IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Joshua Farrell
5/9/24

Outline

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

Executive Summary

METHODOLOGIES:

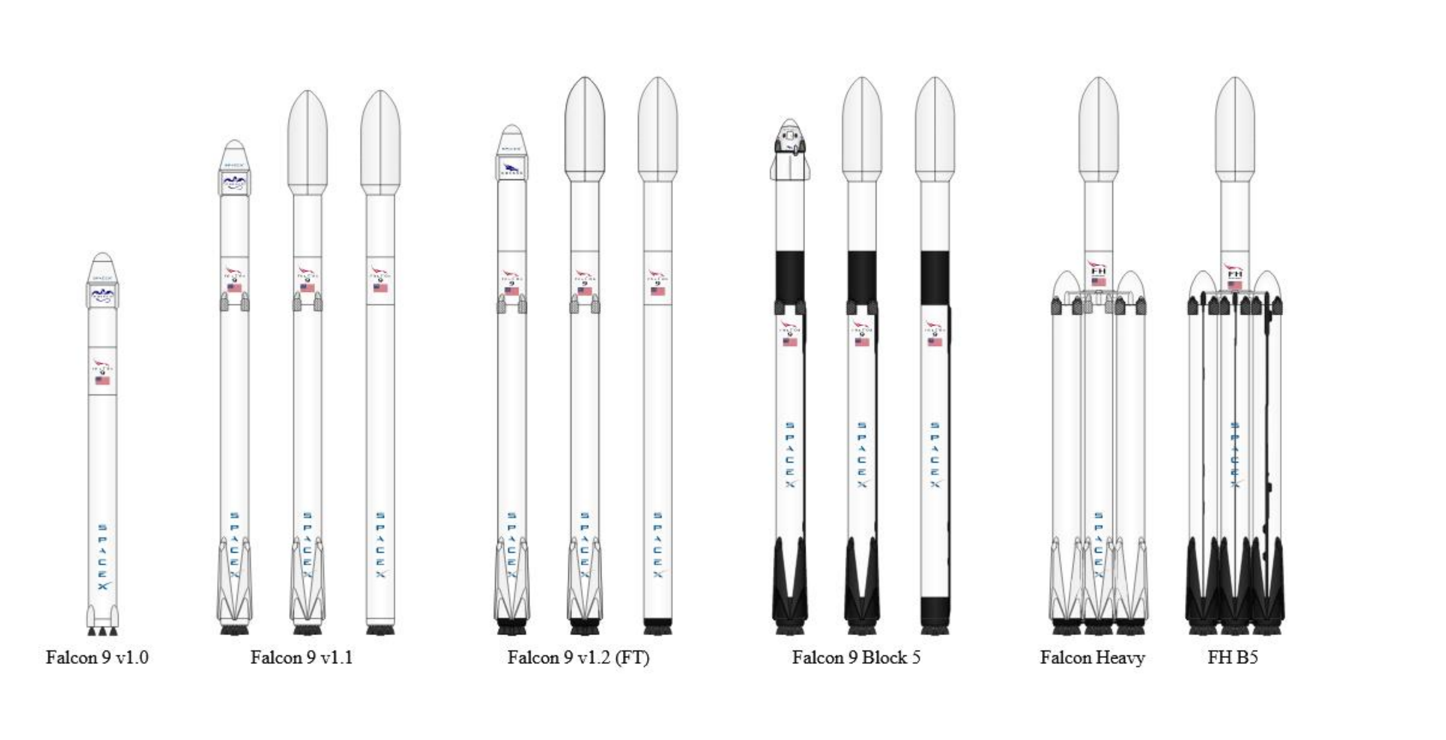
- Develop Python code to manipulate data in a Pandas data frame
- Convert a JSON file into a Create a Python Pandas data frame by converting a JSON file
- Create a Jupyter notebook and make it sharable using GitHub
- Utilize data science methodologies to define and formulate a real-world business problem
- Utilize data analysis tools to load a dataset, clean it, and find out interesting insights from it

RESULTS:

- Falcon9 can land successfully given the right scenarios

Introduction

We will predict if the Falcon 9 first stage will land successfully.



SpaceX 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 SpaceX 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 SpaceX for a rocket launch.

Section 1

Methodology

Methodology

Executive Summary

- Data collection:
  - Use SpaceX REST API Launch data from their website
- Perform data wrangling
  - Scrape data from the site, clean for readability, and process through dozens of formulas to inform insights. Extract, Transfer, Load (ETL)
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build models, test them, and evaluate them using accuracy formulas

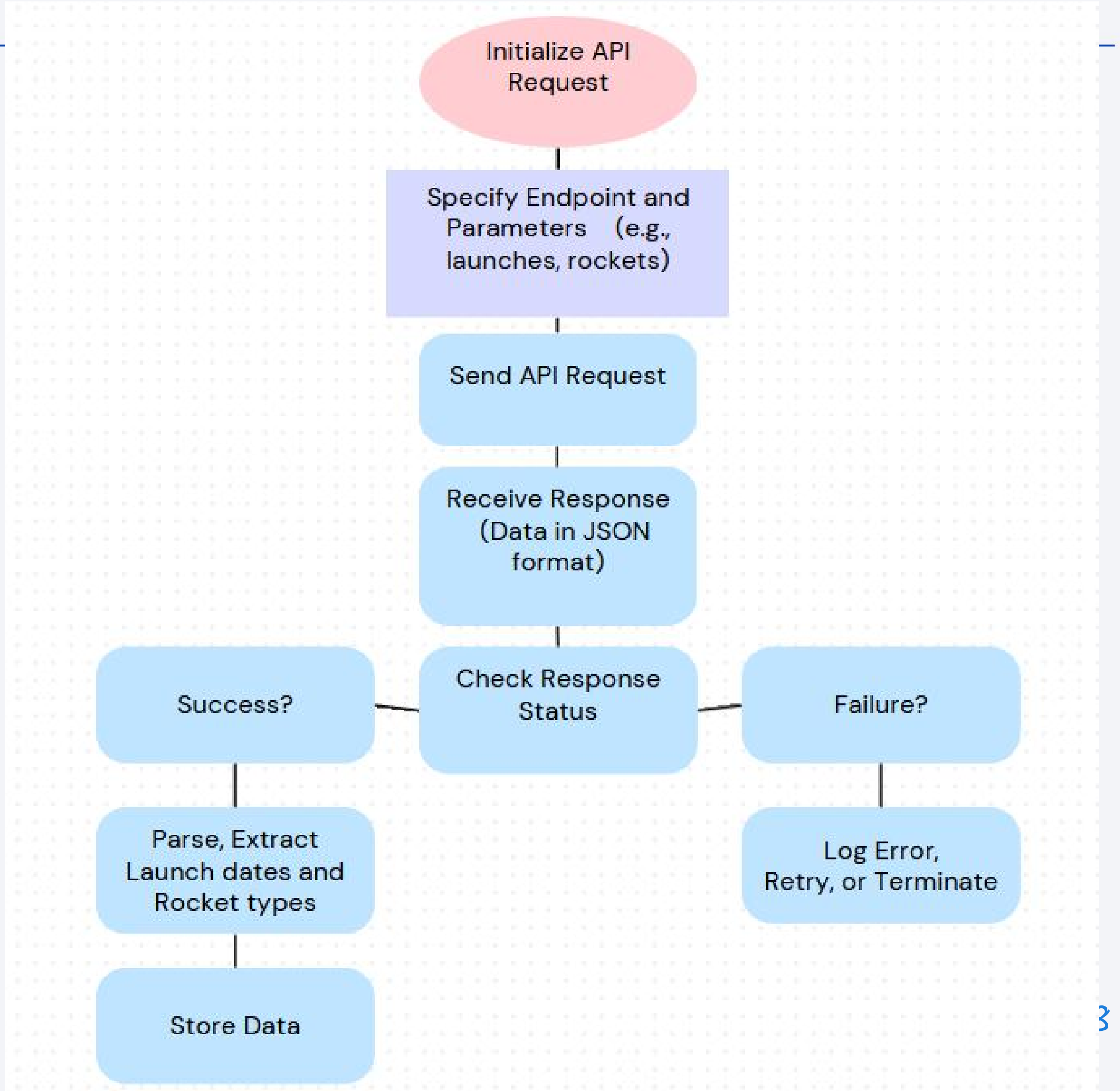
Data Collection

- Collect Data sets by using GET requests in Python to pull the data from the SpaceX site
- Organize into readable data and convert to a dataframe that can be "easily" read
- Drop irrelevant data and manipulate some values to work with data science tools

