Winning Space Race with Data Science

<Bankanidhi Sahoo>
<12-03-2022>

https://github.com/bankanidhi/Python-Data-Science

Outline

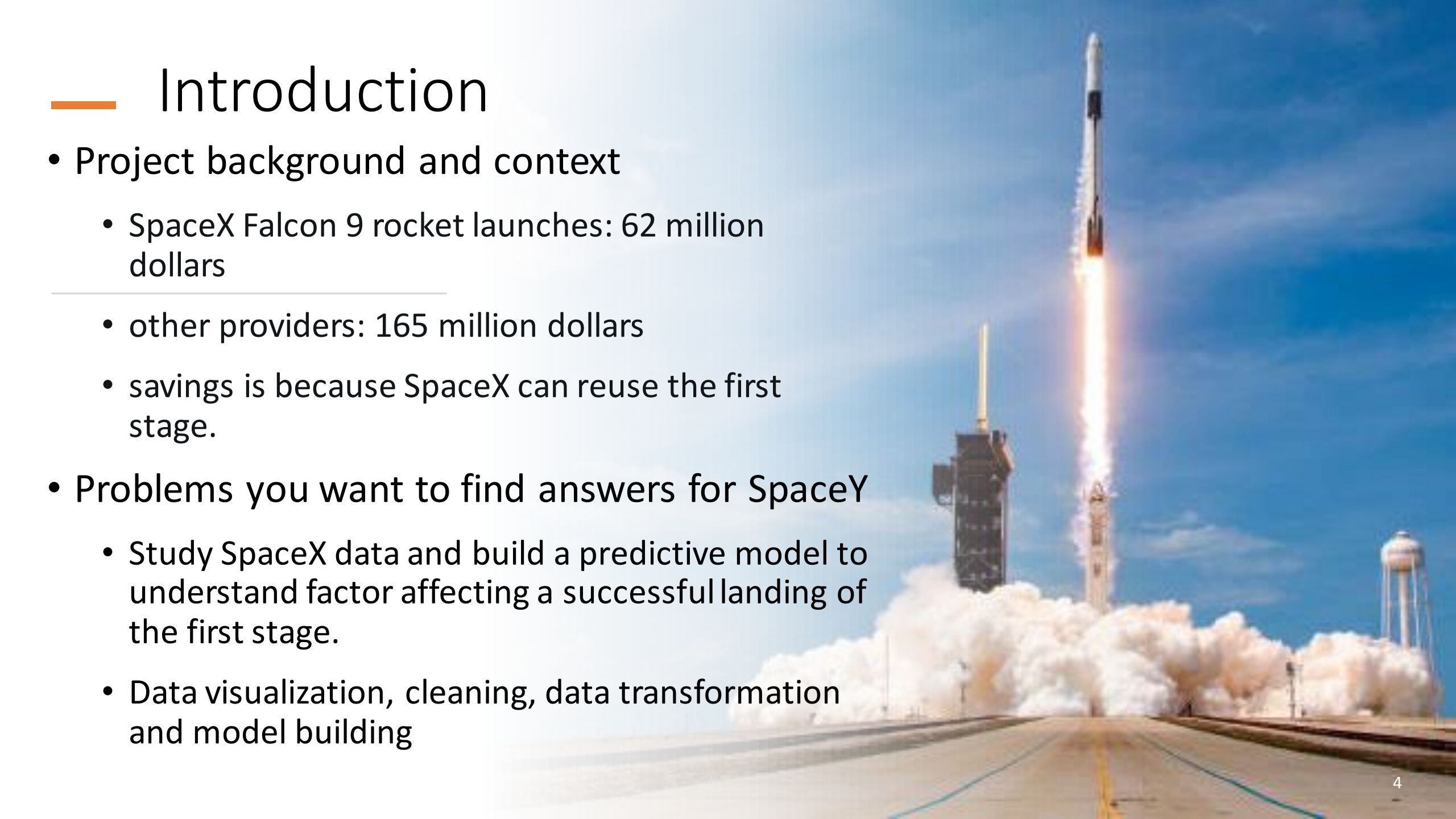
- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- Summary of methodologies
  - Data provided by SpaceX which includes various information along with the success of the projects. The information are variables which includes [BoosterVersion, PayloadMass,Orbit, Launch site etc] alongwith the target (success or failure).
  - Categorical variable are converted to Numerical variables by using one hot encoding followed by standardization using StandardScaler which is like mean centering.
  - The data are then used for building predictive model using 4 methods Logistic regression, Support Vector Machine, Decision tree Classification and K Nearest Neighbors.
- Summary of all results
  - All four method perform equally of test samples with an accuracy of 83%

Introduction

- Project background and context
  - SpaceX Falcon 9 rocket launches: 62 million dollars
  - other providers: 165 million dollars
  - savings is because SpaceX can reuse the first stage.
- Problems you want to find answers for SpaceY
  - Study SpaceX data and build a predictive model to understand factor affecting a successful landing of the first stage.
  - Data visualization, cleaning, data transformation and model building



Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - From SpaceX public API and Wikipedia
- Perform data wrangling
  - number of launches on each site, number and occurrence of orbit, mission outcome per orbit and mission landing outcome by onehot encoding.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Tuned model using GridSearchCV with 4 different models.

Data Collection

- SpaceX API

FlightNumber, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, LandingPad,Longitude, Latitude etc.

