IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ivan Sarmiento
2024-06-14

Outline

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

Executive Summary

- Develop Python code to manipulate data in a Pandas data frame
- Create Jupyter notebooks and make them sharable using GitHub
- Utilize data science methodologies to define and formulate a real-world business problem
- Compare different model of machine learning algorithms
- Decide which model is best suitable for the predict launch success

Introduction

- The main objective is predict if the Falcon 9 first stage will land successfully
- If we can determine if the first stage will land, we can determine the cost of a launch
- The information presented can be used if an alternate company wants to bid against SpaceX for a rocket launch

Section 1

Methodology

Methodology

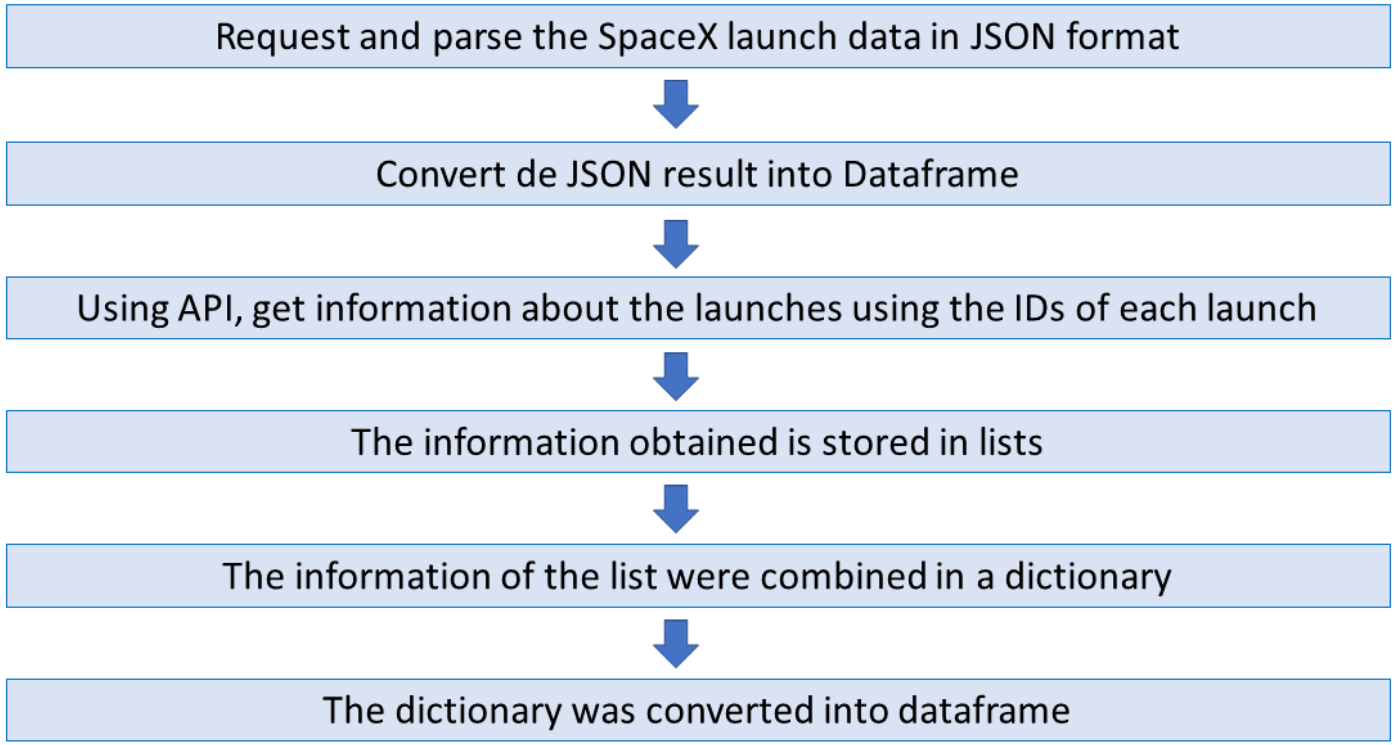
Executive Summary

- Data collection methodology:
  - SpaceX launch data that is gathered from an API, specifically the SpaceX REST API. Another way to collect the data is web scraping related Wiki pages
- Perform data wrangling
  - Landing outcomes were converted to Classes y (either 0 or 1). 0 is a bad outcome, that is, the booster did not land. 1 is a good outcome, that is, the booster did land
- 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

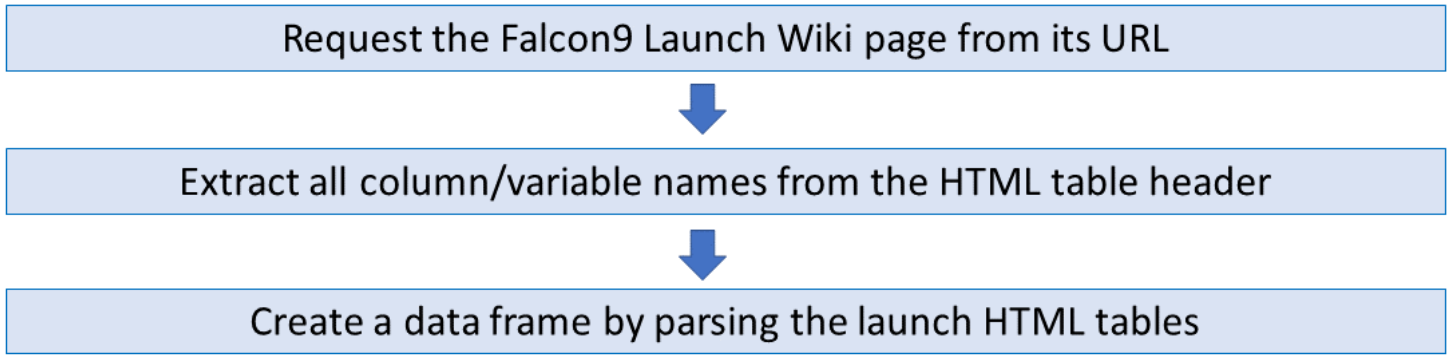
- Data sets were collected using two methods: Space X API and Web Scrapping
- On the following slides we resume the principal aspects of the two methods

Data Collection – SpaceX API



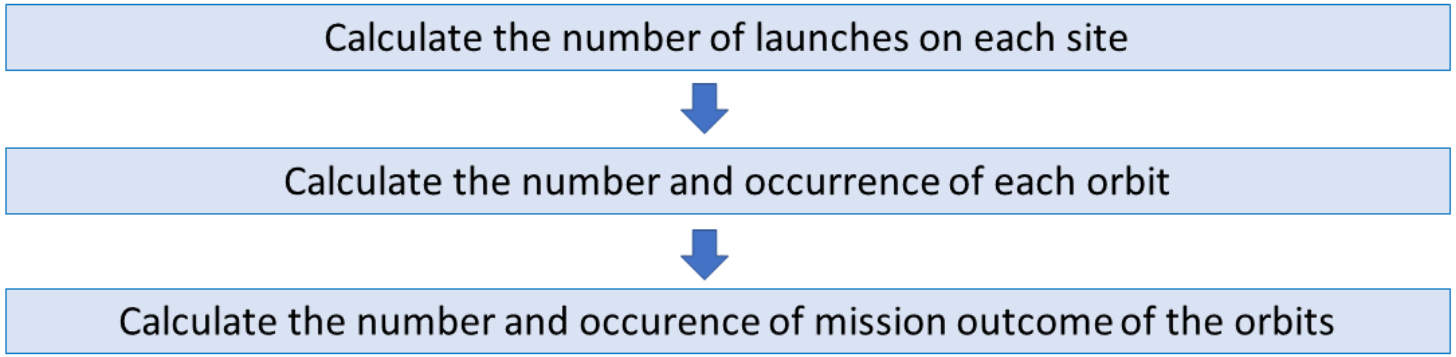
The GitHub URL of the completed SpaceX API calls notebook is:
https://github.com/isarmientop/CourseraApplDSCapstone/blob/main/jupyter_labs_spacex_data_collection_api.ipynb

Data Collection – Scraping



The GitHub URL of the completed web scraping notebook is:
https://github.com/isarmientop/CourseraApplDSCapstone/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling



The GitHub URL of the completed data wrangling notebook is:
https://github.com/isarmientop/CourseraApplDSCapstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

