IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

NUJHATUL JINAN SHITHIL
10/20/2020

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection through API and web scrapping
  - Data wrangling
  - Exploratory data analysis with visualization and database query
  - Further interactive data analysis with maps and other visualizations
  - Predictive analysis with classification algorithms
- Summary of all results
  - Most successful launch site is KSC LC 39 with almost 77% of success in landing outcome
  - Rocket launch to ES-L1, GEO, HEO and SSO was the most successful when it comes to successful landing outcome
  - Decision tree algorithm proven to be the best for predictive modeling.

Introduction

- Project background and context
  - SpaceX provides the best rate in terms of cost for rocket launches than any other companies
  - The lower cost is mainly due to the fact that SpaceX can reuse its first stage
  - By predicting if the first stage will land successfully, we can determine the cost of the rocket launch.
  - If a company (e.g. Space Y) wants to compete against SpaceX, it needs to predict the cost of the launch and provide a better value than SpaceX.
- Problems you want to find answers
  - We want to find out if the first stage can be reused, meaning, if the booster will land successfully in the first stage
  - We want to find out the best classification algorithm to build the predictive model from

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

  Data were collected in two ways. They are:

  - From the SpaceX API
  - From Wikipedia by web scrapping.
- Perform data wrangling
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - We have built, tuned, and evaluated four classification models

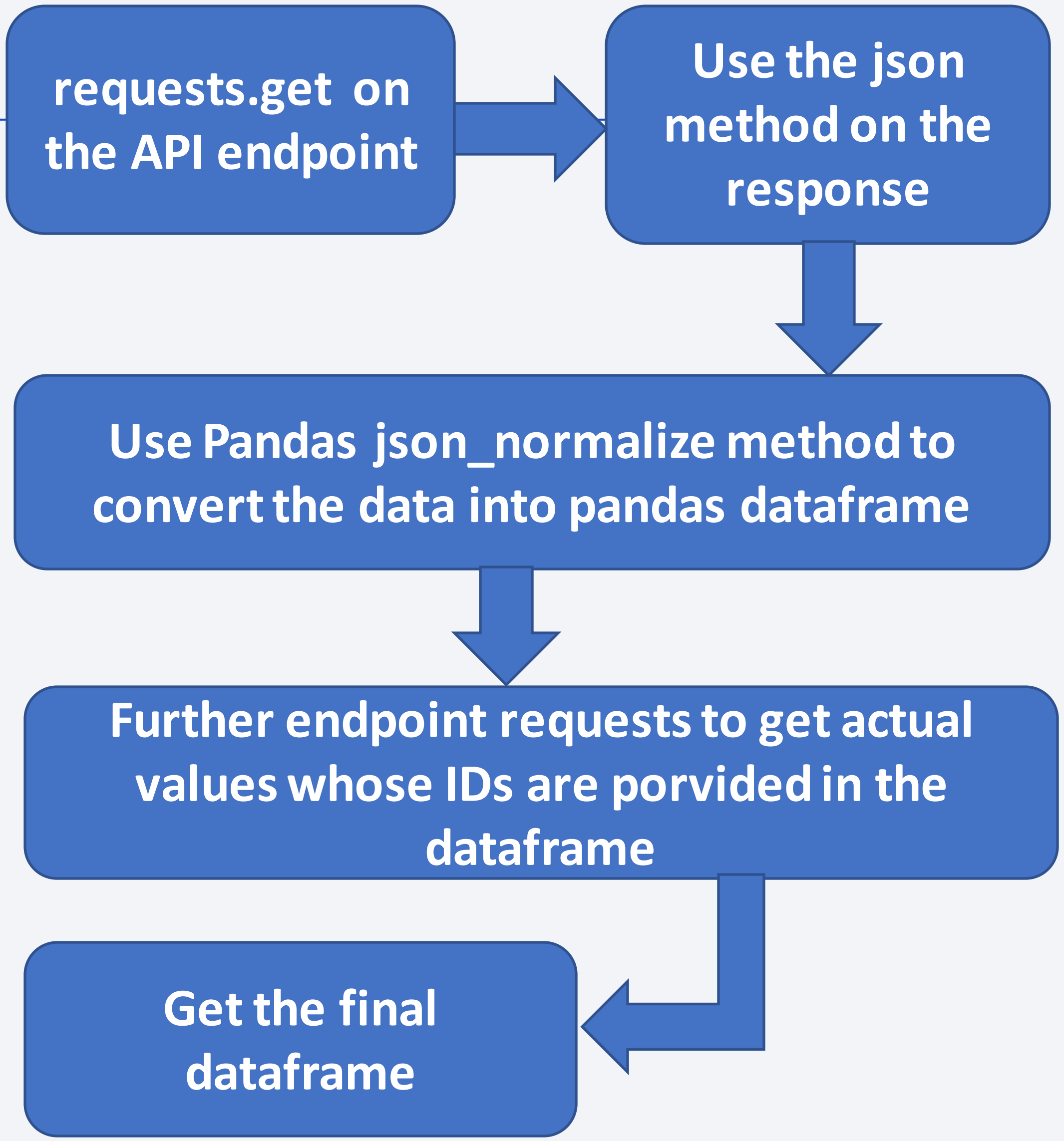
Data Collection

- Data were collected in two ways. They are:
  - From the SpaceX API
  - From Wikipedia by web scrapping.

Data Collection with SpaceX Api	Data Collection with Web Scrapping
• Get the proper URL • Use get method from requests to collect the response • Get the JSON response and then normalize it with pandas DataFrame to get the data table	• Get the proper URL • Use get method from requests to collect the response • Use an HTML parser to get the contents of the html table

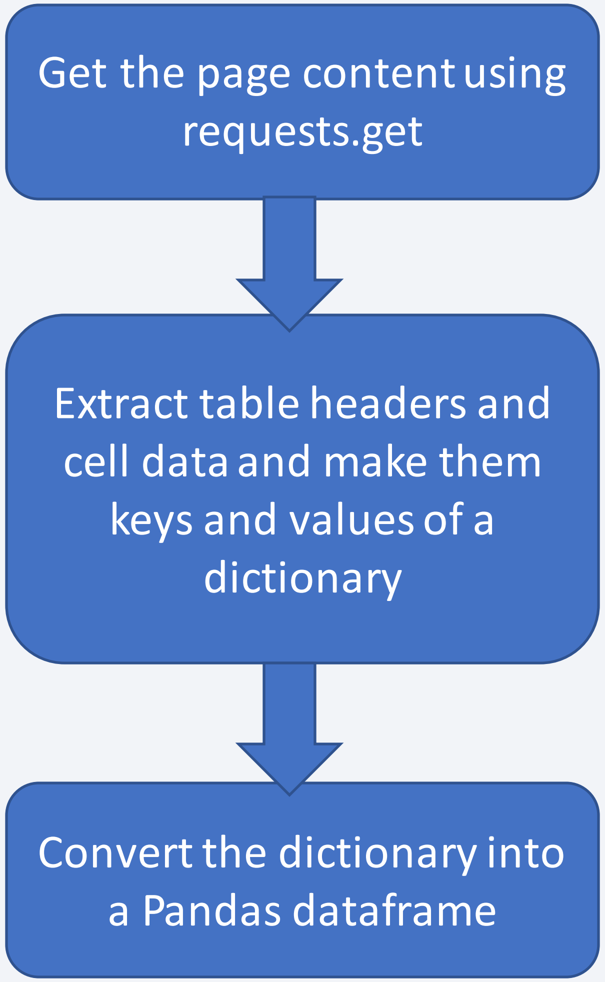
Data Collection – SpaceX API

- Requesting API endpoint
- Converting response into pandas dataframe
- Some attribute values were IDs for real values (e.g. rocket, payload, launchpad, core)
- Further requesting specific API endpoints to get actual relevant values from the given IDs.
- Finally converting those values into the final dataframe that we will use for further analysis.
- GitHub URL of the completed SpaceX API calls notebook: https://github.com/nujhatuljinan/IBMCapstone/blob/58116d3a837dd7cace4dff9bd7e7910e097208ff/spacex-data-collection-api-lab.ipynb



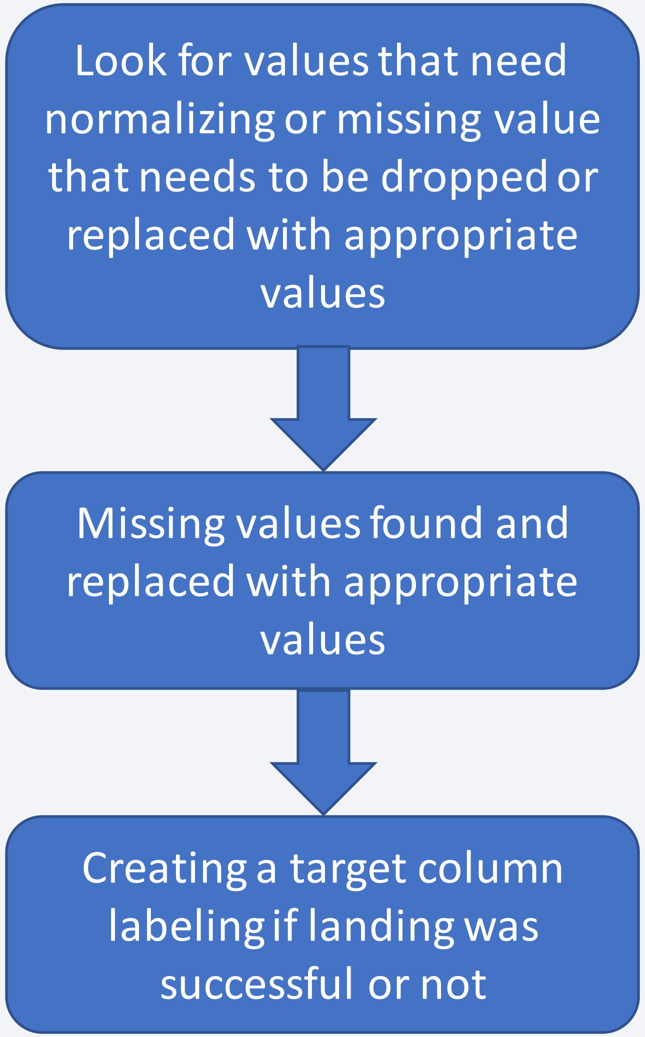
Data Collection - Scraping

- Requesting the Falcon9 Launch Wiki page from its URL
- Extracting all column names from table headers and using them as dictionary keys
- Using Beautiful Soup html parser to get the cell values and assigning them as a list of dictionary value belonging to a proper key
- Converting the dictionary to a Pandas dataframe
- GitHub URL of the completed web scraping notebook: https://github.com/nujhatuljinan/IBMCapstone/blob/58116d3a837dd7cace4dff9bd7e7910e097208ff/spaceX_data_collection_webscraping_lab.ipynb



Data Wrangling

- After converting raw data into a dataframe, missing values were found.
- Missing values regarding payload mass was replaced with the average payload mass.
- Missing values regarding the landing pads remained none, indicating when they were not used.
- Creating a class column that shows if the landing outcome was successful or not and label them.
- GitHub URL of the completed data wrangling related notebooks: https://github.com/nujhatuljinan/IBMCapstone/blob/58116d3a837dd7cace4dff9bd7e7910e097208ff/spacex-Data%20wrangling_lab.ipynb



EDA with Data Visualization

Three types of plots were used for visualization. They are:

1. Scatter plots: Scatter plots were used to visualize relationship between Flight Number and Payload, Flight Number and Launch Site, Payload and Launch Site, Flight Number and Orbit type, and Payload and Orbit type.

