each launch site

Highest Succes Rate – KSC LC-39A, Lowest Success Rate – CCAFS SLC-40



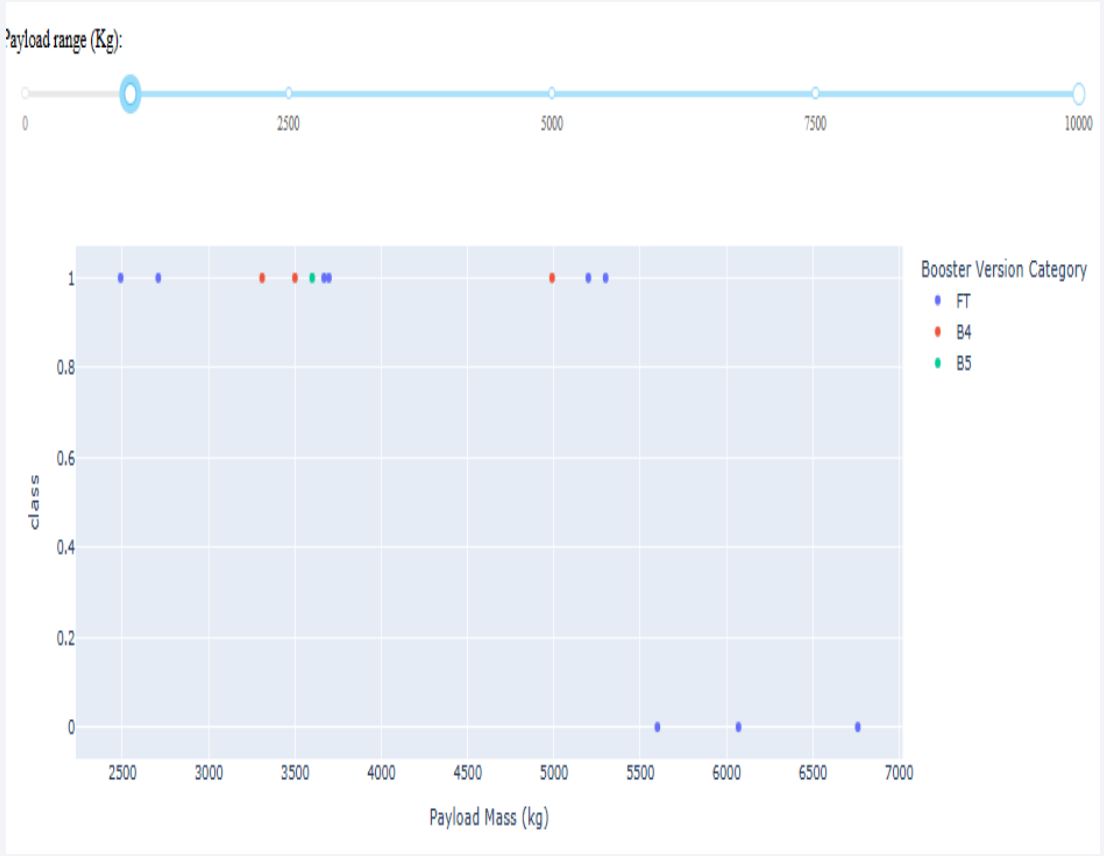
Pie chart showing the Launch site with the highest launch success ratio

