IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ishaan Kondapalli
17th March, 2025

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection via API, Web Scraping
  - Exploratory Data Analysis (EDA) with Data Visualization
  - EDA with SQL
  - Interactive Map with Folium
  - Dashboards with Plotly Dash
  - Predictive Analysis
- Summary of all results
  - Exploratory Data Analysis results
  - Interactive maps and dashboard
  - • Predictive results

Introduction

- Project background and context
  - The aim of the project is to predict if the Falcon 9 rocket will land successfully or not. SpaceX makes Falcon cheaper compared to its competitors because it can reuse its rocket’s first stage. By determining if the first stage will land or not we can determine its cost.
- Problems you want to find answers
  - What are the characteristics of a failed or a successful launch?
  - What’re the conditions that will allow SpaceX to have the best landing success rate?

Section 1

Methodology

Methodology

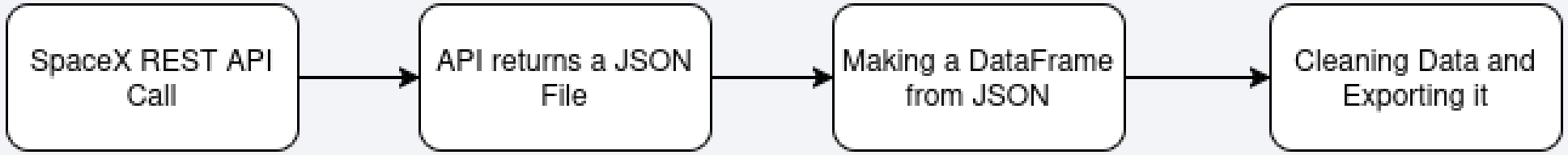
Executive Summary

- Data collection methodology:
  - SpaceX REST API (BeautifulSoup)
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Data Preprocessing and Cleaning NaN Values
  - One Hot Encoding for Classification Models
- 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

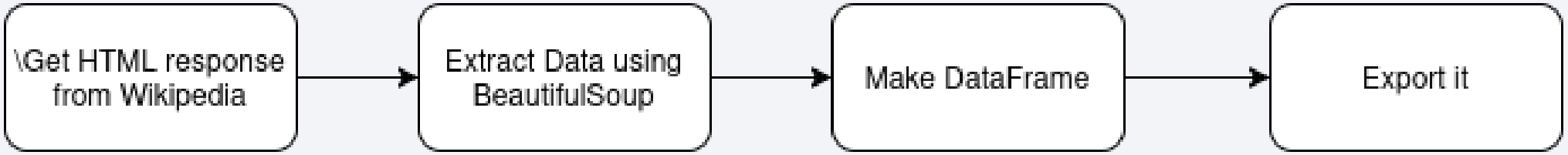
Data Collection

Datasets are collected from SpaceX REST API and webscrapping Wikipedia

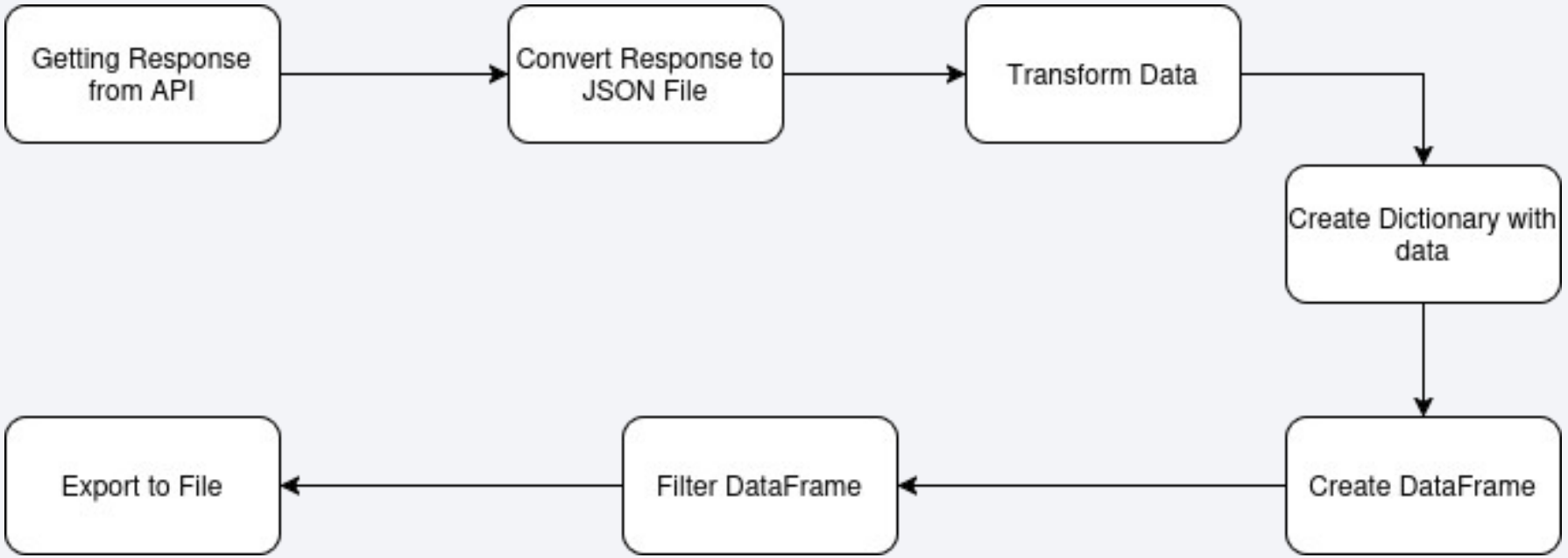
- The information obtained by the API are rocket, launches and payload information
  - The Space X REST API URL is api.spacexdata.com/v4/



- The information obtained by web scrapping of Wikipedia are launches, landing, payload information
  - URL is https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922



