Winning Space Race with Data Science

Yared Getachew
13 November 2021

Outline

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

Executive Summary

- Summary of methodologies
  - Collecting data from various sources (public SpaceX API and SpaceX Wikipedia page)
  - Improve data quality by performing data wrangling
  - Exploring the processed data with SQL query
  - Applying some basic statistical analysis and data visualization
  - Splitting the data into groups defined by categorical variables or factors
  - Build, evaluate, and refine predictive models
- Summary of all results
  - Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K Nearest Neighbors machine learning models were used and produced near identical results with highest accuracy rate of about 83.33% using Support Vector Machine when applied to the test data.

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.
- Problem
  - Determine if the Space X Falcon 9 rocket first stage will land by using different variables to be able to determine the total cost of a launch.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Raw data was collected from various sources including by requesting to the SpaceX API
- Perform data wrangling
  - Perform exploratory data analysis and determine training labels
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build, tune, evaluate classification models

Data Collection

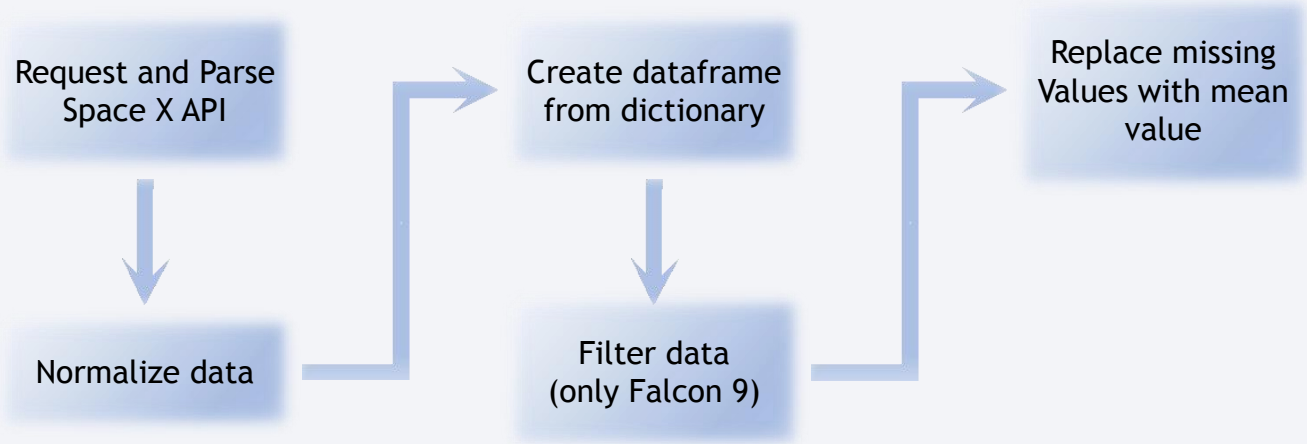
- Request and parse the SpaceX launch data using the GET request
- Normalize data and prepare data by removing irrelevant data
- Filter the dataframe to only include Falcon 9 launches
- Deal with missing values by replacing with mean values

Data Collection – SpaceX API

Data was collected from various sources such as Space X public API requests and web scraping from a table in Space X's Wikipedia page

Space X API Data Columns:
FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude,Latitude

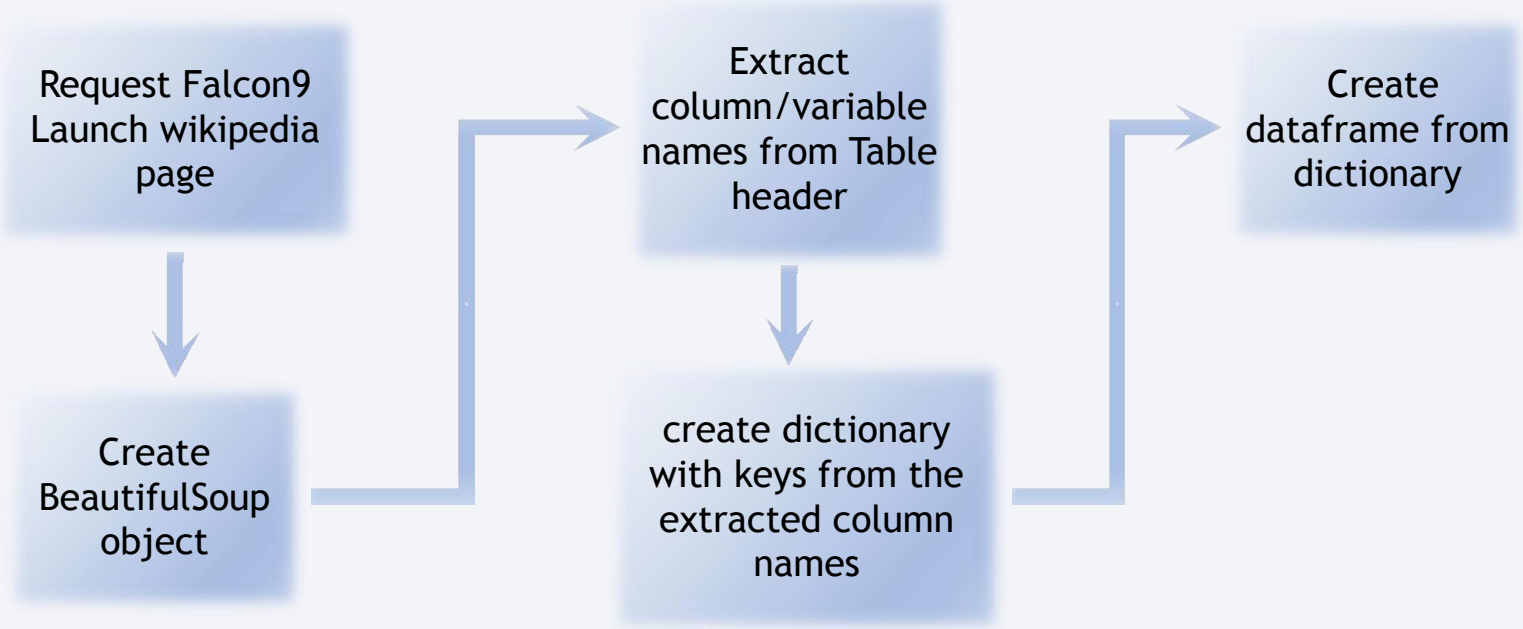
Wikipedia Data Columns:
Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time



GitHub URL:

https://github.com/yaredsha/datascience-ibm/blob/main/data%20collection%20api.ipynb

Data Collection - Scraping



GitHub URL:

https://github.com/yaredsha/datascience-ibm/blob/main/webscraping.ipynb

Data Wrangling

- Load Space X dataset
- Calculate the number of launches on each site
- Calculate the number and occurrence of each orbit
- Calculate the number and occurence of mission outcome per orbit type
  - determine the number of landing outcomes
- Create a landing outcome label from Outcome column
  - landing class = 0 if bad_outcome, landing class = 1 otherwise
- Mapping:
  - True ASDS, True RTLS, & True Ocean – set to -> 1
  - None None, False ASDS, None ASDS, False Ocean, False RTLS – set to -> 0

GitHub URL:

https://github.com/yaredsha/datascience-ibm/blob/main/Data%20wrangling.ipynb

EDA with Data Visualization

- Scatter plots, line charts, and bar plots were used to determine if there exists relationships between different variables and whether the variables should be used in training the models
- Plotted variables:
  - Flight Number vs. Payload Mass, Flight Number vs. Launch Site, Payload Mass vs. Launch Site,
  - Orbit vs. Success Rate, Flight Number vs. Orbit, Payload vs Orbit, and Success Yearly Trend

GitHub URL:

https://github.com/yaredsha/datascience-ibm/blob/main/eda%20dataviz.ipynb

EDA with SQL

- Display the names of the unique launch sites in the space mission
- Display 5 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
- List the date when the first successful landing outcome in ground pad was acheived.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- 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