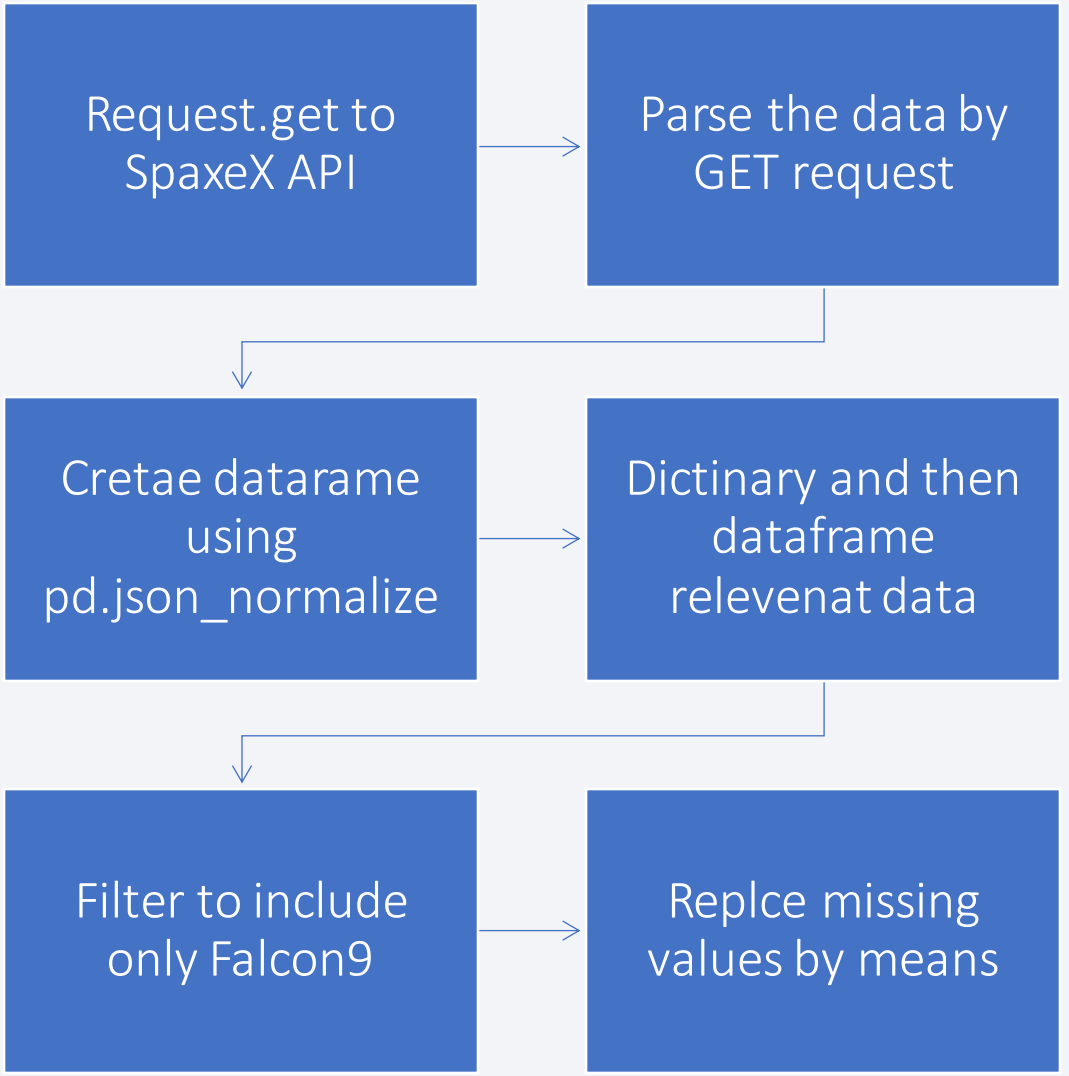
- Webscraping

Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, etc

Data Collection – SpaceX API

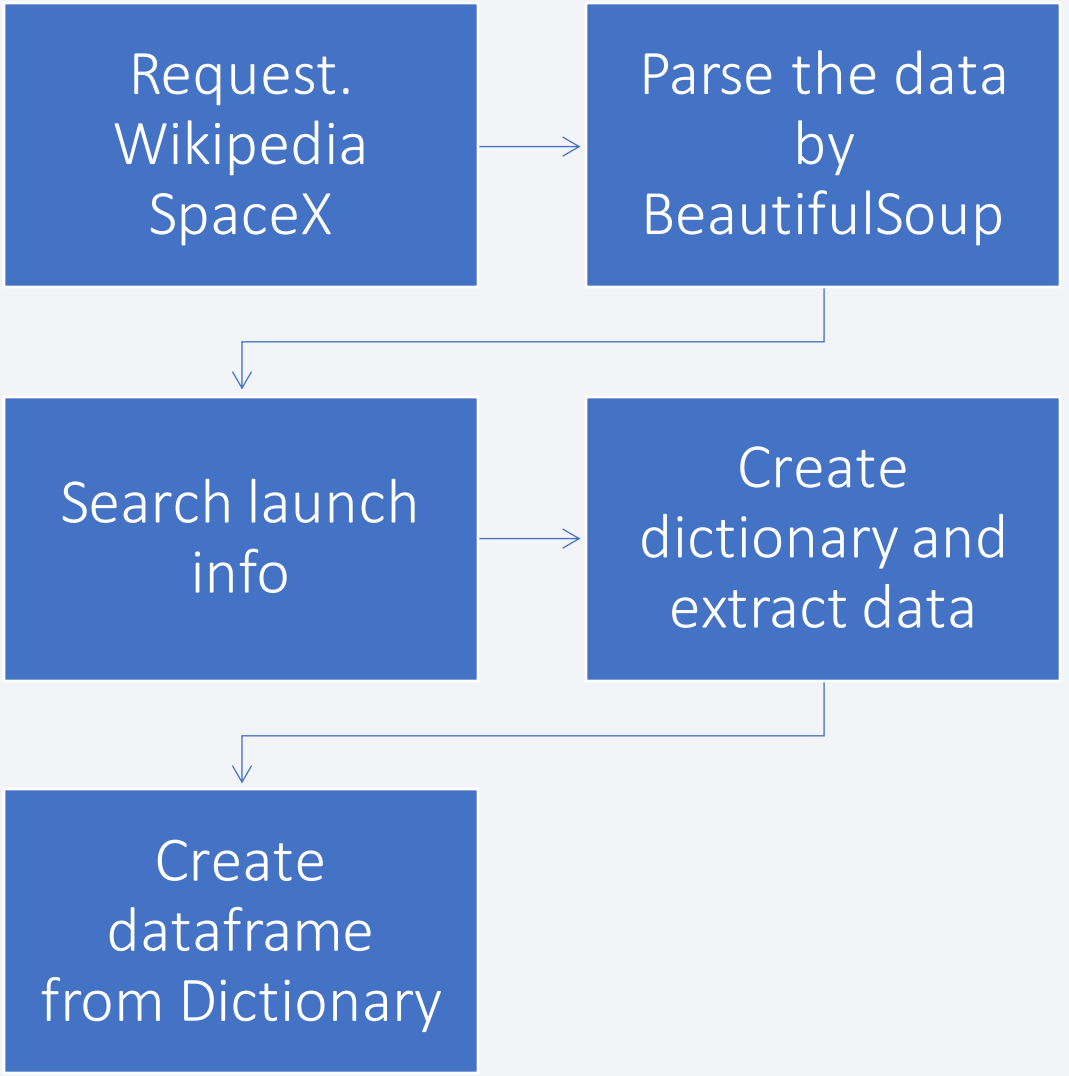
- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- Add the GitHub URL of the completed SpaceX API calls notebook https://github.com/bankanidhi/Python-Data-Science/blob/main/Data_Collection_API.ipynb



Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

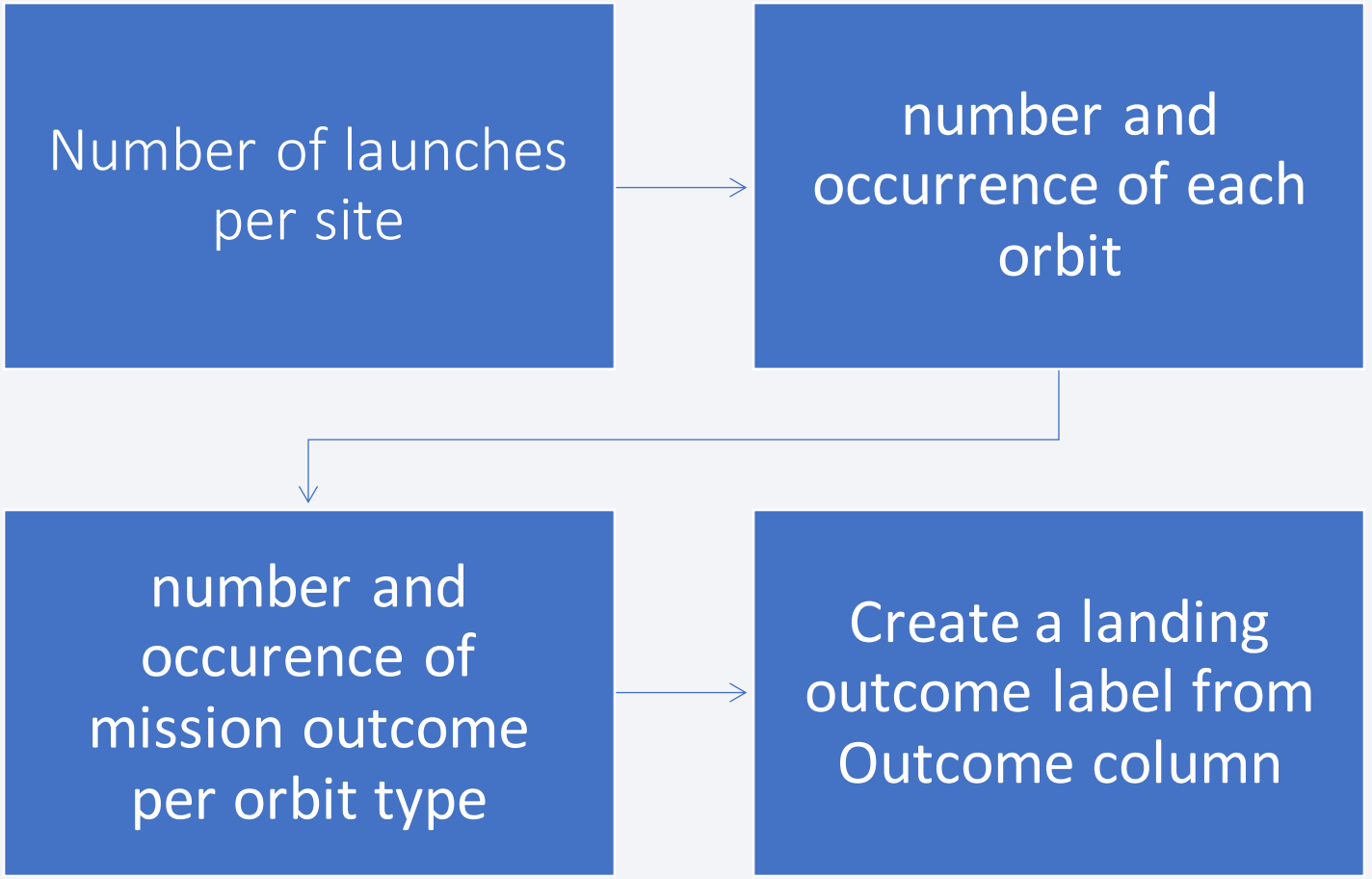


- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose
- https://github.com/bankanidhi/Python-Data-Science/blob/main/Data%20Collection%20with%20Web%20Scraping%20lab.ipynb

Data Wrangling

- Describe how data were processed
- You need to present your data wrangling process using key phrases and flowcharts

- https://github.com/bankanidhi/Python-Data-Science/blob/main/Data_Wrangling.ipynb



EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
  - Scatterplot, histplot and line plot are used to see interdependency of different variables on each other such as 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

- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose
- https://github.com/bankanidhi/Python-Data-Science/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
  - uploaded data set into IBM DB2 Database.
  - Queried using SQL Python.
  - Various queries were performed to get a more understanding of the dataset such as about launch site names, mission outcomes, various pay load sizes of customers and booster versions, and landing outcomes

- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose

- https://github.com/bankanidhi/Python-Data-Science/blob/main/EDL_with_SQL.ipynb

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Map gives information about the location of the launch site and information about the nearby surrounding so all precaution can be taken for the launching.
- Location, succeessful launching, nearby places with distance and line of connection are used.