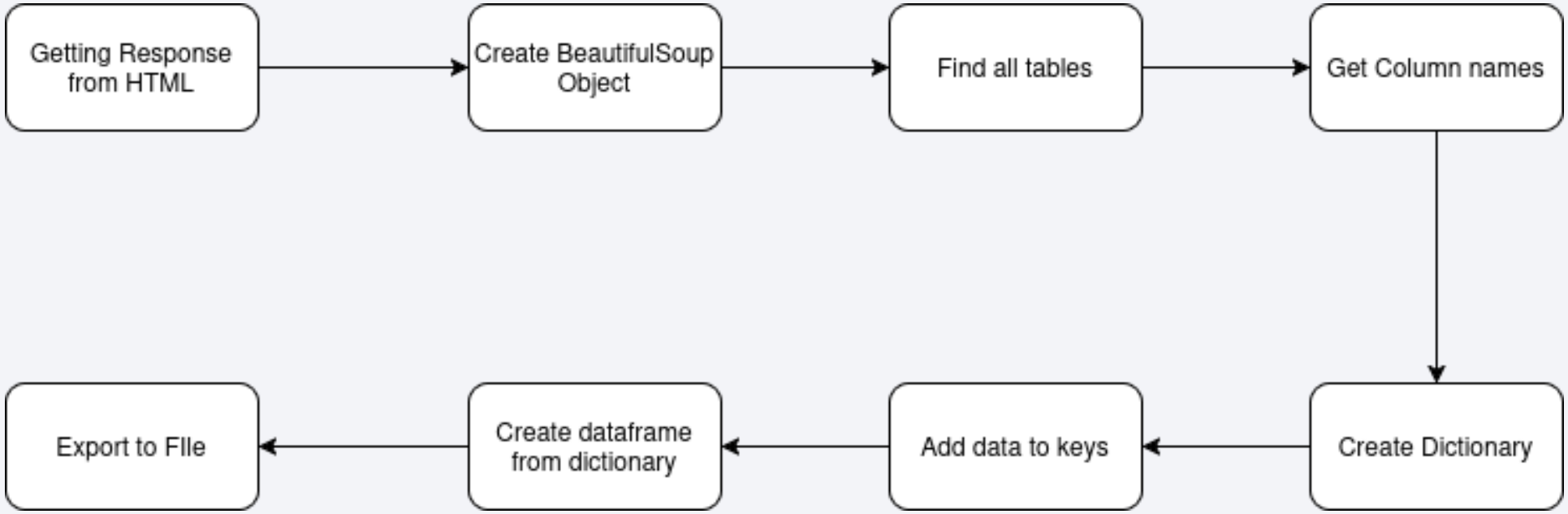
Data Collection – SpaceX API



The notebook has been linked below

Link to Code

Data Collection - Scraping

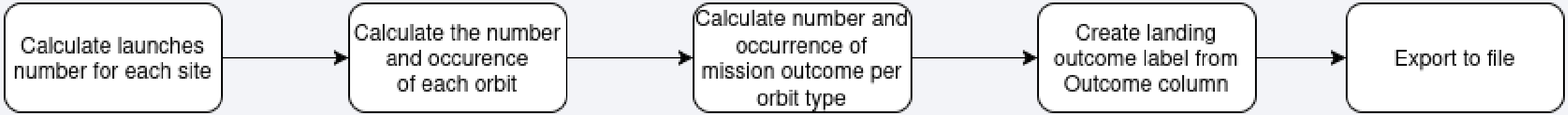


The notebook has been linked below

Link to Code

Data Wrangling

- In the dataset, there are several cases where the booster did not land successully.
  - True Ocean, True RTLS, True ASDS means the mission has been successful.
  - False Ocean, False RTLS, False ASDS means the mission was a failure.
- We need to transform string variables into categorical variables where 1 means the mission has been successful and 0 means the mission was a failure.

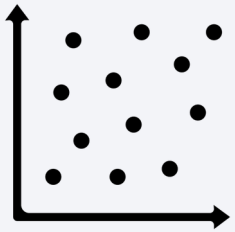


The notebook has been linked below

Link to Code

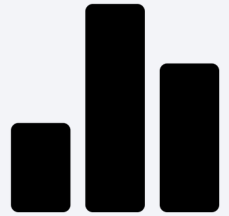
EDA with Data Visualization

- Scatter Graphs
  - Flight Number vs. Payload Mass
  - Flight Number vs. Launch Site
  - Payload vs. Launch Site
  - Orbit vs. Flight Number
  - Payload vs. Orbit Type
  - Orbit vs. Payload Mass



Scatter plots show relationship between variables. This relationship is called the correlation.

- Bar Graph
  - Success rate vs. Orbit



Bar graphs show the relationship between numeric and categoric variables.

- Line Graph
  - Success rate vs. Year

Line graphs show data variables and their trends.
Line graphs can help to show global behavior and make prediction for unseen data.

Link to Code

EDA with SQL

- We performed SQL queries to gather and understand data from dataset:
  - •Displaying the names of the unique lauunch sites in the space mission.
  - • Display 5 records where launch sites begin with the string 'CCA'
  - • Display the taotal 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 achieved.
  - • 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.
  - • List the records which will display the month names, faiilure landing_ouutcomes in drone ship, booster versions, launch_site for the months in year 2015.
  - • Rank the count of successful landiing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

Link to Code

Build an Interactive Map with Folium

- Folium map object is a map centered on NASA Johnson Space Center at Houson, Texas

  • Red circle at NASA Johnson Space Center's coordinate with label showing its name (folium.Circle, folium.map.Marker).

  • Red circles at each launch site coordinates with label showing launch site name (folium.Circle, folium.map.Marker,

  folium.features.DivIcon).

  • The grouping of points in a cluster to display multiple and different information for the same coordinates

  (folium.plugins.MarkerCluster).

  • Markers to show successful and unsuccessful landings. Green for successful landing and Red for unsuccessful landing.

  (folium.map.Marker, folium.Icon).

  • Markers to show distance between launch site to key locations (railway, highway, coastway, city) and plot a line between them .

  (folium.map.Marker, folium.PolyLine, folium.features.DivIcon)

- These objects are created in order to understand better the problem and the data. We can show easily all launch sites, their surroundings and the number of successful and unsuccessful landings.

Link to Code

Build a Dashboard with Plotly Dash

- Dashboard has dropdown, pie chart, rangeslider and scatter plot components
  - Dropdown allows a user to choose the launch site or all launch sites

    (dash_core_components.Dropdown).
  - Pie chart shows the total success and the total failure for the launch site chosen with the

    dropdown component (plotly.express.pie).
  - Rangeslider allows a user to select a payload mass in a fixed range

    (dash_core_components.RangeSlider) .
  - Scatter chart shows the relationship between two variables, in particular Success vs

    Payload Mass (plotly.express.scatter) .

Link to Code

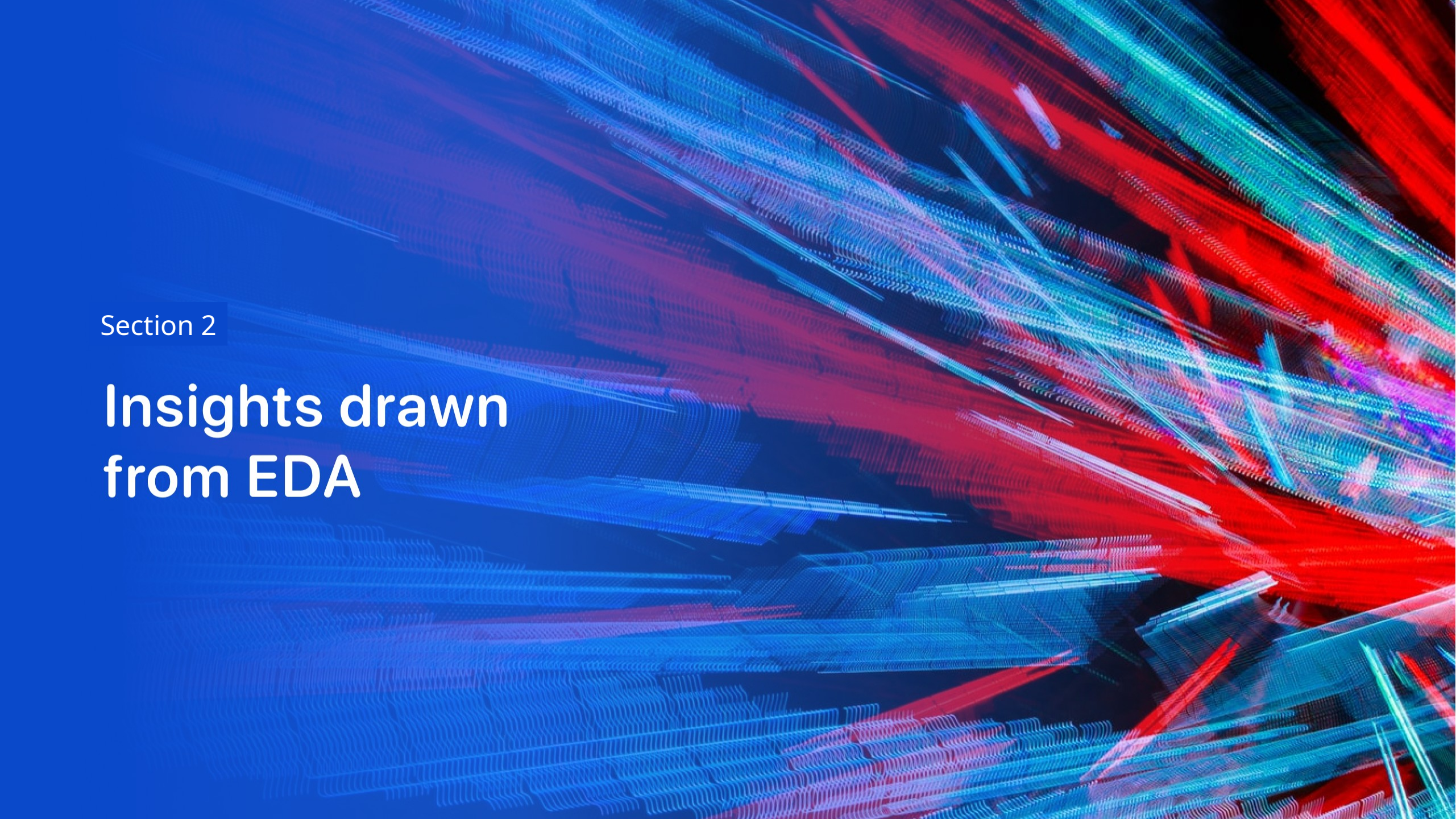
Predictive Analysis (Classification)

