Winning Space Race with Data Science

Denis Levert
October 11, 2021

Outline

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

Executive Summary

- The methodology used in this report follows the CRISP-DM methodology from John Rollins. In summary, it consists of the following main phases:
    1. From problem to approach.
    2. Working with data.
    3. Deriving the answer.

- Summary of all results

    1. Launch outcomes are becoming more successful and payloads can be matched for even higher success rates.
    2. The Orbits can affect the landing predictability and several have 100% success rate.
    3. The launch site can be selected for the highest success rate.
    4. The booster can be selected along with the payload range for the best chance of a successful landing. The most successful booster can be chosen based on payload required to be launched.
    5. Prediction can be made with an 89% accuracy.

Introduction

- Project background and context
  - SpaceY funded by Allon Mask is looking to determine the price of each launch for their new space program.
  - SpaceX has the best data and success rate. Their data is available to analyze.

- Problems you want to find answers
  - What does SpaceX data tell us about the landing predictability of the Falcon 9 booster?
  - Can the probability of a first stage landing be predicted?
  - Can this data be used to predict the cost of a launch for SpaceY?

Section 1

Methodology

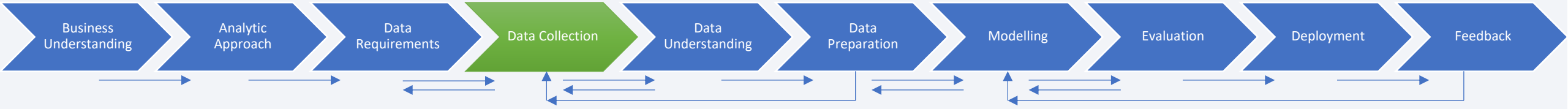
Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using API calls to https://api.spacexdata.com/v4. Then the data was filtered for Falcon 9.
- Perform data wrangling
  - Payload nulls were replaced with the median. Data was explored to understand it, and then a column created to Classify the landings as a success or failure.
- 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

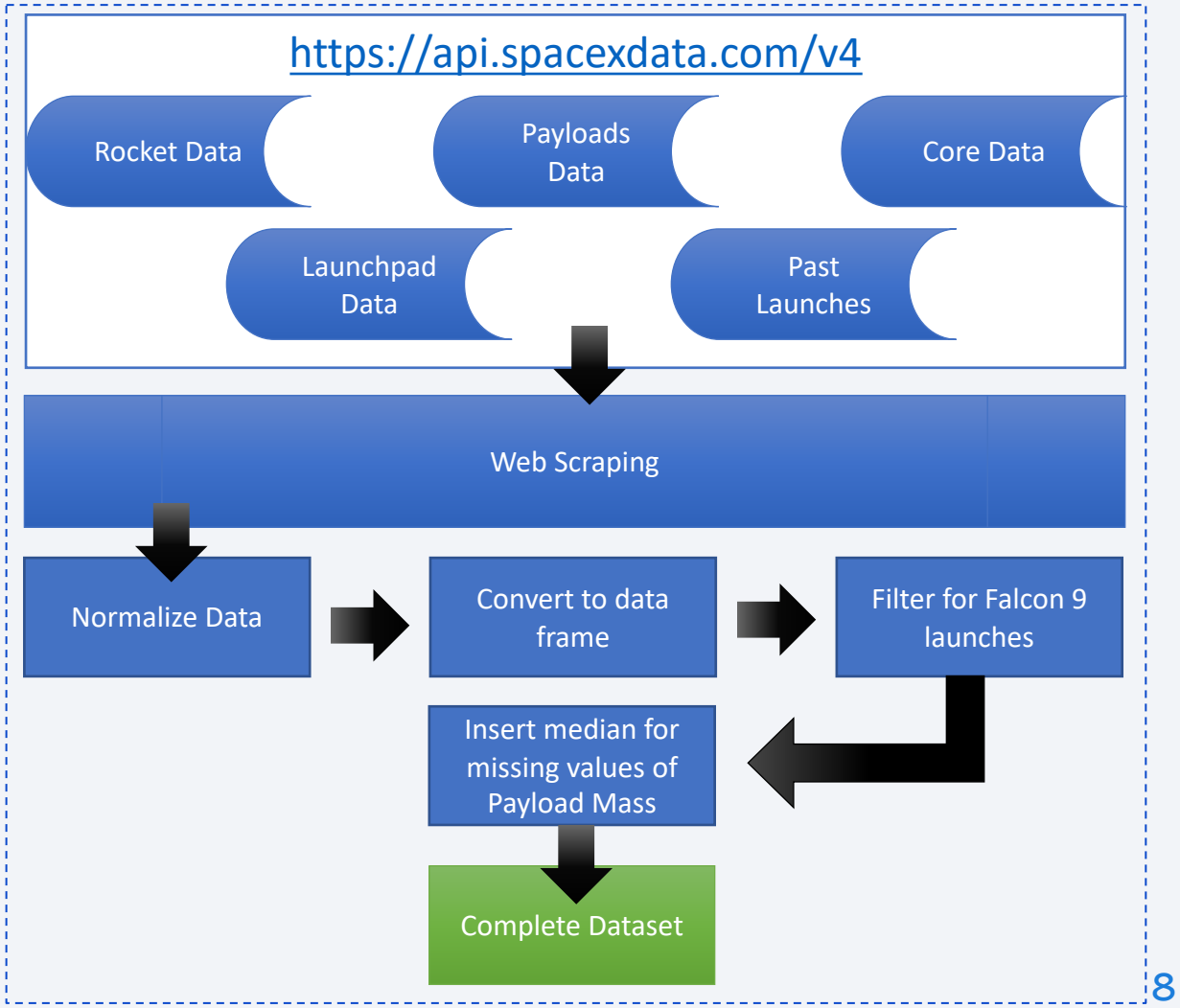
Data Collection

- The datasets were collected using REST API's from the Space X site https://api.spacexdata.com/v4.
- The intent is to collect enough data to determine if the first phase be reused to determine the cost of a launch and if a company can competitively bid against Space X.
- Space X data will be collected, cleaned, analyzed, and then explored further.



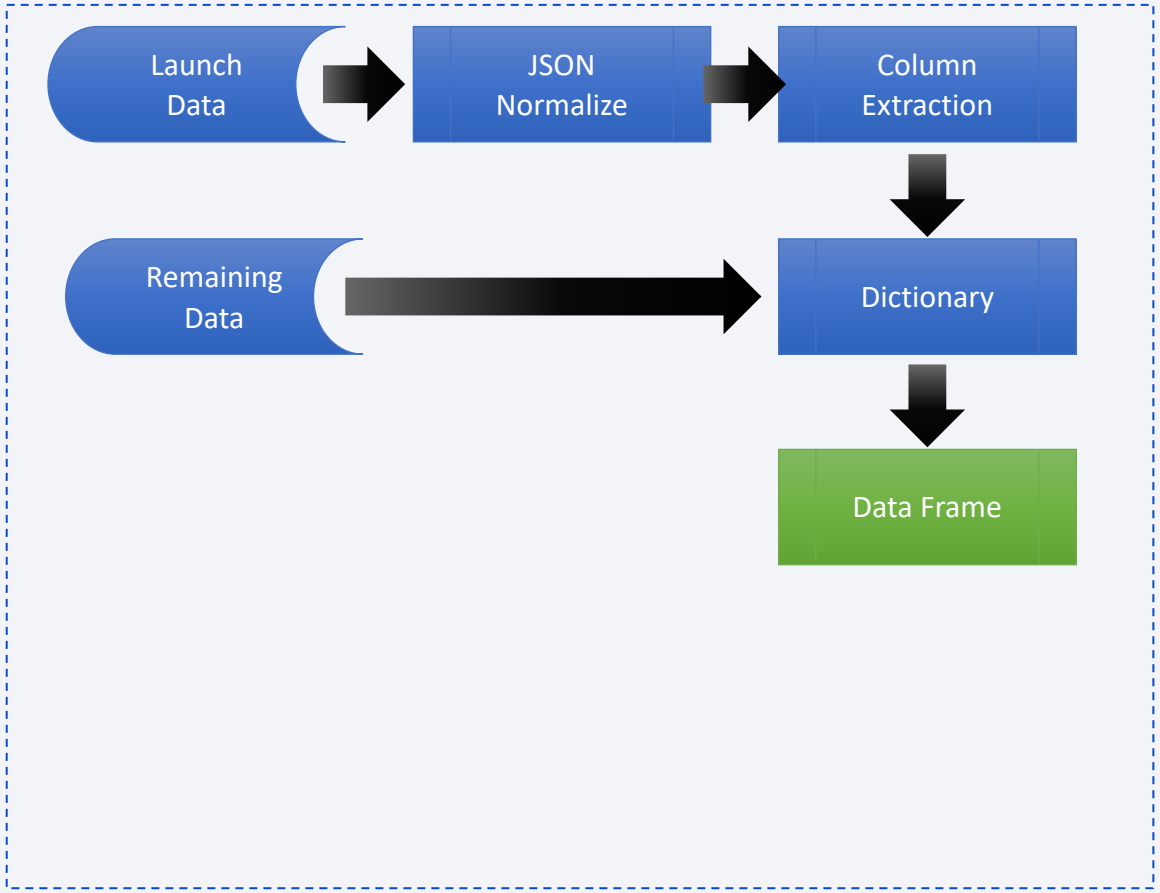
Data Collection – SpaceX API

- The data was collected from https://api.spacexdata.com/v4. 5 separate REST API's were called to collect the Rocket, Launchpad, Payloads, Cores, and Launches data.
- The data was then parsed, stored in lists, converted to dictionaries, using the ID columns converted to a data frame.
- The dataset was filtered to Falcon 9 launches only.
- The Jupyter notebook can be found at: https://github.com/dlevert/DSProject/blob/1058d4efddc46242e8fa28e001fc2cc99c776617/Lab%201%20Collecting%20the%20data.ipynb



