IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Javier Eusamio
31/03/2024

Outline

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

Executive Summary

- Data from SpaceX rocket launches has been collected and analyzed. Data has been gathered from the SpaceX API and scraped from Wikipedia. It has been processed, visualized with Jupyter Notebooks and dashboards, and used to train four machine learning models.
- All four models (logistic regression, SVM, decision tree classifier, and KNN) show an accuracy of 83.3 % when using train-test split data. The test dataset was very small, which could have probably impacted the accuracy.

Introduction

- SpaceX has been developing commercial spaceships in the past years. We want to develop SpaceY, a competing company with a similar business strategy.
- We want to use machine learning to predict the conditions in which the rocket will be successfully recovered, saving the business a lot of money.

Section 1

Methodology

Methodology

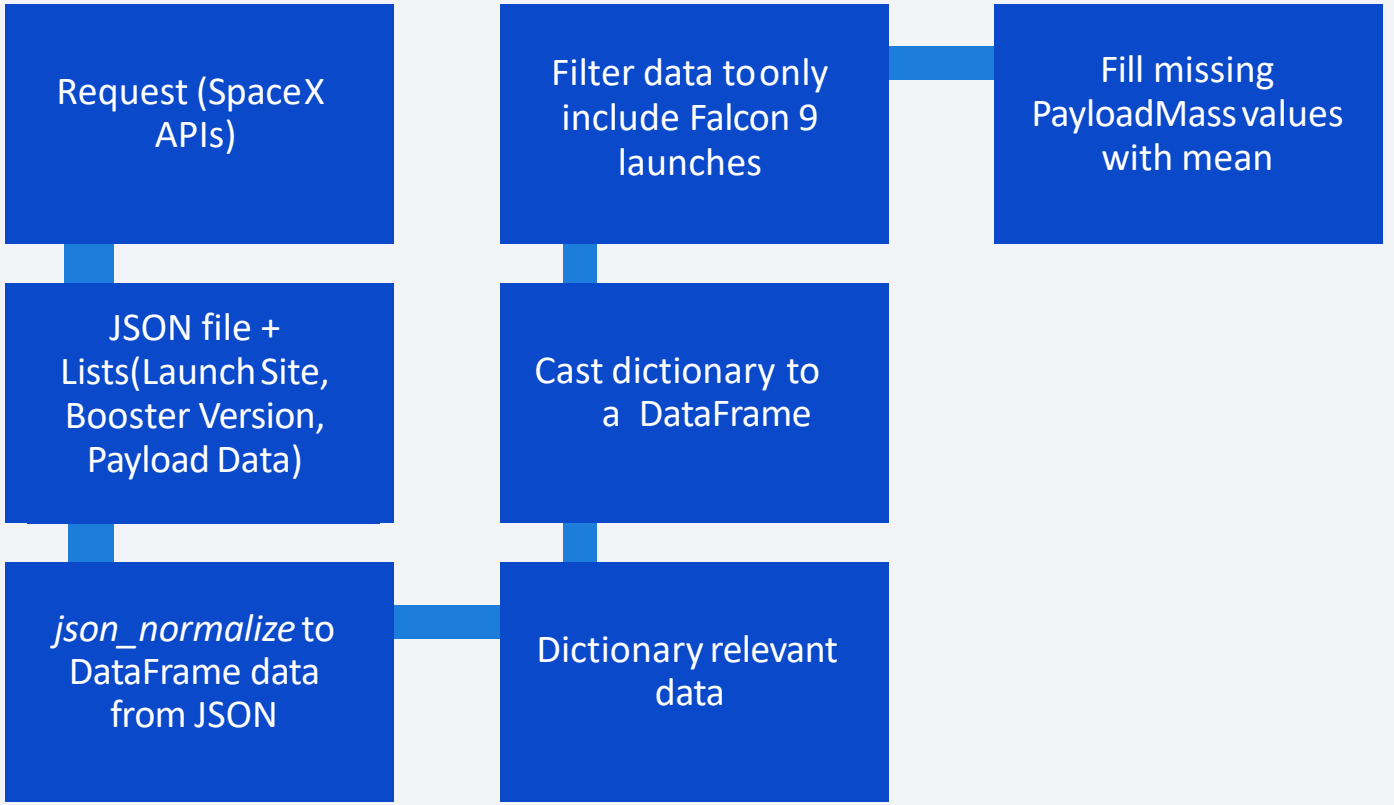
Executive Summary

- Data collection methodology:
  - Collected from scraping Wikipedia and from the SpaceX public API
- Perform data wrangling
  - Landings were classified numerically as successful or unsuccessful
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Found the best parameters using GridSearchCV

Data Collection

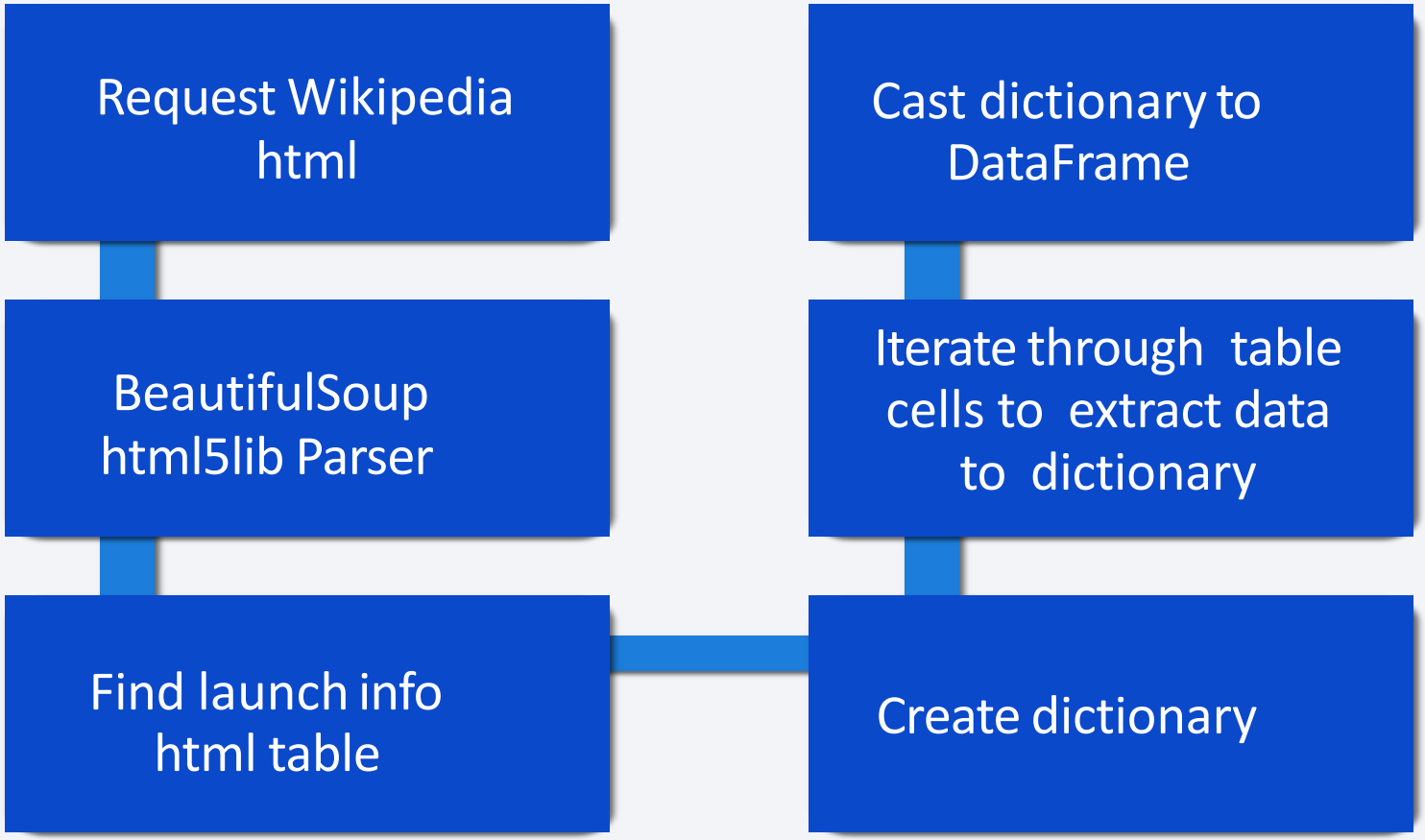
- Data was collected through the SpaceX API and scraping the Wikipedia SpaceX page.
- **Space X API Data Columns**: FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude
- **Wikipedia Webscrape Data Columns**: Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time