- Data preparation
  - Load dataset
  - Normalize data
  - Split data into training and test sets.
- Model preparation
  - Selection of machine learning algorithms
  - Set parameters for each algorithm to GridSearchCV
  - Training GridSearchModel models with training dataset
- Model evaluation
  - Get best hyperparameters for each type of model
  - Compute accuracy for each model with test dataset
  - Plot Confusion Matrix
- Model comparison
  - Comparison of models according to their accuracy
  - The model with the best accuracy will be chosen (see Notebook for result)

Link to Code

Results

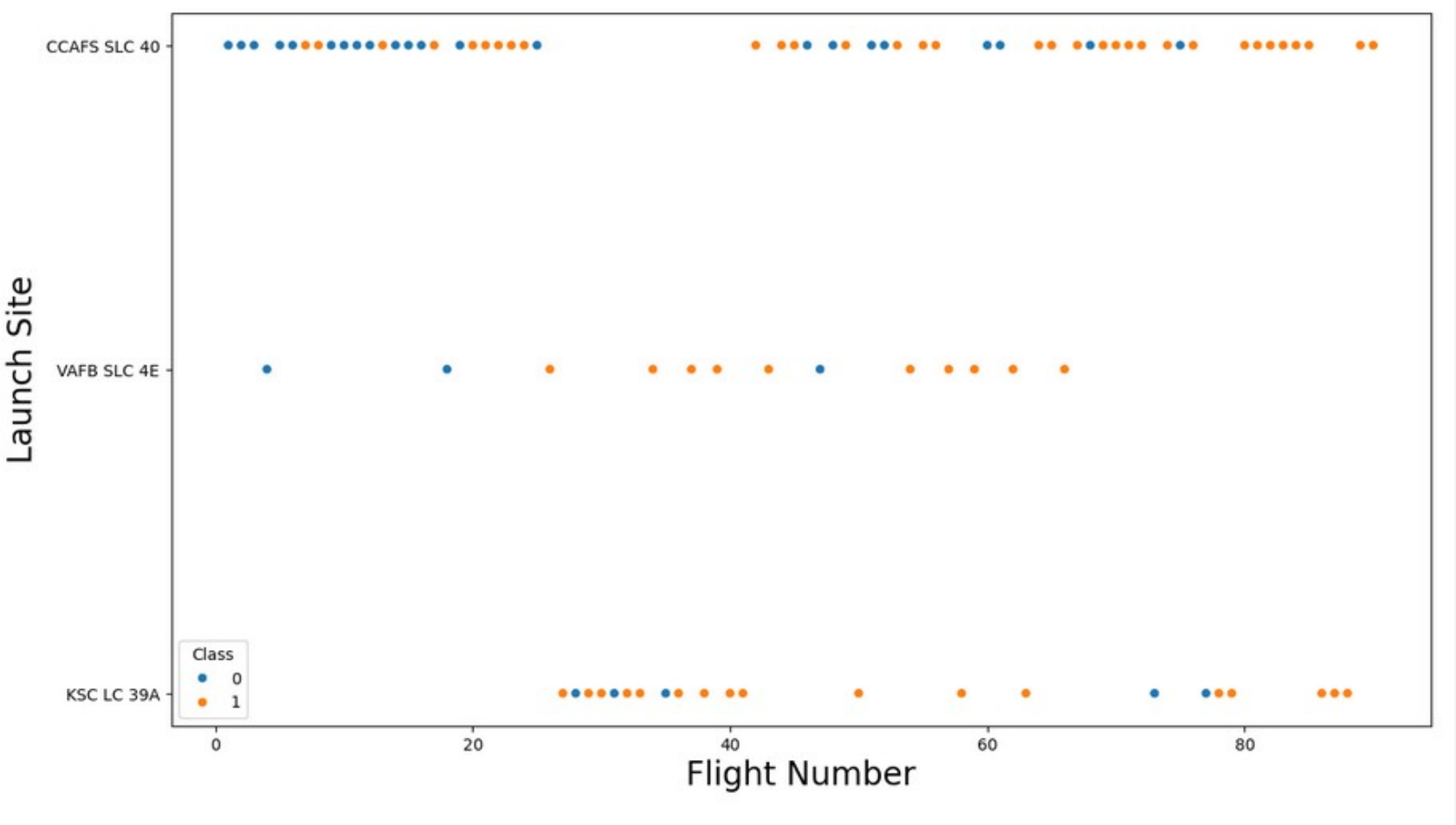
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results



