IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Fernando Gutierrez
April 8th, 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

- SpaceX is a space company owned by billionaire entrepreneur Ellon Musk, whose main objective is to make human life multiplanetary. The starting point will be Mars and to get there, they pioneered a new generation of rockets called Falcon 9. Its main feature is that the booster, which contains the engine (first-stage) is reusable, which makes the process more efficient and saves the company a lot of money.
- Once a rocket is launched and reaches a certain point, its first stage is detached and returned to a specific location on earth. In that context, the main question we want to know is:
- Will the return landing be successful or not?
- What set of features impact the landing success rate the most?

Section 1

Methodology

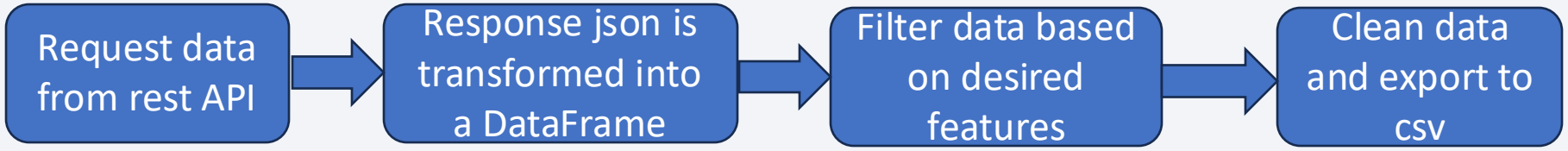
Methodology

Executive Summary

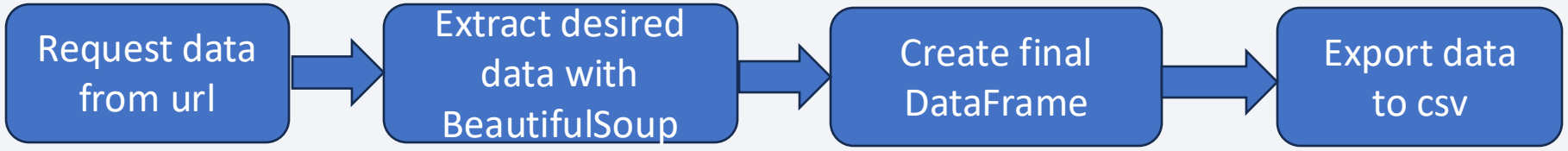
- Data collection methodology:
  - SpaceX Rest API
  - Web scrapping
- Perform data wrangling
  - Dropping unnecessary columns
  - Create a column for target variable
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Logistic regression, Support Vector Machine (SVM), Decision tree and K-nearest neighbors (KNN) models each with a grid a parameters in a grid search object

Data Collection

- Data was collected from 2 sources:
  - .SpaceX Rest API: https://api.spacexdata.com/v4/launches/past



  - Web scrapping Wikipedia tables containing information about past SpaceX launches: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches



Data Collection – SpaceX API

1. Get responde from API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

2. Create df from json

```
data = pd.json_normalize(response.json())
```

3. Obtain desired features

```
# Call getLaunchSite
getLaunchSite(data)

# Call getPayloadData
getPayloadData(data)

# Call getCoreData
getCoreData(data)
```

4. Create dic. based on desired features

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

5. Create clean dataframe

```
# Create a data from launch_dict
df = pd.DataFrame(launch_dict)
```

6. Filter for Falcon 9 launches only

```
data_falcon9 = df[df['BoosterVersion']!='Falcon 1']
```

7. Deal with missing values

```
avg_pmass = data_falcon9['PayloadMass'].mean(axis = 0)
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan, avg_pmass, inplace = True)
```

8. Export data

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

https://github.com/FernandoIGD/Capstone-project/blob/main/spacex_data_collection_api.ipynb

Data Collection - Scraping

1. Get raw data from url

```
response = requests.get(static_url)
response.status_code
```

2. Create soup object

```
soup = BeautifulSoup(response.content)
```

3. Filter for tables

```
html_tables = soup.find_all('table')
```

4. Extract table header names

```
column_names = []
ths = first_launch_table.find_all('th')
for i in range(len(ths)):
    col_name = extract_column_from_header(ths[i])
    if (col_name != None) and (len(col_name)>0):
        column_names.append(col_name)
```

5. Create dic. of desired features

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with eacl
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6. Parse & extract desired data

7. Create final dataframe

8. Export to data to csv

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

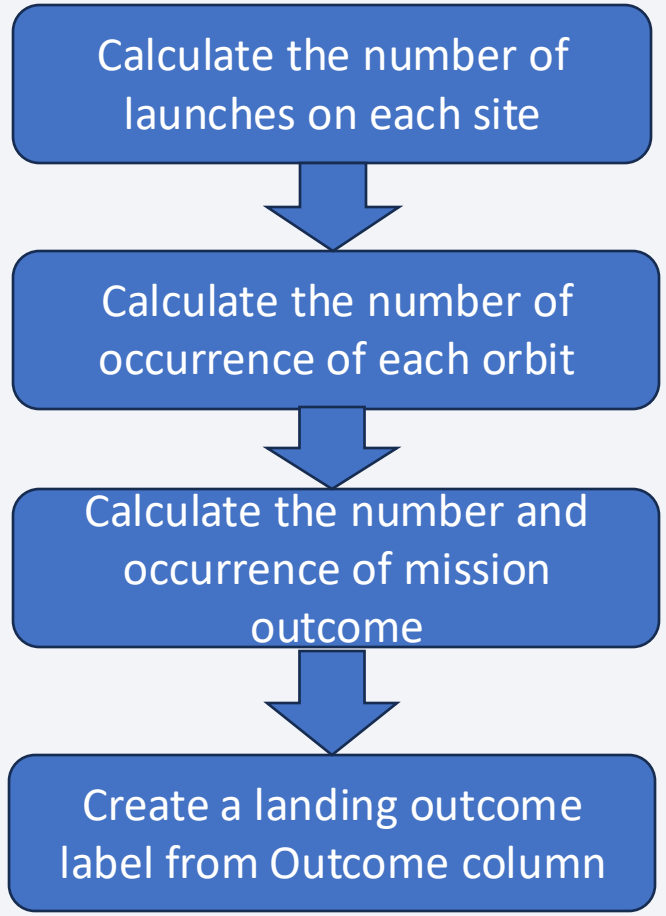
https://github.com/FernandoIGD/Capstone-project/blob/main/spacex_data_collection_webscrapping.ipynb

Data Wrangling

On this step, the main objective was to create a categorical target variable Column that indicates wether the landing was succesful or not.

To do this, we identified the ‘bad outcomes’ from the unique outcomes in The column with the same header name. Then, we created a landing class series that contains only ‘0s’ and ‘1s’ corresponding to an unsuccessful or Successful landing, respectively.

Finally, we added this ’class’ series as the final column in our dataframe containing our data.



https://github.com/FernandoIGD/Capstone-project/blob/main/spacex_data_wrangling.ipynb

EDA with Data Visualization

- Scatter graphs
  - Flight number vs payload mass
  - Flight number vs Launch Site
  - Payload mass vs Launch Site
  - Flight number vs Orbit type
  - Payload mass vs Orbit Type
- Bar plots
  - Orbit type vs Success Rate
- Line chart
  - Year vs Success Rate

- Scatter plots are used to show relationships between the variables, called correlation. Any possible existing relation could be used to in a machine learning model.
- The bar plot displays the success rate value of each unique orbit type.
- Line chart shows the variation of success rate over the years.

https://github.com/FernandoIGD/Capstone-project/blob/main/spacex_EDA_visualization.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 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 all the booster versions that 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

https://github.com/FernandoIGD/Capstone-project/blob/main/spacex_data_EDA_sql.ipynb

Build an Interactive Map with Folium

- The folium map is centered around NASA Johnson Space center. It shows each launch site location (California and Florida) marked with an orange circle and a pop-up icon with each site name.
- Also, the number of launches from each site, marked as successful or failure. To accomplish this, a marker cluster containing each launch location and landing outcome marked as green or red depending on the result.
- Finally, we wanted to calculate the distance between a launch location and its surroundings, i.e. a coastline point. For this, we added a straight line and a a distance marker.

https://github.com/FernandoIGD/Capstone-project/blob/main/spacex_data_folium.ipynb