GitHub URL:

https://github.com/yaredsha/datascience-ibm/blob/main/eda%20sql.ipynb

Build an Interactive Map with Folium

- Map objects such as markers, circles were used
  - To mark all launch sites on a map
  - To mark the success/failed launches for each site on a map
- Map objects such as lines were used
  - To calculate the distances and draw lines between a launch site to its proximities such as Railway, Highway, Coast, and City
- MousePosition map object was used
  - To get coordinate for a mouse over a point on the map

GitHub URL:

https://github.com/yaredsha/datascience-ibm/blob/main/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

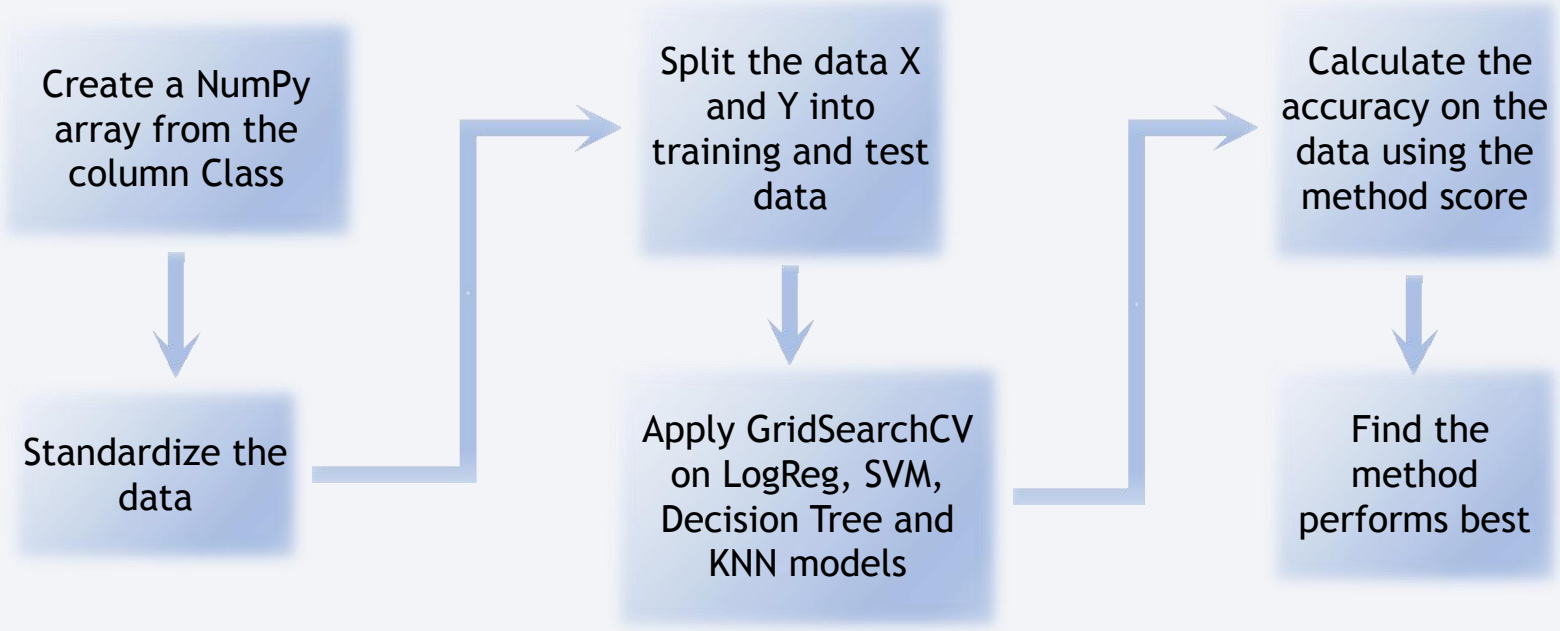
Build a Dashboard with Plotly Dash

- Pie chart
  - show distribution of successful landings across all launch sites
  - select and show individual launch site success rates
  - helps to see how success varies across launch sites, payload mass, and booster version category
- Scatter plot
  - takes two inputs:
    - All sites or individual site
    - payload mass on a slider between 0 and 10000 kg
  - helps to visualize launch site success rate

GitHub URL:

https://github.com/yaredsha/datascience-ibm/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

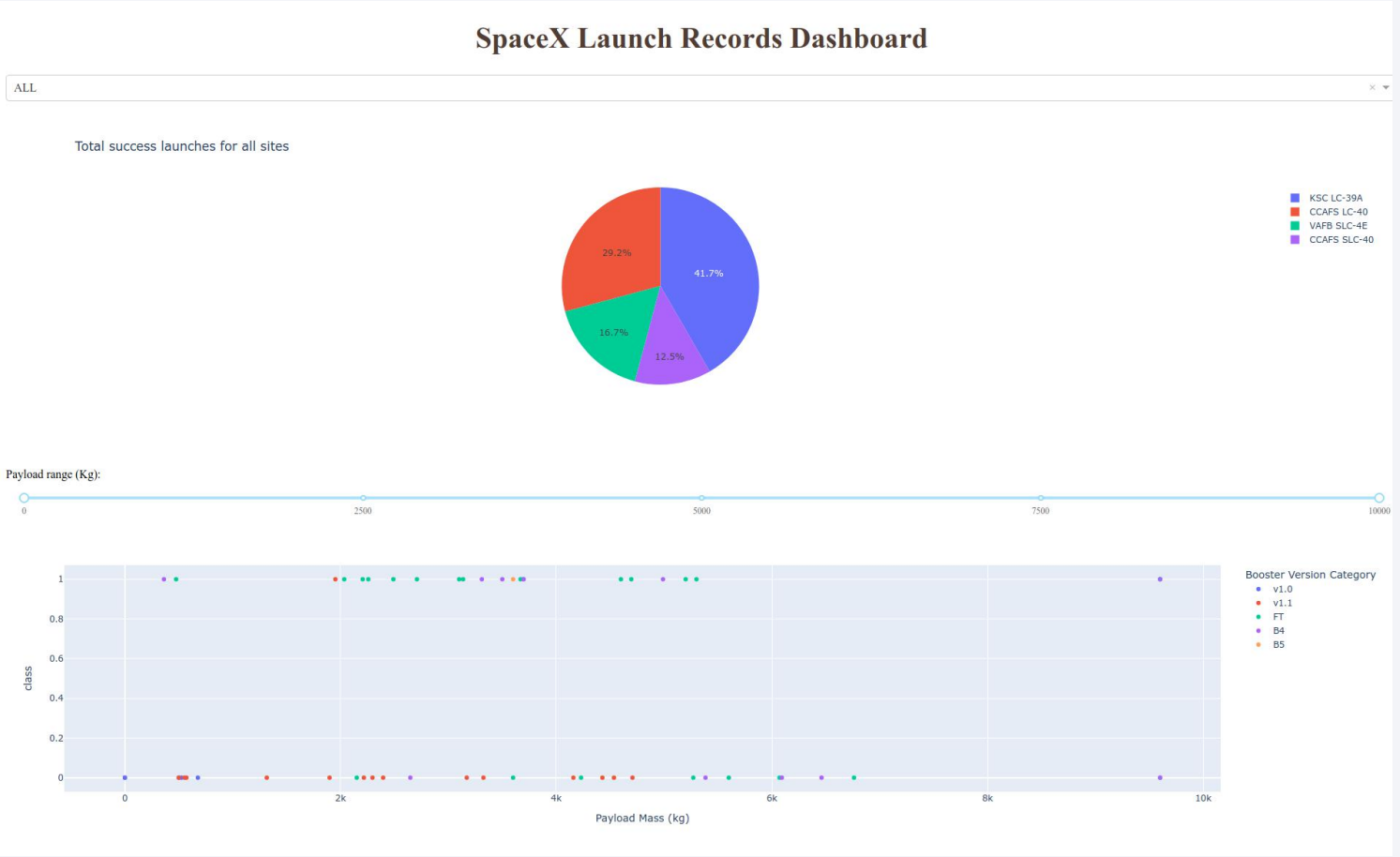


GitHub URL:

https://github.com/yaredsha/datascience-ibm/blob/main/SpaceX%20Machine%20Learning%20Prediction%20Part%205.ipynb

