IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nathan Nyan
4 July 2025

Outline

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Project background and context

  -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.

- Problems you want to find answers

  - To predict if the Falcon 9 first stage will land successfully

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

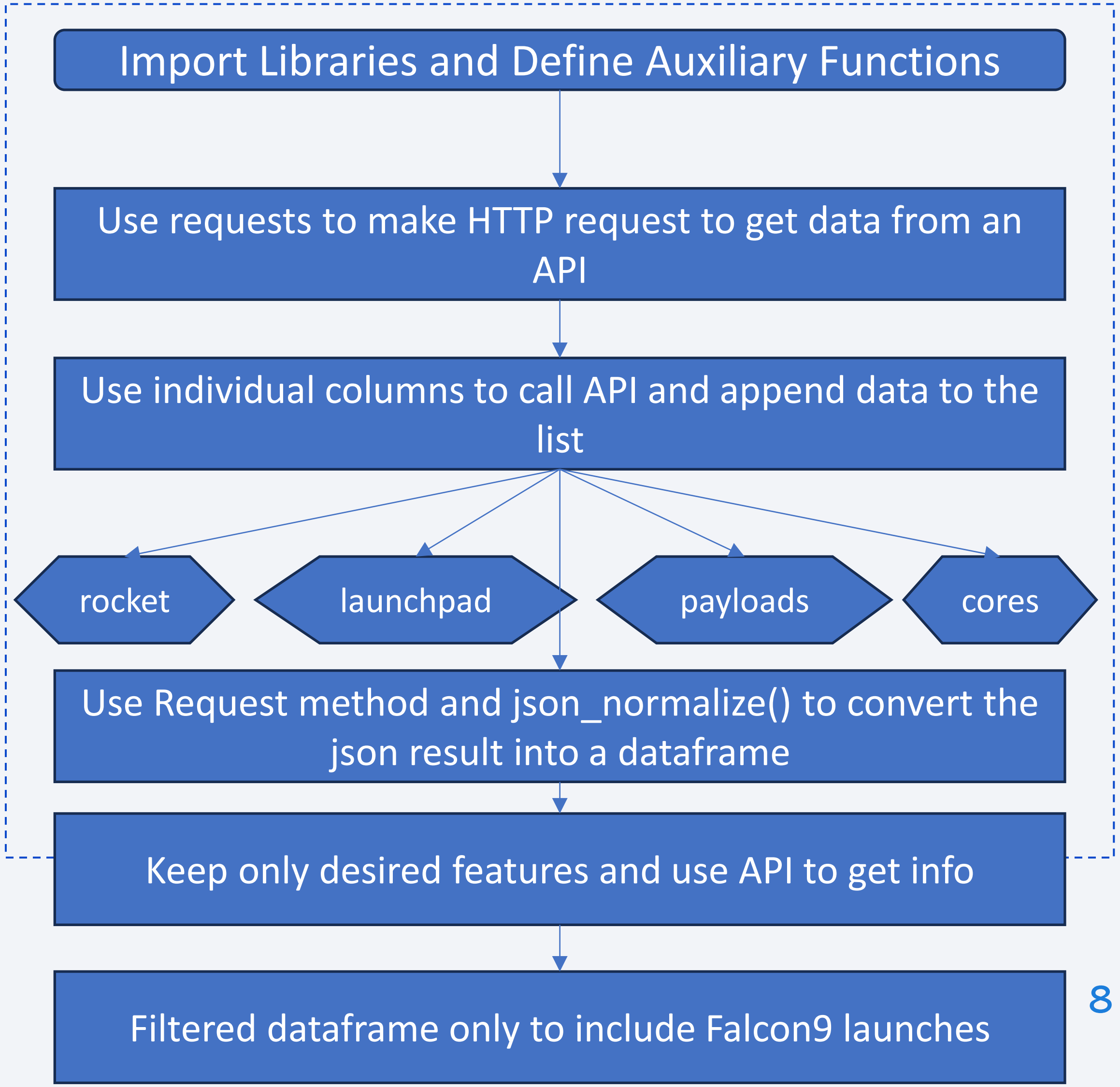
- Dataset are collected using Request and parse the SpaceX launch data using the GET request
- Used Beautiful soup to to response test from Falcon9 launch WIKI page, extract all column/variable names from the HTML table header and create a data frame by parsing the launch HTML tables.

Data Collection – SpaceX API

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

- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose

- https://github.com/nathan-nyan/Applied_Data_Science_Capstone/blob/5b2abc0bbc7000298bc4cffa13aa4e4676cc0ea6/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts
- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose
- https://github.com/nathan-nyan/Applied_Data_Science_Capstone/blob/de21970e816873373f581ae0b23f4c1f469f26b5/jupyter-labs-webscraping.ipynb

Request the Falcon9 Launch Wiki page from its URL

↓

Use BeautifulSoup() to create a BeautifulSoup object from a response text content

↓

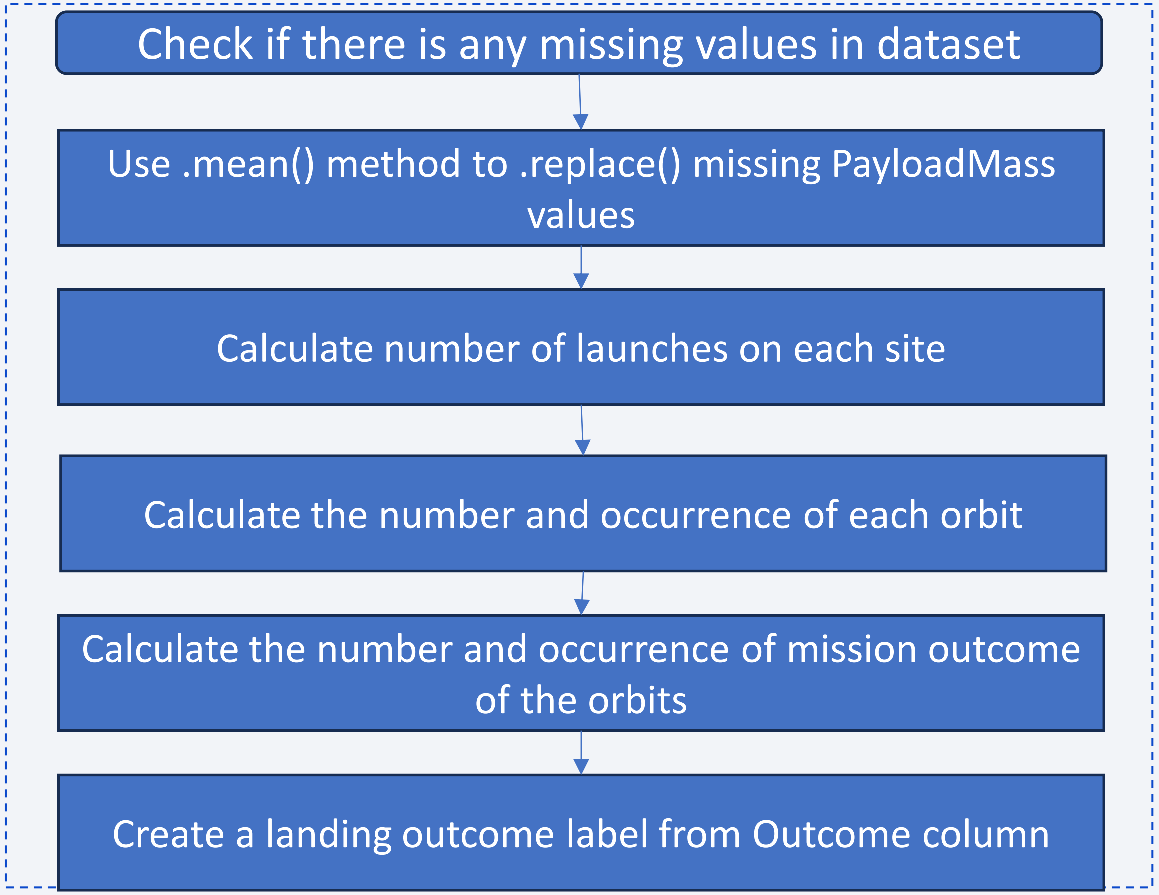
Extract all column/variable names from the HTML table header