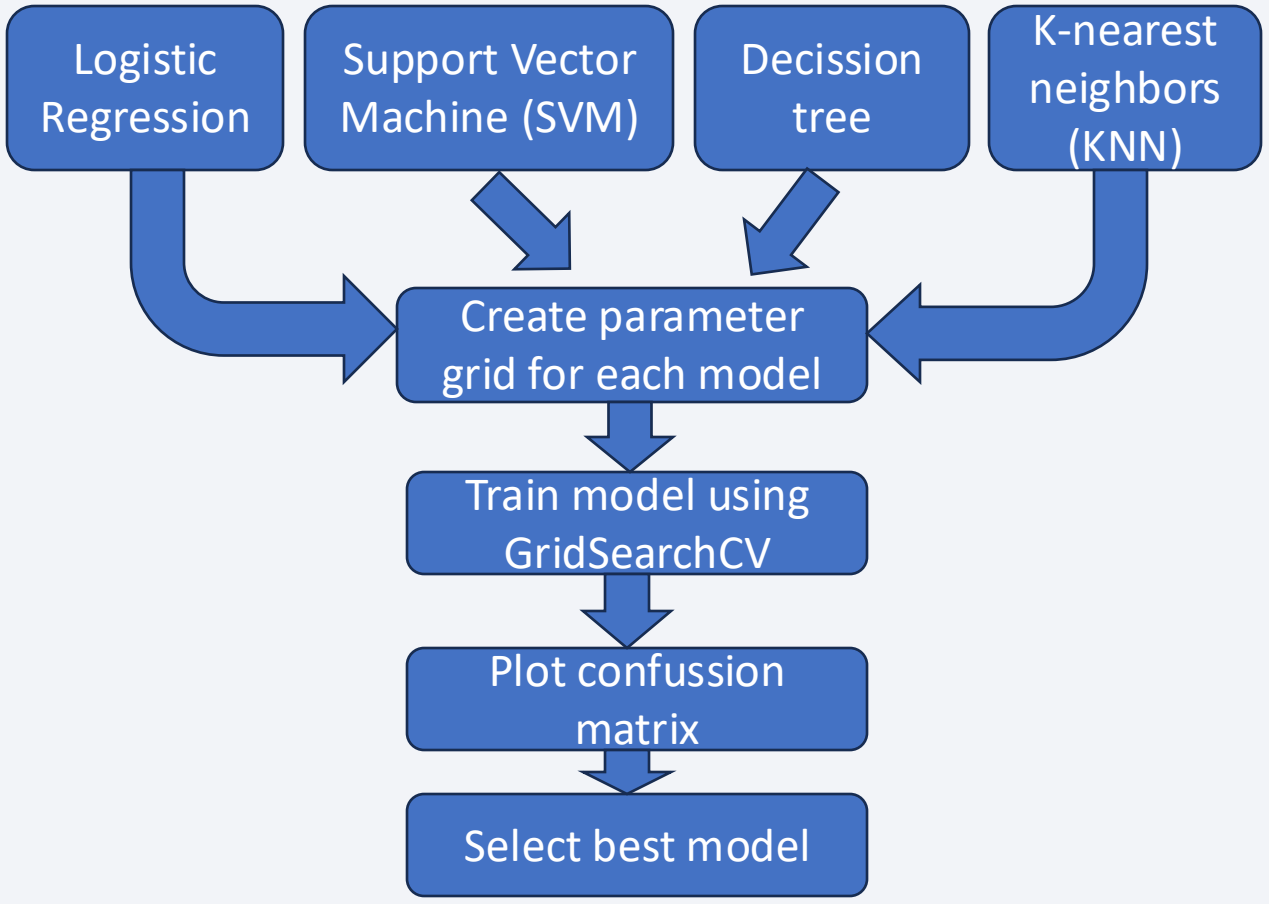
Build a Dashboard with Plotly Dash

- Dropdown list to select either all sites or one individual launch site location.
- Pie chart displaying success and failure percentage for the chosen site.
- A range slider to select payload mass values in a range from 0 to 10 tons.
- Scatter plot between the payload mass and success rate for a given site location.

https://github.com/FernandoIGD/Capstone-project/blob/main/spacex-dash-app.py

Predictive Analysis (Classification)

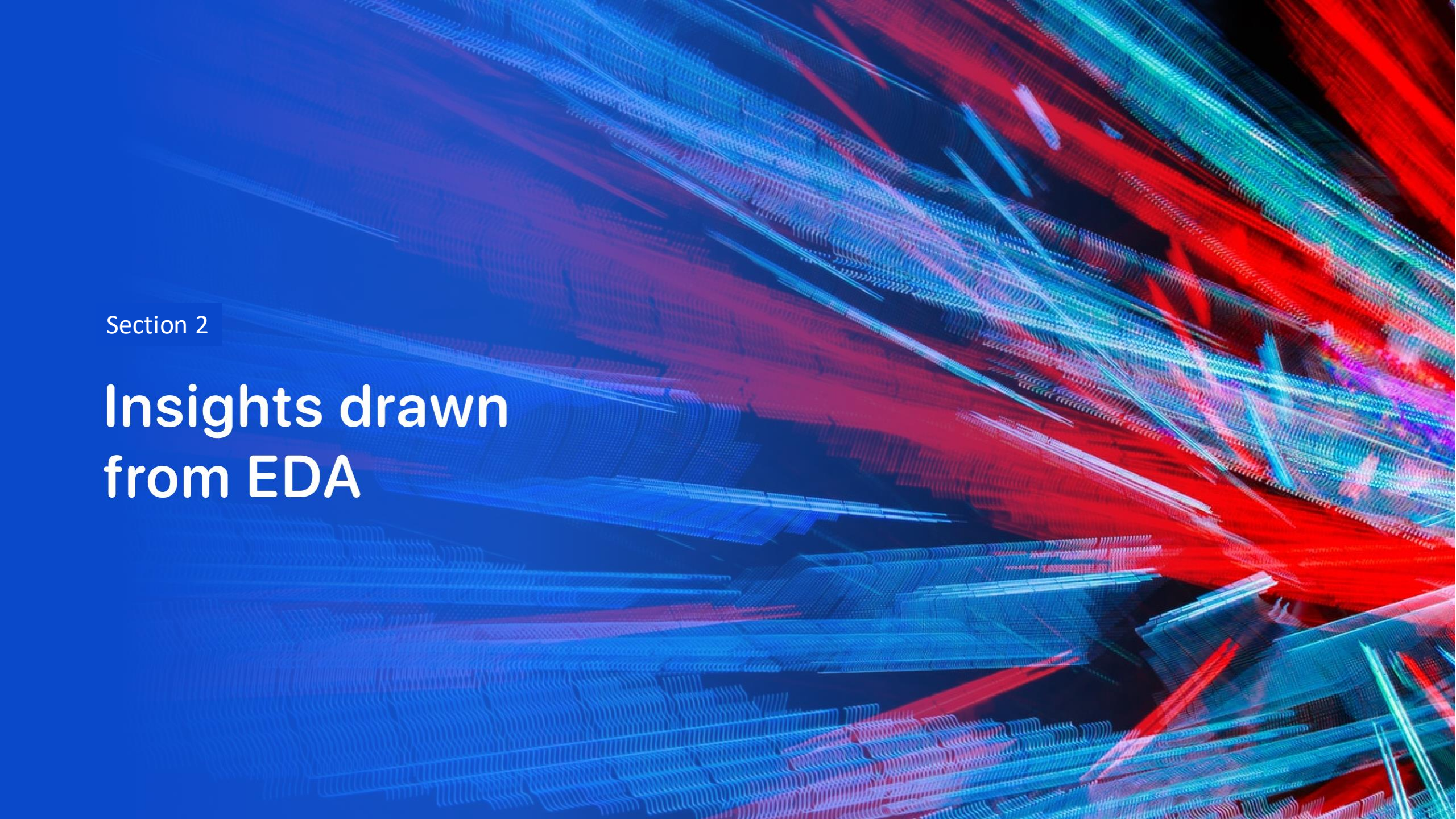
- 4 machine learning models were built to predict the landing outcome given a set of features selected from the previous EDA. Then, all models were evaluated, and the best one was selected by comparing model accuracies.