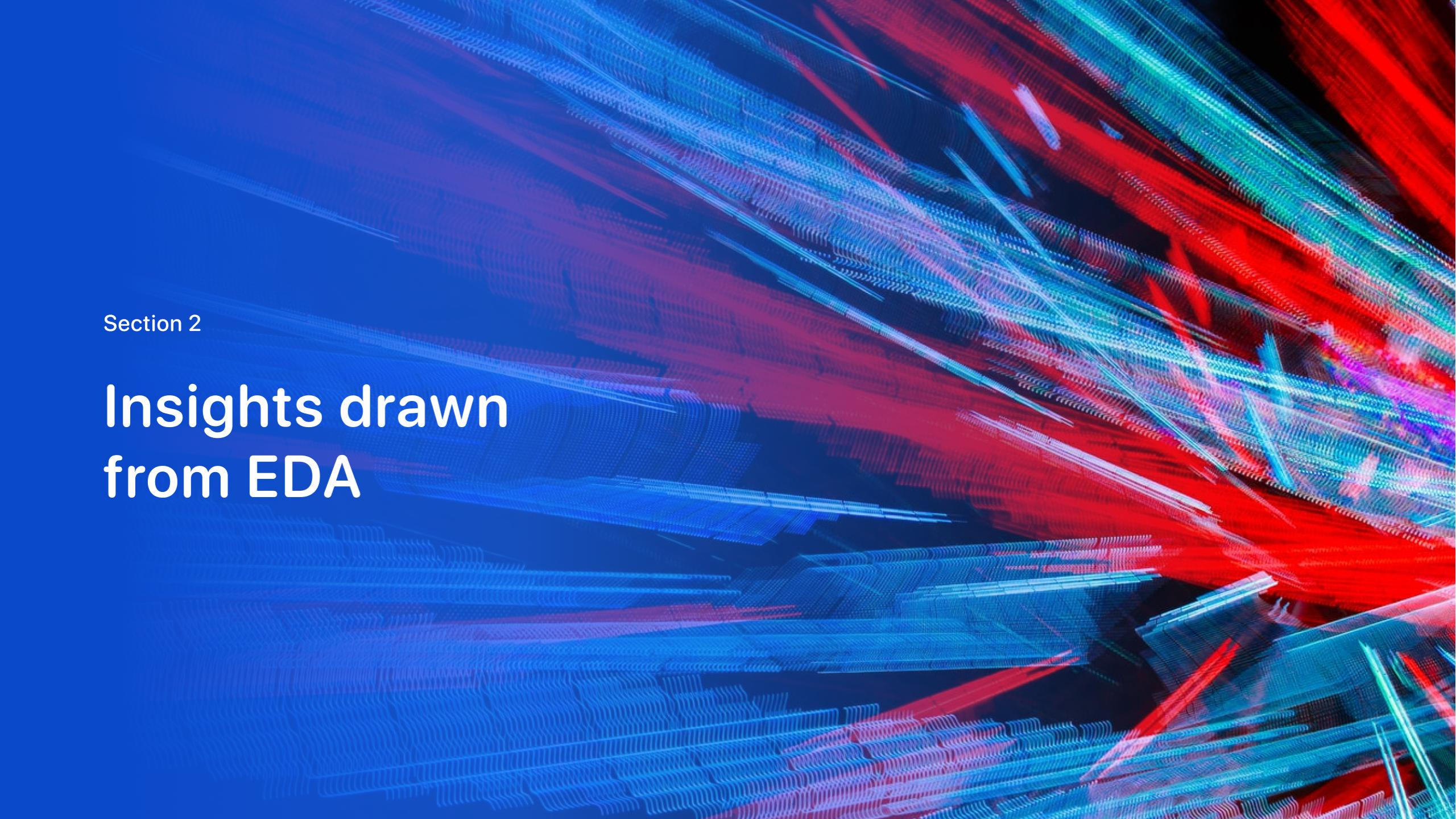
Results

- A preview of the Plotly dashboard (See next slides for the results/insights drawn from from each analysis)

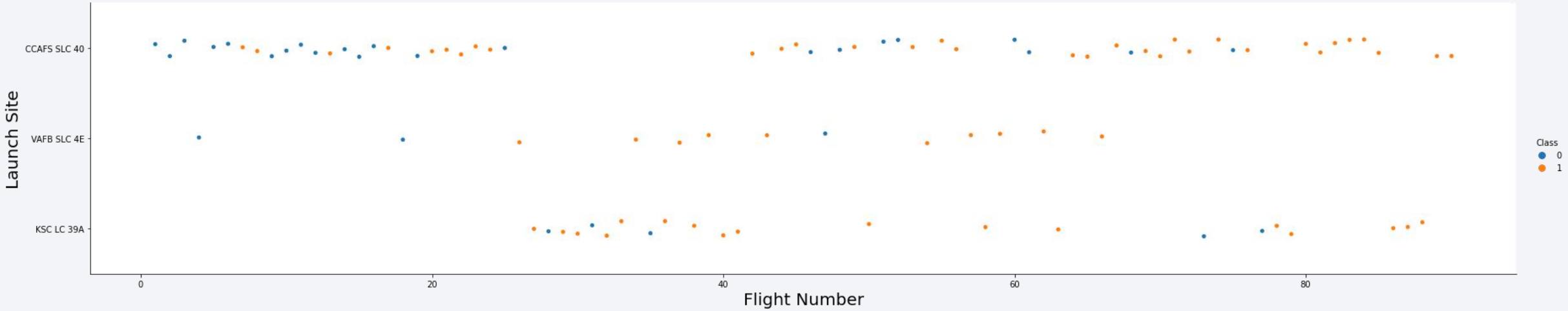


Section 2

Insights drawn from EDA



Flight Number vs. Launch Site

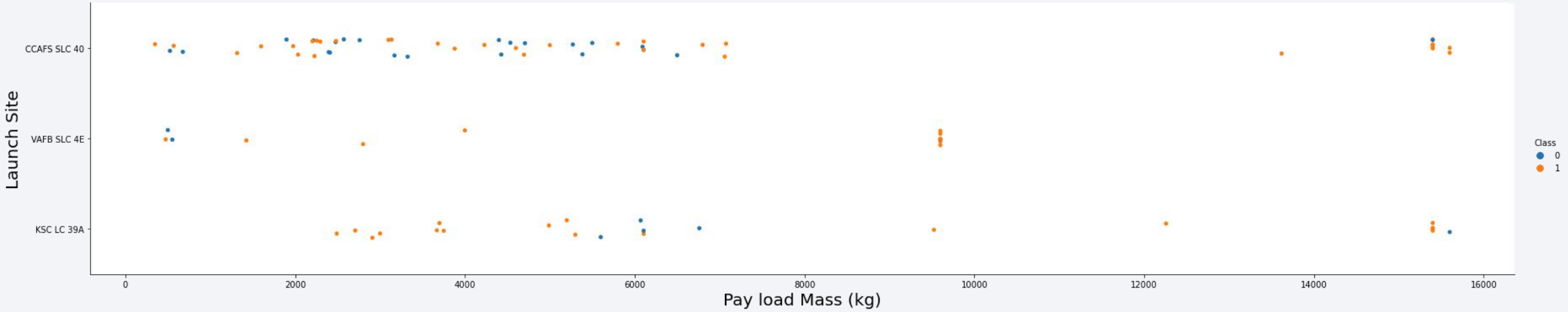


Orange: *successful launch*

Blue: *unsuccessful launch*

- We see different launch sites have different success rates
- Around flight number > 20, we see an increase in success rate
- CCAFS has the most volume of all launch sites indicating a main launch site