↓

Create a data frame by parsing the launch HTML tables

Data Wrangling

- Describe how data were processed
- You need to present your data wrangling process using key phrases and flowcharts
- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose
- https://github.com/nathan-nyan/Applied_Data_Science_Capstone/blob/de21970e816873373f581ae0b23f4c1f469f26b5/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose
- https://github.com/nathan-nyan/Applied_Data_Science_Capstone/blob/bea42843c24df8b293f64beca7bd0f6815c2b5ed/edadataviz%20(1).ipynb

- Used scatter plot to find out relationship between (fight number and launch site), (payload mass and launch site), (flight number and orbit type) and (payload mass and orbit type). **Scatter plot show relationship between two categories.**
- Used bar chart to find out success rate of each orbit type. **Bar chart can visualize categorical data.**
- Line chart to visualize the launch success yearly trend. **Line chart can show continuous data trend.**

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose
- https://github.com/nathan-nyan/Applied_Data_Science_Capstone/blob/f32ac22f2dc528da60505fc096a13d5722b5a77a/jupyter-labs-eda-sql-coursera_sqllite.ipynb

- %sql SELECT DISTINCT "Launch_Site" from SPACEXTABLE
- %sql SELECT * from SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5
- %sql SELECT SUM("PAYLOAD_MASS__KG_") AS total_payload FROM SPACEXTABLE WHERE "Customer" = "NASA (CRS)“
- %sql SELECT AVG("PAYLOAD_MASS__KG_") AS avg_payload FROM SPACEXTABLE WHERE "Booster_Version" = "F9 v1.1“
- %sql SELECT MIN("Date") AS first_succesful_landing_outcome_in_ground_pad , "Landing_Outcome" FROM SPACEXTABLE WHERE "Landing_Outcome" = "Success (ground pad)“
- %sql SELECT "Booster_Version", "PAYLOAD_MASS__KG_" FROM SPACEXTABLE WHERE "Landing_Outcome" = "Success (drone ship)" AND ("PAYLOAD_MASS__KG_" BETWEEN 4000 and 6000)
- %sql SELECT "Mission_Outcome", COUNT(*) FROM SPACEXTABLE GROUP BY "Mission_Outcome“
- %sql SELECT MAX("PAYLOAD_MASS__KG_") AS max_payload , "Booster_Version" FROM SPACEXTABLE GROUP BY "Booster_Version“
- %sql SELECT substr("Date", 6,2) as month , "Date", "Booster_Version", "Landing_Outcome", "Launch_Site" FROM SPACEXTABLE WHERE "Landing_Outcome" = "Failure (drone ship)" AND substr("Date",0,5)='2015'
- %sql SELECT "Landing_Outcome", COUNT(*) as total, "Date" FROM SPACEXTABLE WHERE "Date" BETWEEN "2010-06-04" AND "2017-03-20" GROUP BY "Landing_Outcome" ORDER BY "total" DESC

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Explain why you added those objects
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose
- https://github.com/nathan-nyan/Applied_Data_Science_Capstone/blob/e86d90dd0e973b6ac1059b9026528a52b5a311e2/lab_jupyter_launch_site_location.ipynb

- Markers (launch site and success rates) are added. **This is to mark all lunch sites and labels.**
- Circles (area) are added. **This is to highlight area with a text label on for easy identification.**
- Lines (launch site to coastline, railway, highway and city) are added. **This is to show distance between a launch site to its proximities.**

Build a Dashboard with Plotly Dash

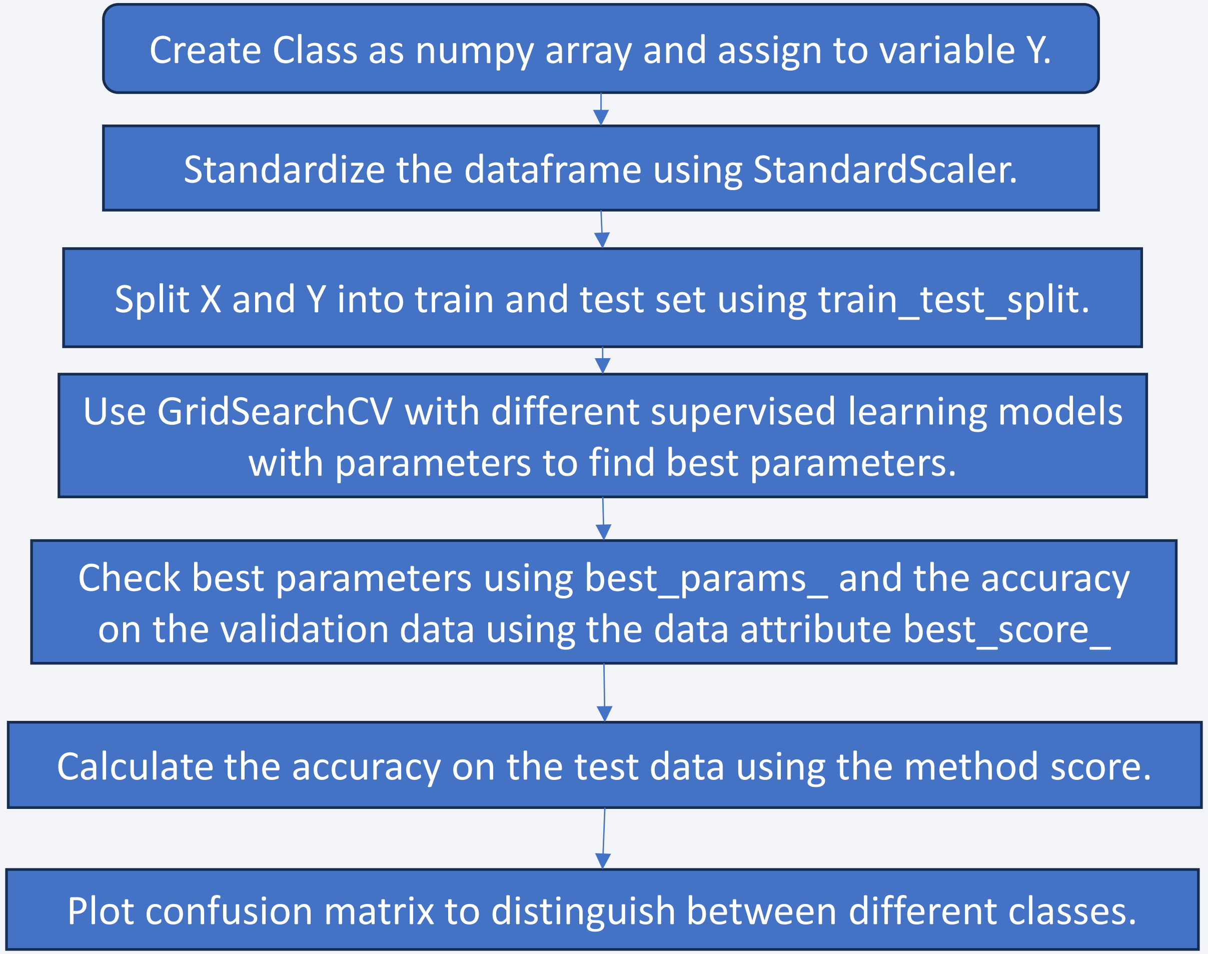
- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose
- https://github.com/nathan-nyan/Applied_Data_Science_Capstone/blob/210521dc50a55f4c9b49f540dfc5c7b6da5ef32e/spacex-dash-app.py