Data Collection – SpaceX API



GitHub URL: [ibm_data_science/1_introduction/jupyter-labs-spacex-data-collection-api.ipynb at main · jaymaeuro/ibm_data_science (github.com)]

Data Collection - Scraping



GitHub URL: [ibm_data_science/1_introduction/jupyter-labs-webscraping.ipynb at main · jaymaeuro/ibm_data_science (github.com)]

Data Wrangling

- A “Class” column was created where the output of the landing was labeled 1 or 0 depending on whether it was a success or a failure, respectively.
- GitHub URL: ibm_data_science/1_introduction/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb at main · jaymaeuro/ibm_data_science (github.com)

EDA with Data Visualization

Exploratory Data Analysis performed on variables Flight Number, Payload Mass, Launch Site, Orbit, Class and Year.

Plots Used: 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.

Scatter plots, line charts, and bar plots were used to compare relationships between variables to decide if a relationship exists so that they could be used in training the machine learning model

GitHub url: ibm_data_science/2_EDA/jupyter-labs-eda-dataviz.ipynb at main · jaymaeuro/ibm_data_science (github.com)

EDA with SQL

The following data was analyzed via SQL queries through Python:

- Launch sites
- Boosters launched by NASA
- Payload mass
- Mission outcome

Github URL: ibm_data_science/2_EDA/jupyter-labs-eda-sql-edx_sqllite.ipynb at main · jaymaeuro/ibm_data_science (github.com)

Build an Interactive Map with Folium

- Folium maps mark Launch Sites, successful and unsuccessful landings, and a proximity example to key locations: Railway, Highway, Coast, and City.
- This allows us to understand why launch sites may be located where they are. Also visualizes successful landings relative to location.

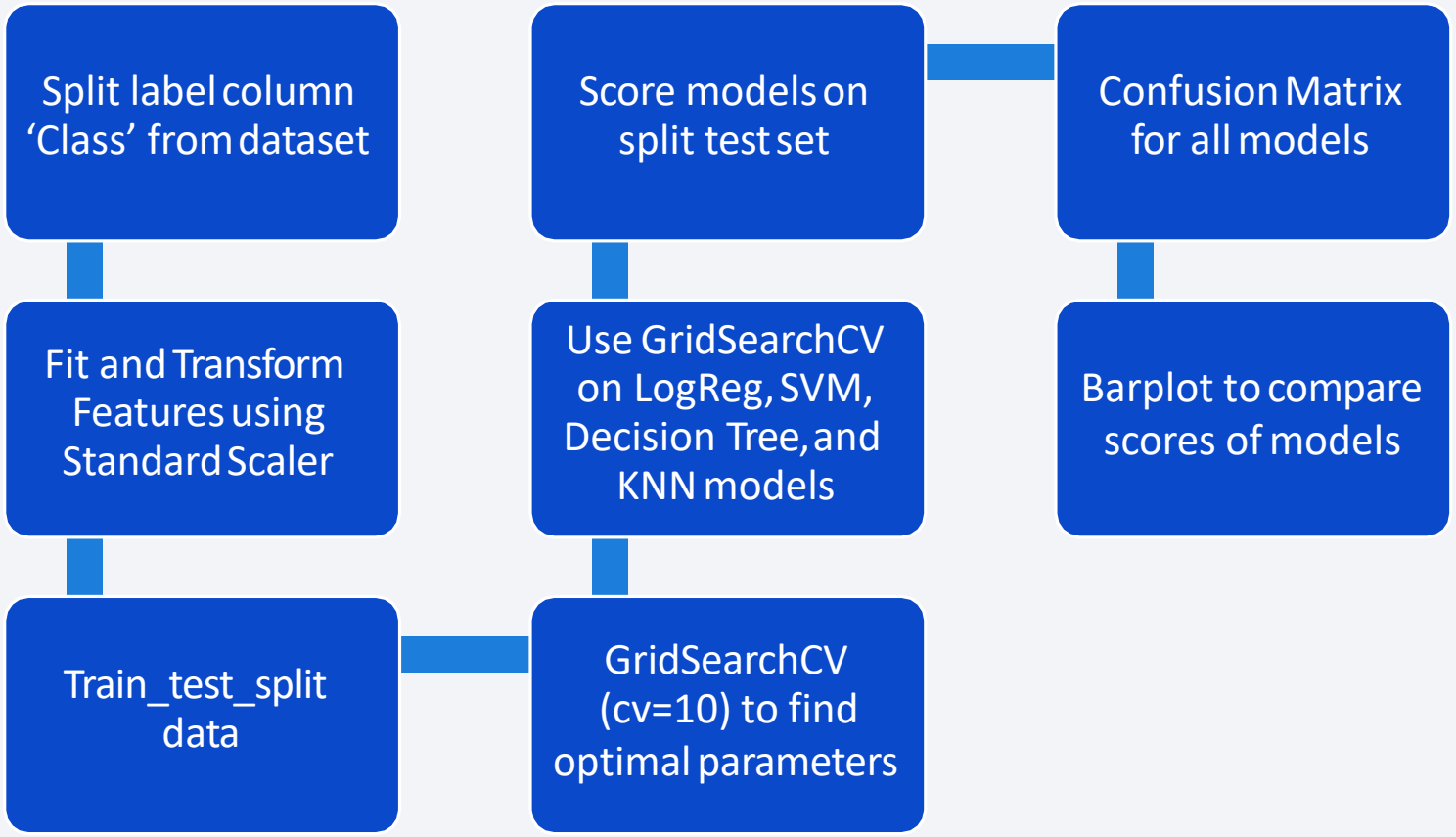
Github URL: ibm_data_science/3_dashboard/lab_jupyter_launch_site_location.ipynb at main · jaymaeuro/ibm_data_science (github.com)

Build a Dashboard with Plotly Dash

- Dashboard includes a pie chart and a scatter plot.
- Pie chart can be selected to show distribution of 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 on a slider between 0 and 10000 kg.
- The pie chart is used to visualize launch site success rate.
- The scatter plot can help us see how success varies across launch sites, payload mass, and booster version category.

