IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rajarshi Dutta
01.11.2023

Outline

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Conclusion
  - Findings & Implications
- Appendix

Executive Summary

- **Summary of methodologies**
  - Data Collection through API and web scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Data visualization using Matplotlib and interactive maps using Folium
  - Interactive dashboard using Plotly
  - Machine learning prediction using classification
- **Summary of all results**
  - The resulting models all produced similar results, with an accuracy rate of ~83.33% when tested with a test data set. The models tended to over predict successful landings. Training the model with more data could lead to improved accuracy.

Introduction

- **Project background and context**

Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- **Problems you want to find answers**
  - To be able to predict the likelihood of a rocket successfully landing.
  - Which factors influence if a rocket will successfully land?
  - Optimal conditions to achieve a successful landing.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

SpaceX REST API

Data was collected from the SpaceX REST API: api.spacexdata.com/v4/ v4/. More information can be found @ https://docs.spacexdata.com/ com/. The data included information about the rocket used, payload, landing outcome, and other launch & landing specifications in the form of a .JSON file. Data is normalized into a flat .csv file.

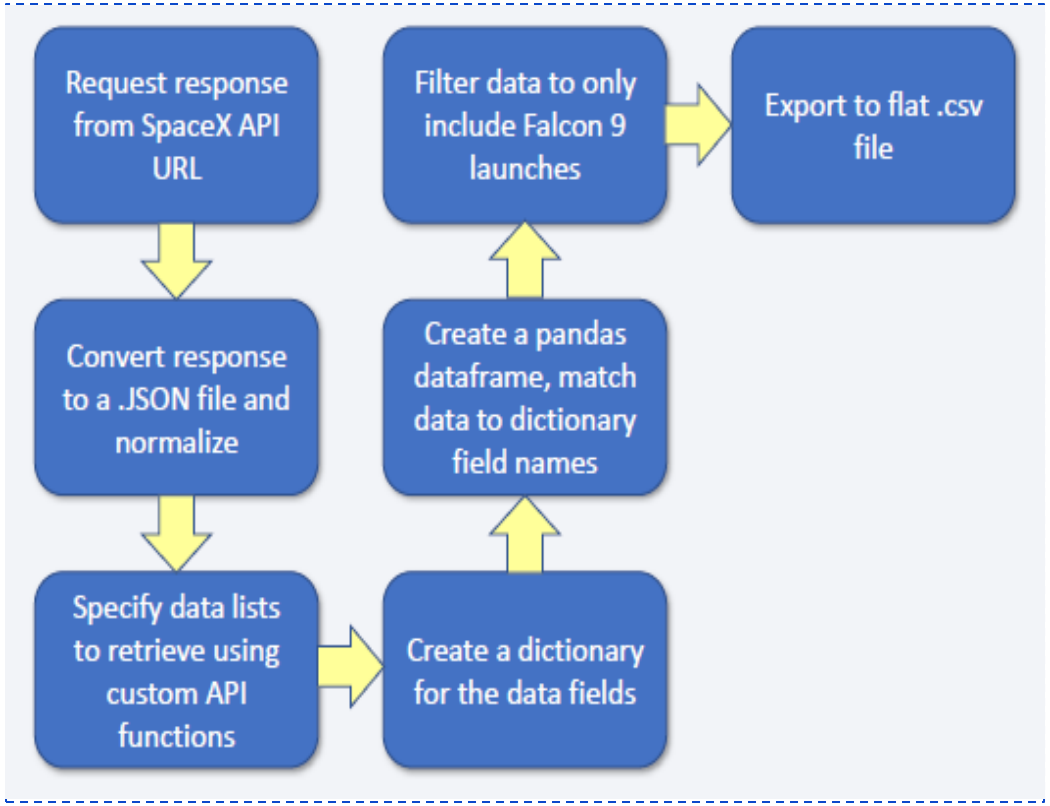
Web Scraping

Data was also collected by web scraping a table from the SpaceX Wikipedia page using the BeautifulSoup python package. Information can be appended to our dataset by using the rocket/flight id as a key. Data is normalized into a flat .csv file.

Data Collection – SpaceX API

Used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.

Github link -
https://github.com/Rajarshi-ctrl/IBM-Data-Science-Capstone-Project/blob/master/jupyter-labs-spacex-data-collection-api.ipynb

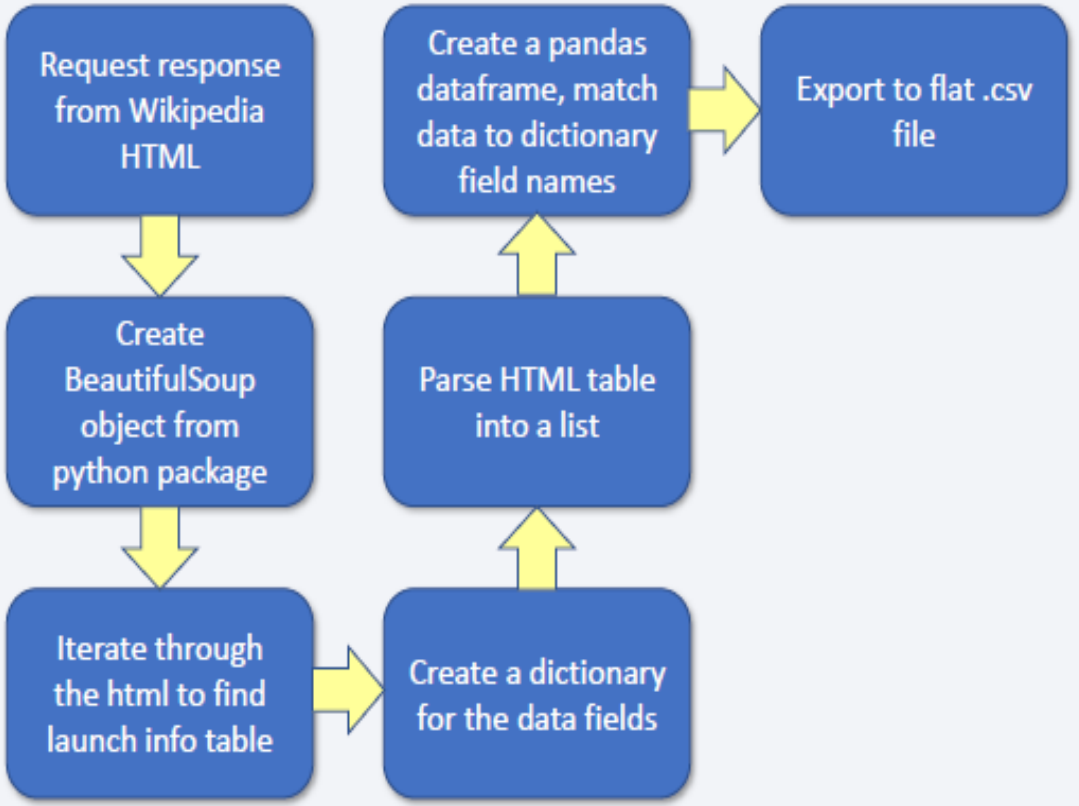


Data Collection - Scraping

- Applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup and parsed the table and converted it into a pandas DataFrame.

