IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Franco Cicirello>
<December 15th 2024>

Outline

Executive Summary

Introduction

Methodology

Results

Conclusion

Appendix

Executive Summary

In the rapidly evolving field of space exploration, data science has emerged as a pivotal tool for driving innovation and achieving mission success. This executive summary outlines the strategic application of data science in space missions, highlighting key areas where data-driven insights can lead to significant advancements.

By harnessing the power of data science, space agencies can achieve unprecedented levels of efficiency, accuracy, and innovation. This strategic approach not only enhances mission outcomes but also paves the way for groundbreaking discoveries and advancements in space exploration.

•Scientific Research and Discoveries

Introduction

Space exploration has always been a frontier of human ingenuity and ambition. With the advent of data science, this field has experienced a transformative shift, enabling more precise, efficient, and groundbreaking missions. Data science, with its powerful analytical tools and methodologies, has become an indispensable asset in the quest to explore and understand the cosmos.

The data science has a lot of areas to offer in this situation like Data Collection and Management, Predictive Analytics and Machine Learning, Scientific Research and Discoveries, Resource Optimization, Develop and Collaboration, etc.

Section 1

Methodology

Methodology

Data collection involved sourcing relevant information from various sources including public databases, official reports, and proprietary datasets. This data encompassed information about rocket launches, mission parameters, pricing strategies, and historical performance metrics of companies such as SpaceX.

- **Data Wrangling:**

Upon collecting the raw data, a meticulous data wrangling process was undertaken to clean, transform, and preprocess the data. This involved handling missing values, removing duplicates, standardizing formats, and ensuring data integrity to prepare the dataset for further analysis.

- **Data Processing:**

Following data wrangling, the processed dataset underwent further processing to extract meaningful insights. This involved aggregating data, calculating relevant metrics, and structuring the dataset in a format suitable for analysis.

- **Exploratory Data Analysis (EDA):**

EDA was conducted using a combination of visualization techniques and SQL queries to uncover patterns, trends, and relationships within the dataset. Visualization tools such as Matplotlib, Seaborn, and SQL queries facilitated the exploration of key metrics and provided valuable insights into the commercial space industry.

- **Interactive Visual Analytics:**

Interactive visual analytics were performed using advanced visualization libraries such as Folium and Plotly Dash. These tools enabled the creation of interactive dashboards and maps, allowing stakeholders to explore data dynamically and gain deeper insights into market dynamics, launch patterns, and geographical trends.

Data Collection

- **Identifying Data Source:**
  Recognizing the need for SpaceX launch data for the capstone assignment.
  Selecting the SpaceX REST API (api.spacexdata.com/v4/) as the primary data source.
- **API Endpoint Selection:**
  Focusing on the endpoint api.spacexdata.com/v4/launches/past to obtain past launch data.
  Understanding that this endpoint provides information about historical launches.
- **Performing GET Request:**
  Utilizing the requests library in Python to perform a GET request to the specified endpoint.
  Initiating the request to retrieve launch data from the API.
- **Response Handling:**
  Receiving a JSON response from the API containing a list of JSON objects, with each representing a launch.
  Recognizing that the JSON format requires conversion into a tabular format for further analysis.

Data Transformation:
Using the json_normalize function to convert the structured JSON data into a flat table format.
Understanding that this transformation facilitates easier manipulation and analysis of the data.

Web Scraping:
Considering alternative data sources, such as web scraping Wikipedia pages for Falcon 9 launch records.
Utilizing the Python BeautifulSoup package to extract data from HTML tables.

Data Parsing and Conversion:
Parsing the extracted data from HTML tables and converting it into a Pandas DataFrame.
Ensuring that the DataFrame is structured appropriately for visualization and analysis.

Filtering Data:
Identifying the need to filter out Falcon 1 launches from the dataset.
Implementing filtering or sampling techniques to focus exclusively on Falcon 9 launches.

Handling Null Values:

Recognizing the presence of null values, particularly in the PayloadMass column.

Devising a method to calculate the mean of the PayloadMass data and replacing null values with this mean.

Acknowledging that the LandingPad column may contain null values, representing instances where a landing pad was not used.

Data Quality Assurance:

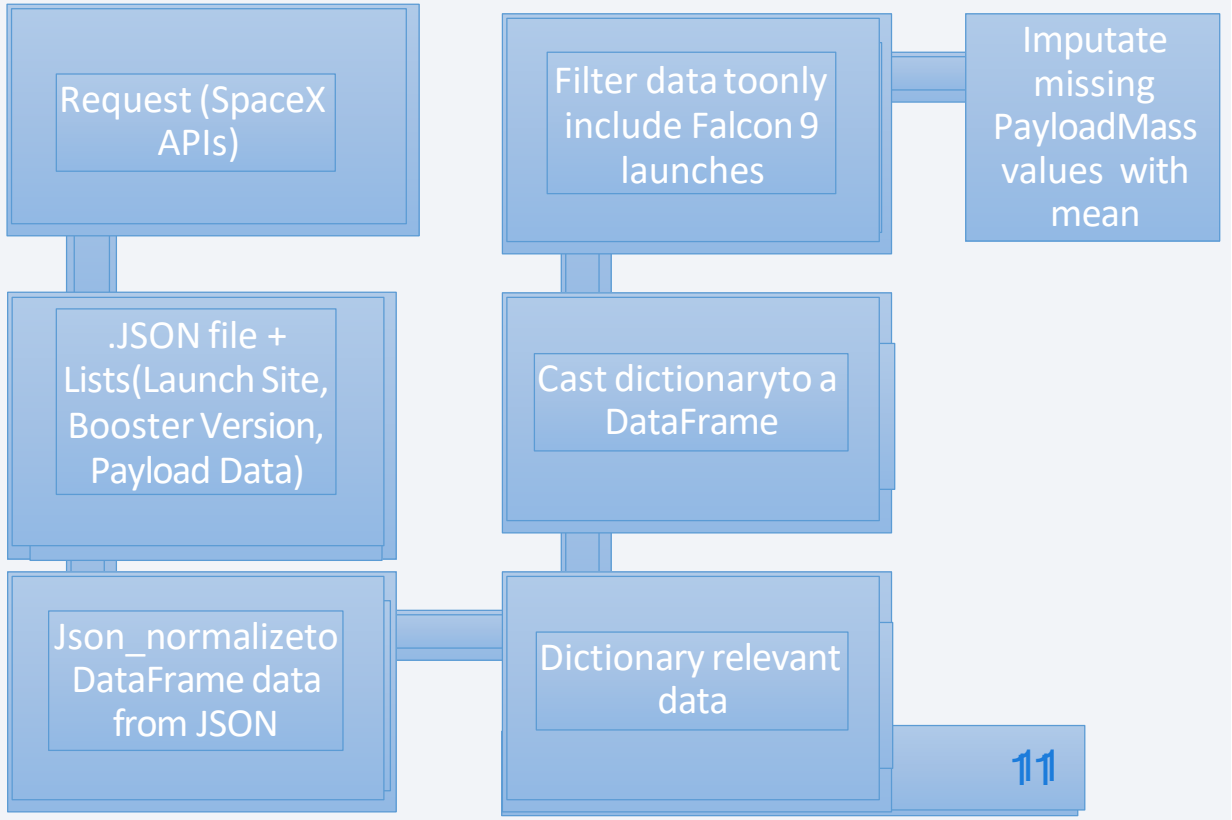
Ensuring that the collected datasets are clean, complete, and ready for further analysis.

Validating the integrity and accuracy of the collected data to mitigate potential errors or inconsistencies.