Section 2

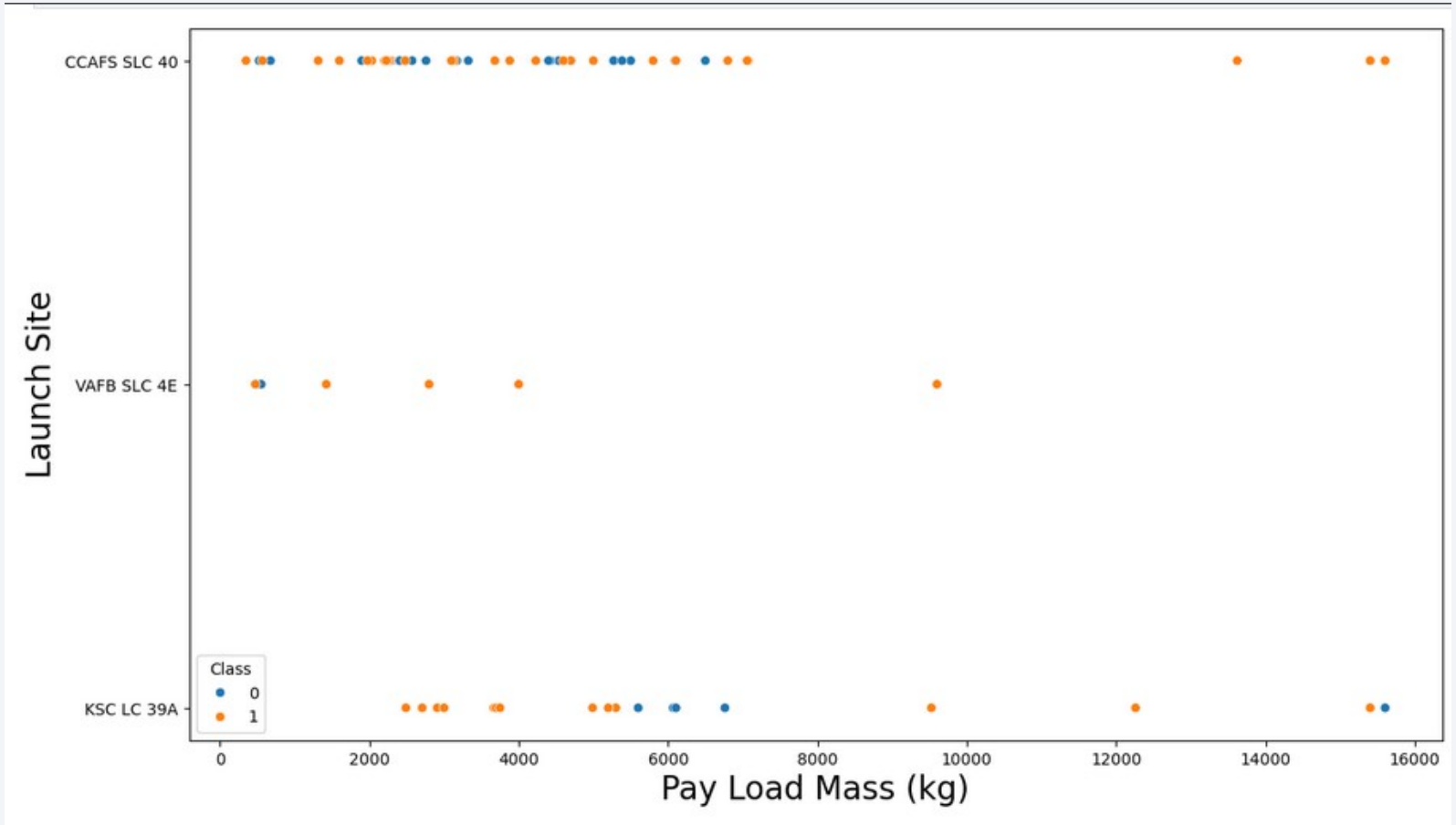
Insights drawn from EDA

Flight Number vs. Launch Site



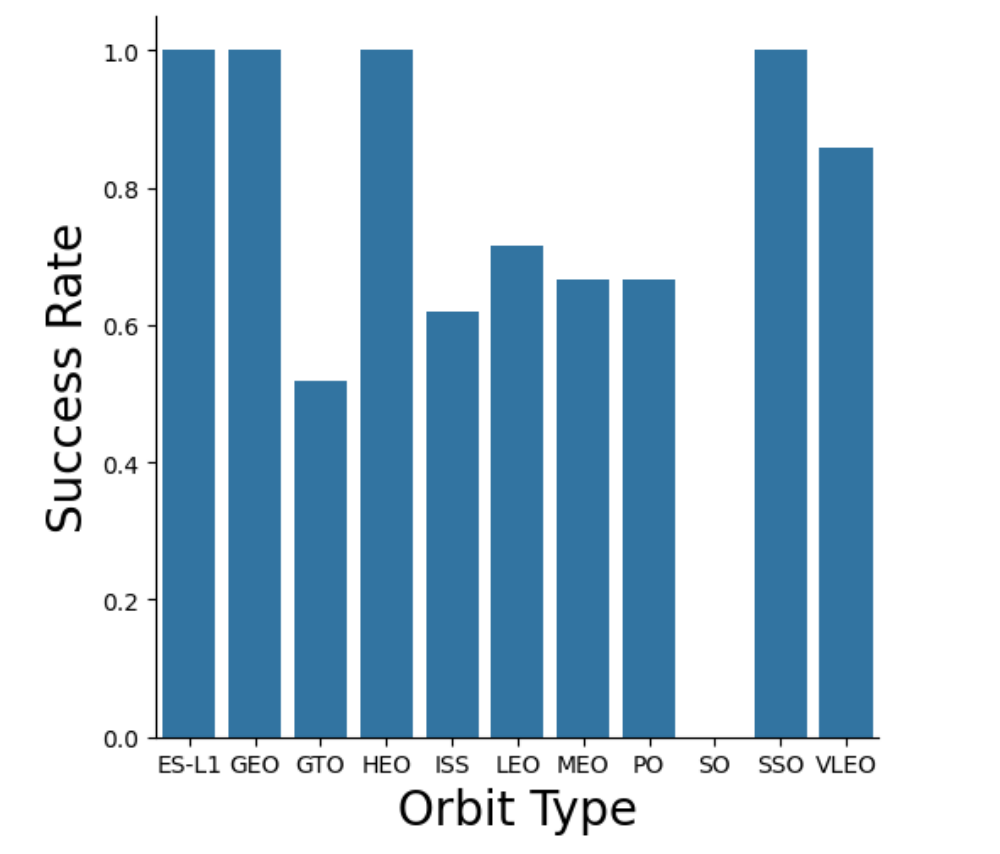
We observe that, for each site, the success rate is increasing.

Payload vs. Launch Site



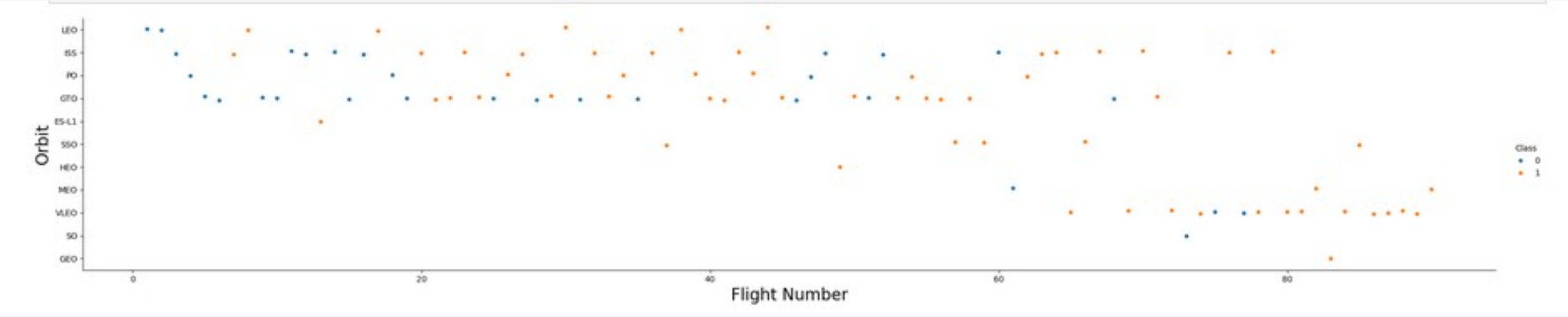
Depending on the launch site, a heavier payload may be a consideration for a successful landing. On the other hand, a too heavy payload can make a landing fail.

Success Rate vs. Orbit Type



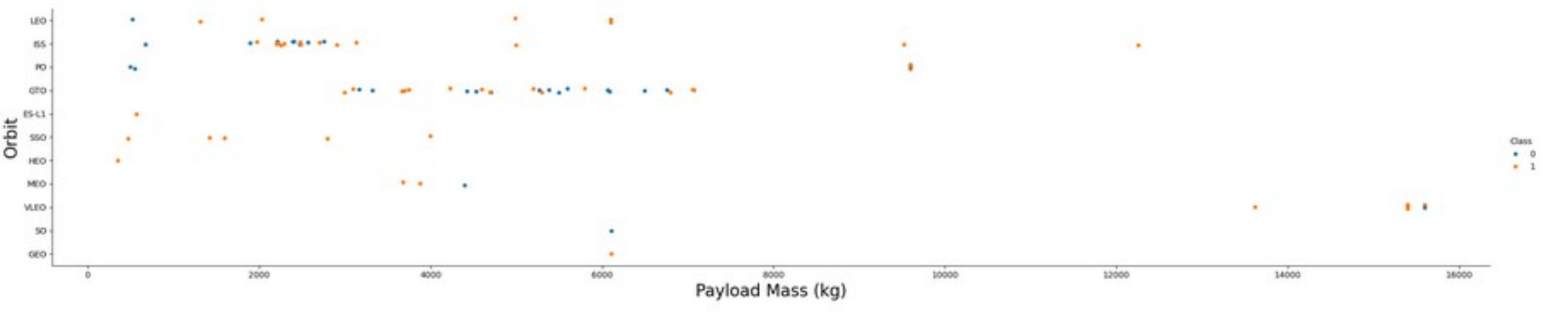
With this plot, we can see success rate for different orbit types. We note that ES-L1, GEO, HEO, SSO have the
best success rate.

Flight Number vs. Orbit Type



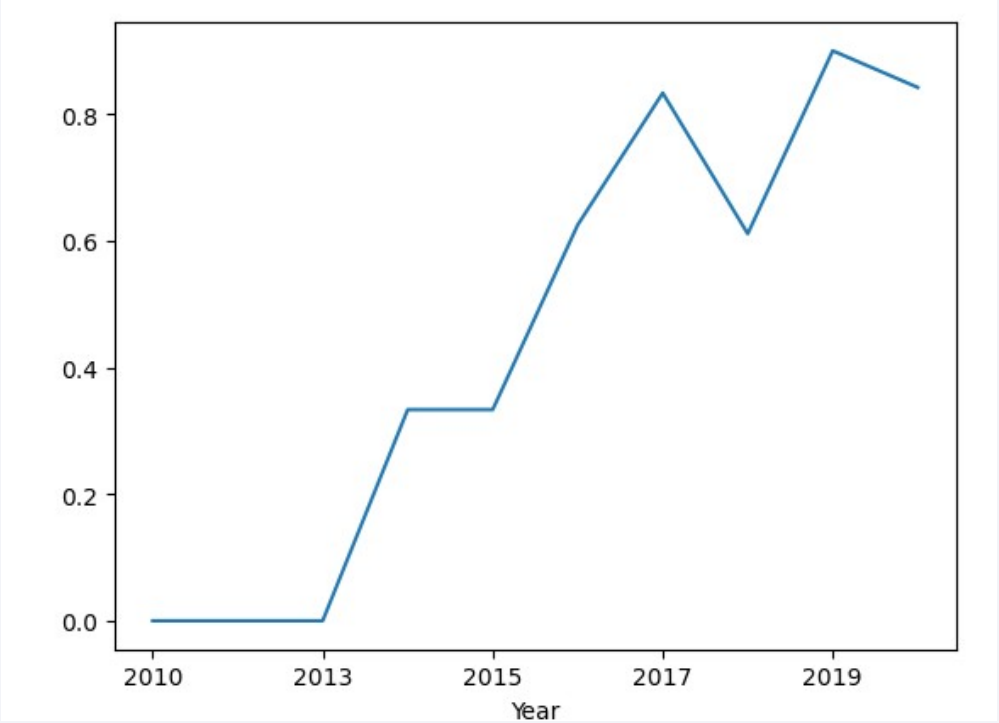
You can observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.