- Pie chart and Scatter chart is added with selection options with drop-down input of "ALL" or "individual launch sites" and range slider to select the Payload.
- **Pie chart can visualize relative proportions of different categories.**
- **Scatter chart can interpret relationship between two categories.**
- **Drop down input component so that user can select based on preference.**
- **Range slider to easy selecting min and max range.**

Micron Confidential

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose
- https://github.com/nathan-nyan/Applied_Data_Science_Capstone/blob/06a758649a1d2e3da0378e7378ca70e4f5bb5593/SpaceX_Machine%20Learning%20Prediction_Part_5%20(1).ipynb



Results

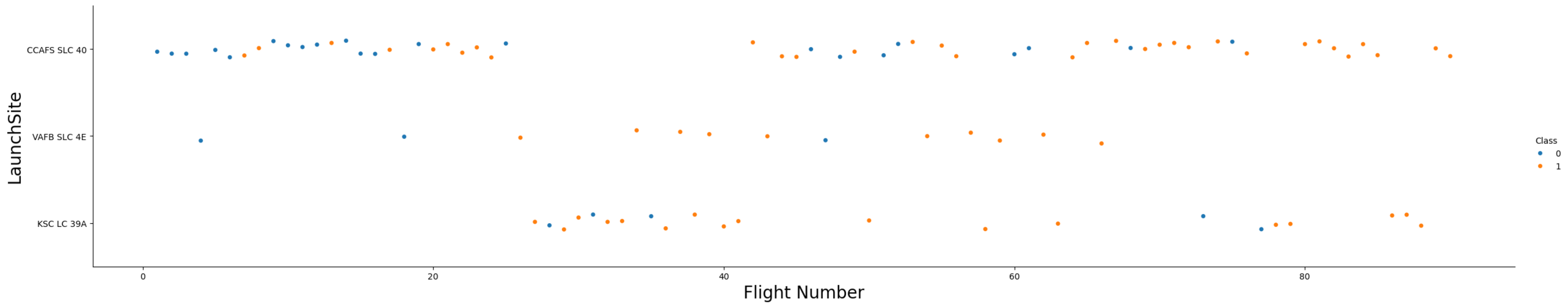
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Micron Confidential

Section 2

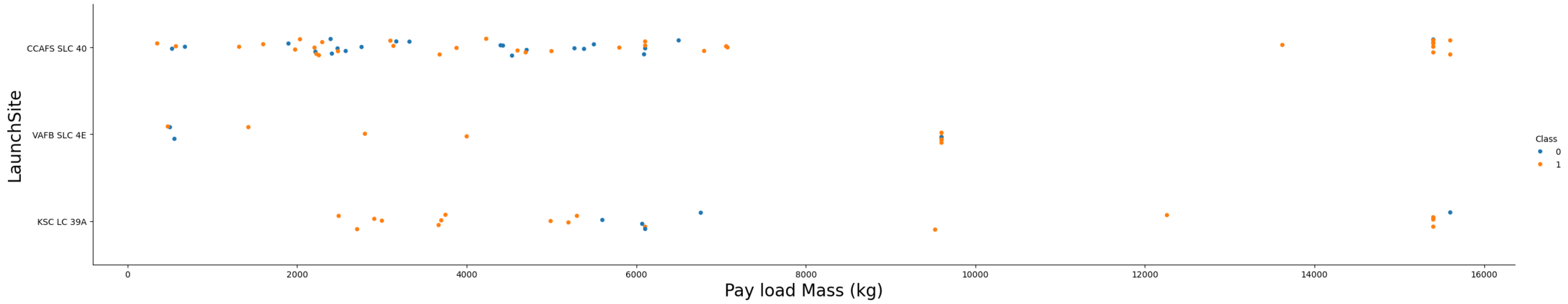
Insights drawn from EDA

Flight Number vs. Launch Site



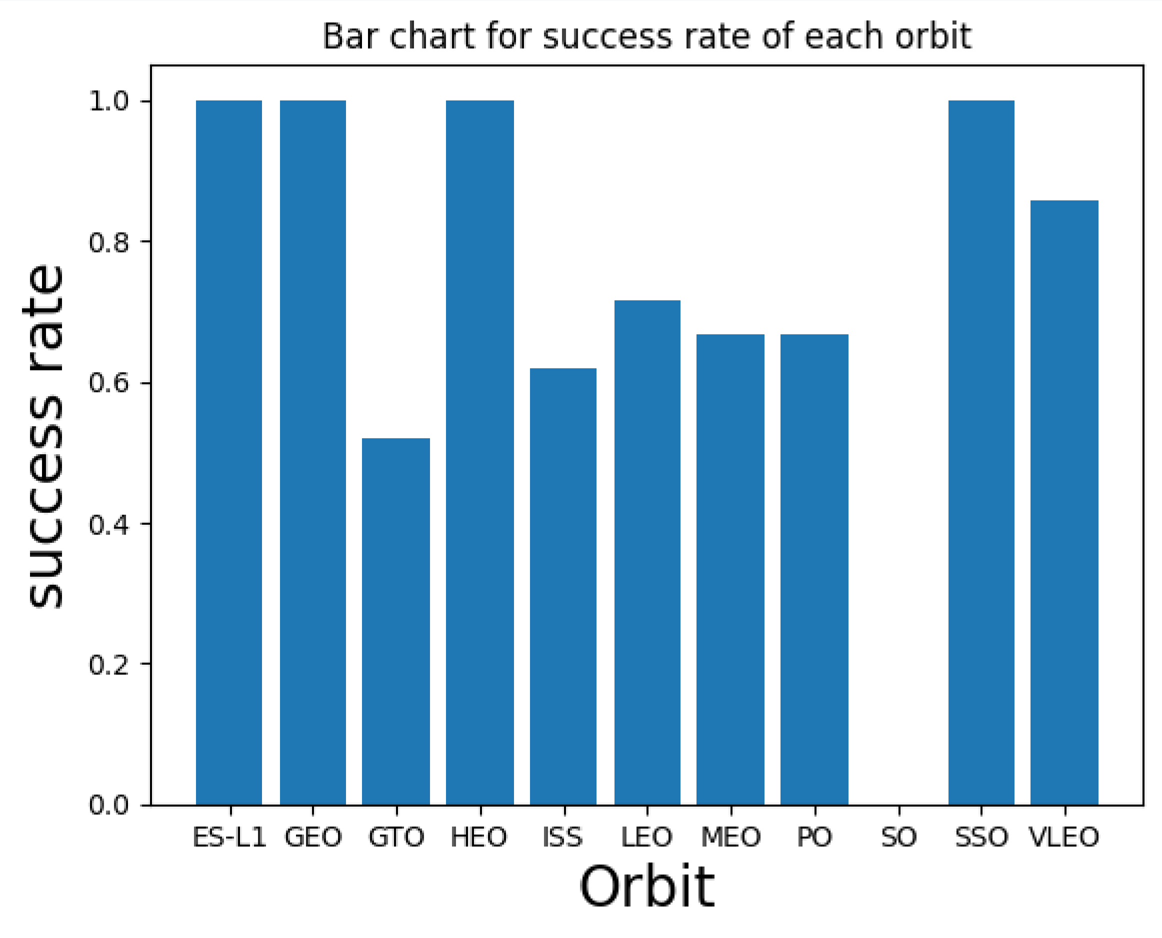
VAFB SLC 4E site has less number of flight but high chance of success , follow by KSC LC 39A site. CCAFS SLC40 has high number of flights and roughly balance mix of class.