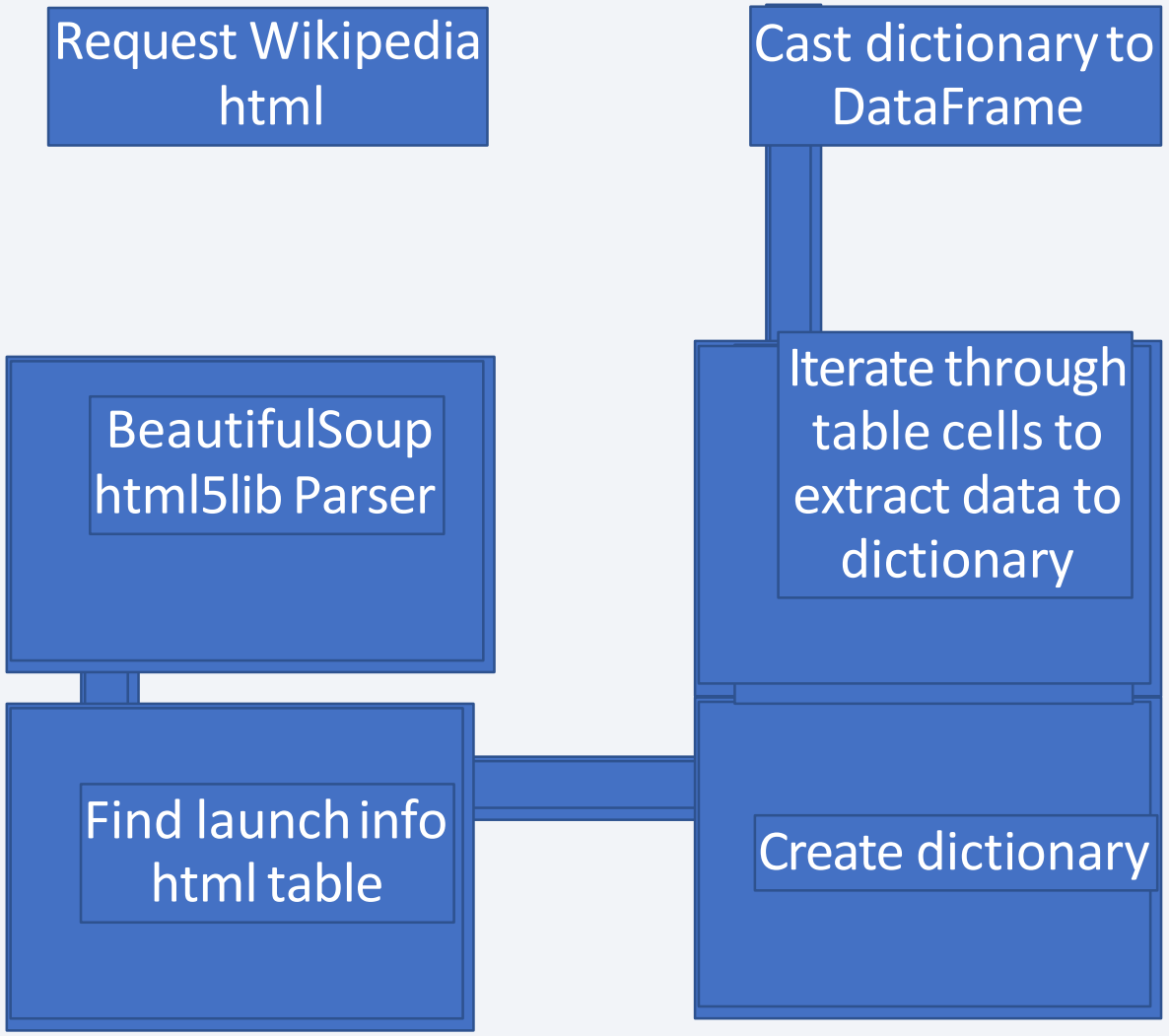
Data Collection - SpaceX API

Data Collection - SpaceX API

git@github.com:DanArtCruzCast/Applied-Data-Science-with-Capstone.git



Data Collection - Scraping



Data Wrangling

- Create a training label with landing outcomes where successful = 1 & failure = 0.
- Outcome column has two components: ‘Mission Outcome’ ‘Landing Location’
- New training label column ‘class’ with a value of 1 if ‘Mission Outcome’ is True and 0 otherwise. Value Mapping:
- True ASDS, True RTLS, & True Ocean – set to -> 1
- None None, False ASDS, None ASDS, False Ocean, False RTLS – set to -> 0

- https://github.com/DanArtCruzCast/Applied-Data-Science-with-Capstone/blob/main/Data%20wrangling%20.ipynb

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

- https://github.com/DanArtCruzCast/Applied-Data-Science-with-Capstone/blob/main/EDA%20with%20Visualization.ipynb

EDA with SQL

- Loaded data set into IBM DB2 Database.
- Queried using SQL Python integration.
- Queries were made to get a better understanding of the dataset.
- Queried information about launch site names, mission outcomes, various pay load sizes of customers and booster versions, and landing outcomes

- https://github.com/DanArtCruzCast/Applied-Data-Science-with-Capstone/blob/main/EDA%20with%20SQL.ipynb

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.

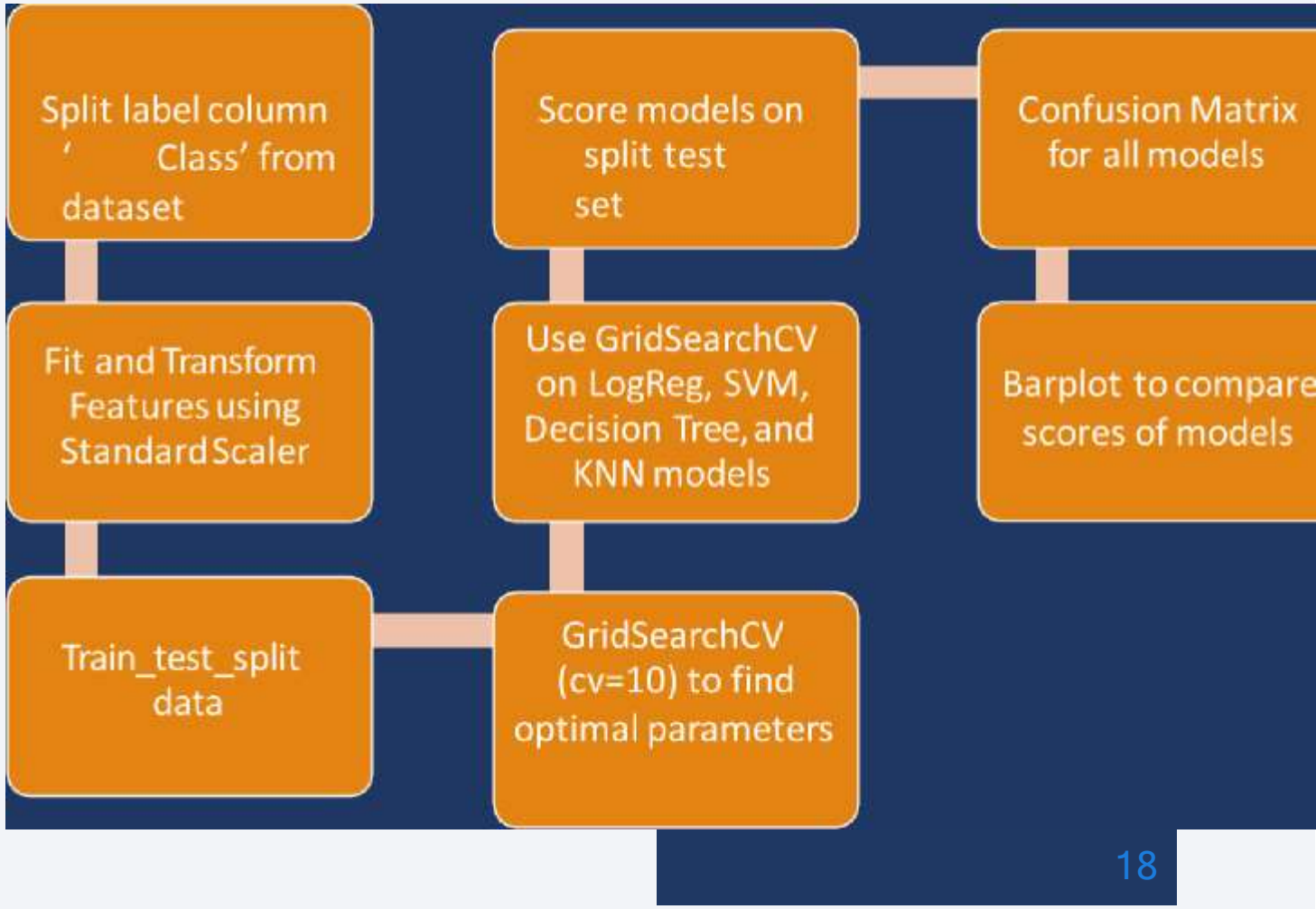
- https://github.com/DanArtCruzCast/Applied-Data-Science-withCapstone/blob/main/Interactive%20Visual%20Analytics%20with%20Folium-checkpoint.ipynb