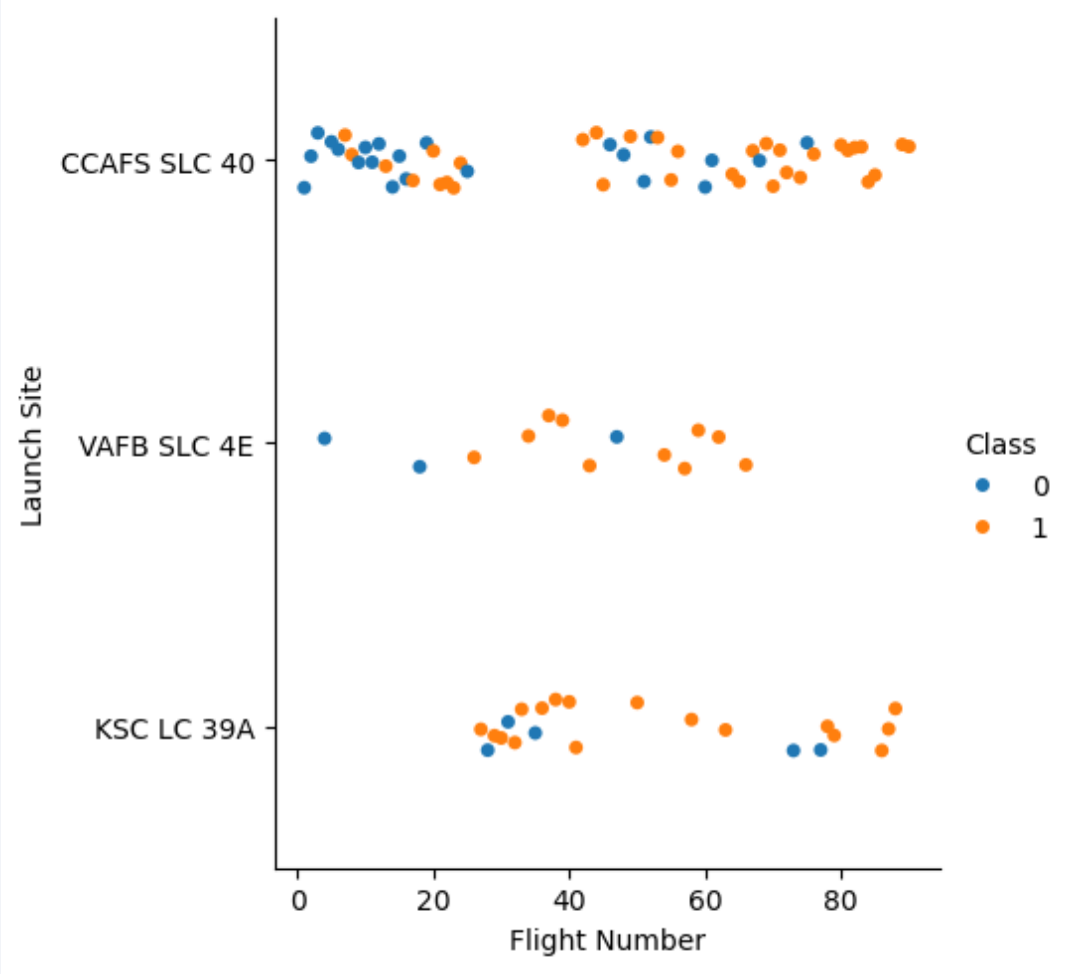
https://github.com/FernandoIGD/Capstone-project/blob/main/spacex_data_predictive_analysis.ipynb

Results

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

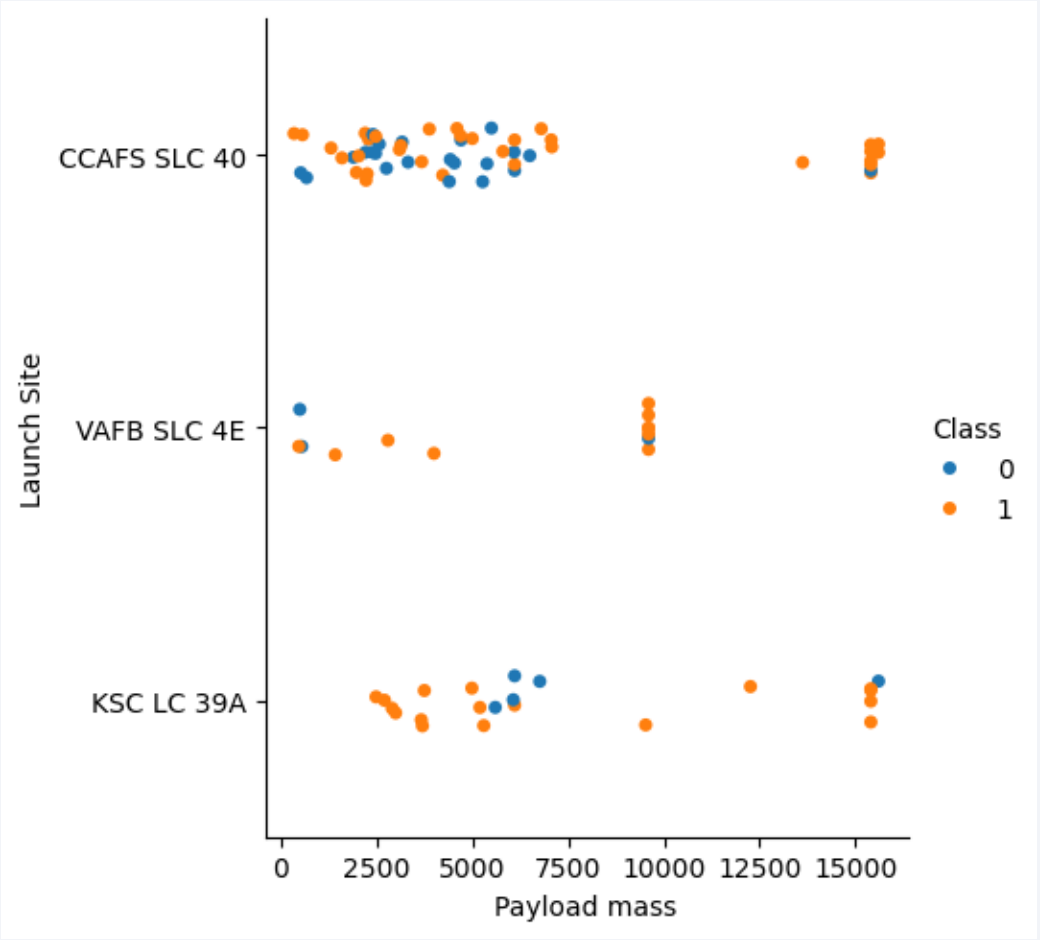

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



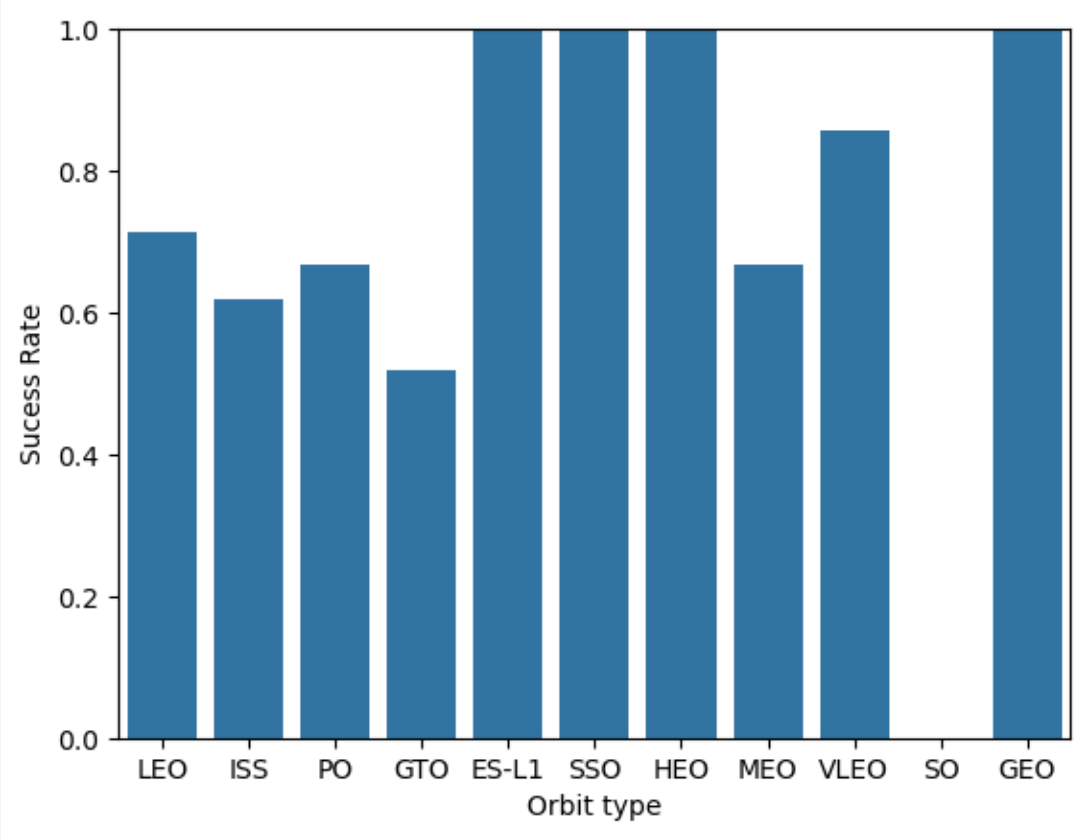
For each site, success rate tends to increase as the more 'flights' occur.