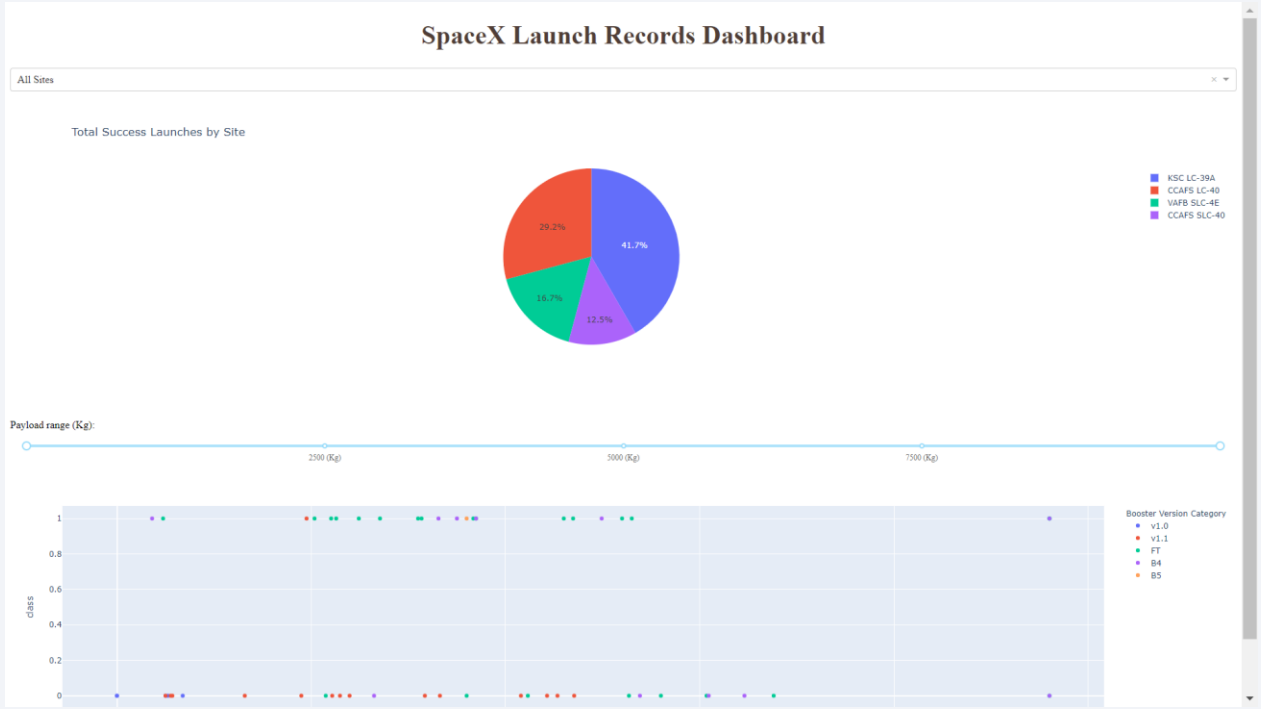
Github URL: ibm_data_science/3_dashboard/spacex_dash_app.py at main · jaymaeuro/ibm_data_science (github.com)

Predictive Analysis (Classification)



GitHub URL:
ibm_data_science/4_predictive_analysis/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb at main · jaymaeuro/ibm_data_science (github.com)

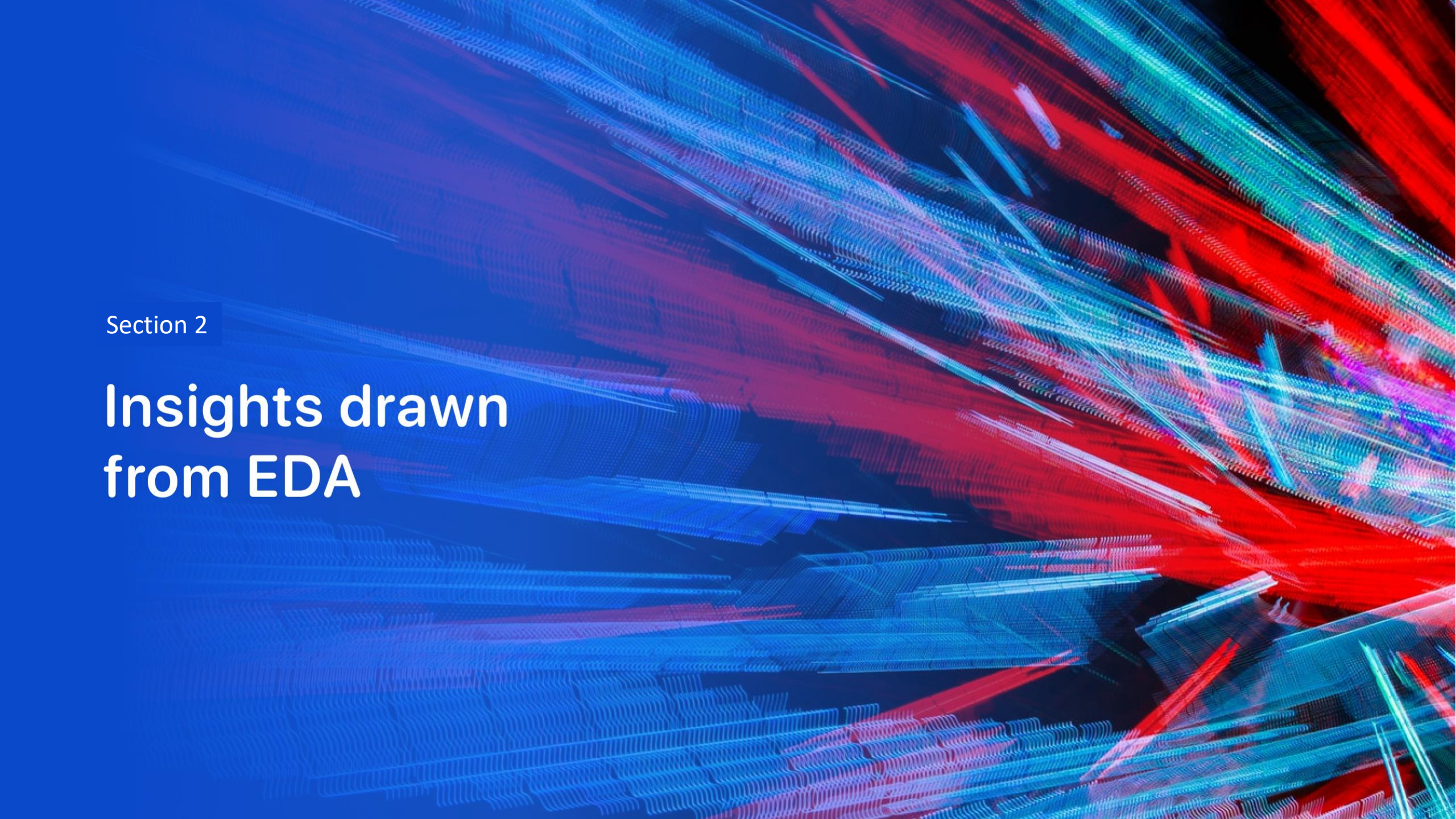
Results



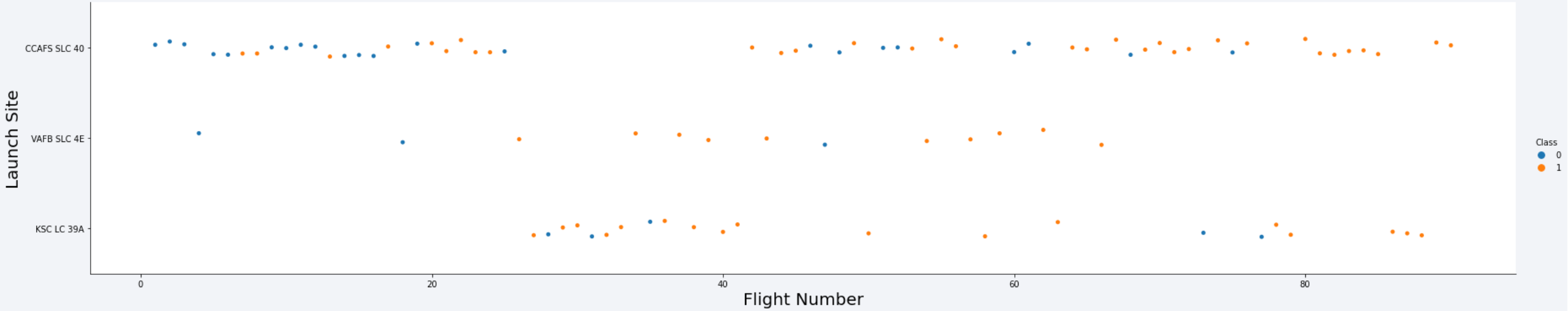
A Plotly dashboard has been built to interactively analyze the data.