Build a Dashboard with Plotly Dash

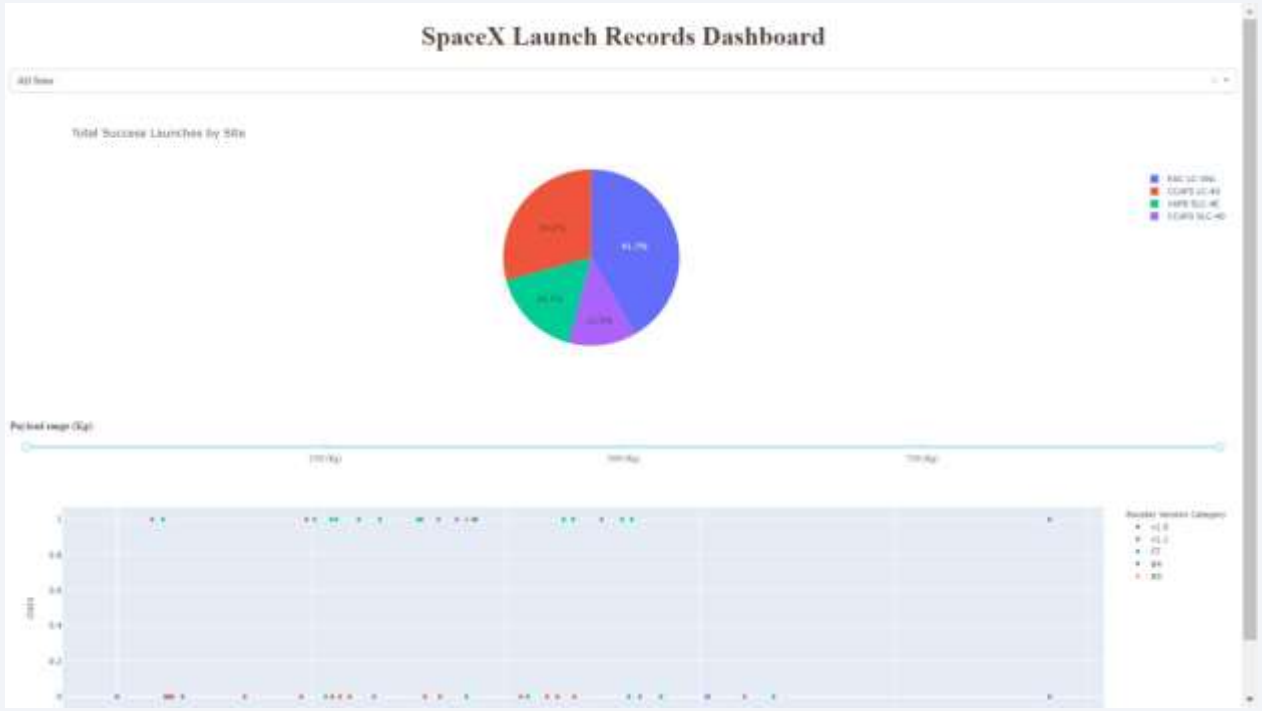
- Dashboard includes a pie chart and a scatter plot.
- Pie chart can be selected to show distribution of successful landings across all launch sites and can be selected to show individual launch site success rates.
- Scatter plot takes two inputs: All sites or individual site and payload mass on a slider 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.
- https://github.com/ DanArtCruzCast /10.-Applied-Data-Science-Capstone/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

https://github.com/DanArtCruzCast/Applied-Data-Science-with-Capstone/blob/main/Machine%20Learning%20Prediction-checkpoint.ipynb



Results

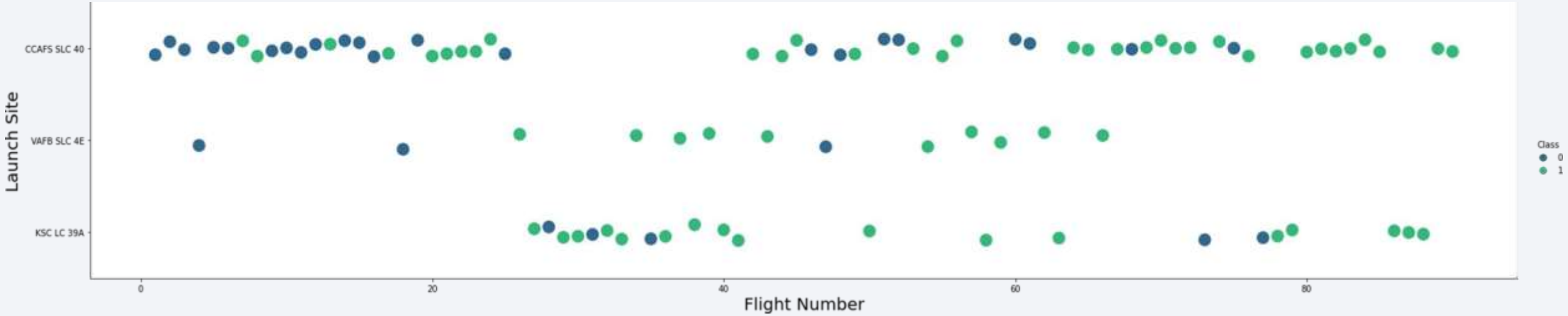


Section 2

Insights drawn from EDA

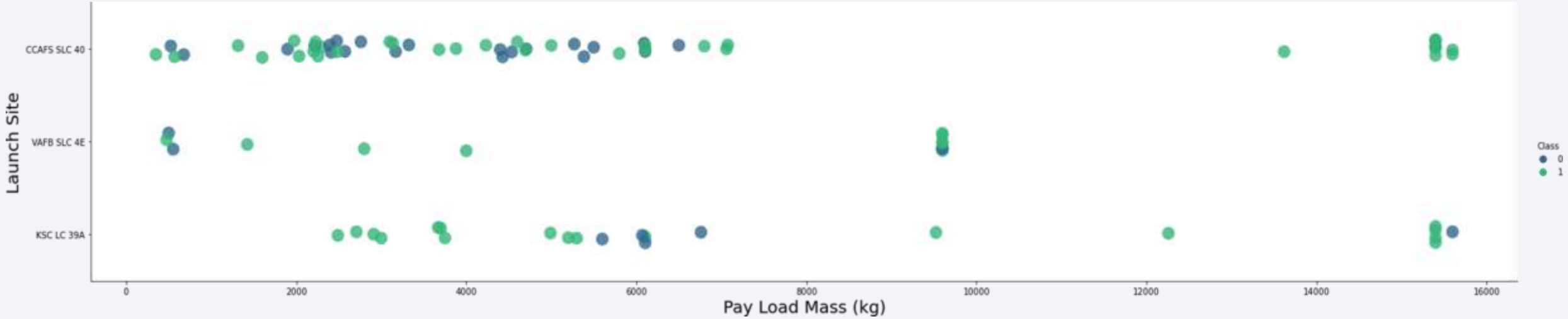
Flight Number vs. Launch Site

- Graphic suggests an increase in success rate over time (indicated in Flight Number). Likely a big breakthrough around flight 20 which significantly increased success rate. CCAFS appears to be the main launch site as it has the most volume.