Data Collection - Scraping

- The launch API returned a lot of data. The data was parsed into a data frame (data) using a JSON normalize function, then relevant columns extracted.
- From here the relevant columns from the rest of the data was extracted into a dictionary.
- The dictionary was converted to a data frame
- The Jupyter notebook can be found at: https://github.com/dlevert/DSProject/blob/1058d4efddc46242e8fa28e001fc2cc99c776617/Lab%201%20Collecting%20the%20data.ipynb



Data Wrangling

- The data was checked for null values. The columns data types were checked. Launches evaluated by site. Orbits evaluated by type and then checked against a positive outcome. A landing outcome was determined and loaded into the 'Class' column. An overall success rate of 66.7% was observed in the data.
- The Jupyter notebook can be found here: https://github.com/dlevert/DSProject/blob/1058d4efddc46242e8fa28e001fc2cc99c776617/Lab%202%20Data%20wrangling.ipynb

EDA with Data Visualization

- Several charts were explored to look for relationship's in the success of a first stage landing.
  - Payload mass vs. flight number shows that generally mass went up over time as well as success rate.
  - Launch site vs. flight number indicates that there are more successful landings with more flights. Landing site 4E has a high success rate.
  - Payload vs. launch site indicates heavier loads have a high success rate.
  - Success rate by orbit type indicates ES-L1, GEO, HEO, and SSO have the highest success rate, while SO has the worst success rate.
  - Orbit vs. flight number does not seem to have an impact.
  - Orbit type vs. payload doesn't seem to make a difference except for GTO orbit it has a negative impact.
  - Success rate over year has a positive impact on success rate, indicating an improvement over time.

- The Jupyter notebook can be found here: https://github.com/dlevert/DSProject/blob/1058d4efddc46242e8fa28e001fc2cc99c776617/Lab%204%20Exploring%20and%20Preparing%20Data.ipynb

EDA with SQL

- The following SQL analysis was performed:
  - Summarized the launch sites.
  - Displayed 5 records starting with 'CCA'.
  - Totaled the payload mass launched by NASA (CRS).
  - Averaged the payload mass carried by booster version F9 v1.1.
  - Queried the first successful ground pad landing.
  - Queried the boosters that had successful drone ship landings carrying a payload mass between 4,000 and 6,000 kg.
  - Summarized the total success and failure mission outcomes.
  - Queried the booster versions that carried the maximum payload mass.
  - Queried the booster version and the launch site that failed a drone ship landing in the year 2015.
  - Ranked the landing outcomes in descending order between the dates 2010-06-04 and 2017-03-20.
- The Jupyter notebook can be found here:
  https://github.com/dlevert/DSProject/blob/1058d4efddc46242e8fa28e001fc2cc99c776617/Lab%203%20SQL%20Notebook.ipynb

Build an Interactive Map with Folium

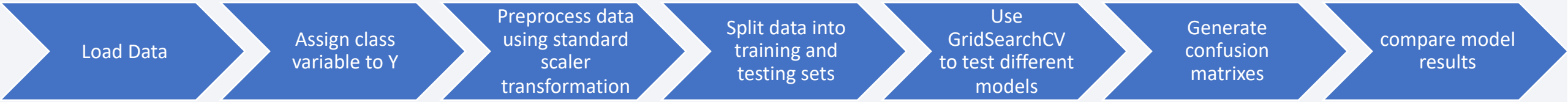
- The Folium map was plotted and the following objects created:
  - Circles: Used to identify geographical locations of sites to clearly show where the launches took place.
  - Markers: used to pinpoint locations.
  - Marker Clusters: used to group the markers and visualize quantity, allowing to zoom into the points to represent quantity and visually display success rates..
  - Text marker: Used to display information (Distance in this example).
  - Lines: Used to reference between two points. The launch site and the nearest points of interest.
- The Jupyter notebook can be found here: https://github.com/dlevert/DSProject/blob/1058d4efddc46242e8fa28e001fc2cc99c776617/Lab%205%20Analysis%20with%20Folium.ipynb
- Afor readability, I have included an NB Viewer link that displays the interactive maps: https://nbviewer.org/github/dlevert/DSProject/blob/1058d4efddc46242e8fa28e001fc2cc99c776617/Lab%205%20Analysis%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- A dashboard is built to allow user interface with the data and allow the ability to drilldown into it. The following graphs and interactions where built:
  - A dropdown was created to select one or all launch sites. The dropdown filters the entire dashboard to focus on location.
  - A pie chart was added; When 'All Sites' is selected it displays the percent of successful outcome by site. When a specific launch site is selected, it displays the ratio between successful and unsuccessful outcome.
  - A scatter chart was added to display payload vs. launch outcome. The color is set to the booster version. This graph filters with the launch site dropdown.
  - A slider was built to range the payload mass in the scatter chart. This allow interactivity and exploring the best combination of payload mass, launch site and booster version.
- The Jupyter notebook can be found here: https://github.com/dlevert/DSProject/blob/1058d4efddc46242e8fa28e001fc2cc99c776617/Lab%206%20Pandas%20Dash.ipynb

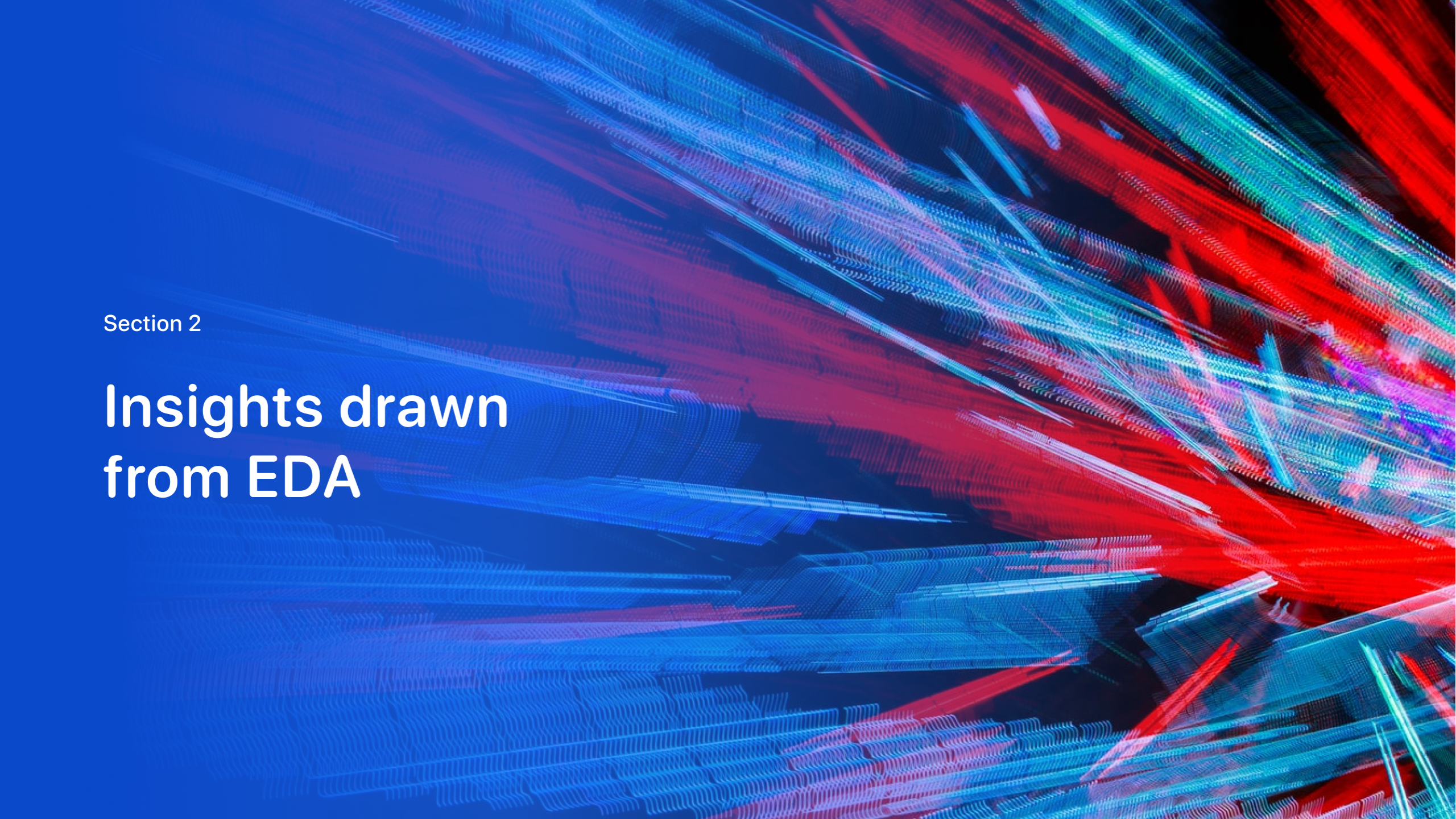
Predictive Analysis (Classification)