Data has been analyzed with SQL and Python.

Four machine learning models have been trained to make predictions with an 83.3 % accuracy.

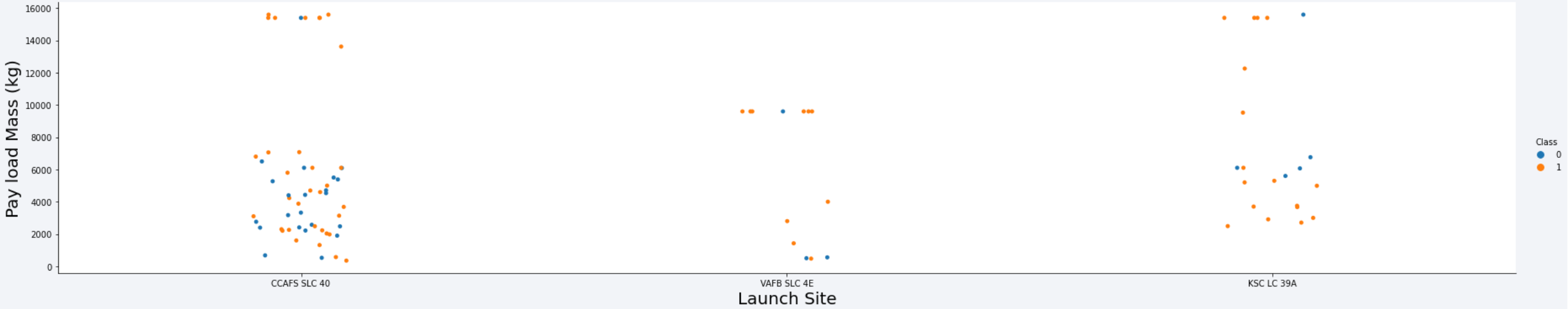

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



- Orange represents a successful landing, and blue an unsuccessful one.

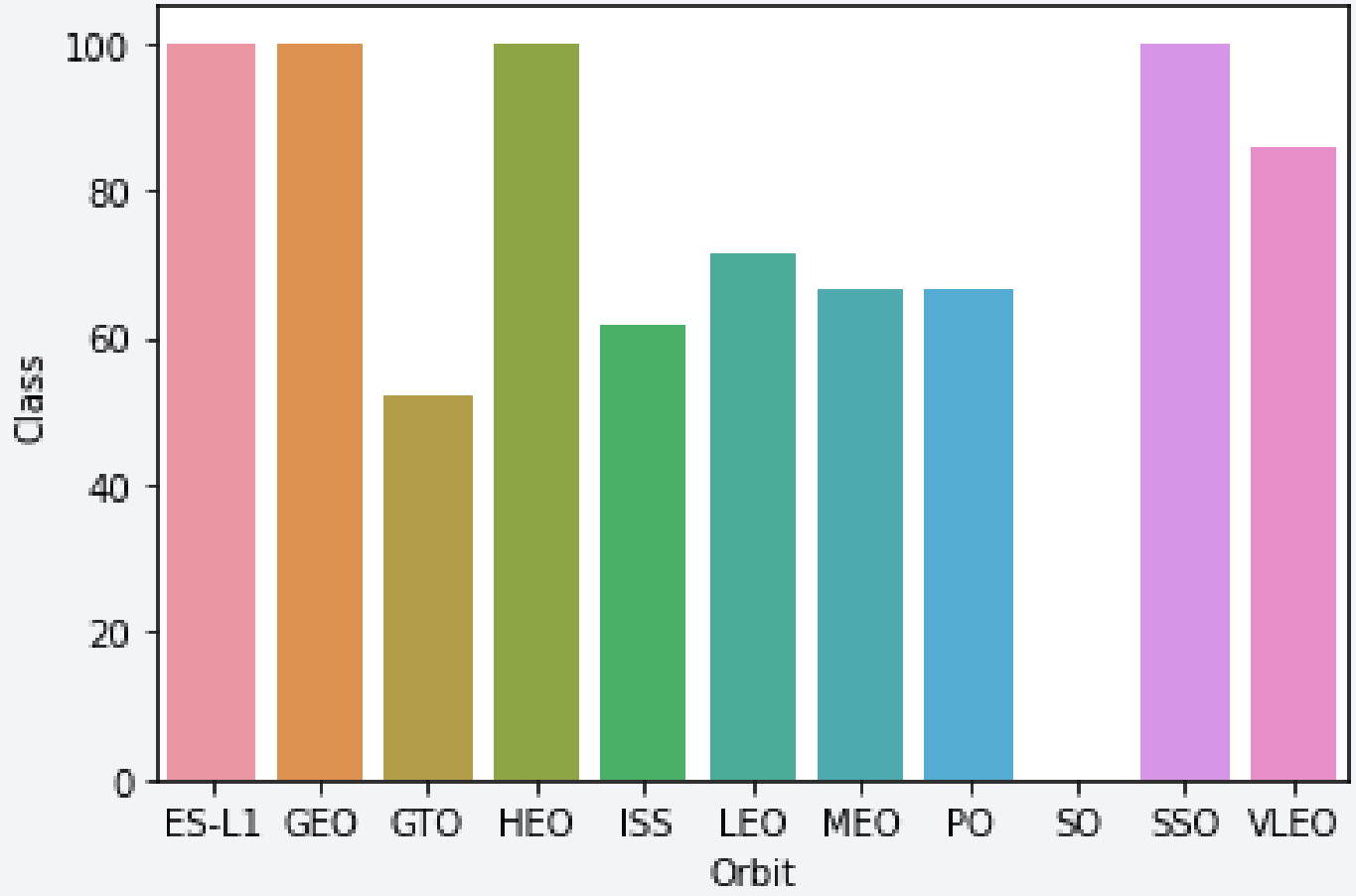
Payload vs. Launch Site



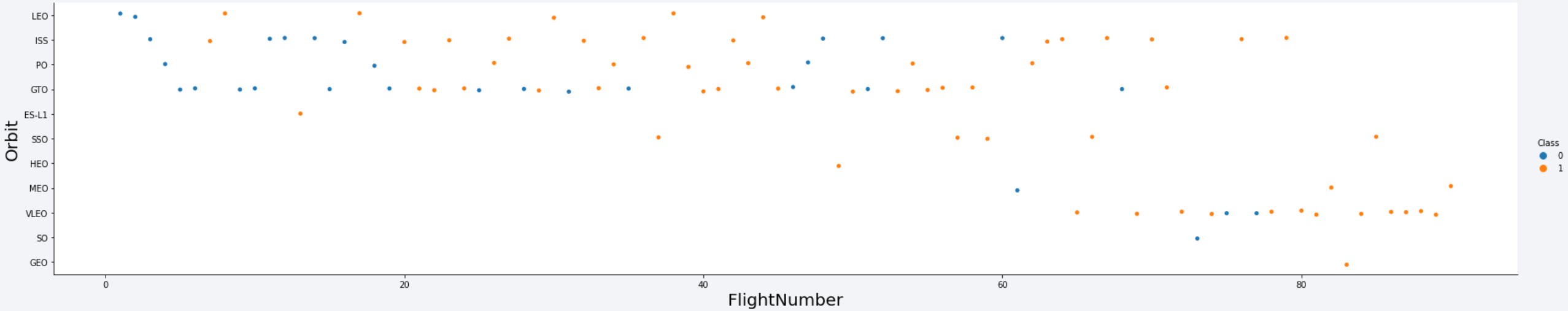
- Orange represents a successful landing, and blue an unsuccessful one.

Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, and SSO are the most successful orbit types.

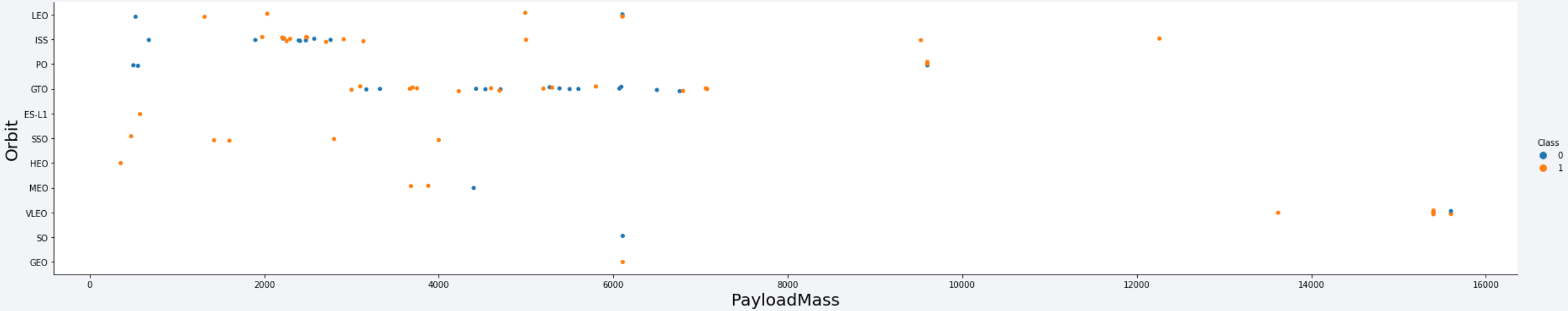


Flight Number vs. Orbit Type



- Orange represents a successful landing, and blue an unsuccessful one.

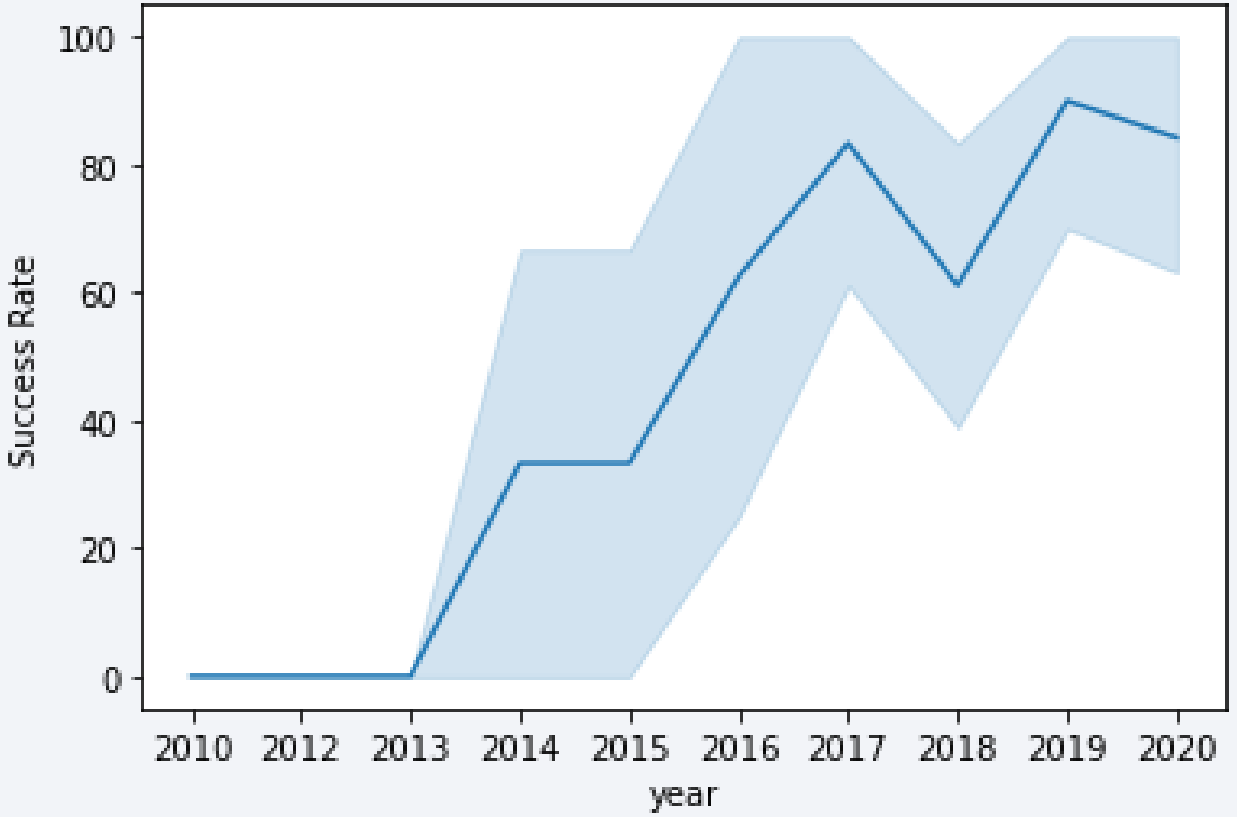
Payload vs. Orbit Type



- Orange represents a successful landing, and blue an unsuccessful one.

Launch Success Yearly Trend

- The sucess rate since 2013 kept increasing till 2020



All Launch Site Names

- There are 5 unique launch sites.

```
%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

Launch Site Names Begin with 'KSC'

```
%sql select * from SPACEXTBL where launch_site like 'KSC%' limit 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-03-16	6:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
2017-05-15	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

- The first 5 results matching the query are displayed.

Total Payload Mass

- Total payload calculated using the sum function.

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

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

sum
45596

Average Payload Mass by F9 v1.1

- Query for finding the average payload mass.

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

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

Average
2534.6666666666665

First Successful Ground Landing Date

- The date was June 4th 2010.

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

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

Date
2010-06-04

Successful Drone Ship Landing with Payload between 4000 and 6000

- There are only two boosters with those characteristics.

```
%sql select booster_version from SPACEXTBL where (mission_outcome like 'Success') AND (payload_mass__kg_ BETWEEN 4000 AND 6
```

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

Booster_Version
F9 FT B1032.1
F9 B4 B1040.1

Total Number of Successful and Failure Mission Outcomes

- 100 successful missions and 1 failure.

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

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

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

Boosters Carried Maximum Payload

- Obtained using a compound query

```
%sql select booster_version from SPACEXTBL where payload_mass__kg_=(select max(payload_mass__kg_) from SPACEXTBL)
```

```
 * sqlite:///my_data1.db
Done.
 * sqlite:///my_data1.db
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 of the records for the months in year 2017, which display the month names, successful landing outcomes in ground pad, booster versions and launch site

```
%sql select substr(Date,6,2) as Month, Landing_Outcome, booster_version, launch_site from SPACEXTBL where DATE like '2017%'
```

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

Month	Landing_Outcome	Booster_Version	Launch_Site
02	Success (ground pad)	F9 FT B1031.1	KSC LC-39A
05	Success (ground pad)	F9 FT B1032.1	KSC LC-39A
06	Success (ground pad)	F9 FT B1035.1	KSC LC-39A
08	Success (ground pad)	F9 B4 B1039.1	KSC LC-39A
09	Success (ground pad)	F9 B4 B1040.1	KSC LC-39A
12	Success (ground pad)	F9 FT B1035.2	CCAFS SLC-40

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

- Obtained using a group by query.