Data Collection – SpaceX API

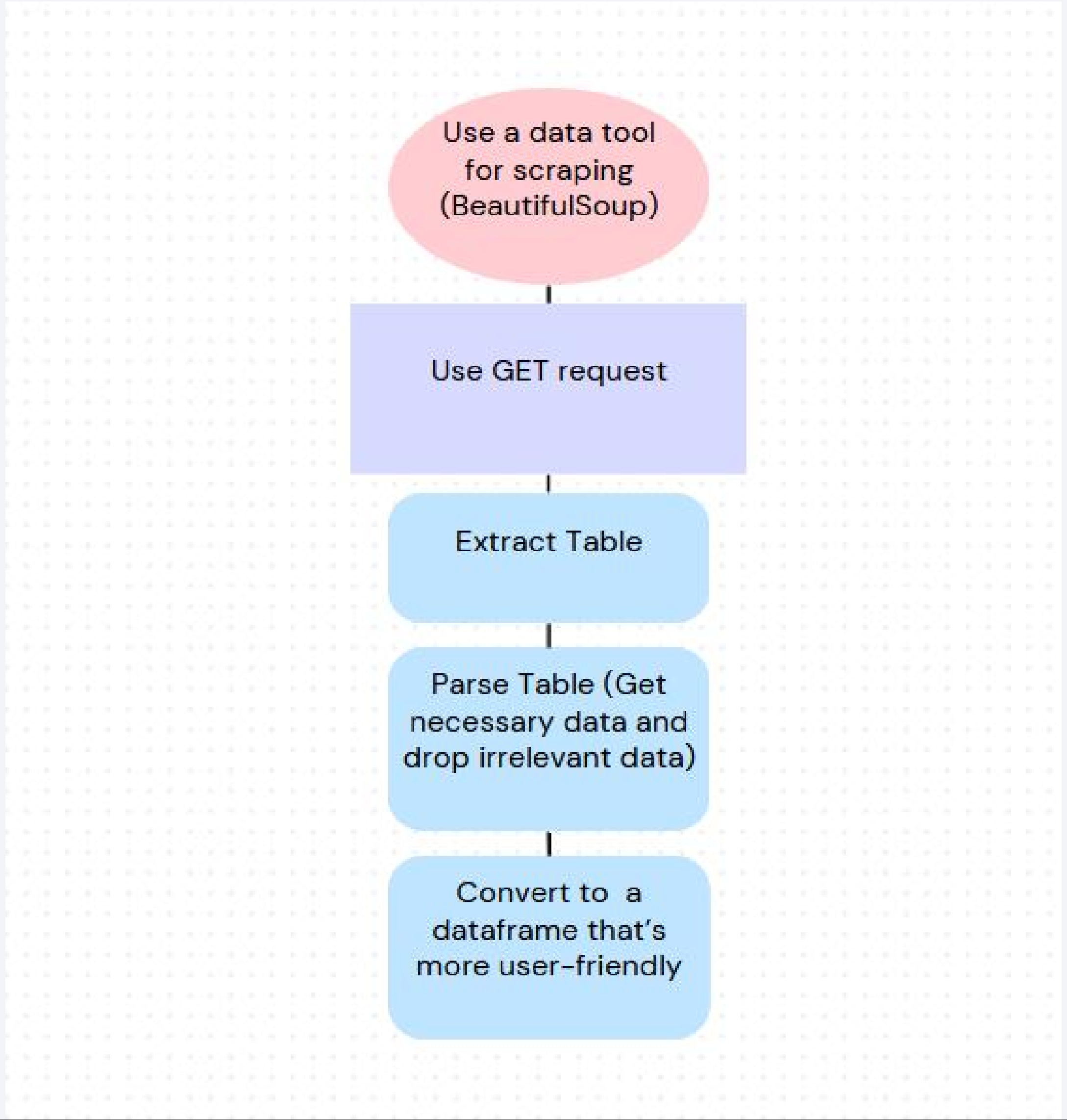
- Calculate the number of launches at each SpaceX site
- Calculate the number of launches in each orbit type (Low-High)
- Create Mission Outcome Labels (Failure or Success)

GitHub URL



Data Collection – Scraping

- Extract Launch records from site using Beautiful Soup
- Parse the resulting table
- Convert into Pandas Dataframe



[GitHub URL]

Data Wrangling

- Convert Data from Null values to mean values and from Categorical entries to Numerical/True/False entries
- Store as CSV

[GitHub URL]

EDA with Data Visualization

- Create scatter plots and bar charts by writing Python code to analyze data in a Pandas data frame
- Write Python code to conduct exploratory data analysis by manipulating data in a Pandas data frame
- Create and execute SQL queries to select and sort data
- Utilize data visualization skills to visualize the data and extract meaningful patterns to guide the modeling process

[GitHub URL]

EDA with SQL

Perform SQL queries:

- %sql sqlite:///my_data1.db
- SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;
- SELECT * FROM SPACEXTABLE

  WHERE "Launch_Site" LIKE 'CCA%'

  LIMIT 5;
- SELECT AVG("PAYLOAD_MASS__KG_") AS AveragePayload

  FROM SPACEXTABLE

  WHERE "Booster_Version" = 'F9 v1.1';

[GitHub URL]

EDA with SQL

Perform SQL queries:

- SELECT MIN("Date") AS FirstSuccessfulLanding
  FROM SPACEXTABLE
  WHERE "Landing_Outcome" = 'Success (ground pad)';
- SELECT "Booster_Version"
  FROM SPACEXTABLE
  WHERE "Landing_Outcome" = 'Success (drone ship)'
  AND "PAYLOAD_MASS__KG_" > 4000
  AND "PAYLOAD_MASS__KG_" < 6000;

[GitHub URL]()

EDA with SQL

Perform SQL queries:

- SELECT "Mission_Outcome",

      COUNT(*) AS OutcomeCount

  FROM Spacextable

  GROUP BY "Mission_Outcome";

  FROM SPACEXTABLE

- SELECT "Booster_Version"

  FROM Spacextable

  WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM Spacextable);

[GitHub URL]

EDA with SQL

Perform SQL queries:

- SELECT SUBSTR("Date", 6, 2) AS Month,

  "Landing_Outcome",

  "Booster_Version",

  "Launch_Site"

  FROM Spacextable

  WHERE SUBSTR("Date", 1, 4) = '2015'

  AND "Landing_Outcome" LIKE '%Failure (drone ship)%';

GitHub URL

EDA with SQL

Perform SQL queries:

```
SELECT "Landing_Outcome",
       COUNT(*) AS OutcomeCount
FROM Spacextable
WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY "Landing_Outcome"
ORDER BY OutcomeCount DESC;
```

[GitHub URL]

Build an Interactive Map with Folium

- Added Markers to Launch Locations
- Added Categorical Info to Markers
- Added Lines between Markers and Important Landmarks
  - These improve map readability and interactivity that help tell the story

GitHub URL

Build a Dashboard with Plotly Dash

- Added dropdown menu and Pie Charts
  - This allows user to toggle between different sites and allows

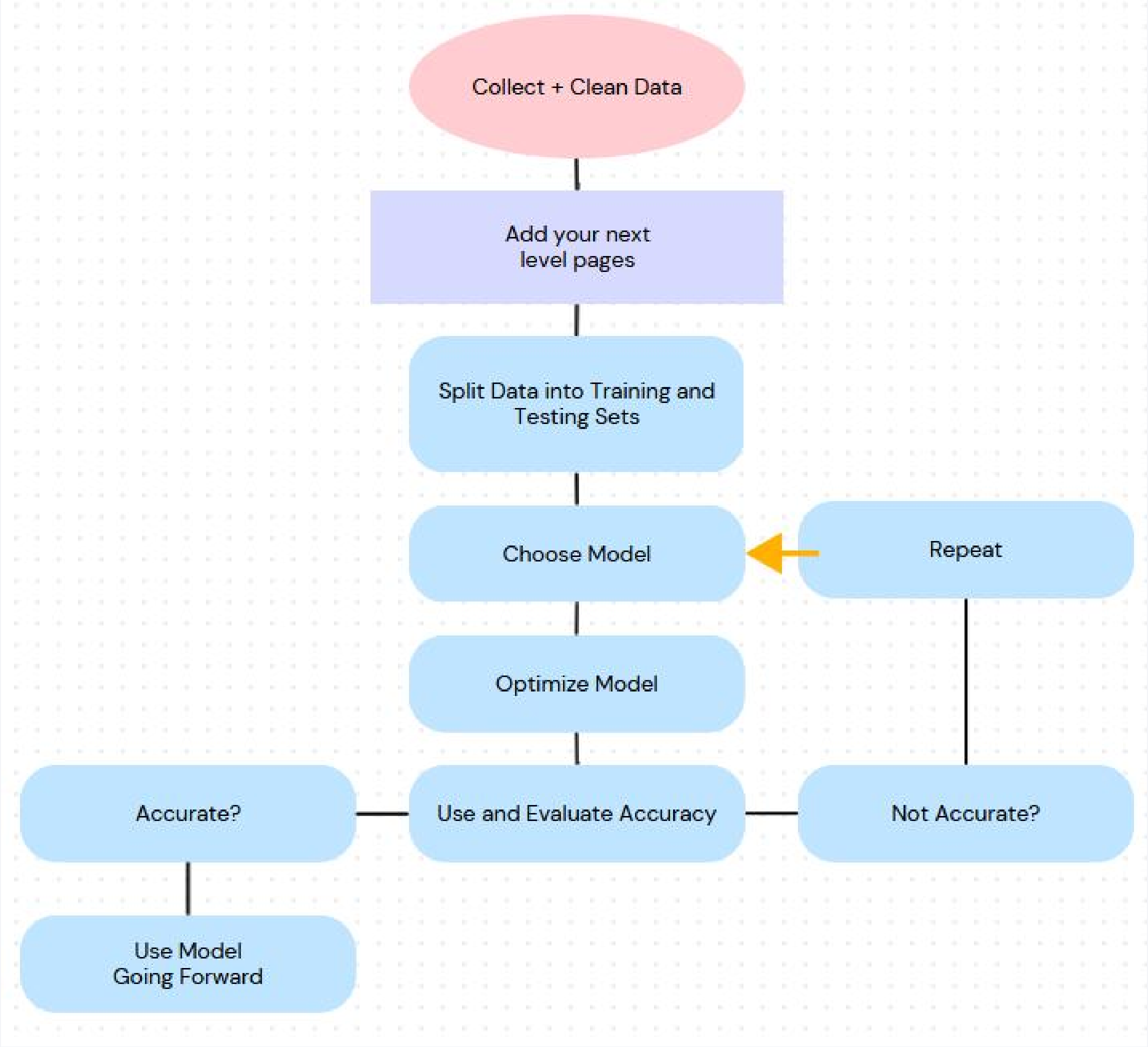
    easy visualization between launch sites