Payload vs. Orbit Type



The weight of the payloads can have a great influence on the success rate of the launches in certain
orbits. For example, heavier payloads improve the success rate for the LEO orbit. Another finding is that
decreasing the payload weight for a GTO orbit improves the success of a launch.

Launch Success Yearly Trend



you can observe that the success rate since 2013 kept increasing till 2020

All Launch Site Names

SQL QUERY:

```
%sql select distinct launch_site from SPACEXTABLE
```

OUTPUT:

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

Explanation:

DISTINCT here helps in finding only unique names in the launch sites

Launch Site Names Begin with 'CCA'

SQL QUERY:

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

OUTPUT:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_O
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (pa
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 (pa
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No

Explanation:

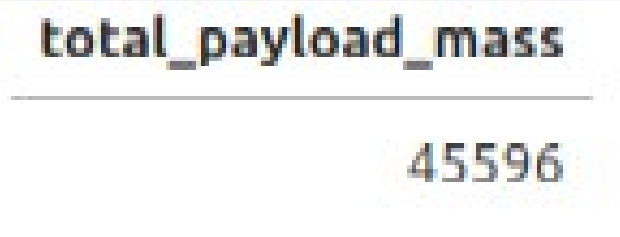
The WHERE clause followed by LIKE filters it to strings only with CCA and LIMIT 5 only shows 5 items

Total Payload Mass

SQL QUERY:

```
%sql select sum(PAYLOAD_MASS__KG_) as total_payload_mass from SPACEXTABLE where customer == 'NASA (CRS)'
```

OUTPUT:



total_payload_mass
45596

Explanation:

The SUM function computes the sum of all items in the column and returns it. And where clause limits to only NASA (CRS).

Average Payload Mass by F9 v1.1

SQL QUERY:

```
%sql select avg(PAYLOAD_MASS__KG_) as average_payload_mass from SPACEXTABLE where Booster_Version like '%F9 v1.1%
```

OUTPUT:

average_payload_mass
2534.6666666666665

Explanation:

The AVG function returns the average of all the items in the column and where clause limits it only to F9 v1.1

First Successful Ground Landing Date

SQL QUERY:

```
%sql select min(date) as first_succesful_landing from SPACEXTABLE where landing_outcome like '%Success%'
```

OUTPUT:

first_succesful_landing
2015-12-22

Explanation:

The MIN function returns the smallest value in the column. Where clause finds the landing outcomes that were successful

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL QUERY:

%sql select booster_version from SPACEXTABLE where landing_outcome == 'Success (drone ship)' and payload_mass__kg_ between 4000 and 6000

OUTPUT:

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

Explanation:

BETWEEN is used to take values for the two values.

Total Number of Successful and Failure Mission Outcomes

SQL QUERY:

```
%sql select mission_outcome, count(*) as total from SPACEXTABLE group by mission_outcome
```

OUTPUT:

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

Explanation:

COUNT returns the no of values in a column or an item. GROUP BY divides based on the values of that column.

Boosters Carried Maximum Payload

SQL QUERY:

%sql select booster_version from SPACEXTABLE where payload_mass__kg_ == (select max(payload_mass__kg_) from SPACEXTABLE)

OUTPUT:

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

Explanation:

MAX function here is used to return the highest value in that column. That is taken from a sub query and returned to the main query.

2015 Launch Records

SQL QUERY:

%sql select substr(Date,6,2) as month, date, booster_version, launch_site, landing_outcome from SPACEXTABLE where landing_outcome == 'Failure (drone ship)' and substr(Date,0,5) == '2015'

OUTPUT:

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

Explanation:

This query returns month, booster version, launch site where landing was unsuccessful and landing date took place in 2015. Substr function process date in order to take month or year. Substr(DATE, 4, 2) shows month. Substr(DATE,7, 4) shows year.

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

SQL QUERY:

%sql select landing_outcome, count(*) as count_outcomes from SPACEXTABLE where date between '2010-06-04' and '2017-03-20' group by landing_outcome order by count_outcomes desc;

OUTPUT:

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

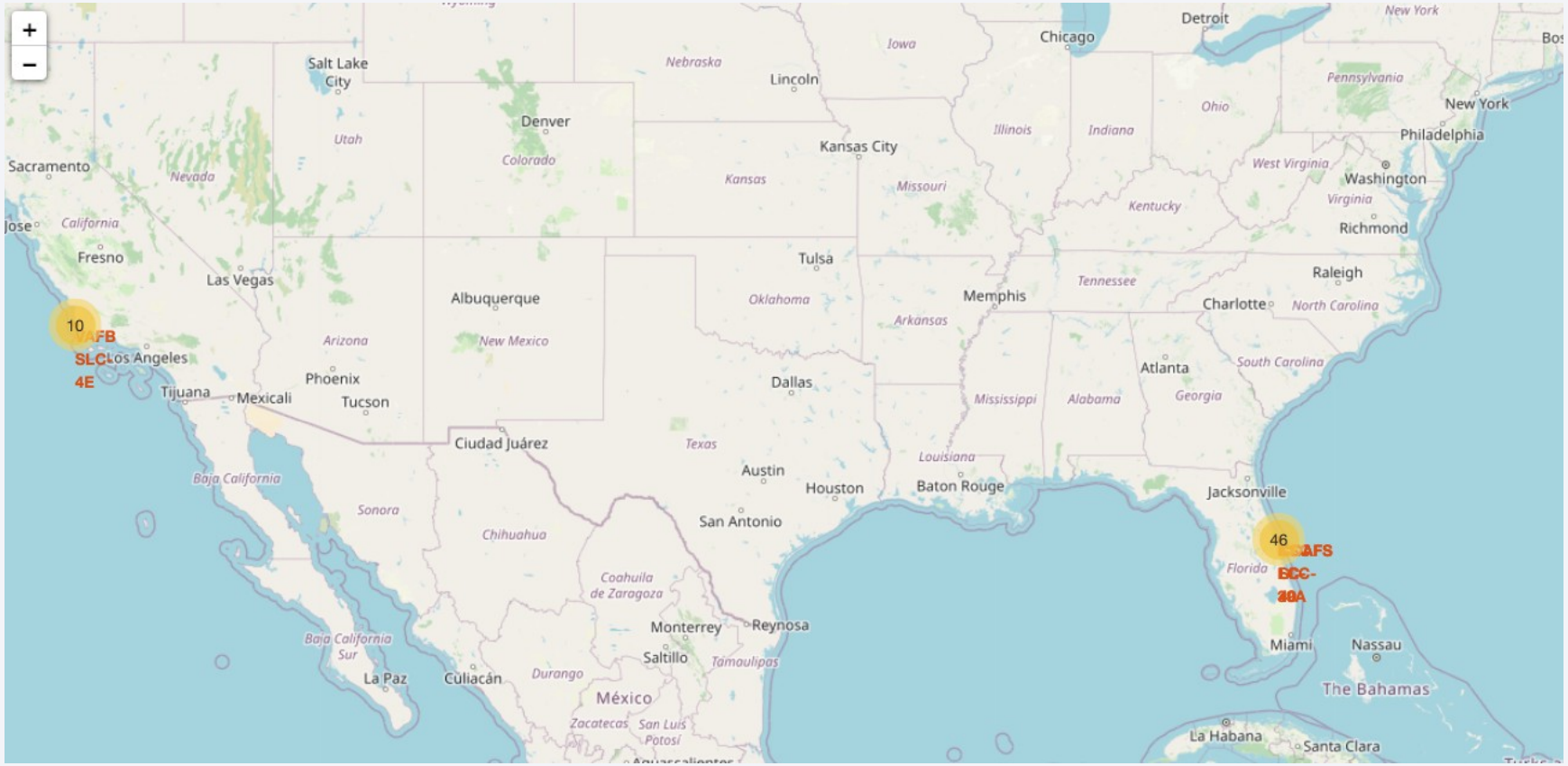
Explanation:

This query returns landing outcomes and their count where mission was successful and date is between 04/06/2010 and 20/03/2017. The GROUPBY clause groups results by landing outcome and ORDER BY COUNT DESC shows results in decreasing order.