- The dataset was loaded then:
  - The Class column was assigned to the 'Y' variable.
  - The remaining variables where preprocessed and fit to a standard scaler transformation, then set to the 'X' variable.
  - The data was split into train(80%) and test(20%) sets.
  - The GridSearchCV method was used to 'fit' and 'score'. The 'best parameters' and 'best score' was evaluated for the following methods:
    - Logistic Regression
    - Support Vector Machine
    - Decision Tree
    - K Nearest Neighbor
  - A confusion matrix was built for each method.
  - I final matrix to display the comparison of the models.
- The Jupyter notebook can be found here: https://github.com/dlevert/DSProject/blob/1058d4efddc46242e8fa28e001fc2cc99c776617/Lab%207%20Landing%20Prediction.ipynb



Results

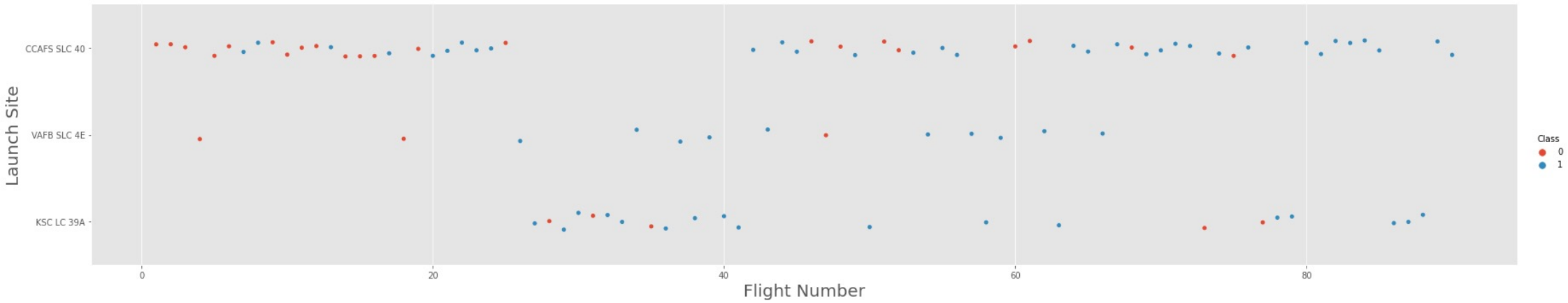
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results



Section 2

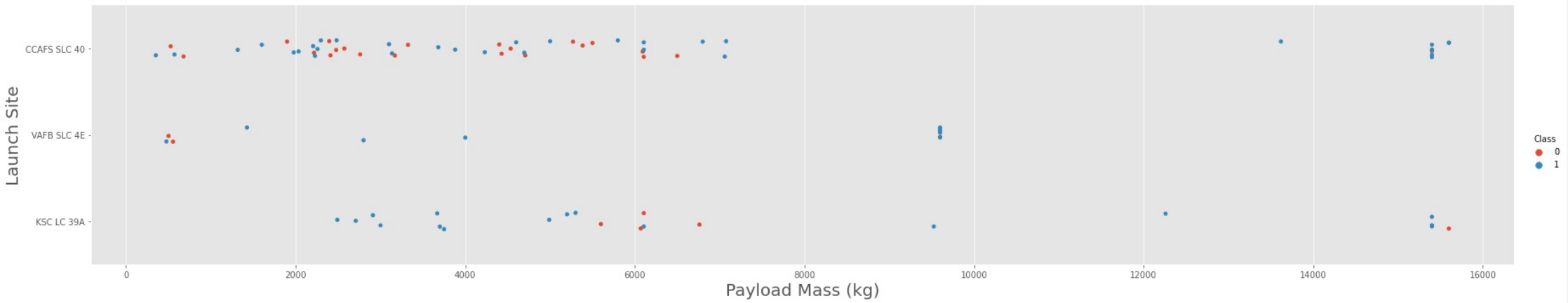
Insights drawn from EDA

Flight Number vs. Launch Site



- This plot shows the flight numbers and which launch site it was from with the color showing the outcome 0 for negative and 1 for successful.
- It does indicate many negative outcomes early and more successful outcomes later.
- CCAFS SLC 40 is the most used launch site, VAFB SLC 4E the least used.

Payload vs. Launch Site

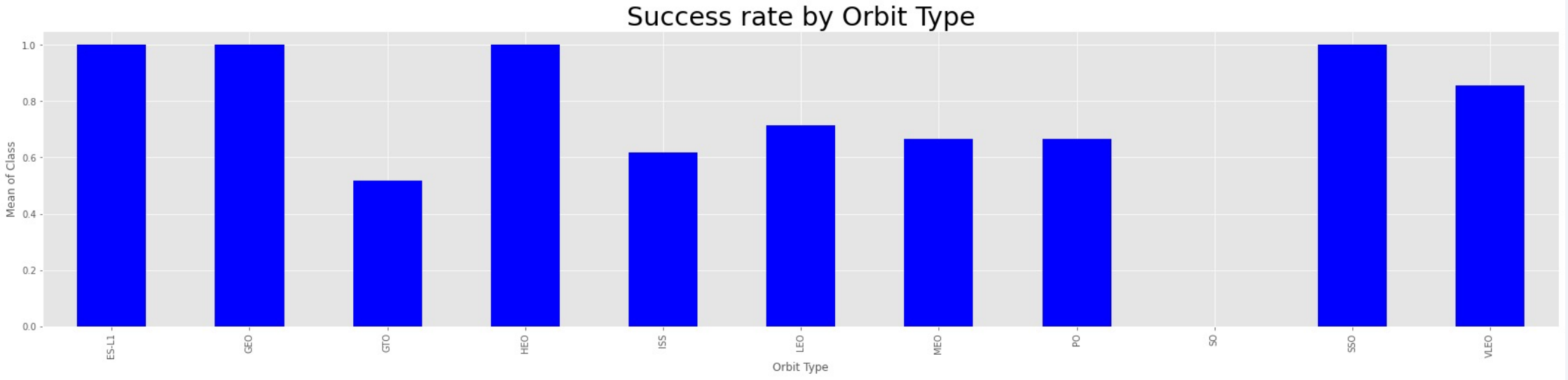


- This plot shows the payload mass and which launch site it was from with the color showing the outcome 0 for negative and 1 for successful.

- The data indicates that as the payload mass increases, so does the probability of a successful outcome.

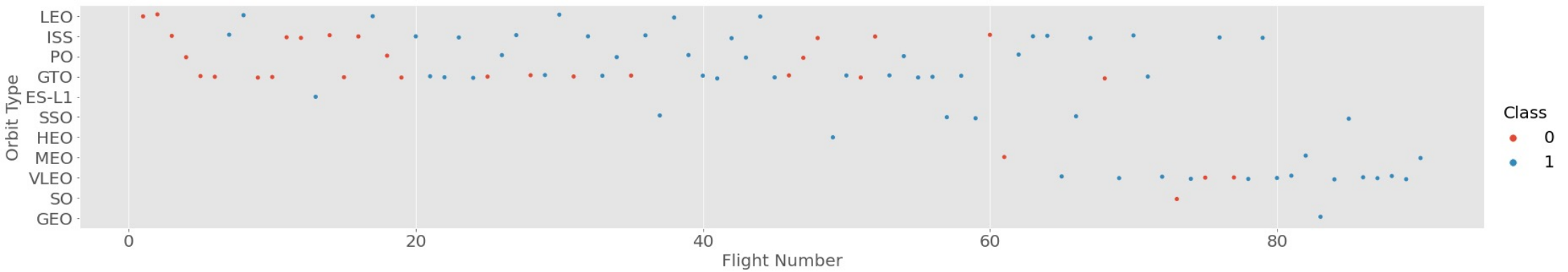
- There are a lot of negative outcomes with lower payload masses.

Success Rate vs. Orbit Type



- This plot shows the orbit type and which the mean of the outcome (Success rate).
- Orbits ES-L1, GEO, HEO, and SSO have a 100% success Rate.
- SO has the lowest success rate.

Flight Number vs. Orbit Type

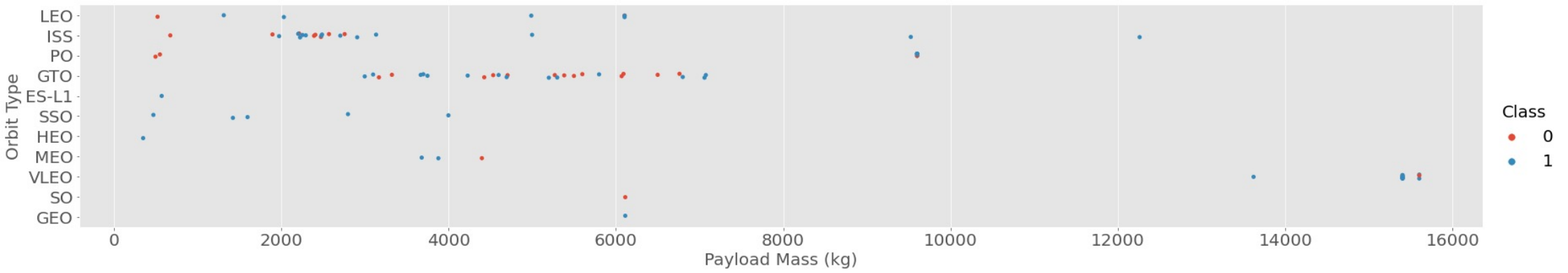


- This plot shows the flight number and which orbit type with the color showing the outcome 0 for negative and 1 for successful.