Payload vs. Launch Site



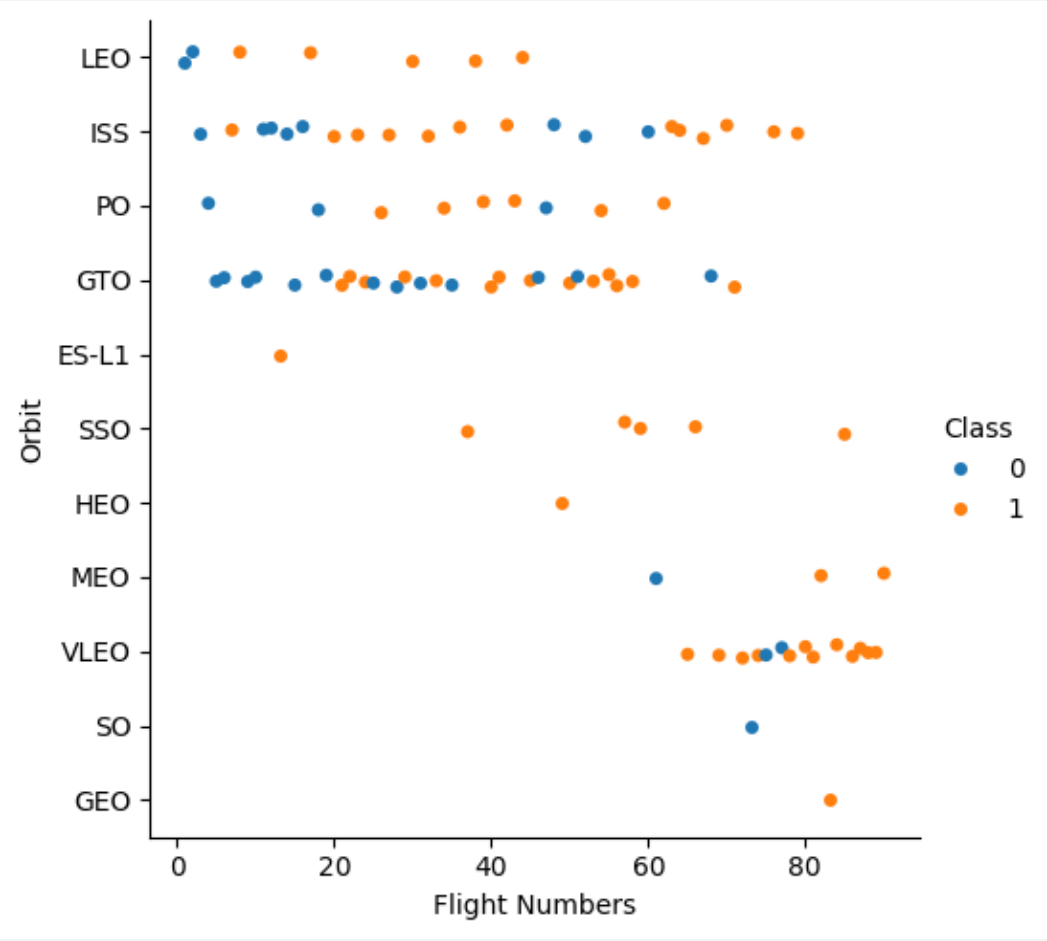
For each launch site, we observe a trend where successful landings increase as the payload mass is greater.

Success Rate vs. Orbit Type



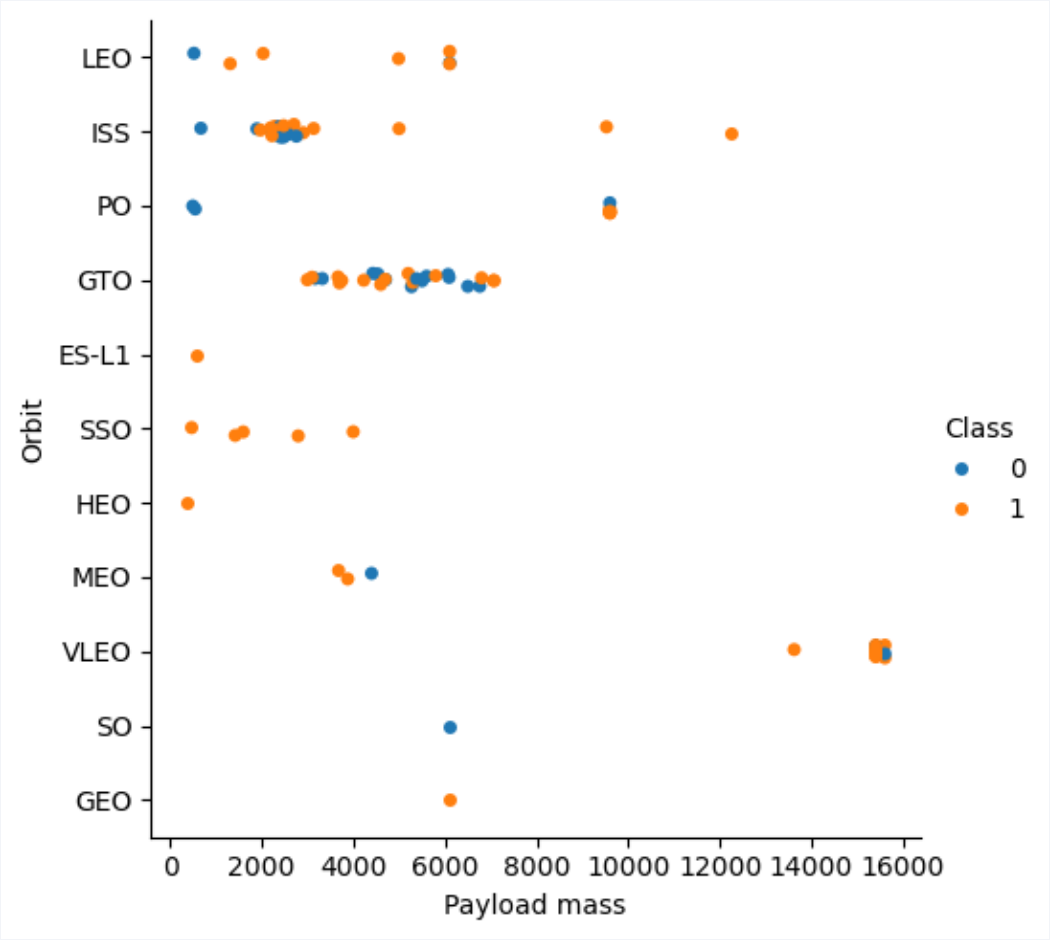
Perfect success rate for geostationary-type orbits

Flight Number vs. Orbit Type



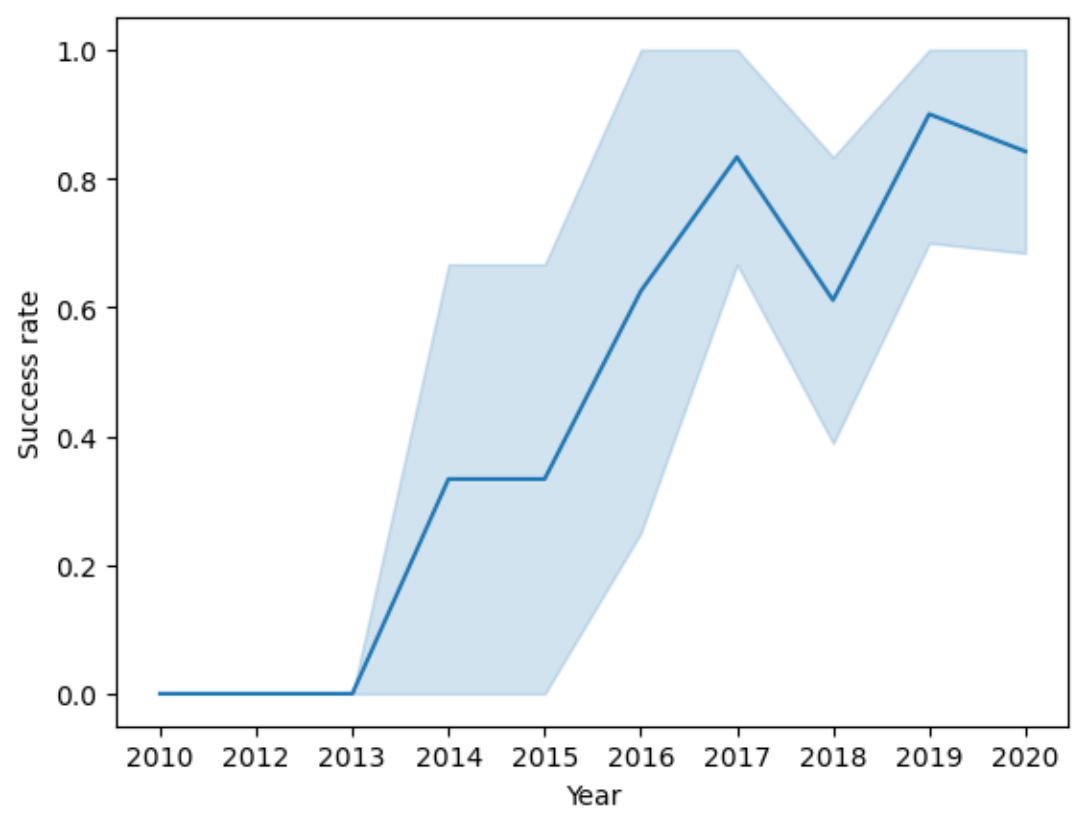
Plot shows an increase in success landings for the LEO orbit as more 'flights' occurred. For GTO, there is no clear insight. HEO, GEO and ES-L1 show only one 'flight' which is not enough information for this orbits even though they were all successful landings.