- Added Slider Bar and Scatterplot
  - This allows user to choose different weight parameters and see how Payload Mass and Booster Versions are correlated

[GitHub URL]

Predictive Analysis (Classification)

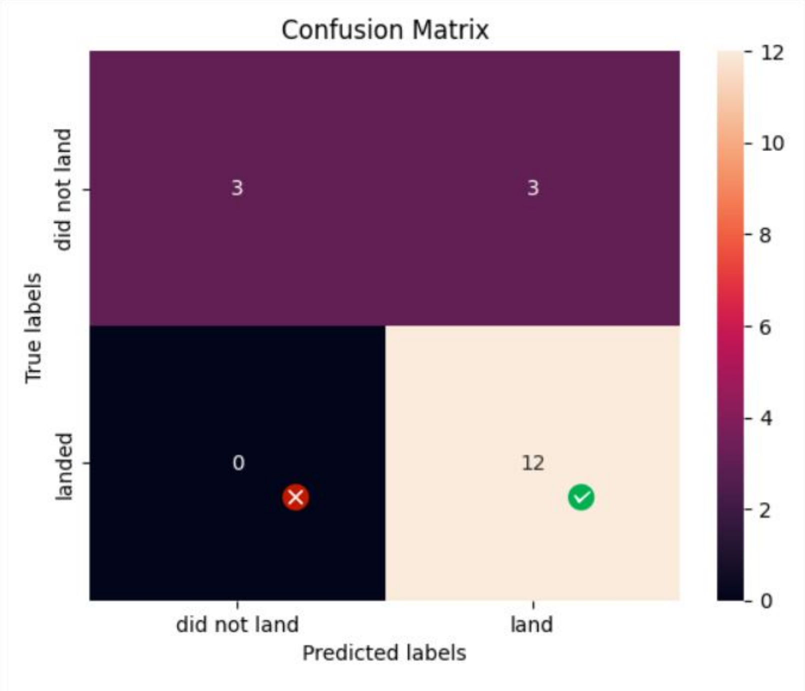
- Perform exploratory Data Analysis and determine Training Labels
    - Create a column for the class
    - Standardize the data using Numpy
    - Split into training data and test data
- Find the method performs best using test data



[GitHub URL]

Results

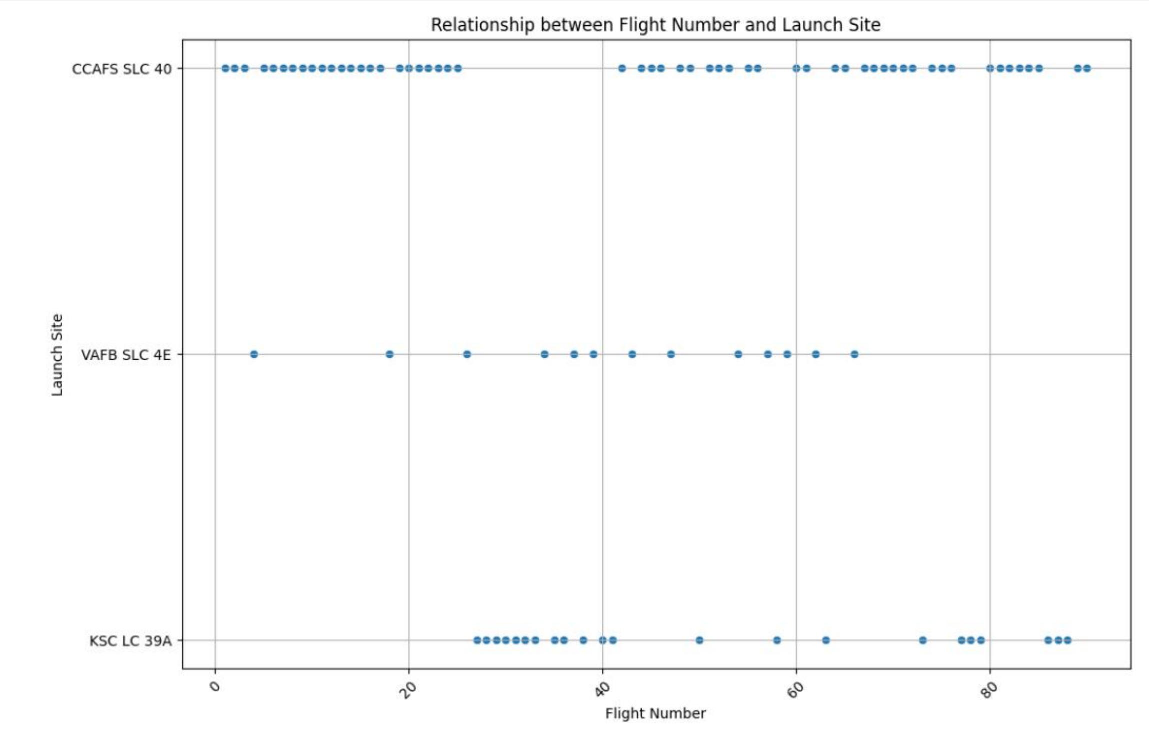
- Exploratory data analysis results
  - K-Nearest Neighbor is the most accurate model to use for predicting landings
- K-Nearest Neighbor was more accurate than Decision Tree, SVM, and Logistic Regression models with an accuracy of 85%. It correctly predicted 12 of 12 landings after testing and training the dataset