- Github link -

  https://github.com/Rajarshi-ctrl/IBM-Data-Science-Capstone-Project/blob/master/jupyter-labs-webscraping.ipynb



Data Wrangling

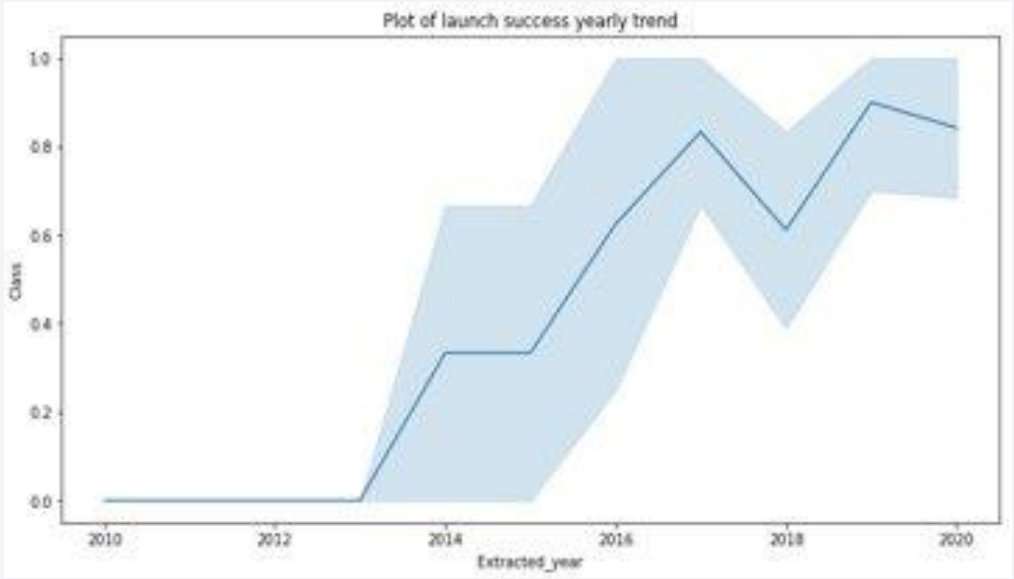
- The outcome field details two components: ‘mission outcome’ and ‘landing location’. We want to create a training label ‘Class’ to indicate successful landing = 1; unsuccessful landing = 0

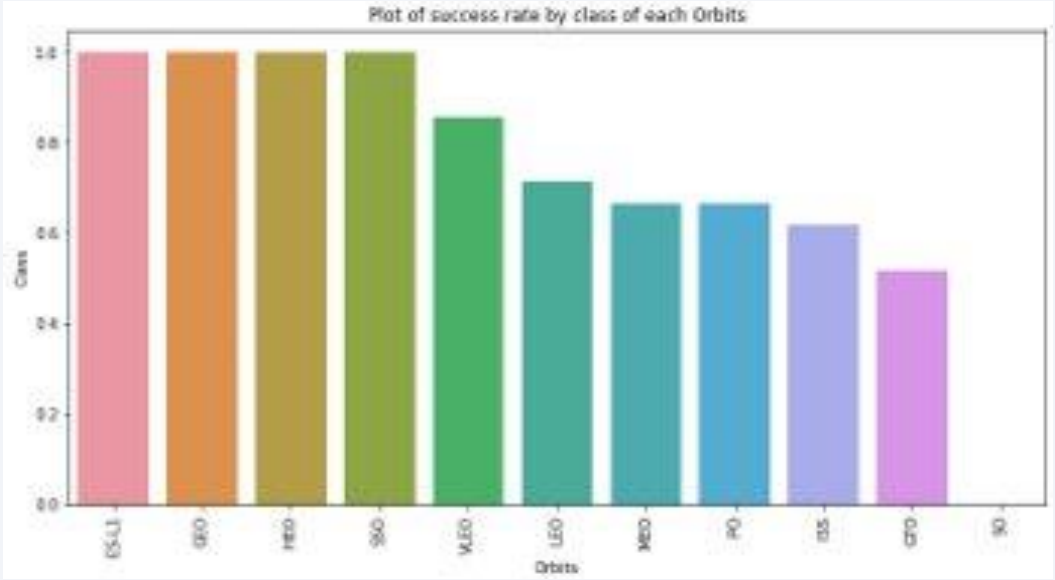
- **Value mapping:**
  - Outcomes ‘True ASDS’, ‘True RTLS’, & ‘True Ocean’ set Class to --> 1
  - Outcomes ‘None None’, ‘False ASDS’, ‘None ASDS’, ‘False Ocean’, ‘False RTLS’ set Class to --> 0

- GitHub URL - https://github.com/Rajarshi-ctrl/IBM-Data-Science-Capstone-Project/blob/master/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend. Github url - https://github.com/Rajarshi-ctrl/IBM-Data-Science-Capstone-Project/blob/master/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb





EDA with SQL

- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string 'CAA'
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the date where the successful landing outcome in drone ship was achieved
- Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster_versions which have carried the maximum payload mass
- Listing the records which will display the month names, successful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2015
- Ranking the count of successful

Github url - *https://github.com/Rajarshi-ctrl/IBM-Data-Science-Capstone-Project/blob/master/jupyter-labs-eda-sql-coursera_sqllite.ipynb*

Build an Interactive Map with Folium

- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distances between a launch site to its proximities. We answered some question for instance:
    - Are launch sites near railways, highways and coastlines.
    - Do launch sites keep certain distance away from cities.

- **Github Url -** *https://github.com/Rajarshi-ctrl/IBM-Data-Science-Capstone Project/blob/master/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb*

Build a Dashboard with Plotly Dash

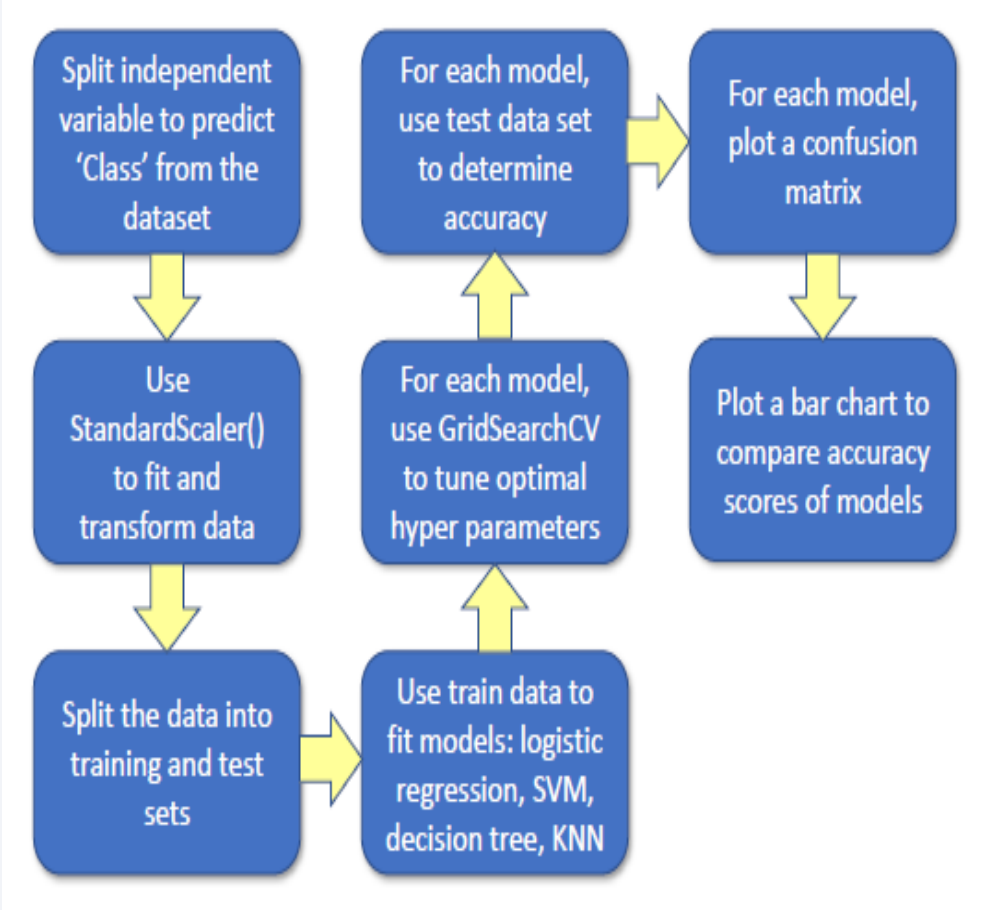
- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

- Github Url - *https://github.com/Rajarshi-ctrl/IBM-Data-Science-Capstone-Project/blob/master/spacex_dash_app.py*

Predictive Analysis (Classification)

Loaded the data using numpy and pandas, transformed the data, split our data into training and testing. Then built different machine learning models and tune different hyperparameters using GridSearchCV and used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning. As found the best performing classification model.