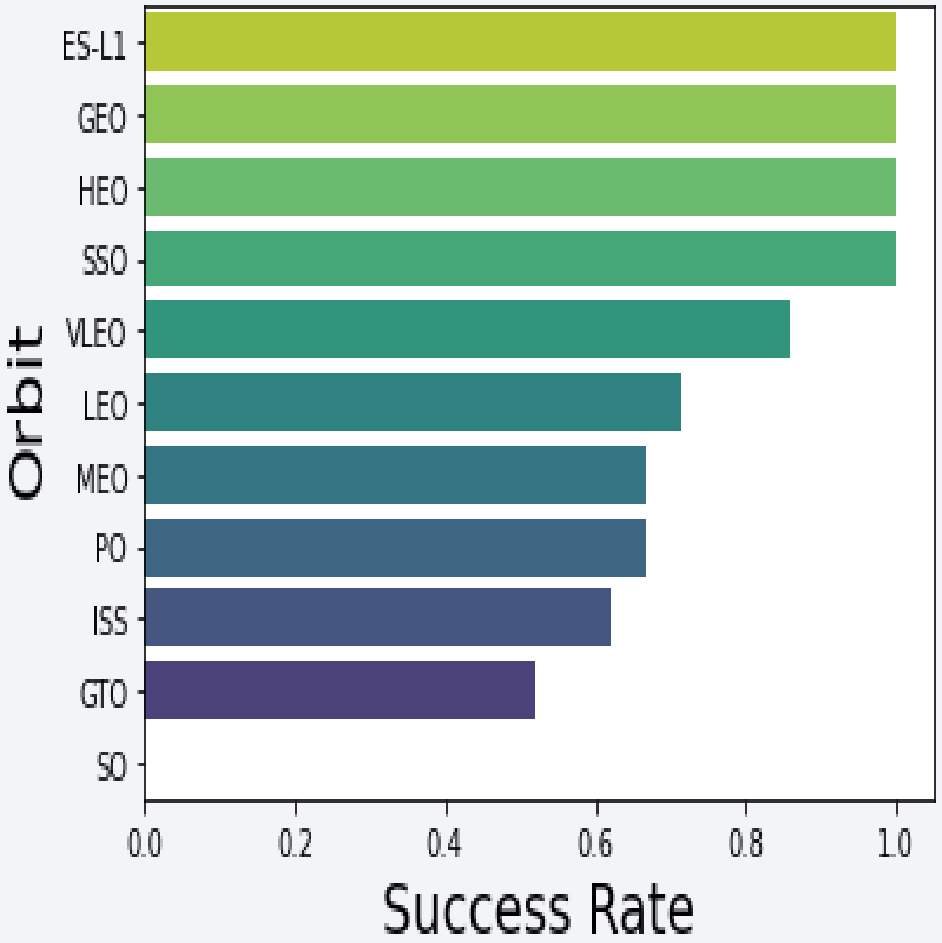


Payload vs. Launch Site

- Payload mass appears to fall mostly between 0-6000 kg. Different launch sites also seem to use different payload mass.



Success Rate vs. Orbit Type



Success Rate Scale with 0 as 0%
0.6 as 60% 1 as 100%

ES-L1 (1), GEO (1), HEO (1) have 100% success rate (sample sizes in parenthesis) SSO (5) has 100% success rate

VLEO (14) has decent success rate and attempts

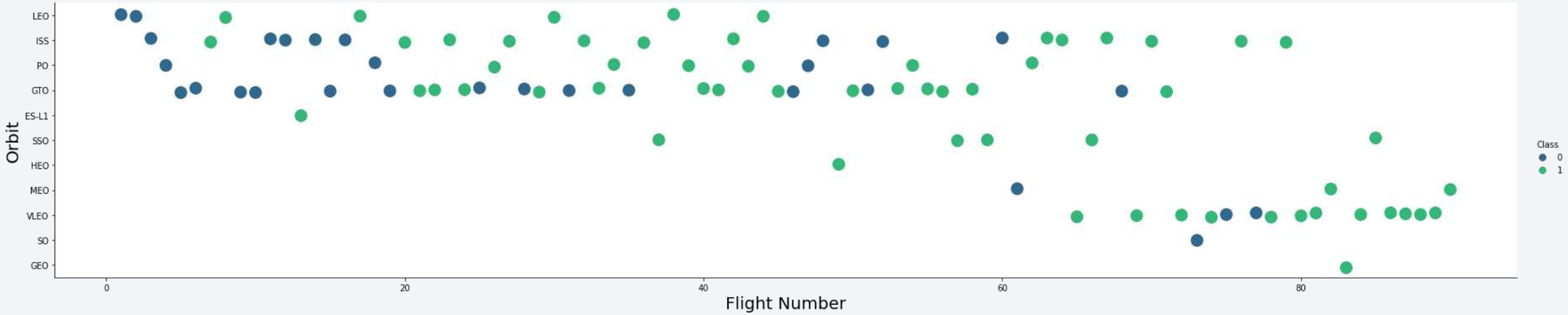
SO (1) has 0% success rate

GTO (27) has the around 50% success rate but largest sample

Flight Number vs. Orbit Type

Launch Orbit preferences changed over Flight Number. Launch Outcome seems to correlate with this preference.

SpaceX started with LEO orbits which saw moderate success LEO and returned to VLEO in recent launches
SpaceX appears to perform better in lower orbits or Sun-synchronous orbits



- Green indicates successful launch; Purple indicates unsuccessful launch.

Payload vs. Orbit Type

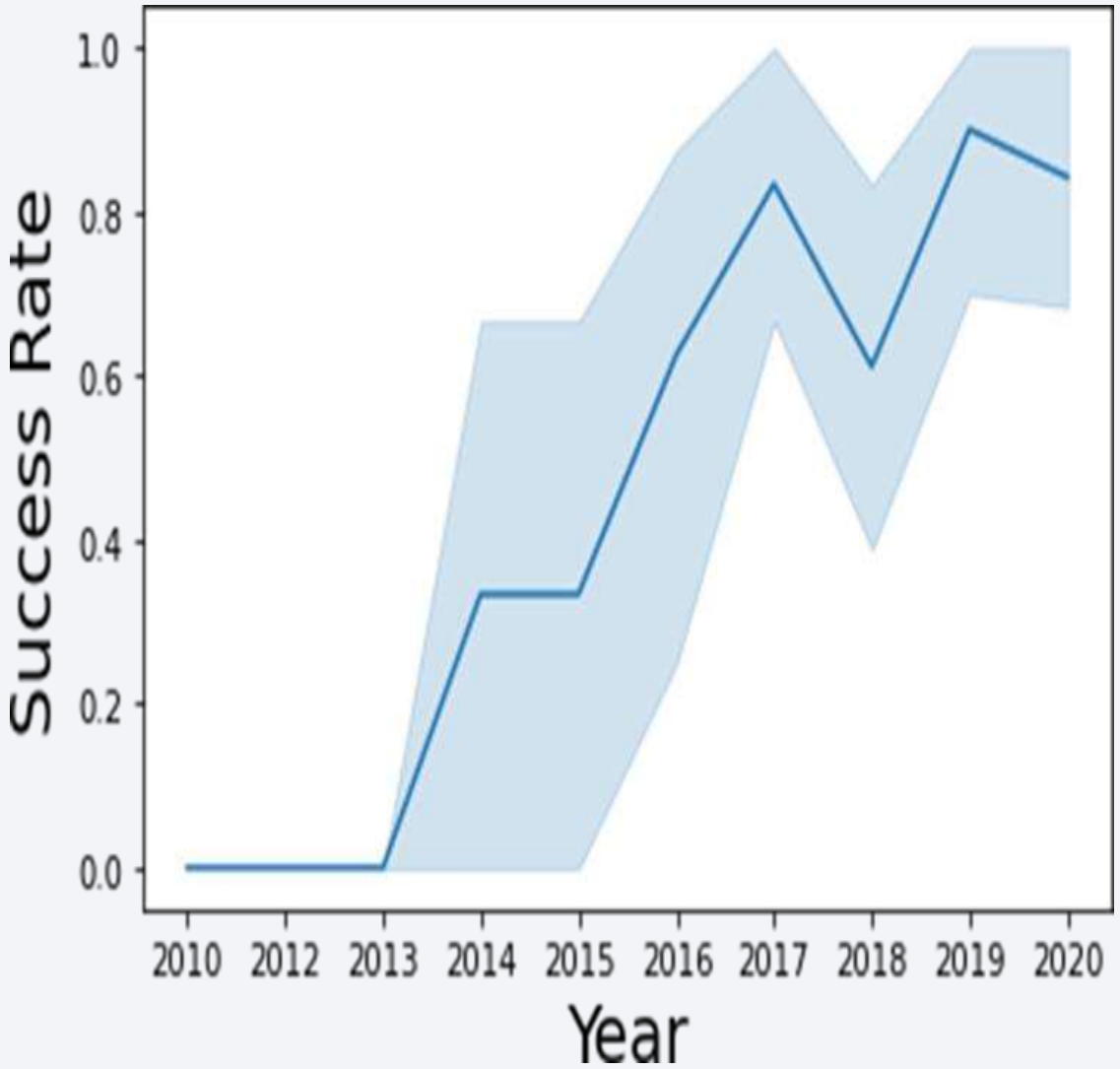
Payload mass seems to correlate with orbit