- The data suggests LEO orbits got better over the number of attempts.

- GTO orbits are spurratic and not getting better over the number of attempts.

Payload vs. Orbit Type

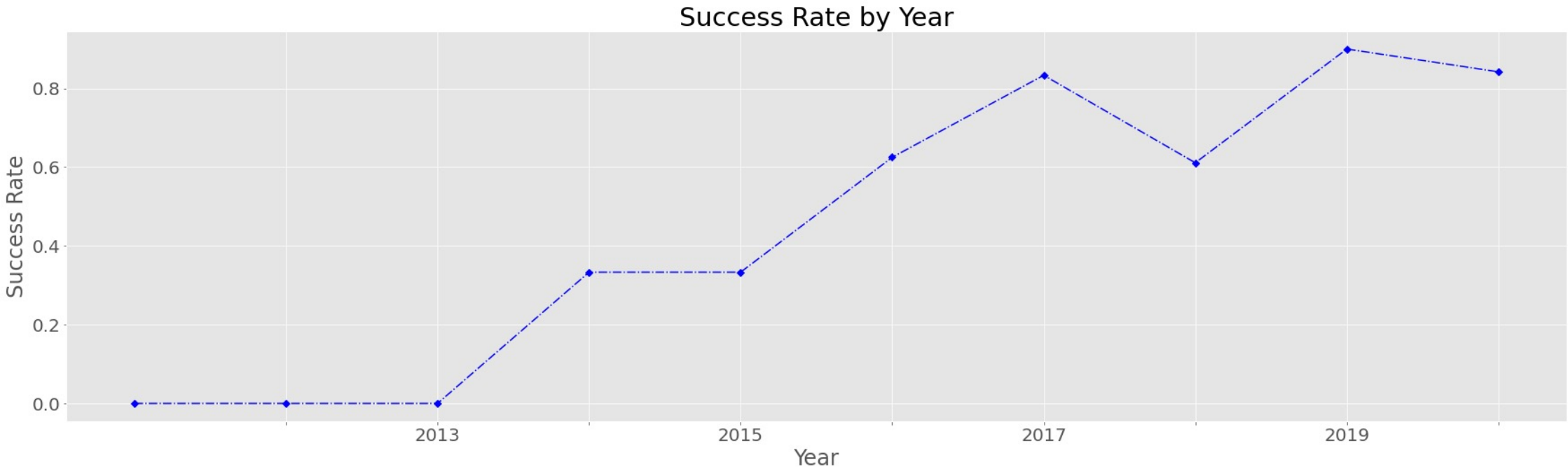


- This plot shows the payload mass and which orbit type with the color showing the outcome 0 for negative and 1 for successful.

- Orbits LEO and ISS seem to have more successful outcomes with heavier payload mass.

- The GTO orbit seems to have more negative outcomes as the payload mass increases.

Launch Success Yearly Trend



- This plot shows the success rate by year.
- The trend shows that the success rate improving over time.
- The only anomaly is 2018 where there was a dip in success rate.

All Launch Site Names

- SQL Statement

```
%sql\
select launch_site\
from SPACEXTBL\
group by launch_site
```

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

- The above statement groups the data to produce the summary table of launch sites in the dataset.

Launch Site Names Begin with 'CCA'

- SQL statement:

```
%sql\
select *\
from SPACEXTBL\
where launch_site like 'CCA%'\
limit(5)
```

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

- The above statement selects the data where the launch sites start with CCA.

Total Payload Mass

- SQL statement:

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

total_mass
45596

- The above statement filters the data for boosters carried by NASA and sums the total mass.

Average Payload Mass by F9 v1.1

- SQL statement:

```
%sql\
select avg(payload_mass__kg_) as avgscore\
from SPACEXTBL\
where booster_version = 'F9 v1.1'
```

avgscore
2928

- The above statement filters the data for booster version ‘F9 v1.1’ and then averages the payload mass.

First Successful Ground Landing Date

- SQL Statement:

```
%sql\
select min(DATE) as First_Succesful_Landing\
from SPACEXTBL\
where landing__outcome like '%ground pad%'
```

first_succesful_landing
2015-12-22

- The above statement creates a subquery to filter the data for a ground pad landing and then filters the date column for it's minimum data to return the first successful ground landing date.

Successful Drone Ship Landing with Payload between 4000 and 6000

- SQL Statement:

```
%sql\
select booster_version\
from SPACEXTBL\
where payload_mass__kg_ >4000 and
 payload_mass__kg_ <6000 and
 landing__outcome like '%drone%'\
group by booster_version
```

booster_version
F9 FT B1021.2
F9 FT B1031.2
F9 FT B1020
F9 FT B1022
F9 FT B1026

- The above statement filters the data first for payload mass over 4,000 kg, then filters for payload mass under 6,00 kg, and finally for a drone landing outcome. It then returns the booster version.

Total Number of Successful and Failure Mission Outcomes

- SQL Statement:

```
%sql\
select sum(mission_outcome like '%Success%') as Success,
 sum(mission_outcome like '%Fail%') as Failure\
from SPACEXTBL
```

success	failure
100	1

- The above statement creates two variables, one for outcomes that contain success and one for outcomes that contain failure and then retuirns the sum of these outcomes.

Boosters Carried Maximum Payload

- SQL Statement:

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

- The above statement subqueries the data for the boosters carrying the maximum payload, then selects the unique booster versions.

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

- SQL Statement:

```
%sql\
select booster_version, launch_site\
from SPACEXTBL\
where landing__outcome =\
 'Failure (drone ship)' and date like('%2015%')
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

- The above statement filters the booster version and launch site from the data where the outcome was a failed drone ship landing in the year 2015.

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

- SQL Statement:

```
%sql\
select landing__outcome, count(landing__outcome)as rank\
from SPACEXTBL\
where DATE > '2010-06-04' and DATE < '2017-03-20'\
group by landing__outcome\
order by rank DESC
```

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

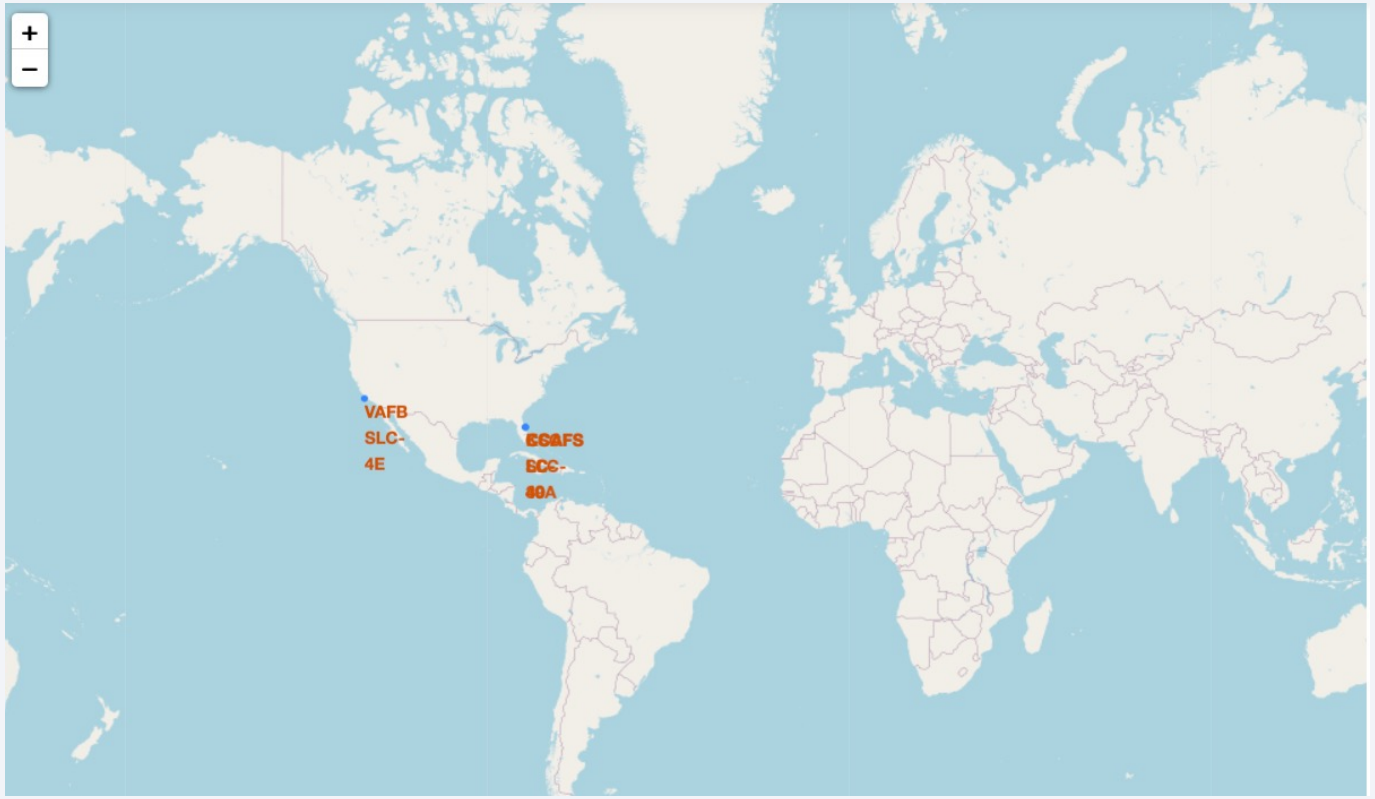
- The above statement groups the landing outcomes and filters the dataset between 2010-06-04 and 2017-03-20 and then orders by the landing outcome rank.

Section 4

Launch Sites Proximities Analysis

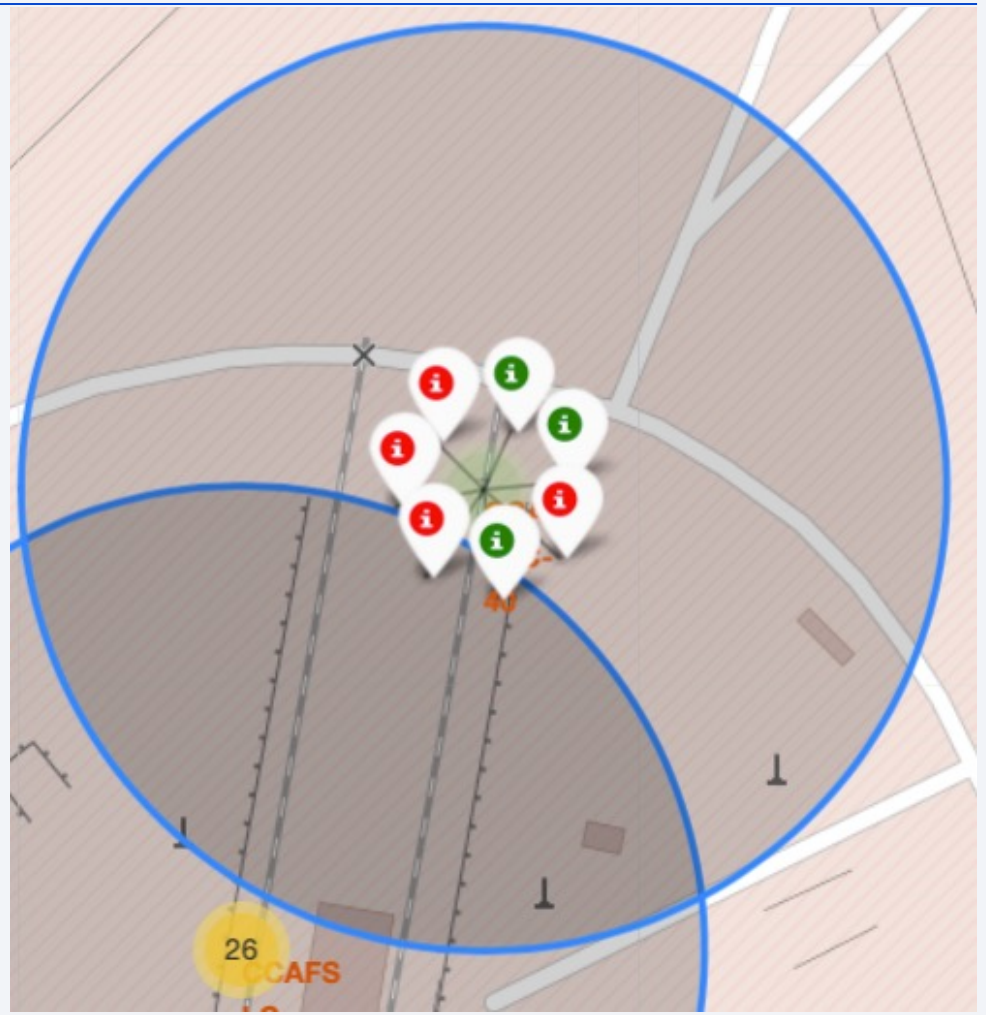
Launch Site Locations

- One launch site is located on the west coast of the United States.
- 3 launch location are located on the east coast of the United States.



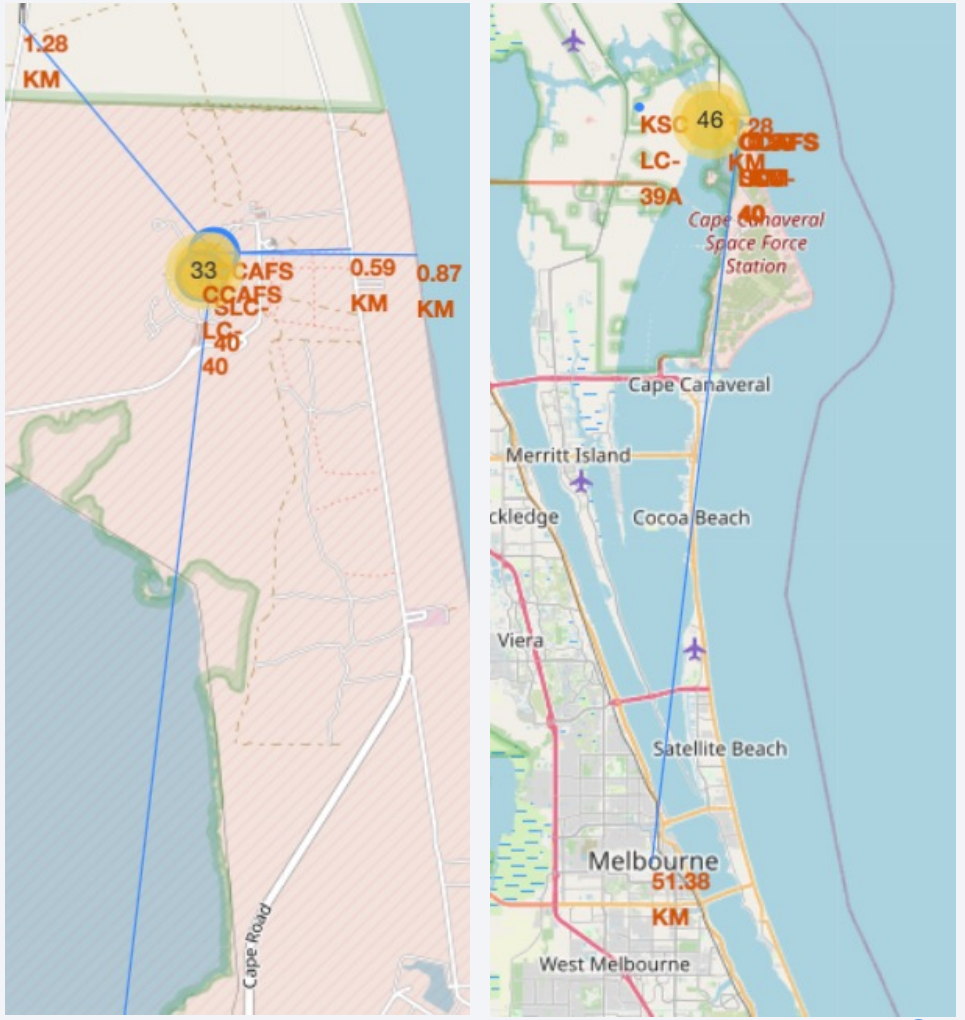
Launch site Outcomes for CCAFS SLC-40

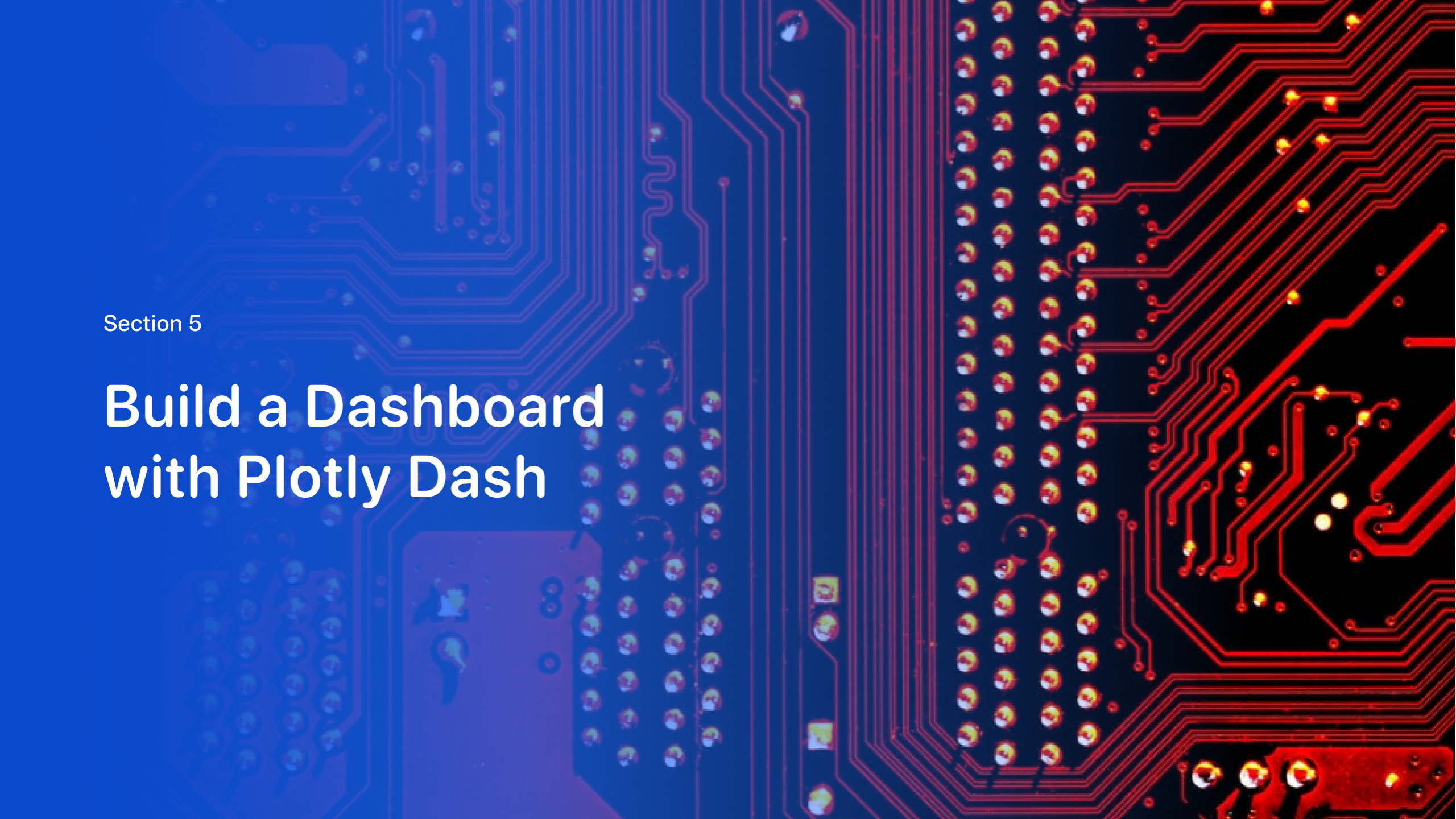
- The map shows the launch location CCAFS SLC-40, located right next to CCAFS LC-40.
- The map shows a total of 7 launches from this site.
- There were 4 unsuccessful landings from this site (indicated in red).
- There were 3 successful launches from this site (indicated in green).



CCAFS SLC-40 Proximity to Landmarks

- CCAFS SLC-40 launch site proximity to landmarks as follows:
  - 1.28 km from a railroad.
  - 0.59 km from a major road.
  - 0.87 km from the ocean.
  - 51.38 km from the nearest city (Melbourne)



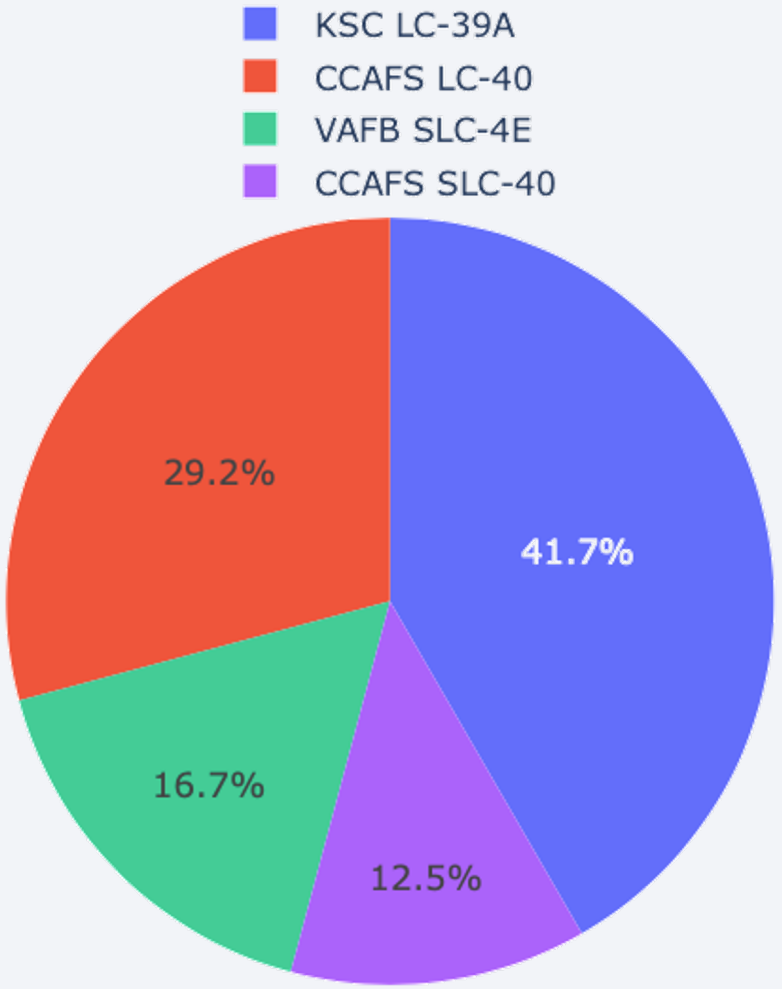


Section 5