Payload vs. Orbit Type



Greater payload mass correlates to a positive outcome for the LEO orbit. For GTO, there is no clear trend.

Launch Success Yearly Trend



Graph shows no successful landings happened before 2013. After that, the rate increased significantly peaking in 2019.

All Launch Site Names

```
%sql select distinct Launch_Site from SPACEXTBL
```

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

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

We used the ‘DISTINCT’ statement in the query to extract the unique launch site names.

Launch Site Names Begin with 'CCA'

```
%sql select * from SPACEXTBL where Launch_site like 'CCA%' limit 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677

The 'like' statement followed by 'CCA%' restrict the search to the desired outcome.

Total Payload Mass

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer like '%CRS%'
```

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

sum(PAYLOAD_MASS__KG_)
48213

Average Payload Mass by F9 v1.1

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

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

avg(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

```
%sql select min(Date) from SPACEXTBL where Landing_Outcome='Success (ground pad)'
```

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

min(Date)
2015-12-22

The first event is selected with the ‘minimum date’. The ‘where’ clause restricts the query results.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select distinct Booster_Version from SPACEXTBL where Landing_Outcome='Success (drone ship)'
    and PAYLOAD_MASS__KG_>4000 and PAYLOAD_MASS__KG_<6000
```

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

```
%sql select Mission_Outcome, count (*) from SPACEXTBL group by Mission_Outcome
```

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

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

The group by statement makes it possible to display each distinct mission outcome and its total values. Note this is mission outcome and not landing outcome, which is what we are trying to predict.

Boosters Carried Maximum Payload

```
%sql select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_=
    (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

It was necessary to use a subquery to find the maximum payload mass and use it as a constraint in the main query.

2015 Launch Records

```
%%sql select strftime('%m', Date) as Month_name, Landing_Outcome, Booster_Version, Launch_Site
from SPACEXTBL sql where substr(Date,1,4)='2015' and Landing_Outcome='Failure (drone ship)'
```

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

Month_name	Landing_Outcome	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

The function substr allows us to select part of the date such as the year, month or day

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

```
%%sql select Landing_Outcome, count(*) from SPACEXTBL where(DATE(Date) between DATE('2010-06-04')
and DATE('2017-03-20')) group by Landing_Outcome order by count(*) desc
```

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

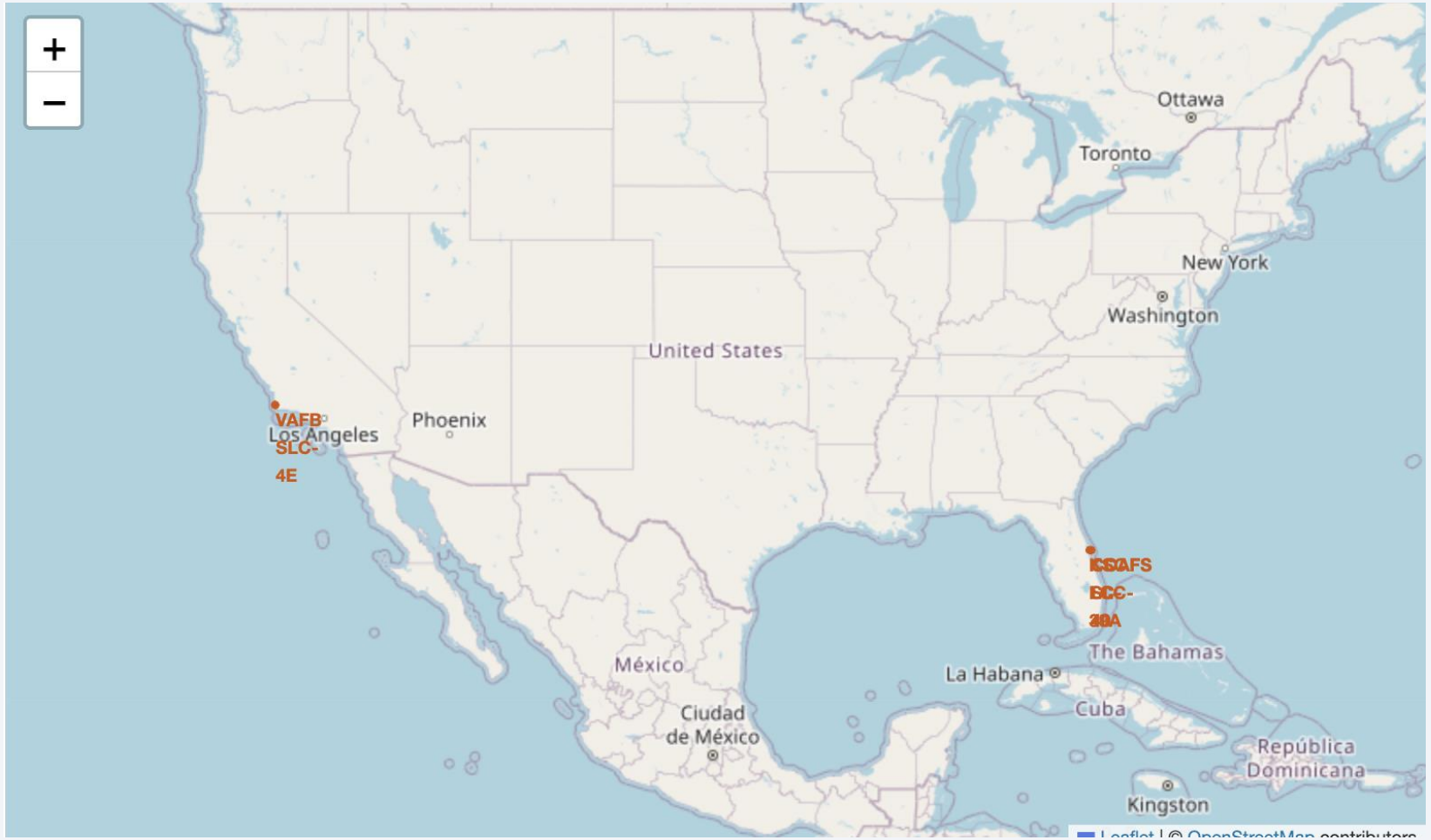
Landing_Outcome	count(*)
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

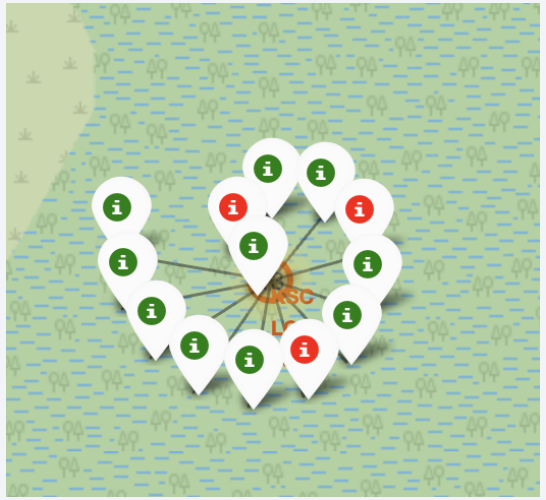
Launch Sites Proximities Analysis

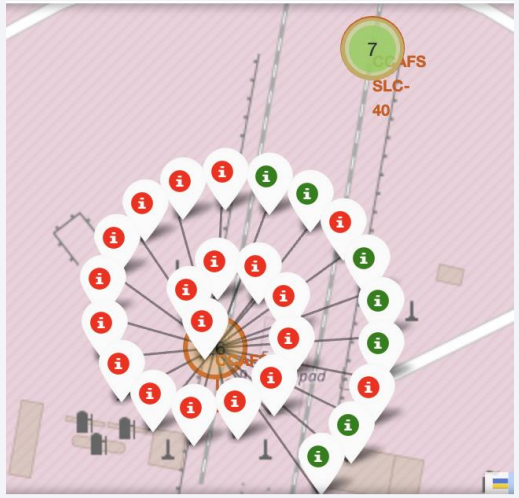
Folium map – All launch sites

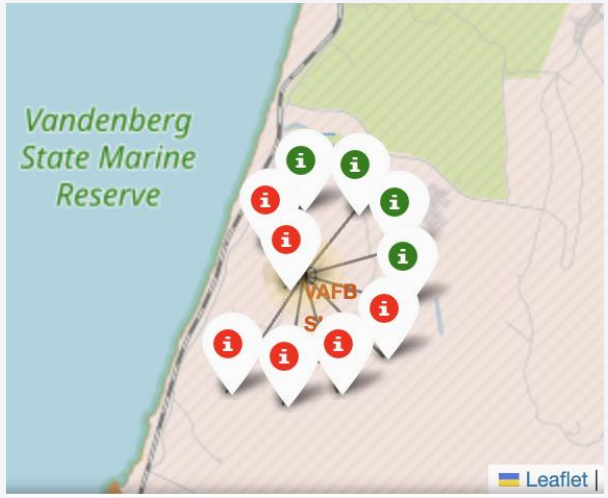
Map displays the 4 launch locations marked with an orange circle. One in California, and 3 in Florida.