2. Bar charts: Bar chart was used to visualize the relationship between success rate of each Orbit type.

3. Line graph: Line graph was used to visualize the trend of success rate throughout the years.

GitHub URL of the completed EDA with data visualization notebook: https://github.com/nujhatuljinan/IBMCapstone/blob/58116d3a837dd7cace4dff9bd7e7910e097208ff/spaceX-labs-eda-dataviz_lab.ipynb

EDA with SQL

Some SQL queries performed:

- Finding out unique launch sites, total payload mass by NASA
- Date of the first landing outcome, total number of successful and failure mission outcome
- Names of boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000, names of the booster_versions which have carried the maximum payload mass
- These queries were performed to figure out the factors involved in the success of a landing outcome.
- GitHub URL of the completed EDA with SQL notebook: https://github.com/nujhatuljinan/IBMCapstone/blob/58116d3a837dd7cace4dff9bd7e7910e097208ff/jupyter-labs-eda-sql-coursera_sqllite_completed.ipynb

Build an Interactive Map with Folium

Some "map objects" added on the map:

- Folium Circle object to the map to mark the unique launch sites.
- Folium Marker cluster object to mark the launch sites with red and green marks according to failure and success respectively from those launch sites.
- Used Folium Marker to mark the distance between nearest coastline, railway, highway and city, also used Polyline object to draw a line between them and added to the map.
- These markers were added in order better understand the relationship between launch sites and the mission outcome as well as to better understand the surroundings with respect to the launch sites.
- Folium lab notebook GitHub URL: https://github.com/nujhatuljinan/IBMCapstone/blob/58116d3a837dd7cace4dff9bd7e7910e097208ff/spaceX_launch_site_location_lab.ipynb

Build a Dashboard with Plotly Dash

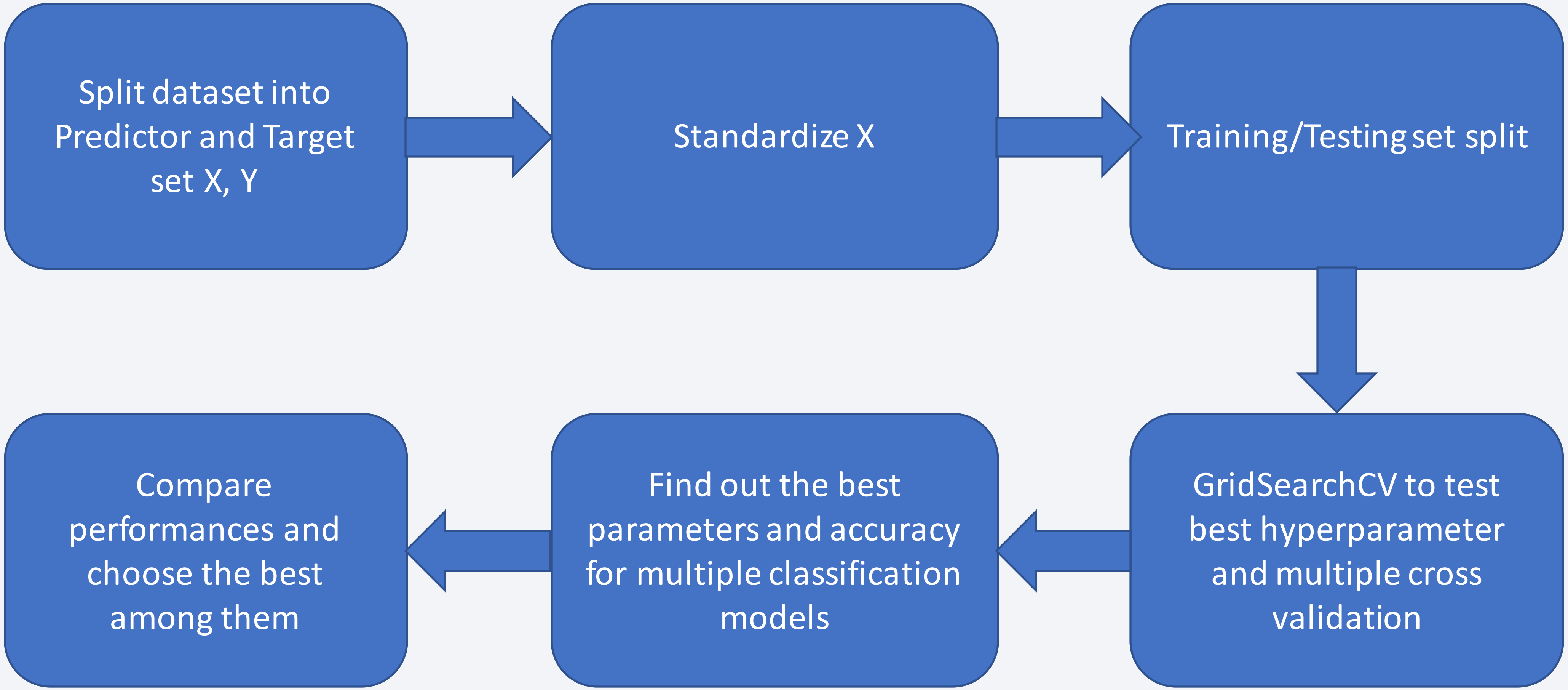
- Added an interactive pie chart that shows the success rate among all sites and success to failure rate in individual launch sites.
- Added an interactive scatter plot representing success and failure due to various payload mass and also for different launch sites.
- These charts allow interaction and make exploratory data analysis easy to find the relationship between different variables.
- GitHub URL to the python file: https://github.com/nujhatuljinan/IBMCapstone/blob/58116d3a837dd7cace4dff9bd7e7910e097208ff/spaceX_plotly_dash_lab.py

Predictive Analysis (Classification)

Process:

- Split the dataset into predictor and target data X and Y respectively
- Standardized X in order to reduce data biases
- Split the data into training and testing sets using train_test_split()
- Used GridSearchCV model in order to select a range of hyper parameters to achieve the best result from the most suitable hyper parameter and 10 folds cross validation as well.
- Used Logistic Regression, SVM, Decision tree, and KNN classification algorithms and compared the best accuracy among them.
- GitHub URL: https://github.com/nujhatuljinan/IBMCapstone/blob/58116d3a837dd7cace4dff9bd7e7910e097208ff/SpaceX_Machine%20Learning%20Prediction_lab.ipynb

Prediction Analysis (Classification) Flow Chart

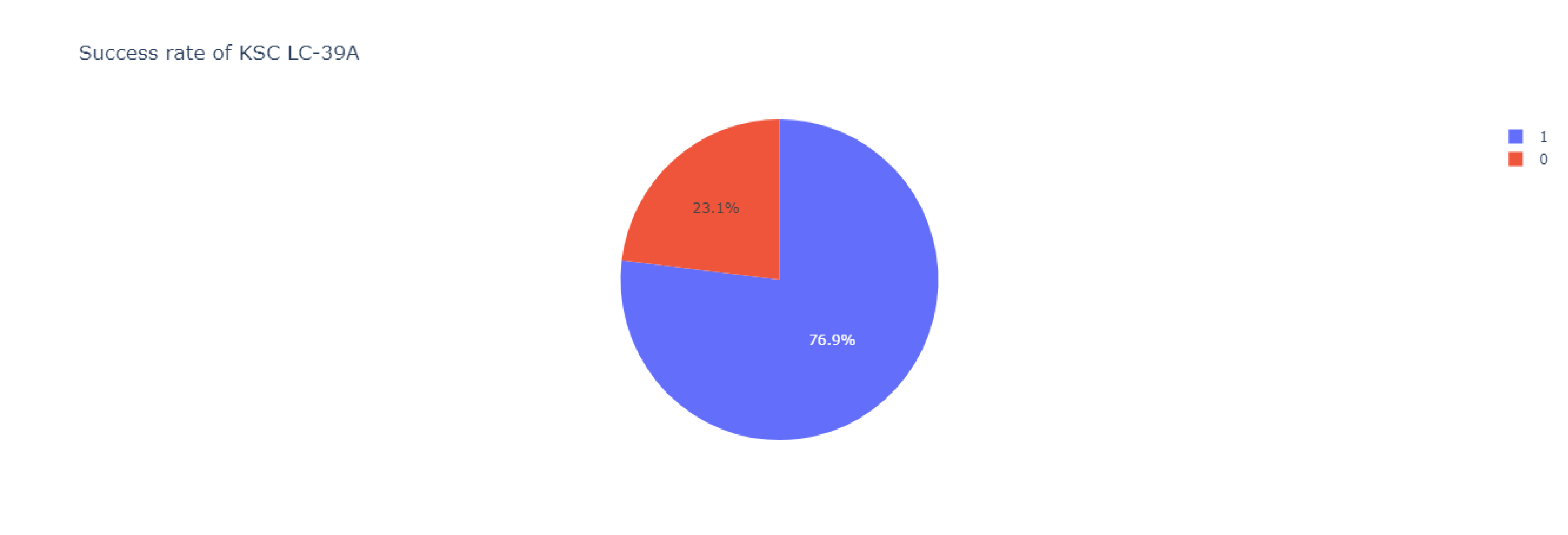


Results

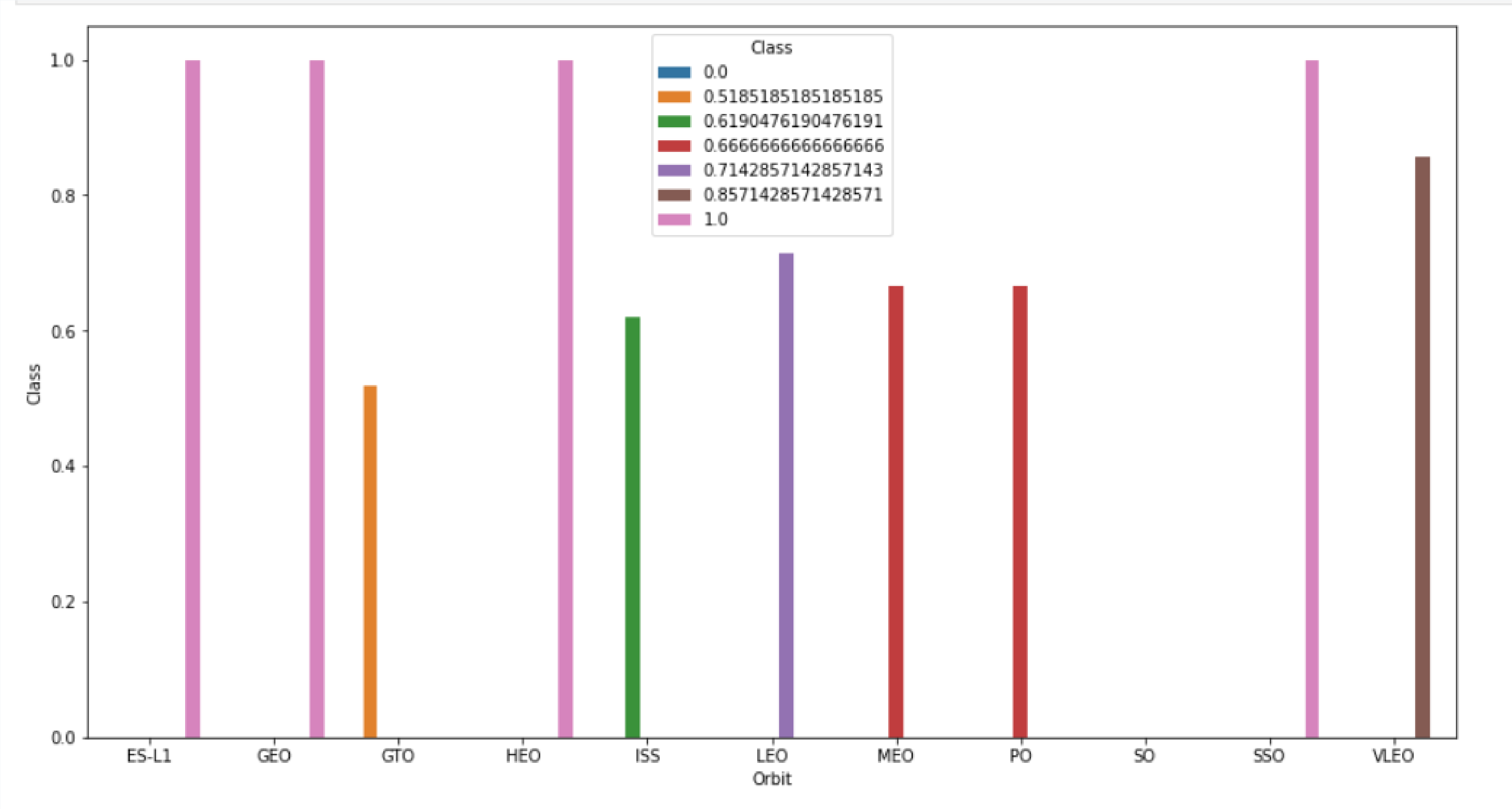
Exploratory data analysis results

- Most successful launch site is KSC LC 39 with almost 77% of success in landing outcome
- Rocket launch to ES-L1, GEO, HEO and SSO was the most successful when it comes to successful landing outcome
- KSC LC 39A and VAFB SLC 4E were most successful in landing outcomes when payload mass is between 2000 kg and 6000 kg