- Github Url - *https://github.com/Rajarshi-ctrl/IBM-Data-Science-Capstone-Project/blob/master/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb*



Results

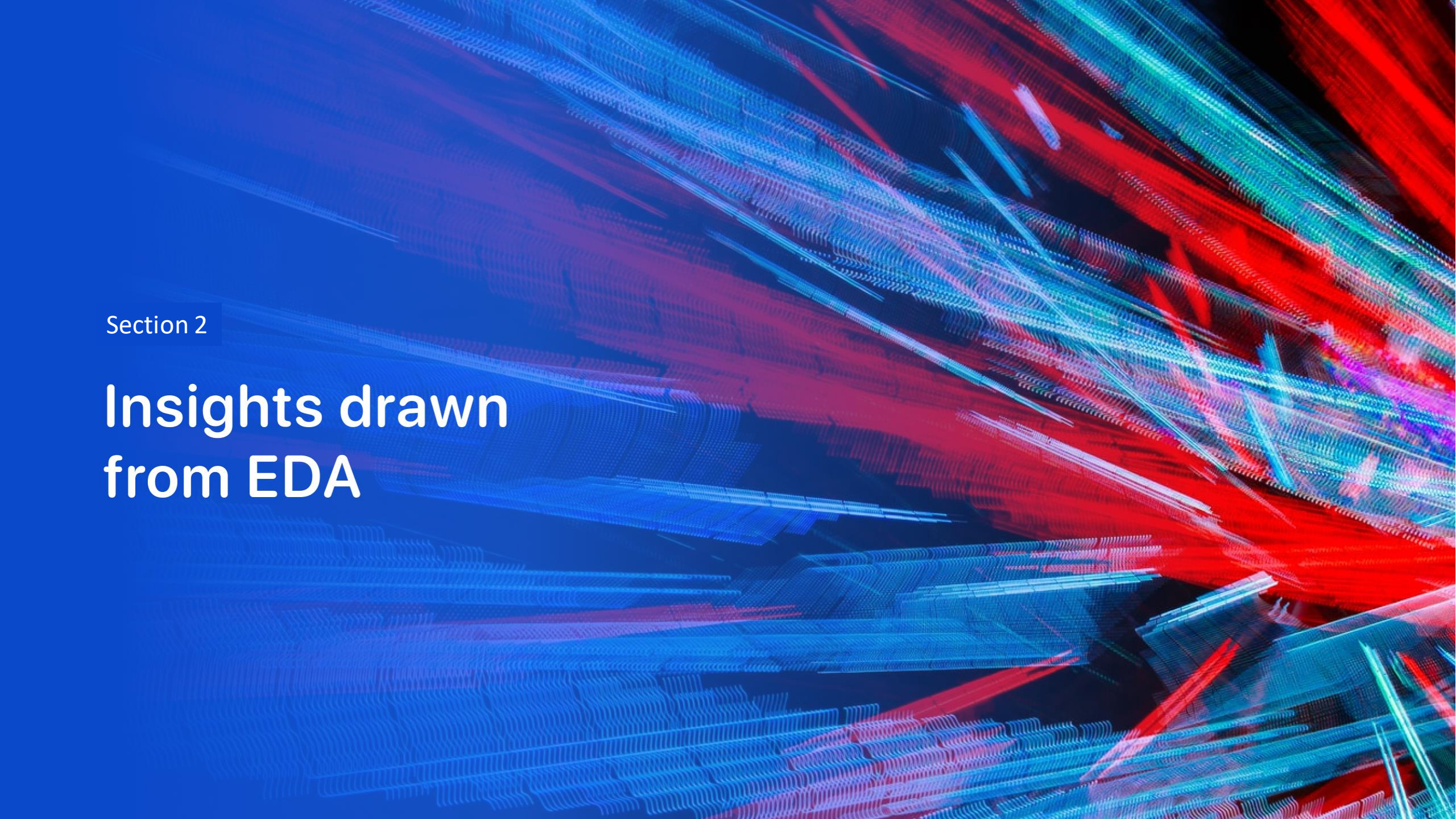
Exploratory data analysis results

Interactive analytics demo in screenshots

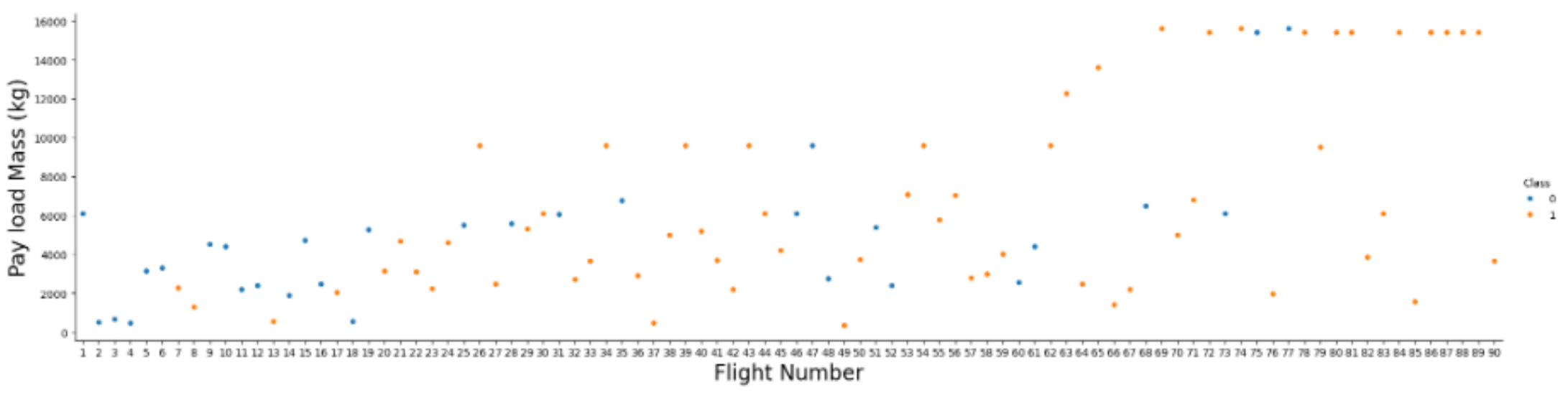
Predictive analysis results

Section 2

Insights drawn from EDA

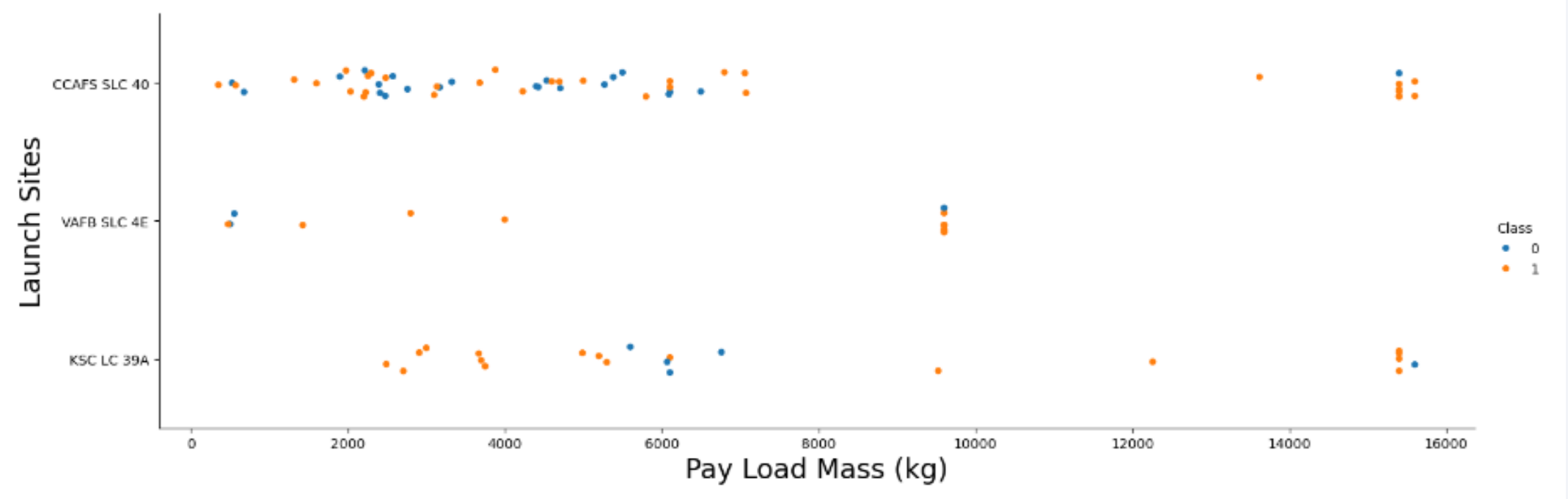


Flight Number vs. Launch Site