```
%sql select Landing_Outcome, count(*) as count from SPACEXTBL where Date >= '2010-06-04' AND Date <= '2017-03-20' GROUP by L
```

```
 * 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

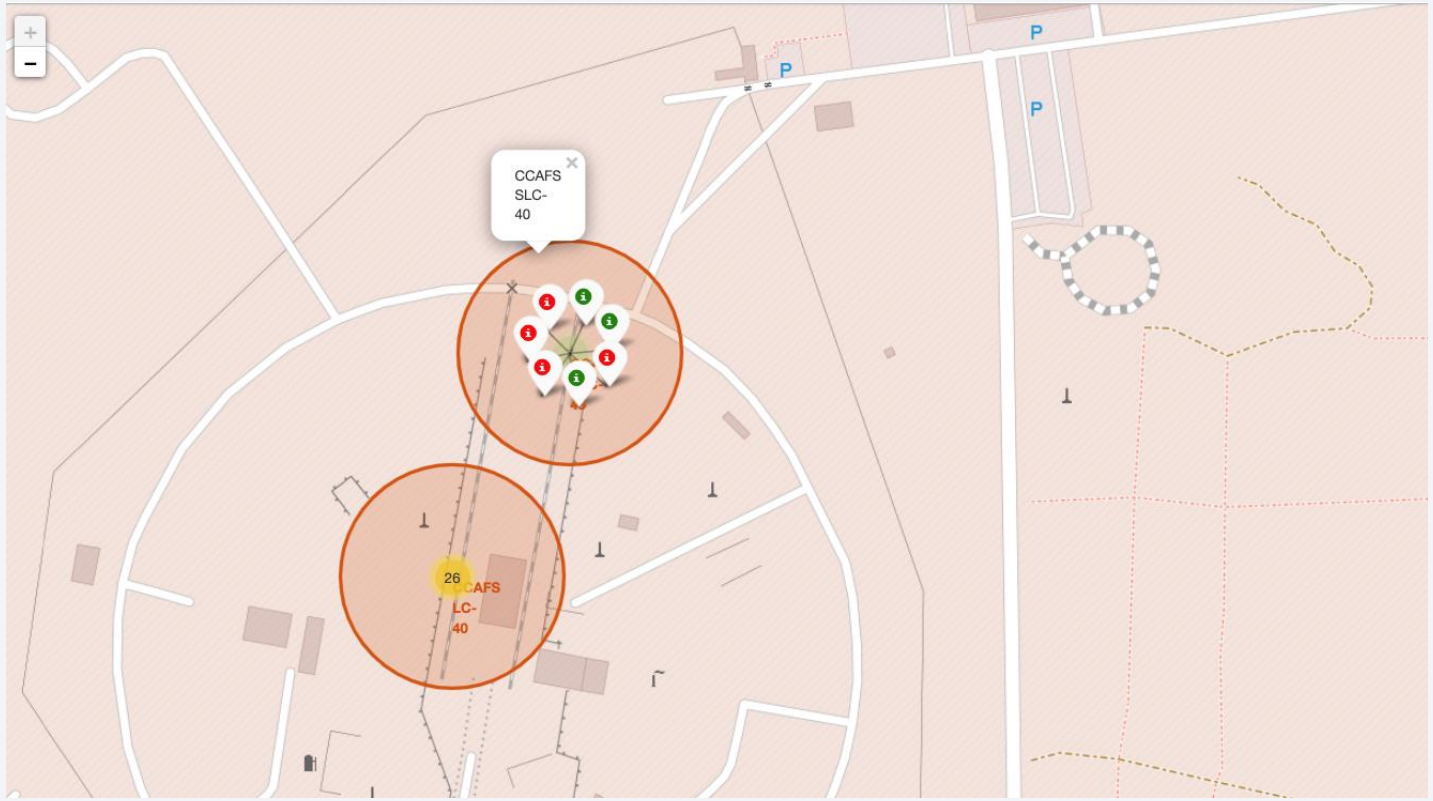
Launch Sites Proximities Analysis

Map with launch sites

- Folium map with markers on the different launch sites.