The following charts were plotted:

- Scatter plot of the Flight Number vs. Payload (to see how the FlightNumber and Payload variables affect the launch outcome)
- Scatter plot of the Flight Number vs. Site Launch (to find and explain the patterns in the Flight Number vs. Launch Site)
- Scatter plot of the Payload vs. Site Launch (to observe the relationship between launch sites and their payload mass)
- Bar chart of the Orbit vs. Average Success(to visually check if there are any relationship between success rate and orbit type)
- Scatter plot of the Flight Number vs. Orbit Type (to visualize and explain the relationship between Flight Number and Orbit type)
- Line plot of the Year vs. Orbit Type (to visualize the launch success yearly trend)

The GitHub URL of the completed EDA with data visualization notebook is:

https://github.com/isarmientop/CourseraApplDSCapstone/blob/main/edadataviz.ipynb

EDA with SQL

The following SQL queries were performed:

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Calculate and display the total payload mass carried by boosters launched by NASA (CRS)
- Calculate and display average payload mass carried by booster version F9 v1.1
- 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, failure landing_outcomes in drone ship ,booster versions, launch_site for the months 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

The GitHub URL of the completed EDA with SQL notebook is:

https://github.com/isarmientop/CourseraApplDSCapstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

The following map objects were created:

- Circle and Markers for all launch sites: to observe proximity to the Equator line and the proximity to the coast of all launch sites;
- Markers for the success/failed launches: to visualize with differents colors the success or failed of the launches
- Circle at NASA Johnson Space Center's coordinate to label and visualize the NASA Johnson Space Center's ubication
- Marker clusters to simplify a map containing many markers having the same coordinate
- MousePosition to get the coordinate (Lat, Long) for a mouse over on the map
- Marker on the selected closest coastline point on the map
- PolyLine to join coastline point and launch site point

The GitHub URL of the interactive Map with Folium notebook is:
https://github.com/isarmientop/CourseraApplDSCapstone/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

The following plots/graphs and interactions were added to a dashboard:

- A pie chart to show total success launches of each launch site in order to visualize launch success counts.
- A pie chart graph to show the success (class=1) count and failed (class=0) count for a selected site in order to visualize for a selected site the percent of success and failed launches
- A Dropdown interaction in order to select a site or ALL sites for the figure of pie chart.
- A scatter chart to find if variable payload is correlated to mission outcome
- A Range Slider interactions to select the range of payload used for the figure of the scatter chart

The GitHub URL of the python script used for the Dashboard is:
https://github.com/isarmientop/CourseraApplDSCapstone/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

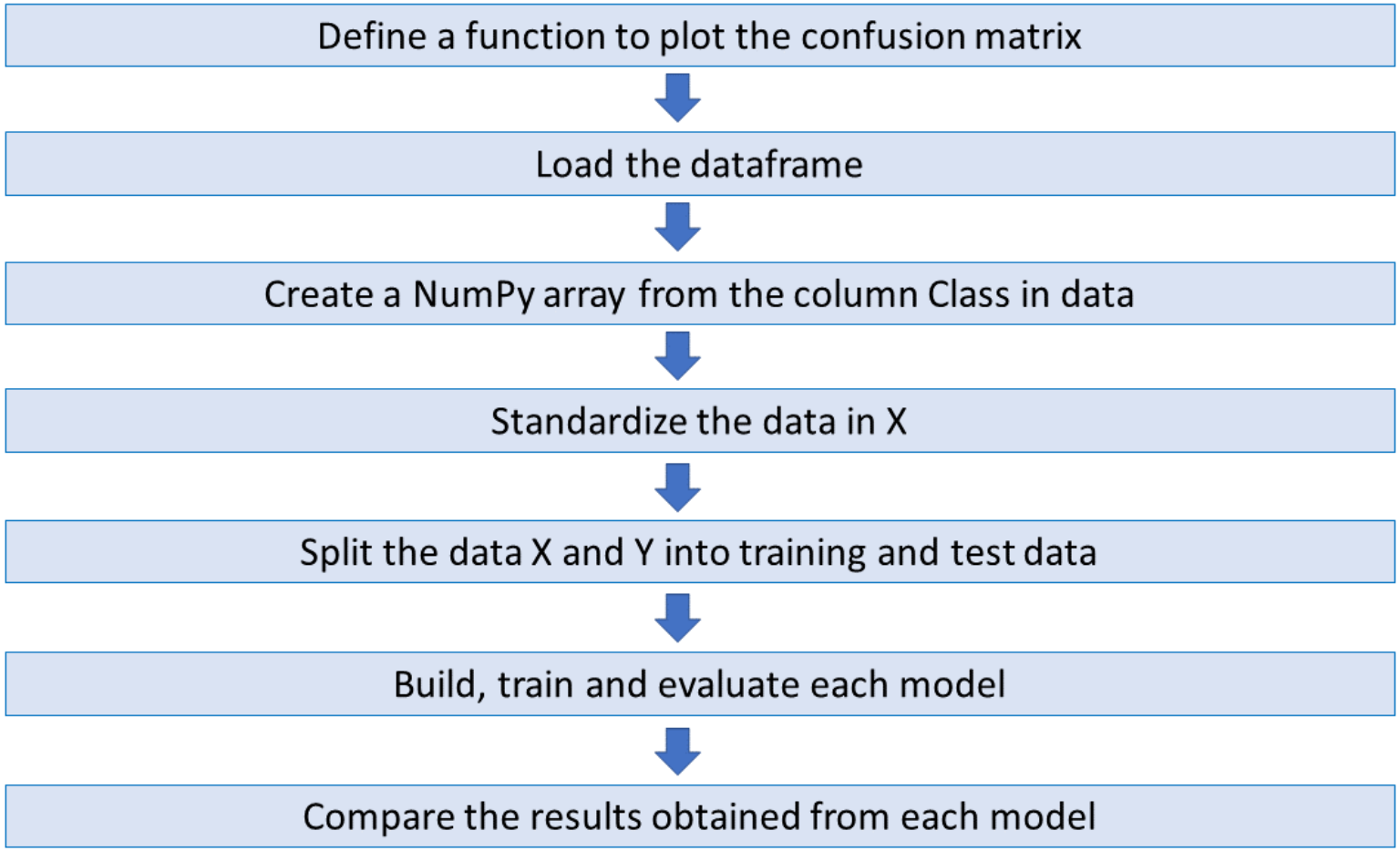
Summary:

- Classification models built and evaluated: logistic regression, support vector machine, decision tree classifier and k nearest neighbors
- For each model the following tasks were carried out:
    1. Create object with the model
    2. Create a GridSearchCV object
    3. Fit GridSearchCV to the data
    4. Display the best parameters using the data attribute best_params_ and the accuracy on the validation data using the data attribute best_score_
    5. Calculate the accuracy on the test data using the method score
    6. Display the confusion matrix

The GitHub URL of the python script with ML prediction is:
https://github.com/isarmientop/CourseraApplDSCapstone/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.ipynb