Payload vs. Launch Site

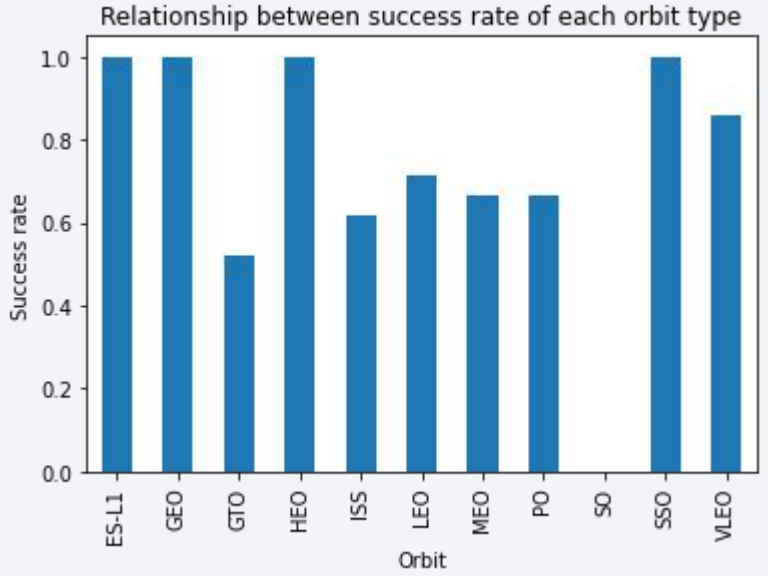


Orange: *successful launch*

Blue: *unsuccessful launch*

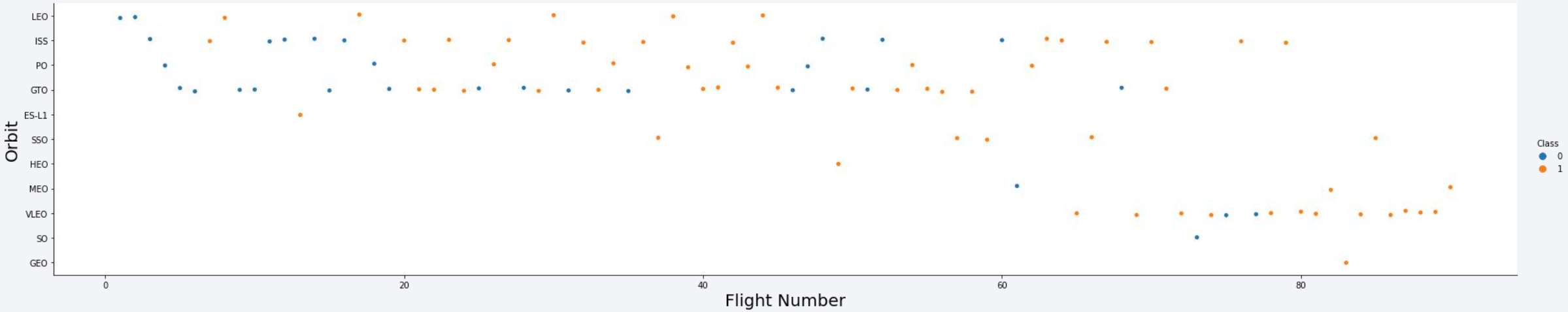
- Different launch sites have different success rates
- For the VAFB-SLC launchsite there are no rockets launched for heavy payload mass (greater than 10000 kg)
- Payoad mostly fall under less than 10000 kg

Success Rate vs. Orbit Type



- Different Orbit types have different success rates
- ES-L1, GEO, HEO and SSO have 100% success rate
- VLEO has around 90% success rate
- SO has 0% success rate
- The rest of the orbits have aound 50% success rate

Flight Number vs. Orbit Type

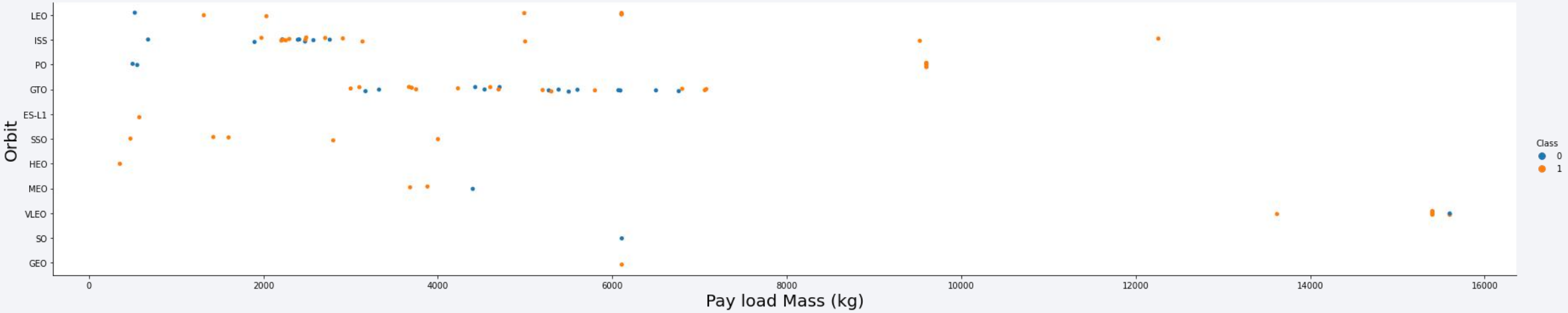


Orange: *successful launch*

Blue: *unsuccessful launch*

- Different Orbits have different success rates
- Launch Orbit preferences changed over Flight Number
- Launch sucess seem to increase over Flight Number and change of Orbit preference

Payload vs. Orbit Type

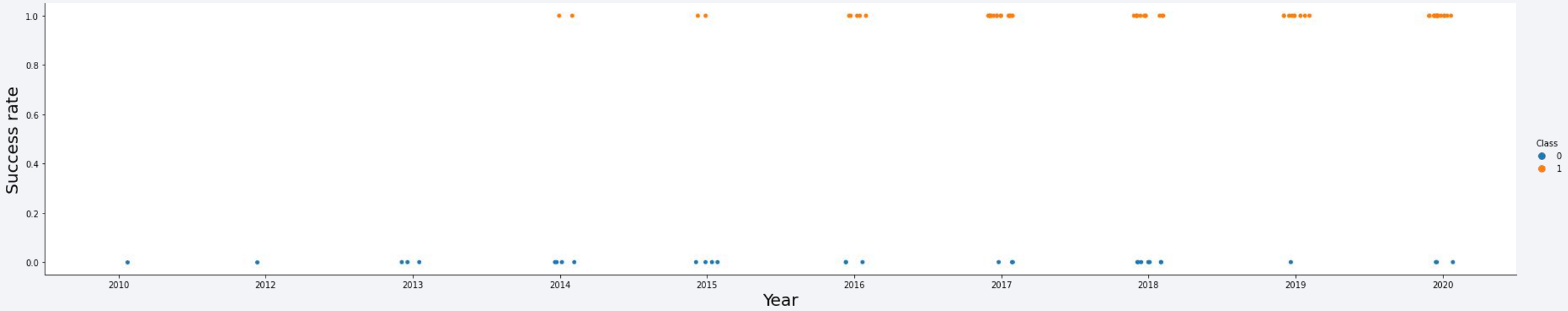


Orange: *successful launch*

Blue: *unsuccessful launch*

- Different Orbits have different success rates
- Payoad mostly fall under less than 10000 kg
- LEO and SSO seem to have relatively low payload mass
- VLEO only has payload mass values in the higher end of the range

Launch Success Yearly Trend



Orange: *successful launch*

Blue: *unsuccessful launch*

- Success rate continuesly increased since 2014 except for the year 2018
- 2017, 2019 and 2020 are the most successful years
- No success until 2014

All Launch Site Names

```
In [26]: %%sql
         SELECT LAUNCH_SITE FROM SPACEX GROUP BY LAUNCH_SITE
          * ibm_db_sa://slj93772:***@ba99a9e6-d59e-4883-8fc0-d6
         Done.
```

Out[26]:

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

- The query lists all launch site names by group the data by lanch sites and selecting the launch site names

Launch Site Names Begin with 'CCA'

```
In [8]: %%sql
SELECT * FROM SPACEX WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://slj93772:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

Out[8]:

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

- The query selects all records where launch sites begin with the string 'CCA'. This is done using the SQL LIKE clause
- It also limits the number of the results by 5 using the LIMIT clause

Total Payload Mass

```
In [10]: %%sql
         SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_MASS FROM SPACEX WHERE CUSTOMER = 'NASA (CRS)'
          * ibm_db_sa://slj93772:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde
         Done.
Out[10]:
| total_mass |
|---|
| 45596 |
```

- The query calculates the total payload mass (in kg) carried by boosters launched by NASA (CRS)
- It uses the SUM function to get the total payload mass value
- The query filters the records using the WHERE clause on the CUSTOMER column to get results only for NASA (CRS)

Average Payload Mass by F9 v1.1