Folium Map – launches for all sites

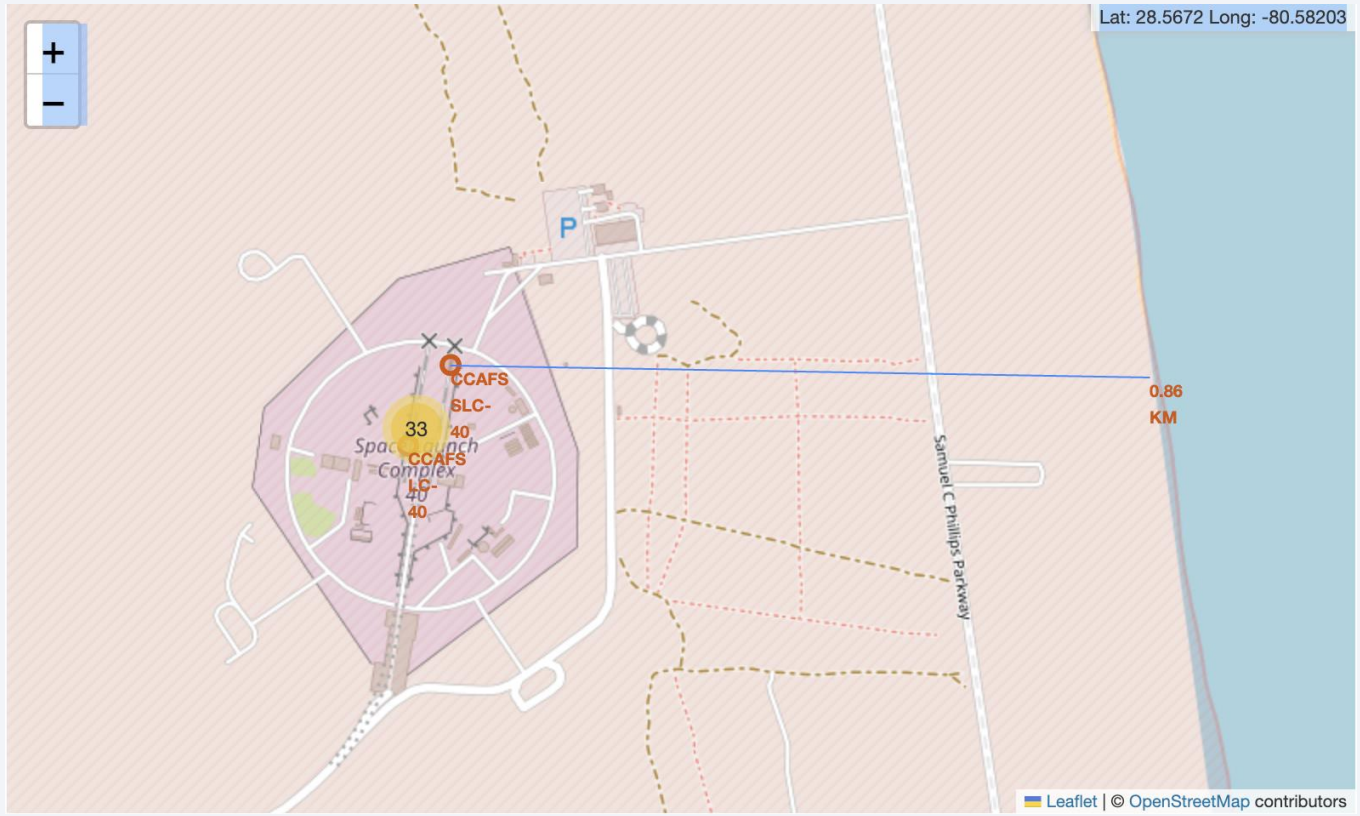




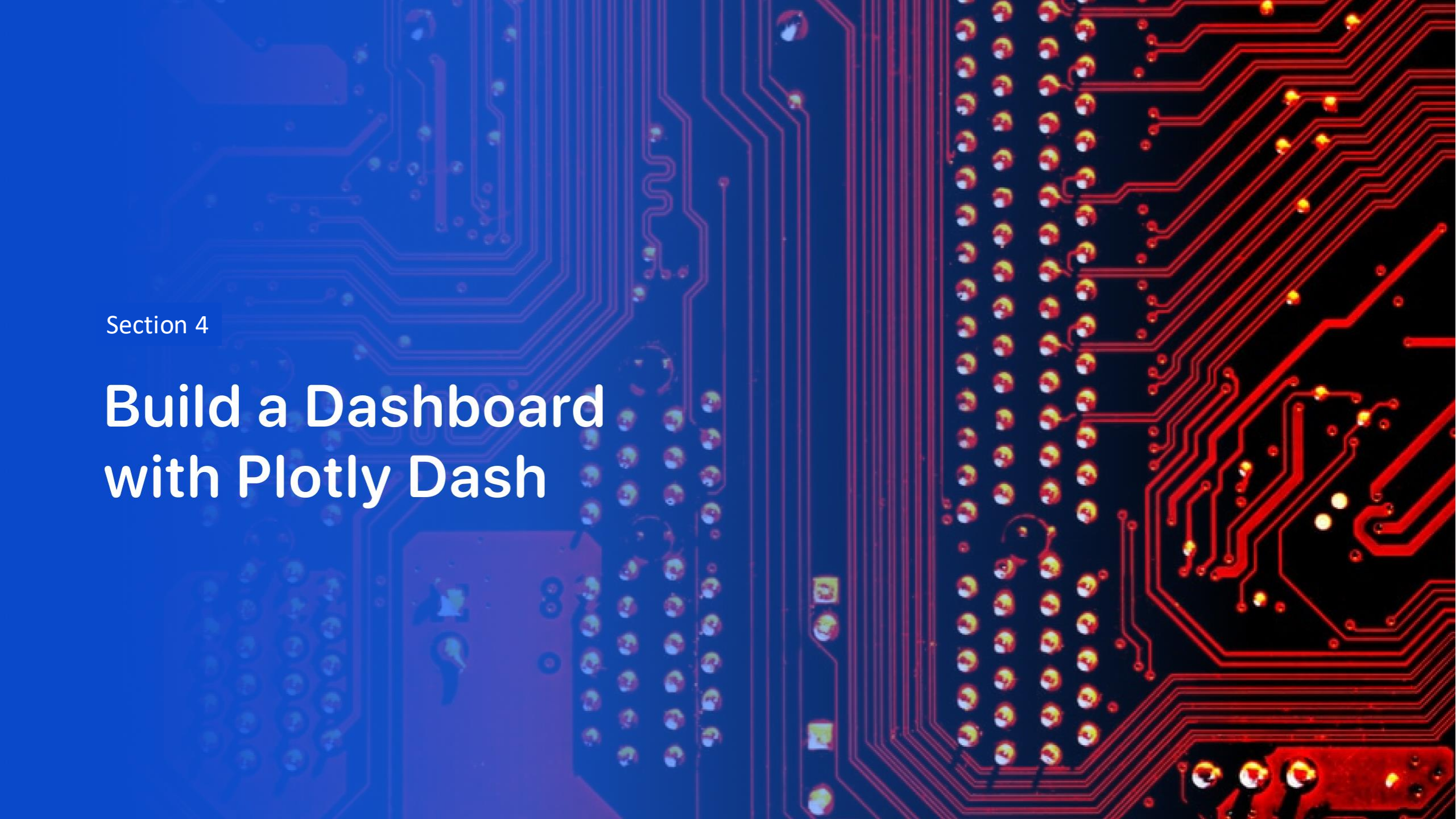


A green mark indicates a succesful launch while a red one indicates a failure.