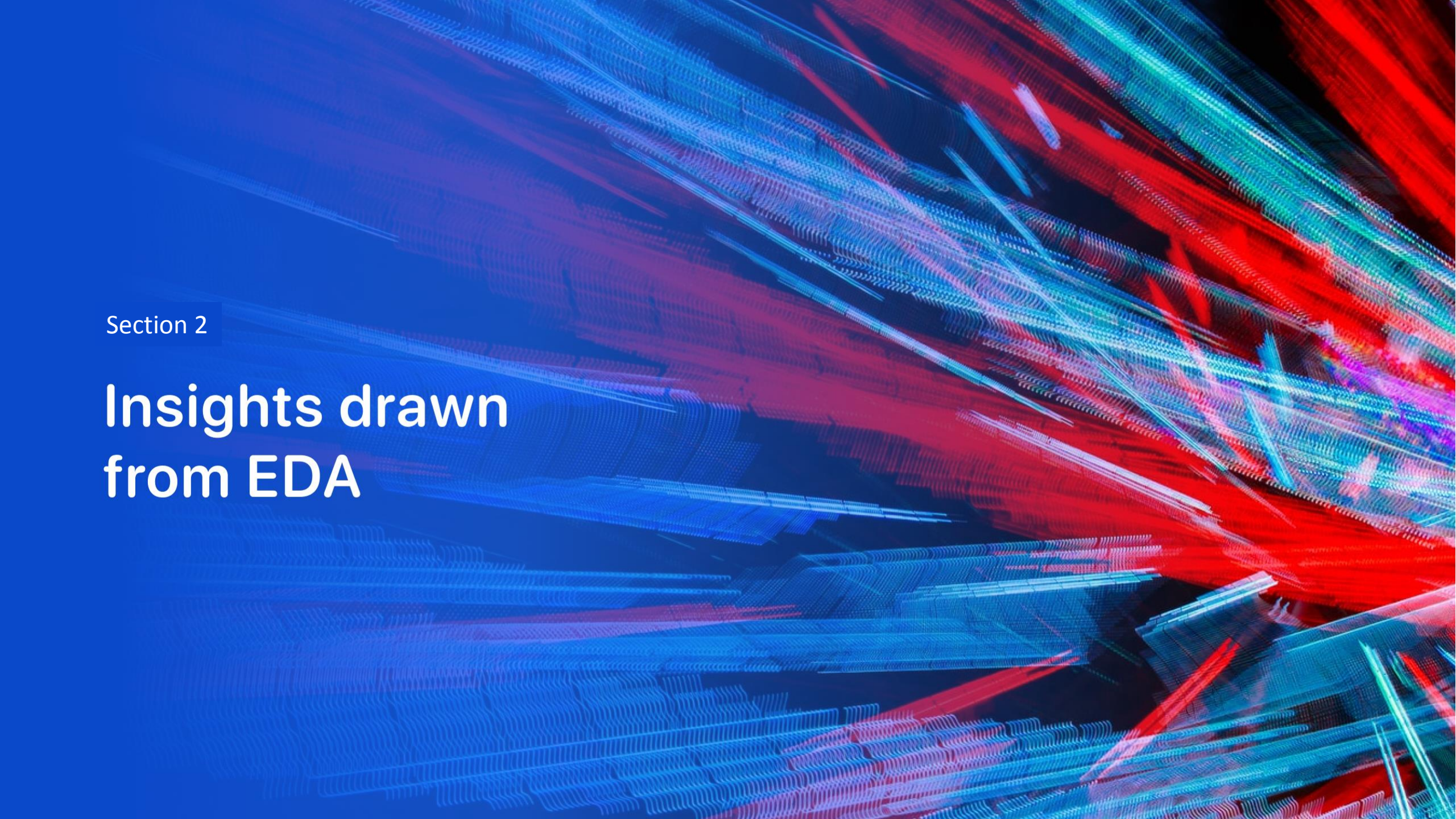
Predictive Analysis (Classification)

Development Process:

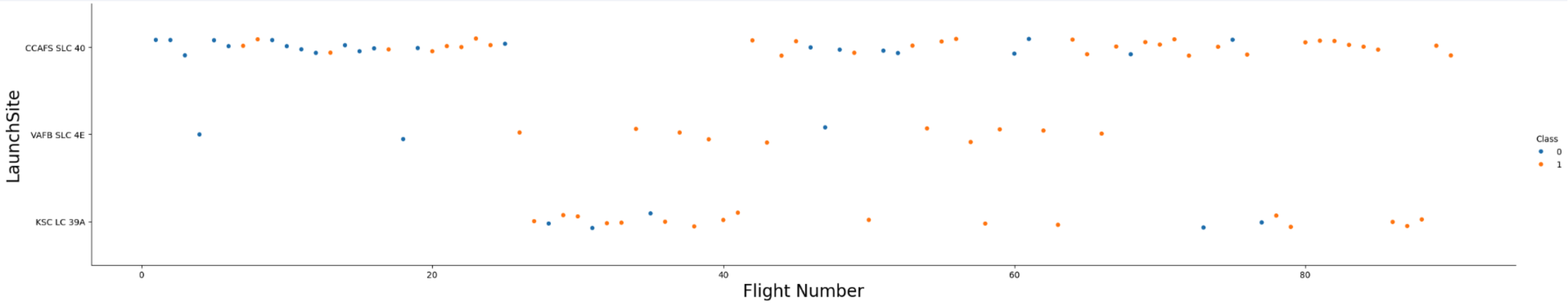


Results

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

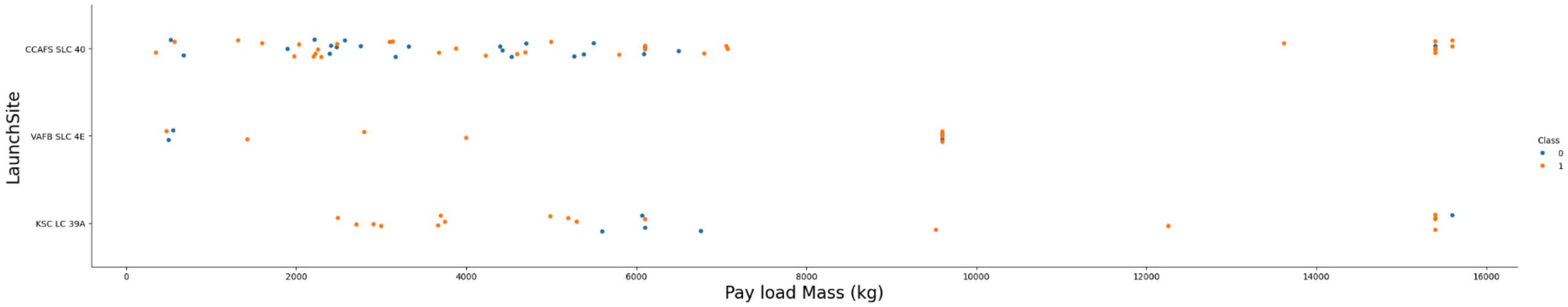

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



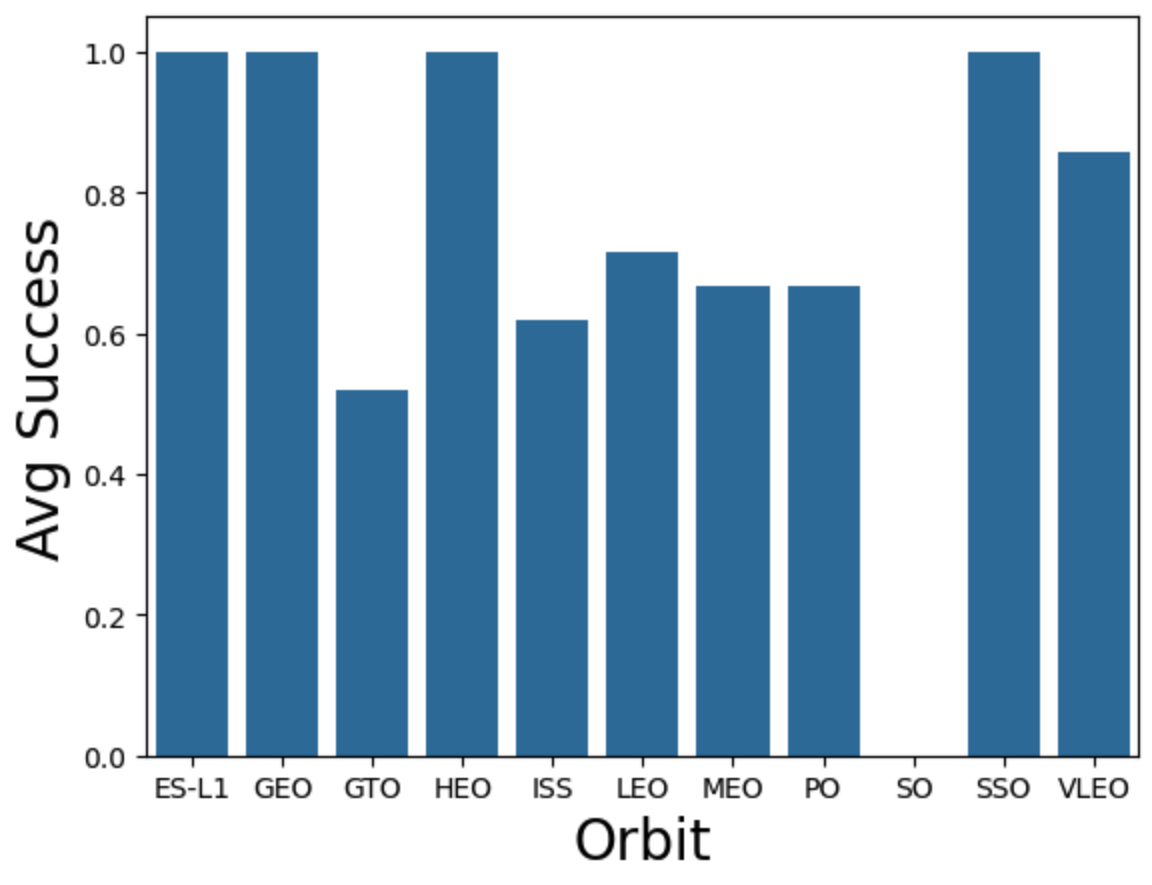
When de Flight Number increases the success for each launch site trend to be better