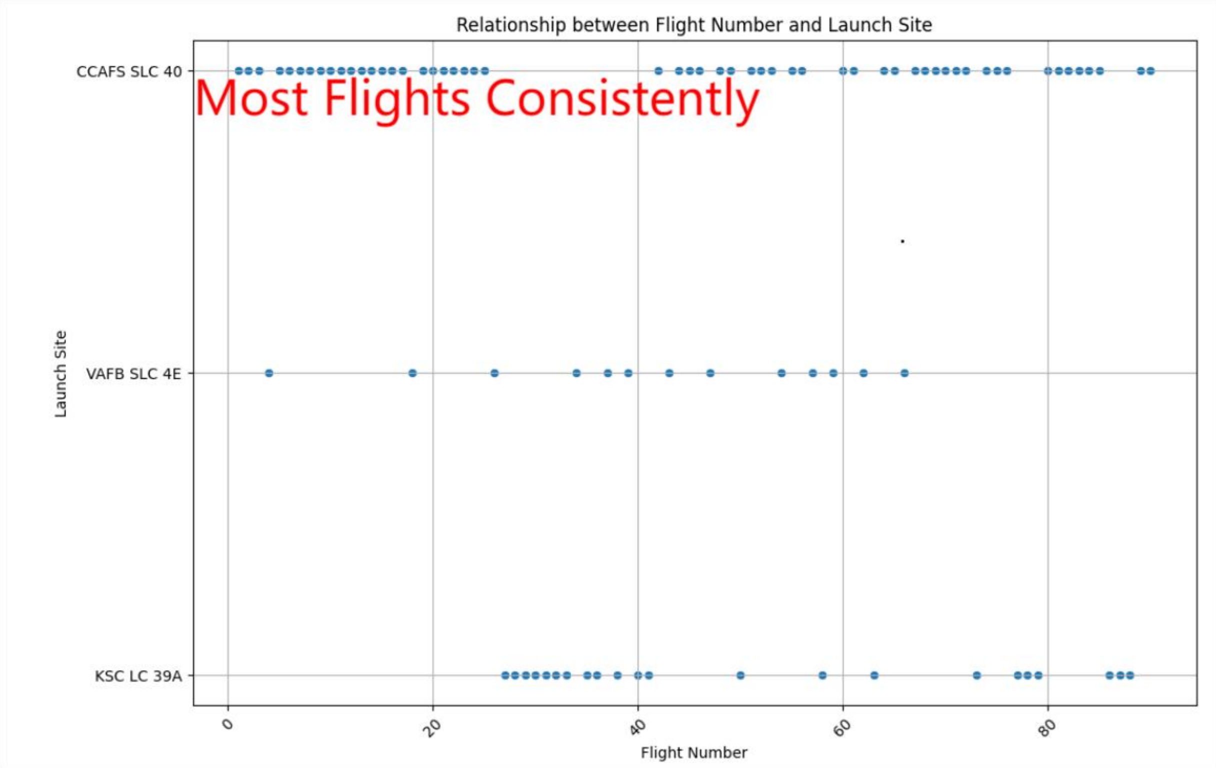


Section 2

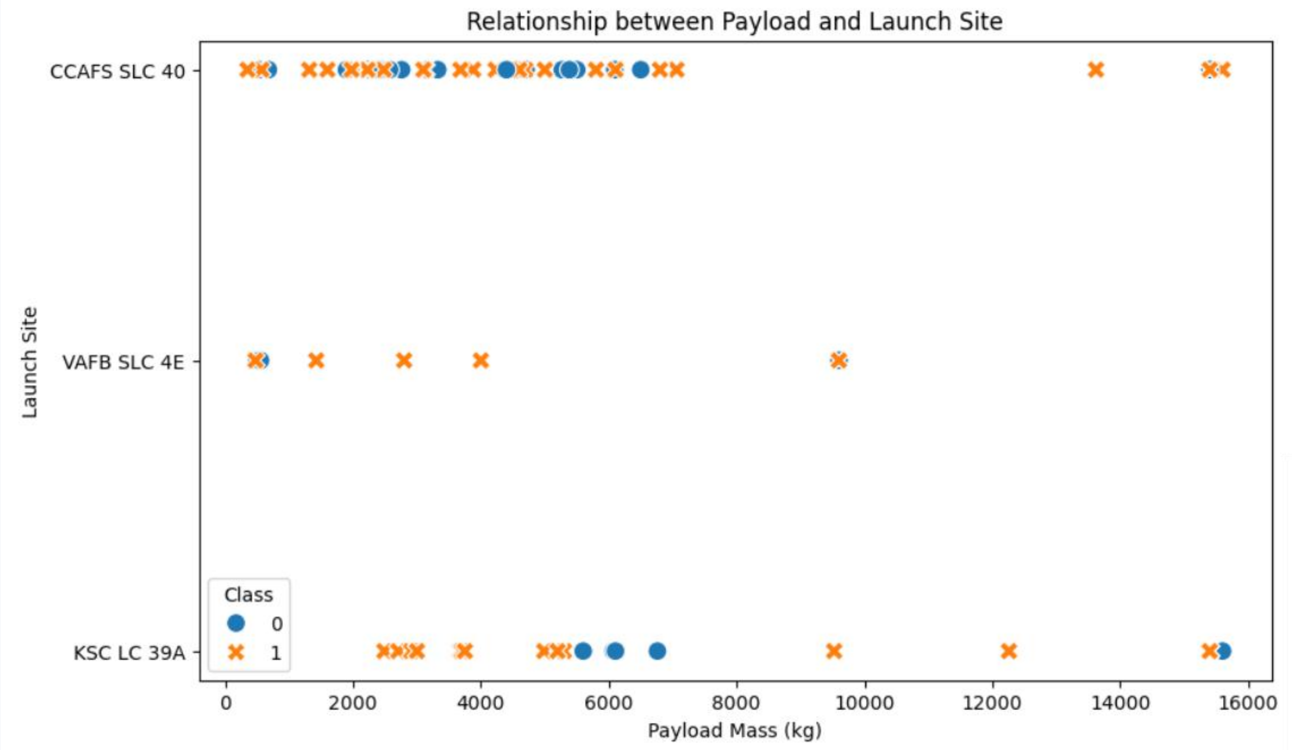
Insights drawn from EDA

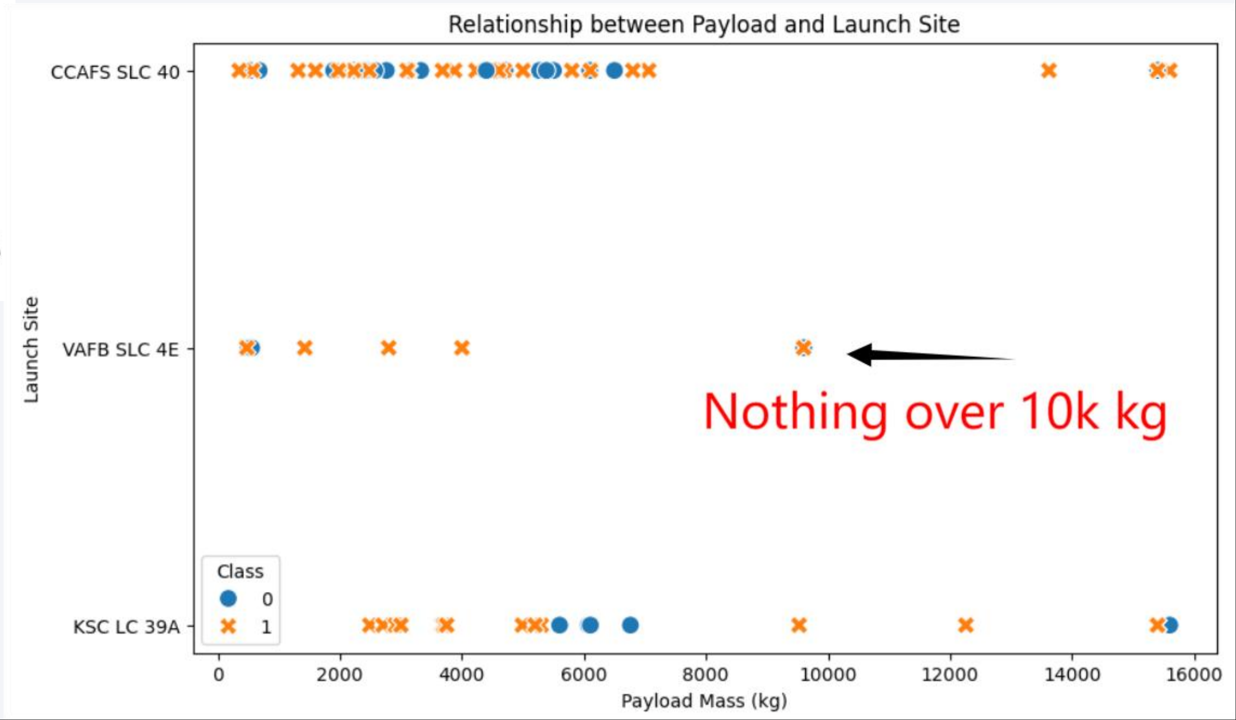
Flight Number vs. Launch Site



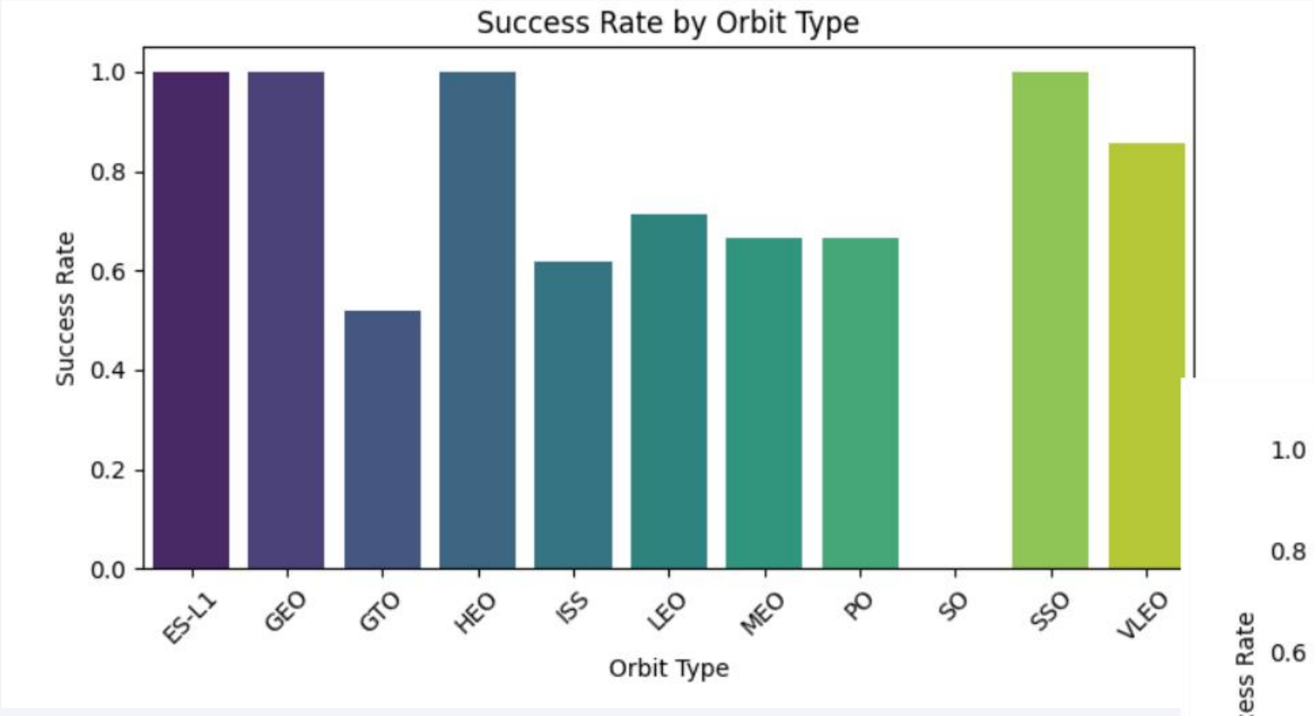


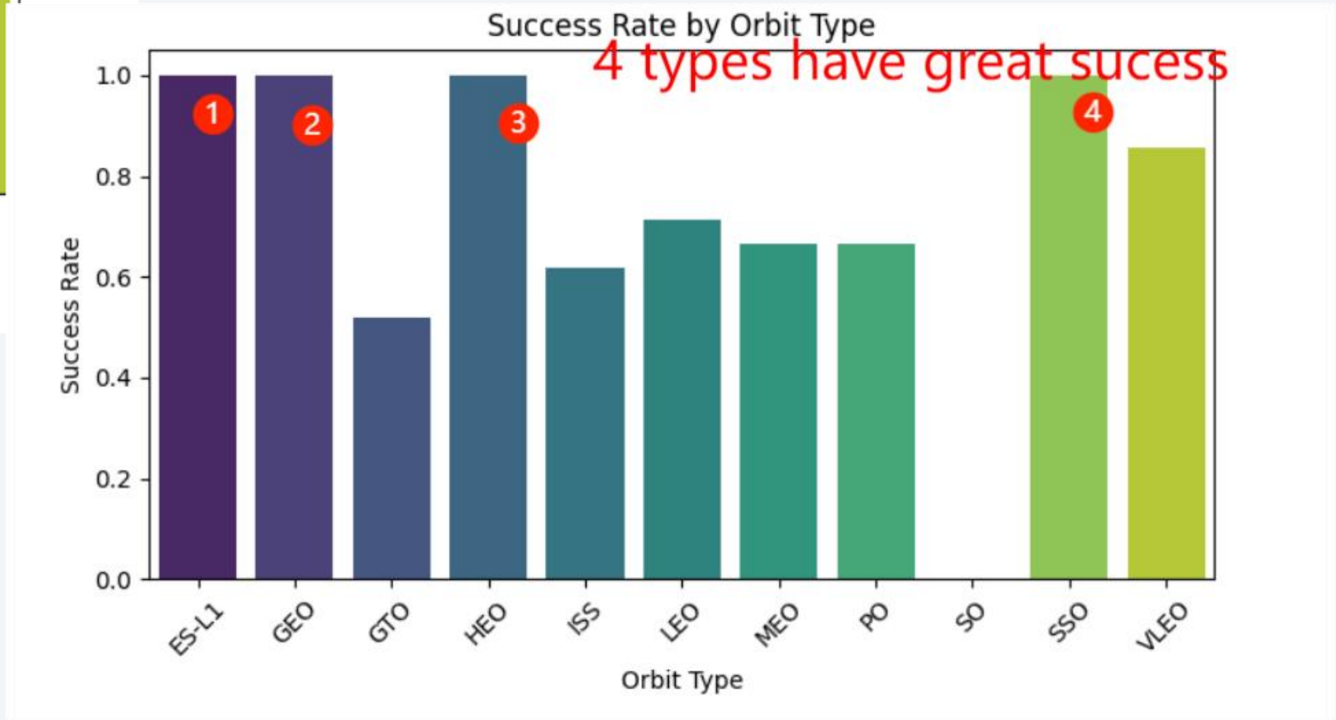
Payload vs. Launch Site



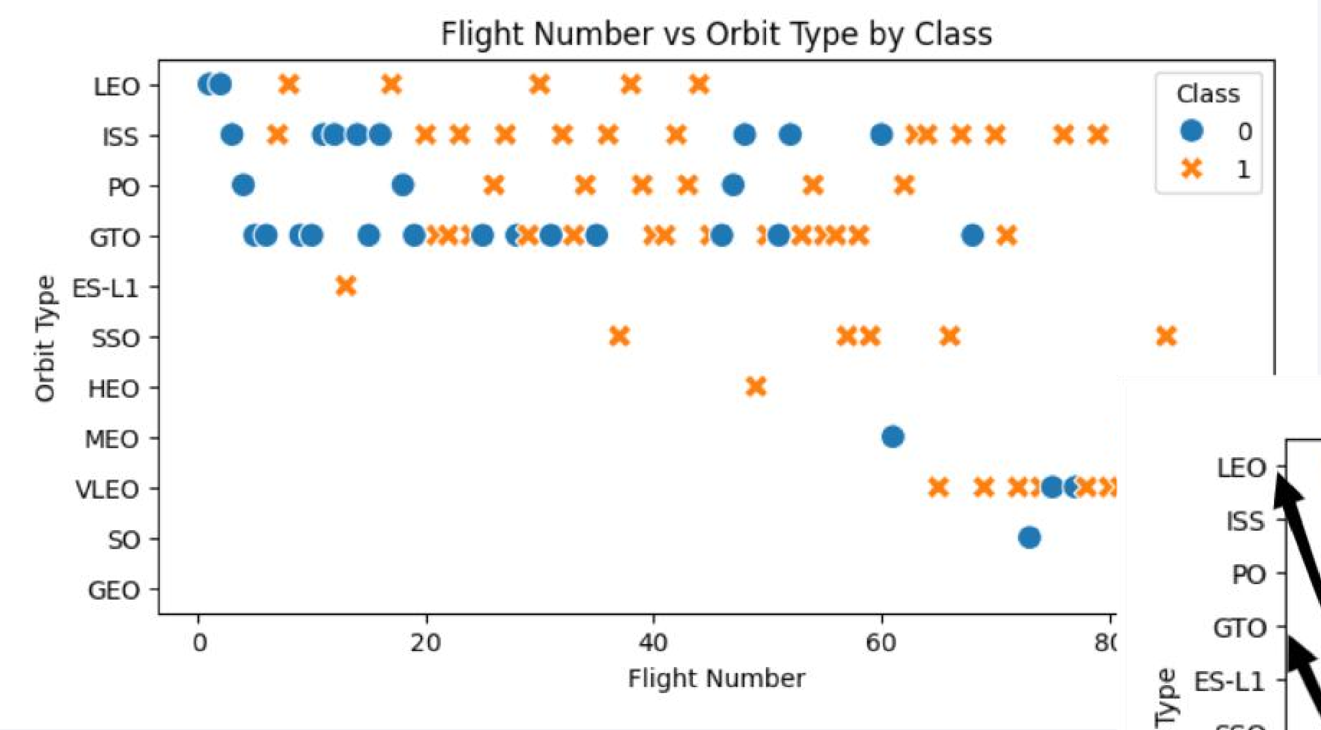


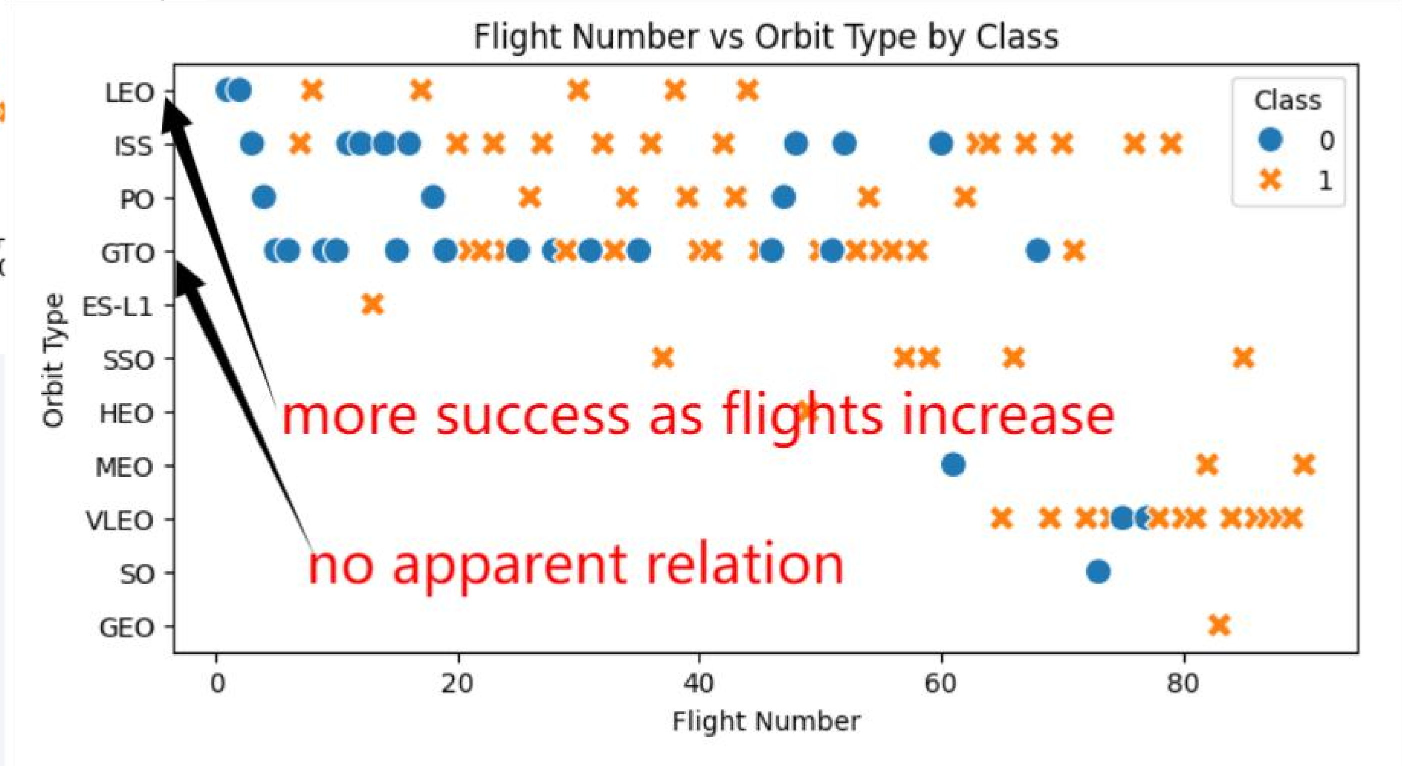
Success Rate vs. Orbit Type



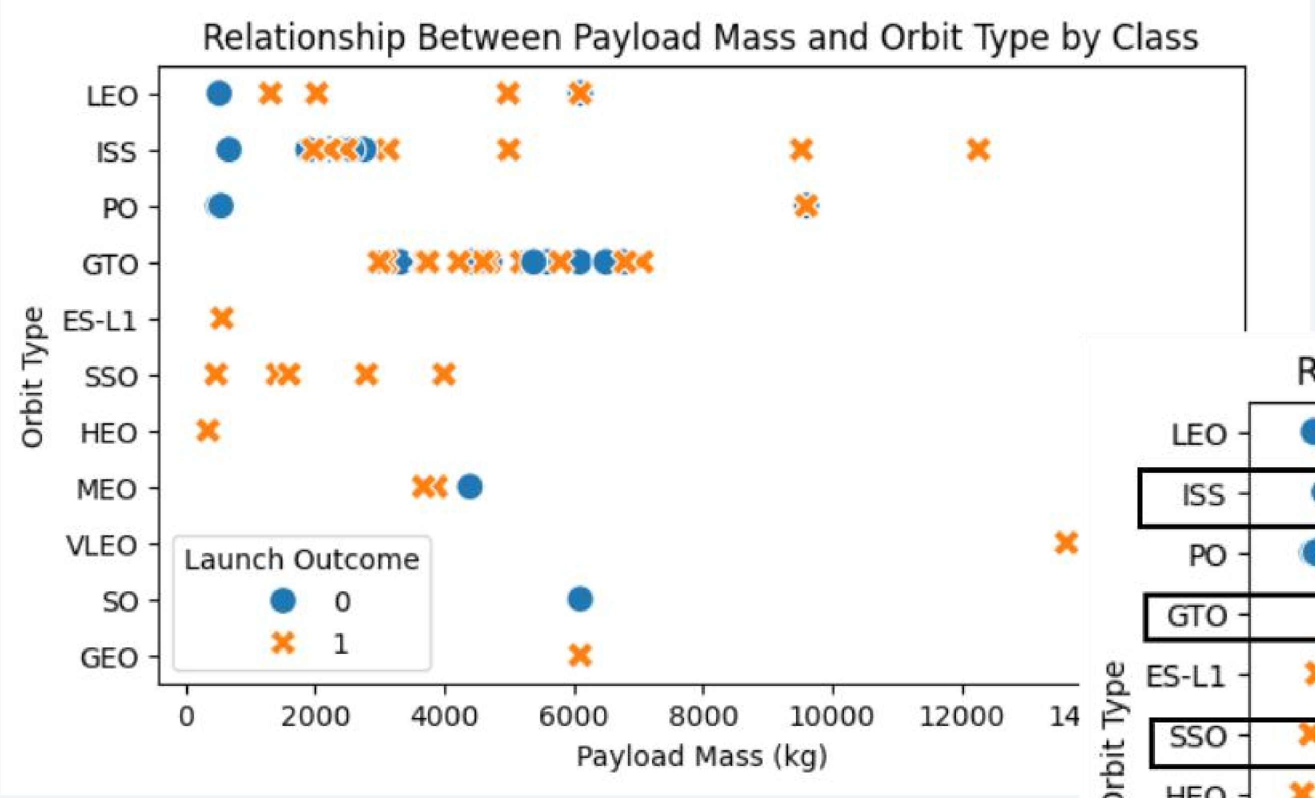


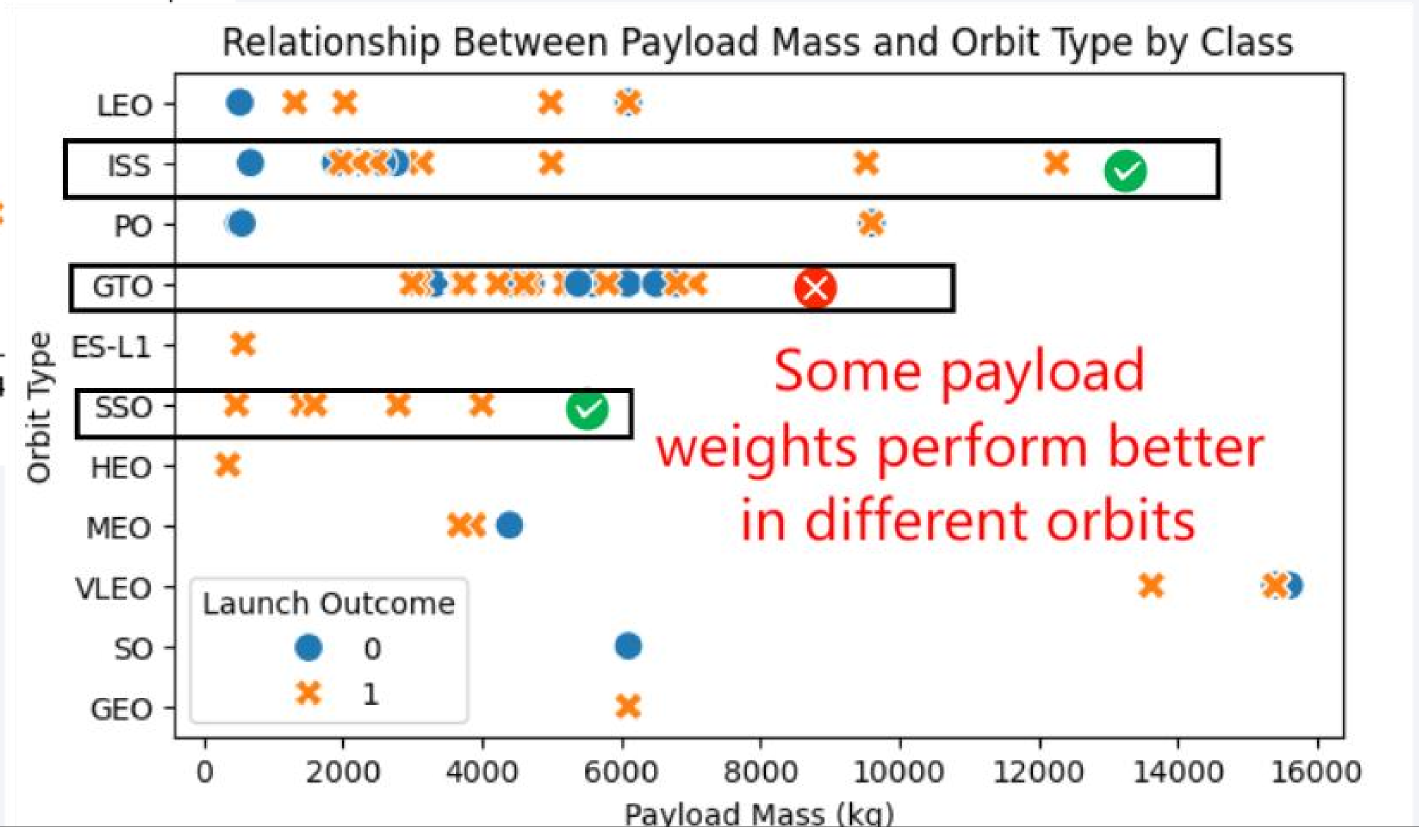
Flight Number vs. Orbit Type



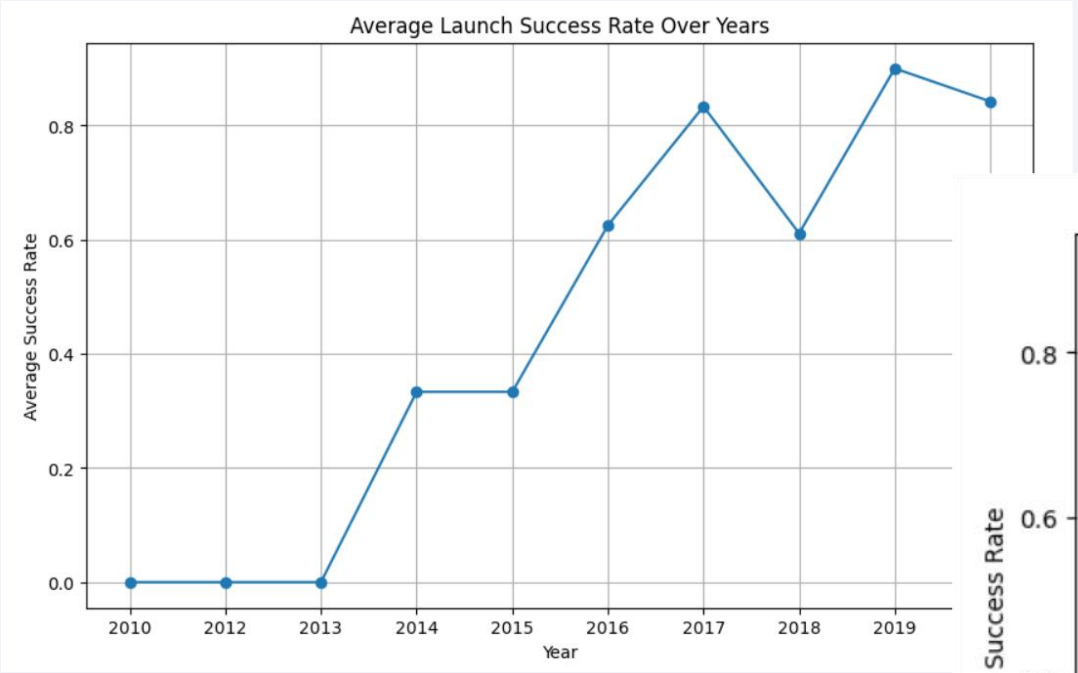


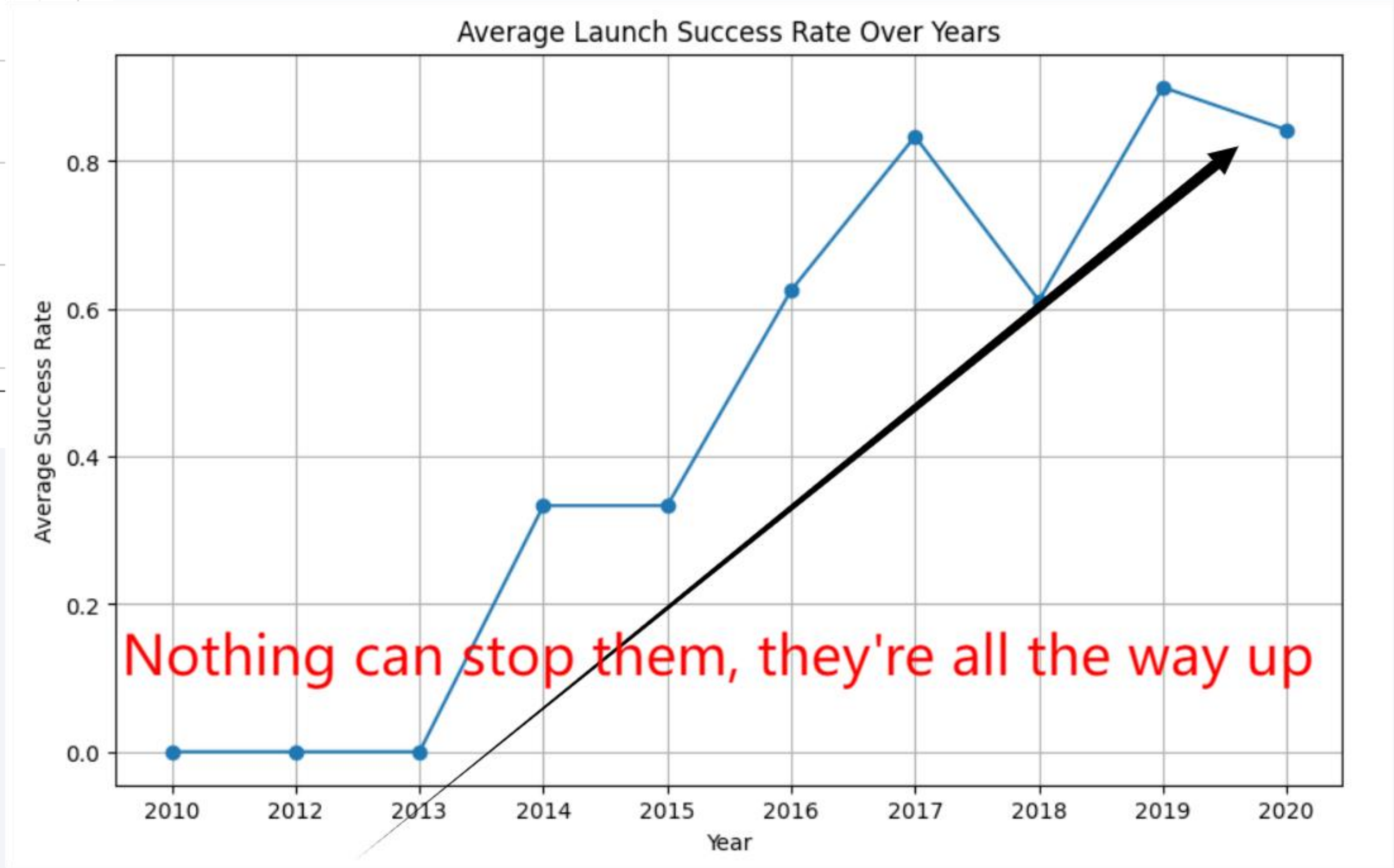
Payload vs. Orbit Type





Launch Success Yearly Trend





All Launch Site Names

Task 1

Display the names of the unique launch sites in the space mission

```
%%sql
SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE;
```

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