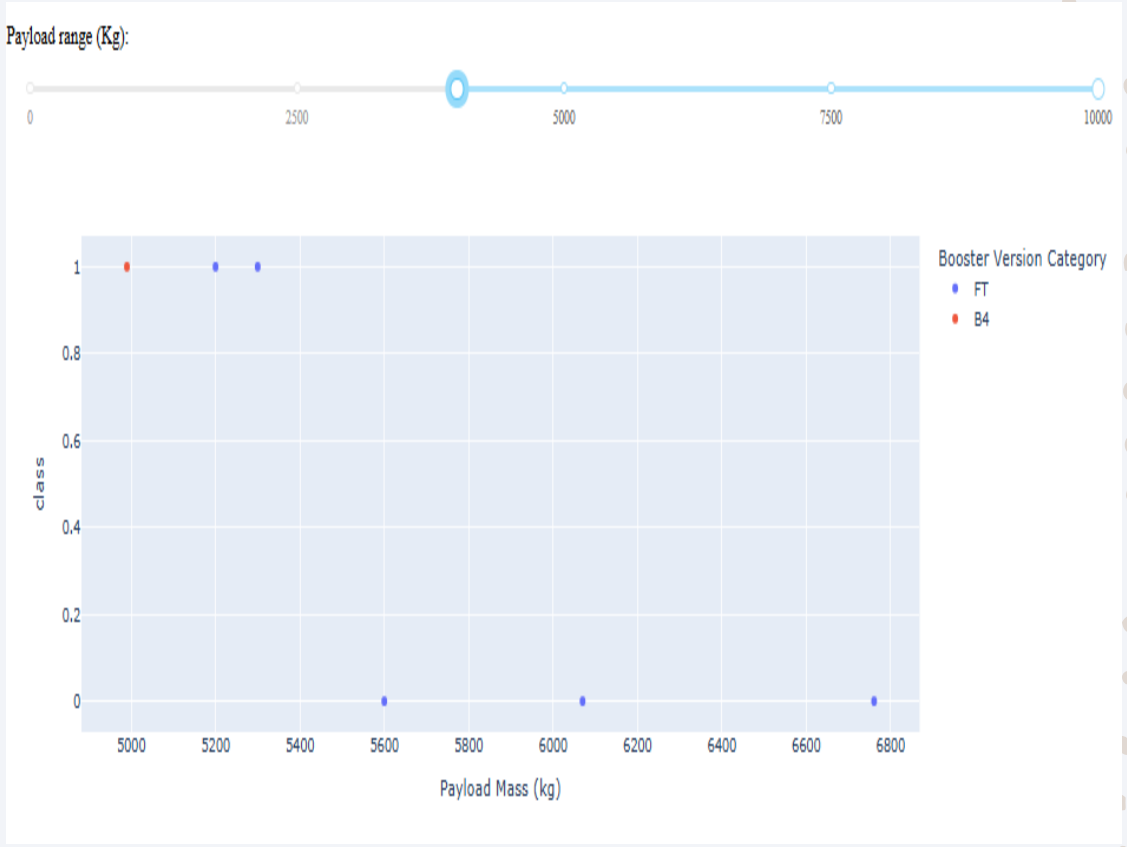
KSC LC-39A Success and Failure percentage



<Dashboard Screenshot 3>

For low mass success rate is high compared to high mass



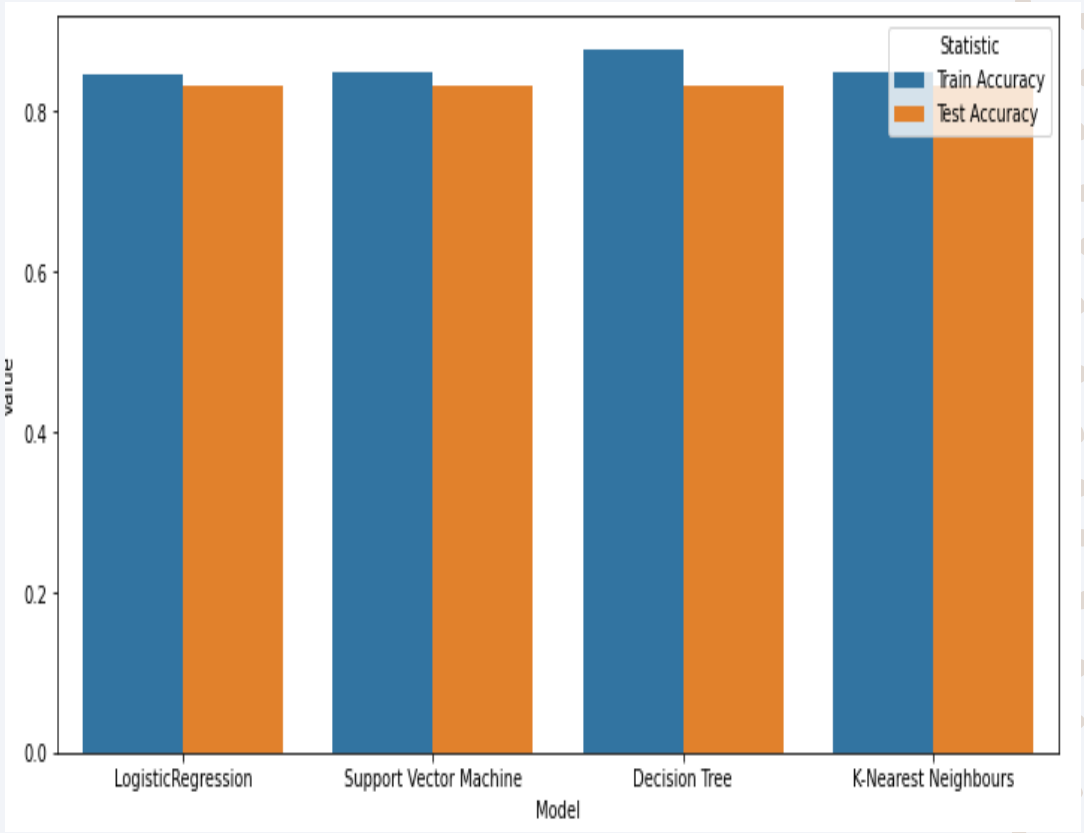


Section 5

Predictive Analysis (Classification)