LEO and SSO seem to have relatively low payload mass

The other most successful orbit VLEO only has payload mass values in the higher end of the range

Green indicates successful launch; Purple indicates unsuccessful launch.

Launch Success Yearly Trend



Success generally increases over time since 2013 with a slight dip in 2018

Success in recent years at around 80%

95% confidence interval (light blue shading)

All Launch Site Names

```
In [4]: %%sql
        SELECT UNIQUE LAUNCH_SITE
        FROM SPACEXDATASET;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2
Done.
```

Out[4]:

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

Query unique launch site names from database.

CCAFS SLC-40 and CCAFSSLC-40 likely all represent the same

launch site with data entry errors.

CCAFS LC-40 was the previous name.
Likely only 3 unique launch_site values:
CCAFS SLC-40, KSC LC-39A, VAFB SLC-4E

Launch Site Names Begin with 'CCA'

In [5]:
```
%%sql
SELECT *
FROM SPACEXDATASET
WHERE LAUNCH_SITE LIKE 'CCA%'
LIMIT 5;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[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

First five entries in database with Launch Site name beginning with CCA

Total Payload Mass

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) AS SUM_PAYLOAD_MASS_KG
FROM SPACEXDATASET
WHERE CUSTOMER = 'NASA (CRS)';
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86
Done.
```

sum_payload_mass_kg
45596

- This query sums the total payload mass in kg where NASA was the customer.
- CRS stands for Commercial Resupply Services which indicates that these payloads were sent to the International Space Station (ISS).

Average Payload Mass by F9 v1.1

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS_KG
FROM SPACEXDATASET
WHERE booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-8(
Done.
```

avg_payload_mass_kg
2928

- This query calculates the average payload mass or launches which used booster version F9 v1.1
- Average payload mass of F9 1.1 is on the low endof our payload mass range

First Successful Ground Landing Date

```
%%sql
SELECT MIN(DATE) AS FIRST_SUCCESS
FROM SPACEXDATASET
WHERE landing__outcome = 'Success (ground pad)';
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81
Done.
```

first_success
2015-12-22

This query returns the first successful ground pad landing date.

First ground pad landing wasn't until the end of 2015.

Successful landings in general appear starting 2014

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
SELECT booster_version
FROM SPACEXDATASET
WHERE landing__outcome = 'Success (drone ship)' AND payload_mass__kg_ BETWEEN 4001 AND 5999;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.database
Done.
```

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

- This query returns the four booster versions that had successful drone ship landings and a payload mass between 4000 and 6000 noninclusively.

Total Number of Successful and Failure Mission Outcomes

```
%%sql
SELECT mission_outcome, COUNT(*) AS no_outcome
FROM SPACEXDATASET
GROUP BY mission_outcome;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-
Done.
```

mission_outcome	no_outcome
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

This query returns a count of each

mission outcome.

SpaceX appears to achieve its mission outcome nearly 99% of the time.

This means that most of the landing

failures are intended.

Interestingly, one launch has an unclear payload status and unfortunately one failed in flight.

Boosters Carried Maximum Payload

```
%%sql
SELECT booster_version, PAYLOAD_MASS__KG_
FROM SPACEXDATASET
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXDATASET);
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1
Done.
```

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

This query returns the booster versions that carried the highest payload mass of 15600 kg.

These booster versions are very similar and all are of the F9 B5 B10xx.x variety.

This likely indicates payload mass correlates with the booster version that is used

2015 Launch Records

```
%%sql
SELECT MONTHNAME(DATE) AS MONTH, landing__outcome, booster_version, PAYLOAD_MASS__KG_, launch_site
FROM SPACEXDATASET
WHERE landing__outcome = 'Failure (drone ship)' AND YEAR(DATE) = 2015;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.app
Done.
```

MONTH	landing__outcome	booster_version	payload_mass__kg_	launch_site
January	Failure (drone ship)	F9 v1.1 B1012	2395	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	1898	CCAFS LC-40

This query returns the Month, Landing Outcome, Booster Version, Payload Mass (kg), and Launch site of 2015 launches where stage 1 failed to land on a drone ship.

There were two such occurrences.

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

```
%%sql
SELECT landing__outcome, COUNT(*) AS no_outcome
FROM SPACEXDATASET
WHERE landing__outcome LIKE 'Succes%' AND DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY landing__outcome
ORDER BY no_outcome DESC;
```

```
 * ibm_db_sa://ftb12020:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg
Done.
```

landing__outcome	no_outcome
Success (drone ship)	5
Success (ground pad)	3

This query returns a list of successful landings and between 2010-06-04 and 2017-03-20 inclusively.

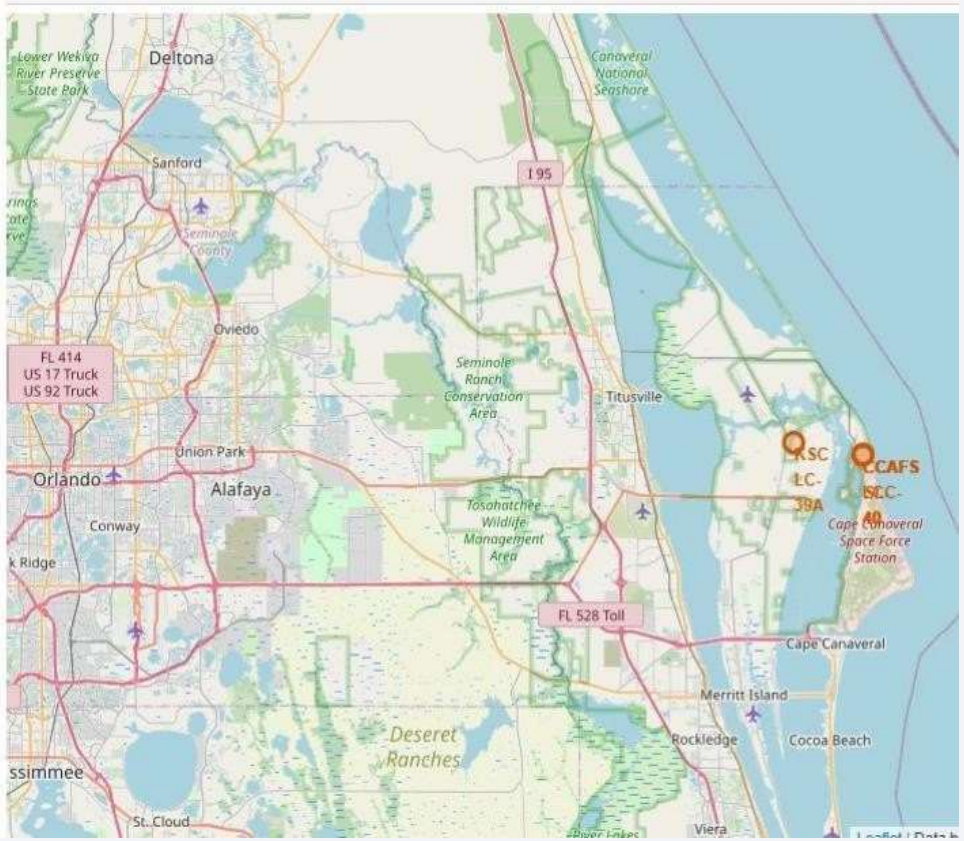
There are two types of successful landing outcomes: drone ship and ground pad landings.