Payload vs. Launch Site



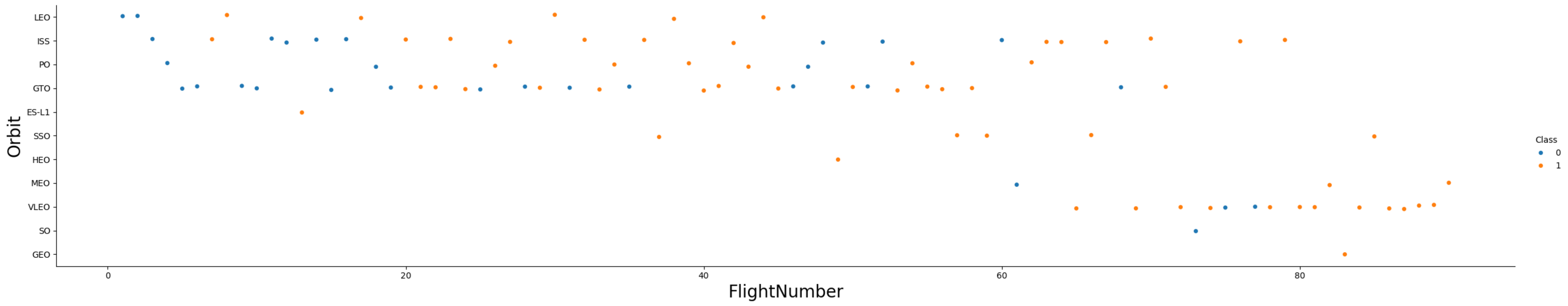
VAFB-SLC launchsite there are no rockets launched for heavypayload mass (greater than 10000)

Success Rate vs. Orbit Type



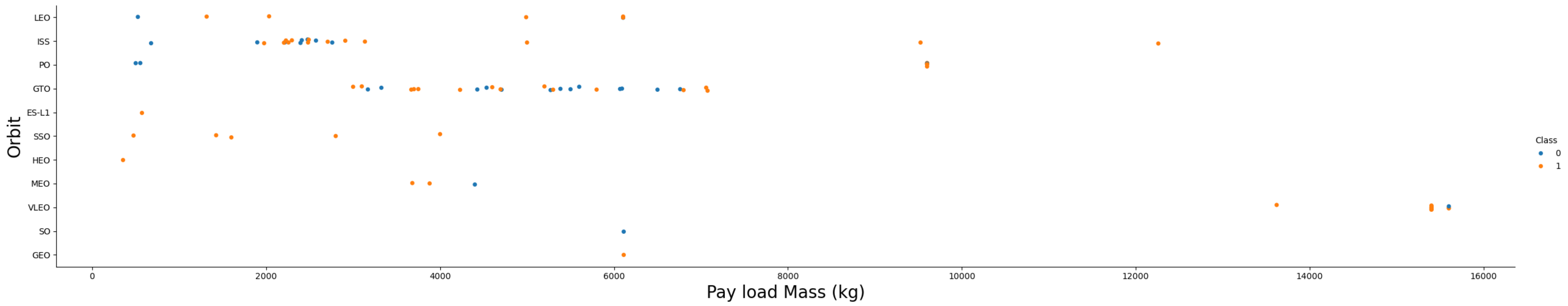
Orbit : ES-LS1, GEO, HEO and SSO has highest success rates.

Flight Number vs. Orbit Type



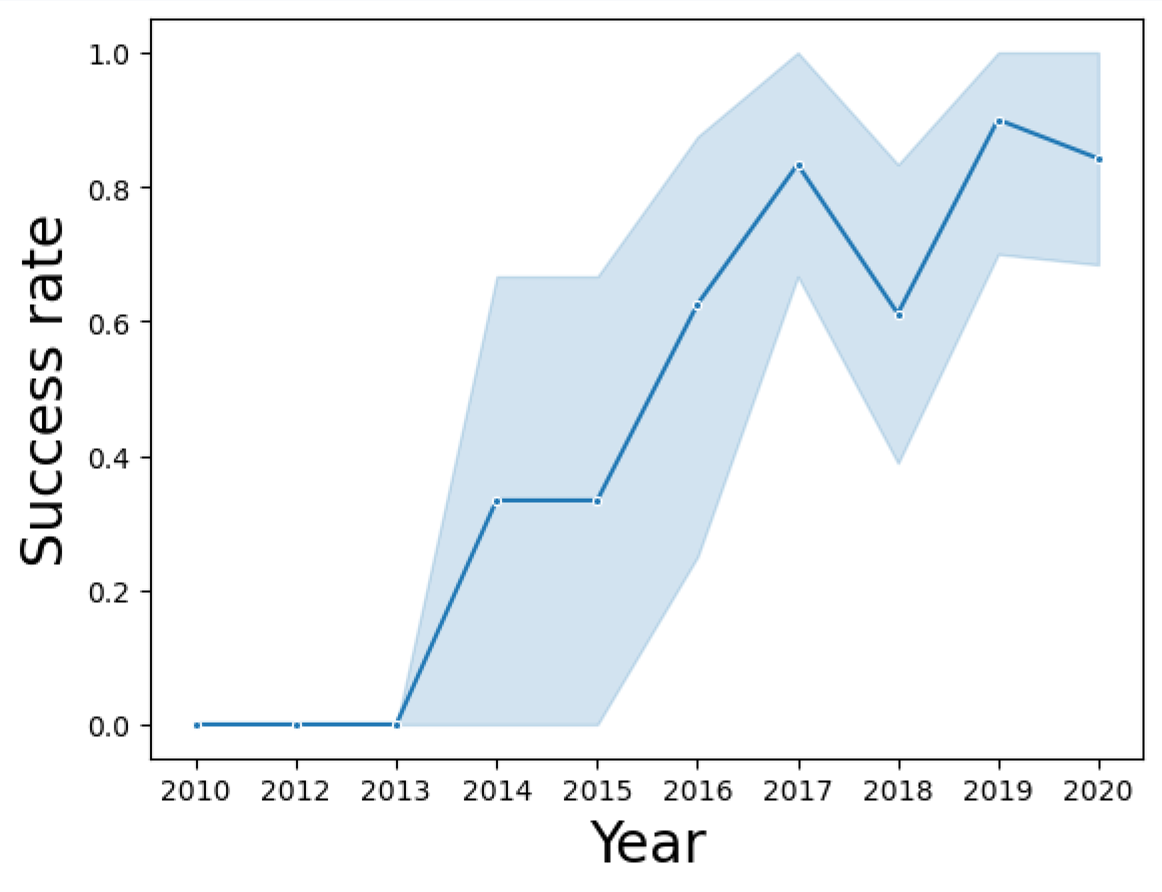
In the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.

Payload vs. Orbit Type



With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.

Launch Success Yearly Trend



The success rate since 2013 kept increasing till 2020

All Launch Site Names

- Find the names of the unique launch sites

```
In [12]: %sql SELECT DISTINCT "Launch_Site" from SPACEXTABLE

 * sqlite:///my_data1.db
Done.
Out[12]:
```

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

- Use DISTINCT in query so that unique values are returned.

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

```
In [14]: %sql SELECT * from SPACEXTABLE WHERE "Launch_Site" LIKE 'CCA%'  LIMIT 5
```

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

Out[14]:

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

- Used Where clause on column name “Launch_site” and use LIKE for related keywords with Limit for desired number of rows to query.

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

```
In [16]: %sql SELECT SUM("PAYLOAD_MASS__KG_") AS total_payload FROM SPACEXTABLE WHERE "Customer" = "NASA (CRS)"

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

Out[16]:

total_payload
45596

- Use SUM() and WHERE clause for “NASA (CRS)”.

Average Payload Mass by F9 v1.1

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

```
In [17]: %sql SELECT AVG("PAYLOAD_MASS__KG_") AS avg_payload FROM SPACEXTABLE WHERE "Booster_Version" = "F9 v1.1"

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

Out[17]:

avg_payload
2928.4

- Use AVG() and WHERE clause for “F9 v1.1”.