Build a Dashboard with Plotly Dash

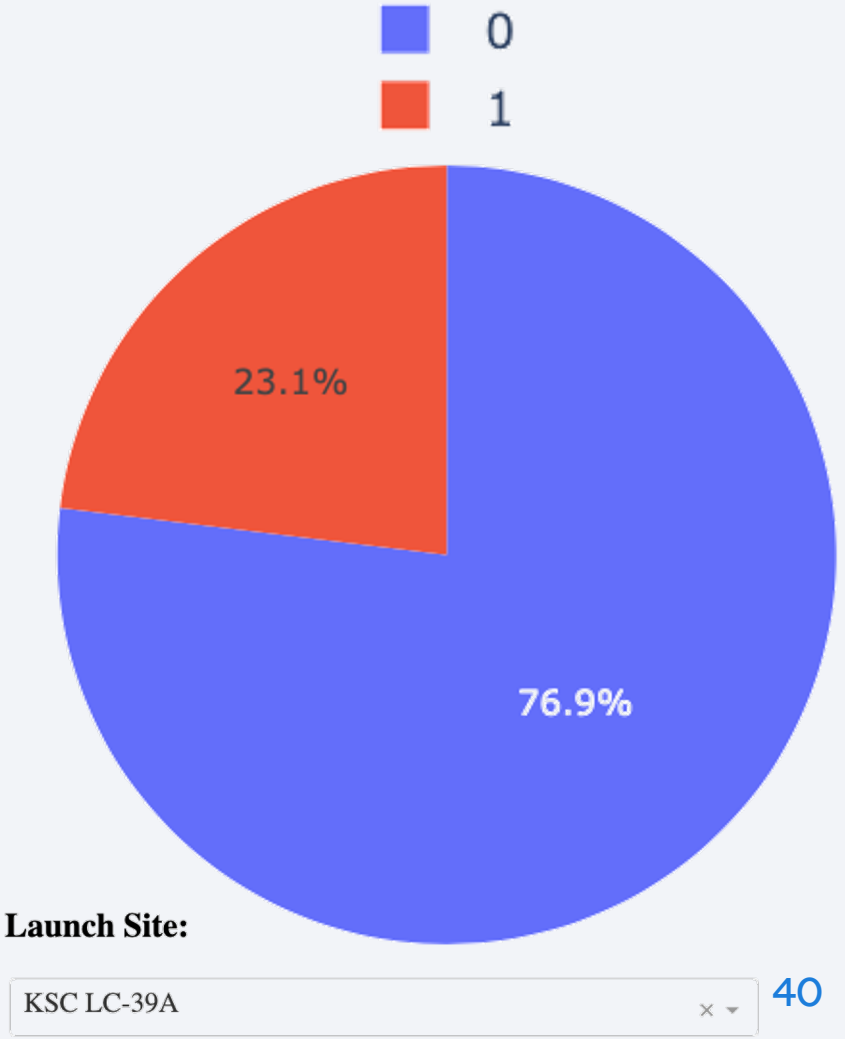
Launch Outcomes Across All Sites

- The pie chart shows that 41.7% of the successful mission outcomes come from KSC LC-391 Site.
- The second most successful mission outcomes are from CCAFS LC-40.
- CCAFS SLC-40 had the least amount of successful mission outcomes.

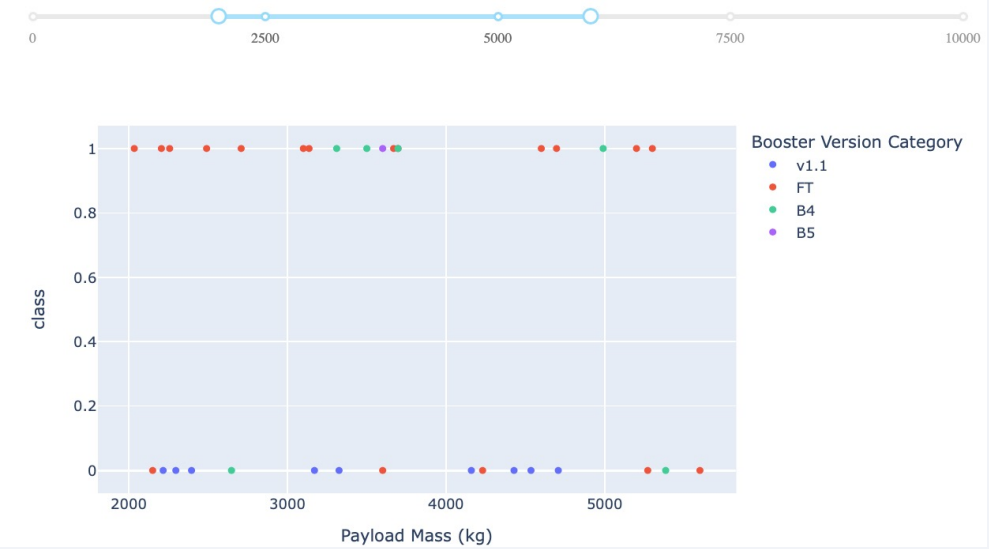


Launch Site KSC LC-39A Success Rate

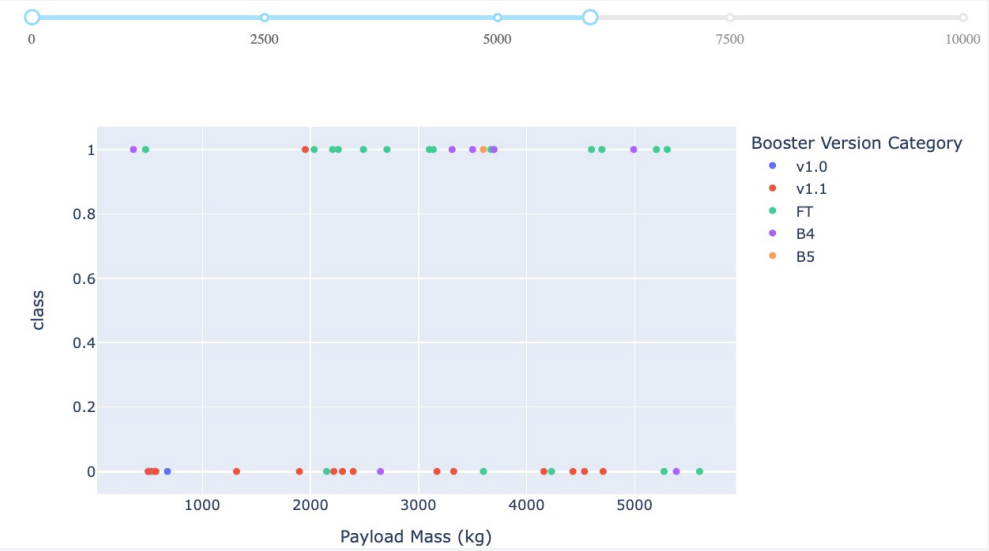
- The launch site KSC LC-391 has the highest success rate amongst all the sites.
- 23.1% failure rate in this site.
- 76.9% success rate.



<Dashboard Screenshot 3>



- The FT booster operated very well in the 2,000 to 6,000 kg payload range.

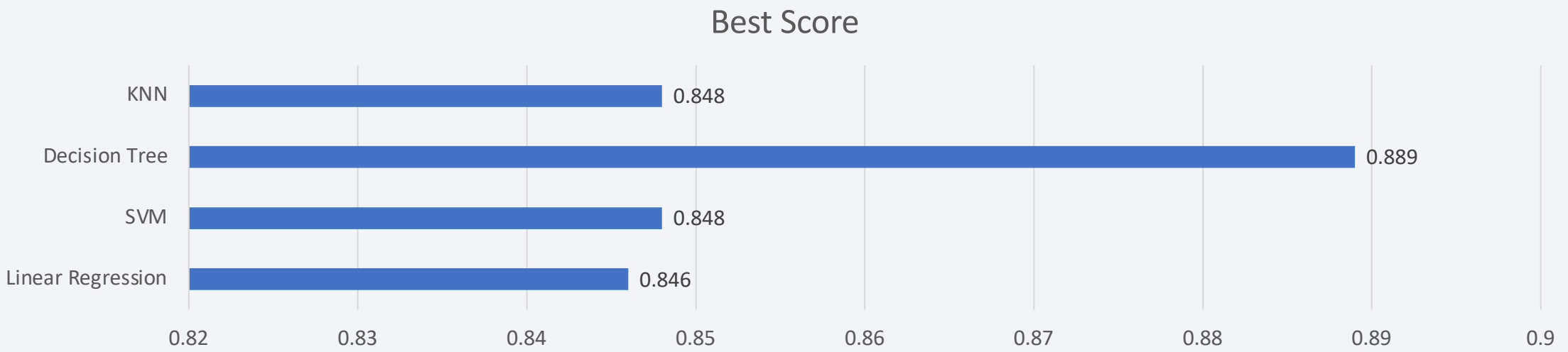


- The B4 booster does not seem to get constant outcomes regardless of payload.

Section 6

Predictive Analysis (Classification)

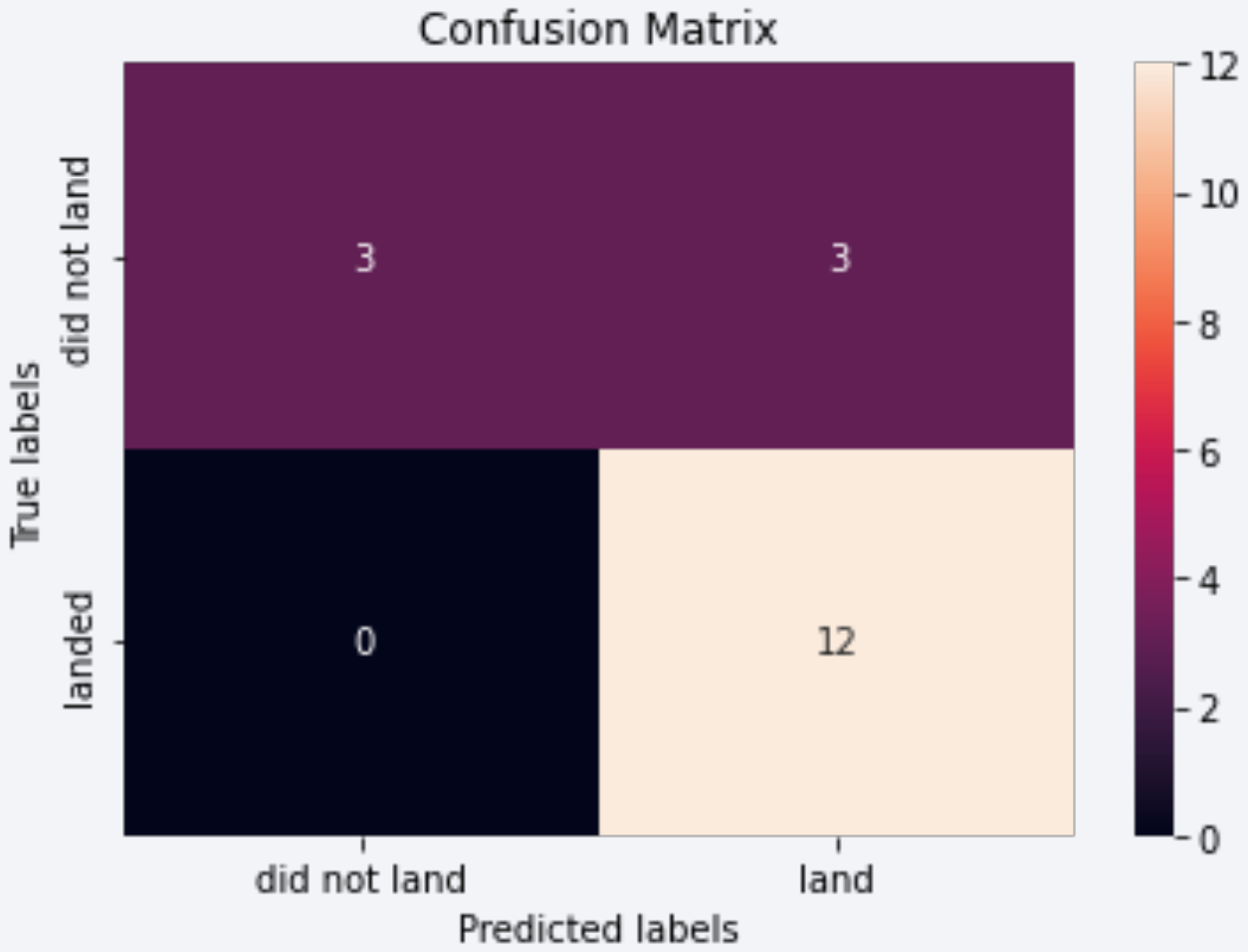
Classification Accuracy



- The decision tree algorithm provides the most accurate results.

Confusion Matrix

- The decision tree confusion matrix is shown here.
- It predicts the true landed variable accurately.
- It does not accurately predict when the outcome is “did not land’



Conclusions

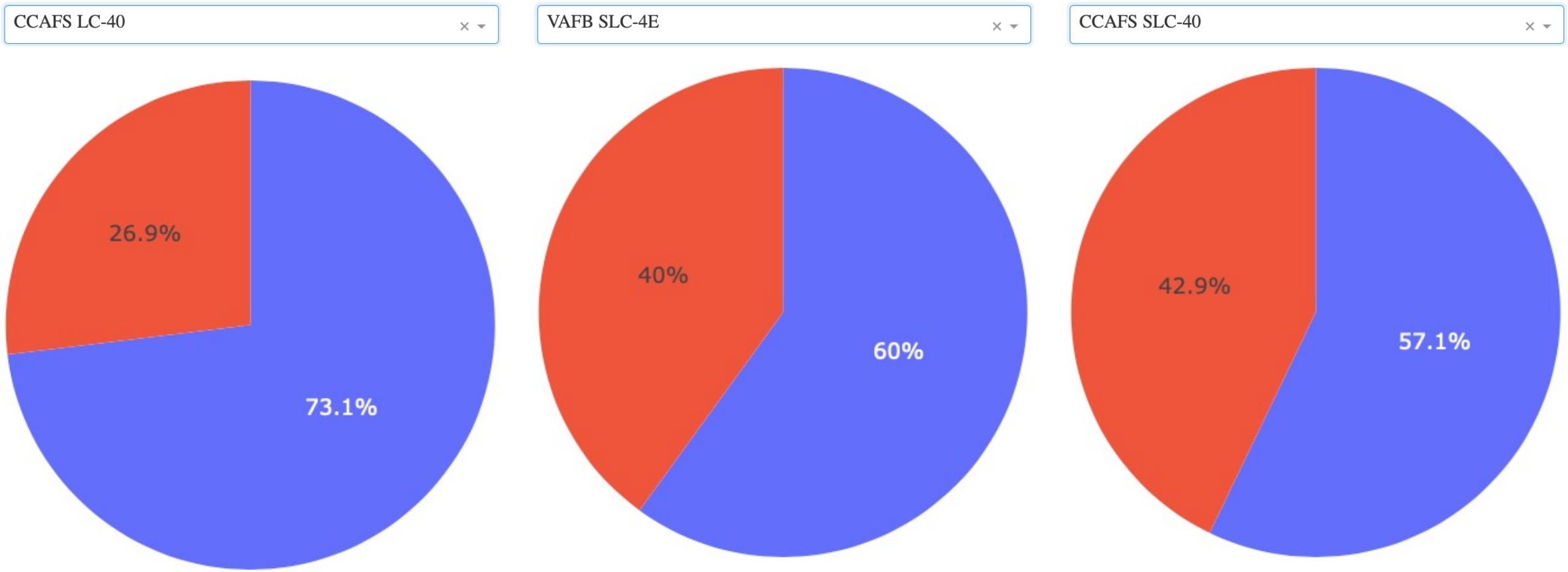
- Launch outcomes have become more positive over time.
- Launch outcomes become more positive with higher payloads.