- https://github.com/bankanidhi/Python-Data-Science/blob/master/Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Build a Dashboard with Plotly Dash

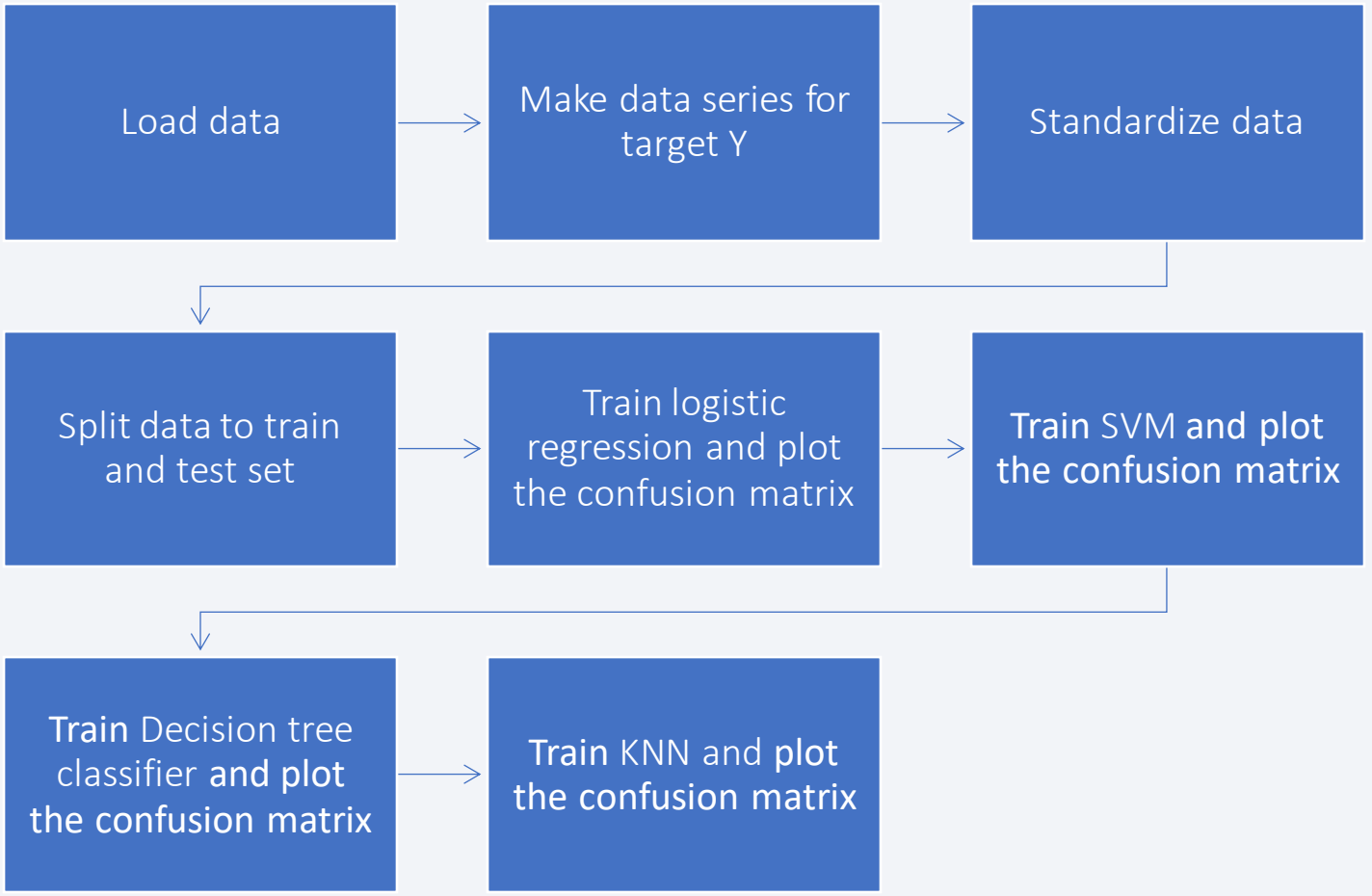
- A pie chart and a scatter plot are used.
- Pie chart shows successful landings across all launch sites and can be selected to show individual launch site success rates.
- Scatter plot takes two inputs: All sites or individual site and payload mass as a slider .

- https://github.com/bankanidhi/Python-Data-Science/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Models are trained to check the accuracy. As shown in the flowchart

- Add the GitHub URL: https://github.com/bankanidhi/Python-Data-Science/blob/main/Machine%20Learning%20Prediction.ipynb