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

```
In [22]: %sql SELECT MIN("Date") AS first_succesful_landing_outcome_in_ground_pad , "Landing_Outcome" FROM SPACEXTABLE WHERE "Landing
```

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

Out[22]:

first_succesful_landing_outcome_in_ground_pad	Landing_Outcome
2015-12-22	Success (ground pad)

- Use MIN("Date") to get first data and WHERE clause for "Success (ground Pad)".

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
In [21]: %sql SELECT "Booster_Version", "PAYLOAD_MASS__KG_"  FROM SPACEXTABLE WHERE "Landing_Outcome" = "Success (drone ship)" AND ('

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

Out[21]:

Booster_Version	PAYLOAD_MASS__KG_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

- Select "Booster_Version", "PAYLOAD_MASS__KG_" and use WHERE clause on "Landing_Outcome" = "Success (drone ship)" AND ("PAYLOAD_MASS__KG_" BETWEEN 4000 and 6000) for desired payload range.

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

```
In [24]: %sql SELECT "Mission_Outcome", COUNT(*) FROM SPACEXTABLE GROUP BY "Mission_Outcome"

 * sqlite:///my_data1.db
Done.
Out[24]:
```

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

- Used GROUP BY to group Mission_outcome and COUNT(*) to get total number of counts on each group.

Boosters Carried Maximum Payload

```
In [33]: %sql SELECT MAX("PAYLOAD_MASS__KG_") AS max_payload , "Booster_Version" FROM SPACEXTABLE GROUP BY "Booster_Version"
```

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

Out[33]:

max_payload	Booster_Version
2647	F9 B4 B1039.2
5384	F9 B4 B1040.2
9600	F9 B4 B1041.2
6460	F9 B4 B1043.2
3310	F9 B4 B1039.1
4990	F9 B4 B1040.1
9600	F9 B4 B1041.1
3500	F9 B4 B1042.1
5000	F9 B4 B1043.1
6092	F9 B4 B1044
362	F9 B4 B1045.1
2697	F9 B4 B1045.2
3600	F9 B5 B1046.1
5800	F9 B5 B1046.2
4000	F9 B5 B1046.3
12050	F9 B5 B1046.4
5300	F9 B5 B1047.2
6500	F9 B5 B1047.3
3000	F9 B5 B1048.2
4850	F9 B5 B1048.3
15600	F9 B5 B1048.4
15600	F9 B5 B1048.5
9600	F9 B5 B1049.2
13620	F9 B5 B1049.3

Only F9 V1.1 has 5 times with different payload.

PAYLOAD_MASS__KG_	Booster_Version
3170	F9 v1.1
3325	F9 v1.1
2296	F9 v1.1
1316	F9 v1.1
4535	F9 v1.1

Use GROUP BY to group each booster_version and MAX() to get maximum payload of each boosters.

525	F9 v1.0 B0005
500	F9 v1.0 B0006
677	F9 v1.0 B0007
4535	F9 v1.1
500	F9 v1.1 B1003
2216	F9 v1.1 B1010
4428	F9 v1.1 B1011
2395	F9 v1.1 B1012

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
In [53]: %sql SELECT substr("Date", 6,2) as month , "Date", "Booster_Version", "Landing_Outcome", "Launch_Site" FROM SPACEXTABLE WHEI

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

Out[53]:

month	Date	Booster_Version	Landing_Outcome	Launch_Site
01	2015-01-10	F9 v1.1 B1012	Failure (drone ship)	CCAFS LC-40
04	2015-04-14	F9 v1.1 B1015	Failure (drone ship)	CCAFS LC-40

- Use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year, WHERE "Landing_Outcome" = "Failure (drone ship)" AND substr("Date",0,5)='2015', for 2015 and fail records.

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

```
In [59]: %sql SELECT "Landing_Outcome", COUNT(*) as total, "Date" FROM SPACEXTABLE WHERE "Date" BETWEEN "2010-06-04" AND "2017-03-20'

 * sqlite:///my_data1.db
Done.
Out[59]:
```

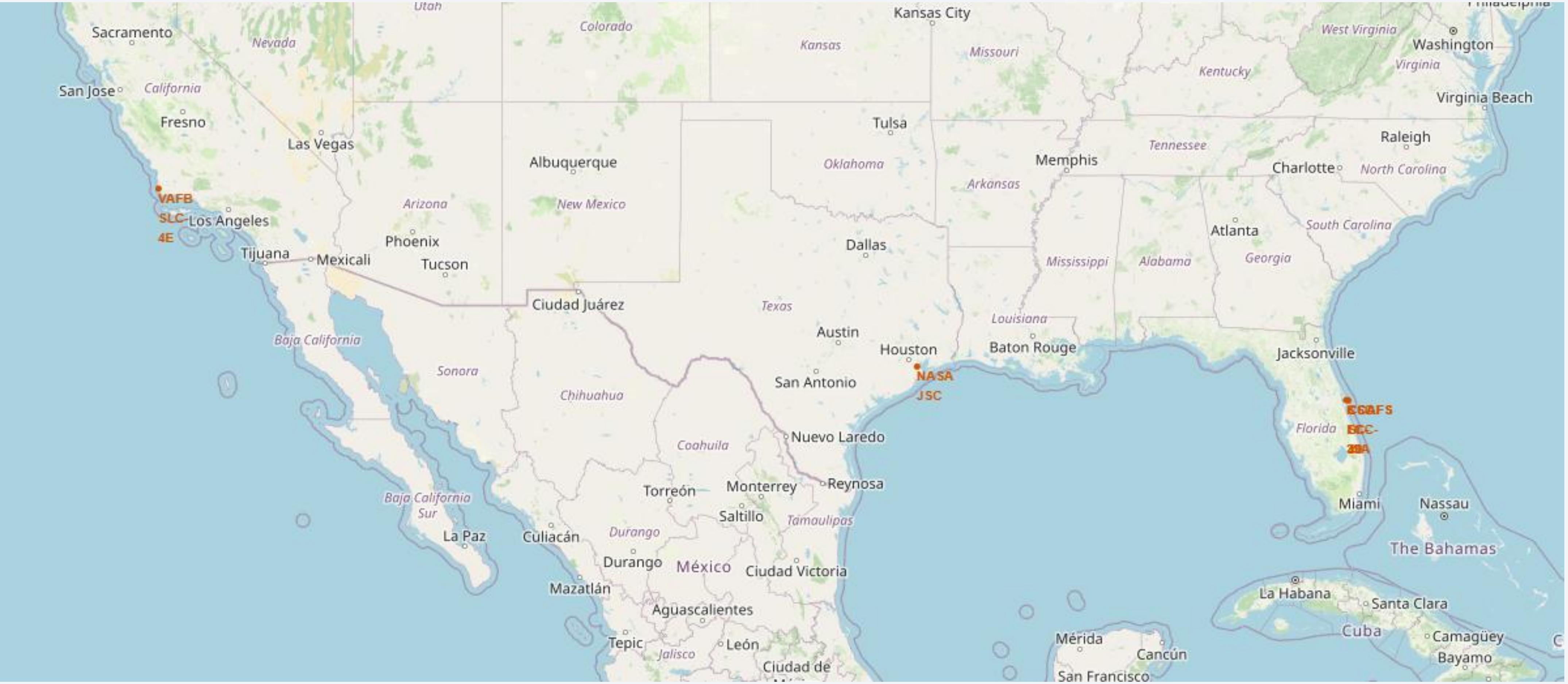
Landing_Outcome	total	Date
No attempt	10	2012-05-22
Success (drone ship)	5	2016-04-08
Failure (drone ship)	5	2015-01-10
Success (ground pad)	3	2015-12-22
Controlled (ocean)	3	2014-04-18
Uncontrolled (ocean)	2	2013-09-29
Failure (parachute)	2	2010-06-04
Precluded (drone ship)	1	2015-06-28

- Use COUNT(*) to get total on "Landing Outcome" group by using GROUP BY, Where condition for selected date range and ORDER by count and DESC for descending order.