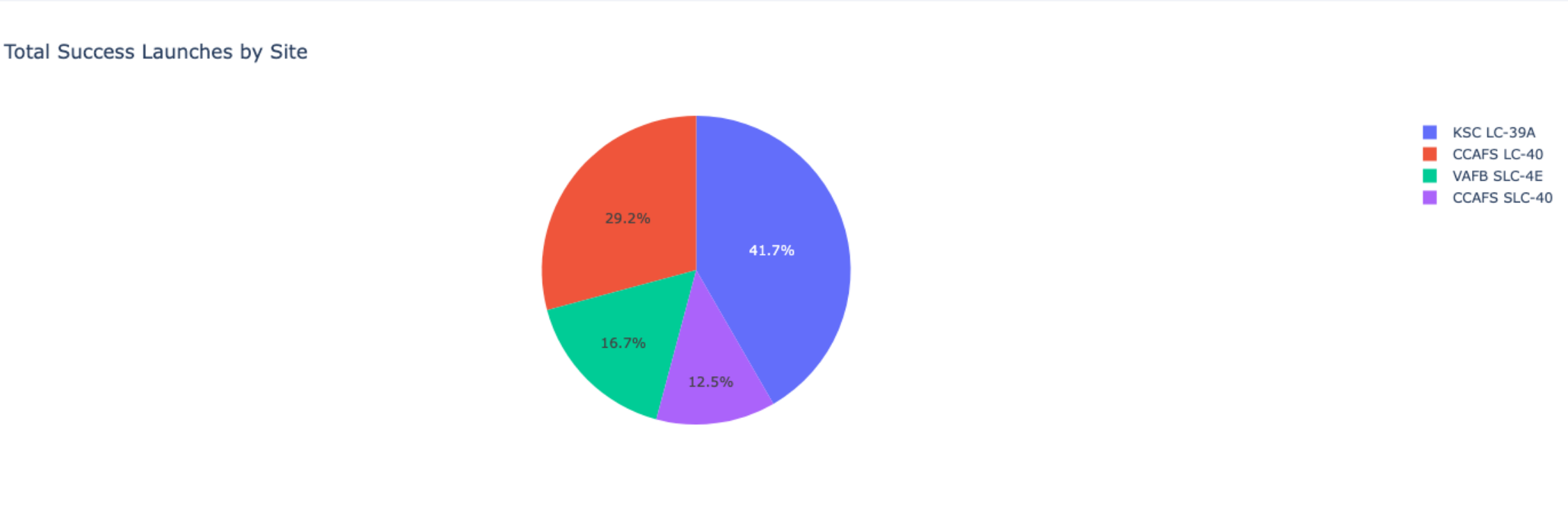
<Folium Map Screenshot 3>



CCAFS SLC-40 launch site proximity to a coastline point. Distance is shown to be 0.86 km.

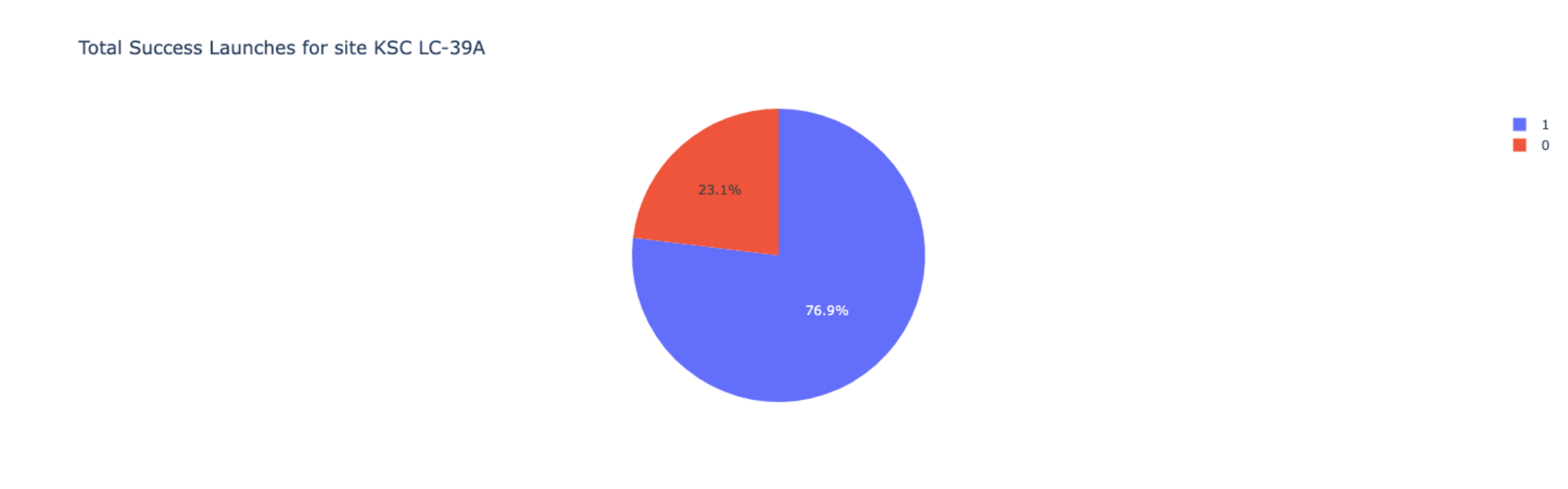

Section 4
Build a Dashboard with Plotly Dash

Pie chart of successful launches by site



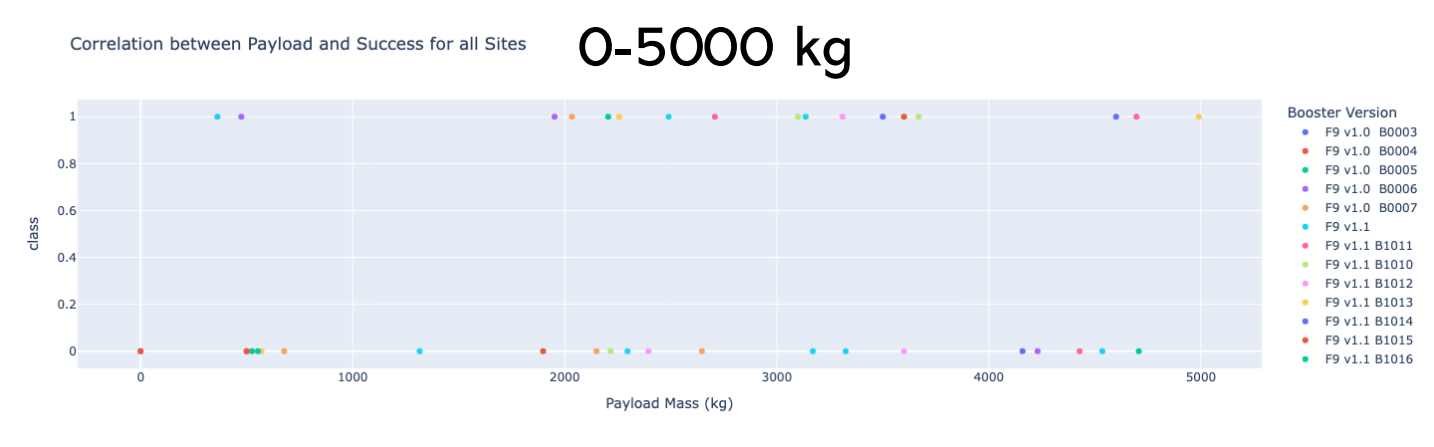
Site KSC LC-39A has the largest share of successful launches.

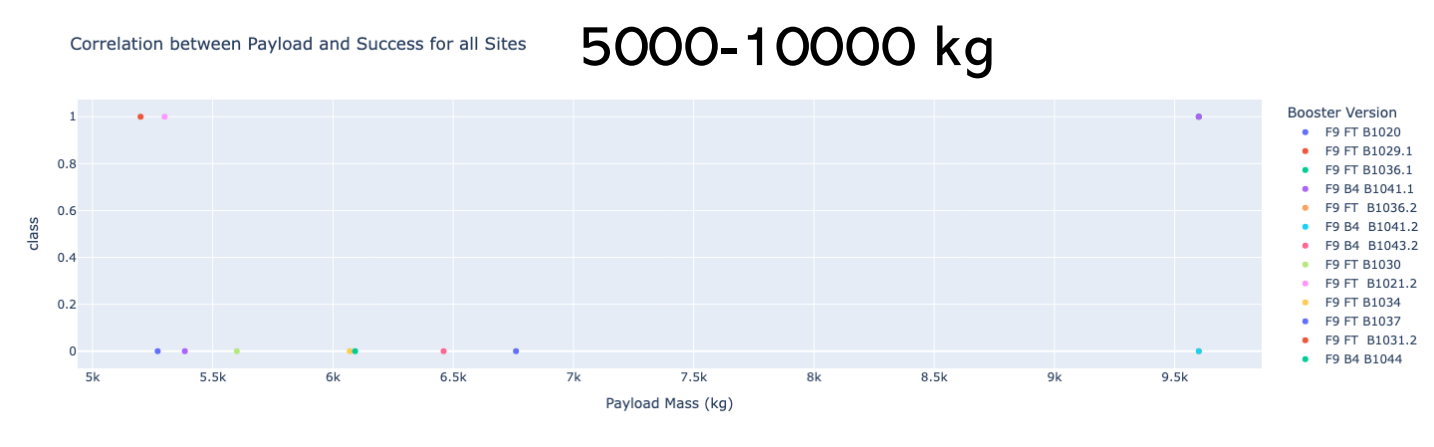
Success rate of KSC-LC39A



We observe a success rate of 76.9%.