- Green indicates a successful launch. Purple indicates an unsuccessful launch.
- Unsuccessful launches were more frequent in the early flight numbers, success rate has improved for more recent flights.

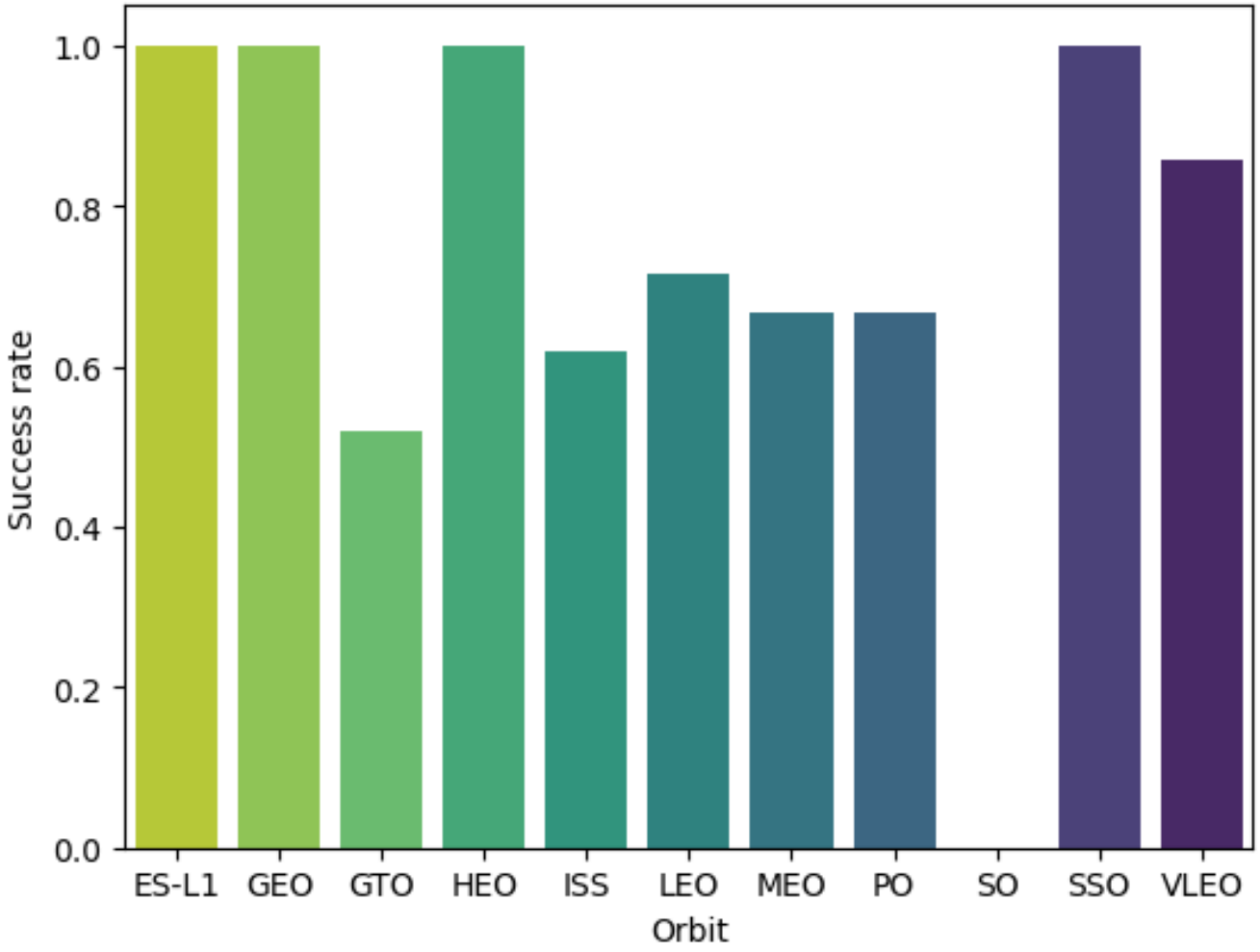
Payload vs. Launch Site



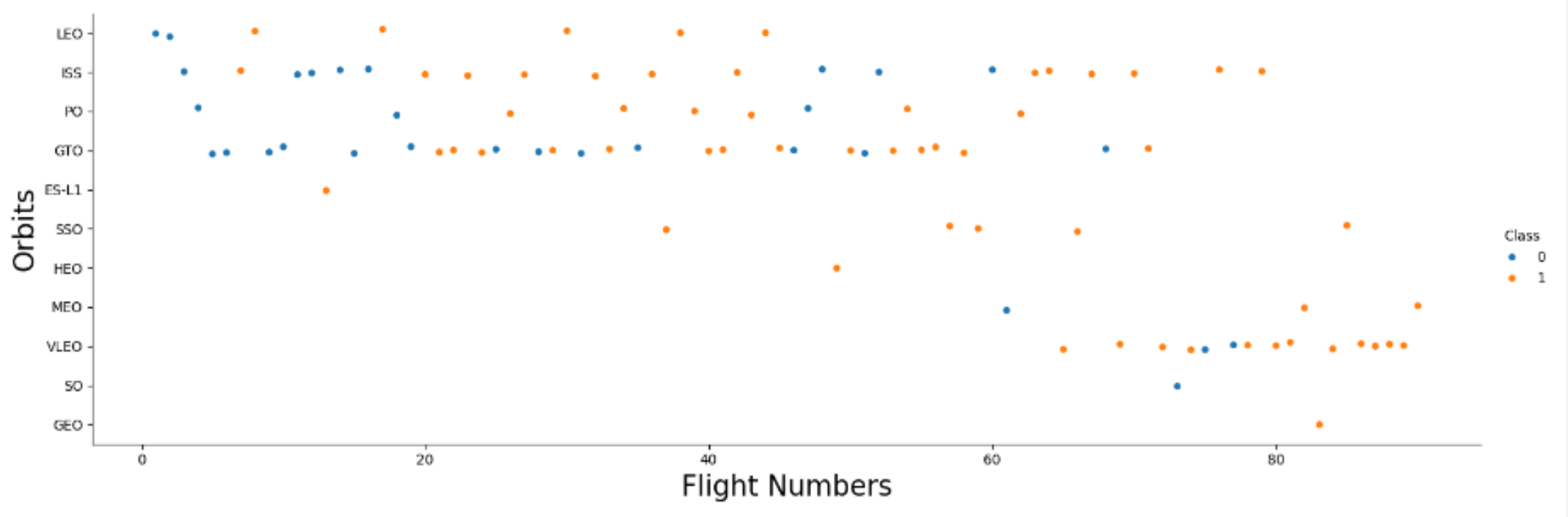
- Green indicates a successful launch. Purple indicates an unsuccessful launch.
- Unsuccessful launches are more frequent in flights with mid lower pay load mass.