There were 8 successful landings in total during this time period

Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>



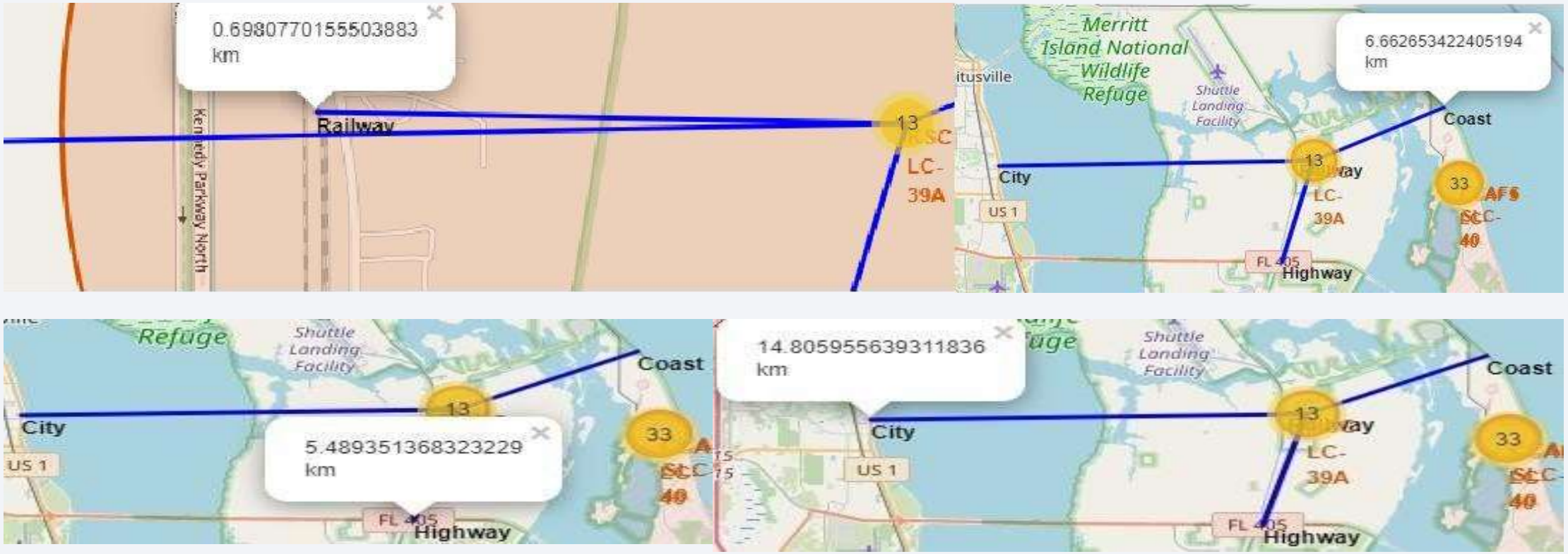
The left map shows all launch sites relative US map. The right map shows the two Florida launch sites since they are very close to each other. All launch sites are near the ocean.

<Folium Map Screenshot 2>

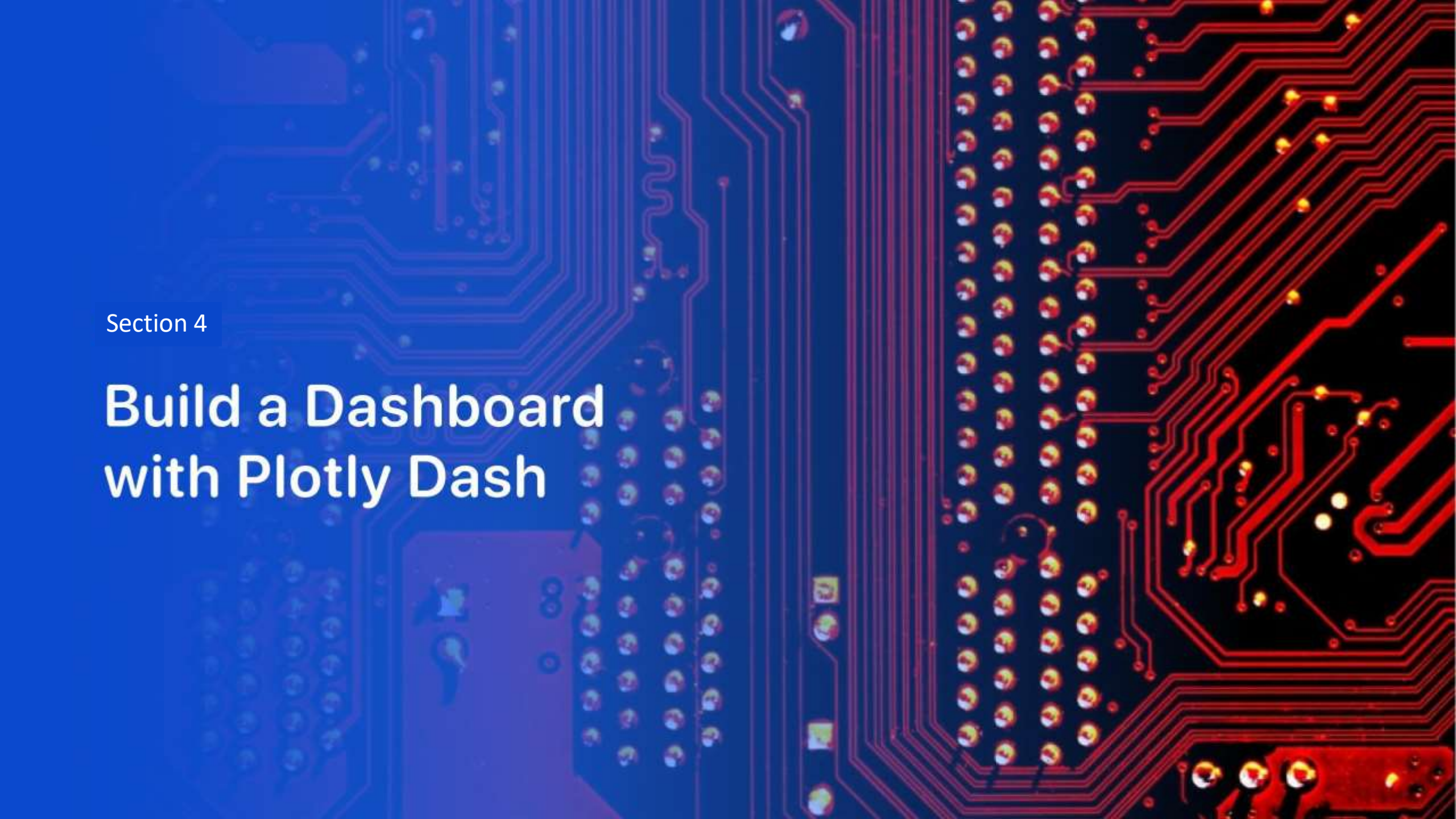
Clusters on Folium map can be clicked on to display each successful landing (green icon) and failed

landing (red icon). In this example VAFB SLC-4E shows 4 successful landings and 6 failed landings.