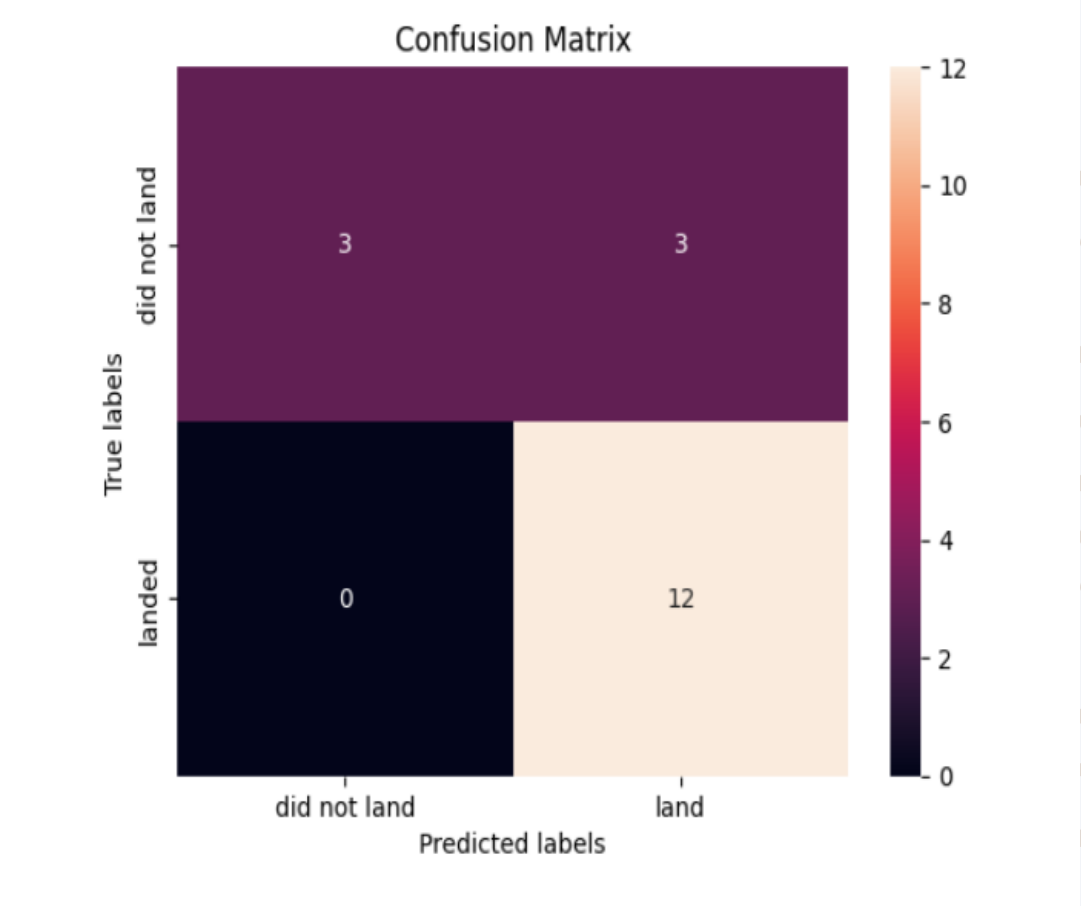
Classification Accuracy

From the graph it can be observed, that although all the models performed approximately the same, Decision Tree models had a slight edge over the other models in terms of accuracy.



Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.



Conclusions

- Decision Tree Model is the best algorithm for this dataset.
- Launches with a low payload mass show better results than launches with a larger payload mass.
- Most of launch sites are in proximity to the Equator line and all the sites are in very close proximity to the coast.
- The success rate of launches increases over the years. • KSC LC-39A has the highest success rate of the launches from all the sites.
- Orbits ES-L1, GEO, HEO and SSO have 100% success rate.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!