```
In [11]: %%sql
         SELECT AVG(PAYLOAD_MASS__KG_) AS TOTAL_MASS FROM SPACEX WHERE BOOSTER_VERSION = 'F9 v1.1'
          * ibm_db_sa://slj93772:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.da
         Done.
Out[11]: total_mass
         2928
```

- The query selects the average payload mass carried by booster version F9 v1.1
- It uses the AVG function to get the average payload mass value
- The query filters the records using the WHERE clause on the BOOSTER_VERSION column to get results only for F9 v1.1

First Successful Ground Landing Date

```
In [14]: %%sql
         SELECT MIN(DATE) AS DATE FROM SPACEX WHERE LANDING__OUTCOME = 'Success (ground pad)'
          * ibm_db_sa://slj93772:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqd
        Done.
```

Out[14]:

DATE
2015-12-22

- The query selects the date when the first successful landing outcome in ground pad was acheived
- It uses the MIN function to get oldest date
- The query filters the records using the WHERE clause on the LANDING__OUTCOME column to get successful landings only for ground pad

Successful Drone Ship Landing with Payload between 4000 and 6000

```
In [15]: %%sql
SELECT DISTINCT(BOOSTER_VERSION) AS BOOSTER_VERSION FROM SPACEX WHERE LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000
 * ibm_db_sa://slj93772:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

Out[15]:

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

- The query lists the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- It uses the DISTINCT function to get distinct booster names
- The query filters the records using the WHERE clause on the LANDING__OUTCOME column to get successful landings only for drone ship and also to get only those records with payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

```
In [19]: %%sql
SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS COUNT FROM SPACEX GROUP BY MISSION_OUTCOME
 * ibm_db_sa://slj93772:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.datab
Done.
```

Out[19]:

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

- The query list the total number of successful and failure mission outcomes
- It groups the records by MISSION_OUTCOME column
- The query uses the COUNT function to get the number of outcomes per MISSION_OUTCOME group

Boosters Carried Maximum Payload

```
In [20]: %%sql
SELECT DISTINCT(BOOSTER_VERSION) FROM SPACEX WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX)
 * ibm_db_sa://slj93772:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:
Done.
```

Out[20]:

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

- The query selects the names of the booster_versions which have carried the maximum payload mass
- It uses a subquery to get the maximum playload mass value
- It also uses the WHERE clause on PAYLOAD_MASS__KG_ column to filter only those records that have the maximum payload mass
- The query uses the DISTINCT function to get distinct booster names

2015 Launch Records

```
In [21]: %%sql
         SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND YEAR(DATE) = 2015
          * ibm_db_sa://slj93772:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
         Done.
```

Out[21]:

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

- The query selects the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- It filters the records using the WHERE clause on the LANDING__OUTCOME column to get only failed landing outcomes
- Additionally the column DATE is used in the WHERE clause to get results only for the year 2015
- The function YEAR is used to extract the year from the DATE column