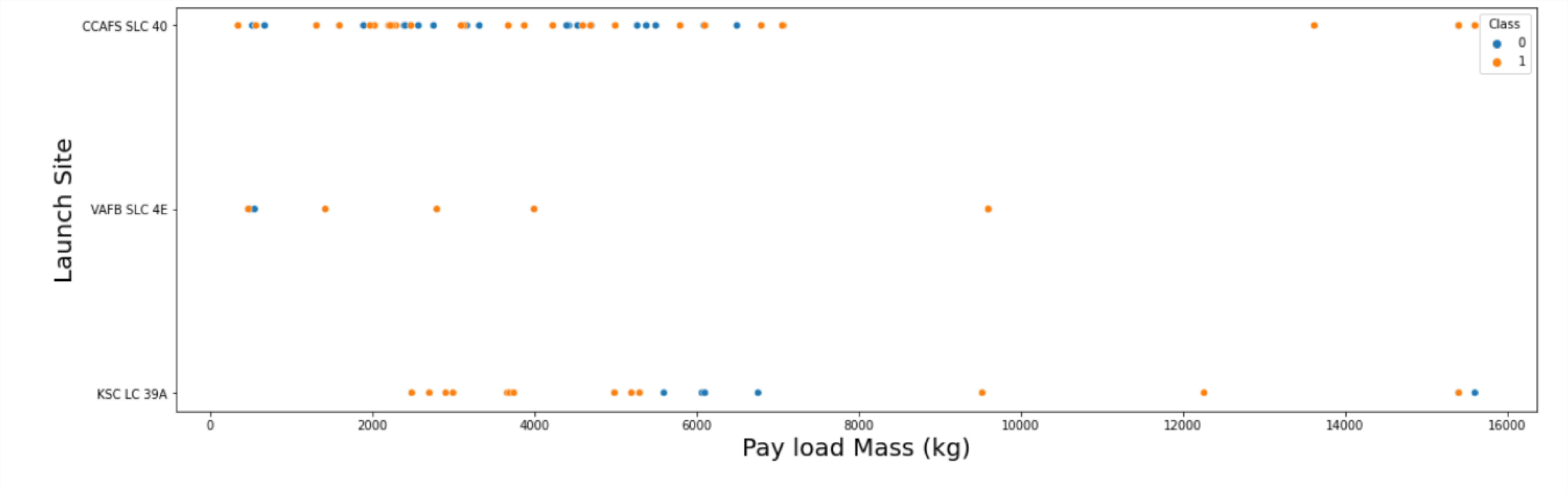
KSC LC-39A launch site success rate



Success rate of each orbit



Launch Site vs Payload Mass



Predictive Data Analysis Results:

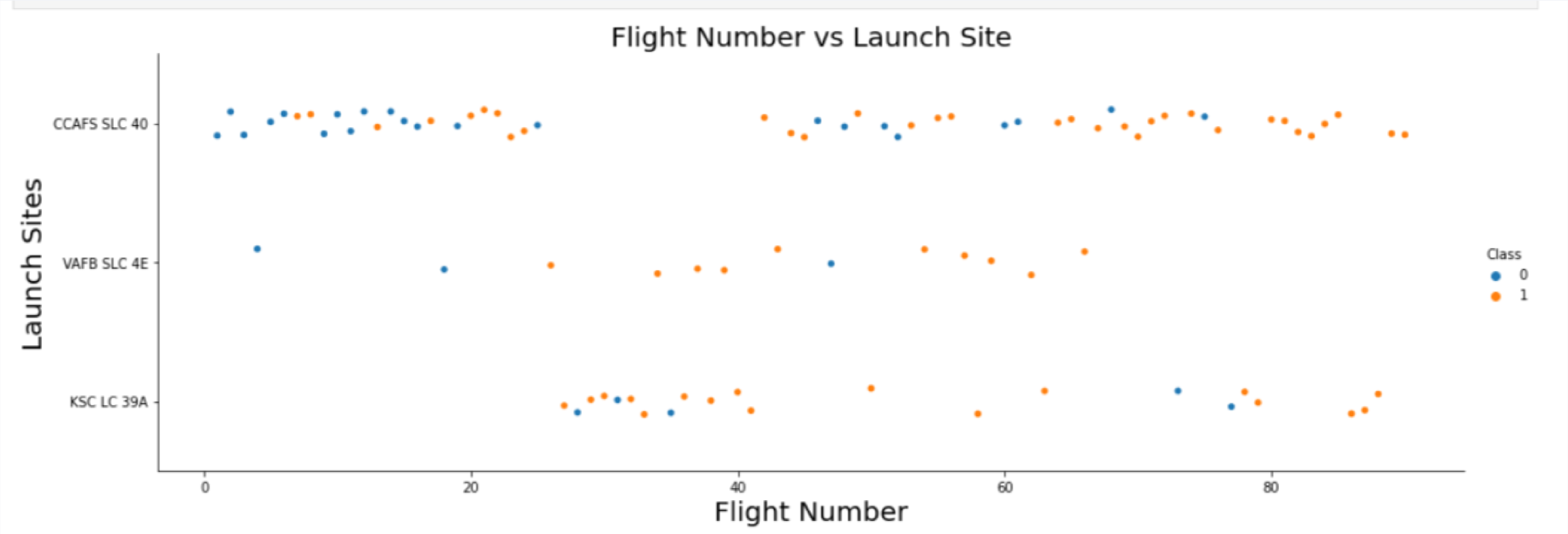
- Logistic Regression, Decision Tree, KNN and SVM all had test set accuracy of 83.333% and all have the same confusion matrix.
- Decision Tree had the highest training set accuracy of 88.75%

	Methods	Validation Set Accuracy	Test Set Accuracy
0	Logistic Regression	0.846429	0.833333
1	SVM	0.848214	0.833333
2	Decision Tree	0.887500	0.833333
3	KNN	0.848214	0.833333

Section 2

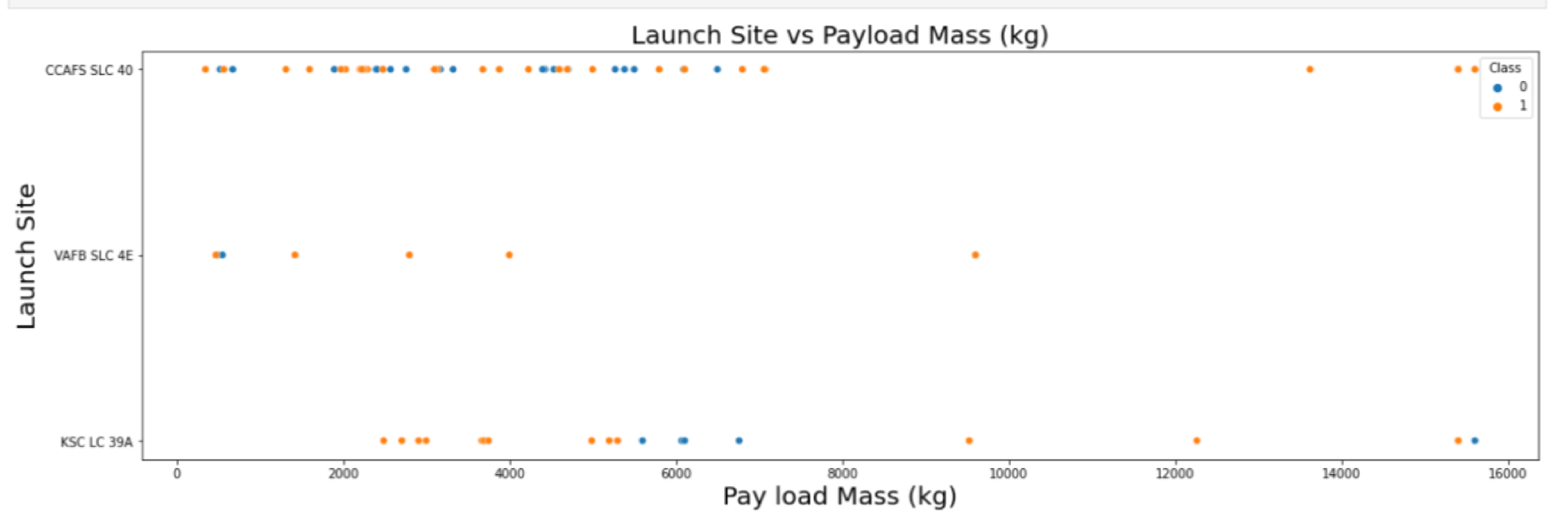
Insights drawn from EDA

Flight Number vs. Launch Site



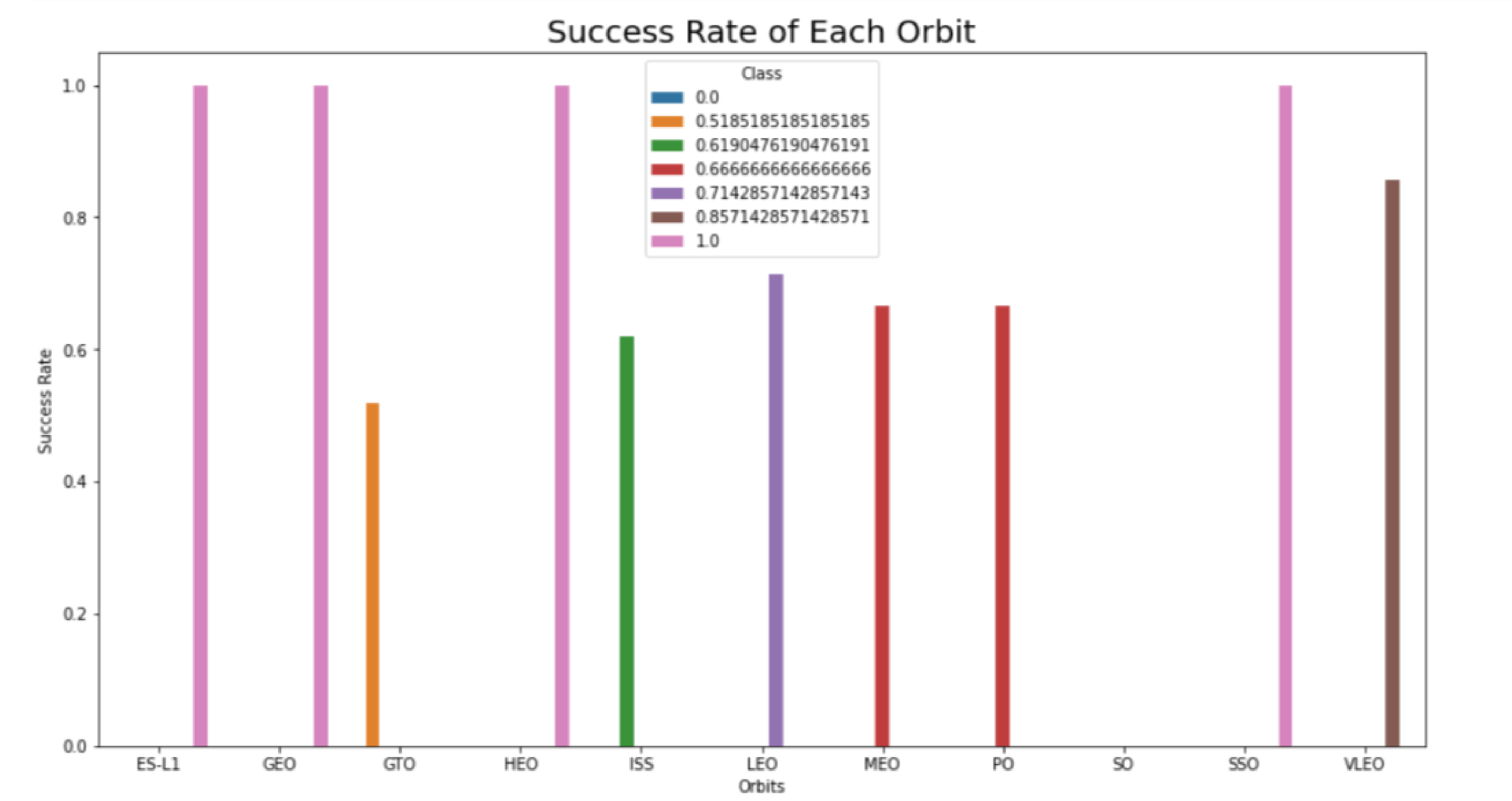
- KSC LC 39A has the most success with landing outcomes from other launch sites

Payload vs. Launch Site



- VAFB SLC 4E and KSC LC 39A, both of these launch sites were really successful when carrying payloads between 2000 kg and 6000 kg
- For VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).