Payload vs. Launch Site



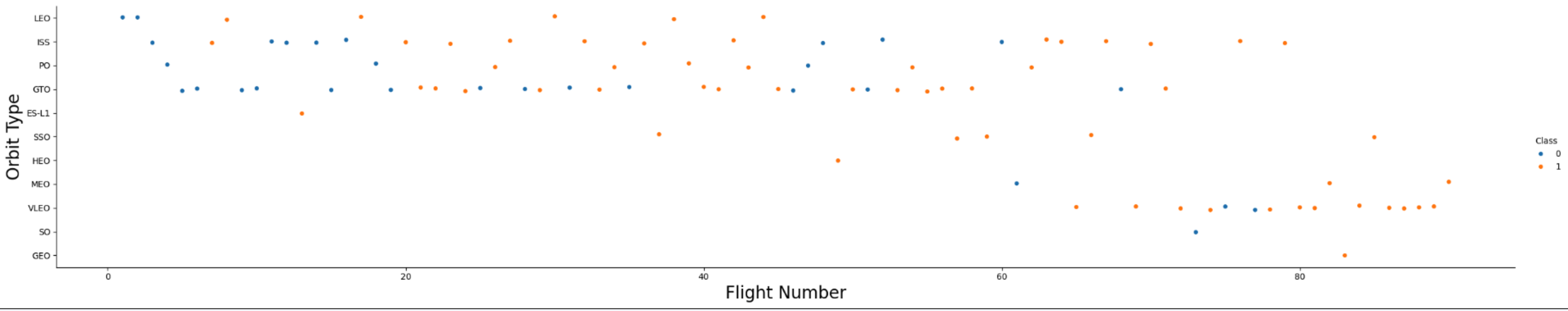
Most launches on all sites were made with payload < 8000 Kg

Success Rate vs. Orbit Type



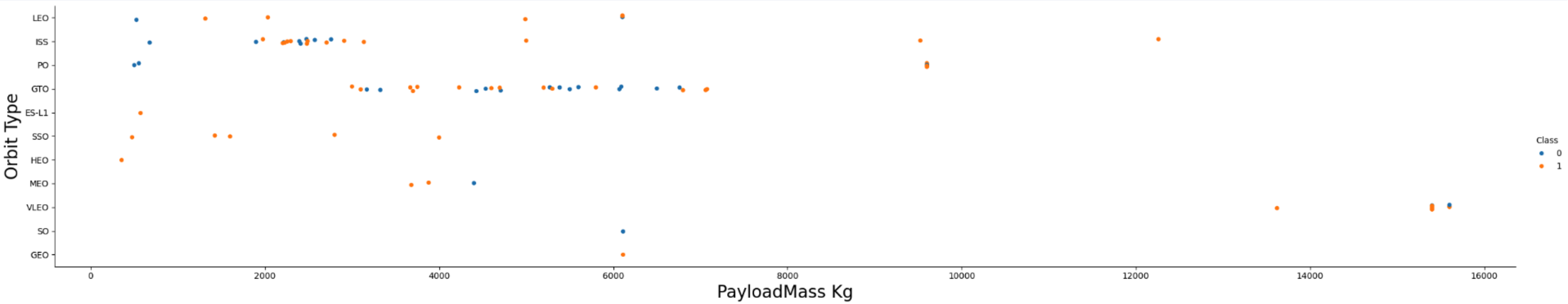
The orbits with higher success rate are: ES-L1, GEO, HEO and SSO
The SO orbit has no success rate

Flight Number vs. Orbit Type



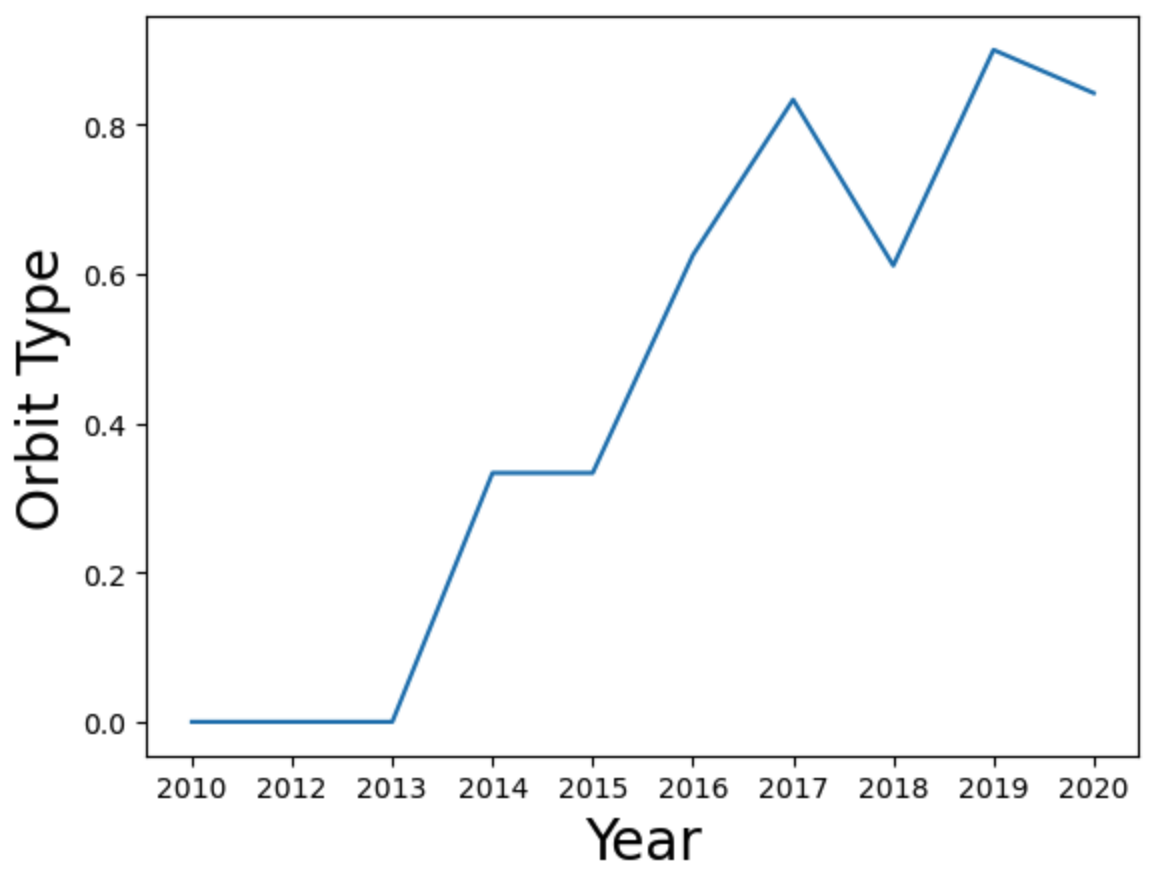
It seems in the LEO orbit the Success appears related to the number of flights.
It seems to be no relationship between flight number and success in GTO orbit

Payload vs. Orbit Type



It seems in the SSO orbit is always Success independent of the payload.
It seems to be no relationship between payload and success in GTO orbit

Launch Success Yearly Trend



The sucess rate since 2013 kept increasing till 2020

All Launch Site Names

Task 1

Display the names of the unique launch sites in the space mission

[9]:
```
%sql select distinct Launch_Site from SPACEXTABLE
```

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

[9]:

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

Comment: Obtain the Launch Site Names using option DISTINCT on SELECT statement

Launch Site Names Begin with 'CCA'

Task 2

Display 5 records where launch sites begin with the string 'CCA'