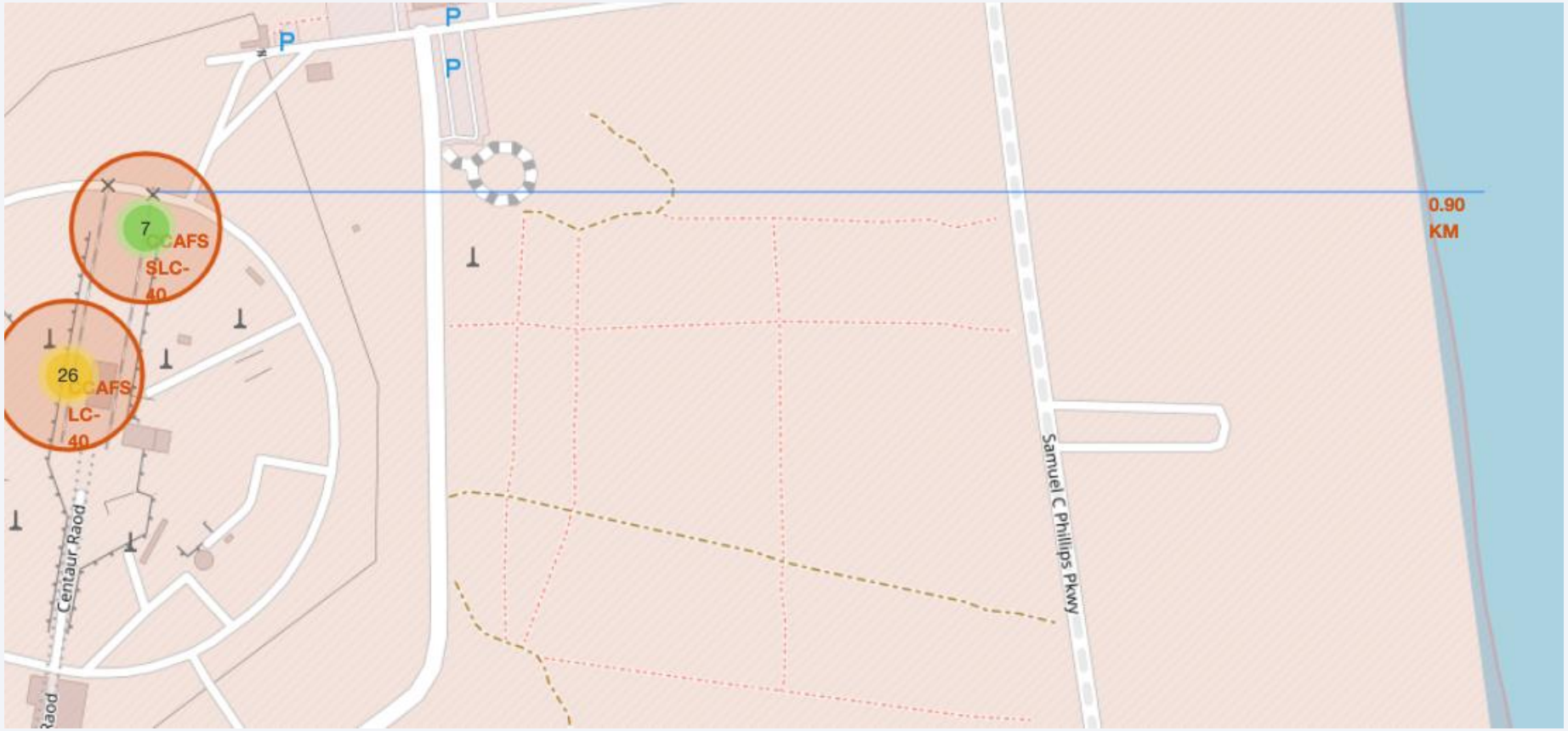
Clustering with marker colors

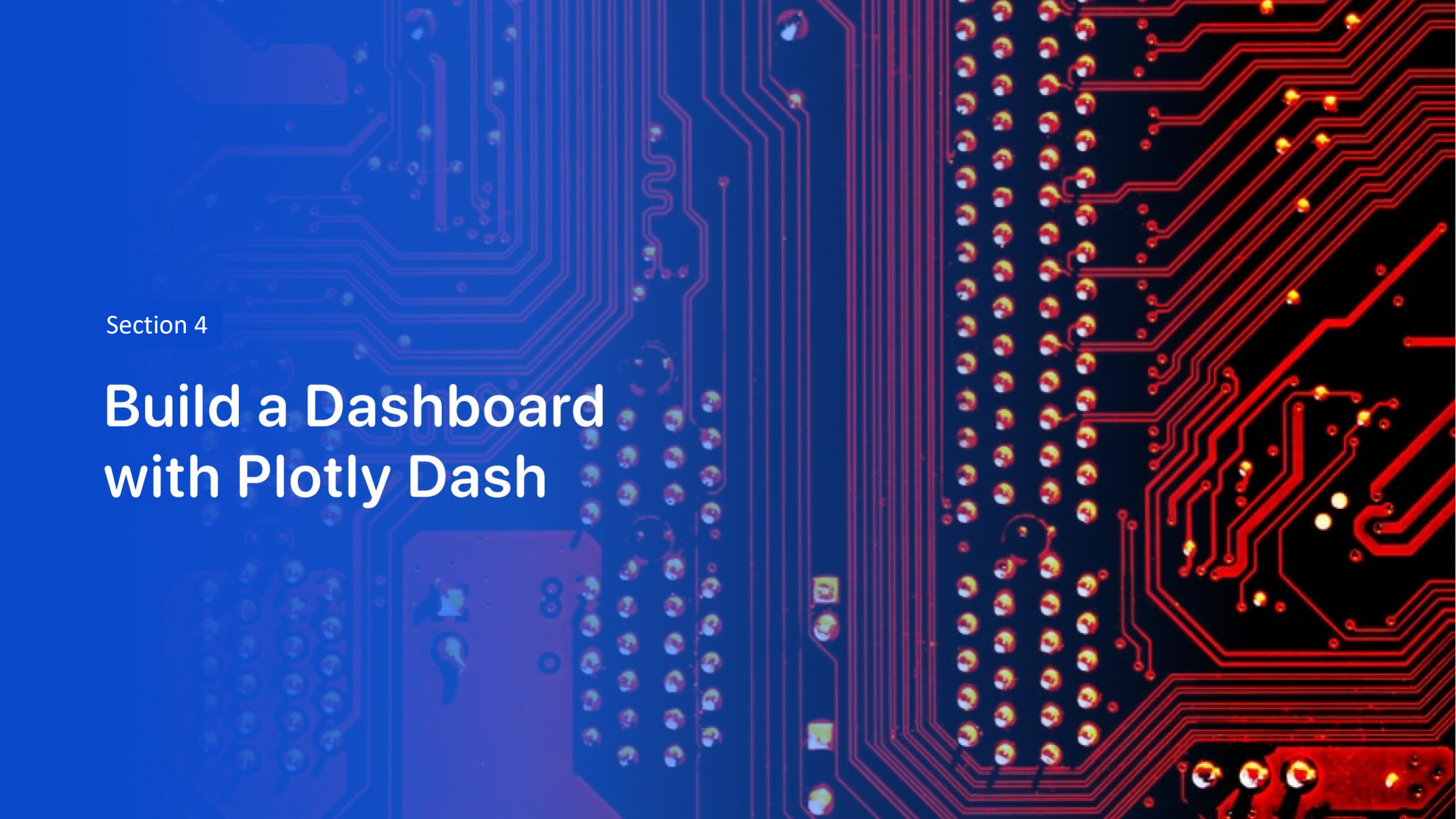
- Markers appear in green for successful landings and in red for failed ones.



Calculating distances

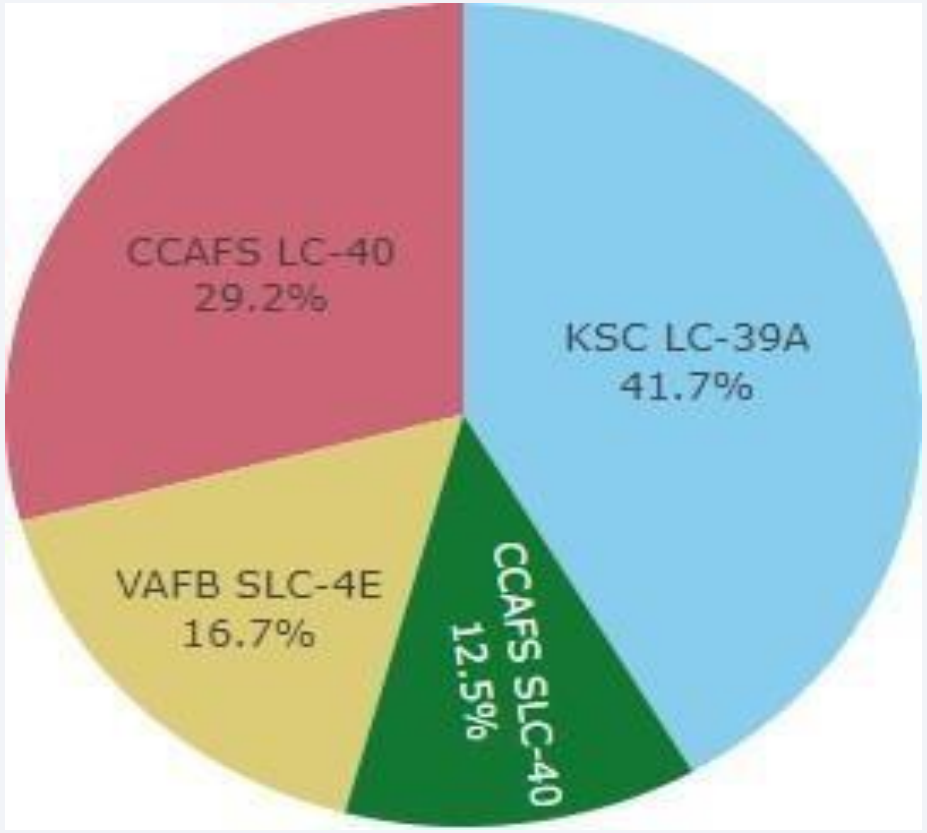
- Map showing the distance between a launch site and the coast.