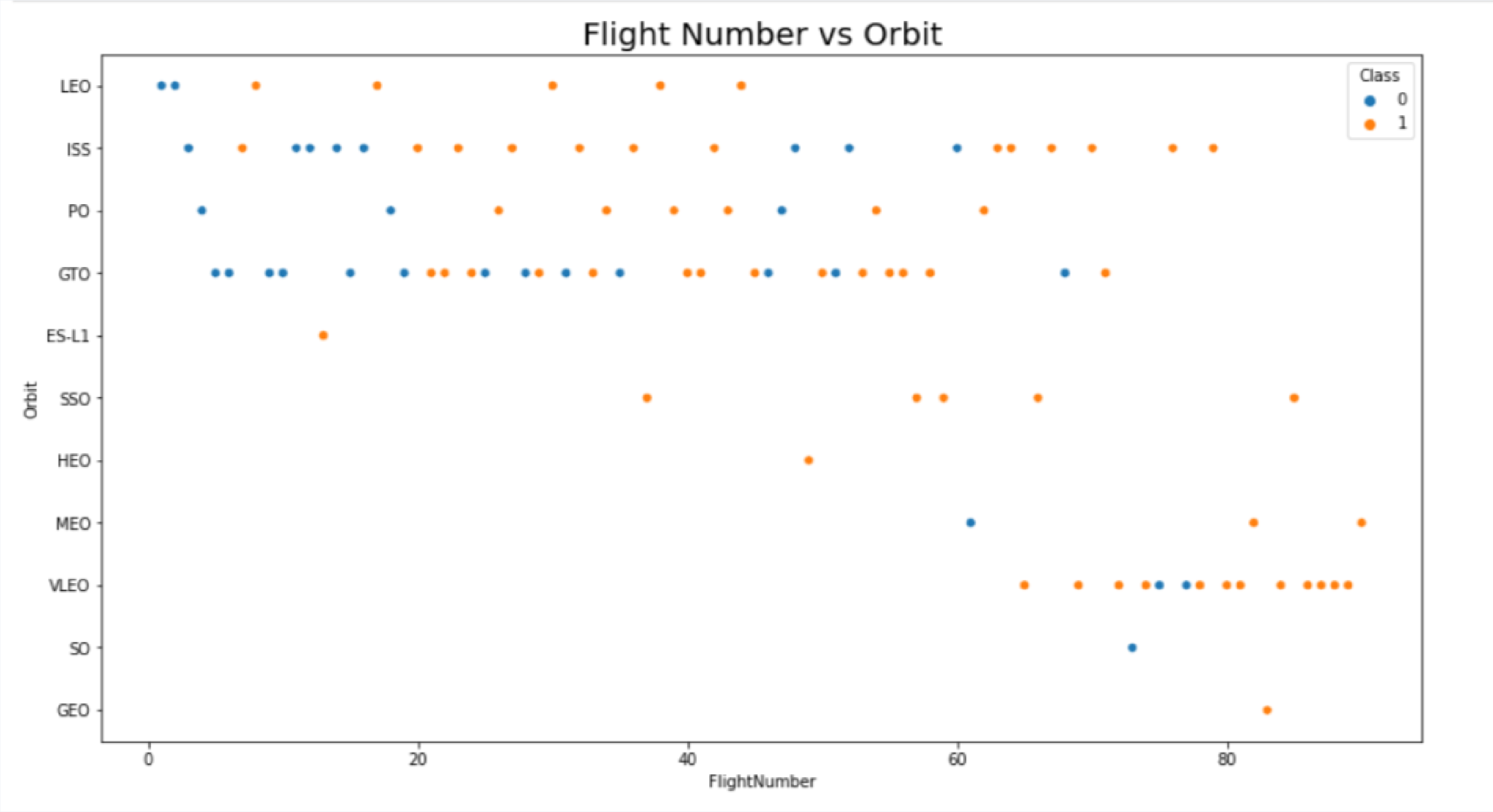
Success Rate vs. Orbit Type



Most successful orbits in terms of landing outcomes are -

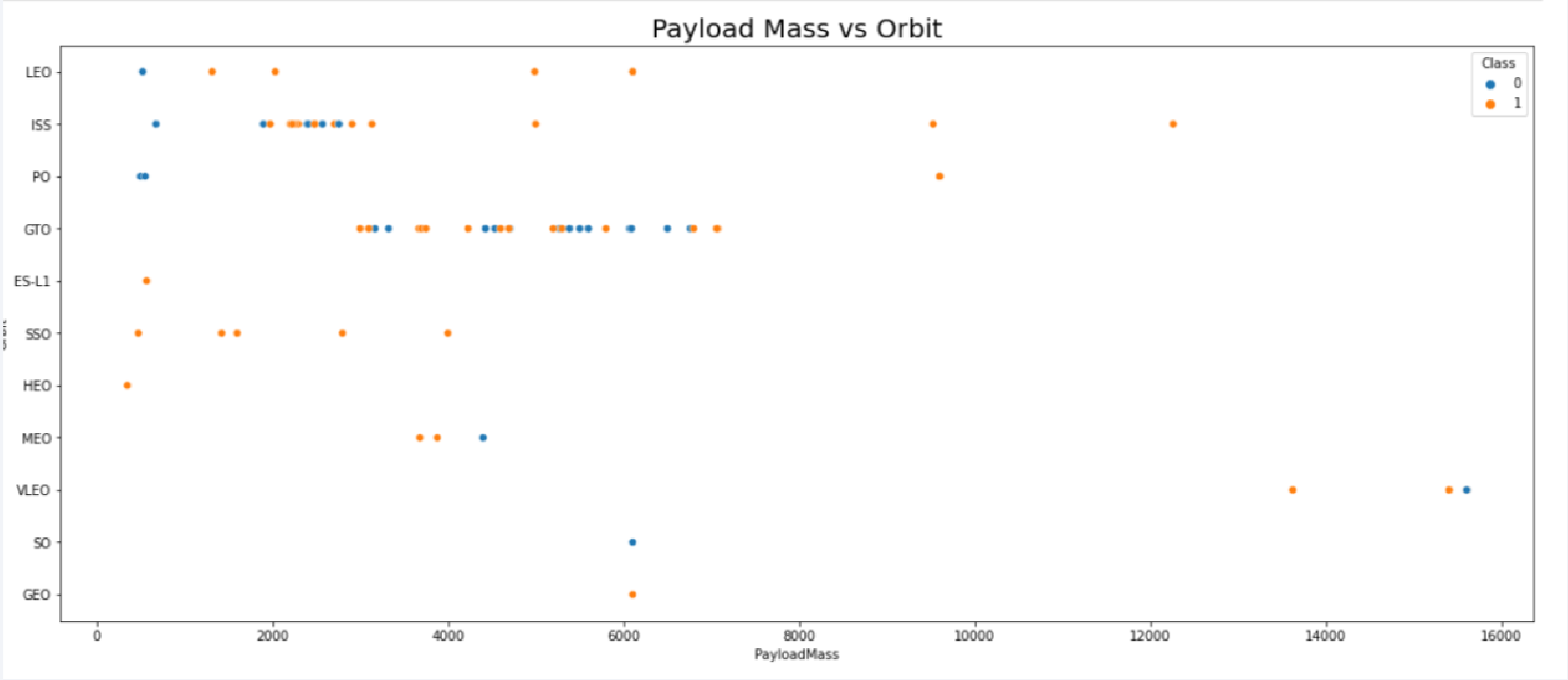
- ES-L1
- GEO
- HEO
- SSO

Flight Number vs. Orbit Type



- All SSO flights were successful
- Other than that there is no visible relationship between Orbit types and Flight Numbers

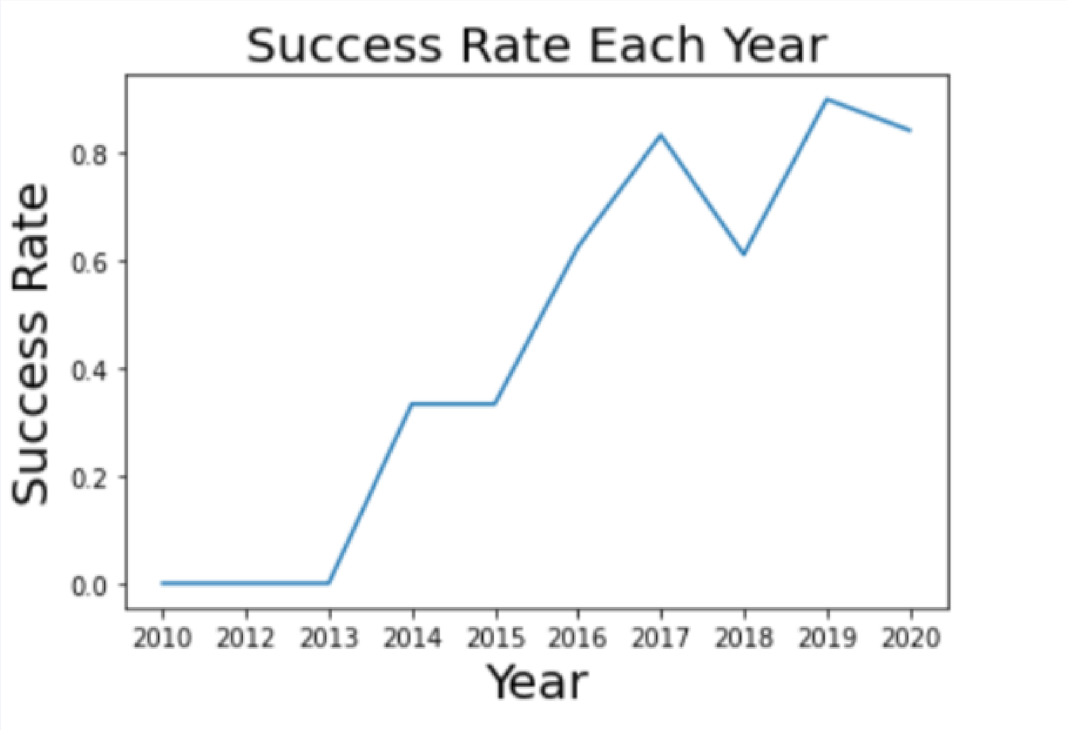
Payload vs. Orbit Type



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

Launch Success Yearly Trend

- Success rate kept increasing from its inception.
- Only decrease in 2018.



All Launch Site Names

```
[8]: %%sql
SELECT DISTINCT "Launch_Site"
FROM SPACEXTBL

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

[8]:

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

- DISTINCT keyword is used in order to select unique launch site names from the SPACEXTBL relationship table
- Four unique launch sites are shown; namely, CCAFS LC-40, VAFB SLC-4E, KSC LC-39A,and CCAFS SLC-40

Launch Site Names Begin with 'CCA'

[9]:
```
%%sql
SELECT *
FROM SPACEXTBL
WHERE "Launch_Site" LIKE "CCA%"
LIMIT 5
```

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

[9]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- Queried for the first five rows where launch site name begins with 'CCA' from the relationship table using LIKE keyword to match the string

Total Payload Mass

```
%%sql
SELECT SUM("PAYLOAD_MASS__KG_") AS "TOTAL PAYLOAD MASS BY NASA (CRS)"
FROM SPACEXTBL
WHERE SPACEXTBL.Customer = "NASA (CRS)"
```

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

TOTAL PAYLOAD MASS BY NASA (CRS)
45596

- Total payload mass sent by NASA is 45596 kg .
- In the WHERE clause Customer property is predicated to be "NASA (CRS)" as it is listed like this in the relationship table.

Average Payload Mass by F9 v1.1

```
[11]: %%sql
      SELECT AVG("PAYLOAD_MASS__KG_") AS "AVERAGE PAYLOAD MASS BY F9 v1.1"
      FROM SPACEXTBL
      WHERE SPACEXTBL."Booster_Version" = "F9 v1.1"

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

[11]:

AVERAGE PAYLOAD MASS BY F9 v1.1
2928.4

- AVG aggregate function was used to find out the average payload mass
- For F9 v1.1 booster version average payload mass happens to be 2928.4 kg

First Successful Ground Landing Date

```
[12]: %%sql
SELECT "Date"
FROM SPACEXTBL
WHERE SPACEXTBL."Landing _Outcome" = "Success (ground pad)"
ORDER BY substr(Date, 7, 4) ASC, substr(Date, 4, 2) ASC, substr(Date, 1, 2) ASC
LIMIT 1
```

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

[12]:

Date
22-12-2015

- As the query was done on sqlite3, some functions were not available so ORDER BY is used to order the date in ascending order using substr() function.
- Only first date is shown from the matched query as only the first successful ground landing date war required

Successful Drone Ship Landing with Payload between 4000 and 6000

```
[13]: %%sql
SELECT "Booster_Version"
FROM SPACEXTBL
WHERE "Landing _Outcome" = "Success (drone ship)" AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000

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

[13]:

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

- Booster versions that were successful to land on drone ship and carried payload mass between 4000kg and 6000kg were shown.
- There were 4 of them as shown on the screenshot.

Total Number of Successful and Failure Mission Outcomes

```
[14]: %%sql
SELECT COUNT("Mission_Outcome"), "Mission_Outcome"
FROM SPACEXTBL
GROUP BY "Mission_Outcome"

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

[14]:

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

- Count of mission outcome from the SPACEXTBL relation table
- GROUP BY was used to group the mission outcomes in the existing categorical values present in the table

Boosters Carried Maximum Payload

[15]:
```
%%sql
SELECT "Booster_Version"
FROM SPACEXTBL
WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

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

[15]:

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3

[15]:

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

- 12 booster versions in total were the ones that carried maximum payloads among all the boosters.

2015 Launch Records

[16]:
```
%%sql