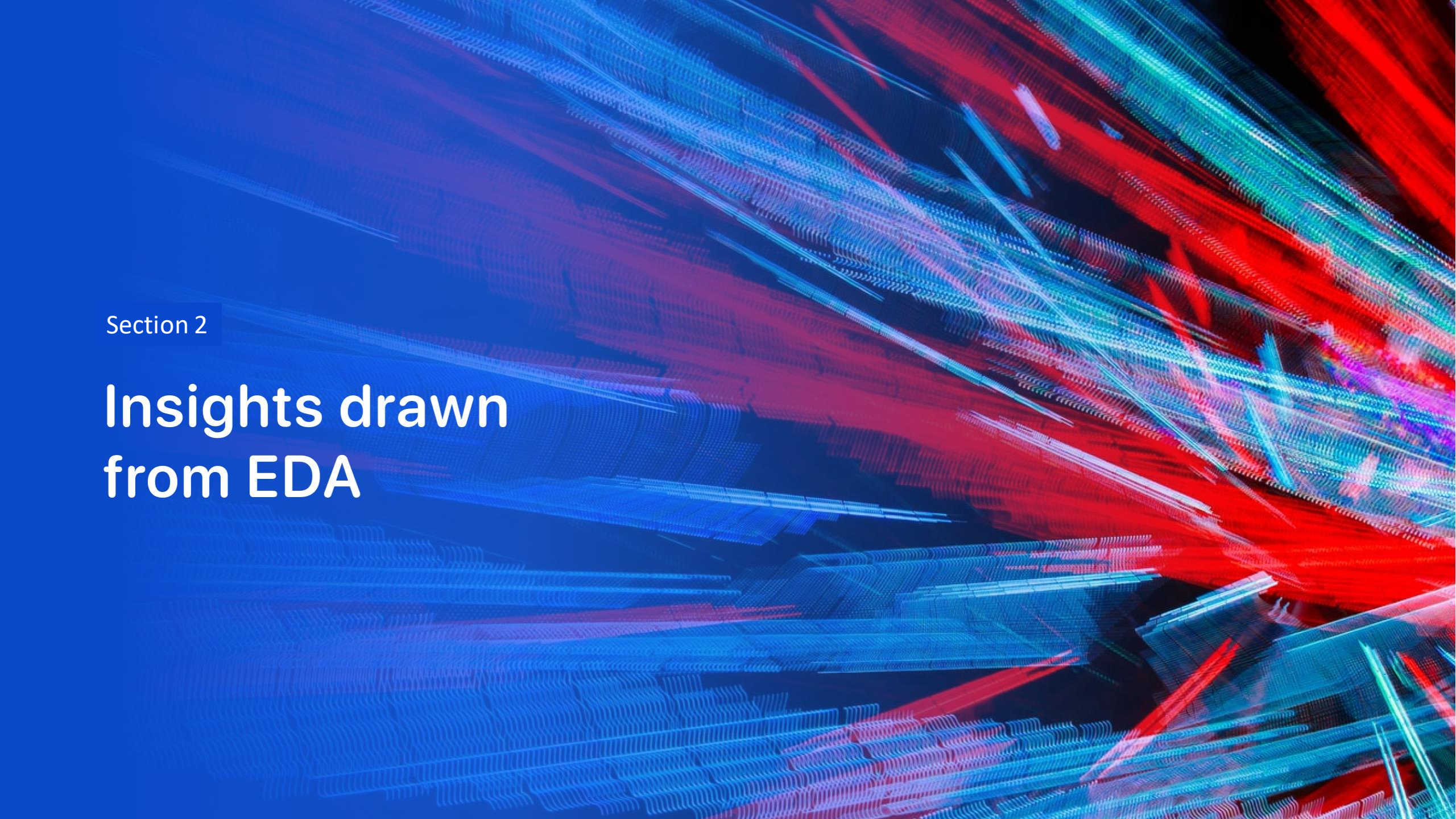


Results

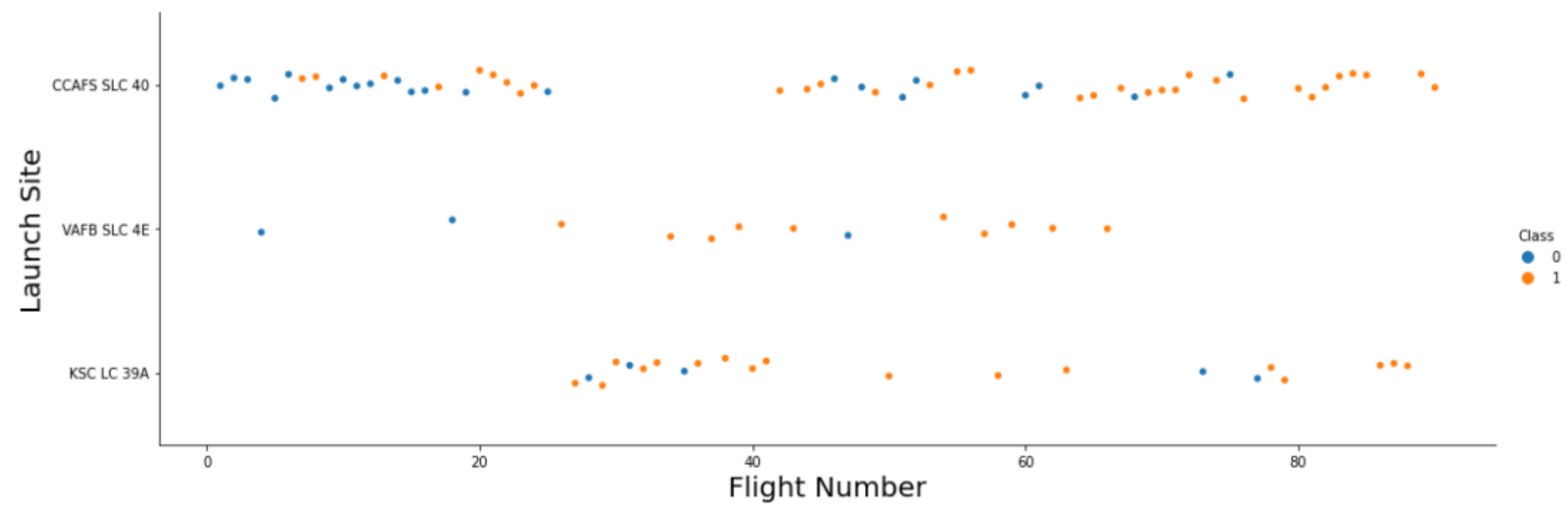
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA



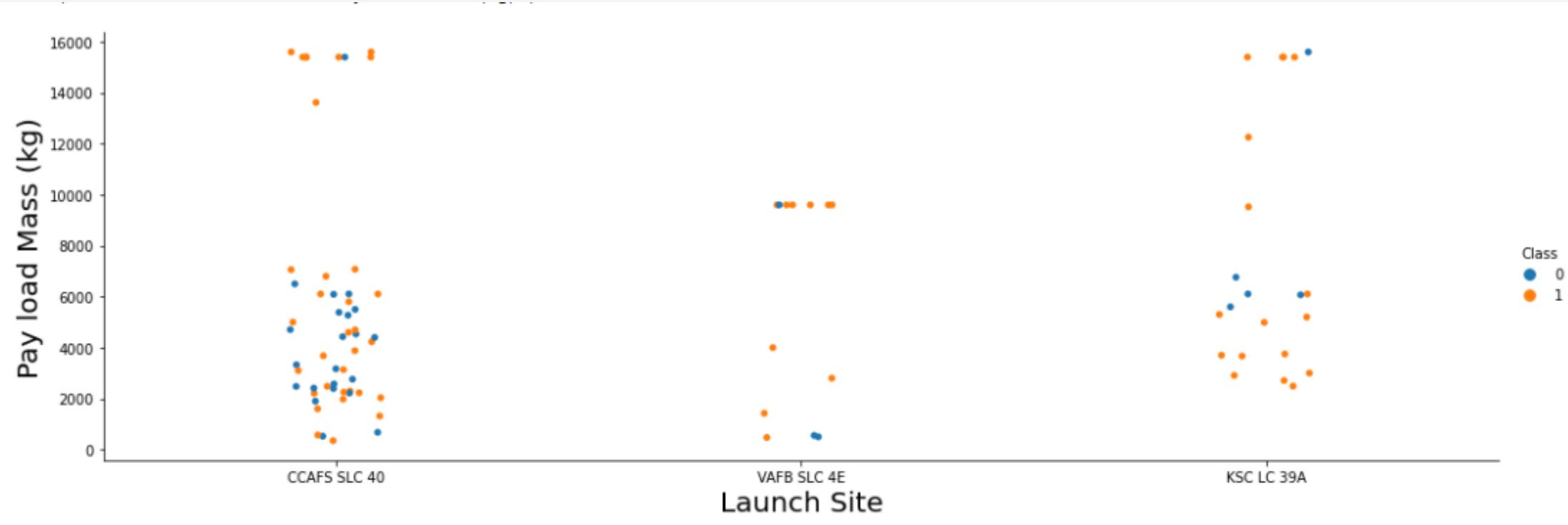
Flight Number vs. Launch Site



Blue shows unsuccessful, and orange for successful launch.

CCAFS is the main launch site as it has the most number of launches.