[10]:
```
%sql select *  from SPACEXTABLE where Launch_Site like 'CCA%' limit 5
```

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

[10]:

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

Comment: Obtain records using option LIKE on WHERE clause of SELECT statement.
Limit the number of records using LIMIT 5

Total Payload Mass

Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

[26]:
```
%sql select sum(PAYLOAD_MASS__KG_) as TotalPayloadMass  from SPACEXTABLE where Customer = 'NASA (CRS)'
```

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

[26]:

TotalPayloadMass
45596

Comment: Obtain the total payload mass using the function SUM.
Limit records to launched by NASA (CRS) using condition on WHERE clause

Average Payload Mass by F9 v1.1

Task 4

Display average payload mass carried by booster version F9 v1.1

[27]:
```
%sql select avg(PAYLOAD_MASS__KG_) as AvePayloadMass  from SPACEXTABLE where Booster_Version = 'F9 v1.1'
```

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

[27]:

AvePayloadMass
2928.4

Comment: Obtain the average payload mass using the function AVG.
Limit records to carried by booster version F9 v1.1 using condition on WHERE clause

First Successful Ground Landing Date

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

[28]:
```
%sql select date as DateFirstSucces from SPACEXTABLE where Landing_Outcome  like '%ground%' order by date limit 1
```

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

[28]:

DateFirstSucces
2015-12-22

Comment: Obtain records using option LIKE on WHERE clause of SELECT statement.
Use ORDER BY clause to obtain first records with earliest date
Limit the number of records using LIMIT 1

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
[29]: %sql select Booster_Version as BoosterSuccessDroneShip from SPACEXTABLE where Landing_Outcome  like '%Success (drone ship)%' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ <
```

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

[29]:

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

Complete SQL: select Booster_Version as BoosterSuccessDroneShip from SPACEXTABLE where Landing_Outcome like '%Success (drone ship)%' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000 order by date
Limit records using composited condition on WHERE clause

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

[17]:
```
%sql select count(Mission_Outcome) as TotalSuccess  from SPACEXTABLE where Mission_Outcome like '%Success%'
```

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

[17]:

TotalSuccess
100

[18]:
```
%sql select count(Mission_Outcome) as TotalFailure  from SPACEXTABLE where Mission_Outcome like '%Failure%'
```

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

[18]:

TotalFailure
1

Comment: Obtain the number of successful and failures mission outcomes using the function COUNT.
Limit records to successful or failure using condition on WHERE clause

Boosters Carried Maximum Payload

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

[30]:
```
%sql select Booster_Version as BoostesCarriedMaxPayload from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

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

[30]:

BoostesCarriedMaxPayload
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

Complete SQL: select Booster_Version as BoostesCarriedMaxPayload from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)

Used subquery and function MAX to obtain maximum value of payload mass

2015 Launch Records

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

[31]:
```
%sql select substr(Date, 6,2) as Month, Landing_Outcome as FailureLandigAoutcome, Booster_Version, Launch_Site  from SPACEXTABLE where Landing_Outcome like '%Failure%' and subst
```

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

[31]:

Month	FailureLandigAoutcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Complete SQL: select substr(Date, 6,2) as Month, Landing_Outcome as FailureLandigAoutcome, Booster_Version, Launch_Site from SPACEXTABLE where Landing_Outcome like '%Failure%' and substr(Date,0,5)='2015'

Used SUBSTR function to obtain month and year from Date

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

Task 10

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.

[32]:
```
%sql select Landing_Outcome, count(Landing_Outcome) as CountLandingAoutcome from SPACEXTABLE where (Landing_Outcome like '%Success (ground pad)%' or Landing_Outcome like '%Failu
```

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

[32]:

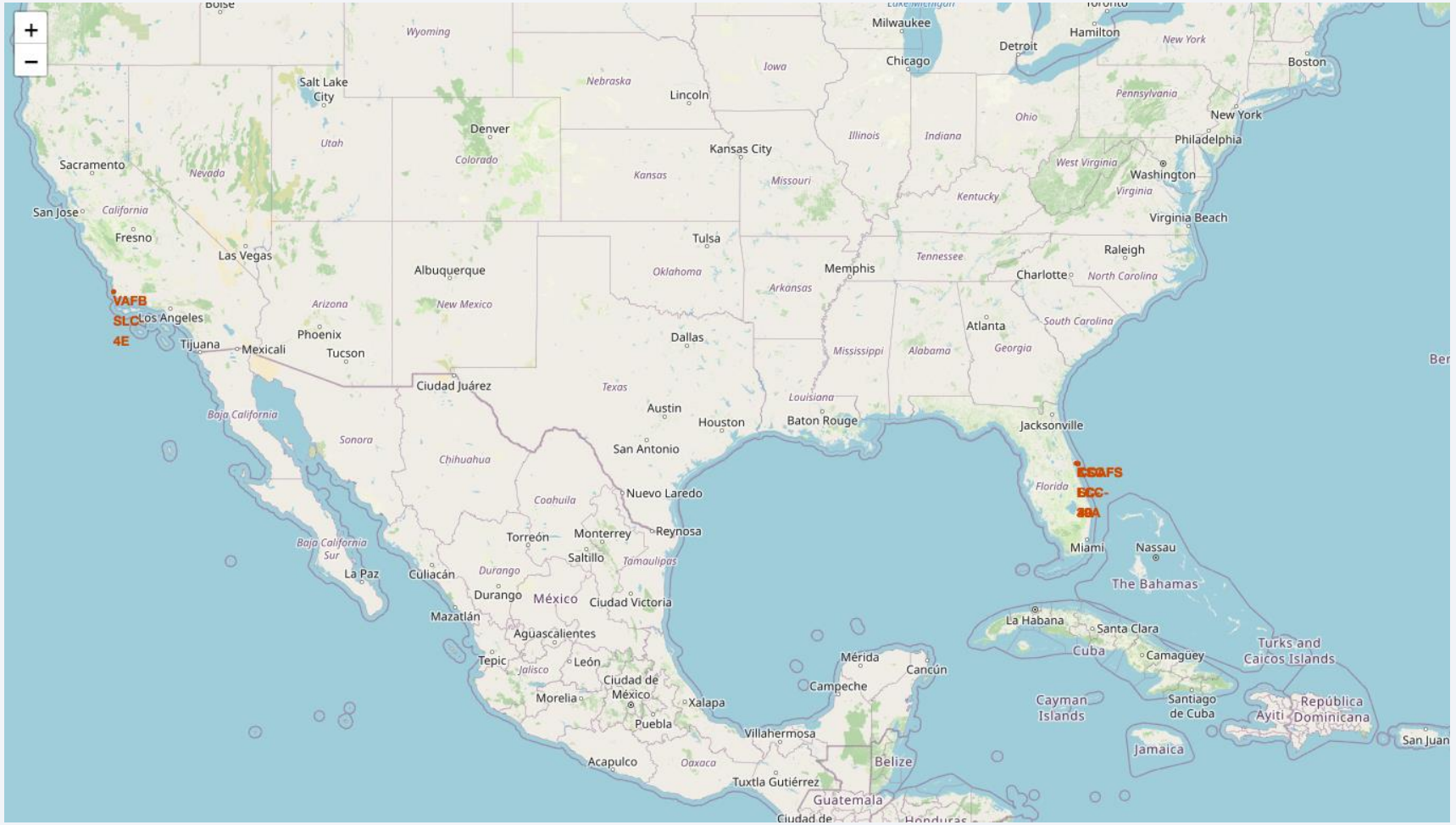
Landing_Outcome	CountLandingAoutcome
Failure (drone ship)	5
Success (ground pad)	3

Complete SQL: sql select Landing_Outcome, count(Landing_Outcome) as CountLandingAoutcome from SPACEXTABLE where (Landing_Outcome like '%Success (ground pad)%' or Landing_Outcome like '%Failure (drone ship)%') and (date >= '2010-06-04' and date <= '2017-03-20') group by Landing_Outcome

Used GROUP BY clause to resume results by Landing_Outcome. Use COUNT function to obtain total records for each Landing_Outcome