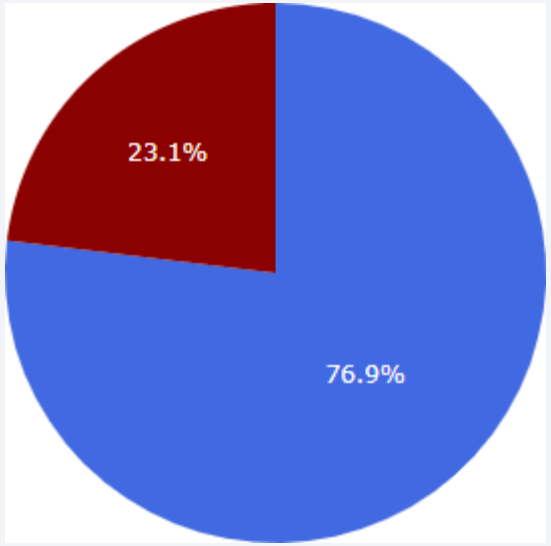

Section 4
Build a Dashboard with Plotly Dash

Piechart of successful launches

- KSC has the largest amount of successful launches



Most successful launch site



KSC LC-39A has the highest success rate with 10 successful landings and 3 failed landings.

Payload vs. Launch Outcome scatter plot

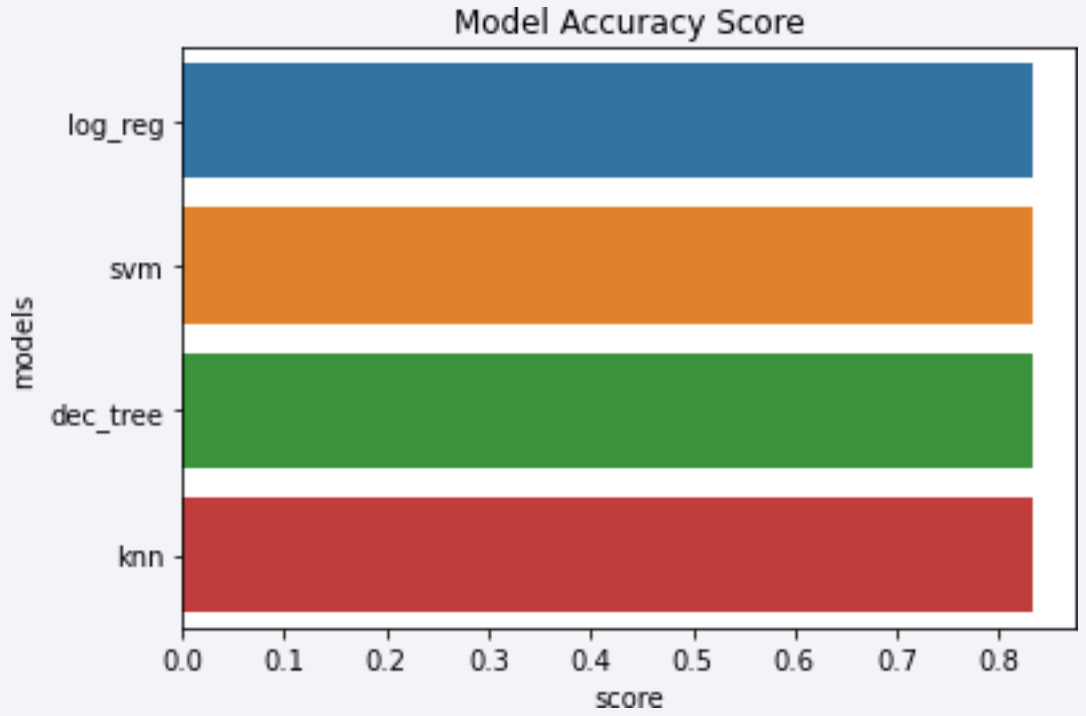
- In the image, max payload is set to 6000. The color corresponds to the launch site, the size to the number of launches, and the y value to the success state.

Section 5

Predictive Analysis (Classification)

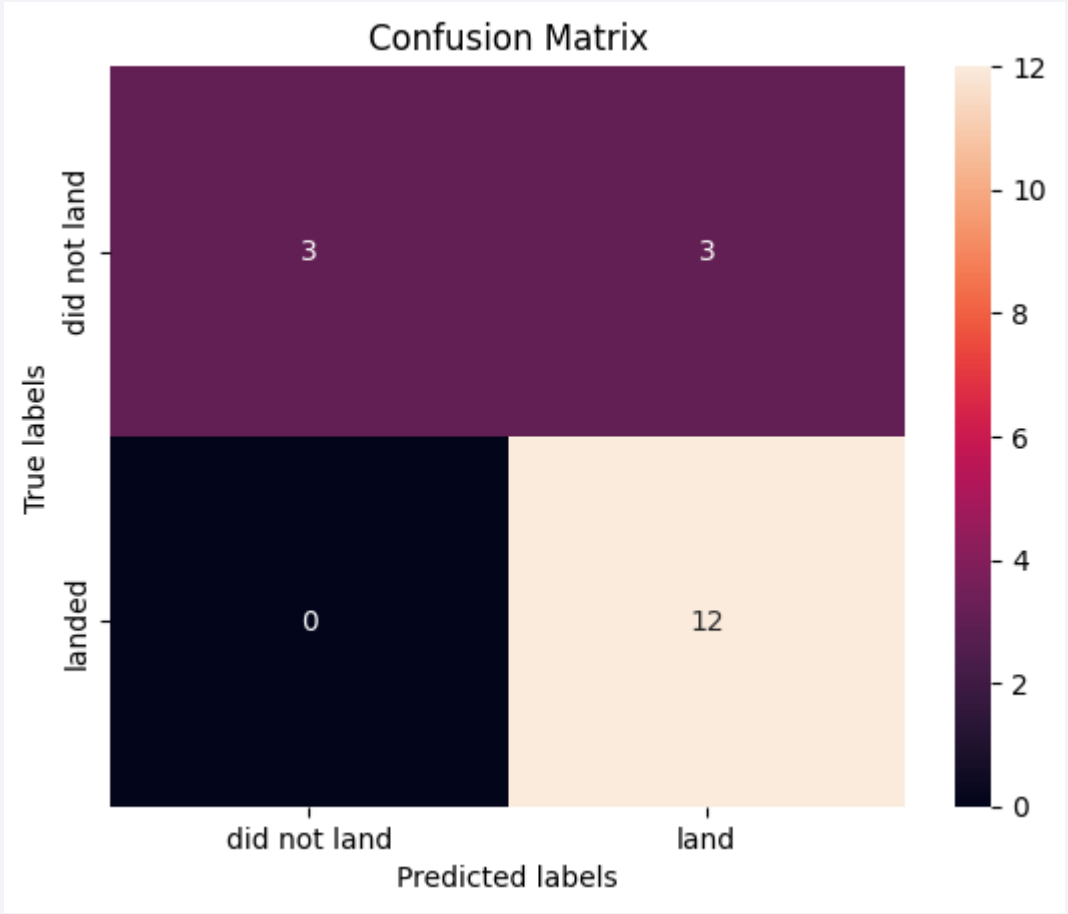
Classification Accuracy

- All models had the same accuracy when tested with the test dataset. This may be due to the reduced size of dataset (only 18 samples).



Confusion Matrix

- All models have the same accuracy and the same confusion matrix.
- The model performs well except for three false positive results.



Conclusions

- We have extracted the data from Wikipedia and the SpaceX public API and wrangled it to obtained our desired dataset.
- The data has been analyzed (EDA) through Python and SQL.
- Visualizations for the data have been created through several graphs (Matplotlib and Seaborn), a dashboard (Plotly Dash), and maps (Folium).
- The data has been used to train four machine learning models and make predictions about the output of future rocket launches.
- An accuracy of 83.3 % has been obtained for all four models.

Appendix

- All the code for the project can be found in the corresponding GitHub repository: jaymaeuro/ibm_data_science (github.com)

Thank you!