Section 3

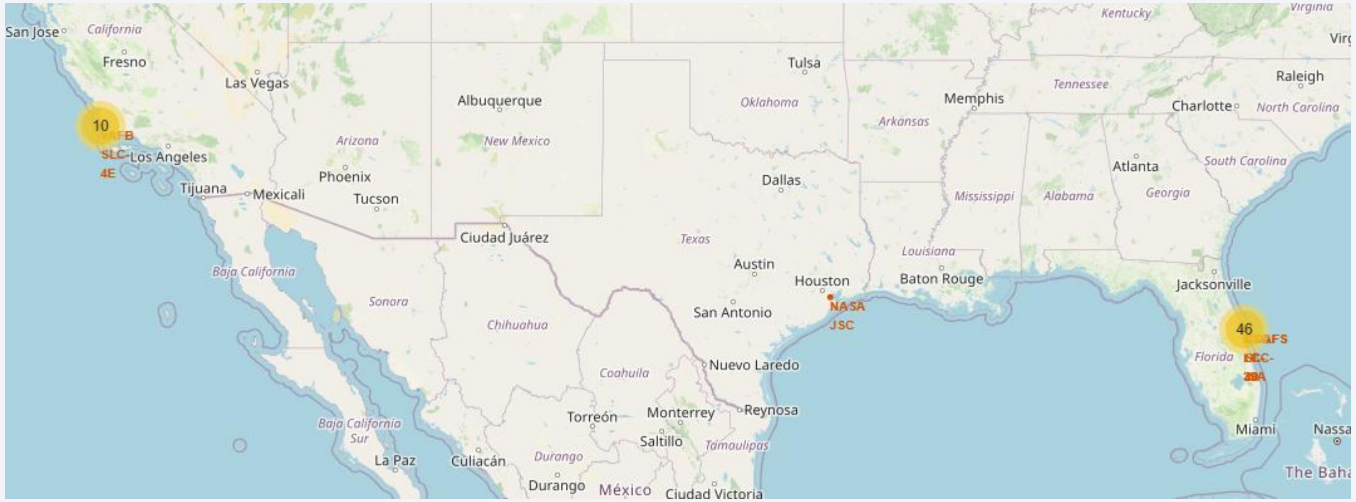
Launch Sites Proximities Analysis

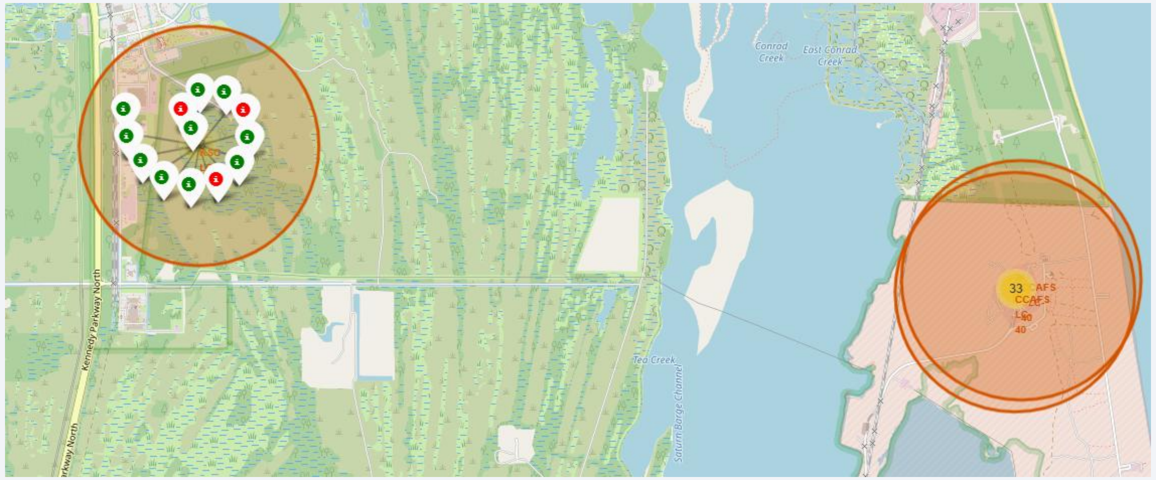
All Marked Launch Sites



Marker and Circle are added on each Launch Site for easy identification.

All Launch Sites with launch records

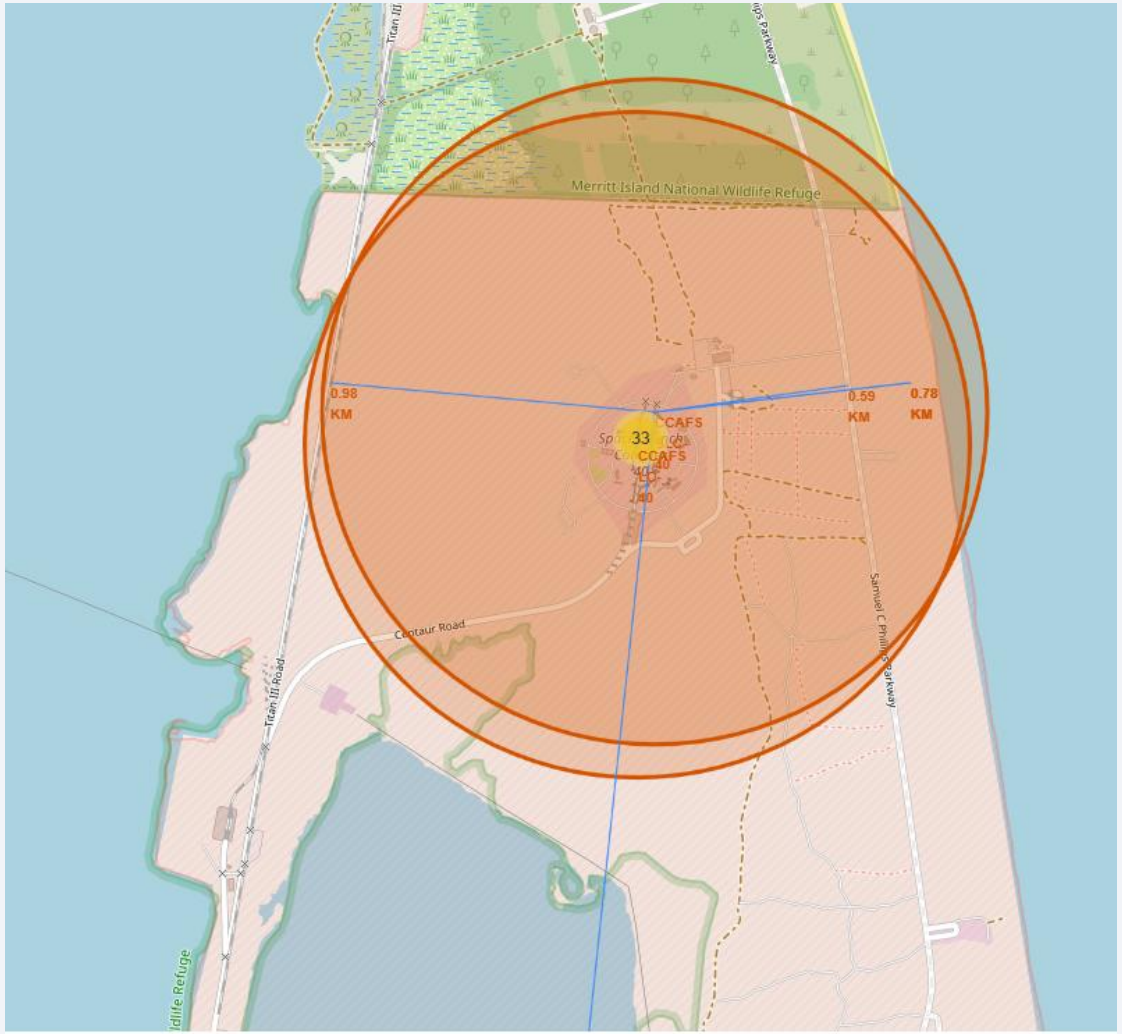




Different markers with different colors are used for success and fail launch.