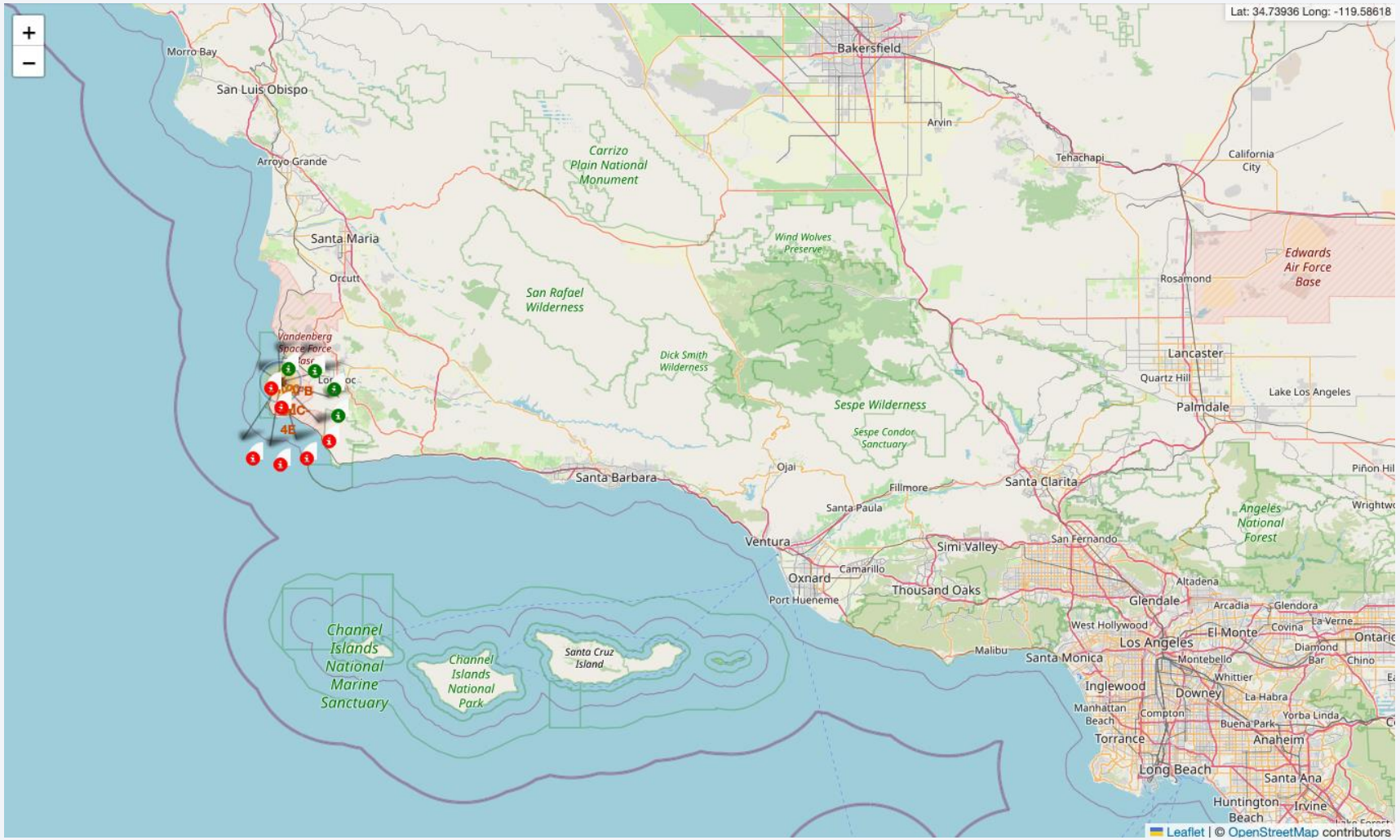

Section 3
Launch Sites
Proximities Analysis

Visualize Sites Locations with Folium



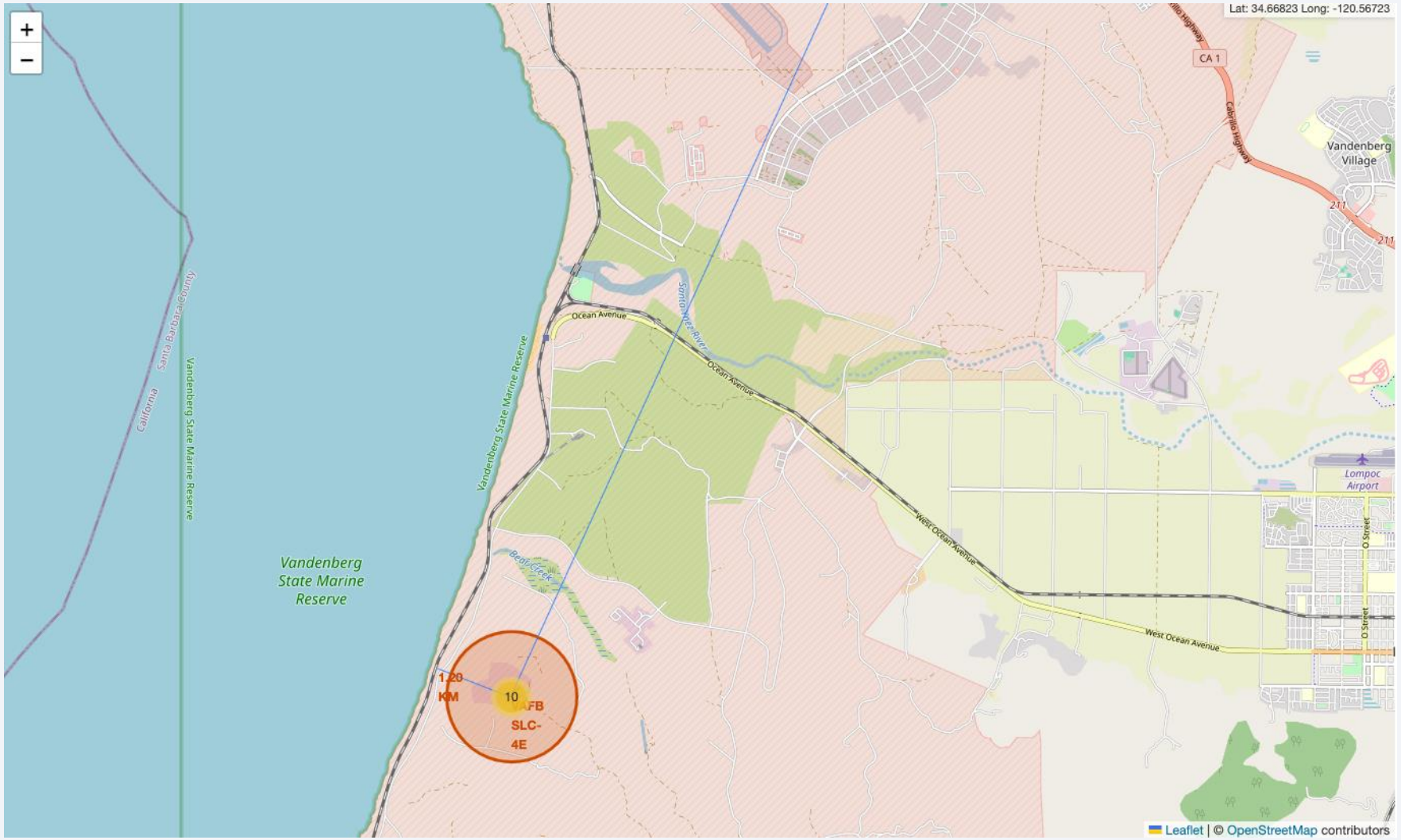
Comment: Launch sites are always close to coastlines and far from cities.