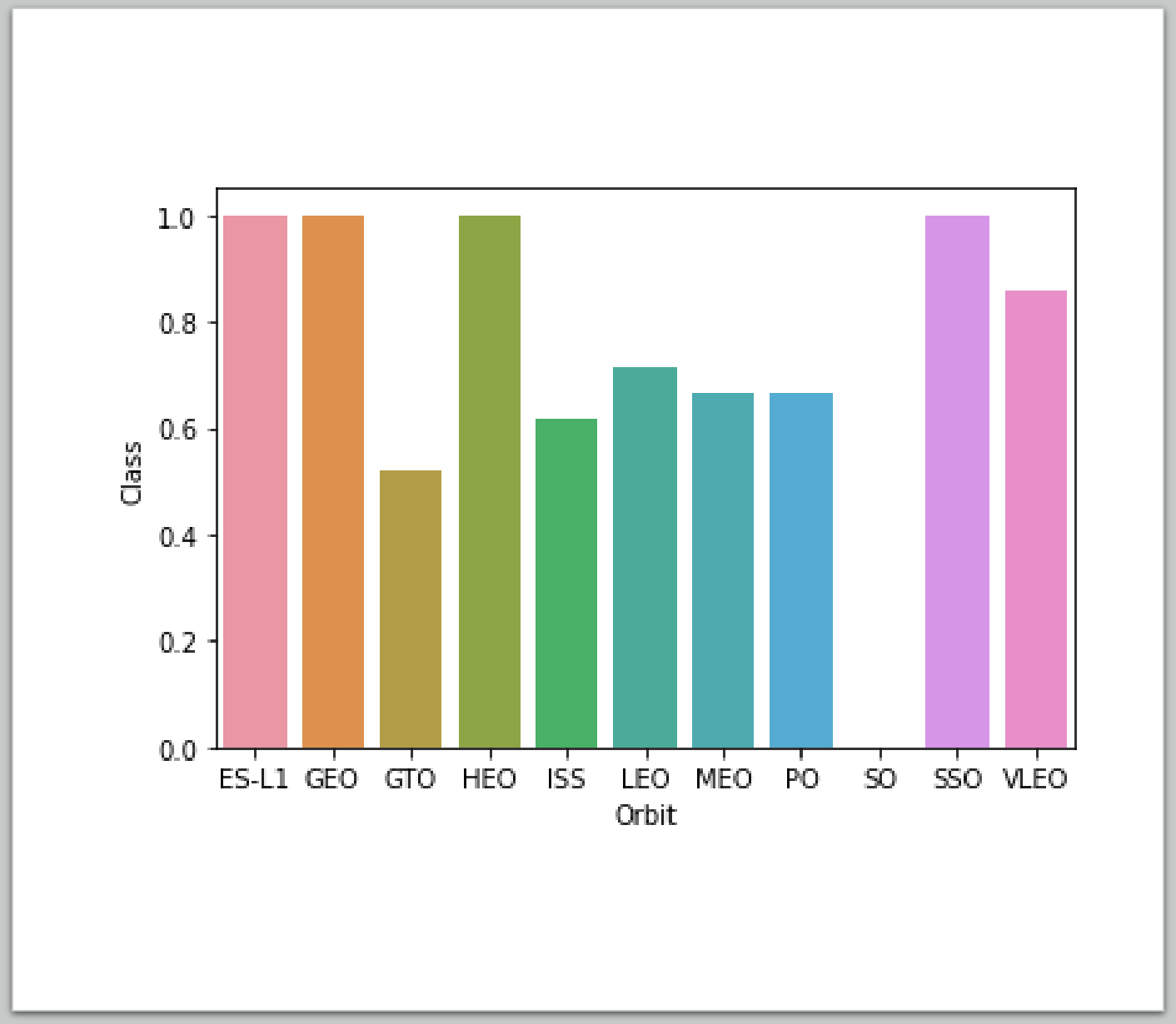
Payload vs. Launch Site



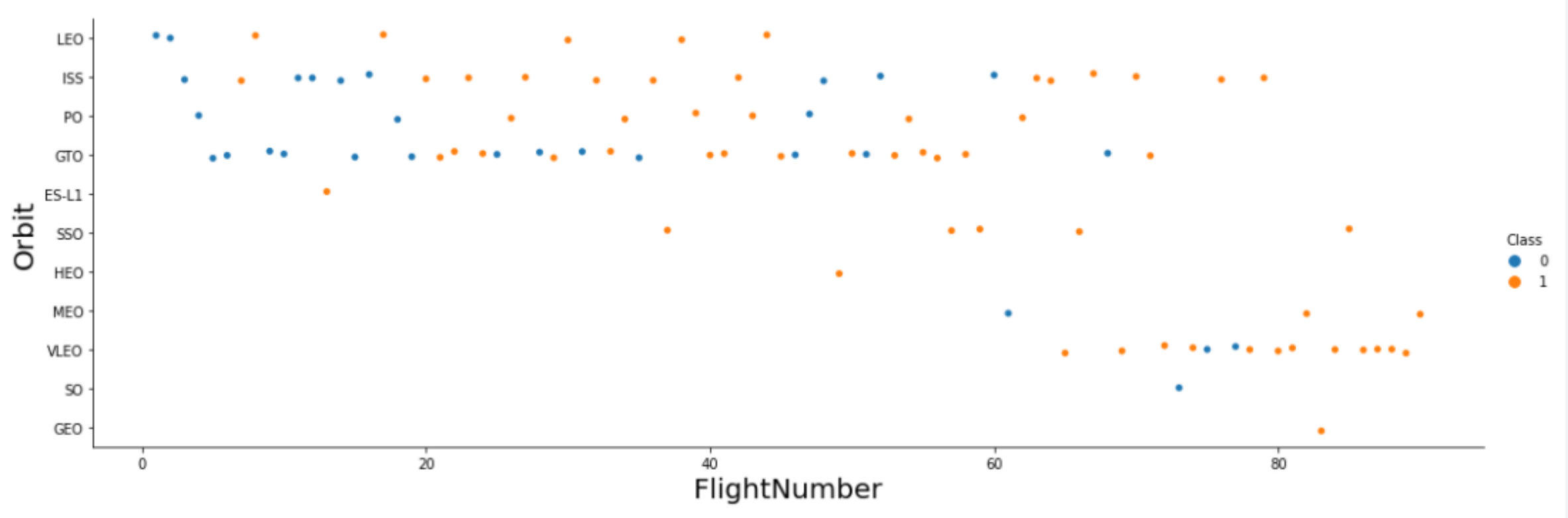
- Higher load are only launched from CCAF SLC 40 and KSC39A

Success Rate vs. Orbit Type

- ELS1, GEO, HEO and SEO has highest success rate

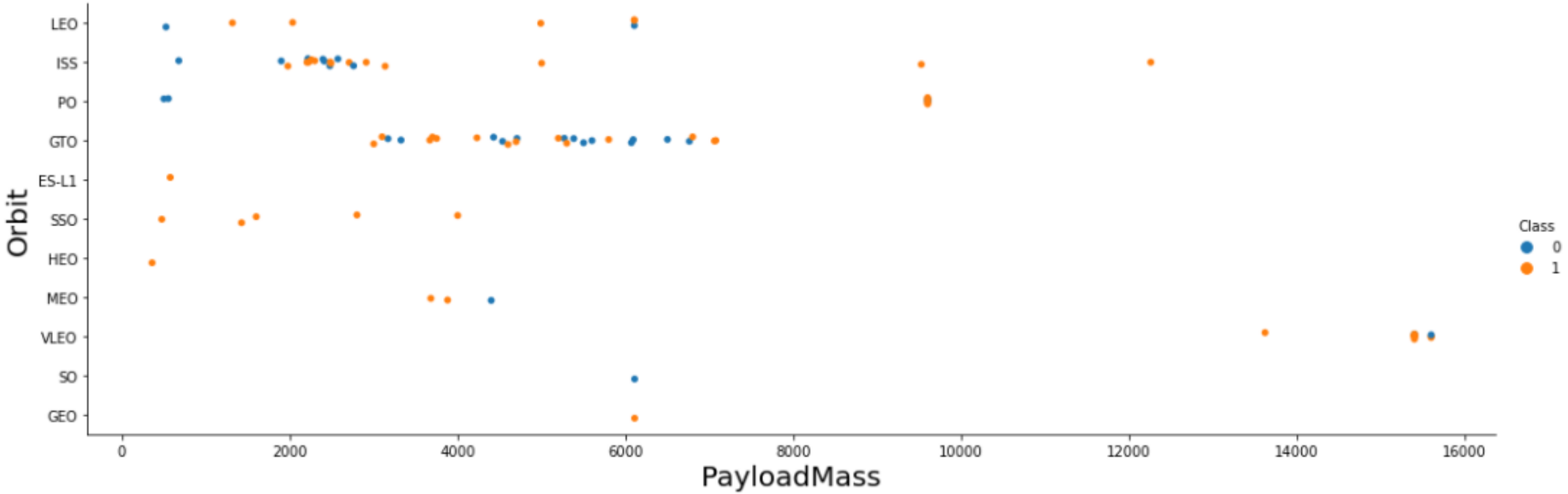


Flight Number vs. Orbit Type



- Orange indicates successful launch; blue indicates unsuccessful launch

- LEO shows more consecutive success

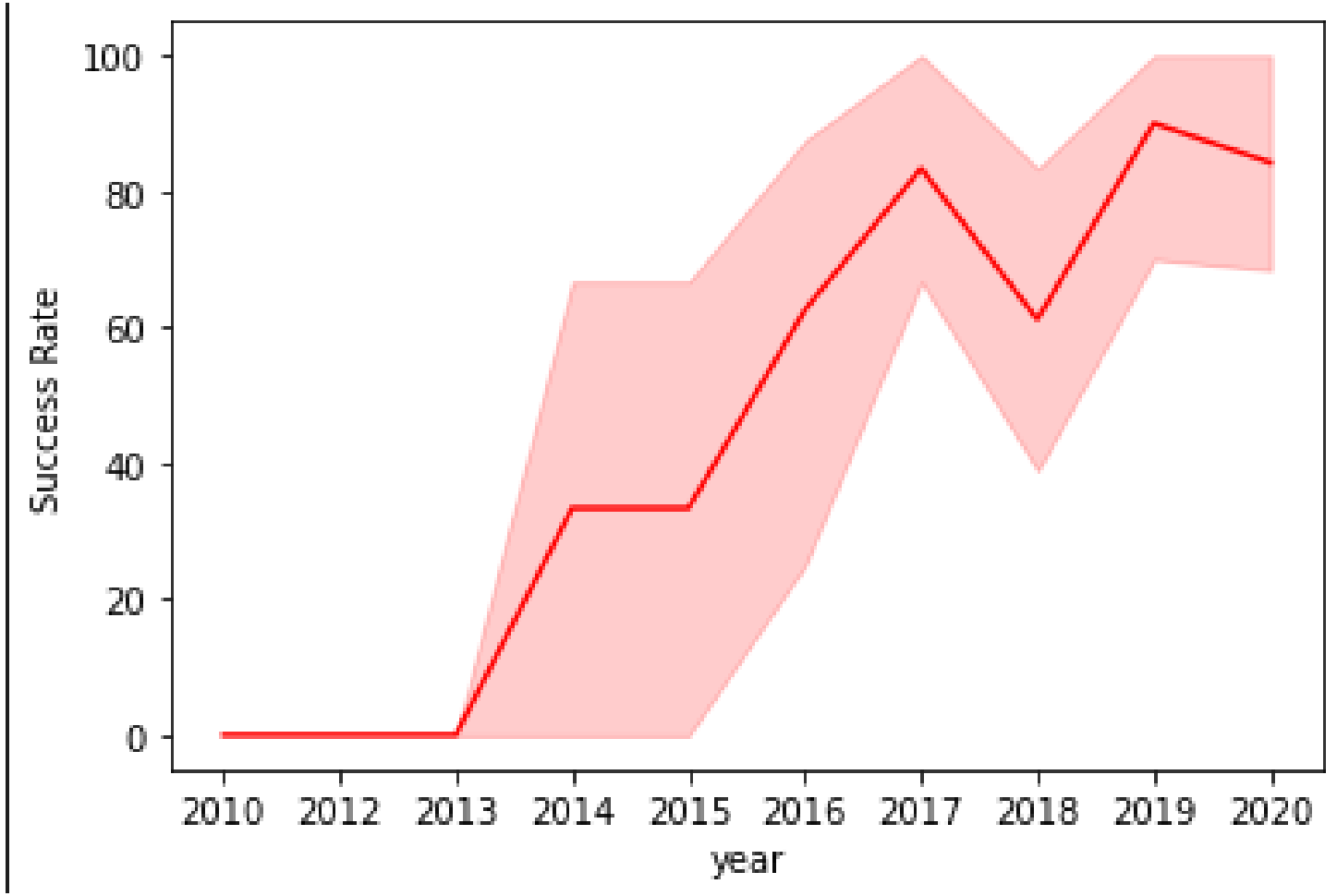


Payload vs. Orbit Type

- Orange: Successful, Blue: Unsuccessful
- There has been more unsuccessful of GTO for higher paylodmass