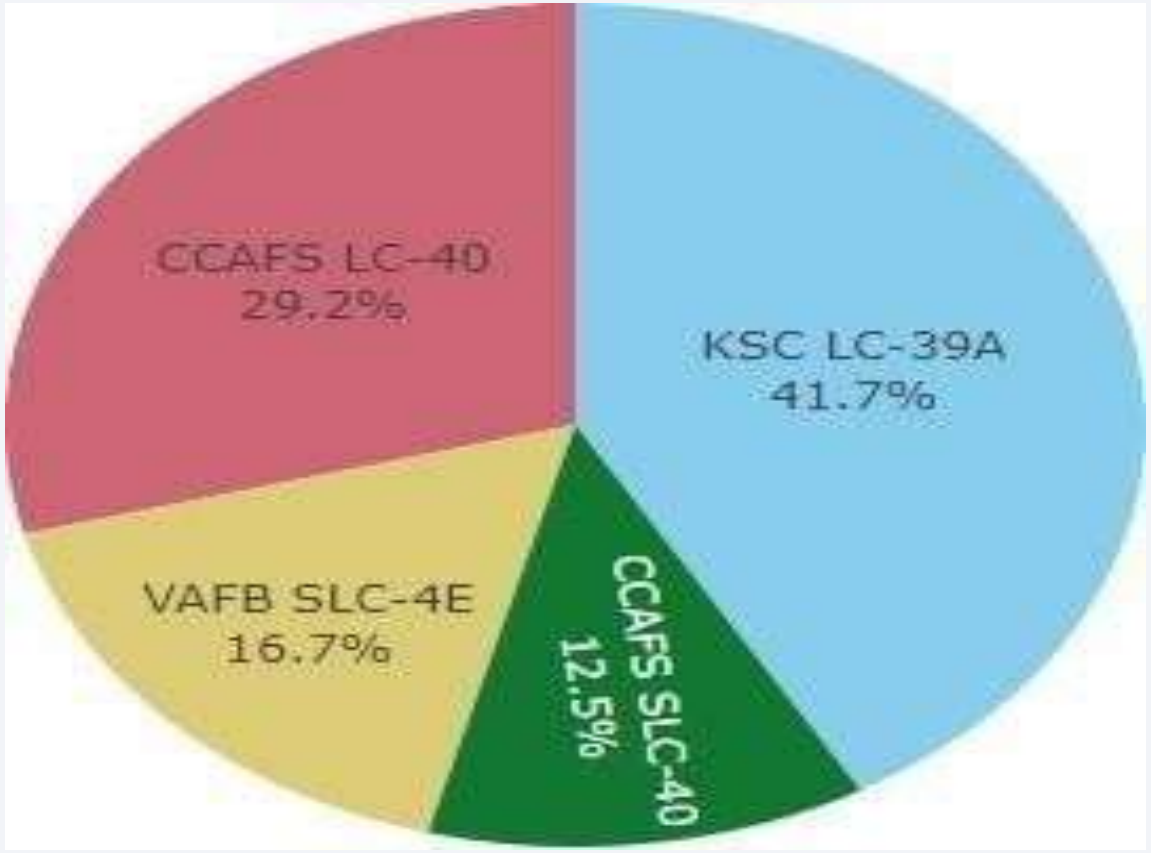
<Folium Map Screenshot 3>



Using KSC LC-39A as an example, launch sites are very close to railways for large part and supply transportation. Launch sites are close to highways for human and supply transport. Launch sites are also close to coasts and relatively far from cities so that launch failures can land in the sea to avoid rockets falling on densely populated areas.