Payload mass vs Outcome for all sites



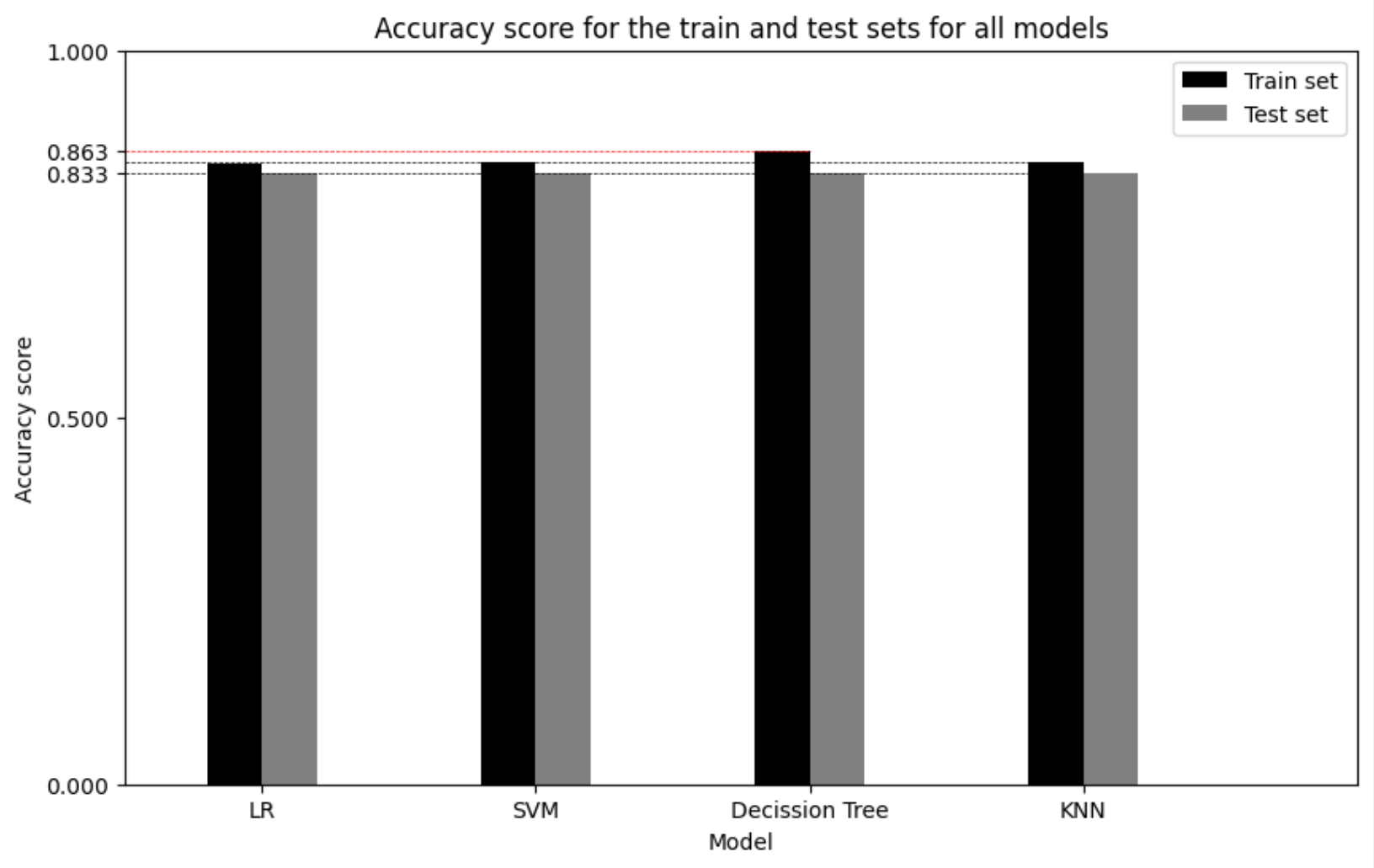


The graphs show that heavier payloads tend to result in higher mission failure percentage.

Section 5

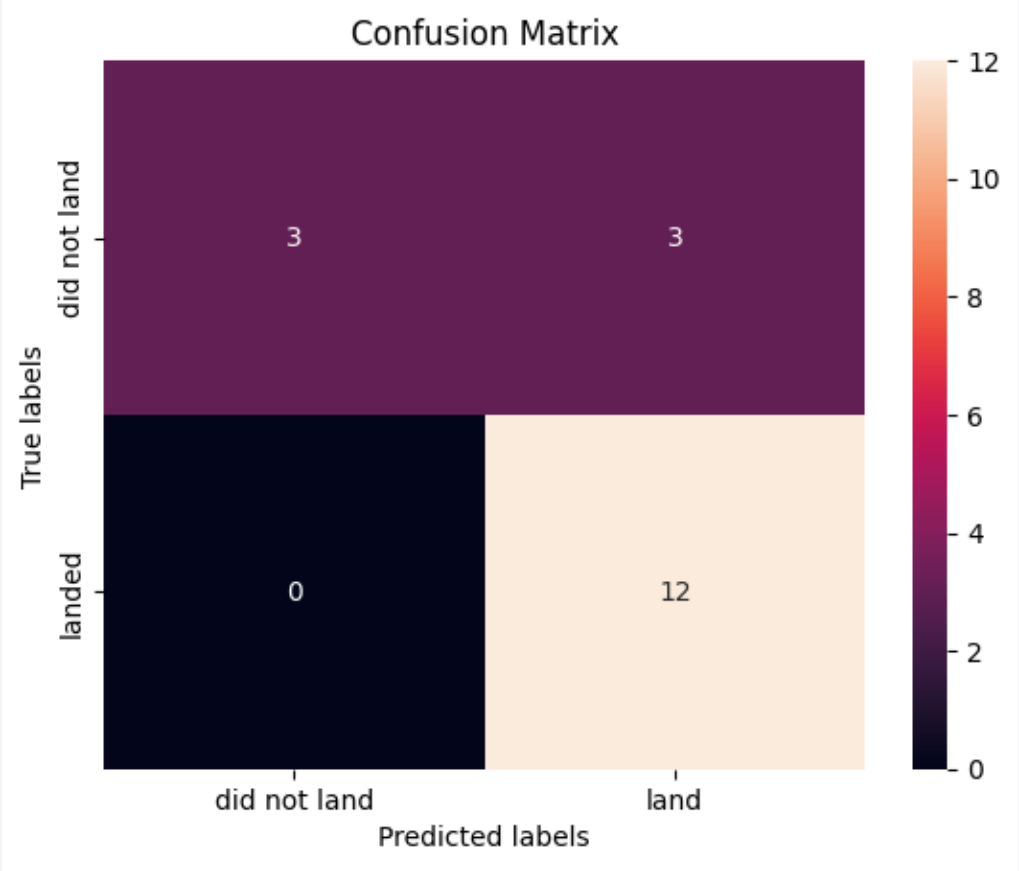
Predictive Analysis (Classification)

Classification Accuracy



We observe that the testing accuracy is the same for all models. In consequence, we select the best model based on the accuracy score for the training set and that is decision tree classifier which has a 86.3% precision.

Confusion Matrix



The confusion matrix for the decision tree classifier is the same as the other models because the accuracy score on the test data is equal for all models.

Conclusions

- It is possible to predict the the outcome with acceptable accuracy based on the parameters selected. Nevertheless, training data is limited. Given the nature of the business, the number of launches is limited.
- It is necessary to refine the selected model as more launches take place.
- Payload mass is a good indicator of successful or failed landing as these variables show some correlation.
- Success rate has increased over the last ten years, so as more experience is gained after each launch, the mission outcome will success rate will also tend to increase.

Thank you!