Launch Success Yearly Trend

- Success rate has a positive trend over time



All Launch Site Names

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

```
%sql select DISTINCT launch_site from SPACEXDATASET
```

All unique sites are returned properly

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Present your query result with a short explanation here

```
%sql select * from SPACEXDATASET where launch_site like 'CCA%' limit 5
```

```
 * ibm_db_sa://zsk66794:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

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

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here

```
%sql select sum(payload_mass__kg_) as sum from SPACEXDATASET where customer like 'NASA (CRS)'
```

```
 * ibm_db_sa://zsk66794:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases
Done.
```

SUM
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here

```
%sql select avg(payload_mass__kg_) as average from SPACEXDATASET where booster_version like 'F9 v1.1'
```

```
 * ibm_db_sa://zsk66794:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

average
2928

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here

```
%sql select min(date) as Date from SPACEXDATASET where mission_outcome like 'Success'
```

```
 * ibm_db_sa://zsk66794:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

DATE
2010-06-04

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

- Present your query result with a short explanation here

```
%sql select booster_version from SPACEXDATASET where (mission_outcome like 'Success') AND (payload_mass__kg_ BETWEEN 4000 AND 6000) AND (landing__outco
```

```
 * ibm_db_sa://zsk66794:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here

List the total number of successful and failure mission outcomes

```
%sql SELECT mission_outcome, count(*) as Count FROM SPACEXDATASET GROUP by mission_outcome ORDER BY mission_outcome
```

```
 * ibm_db_sa://zsk66794:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

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

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

```
maxm = %sql select max(payload_mass__kg_) from SPACEXDATASET
maxv = maxm[0][0]
%sql select booster_version from SPACEXDATASET where payload_mass__kg_=(select max(payload_mass__kg_) from SPACEXDATASET)
```

```
 * ibm_db_sa://zsk66794:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
 * ibm_db_sa://zsk66794:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
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

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Present your query result with a short explanation here

```
%sql select MONTHNAME(DATE) as Month, landing__out
```

```
 * ibm_db_sa://zsk66794:***@9938aec0-8105-433e-8bf
Done.
```

MONTH	landing__outcome	booster_version	launch_site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

```
%sql select landing__outcome, count(*) as count from SPACEXDATASET where Date
```

```
 * ibm_db_sa://zsk66794:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0
Done.
```

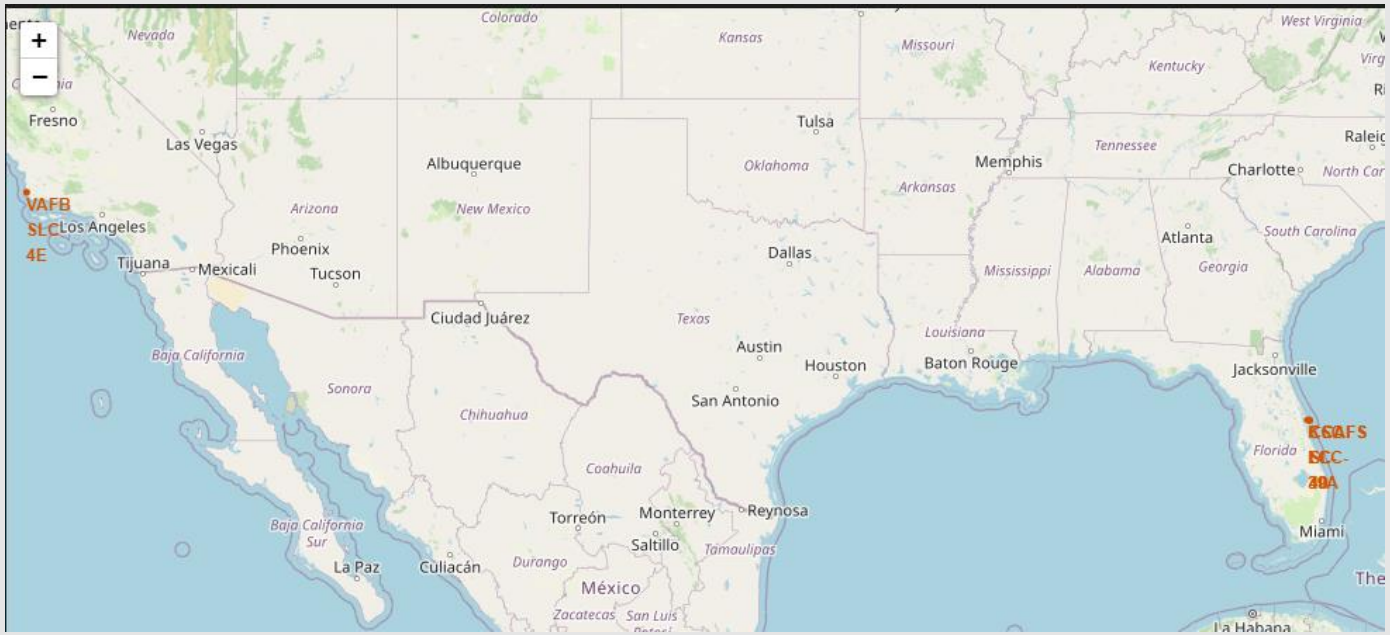
landing__outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

- 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

- Present your query result with a short explanation here


Section 3
Launch Sites
Proximities Analysis

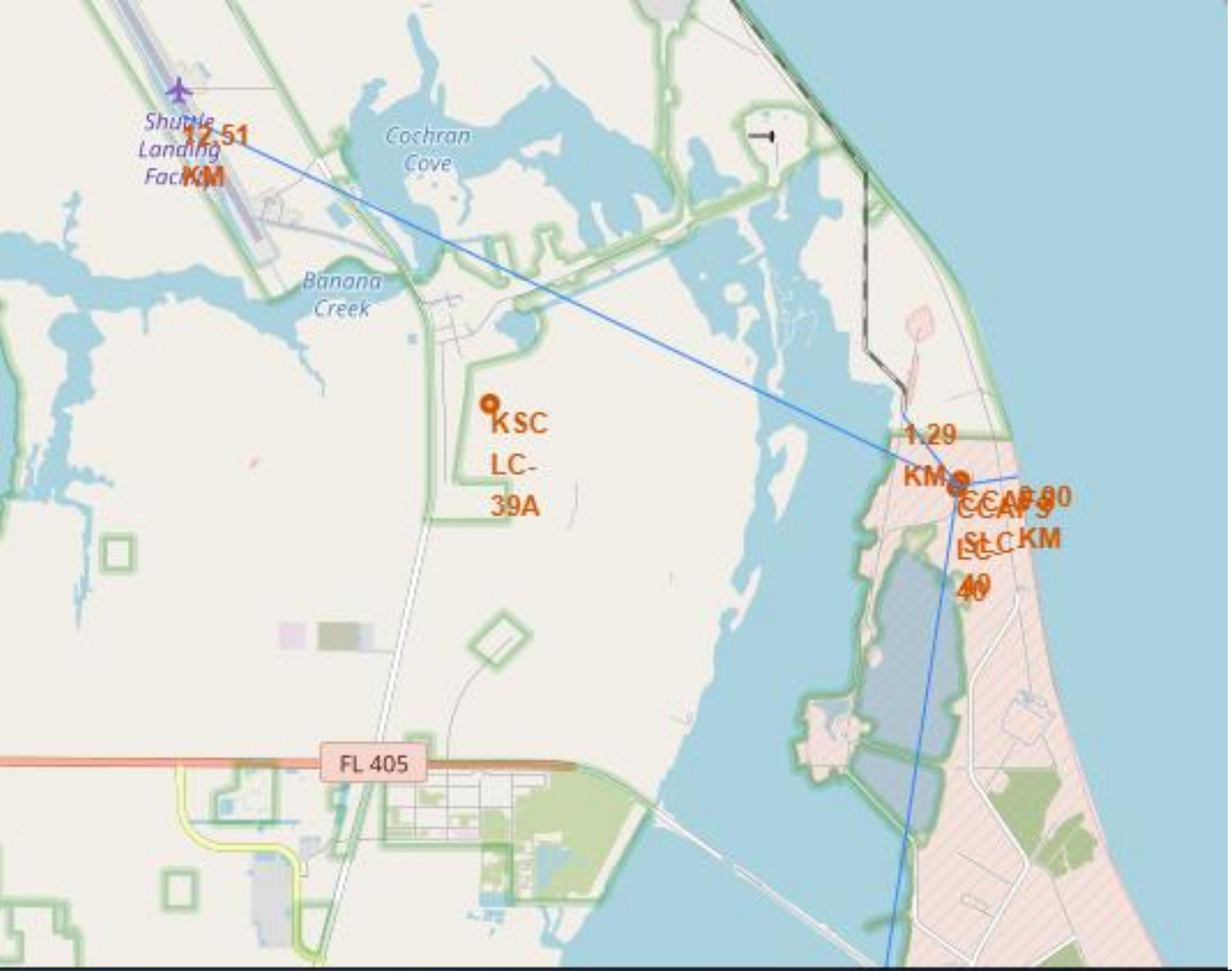
<All launch sites>



- Replace <Folium map screenshot 1> title with an appropriate title
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
- All sites are near to ocean