Section 3

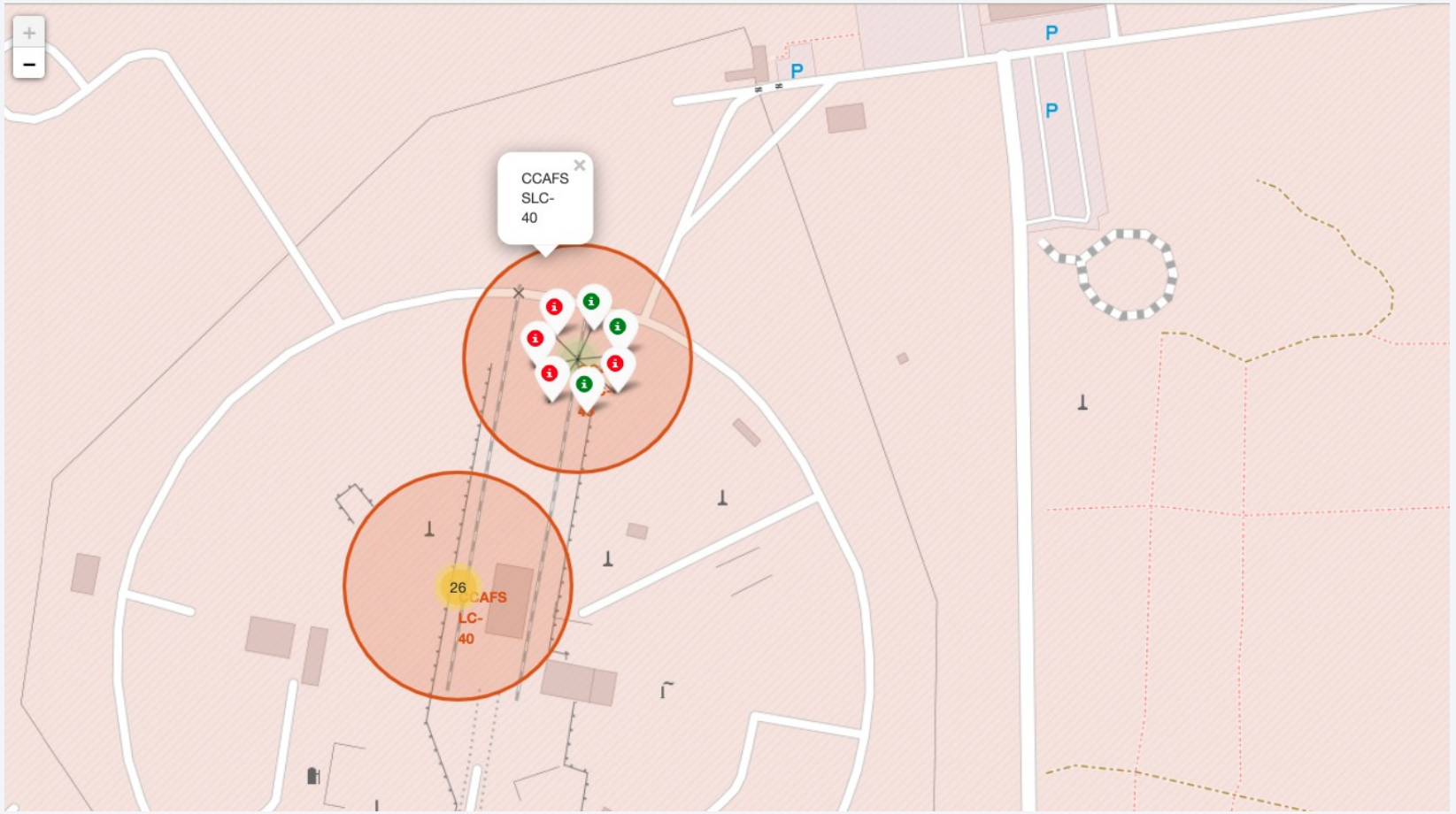
Launch Sites Proximities Analysis

Folium map – Ground stations



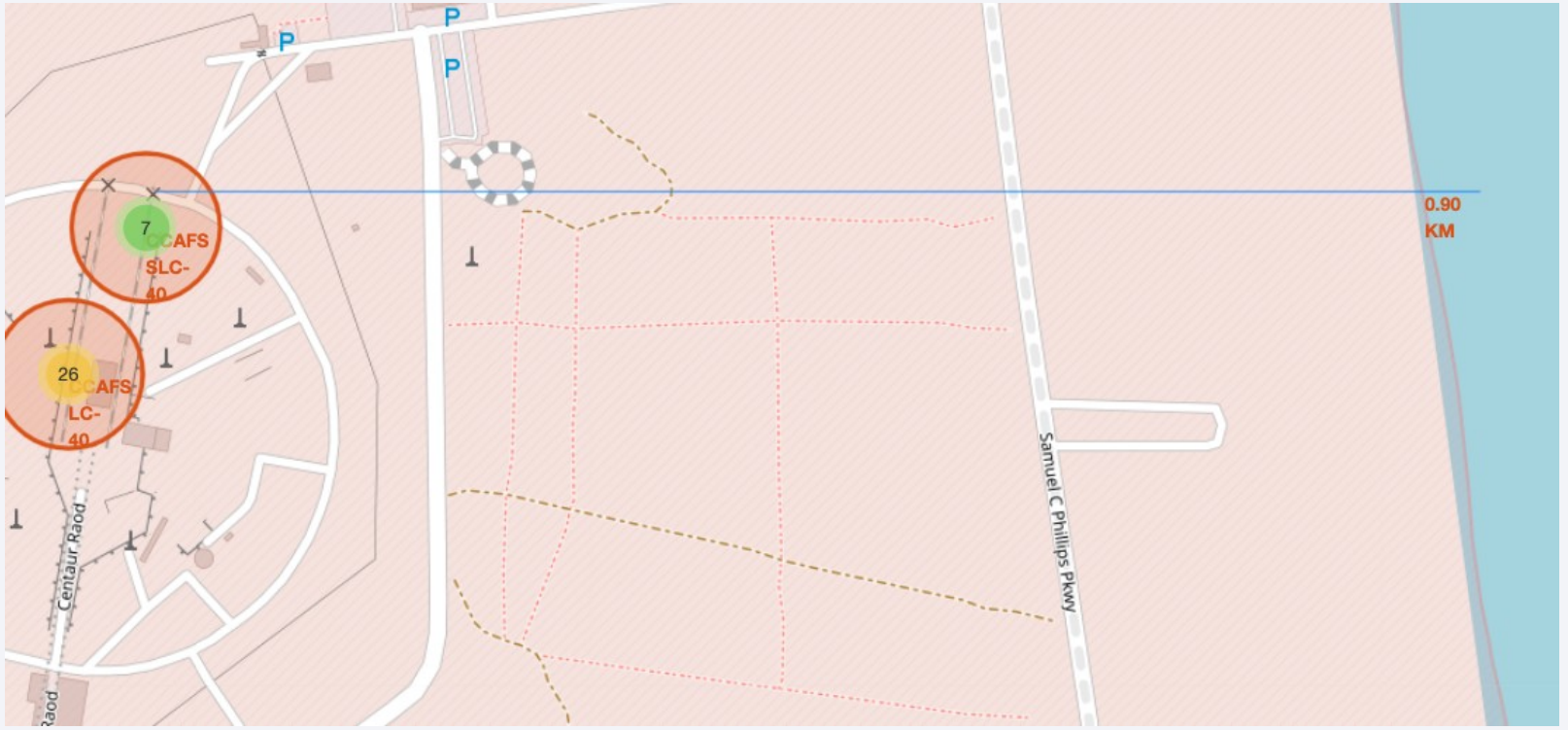
We see that Space X launch sites are located on the coast of the United States