Success/failed launches for site VAFB SLC-4E

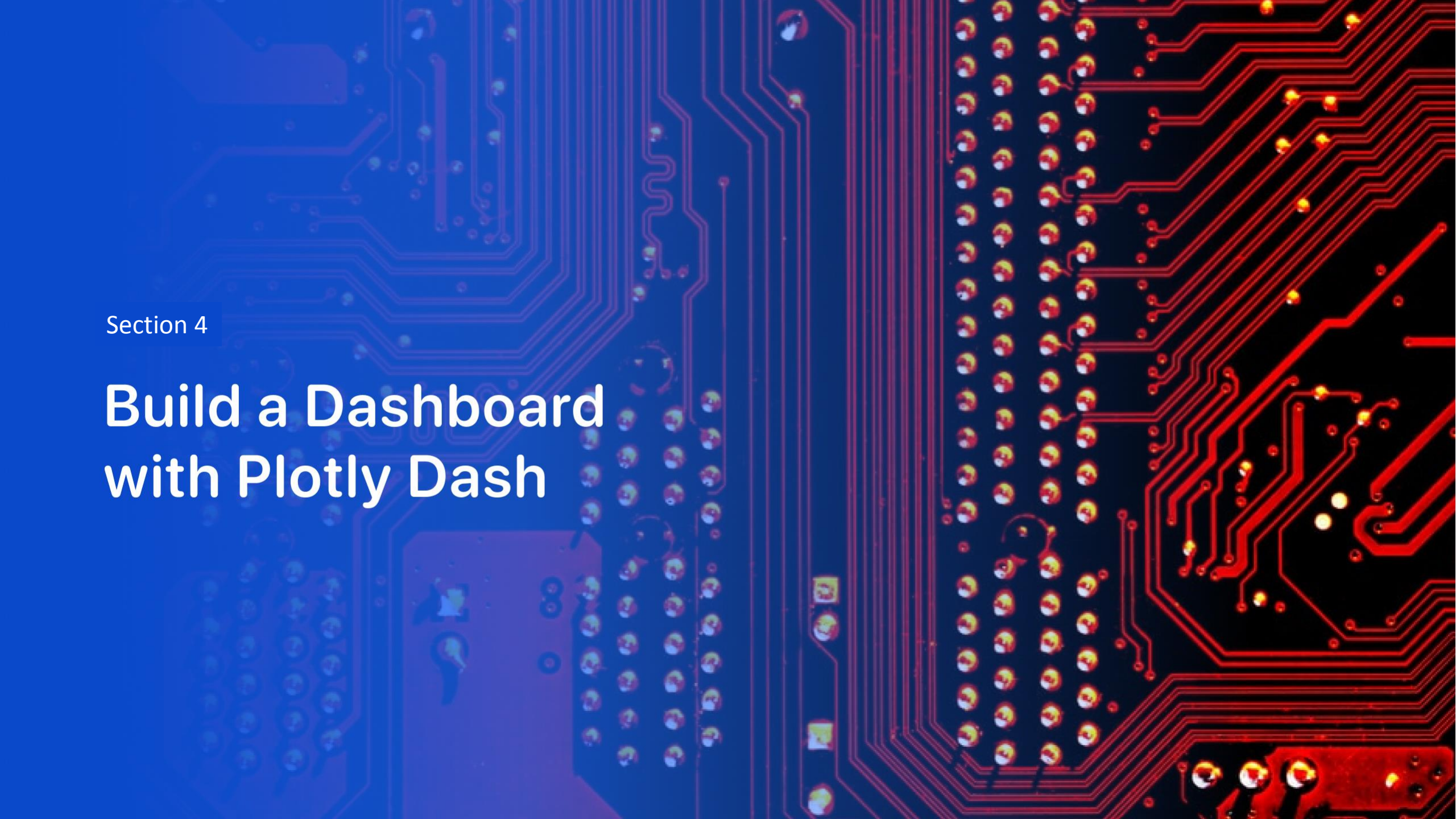


Comment: Need sum the map to observe markers of success / failed launches.

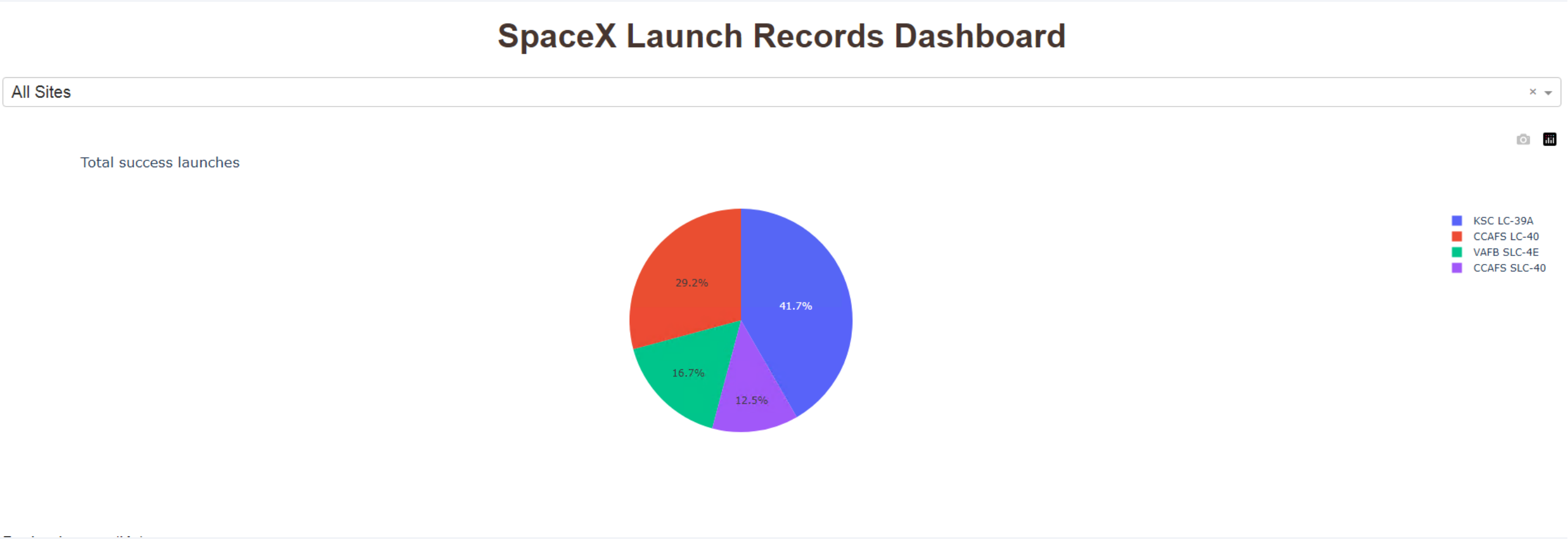
Show coastline and city distance



Comment: Show distance from launch site VAFB SLC-4E to coastline and to the city Santa Maria.

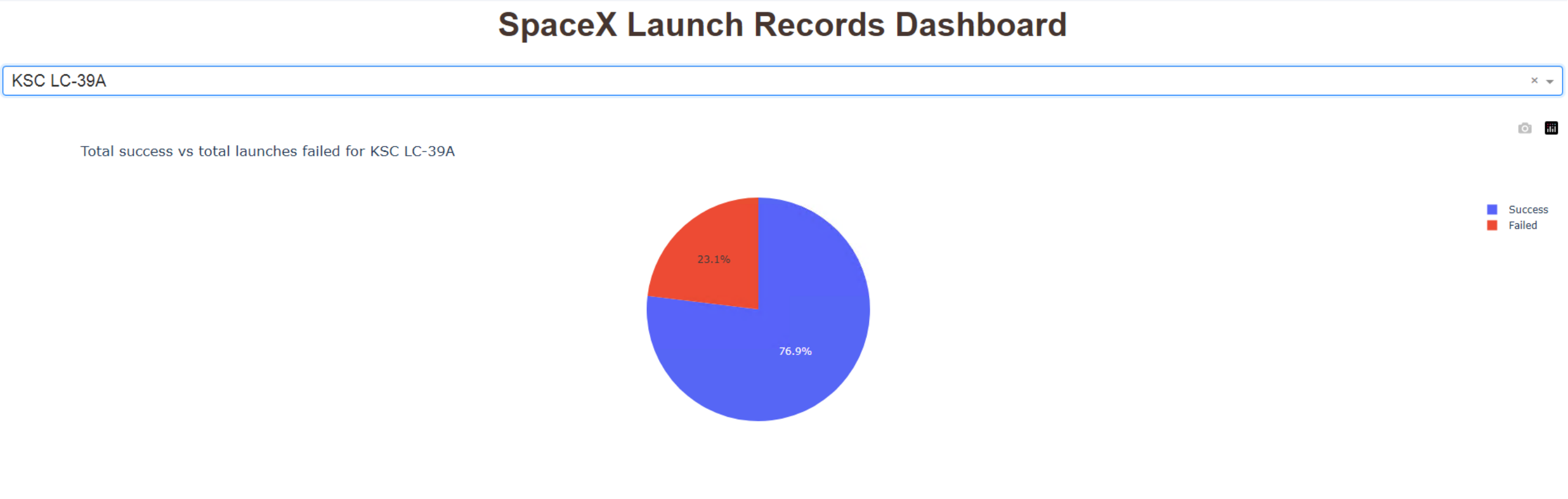

Section 4
Build a Dashboard with Plotly Dash

Total success launches



Comment: The site KSC LC-39A has the largest success launches. The site CCAFS SLC-40 has minor success launches