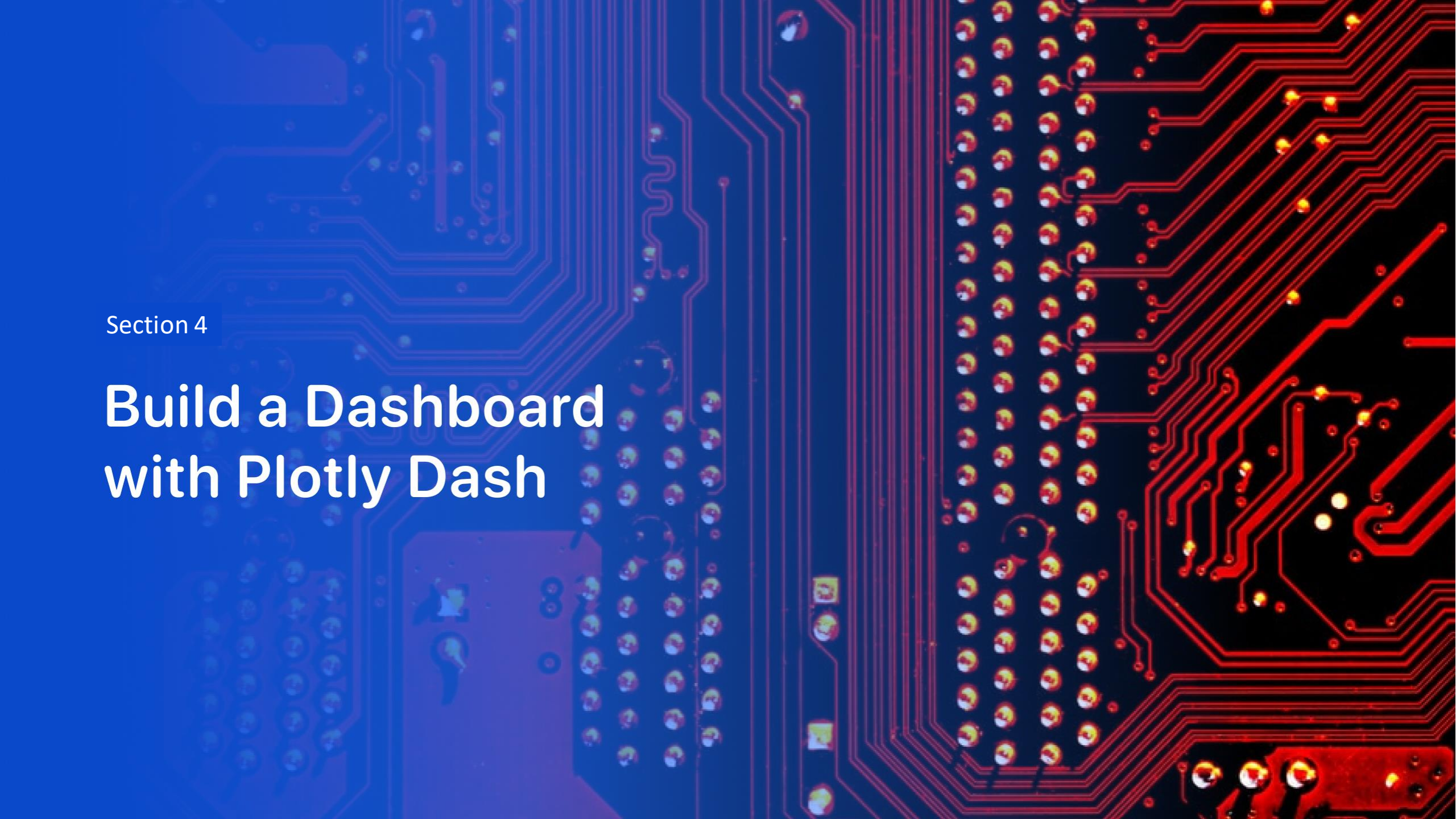
<launch outcomes (Labeled in colors)>

- Replace <Folium map screenshot 2> title with an appropriate title
- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map
- In pictures, success and failure shown.



<map showing nearby places>

- Replace <Folium map screenshot 3> title with an appropriate title
- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed
- This can be used for proper planning in case of any catastrophe

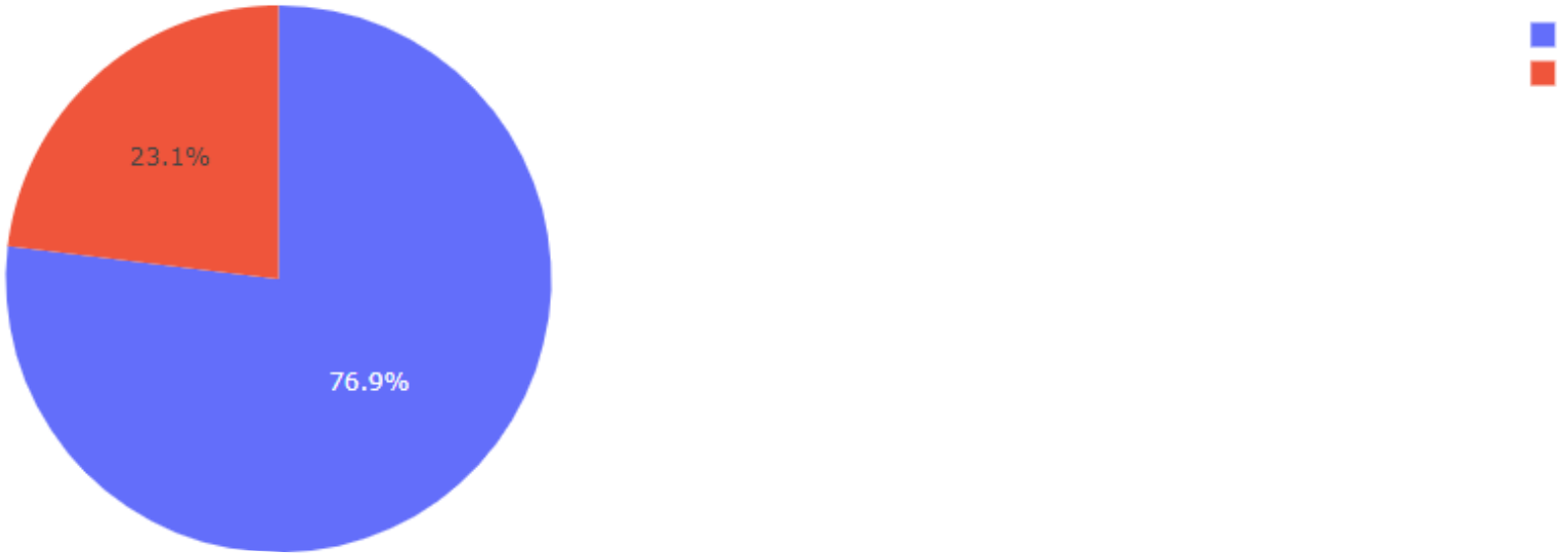

Section 4
Build a Dashboard with Plotly Dash

Total Success Launches by Site

<Success launch stats>

- Show the screenshot of launch success count for all sites, in a piechart
- KSC LC 39A launch Centre has maximum success