Folium map – Color Labeled Markers

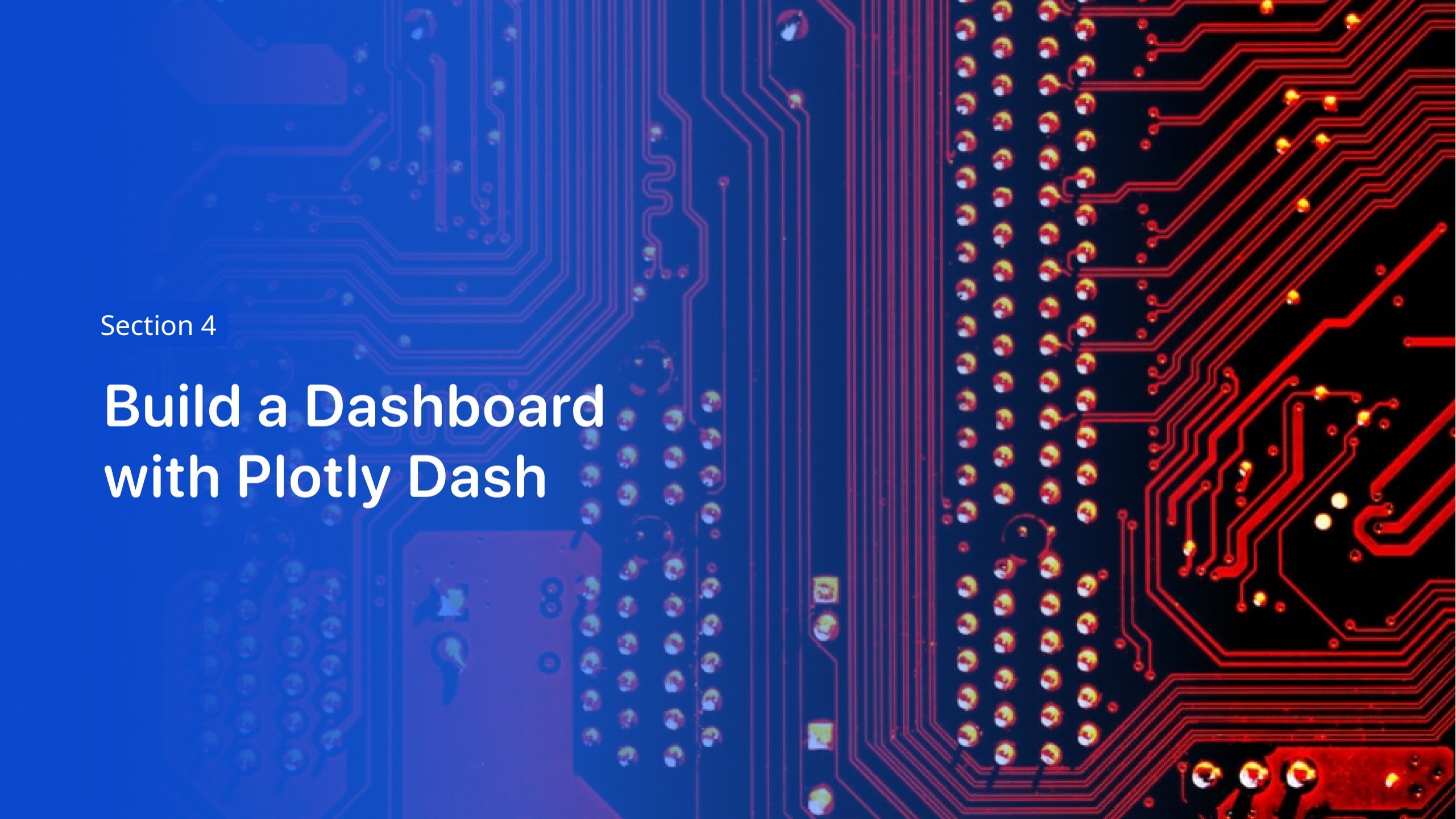


Green marker represents successful launches. Red marker represents unsuccessful launches. We
note that KSC LC-39A has a higher launch success rate.

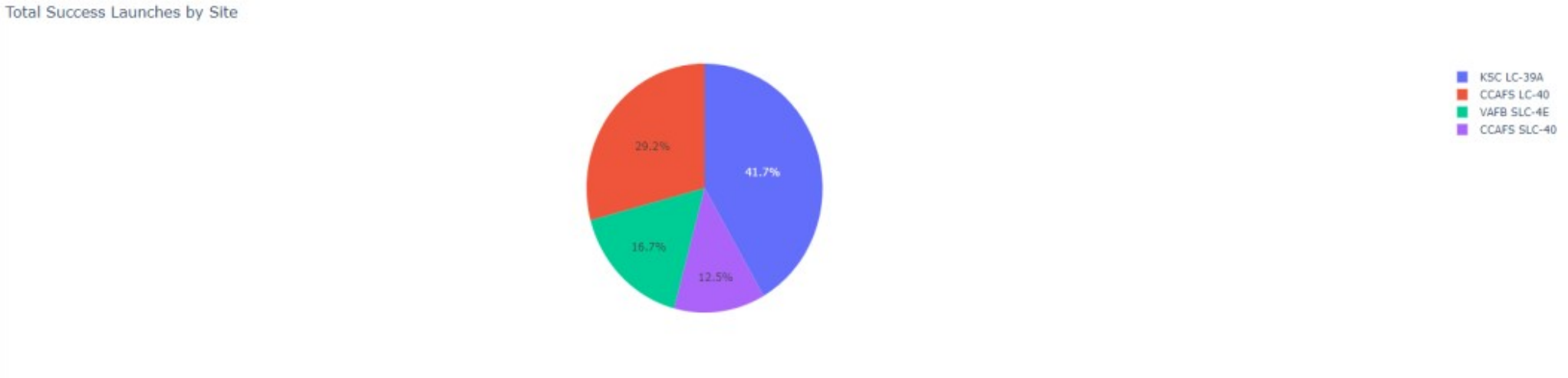
Folium Map – Distances between CCAFS SLC-40 and its proximities



Is CCAFS SLC-40 in close proximity to railways ? Yes
Is CCAFS SLC-40 in close proximity to highways ? Yes
Is CCAFS SLC-40 in close proximity to coastline ? Yes
Do CCAFS SLC-40 keeps certain distance away from cities ? No