SELECT
"Date",
CASE
WHEN SUBSTR(Date,4,2) = '01' THEN 'January'
WHEN SUBSTR(Date,4,2) = '02' THEN 'February'
WHEN SUBSTR(Date,4,2) = '03' THEN 'March'
WHEN SUBSTR(Date,4,2) = '04' THEN 'April'
WHEN SUBSTR(Date,4,2) = '05' THEN 'May'
WHEN SUBSTR(Date,4,2) = '06' THEN 'June'
WHEN SUBSTR(Date,4,2) = '07' THEN 'July'
WHEN SUBSTR(Date,4,2) = '08' THEN 'August'
WHEN SUBSTR(Date,4,2) = '09' THEN 'September'
WHEN SUBSTR(Date,4,2) = '10' THEN 'October'
WHEN SUBSTR(Date,4,2) = '11' THEN 'November'
WHEN SUBSTR(Date,4,2) = '12' THEN 'December'
END AS Month, "Landing _Outcome", "Booster_Version","Launch_Site"
FROM SPACEXTBL
WHERE "Landing _Outcome" = "Failure (drone ship)" AND SUBSTR(Date,7,4) = "2015"
```

* sqlite:///my_data1.db

5]:

Date	Month	Landing _Outcome	Booster_Version	Launch_Site
10-01-2015	January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
14-04-2015	April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- In the year 2015, two drone ship failure occurred.
- Related booster version and launch site were queried to be shown.
- Month name was extracted from the date column using CASE END

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

```
[17]: %%sql
SELECT COUNT("Landing _Outcome"), "Landing _Outcome"
FROM (SELECT *
      FROM SPACEXTBL
      ORDER BY substr(Date, 7, 4) DESC, substr(Date, 4, 2) DESC, substr(Date, 1, 2) DESC)
WHERE "Landing _Outcome" LIKE "Success%"
AND substr("Date", 7, 4) <= "2017"
AND NOT(substr("Date", 7, 4) = "2017" AND substr("Date",4,2) > "03")
AND NOT(substr("Date", 7, 4) = "2017" AND substr("Date",4,2) = "03" AND substr("Date",1,2) > "20" )
GROUP BY "Landing _Outcome"

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

[17]:

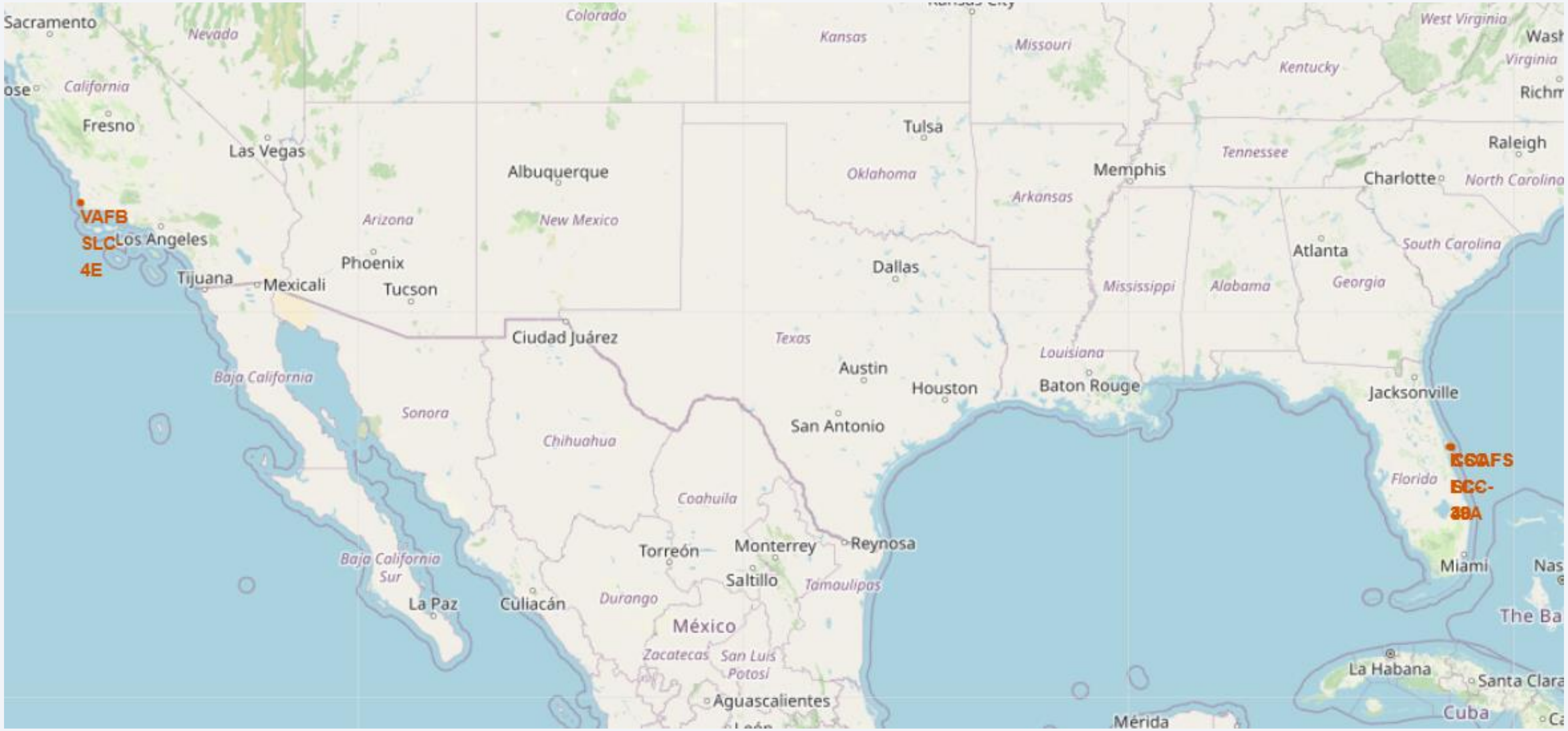
COUNT("Landing _Outcome")	Landing _Outcome
5	Success (drone ship)
3	Success (ground pad)

- As we were using SQLite3 ORDER BY was used to order the dates in the descending order using substr()
- Multiple logics were used in the WHERE section in order to choose between the titled dates
- In that time period 5 successful drone ship landing and 3 successful ground pad landing occurred

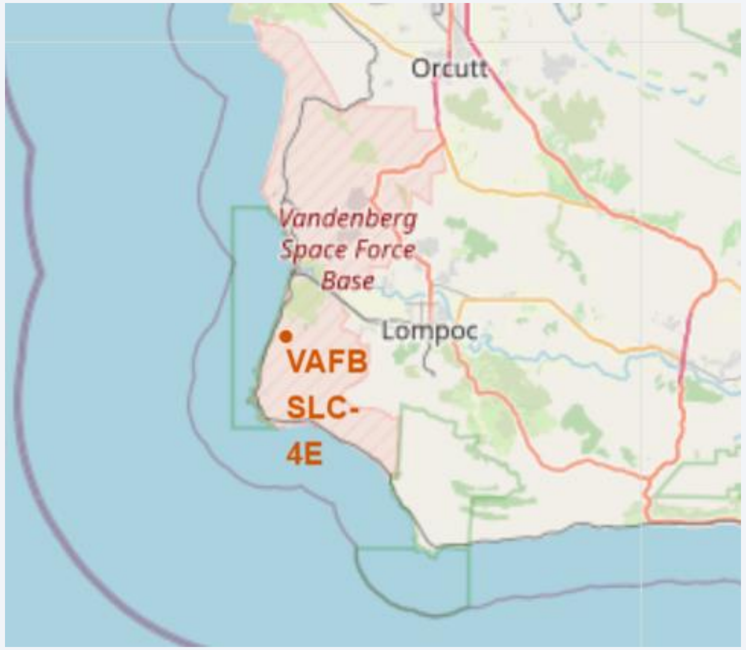
Section 3

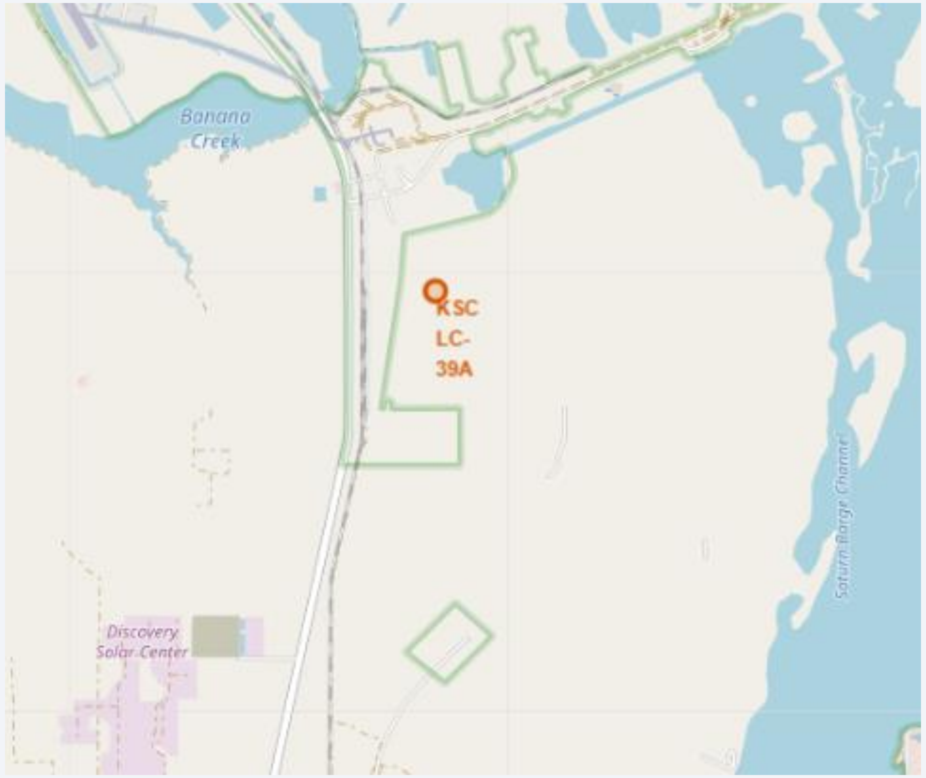
Launch Sites Proximities Analysis

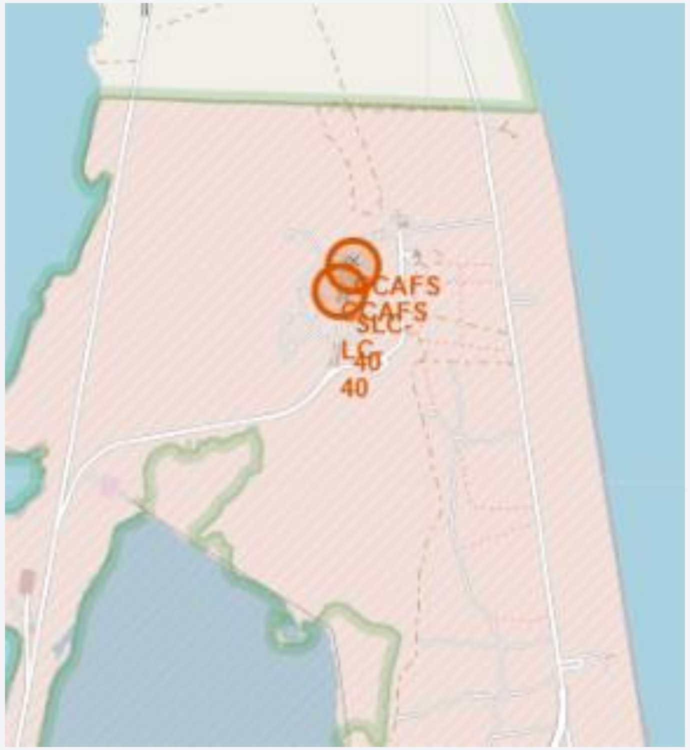
Marked Launch Site Locations



Zoomed out view of all for launch site locations for SpaceX. One in California and three others in Florida

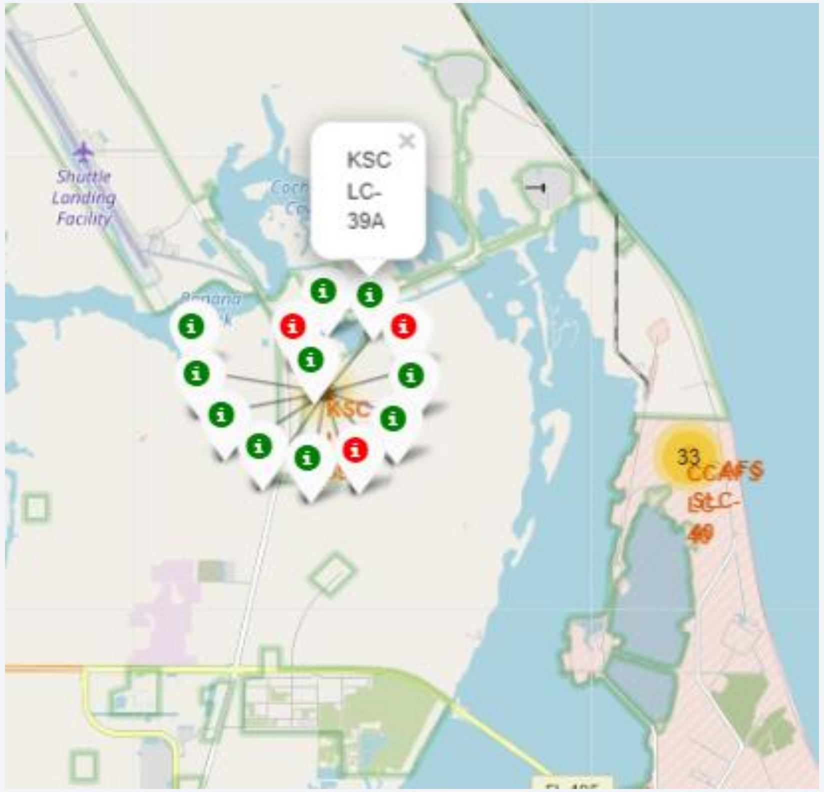






These pictures show a more zoomed in look of the map containing the marked launch site with Circle marker objects in the map and also the Marker object to mark them with the respective launch site names.