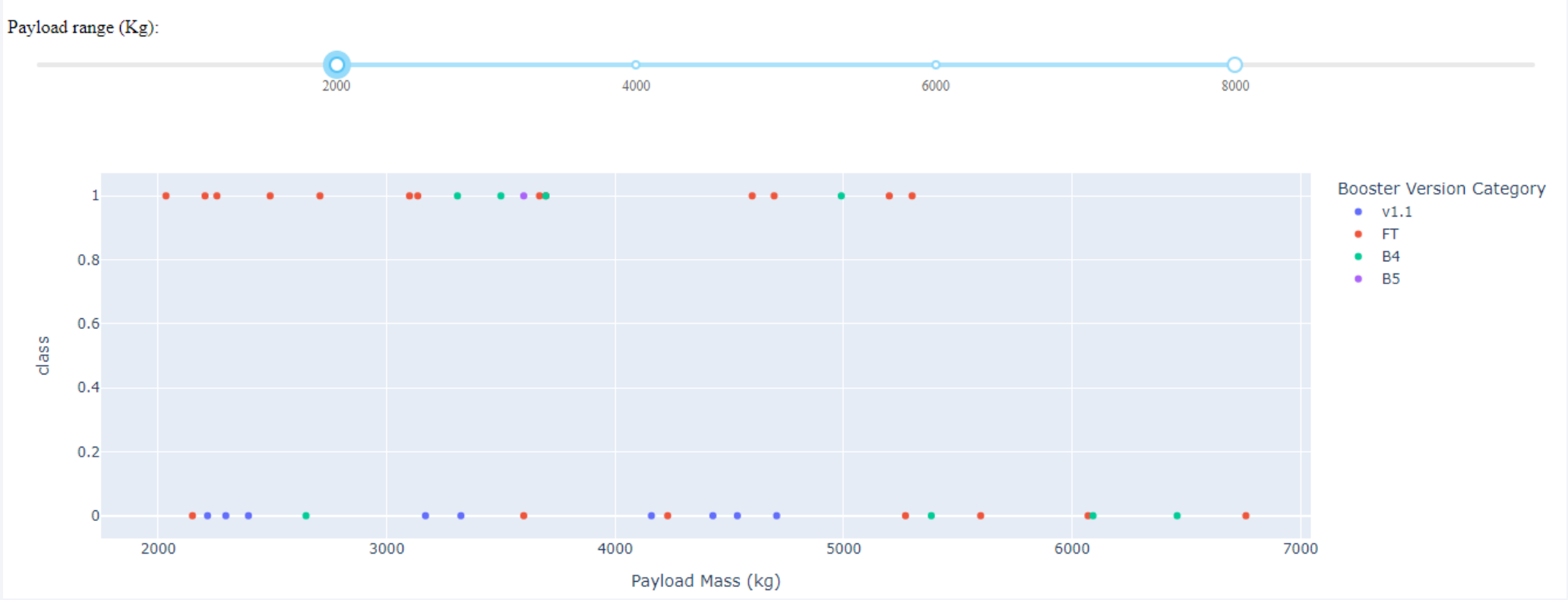
Total Success Launches for KSC LC-39A



<Success of best launching site>

- Show the screenshot of the pie chart for the launch site with highest launch success ratio
- KSC LC-39A has 77% success rate overall.

<paylod with a range slider>



- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

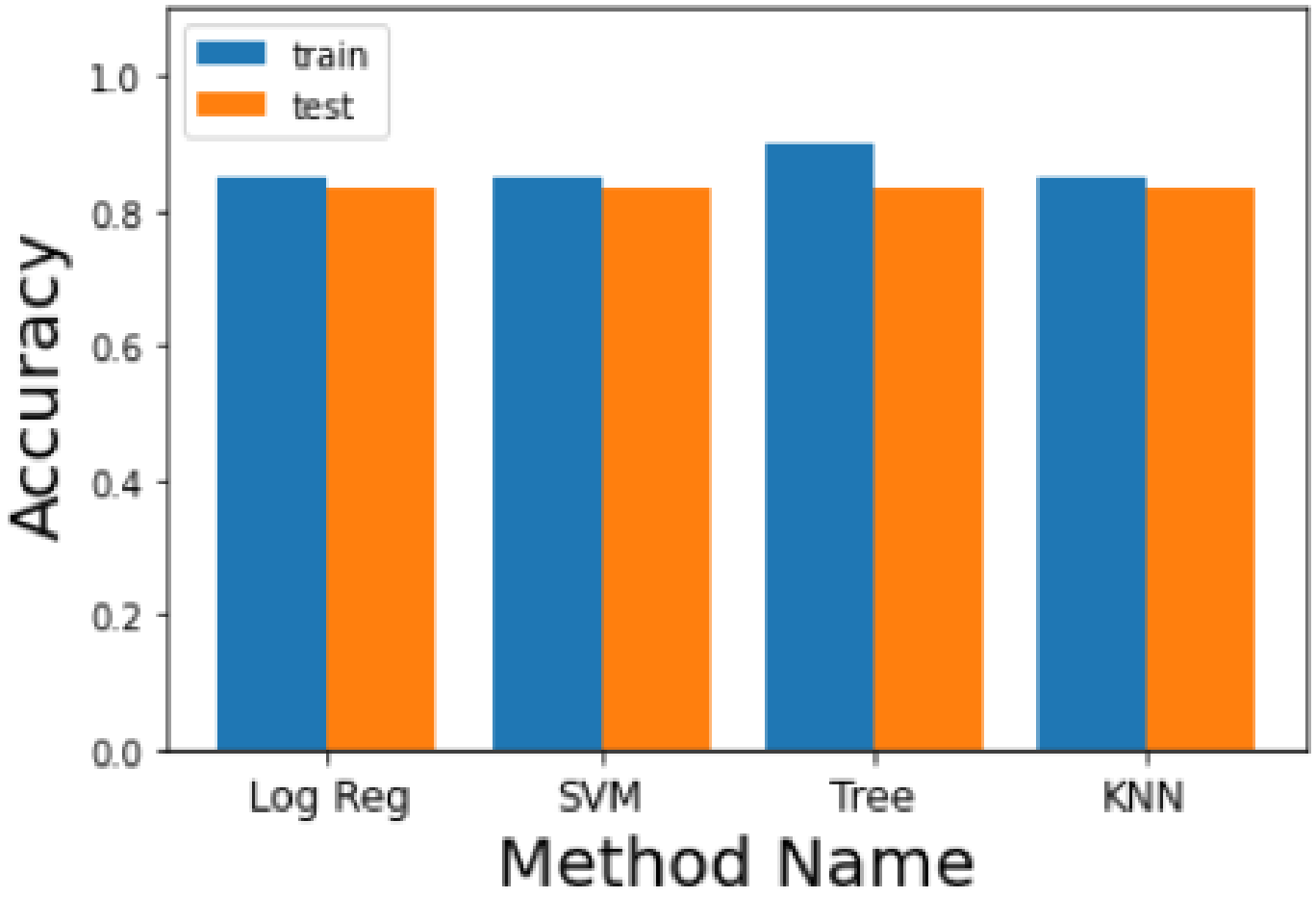
- This explain the payload and success in different range as chosen by the slider.

Section 5

Predictive Analysis (Classification)

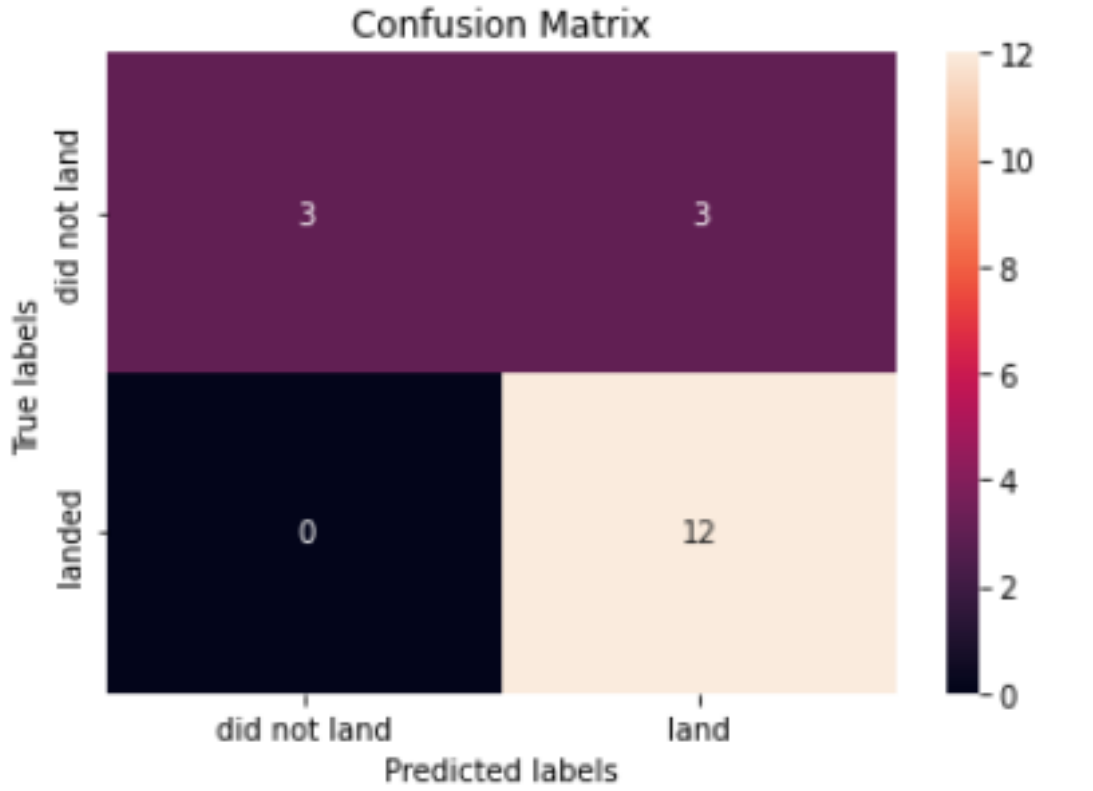
Classification Accuracy

- In training, Decision tree has better accuracy, however, when we look at the accuracy in test set, all the models are same with 83% accuracy



Confusion Matrix

```
yhat = svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



All model has same accuracy.

Conclusions

- Point 1: Data from API and webscraping are good source for data analysis
- Point 2: data can be ordered, properly shaped and preprocessed before model building
- Point 3: Data can be better understood by various kind of plotting including raw python plotting and also using folium for map plotting
- Point 4: The model predicts really good with 83 % correct. Only problem observed are false positive of 3 cases.

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!