Used MarkerCluster to simply a map containing many markers having the same coordinate.

Launch site to its proximities



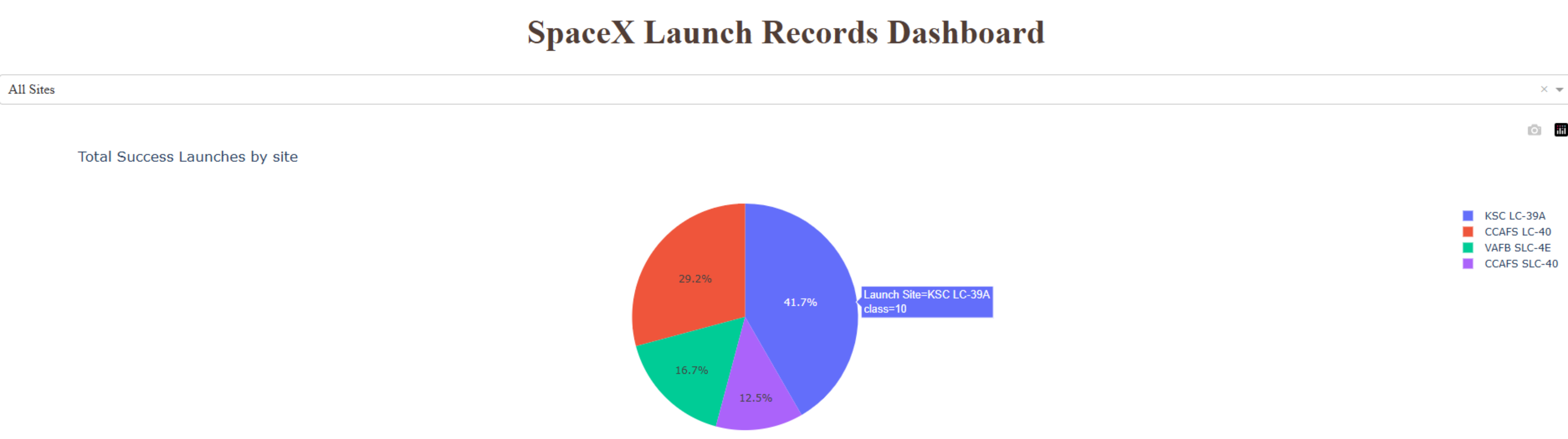
PolyLine is added to show the distance between Launch site and nearby proximities.

Added marker at target proximity location to show total distance in KM from launch site.

Section 4

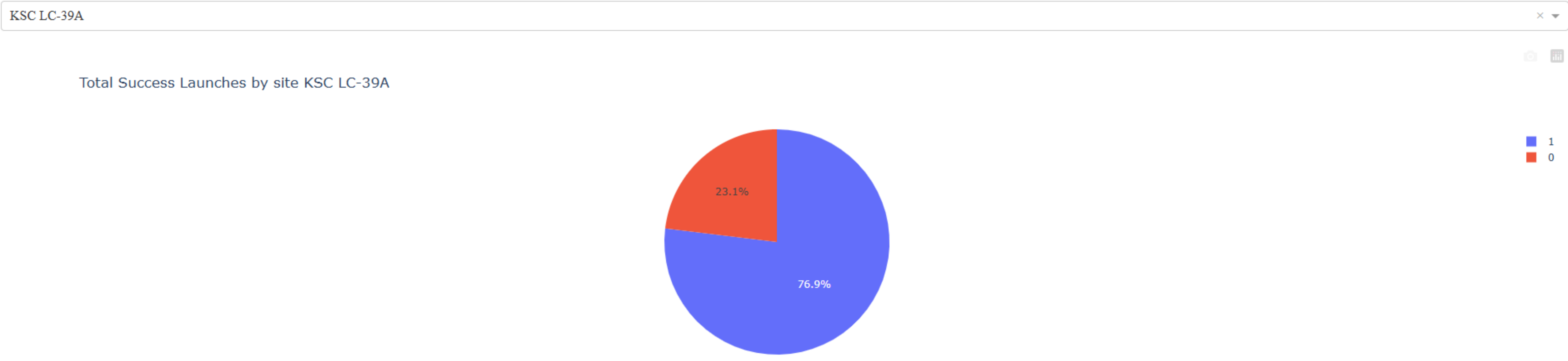
Build a Dashboard with Plotly Dash

Total Success Launches by All Sites



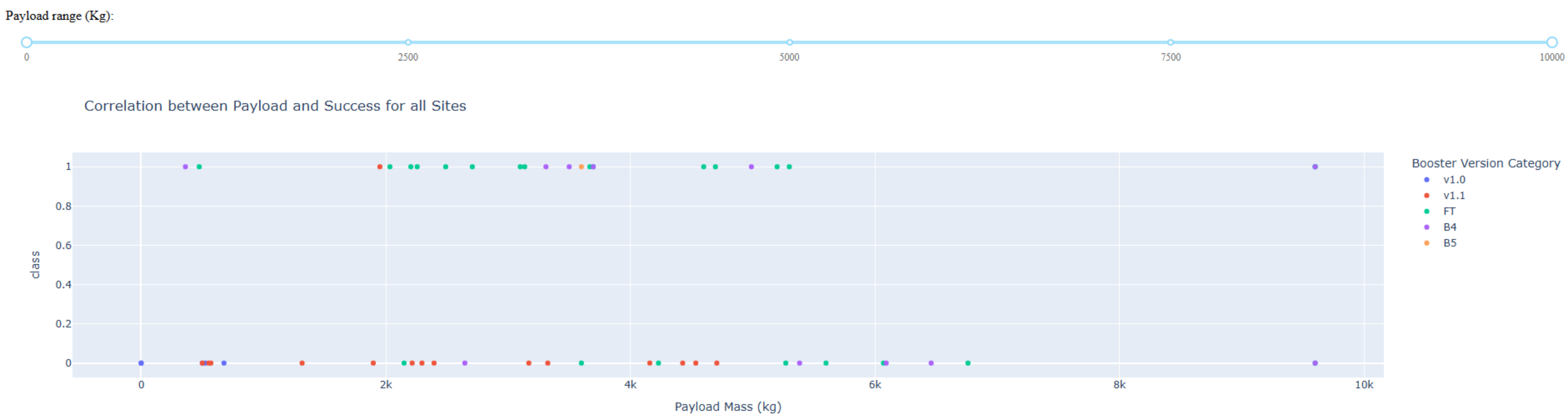
Drop-down component is used for user input where All sites selection as default.
KSC LC-39A has highest success rate with 10 successful launches.

KSC LC-39A Total Success Launches



Drop-down selection is selected as KSC LC-39A.
The pie chart has changed accordingly and showing 77% of successful launches rate from KSC LC-39A site.