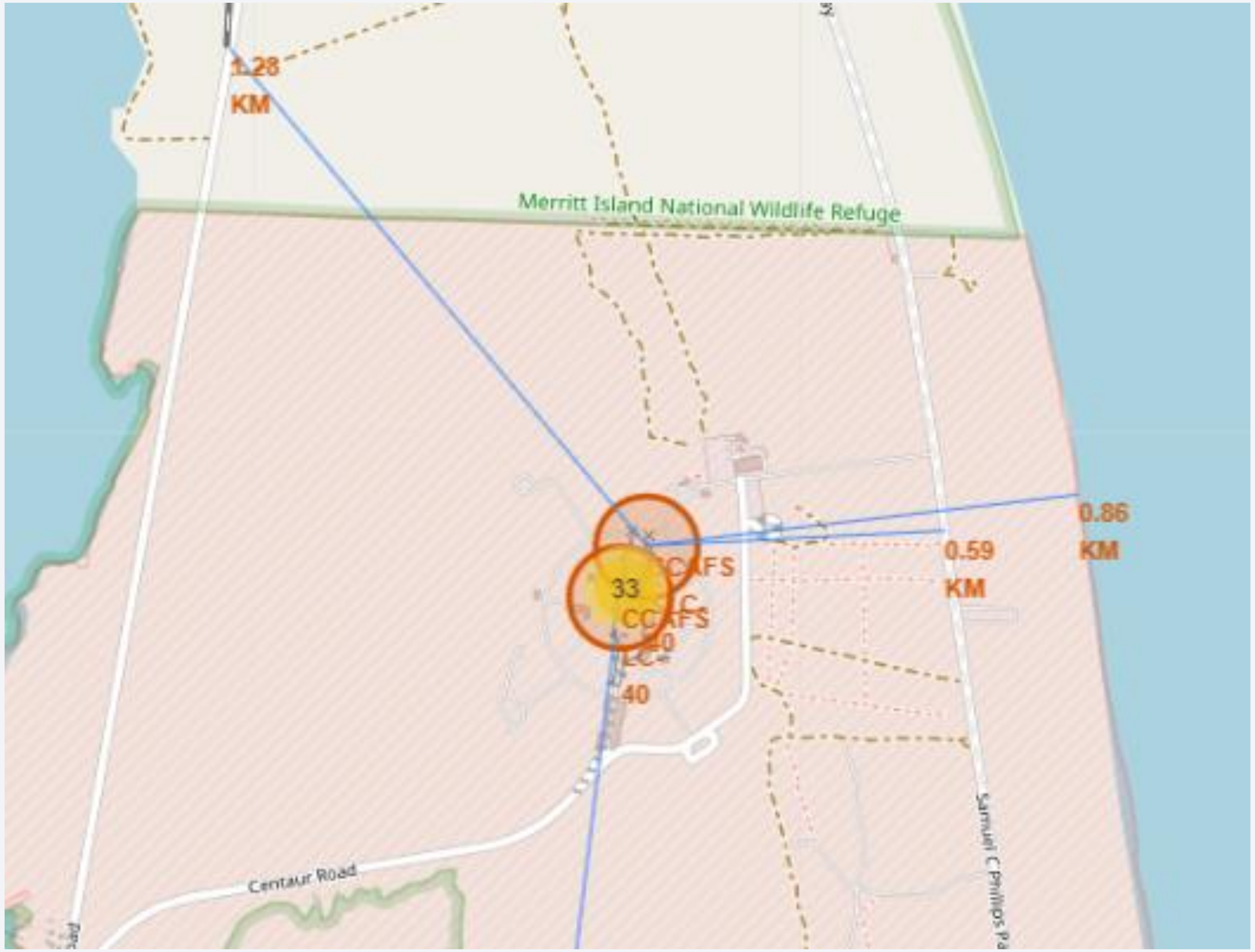
Launch Sites Marked with Success and Failure



Launch sites were clustered with number of launches and each launch in marked with red color representing a failure in landing outcome and green representing a success in landing outcome. These two picture shows the landing outcome of VAFB SLC-4E and KSC LC-39A launch site. The later have more success rate.

These two pictures shows the marked successful and failure landings of other two sites ,namely, CCAFS LC-40 and CCAFS SLC-40
CCAFS LC-40 had far more launches than the other one.

Closest Coastline, Railway, Highway and City



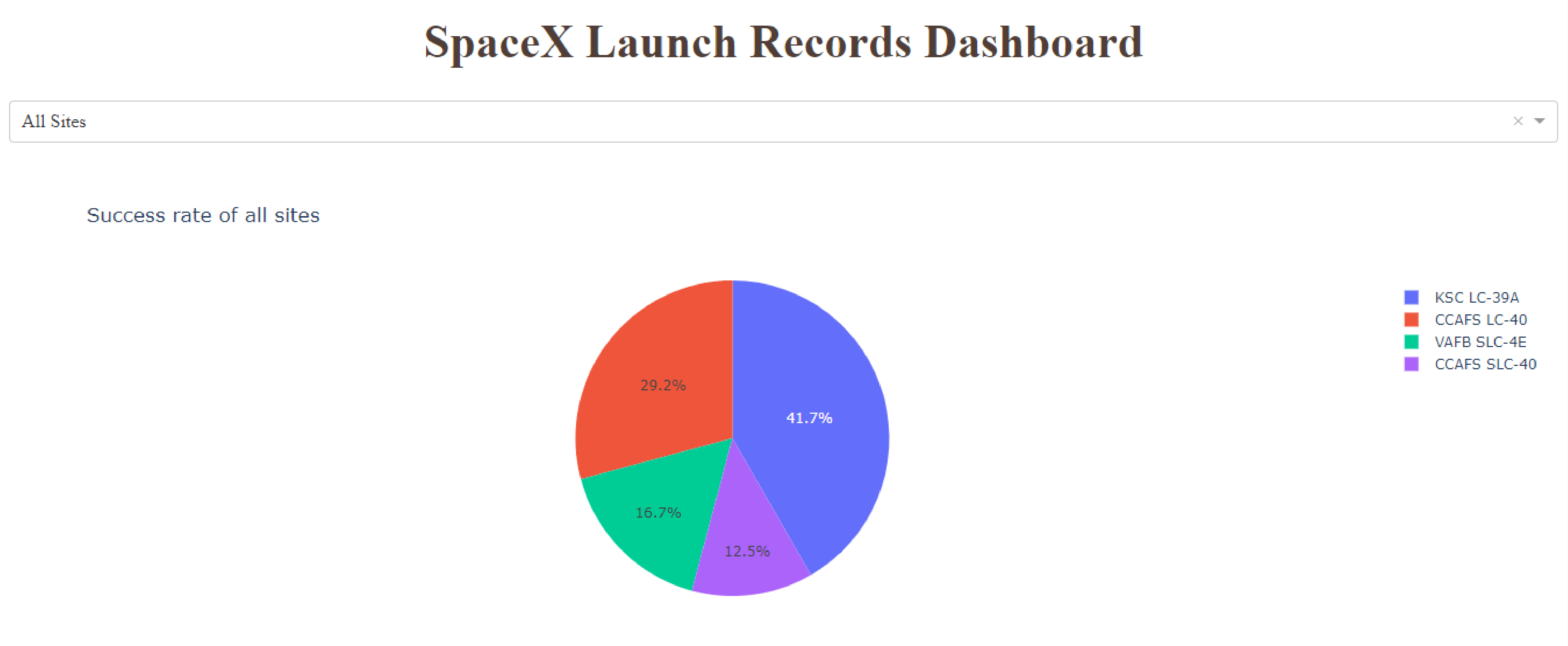
- The closest coastline is from CCAFS SLC-40 and it is 0.86 km far away.
- The closest railway is 1.28 km away as it is shown in the picture.
- The closest highway is 0.59 km away

Closest city is Melbourne which is 51.08 km away and this city is calculated from Florida launch sites