Success launches vs Failed launches for KSC LC-39A



Comment: For the site KSC LC-39A it seems 3 of each 4 launches are success

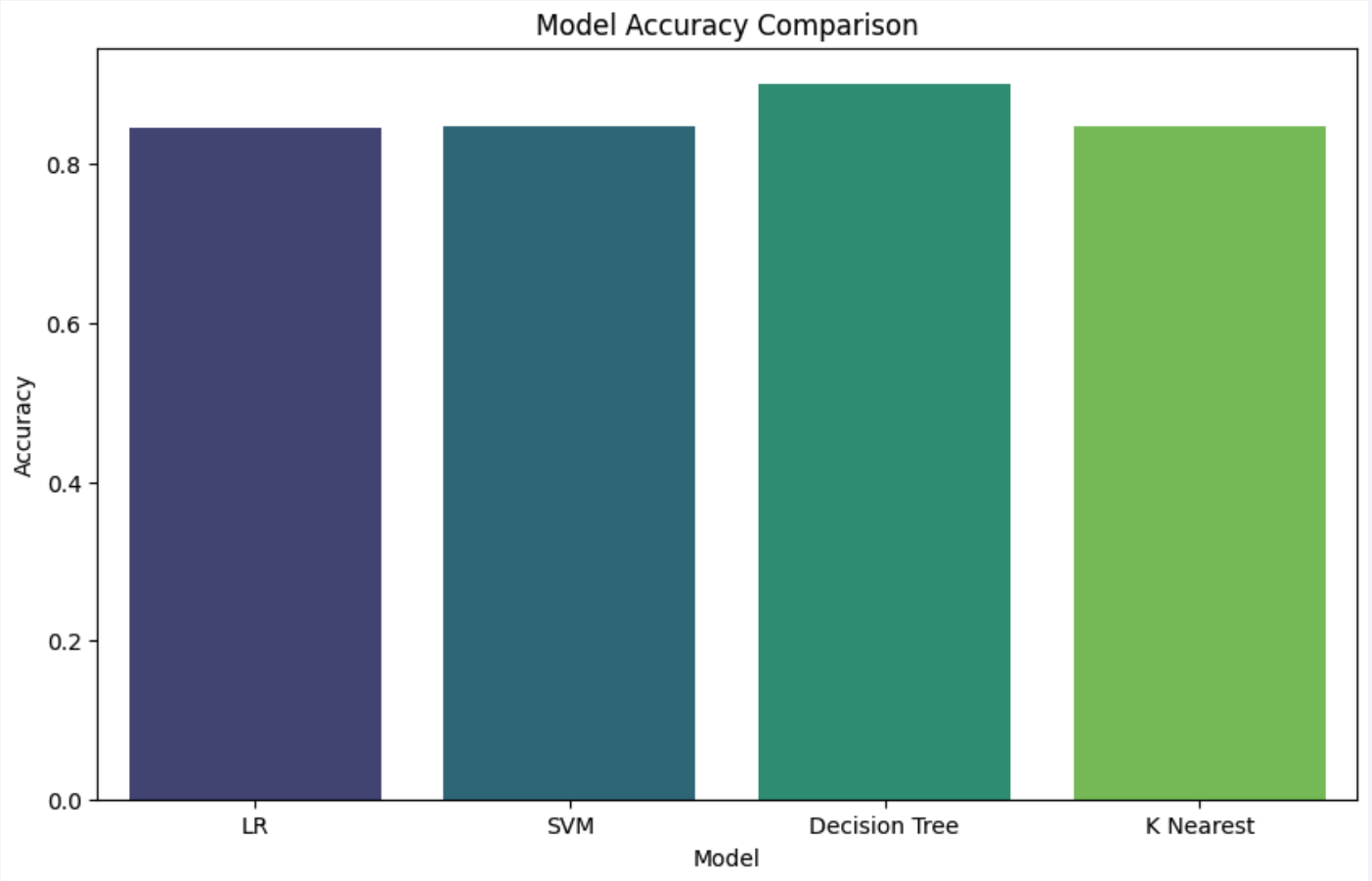
Payload mass vs Class for ALL Sites Launch

Comment: Booster version FT in range 2000-3500 of payload has a higher success.
Booster version v1.1 seems to failed independent of payload mass

Section 5

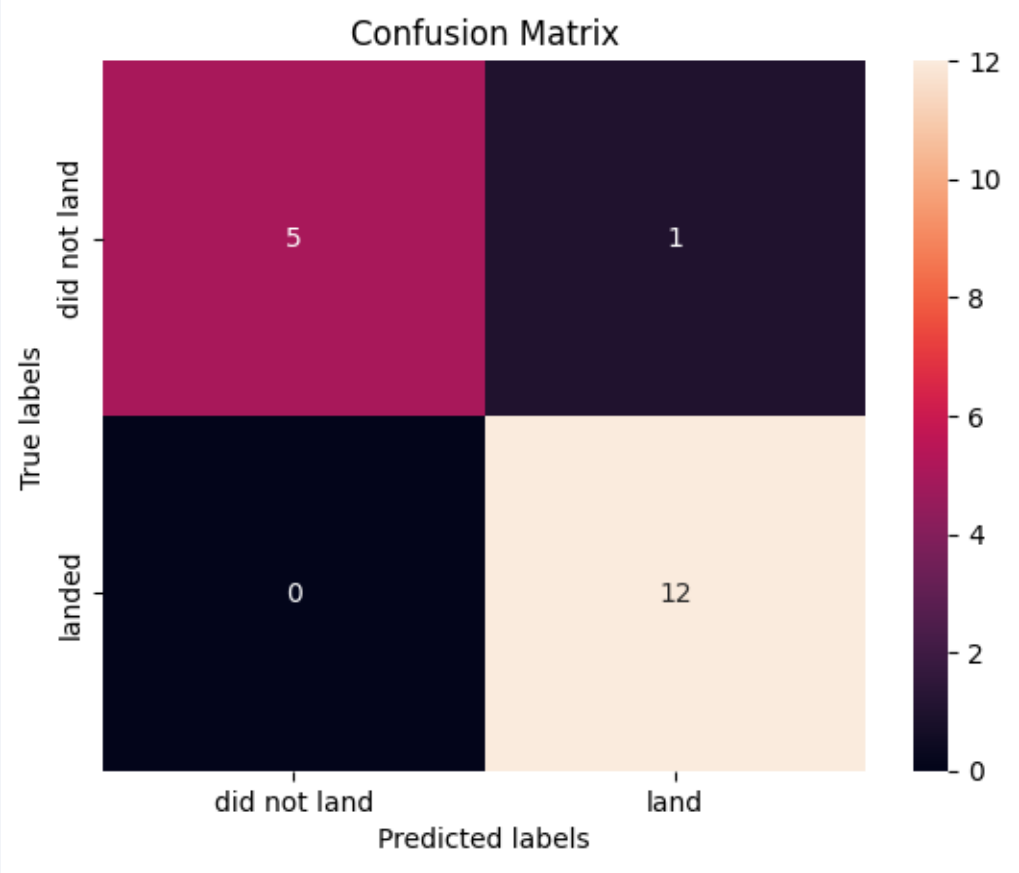
Predictive Analysis (Classification)

Classification Accuracy



Comment: The model with the best accuracy is decision tree

Confusion Matrix



Comment: The Confusion Matrix for the model decision trees show 17 (of 18) good predictions

Conclusions

- The four model build and evaluated shows good results
- The Confusion Matrix is and excellent tool for visualize the models predictions
- According with the data used the best model is Decision Tree
- Folium is an excellent tools for visualize data in maps
- Ploty Dash is an excellent tool for presenting results in an interactive form

Appendix

Python notebook was used for build bar chart comparing models accuracy. The URL of the notebook is:

https://github.com/isarmientop/CourseraApplDSCapstone/blob/main/build_barchar_modelaccuracy.ipynb

Thank you!