Payload and success rate correlation for all sites



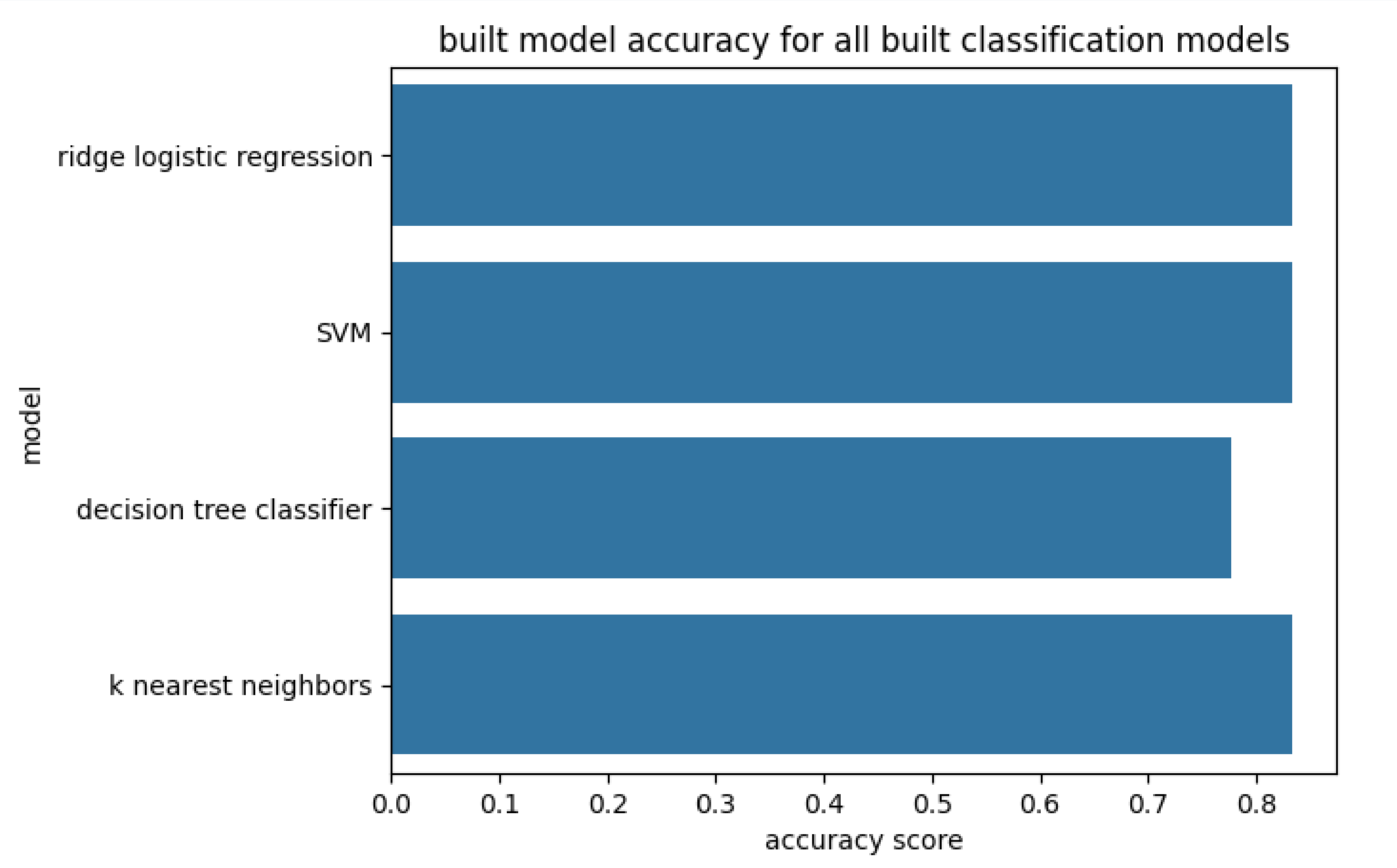
Payload range slider is used to select desired range for payload to check.
Payload range from 0 up to 6000kg showed largest success rate and mostly from FT booster version.

Section 5

Predictive Analysis (Classification)

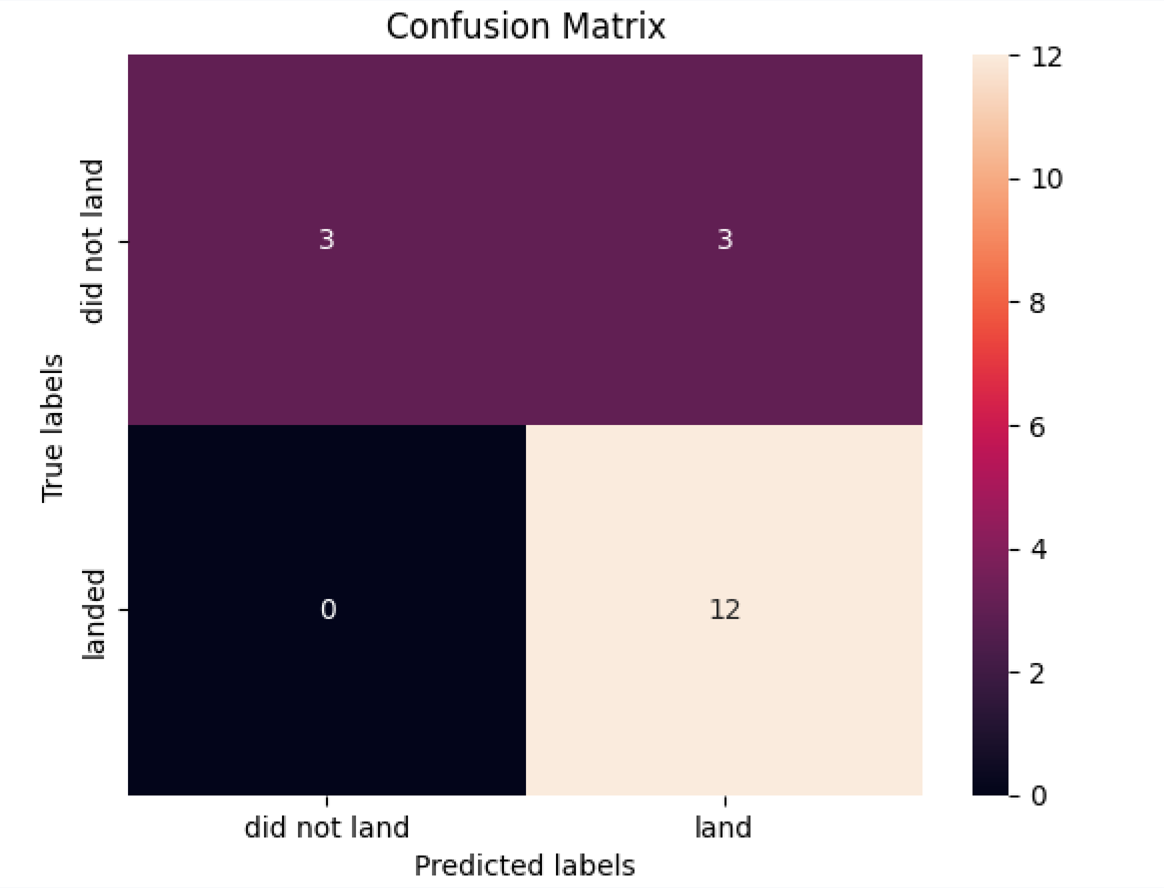
Micron Confidential

Classification Accuracy



Ridge logistic regression, SVM and K nearest neighbors model has best model accuracy.

Confusion Matrix



Ridge logistic regression can distinguish between the different classes where:

True Postive - 12 (True label is landed, Predicted label is also landed)

False Postive - 3 (True label is not landed, Predicted label is landed)

Conclusions

- Best parameter for Ridge logistic regression model is {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'} with accuracy : 0.8464285714285713
- Best parameter for SVM model is {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'} with accuracy : 0.8482142857142856
- Best parameter for decision tree model is {'criterion': 'entropy', 'max_depth': 16, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitter': 'random'} with accuracy : 0.9017857142857142
- Best parameter for K nearest neighbor model is {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1} with accuracy : 0.8482142857142858

Even though decision tree model has highest accuracy in train model however it has lower accuracy on test set. While deciding model, confusion matrix should be used to evaluate the performance of a classification model, especially in supervised learning.

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!

Micron Confidential