Section 4

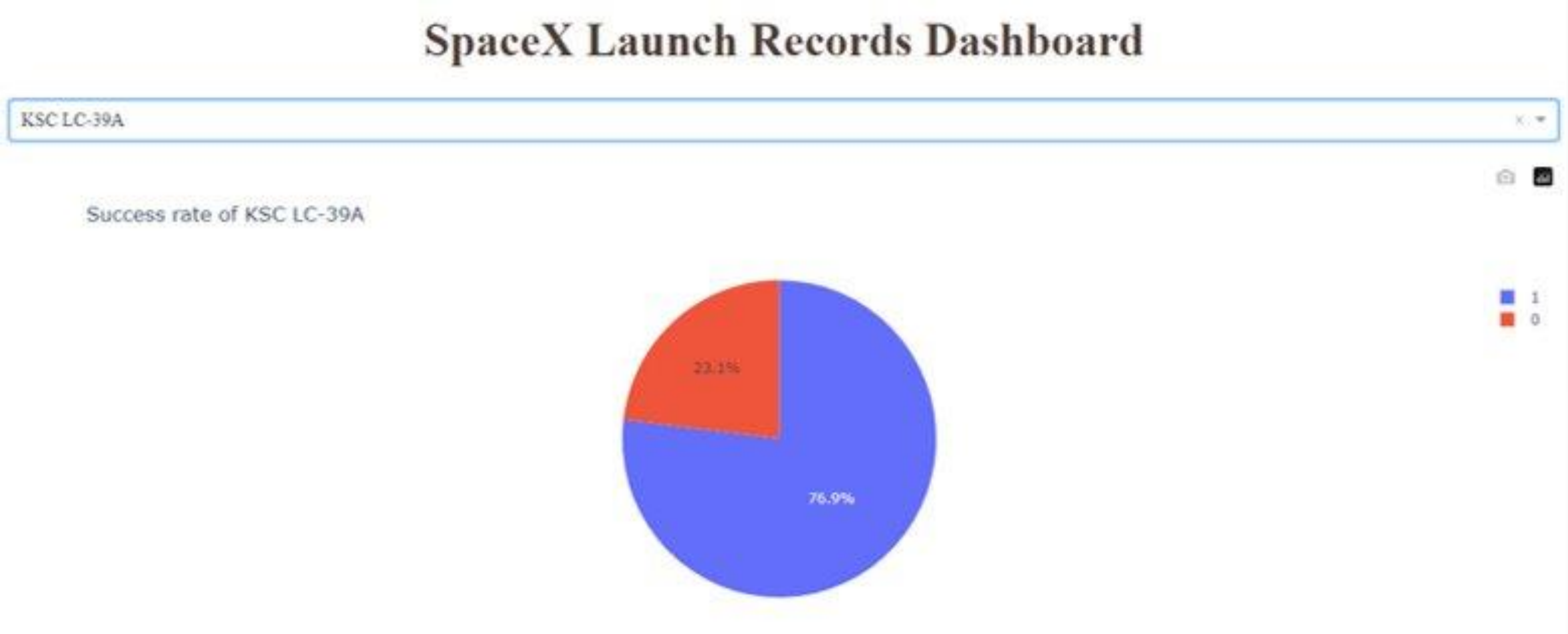
Build a Dashboard with Plotly Dash

Success Rate of All Sites



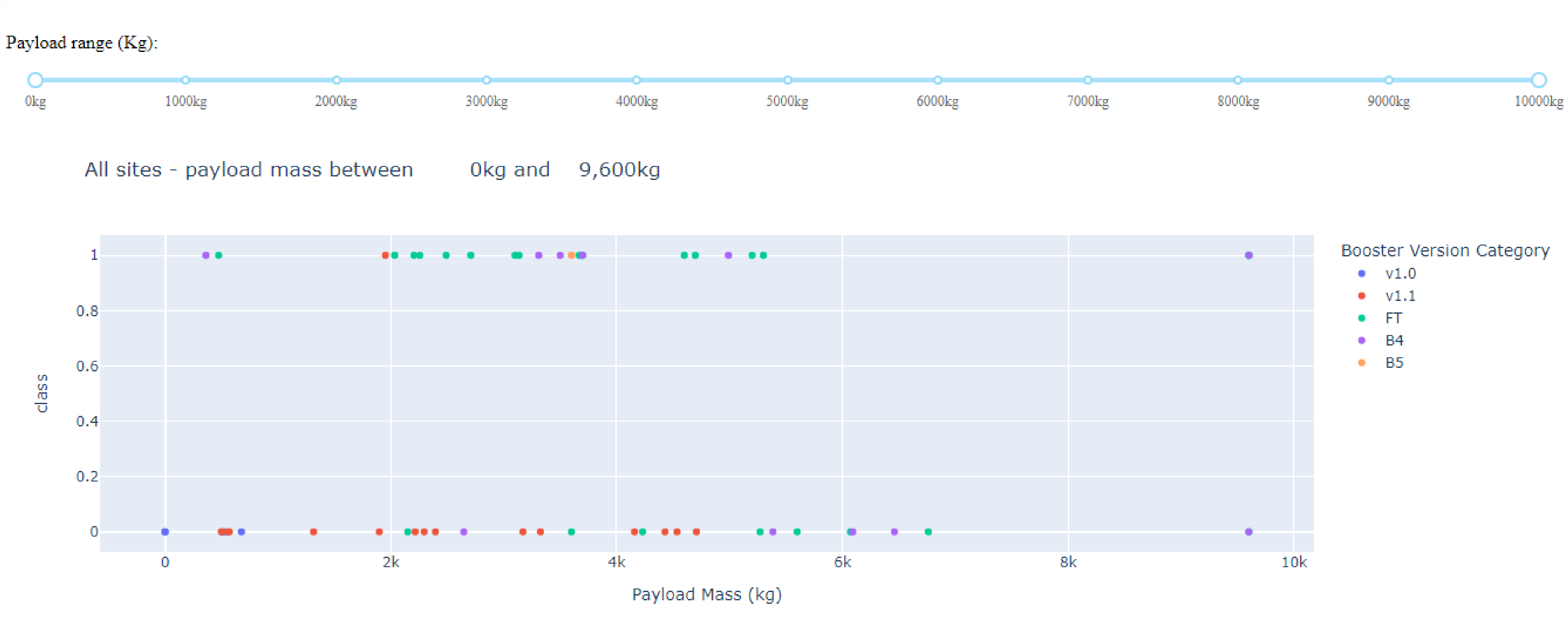
KSC LC-39A and CCAFS LC-40 have almost 71% of all successful landing outcomes combined together.

Most Successful Launch Site

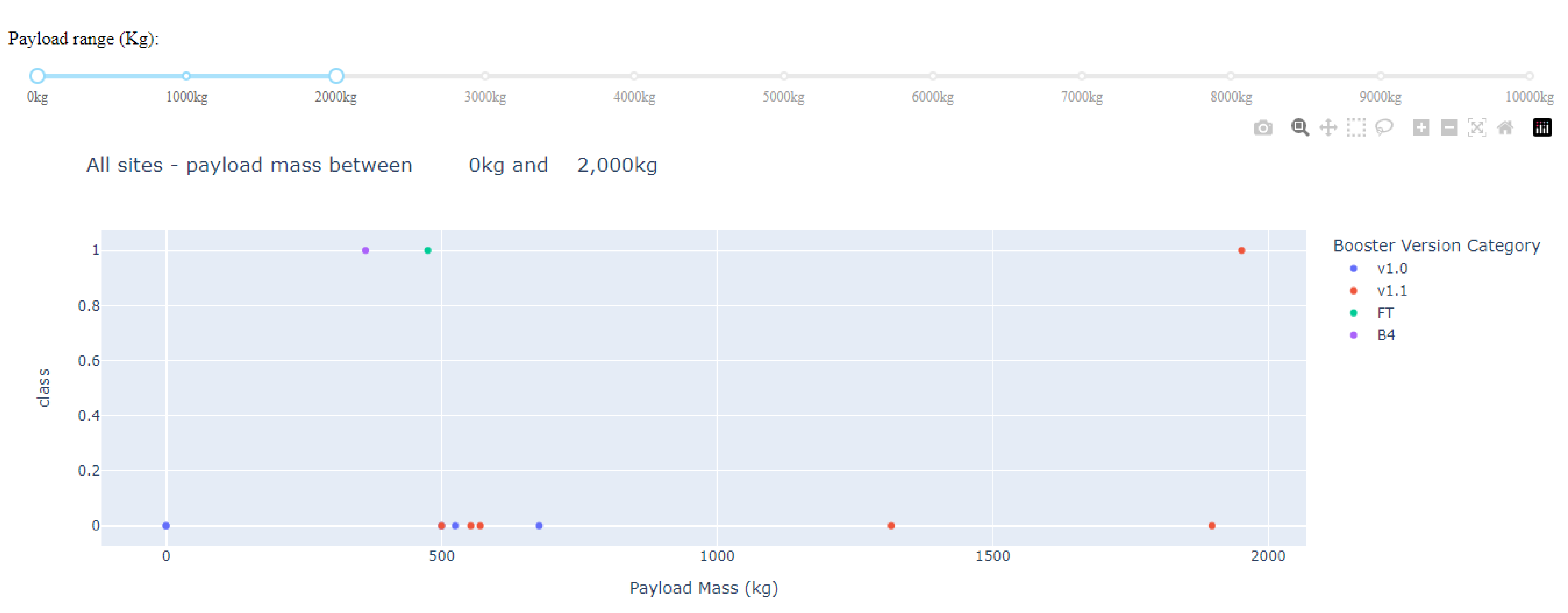


KSC LC 39-A is the most successful in landing outcomes among all the launching sites with success rate of almost 77%

Launch Outcomes Considering Payloads and Booster Versions

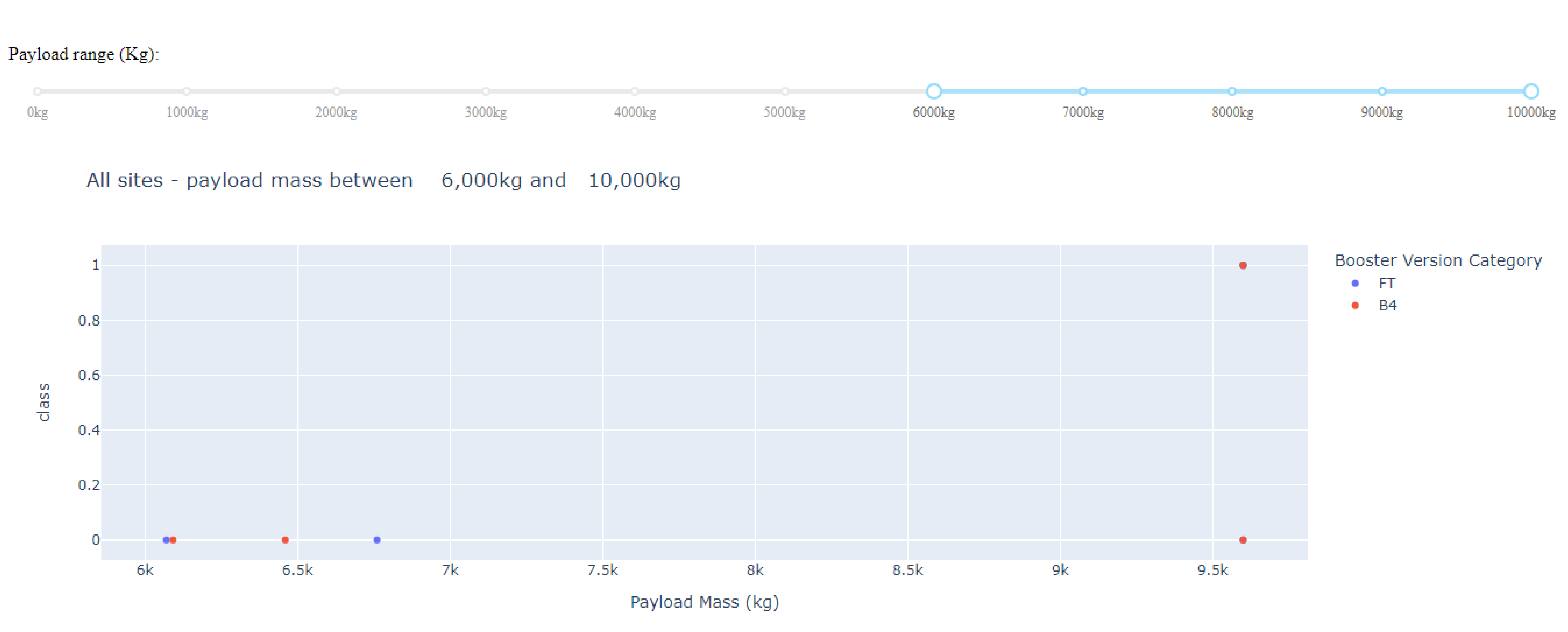


Considering all sites and all payload ranges booster version FT was the most successful



From 0 to 2kg payload range, it is visible that success rate is very low and v1.1 is the most unsuccessful booster version in this range.

When payload mass is between 2000kg and 6000kg success rate is pretty high and FT is the most successful booster version.

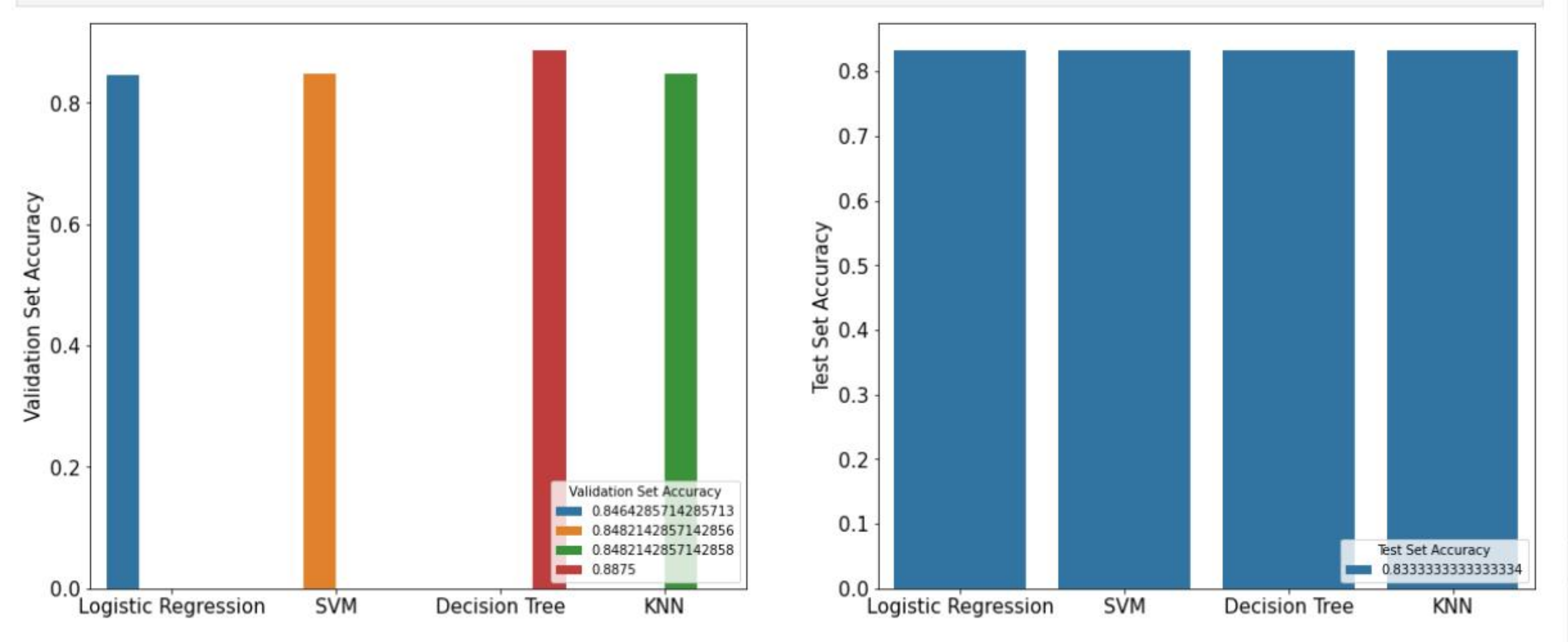


The payload mass range of 6000kg to 10000kg the success rate of landing outcomes by the existing booster versions were the lowest.