- Orbits ES-L1, GEO, HEO, and SSO have a 100% success Rate.
- The launch site KSC LC-391 has the highest success rate at 76.9%.
- The FT booster has the highest success rate and works best in a payload range of 2,000 to 6,000 kg range.
- The most successful booster can be chosen based on payload required to be launched.
- Prediction can be made with an 89% accuracy.

Appendix

- Relevant Datasets
  - Dataset 1 - https://github.com/dlevert/DSProject/blob/1058d4efddc46242e8fa28e001fc2cc99c776617/dataset_part_1.csv
  - Dataset 2 - https://github.com/dlevert/DSProject/blob/1058d4efddc46242e8fa28e001fc2cc99c776617/dataset_part_2.csv
  - Dataset 3 - https://github.com/dlevert/DSProject/blob/1058d4efddc46242e8fa28e001fc2cc99c776617/dataset_part_3.csv
  - Dash dataset - https://github.com/dlevert/DSProject/blob/1058d4efddc46242e8fa28e001fc2cc99c776617/spacex_launch_dash.csv
  - Geo Data - https://github.com/dlevert/DSProject/blob/1058d4efddc46242e8fa28e001fc2cc99c776617/spacex_launch_geo.csv
- Python Notebooks
  - Collecting Data - https://github.com/dlevert/DSProject/blob/1058d4efddc46242e8fa28e001fc2cc99c776617/Lab%201%20Collecting%20the%20data.ipynb
  - Data Wrangling - https://github.com/dlevert/DSProject/blob/1058d4efddc46242e8fa28e001fc2cc99c776617/Lab%202%20Data%20wrangling.ipynb
  - SQL - https://github.com/dlevert/DSProject/blob/1058d4efddc46242e8fa28e001fc2cc99c776617/Lab%203%20SQL%20Notebook.ipynb