Success Rate vs. Orbit Type

- ES L1, GEO, HEO, SSO orbits have 100% successful launch rate.
- SO orbits have 0% successful launch rate.

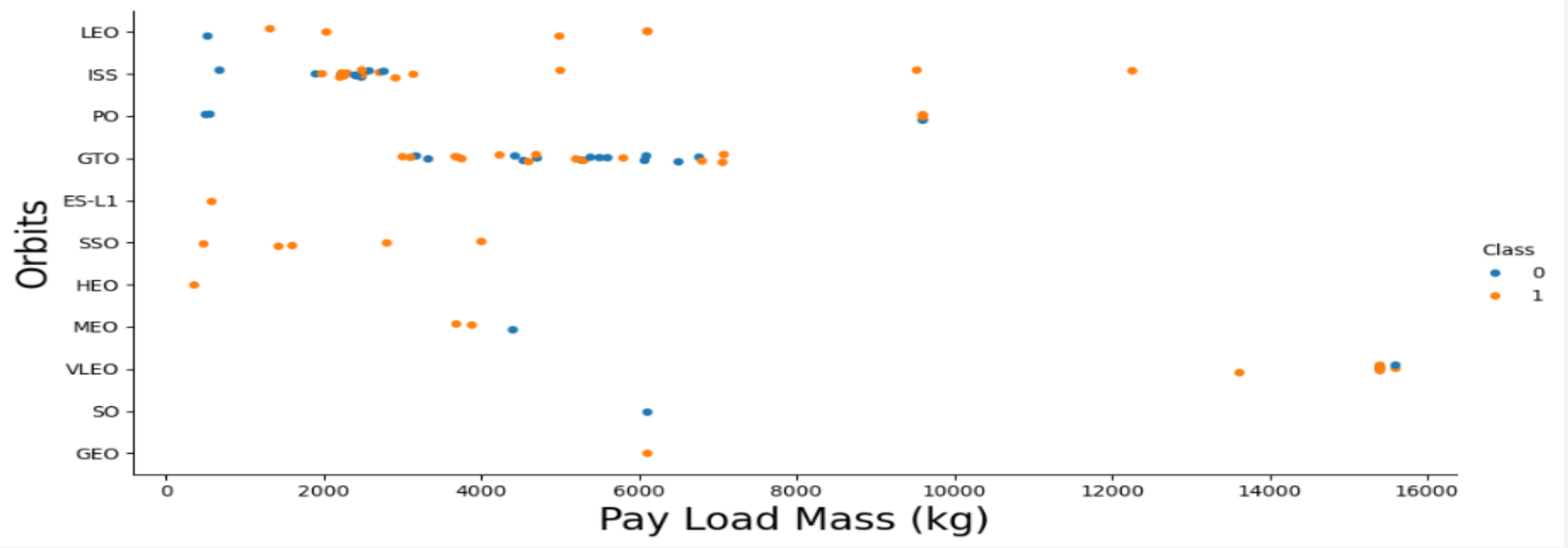


Flight Number vs. Orbit Type

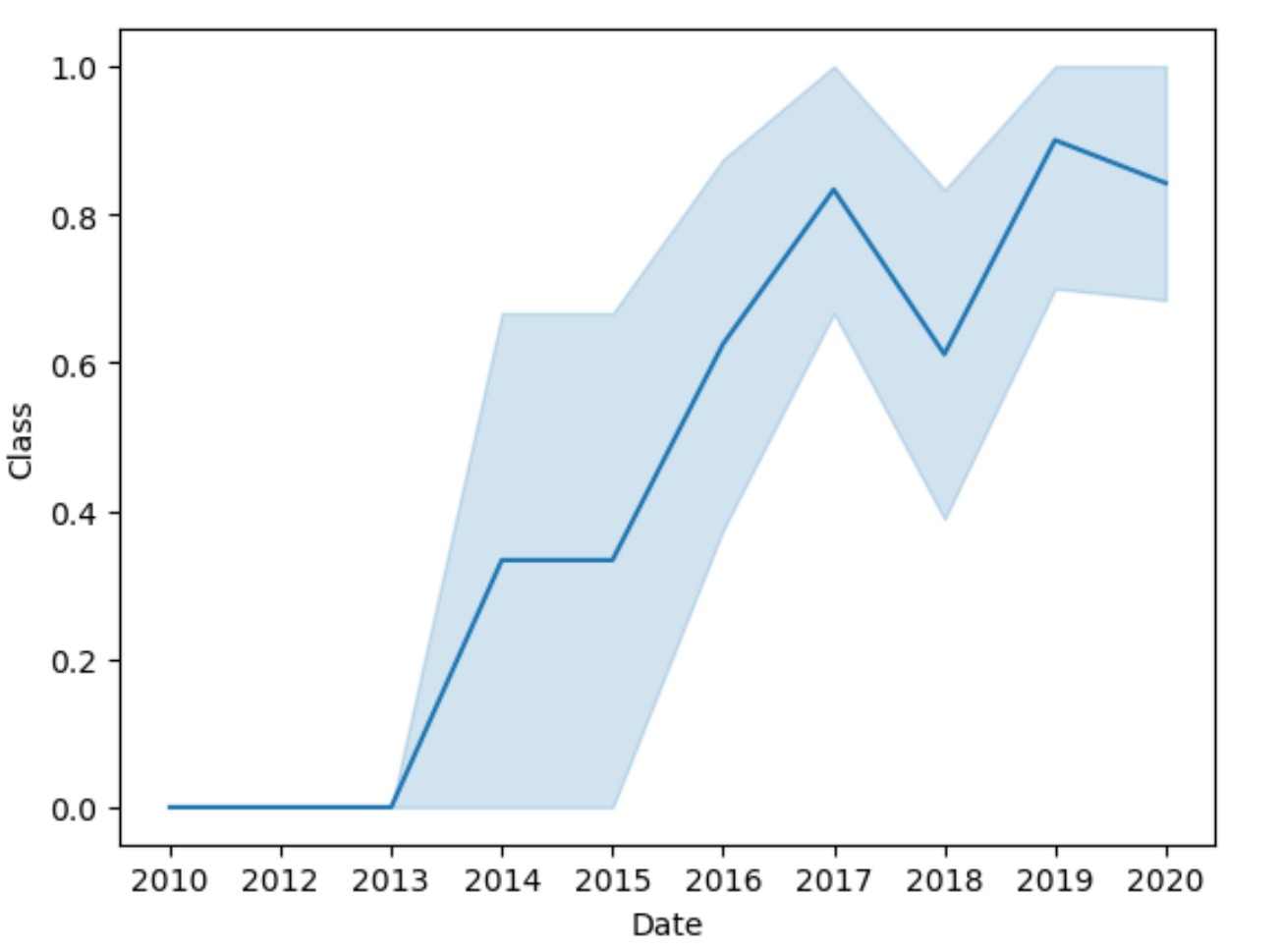


- Green indicates a successful launch. Purple indicates an unsuccessful launch.
- Orbit preference appears to have changed over time. We see some correlation of a higher success rate with the more recent orbit preferences.