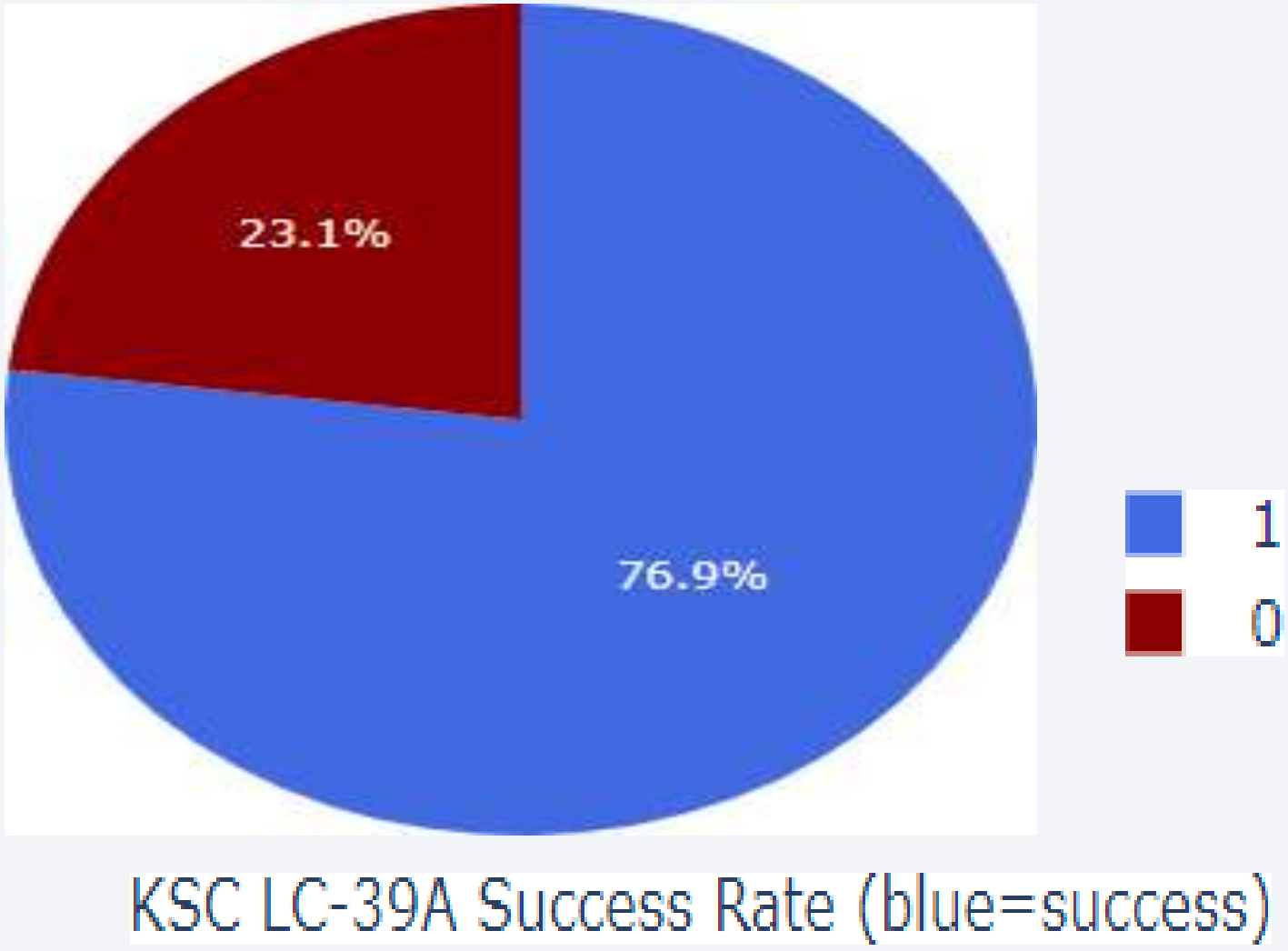

Section 4
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>



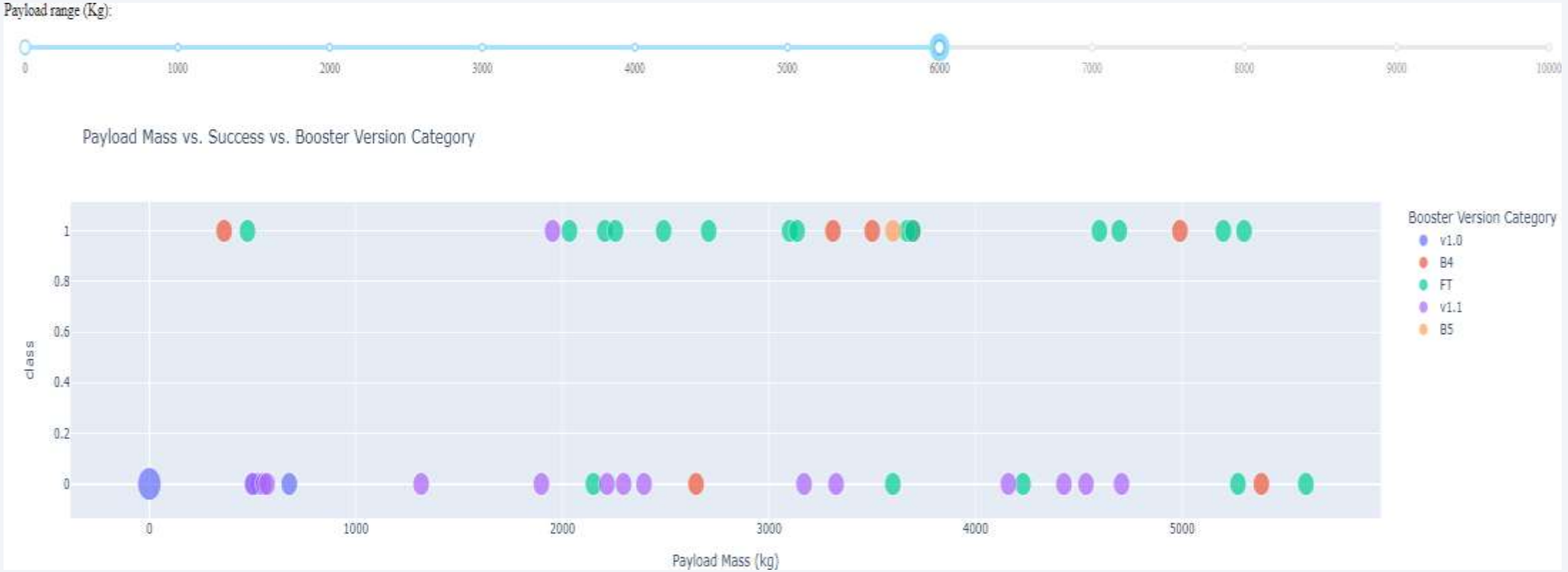
This is the distribution of successful landings across all launch sites. CCAFS LC-40 is the old name of CCAFS SLC-40 so CCAFS and KSC have the same amount of successful landings, but a majority of the successful landings where performed before the name change. VAFB has the smallest share of successful landings. This may be due to smaller sample and increase in difficulty of launching in the west coast.

<Dashboard Screenshot 2>



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

<Dashboard Screenshot 3>

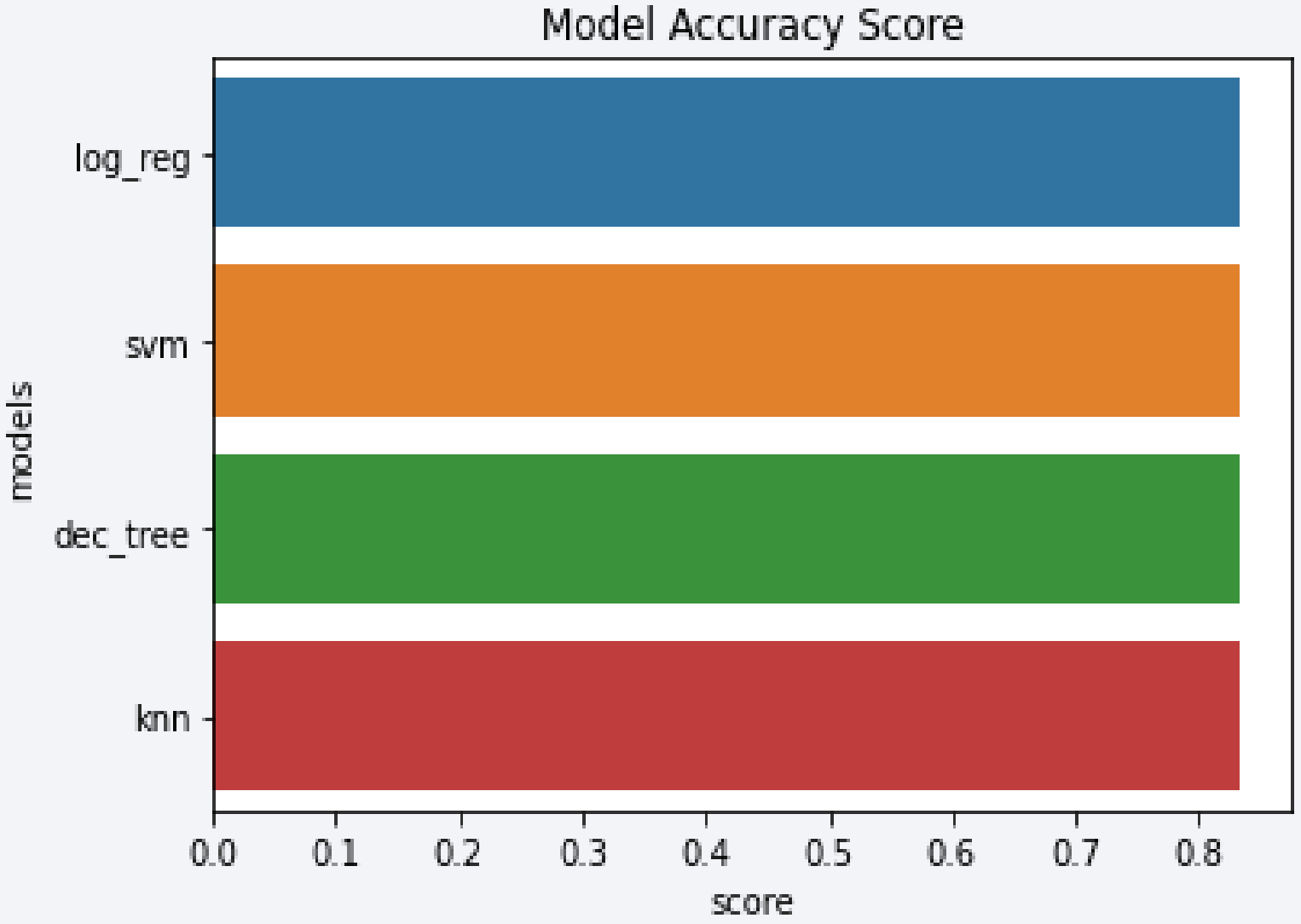


Plotly dashboard has a Payload range selector. However, this is set from 0-10000 instead of the max Payload of 15600. Class indicates 1 for successful landing and 0 for failure. Scatter plot also accounts for booster version category in color and number of launches in point size. In this particular range of 0-6000, interestingly there are two failed landings with payloads of zero kg.

Section 5

Predictive Analysis (Classification)

Classification Accuracy



All models had virtually the same accuracy on the test set at 83.33% accuracy. It should be noted that test size is small at only sample size of 18.

This can cause large variance in accuracy results, such as those in Decision Tree Classifier model in repeated runs.

We likely need more data to determine the best model.

Confusion Matrix

Correct predictions are on a diagonal from top left to bottom right.

Since all models performed the same for the test set, the confusion matrix is the same across all models. The models predicted 12 successful landings when the true label was successful landing.

The models predicted 3 unsuccessful landings when the true label was unsuccessful landing.

The models predicted 3 successful landings when the true label was unsuccessful landings (false positives). Our models over predict successful landings.

Conclusions

- In a very specific and mathematical way, with the help of software, it is possible to analyze SpaceX's business models and how to create a fictitious company like its competitor and thus infer in which models or characteristics of rockets it has had a higher success rate and how to replicate it in those that have not, that is, the areas of opportunity that generate a strategy of continuous improvement.

Appendix

- import plotly.express as px

- # Crear un gráfico de barras interactivo
- fig = px.bar(df, x='x_column', y='y_column', title='Bar Chart')
- fig.show()

- # Crear un gráfico de dispersión interactivo
- fig = px.scatter(df, x='x_column', y='y_column', color='category_column', title='Scatter Plot')
- fig.show()

- # Crear un gráfico de líneas interactivo
- fig = px.line(df, x='x_column', y='y_column', title='Line Plot')
- fig.show()

Thank you!