  - Exploring & Preparing Data - https://github.com/dlevert/DSProject/blob/1058d4efddc46242e8fa28e001fc2cc99c776617/Lab%204%20Exploring%20and%20Preparing%20Data.ipynb
  - Location Analysis - https://github.com/dlevert/DSProject/blob/1058d4efddc46242e8fa28e001fc2cc99c776617/Lab%205%20Analysis%20with%20Folium.ipynb
  - Dash Dashboard - https://github.com/dlevert/DSProject/blob/1058d4efddc46242e8fa28e001fc2cc99c776617/Lab%206%20Pandas%20Dash.ipynb
  - Landing Prediction - https://github.com/dlevert/DSProject/blob/1058d4efddc46242e8fa28e001fc2cc99c776617/Lab%207%20Landing%20Prediction.ipynb
- Github Repository - https://github.com/dlevert/DSProject.git

Appendix

- Other Dashboard Pie Charts

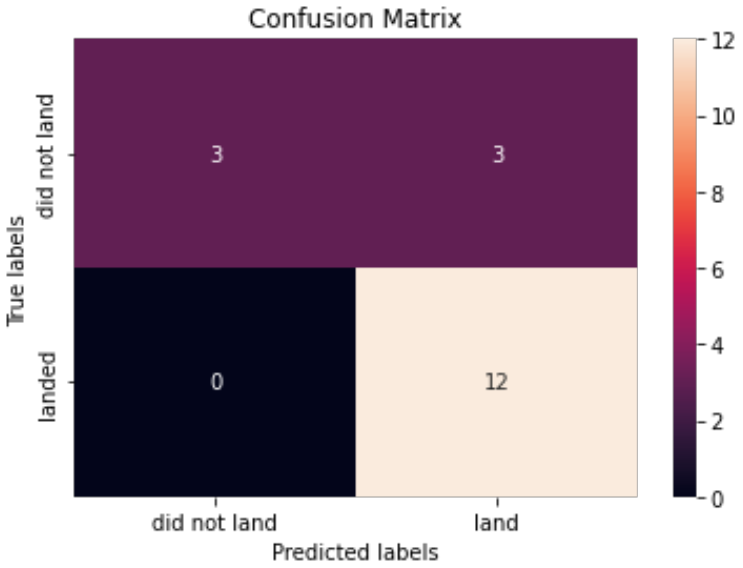


Appendix

Method	Score	Best Score
Linear Regression	0.833333	0.846429
SVM	0.833333	0.848214
Decision Tree	0.833333	0.889286
KNN	0.833333	0.848214

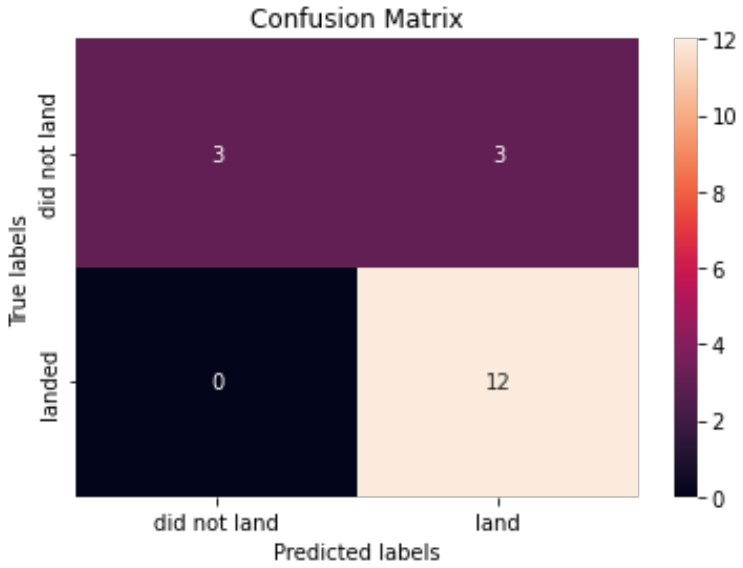
- Other Confusion Matrix results

Logistic Regression



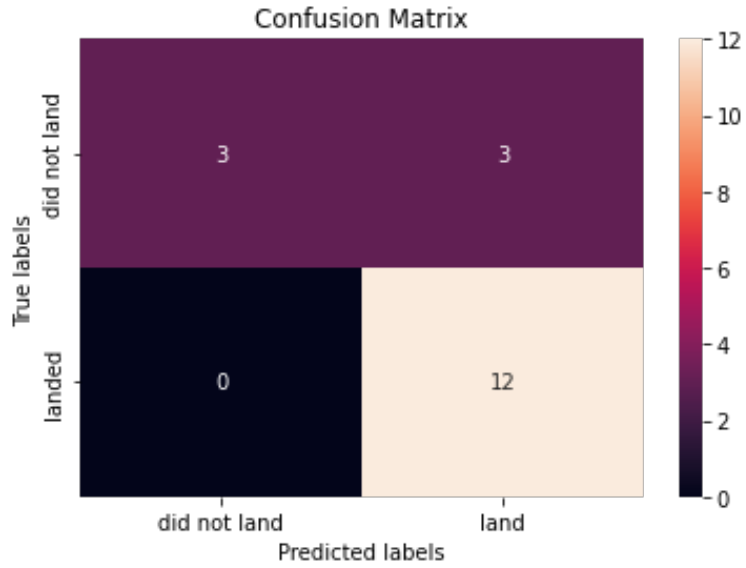
(best parameters) {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
accuracy : 0.8464285714285713

SVM



(best parameters) {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
accuracy : 0.8482142857142856

KNN



(best parameters) {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
accuracy : 0.8482142857142858

Thank you!