Payload vs. Orbit Type



- Green indicates a successful launch. Purple indicates an unsuccessful launch.
- There doesn't seem to be much correlation with pay load mass for GTO orbits. Some other orbits were more successful with heavier payloads.



Launch Success Yearly Trend

- We can see that success rate has generally increased from 2013 2020, with a slight decrease in 2018, and the highest success rate so far being observed in 2019.

All Launch Site Names

- We used the key word **"DISTINCT"** to show only unique launch sites from the SpaceX data.

```
%sql select DISTINCT Launch_Site from SPACEXTABLE
```

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

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

Launch Site Names Begin with 'CCA'

Used the mentioned query to display 5 records where launch sites begin with **"CCA"**

```
%sql select * from SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

Calculated the total payload carried by boosters from NASA as 45596 using the query below

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer='NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

Calculated the average payload mass carried by booster version F9 v1.1 as 2928.4 using below query

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

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

avg(PAYLOAD_MASS__KG_)
2928.4

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

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

min(Date)
2015-12-22

First Successful Ground Landing Date

We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015

Successful Drone Ship Landing with Payload between 4000 and 6000

Used the **WHERE** clause to filter for boosters which have successfully landed on drone ship and applied the **AND** condition to determine successful landing with payload mass greater than 4000 but less than 6000

```
%sql select Booster_Version from SPACEXTABLE where Landing_Outcome='Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 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

Used wildcard like '%' to filter for **WHERE** MissionOutcome was a success or a failure.

```
%sql SELECT mission_outcome, count(*) as Count FROM SPACEXTABLE 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

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

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

Determined the booster that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX()** function.

2015 Launch Records

Used a combinations of the WHERE clause, LIKE, AND, and BETWEEN conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