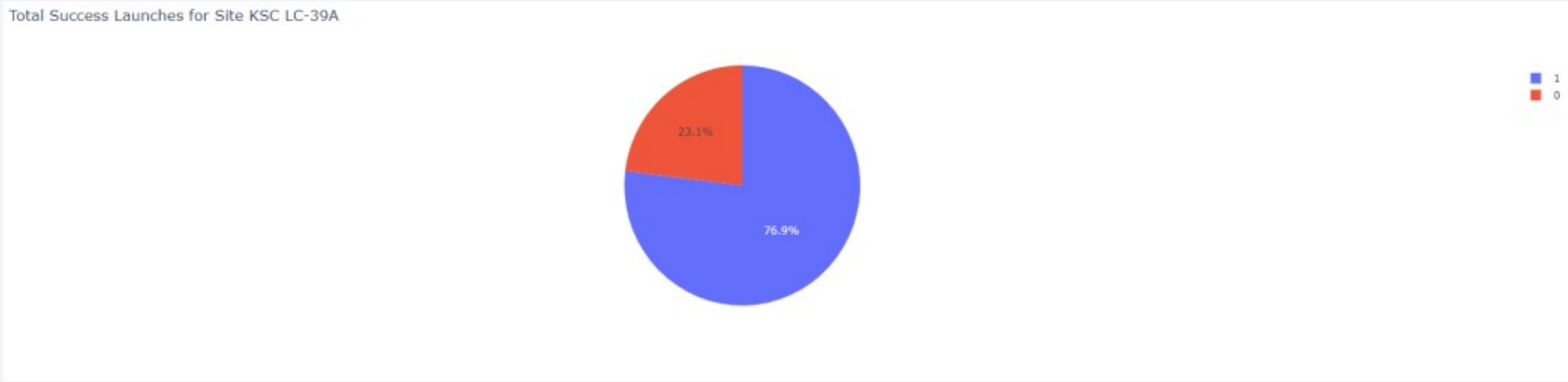

Section 4
Build a Dashboard with Plotly Dash

Dashboard – Total success by Site



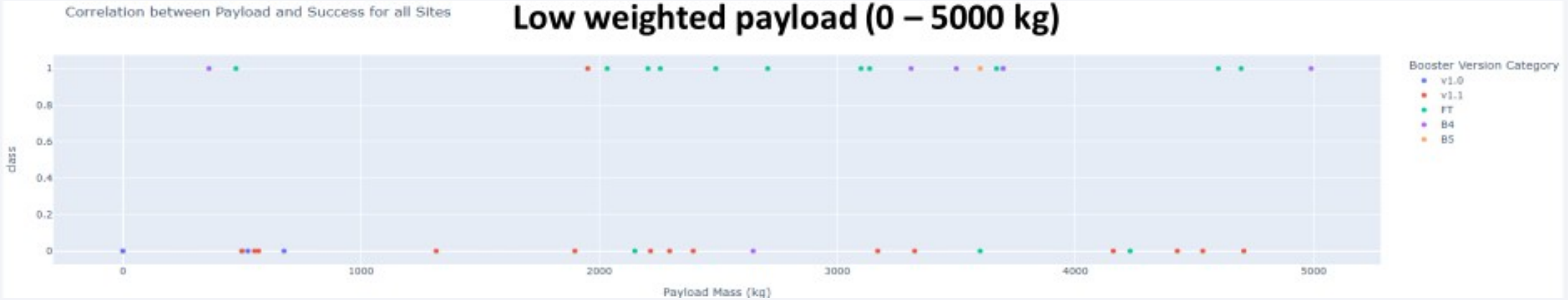
We see that KSC LC-39A has the best success rate of launches.

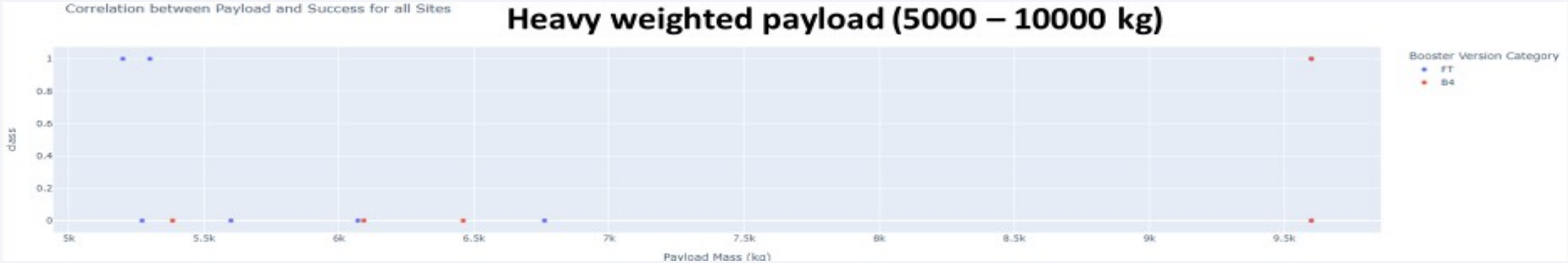
Dashboard – Total success launches for Site KSC LC-39A



We see that KSC LC-39A has achieved a 76.9% success rate while getting a 23.1% failure rate.

Dashboard – Payload mass vs Outcome for all sites with different payload mass selected





Low weighted payloads have a better success rate than the heavy weighted payloads.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

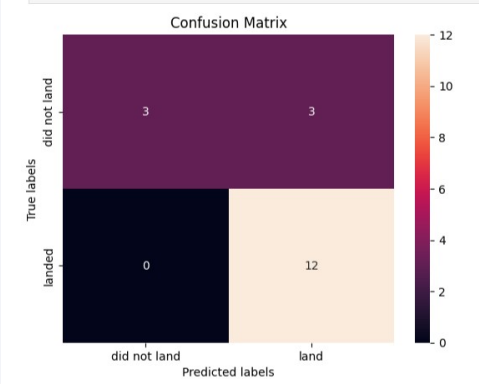
For accuracy test, all methods performed similar. We could get more test data to decide between them. But if we really need to choose one right now, we would take the decision tree.

```
# Since their accuracies are all the same, we pick based on their best scores
algo_score = {'Logistic regresssion': [logreg_cv.best_score_], 'SVM': [svm_cv.best_score_], 'Decision tree': [tre
df = pd.DataFrame.from_dict(algo_score, orient='index', columns=['Best scores'])
df
```

	Best scores
Logistic regresssion	0.846429
SVM	0.848214
Decision tree	0.875000
KNN	0.848214

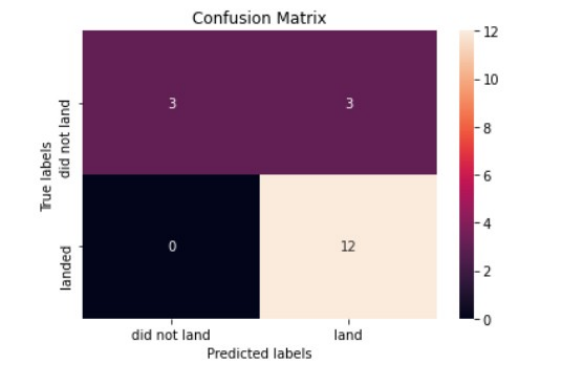
Confusion Matrix

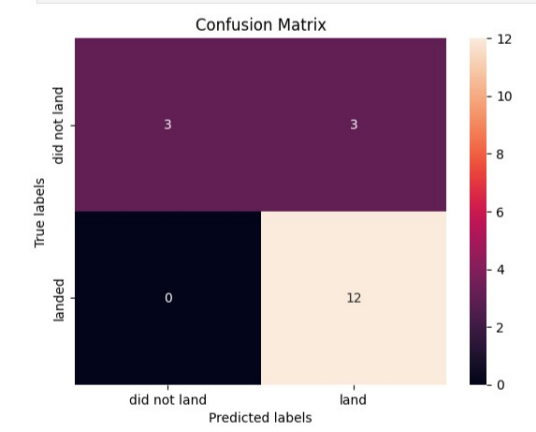
- As the test accuracy all of them are equal, the matrices are also identical. The main problem is false positives



Logistic Regression

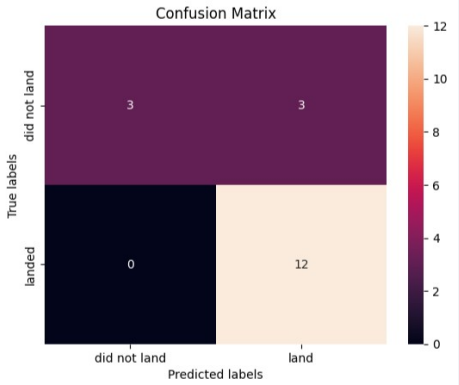
Decision Tree





KNN

SVM



Conclusions

- The success of a mission can be explained by several factors such as the launch site, the orbit and especially the number of previous launches. Indeed, we can assume that there has been a gain in knowledge between launches that allowed to go from a launch failure to a success.
- The orbits with the best success rates are GEO, HEO, SSO, ES-L1.
- Depending on the orbits, the payload mass can be a criterion to take into account for the success of a mission. Some orbits require a light or heavy payload mass. But generally low weighted payloads perform better than the heavy weighted payloads.
- With the current data, we cannot explain why some launch sites are better than others (KSC LC-39A is the best launch site). To get an answer to this problem, we could obtain atmospheric or other relevant data.
- For this dataset, we choose the Decision Tree Algorithm as the best model even if the test accuracy between all the models used is identical. We choose Decision Tree Algorithm because it has a better train accuracy.

Thank you!