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

```
In [5]: %%sql
SELECT LANDING__OUTCOME, COUNT(LANDING__OUTCOME) AS RANK FROM SPACEX WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING__OUTCOME ORDER BY RANK DESC
 * ibm_db_sa://slj93772:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

Out[5]:

landing__outcome	RANK
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

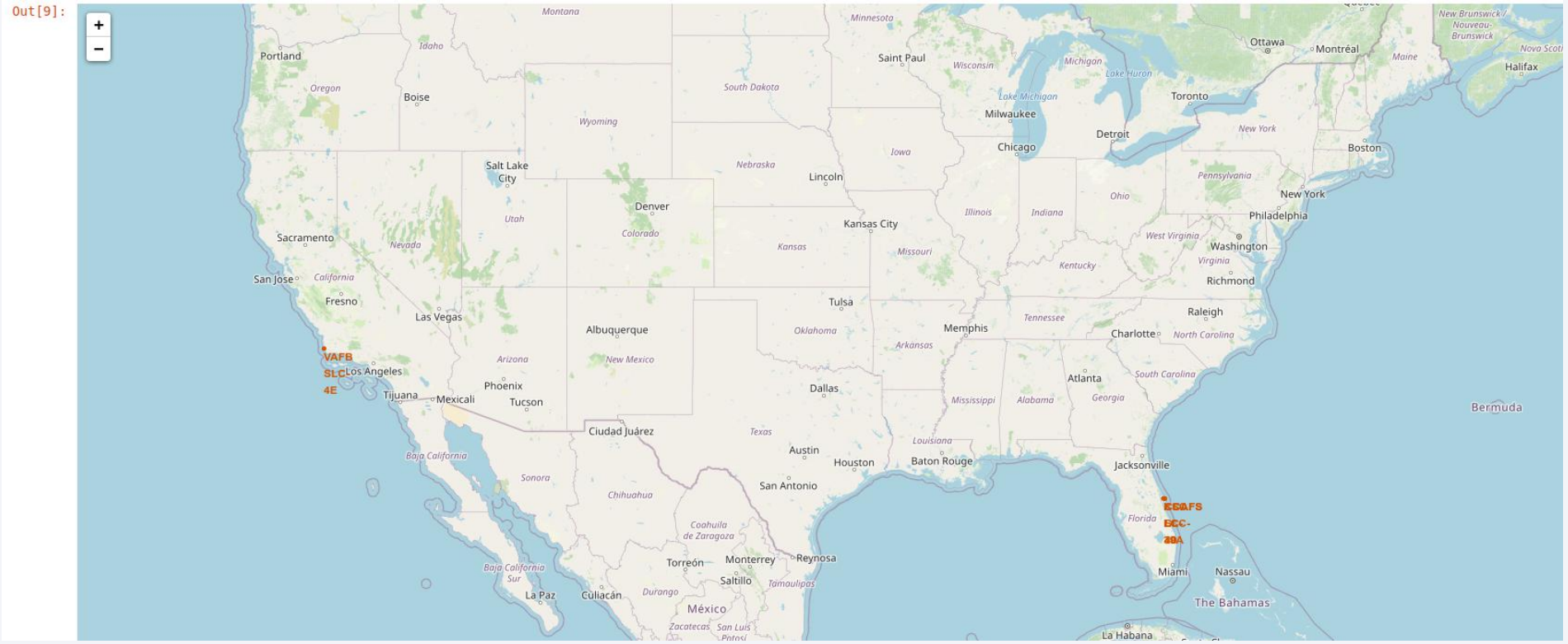
- The query ranks the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order
- It groups the records by LANDING__OUTCOME column
- The query uses the COUNT function to get the number of outcomes per LANDING__OUTCOME group and displays them as RANK
- It filters the records using the WHERE clause on the DATE column to get only records between 2010-06-04 and 2017-03-20
- It uses the ORDER BY clause with DESC to display the records in descending order

Section 4

Launch Sites Proximities Analysis

All launch Sites‘ Map

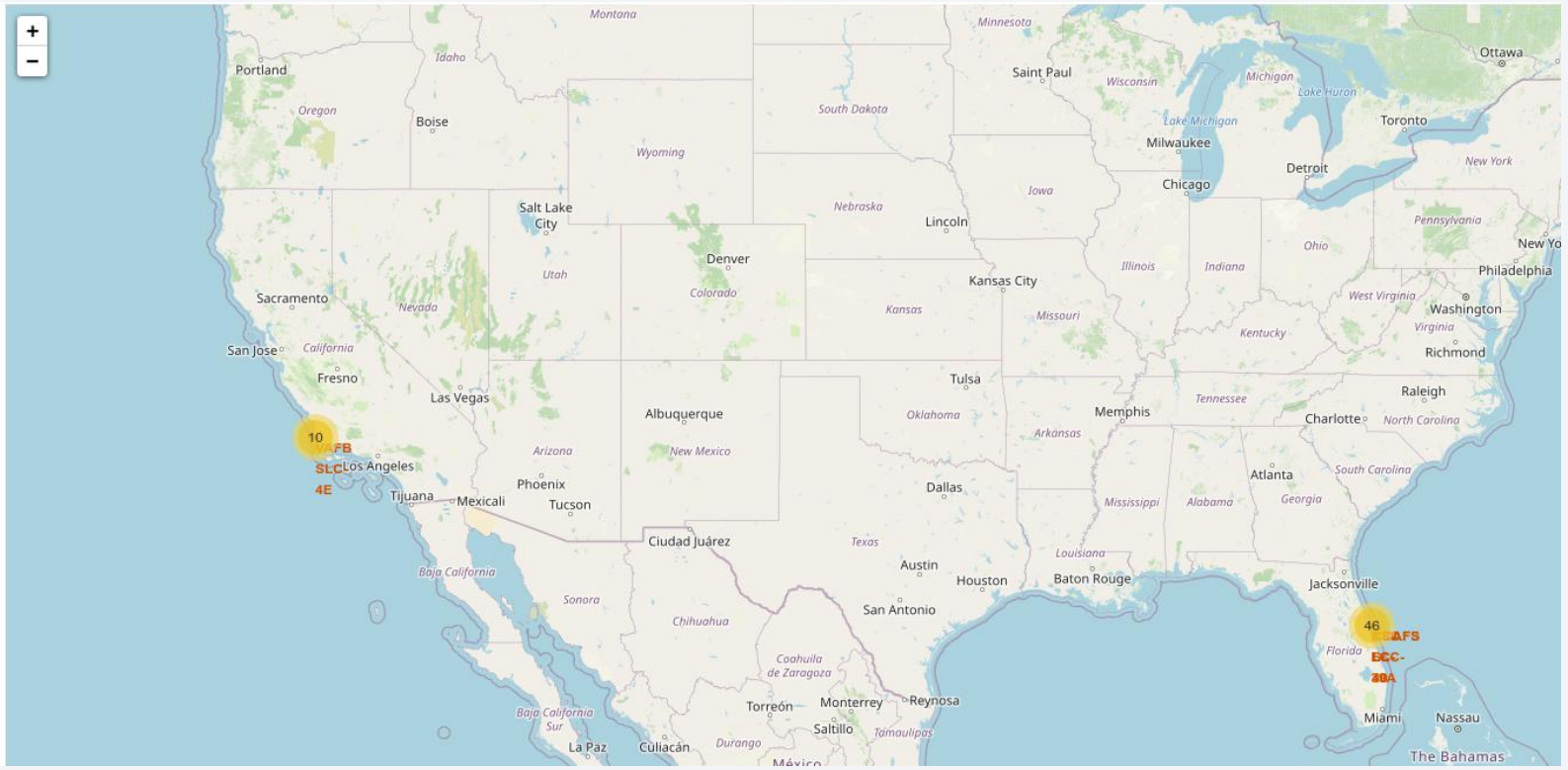
Out[9]:



- Generated folium map displaying all launch sites’ location markers on a global map
- Most of the launch sites are located in Florida
- All launch sites are near the ocean