Section 5

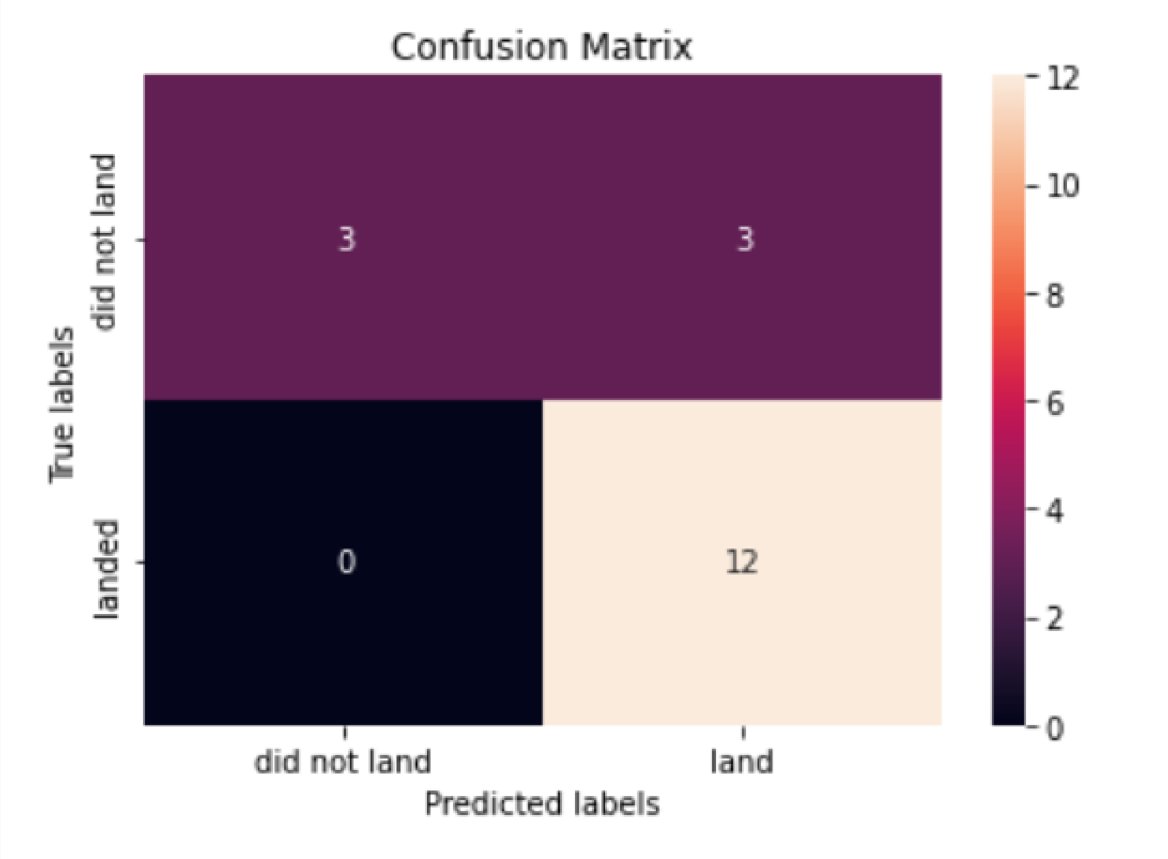
Predictive Analysis (Classification)

Classification Accuracy



We have examined for 4 machine learning algorithms and among them Decision Tree has proven to be the best one with 88.75% validation set accuracy score. All the 4 models had same test set accuracy rate of 83.33%

Confusion Matrix

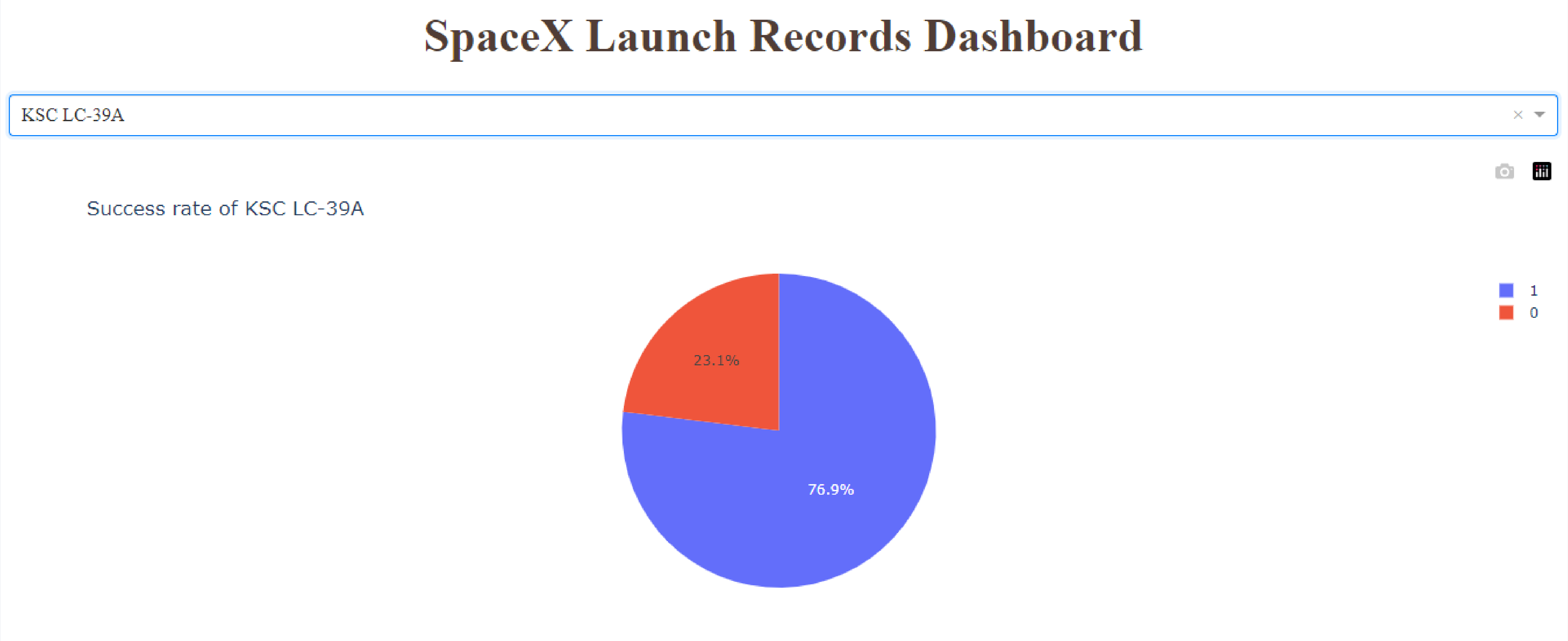


- All of our models showed the same confusion matrix.
- This confusion matrix shows that the model gives 0 True Negatives, and 100% True Positives.
- This confusion matrix shows that model gives 50% False Positives.

Conclusions

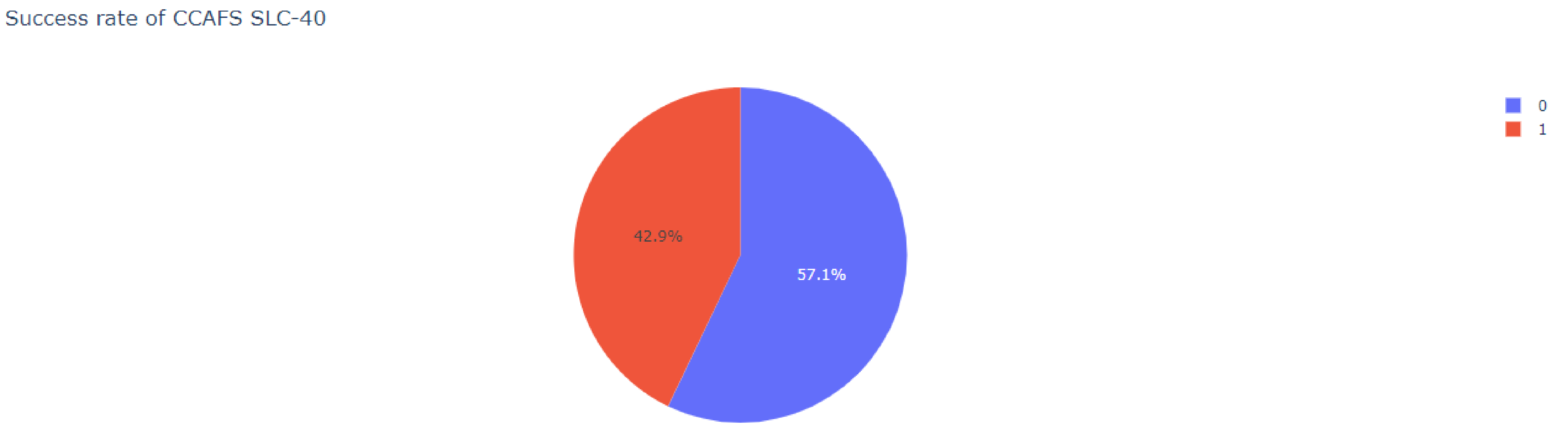
- Our problem was to predict the landing outcome of a rocket launch of SpaceX in order to save money by reusing the first stage which decreases the cost dramatically.
- We have collected data with SpaceX API and from Wikipedia as well and cleaned the dataset.
- Exploratory Data Analysis was done with database and other visualization tools in order to find out insights from the historical data that would benefit to form a more appropriate dataset for model building.
- Four popular classification algorithms were chosen for predictive modeling as our target variable could only have two outcomes, success or failure.
- All models had the test set accuracy score of 83.33% and this can be due to the relatively small dataset we have worked with.
- Decision Tree was the best among them considering validation set accuracy of almost 89%.

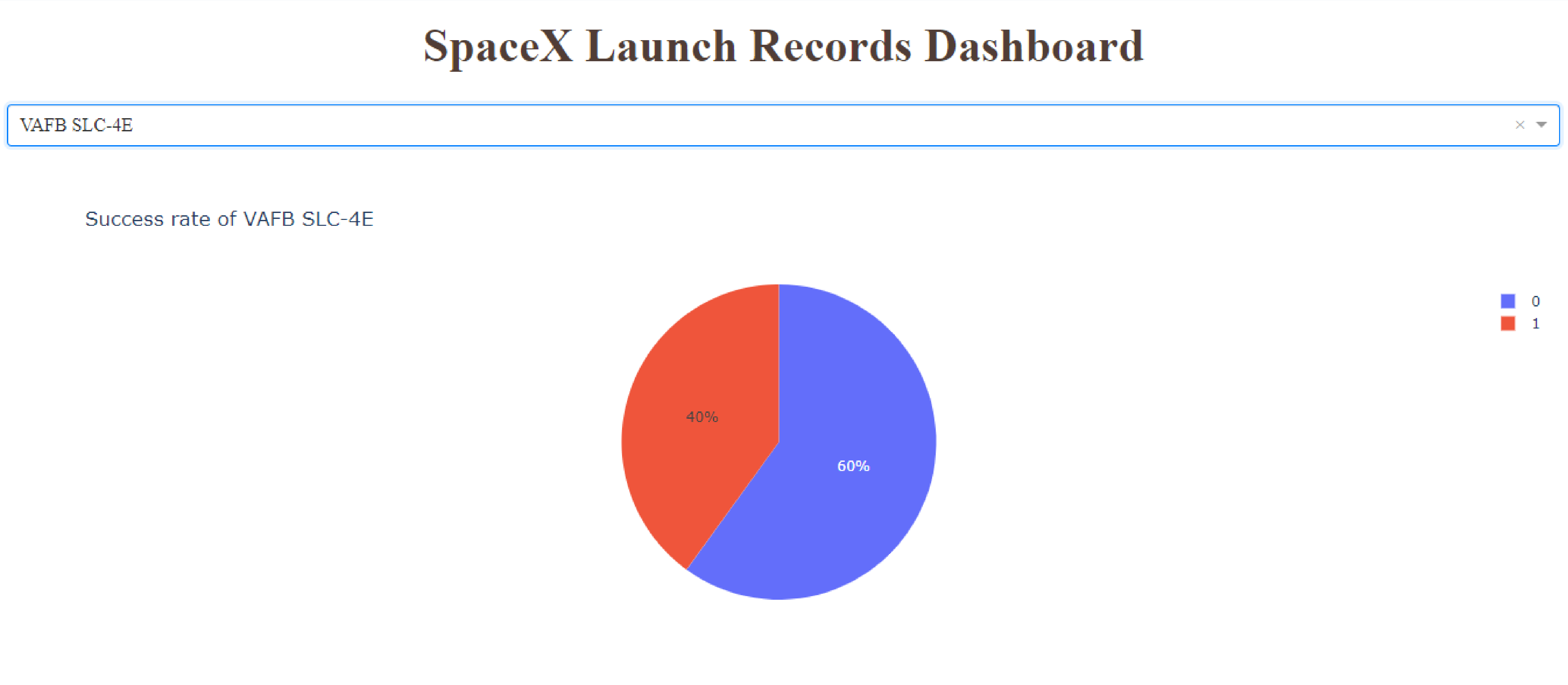
Appendix


SpaceX Launch Records Dashboard
KSC LC-39A
Success rate of KSC LC-39A
23.1%
76.9%
1
0

SpaceX Launch Records Dashboard

CCAFS SLC-40




SpaceX Launch Records Dashboard
VAFB SLC-4E
Success rate of VAFB SLC-4E
40%
60%
0
1

SpaceX Launch Records Dashboard

CCAFS LC-40

Success rate of CCAFS LC-40

Thank you!