```
%sql select substr(Date,0,5) as Year, substr(Date, 6,2) as Month, Booster_Version, Landing_Outcome
    from SPACEXTABLE where Landing_Outcome='Failure (drone ship)' and Date like '2015%'
```

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

Year	Month	Booster_Version	Landing_Outcome
2015	10	F9 v1.1 B1012	Failure (drone ship)
2015	04	F9 v1.1 B1015	Failure (drone ship)

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

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

- Selected Landing outcomes and **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for LandingOutcomes **BETWEEN** 2 010-06-04 to 2010-03-20.
- Applied **GROUP BY** clause to group landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.

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

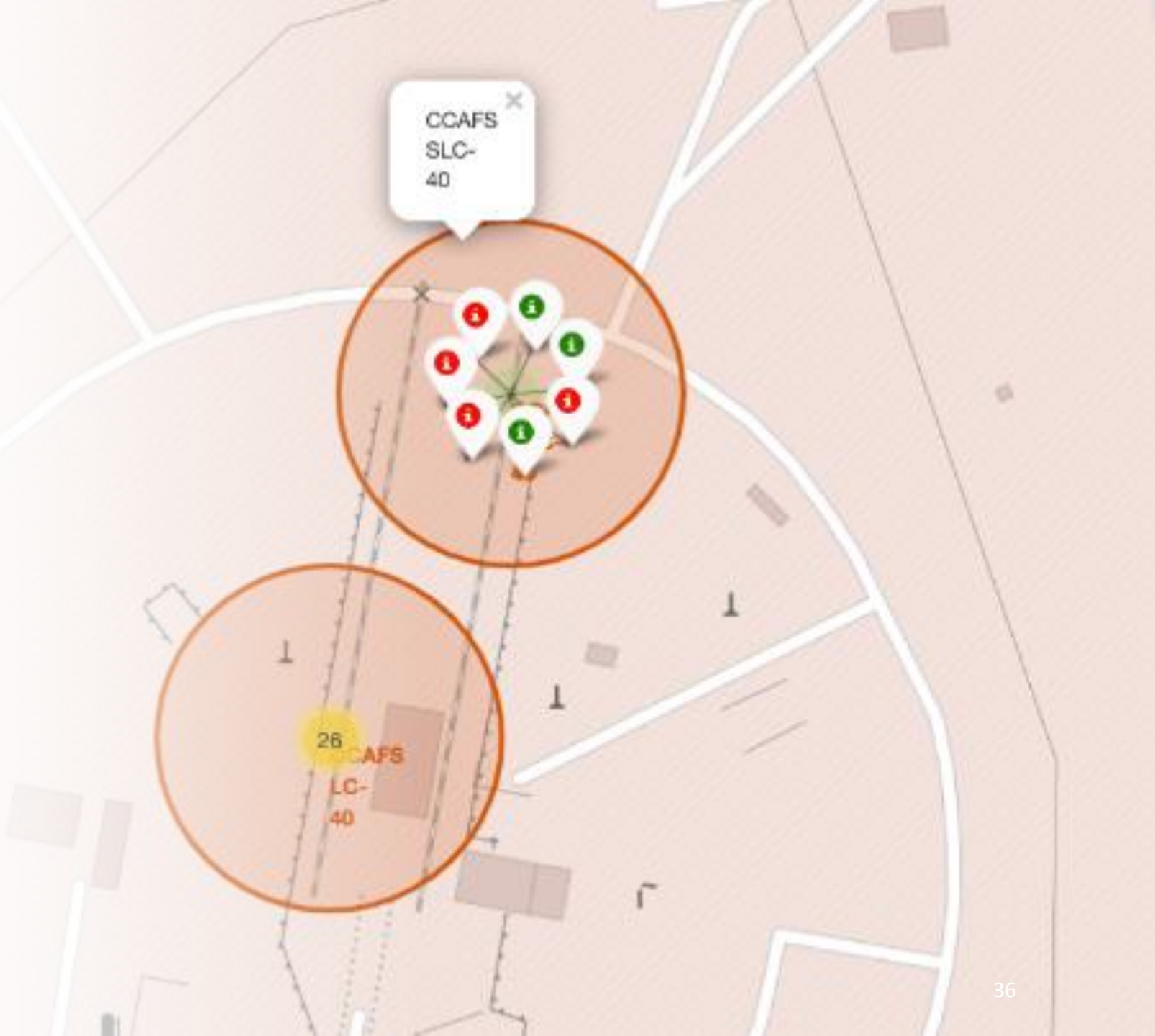

Section 3
Launch Sites
Proximities Analysis

Folium Map: Launch site locations

- We can see that all launch sites are located in North America and that all launch sites are located near to coastlines, specifically the coasts of Florida and California.

Launch sites with color labels

- 22 launch records are clustered at this launch location (VAFB SLC 4E).
- This tells us that there were 4 successful landings (green) and 6 unsuccessful landings (red).

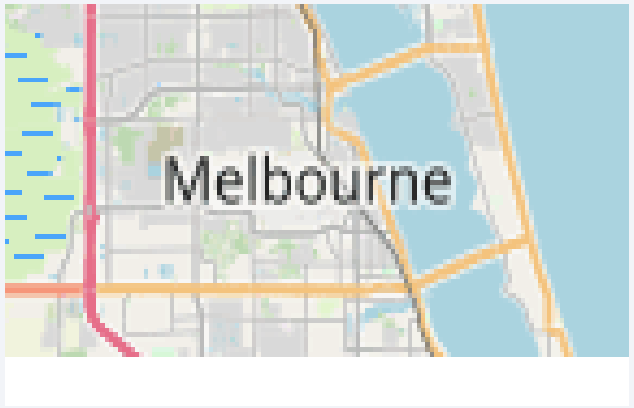


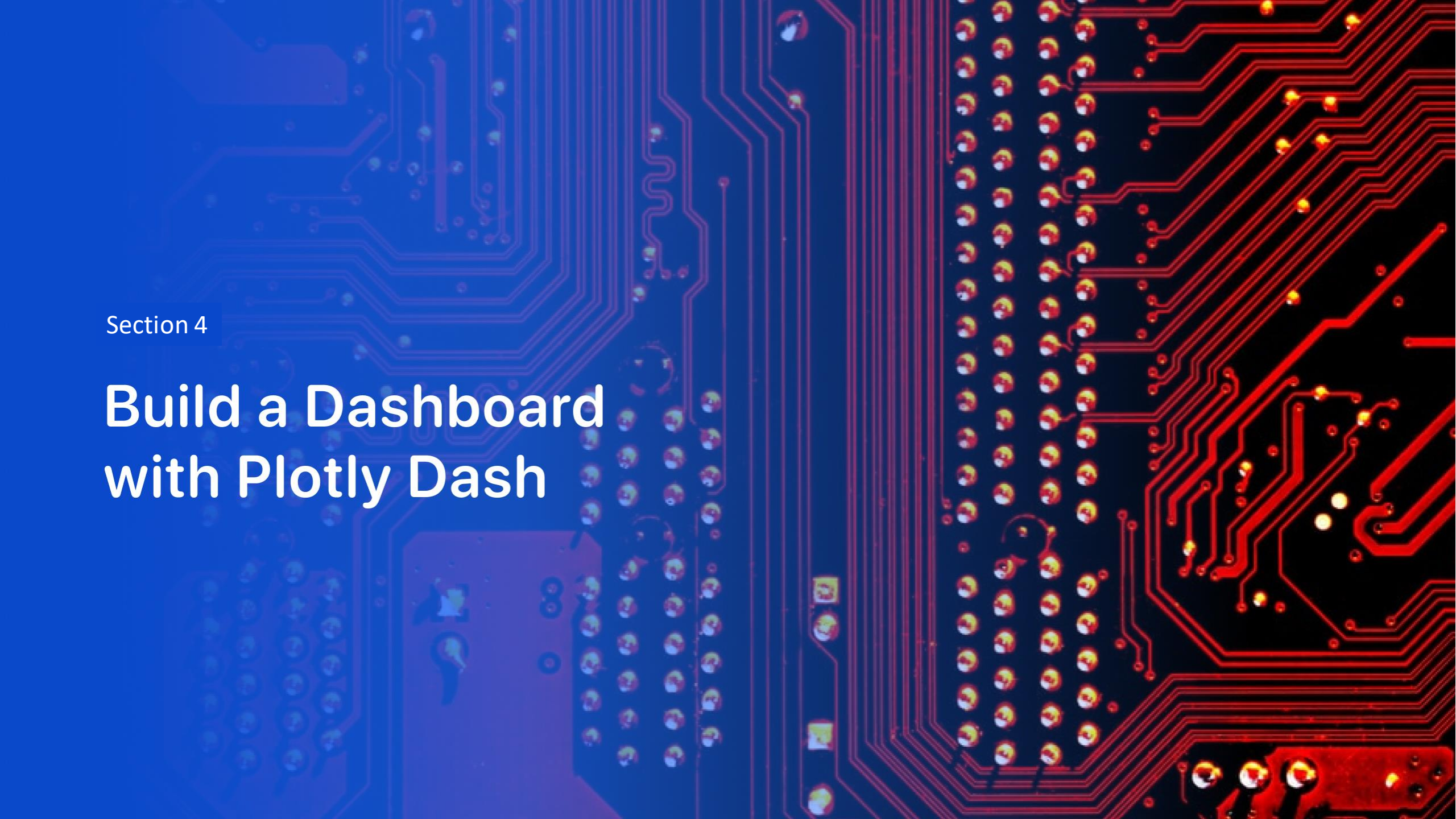
Maps in details

A Railway Maps Looks like :

A Highway Maps Looks like :

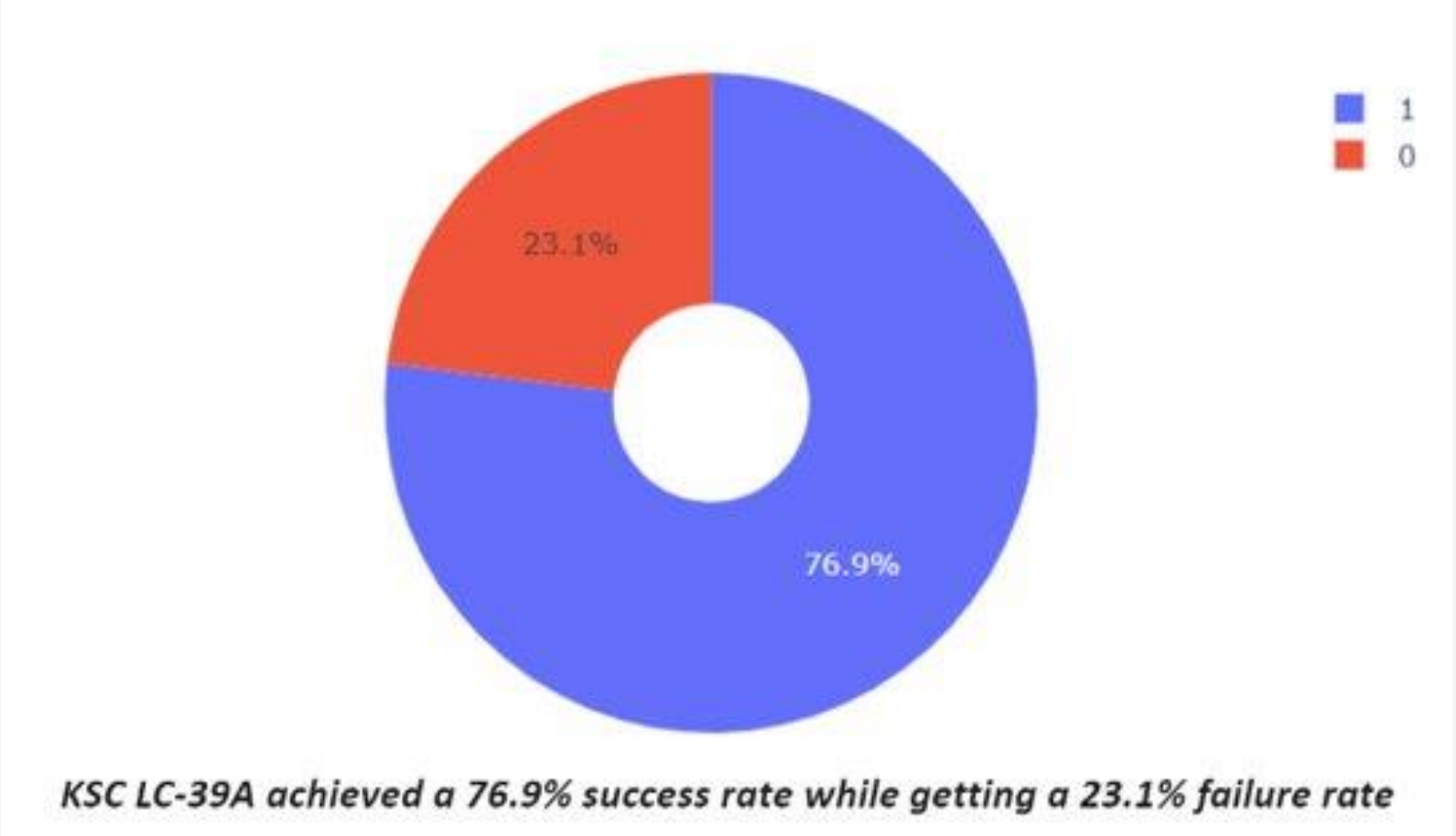
A City Maps symbol Looks like :




Section 4