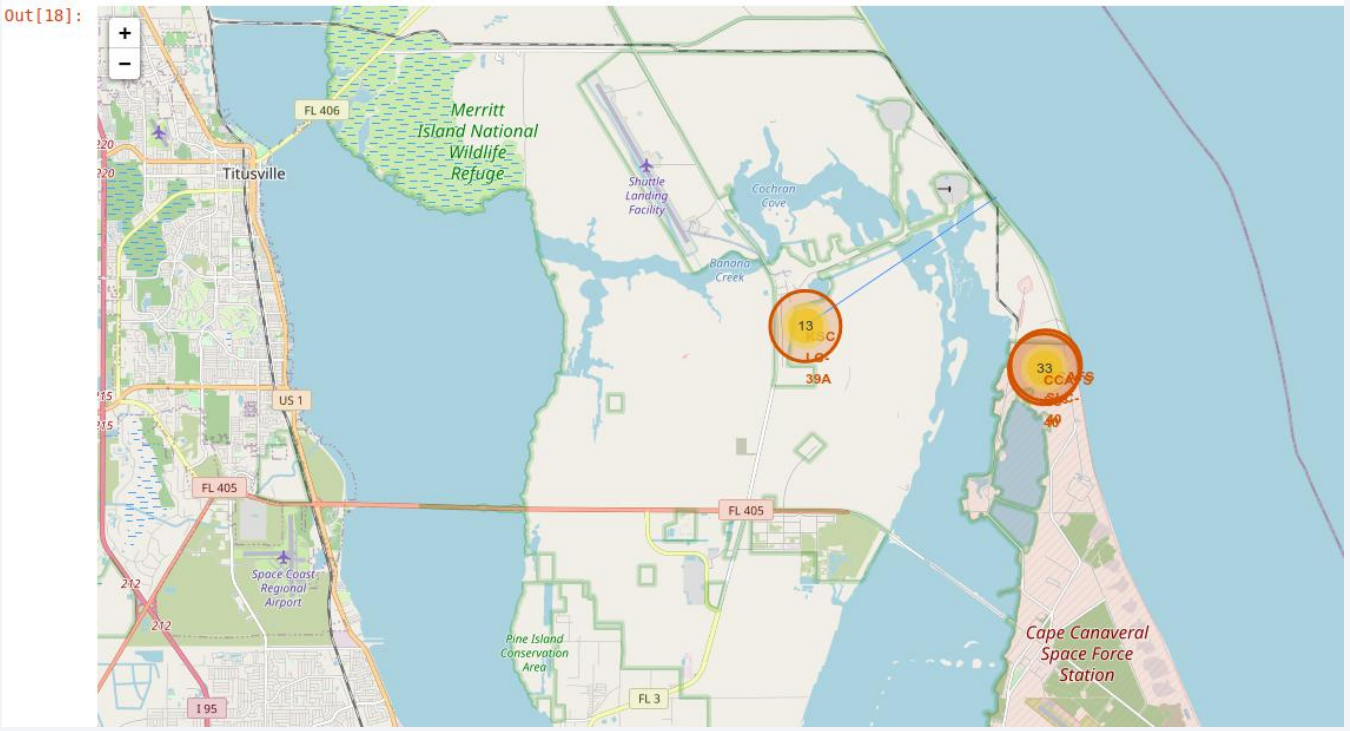
Success/failed launches for each site on the map

Out[13]:

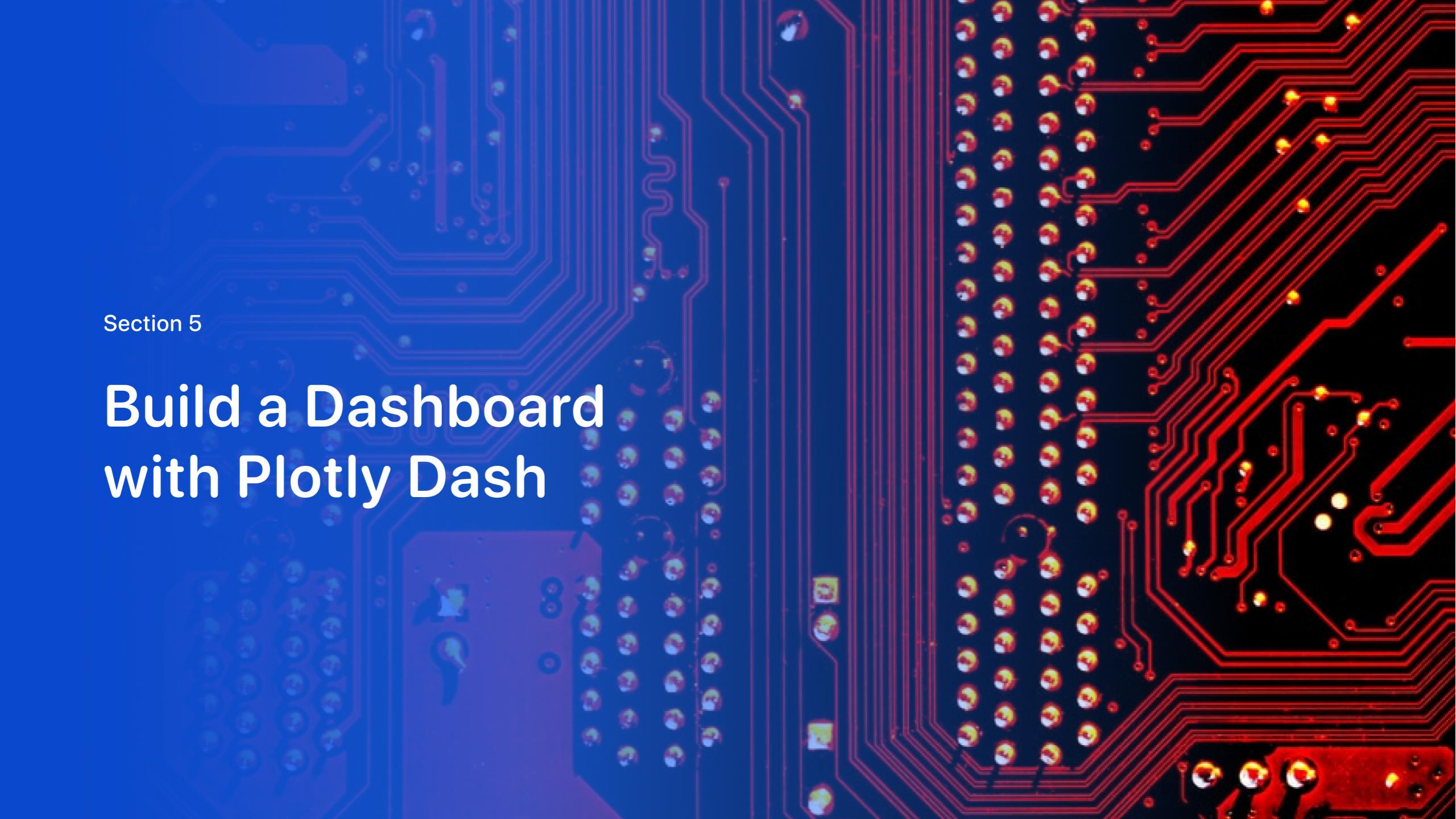


- Generated folium map displaying all launch sites' location markers on a global map
- The success/failed launches for each site are marked on the map

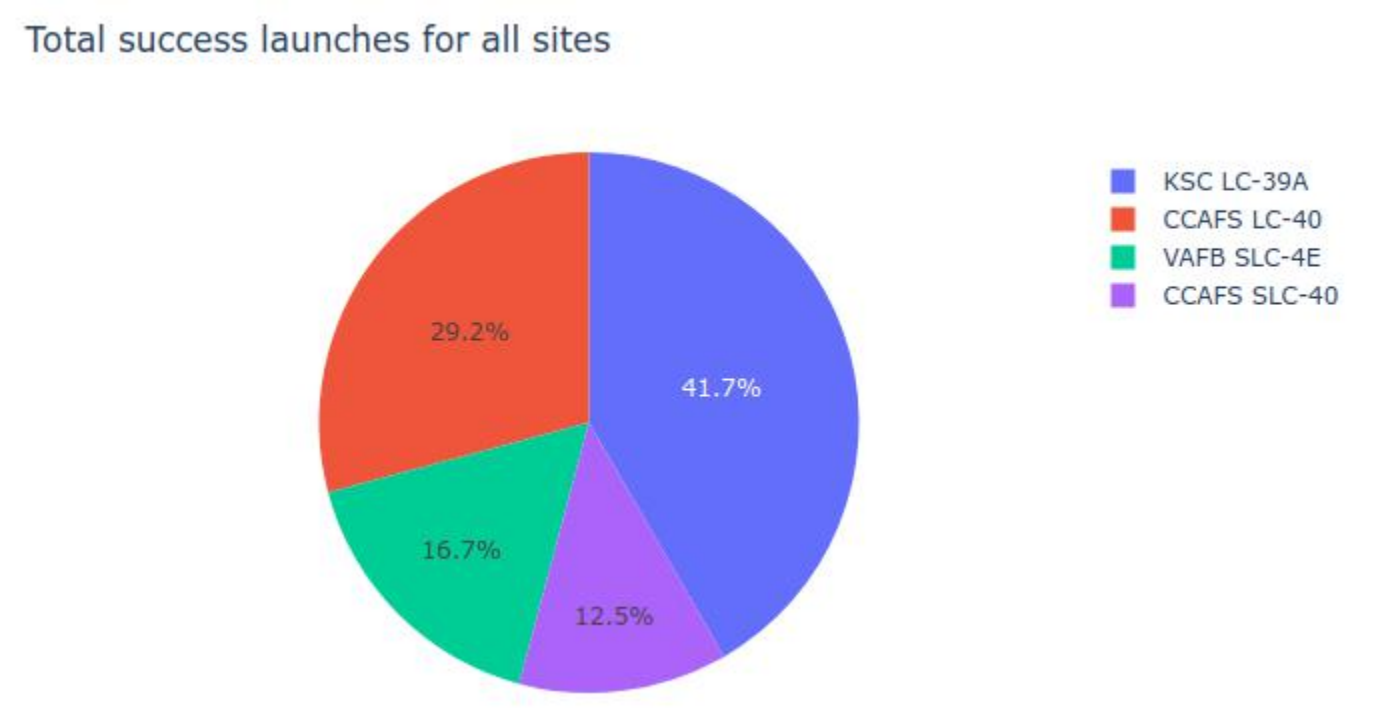
Launch sites to proximities



- The distances between a launch site to its proximities are calculated and drawn with a blue line

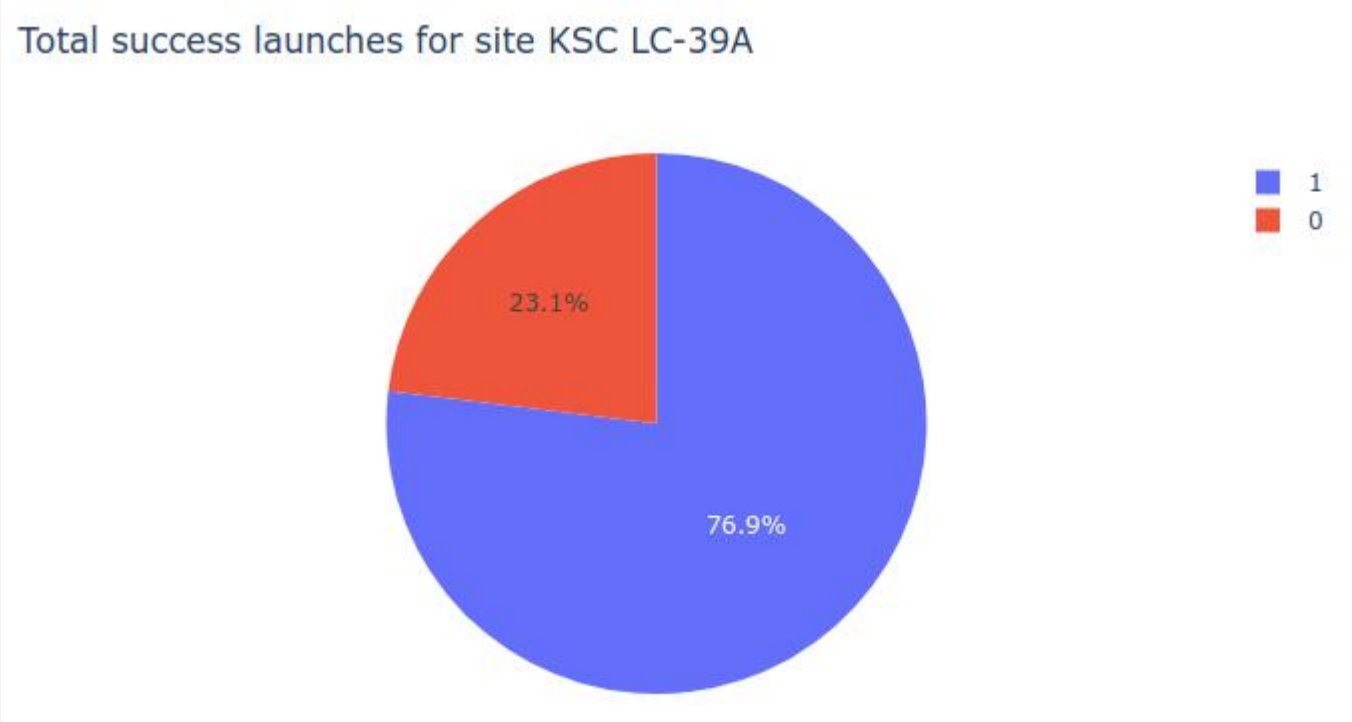

Section 5
Build a Dashboard with Plotly Dash

Total success launches for all sites



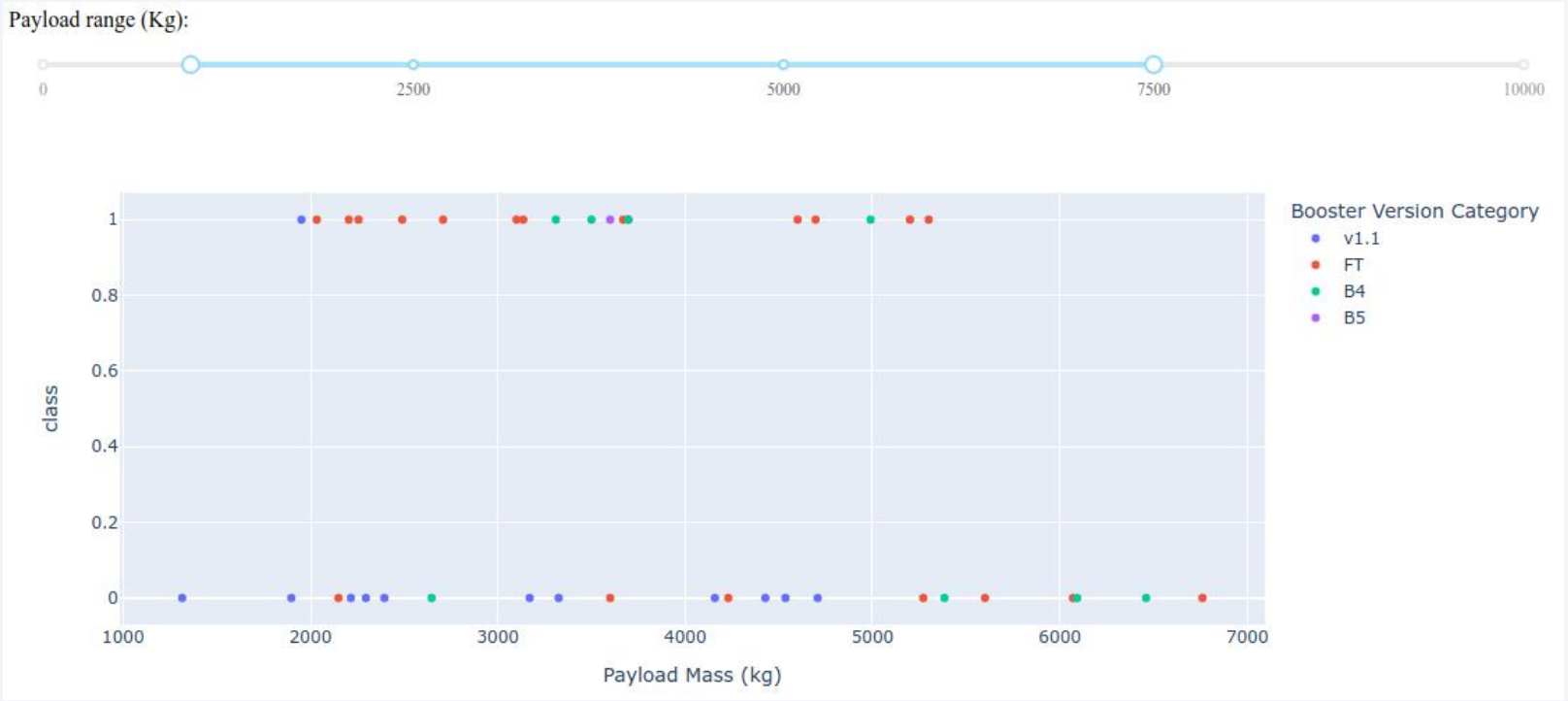
- The piechart shows the total success launches for all sites
- KSC LC-39A has the most number of successful launches
- CCAFS SLC-40 has the least number of successful launches

Launch site with highest launch success ratio



- KSC LC-39A has the highest launch success ratio
- It has 76.9% success ratio

Payload vs. Launch Outcome for all sites

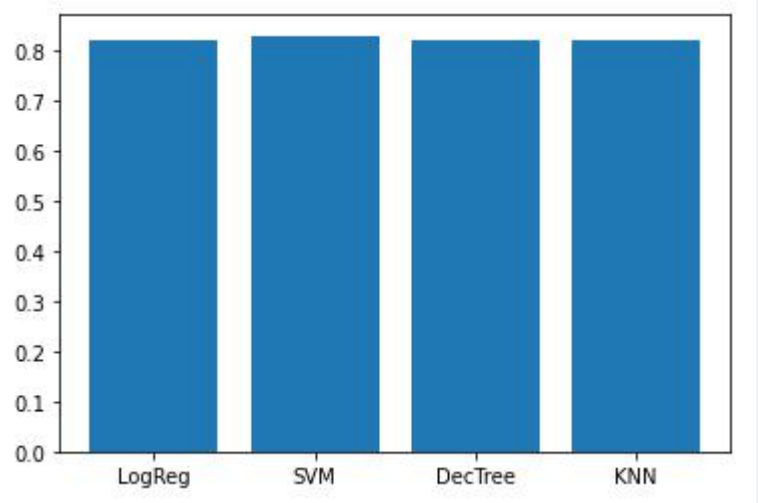


- Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider
- Payload between 1000 kg to 7500 kg selected in the range slider

Section 6

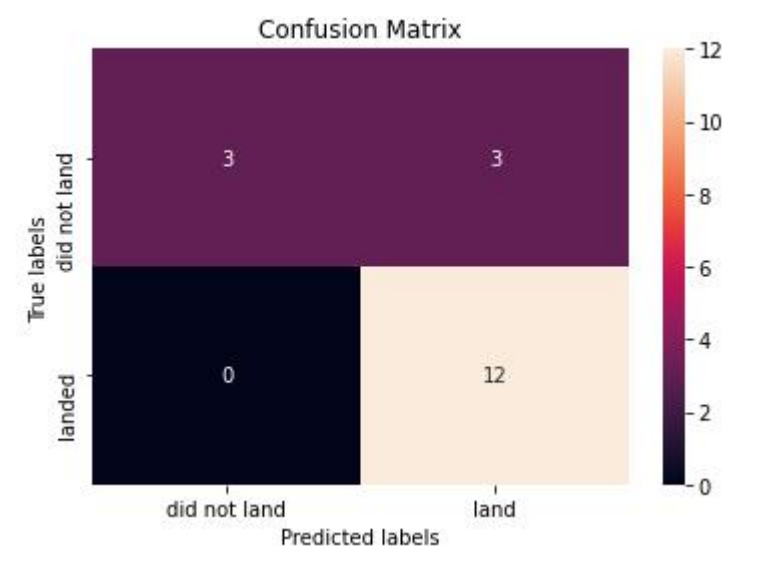
Predictive Analysis (Classification)

Classification Accuracy



- Model accuracy for all built classification models
- SVM has the highst model accuracy of 83%
- Overall the model accuracy of all models is similar most likely due to small data size

Confusion Matrix



- Confusion matrix of the best performing model (SVM)
- The confusion matrix is similar for all models due to small data size
- The models predicted 12 successful landings when the true label was successful landing
- The models predicted 3 unsuccessful landings when the true label was unsuccessful landing
- The models predicted 3 successful landings when the true label was unsuccessful landings (false positives)

Conclusions

- Goal of the project
  - develop the best machine learning model for a company that wants to bid against Space X
- Predict when Stage 1 will successfully land to save $100 million USD
- Data collected from a public Space X API and web scraping Falcon 9 Wikipedia page
- Created an interactive dashboard for visualization
- Created a machine learning model with an accuracy of 83%

Appendix

GitHub URL:

GIT: https://github.com/yaredsha/datascience-ibm.git

WEB: https://github.com/yaredsha/datascience-ibm

Thank you!