Build a Dashboard with Plotly Dash

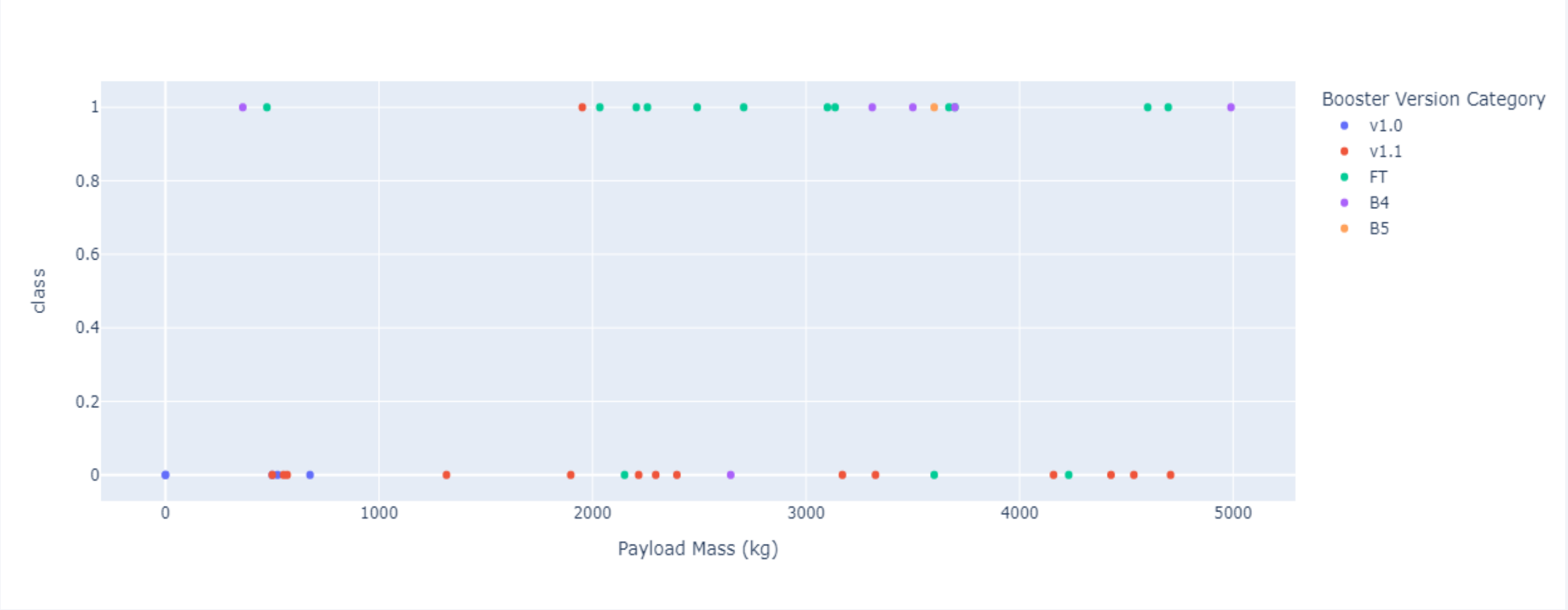
Success Percentage using Pie-chart

Launch Site with Highest Success Ratio



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Payload Mass vs. Success vs. Booster Version Category



Section 5

Predictive Analysis (Classification)

Classification Accuracy

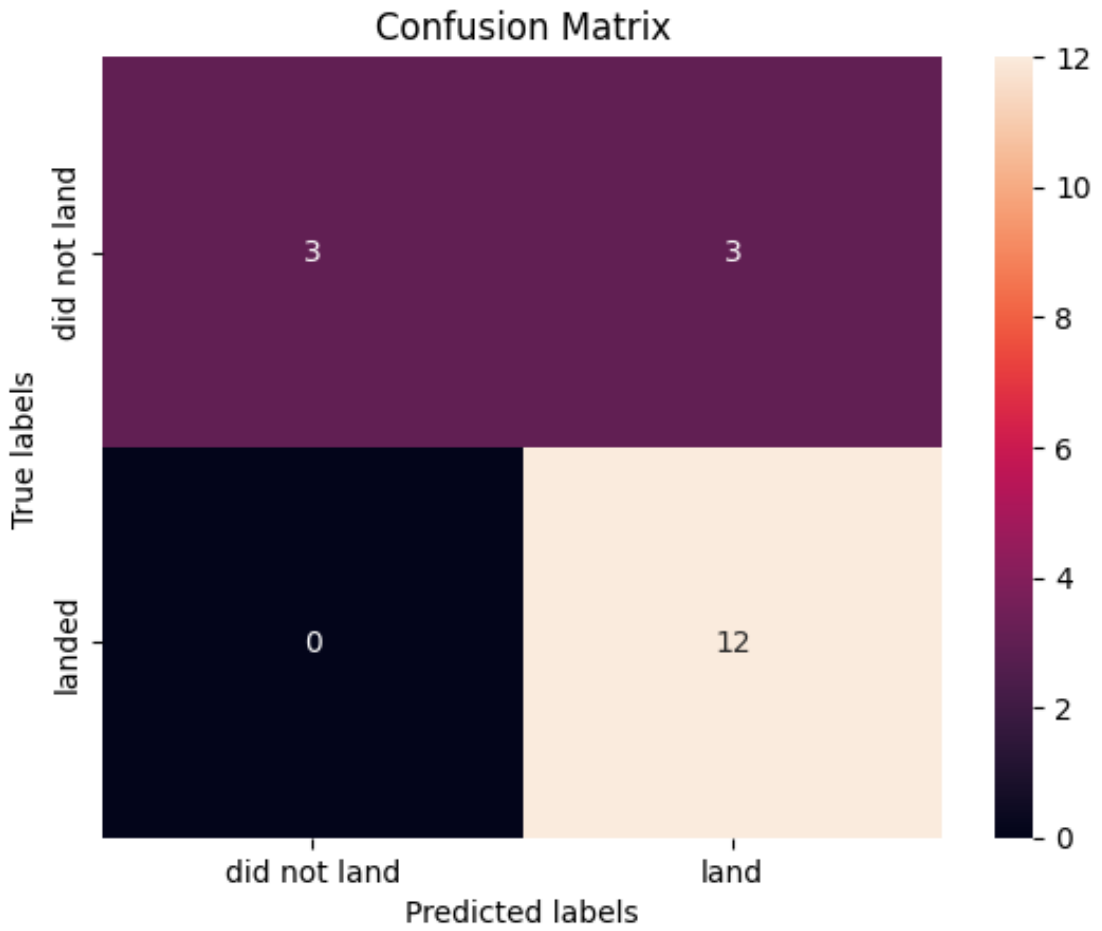
- All models produced the same accuracy against the test data set (~83.33%).
- This is likely due to the limited data used, training and testing the models on larger data sets may produce more varied results.

```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.



Conclusions

- Our goal was to develop a machine learning model to predict if stage 1 will successfully land for a given launch.

- We developed four machine learning models, which all predicted successful landings with ~83.33% accuracy for some test data. The models tend to over predict successful landings, the models could be improved by using more data.

- In addition, we found that:
  - Success rate of stage 1 landings has improved over time.
  - ES L1, GEO, HEO, SSO orbits have the best success rate.
  - Launch sites are typically located close to coastlines.

Appendix

Please find all the Project Notebooks through the GitHub link - https://github.com/Rajarshi-ctrl/IBM-Data-Science-Capstone-Project

Special acknowledgement to the IBM Data Science Professional Certificate Course (https://www.coursera.org/professional-certificates/ibm-data-science?)

Thank you!