This query organizes these results

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

Display 5 records where launch sites begin with the string 'CCA'

In [60]:
```
%%sql
SELECT * FROM SPACEXTABLE
WHERE "Launch_Site" LIKE 'CCA%'
LIMIT 5;
```

This query finds and displays only the CCA Site data

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

Out[60]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	La
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Fa
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	Fa
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

Display the total payload mass carried by boosters launched by NASA (CRS)

In [68]:
```
%%sql
SELECT SUM("PAYLOAD_MASS__KG_") AS TotalPayload
FROM SPACEXTABLE
WHERE Customer LIKE '%NASA (CRS)%';
```

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

Out[68]:

TotalPayload
48213

Calculates Sum of a Column and displays the result

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

Display average payload mass carried by booster version F9 v1.1

In [70]:
```
%%sql
SELECT AVG("PAYLOAD_MASS__KG_") AS AveragePayload
FROM SPACEXTABLE
WHERE "Booster_Version" = 'F9 v1.1';
```

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

Out[70]:

AveragePayload
2928.4

Finds Average Payload of F9 Falcon Rocket and displays result

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

```
In [71]:
%%sql
SELECT MIN("Date") AS FirstSuccessfulLanding
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (ground pad)';
```

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

Out[71]:

FirstSuccessfulLanding
2015-12-22

Finds the date of the first successful landing of a rocket landing on a ground pad and displays the result

Successful Drone Ship Landing with Payload between 4000 and 6000

In [72]:

```sql
%%sql
SELECT "Booster_Version"
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (drone ship)'
  AND "PAYLOAD_MASS__KG_" > 4000
  AND "PAYLOAD_MASS__KG_" < 6000;
```

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

Out[72]:

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

Shows Booster Versions with 4-6k kg Payloads that landed successfully on a drone ship

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

In [73]:
```
%%sql
SELECT "Mission_Outcome",
       COUNT(*) AS OutcomeCount
FROM Spacextable
GROUP BY "Mission_Outcome";
```

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

Organizes Success Counts

Out[73]:

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

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

In [76]:

```
%%sql
SELECT "Booster_Version"
FROM Spacextable
WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM Spacextable);
```

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

Out[76]:

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

Finds heaviest Booster Payloads and displays them

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

In [77]:

```
%%sql
SELECT SUBSTR("Date", 6, 2) AS Month,
       "Landing_Outcome",
       "Booster_Version",
       "Launch_Site"
FROM Spacextable
WHERE SUBSTR("Date", 1, 4) = '2015'
  AND "Landing_Outcome" LIKE '%Failure (drone ship)%' ;
```

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

Out[77]:

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

There were drone ship failuresin January and April by two different versions at the same site

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

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

In [79]:

```
%%sql
SELECT "Landing_Outcome",
       COUNT(*) AS OutcomeCount
FROM Spacextable
WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY "Landing_Outcome"
ORDER BY OutcomeCount DESC;
```

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

Out[79]:

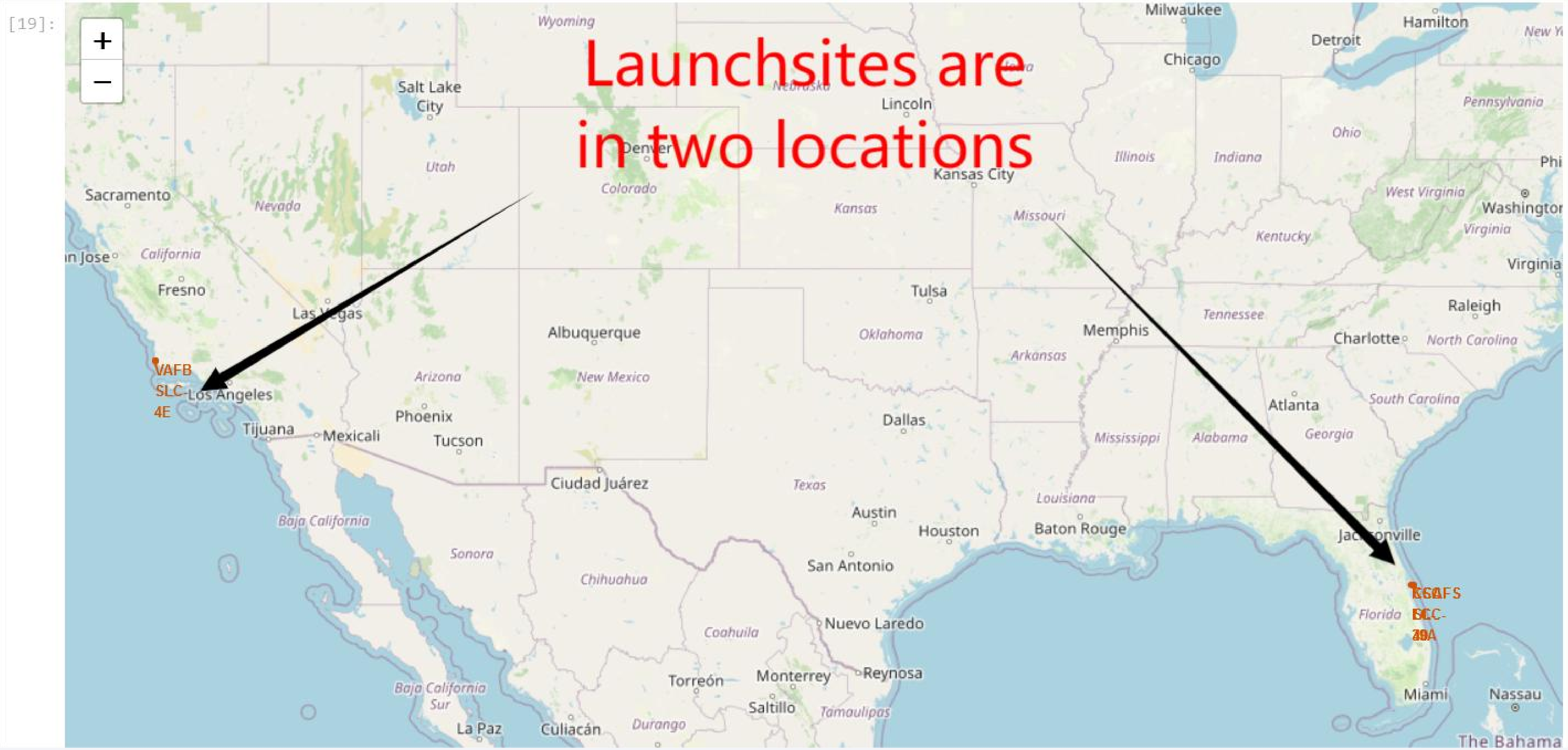
Landing_Outcome	OutcomeCount
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Things were pretty 50/50 regarding success/failure from 2010-17

Section 3

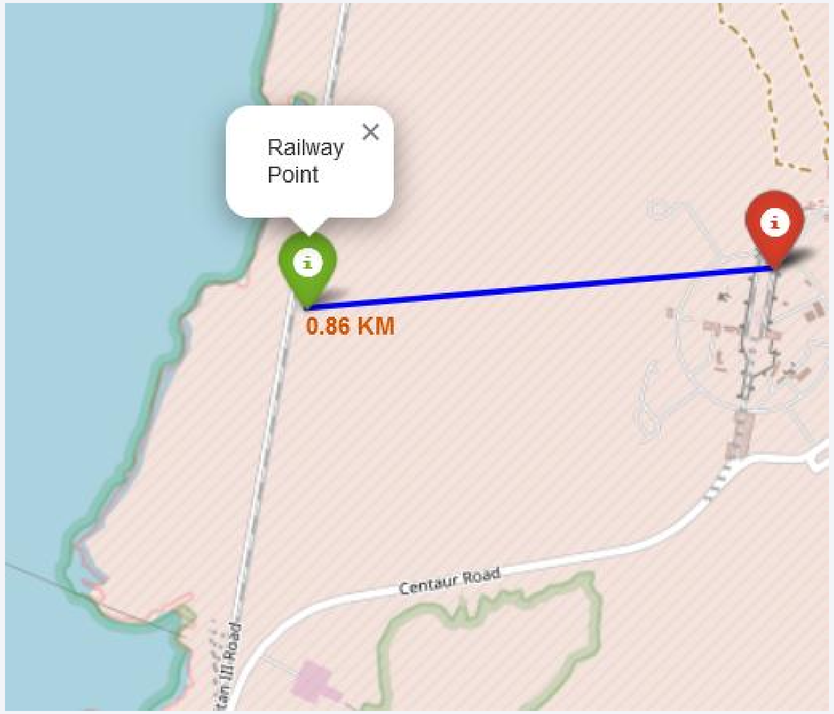
Launch Sites Proximities Analysis

Launch Sites

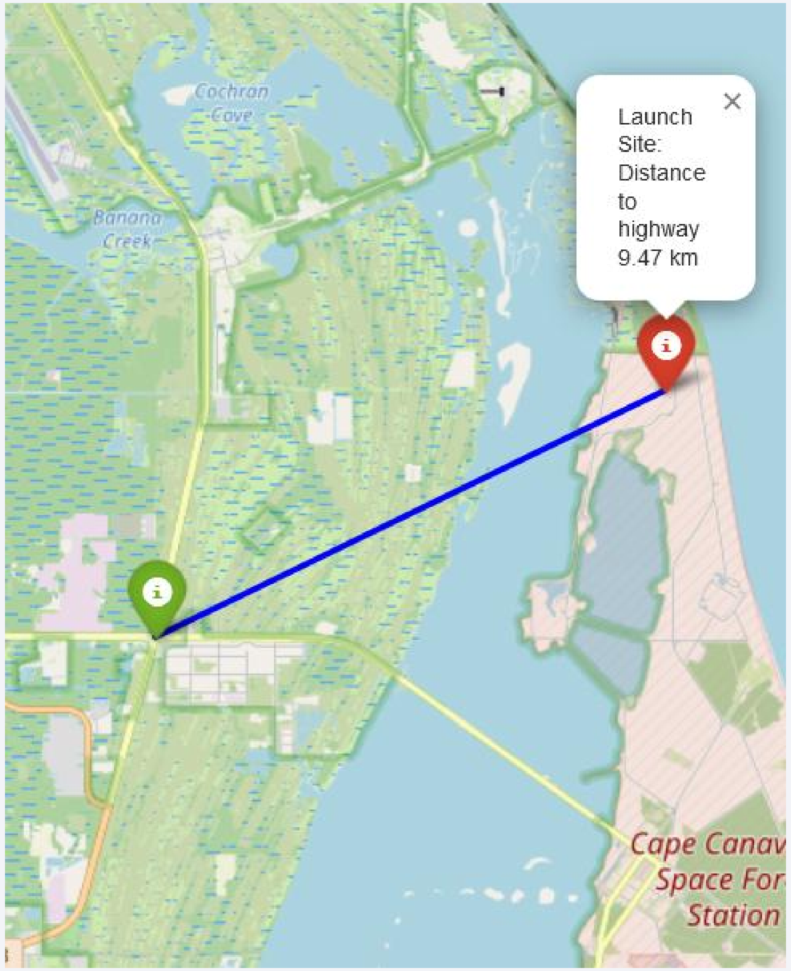


Launch Markers and Success Rates

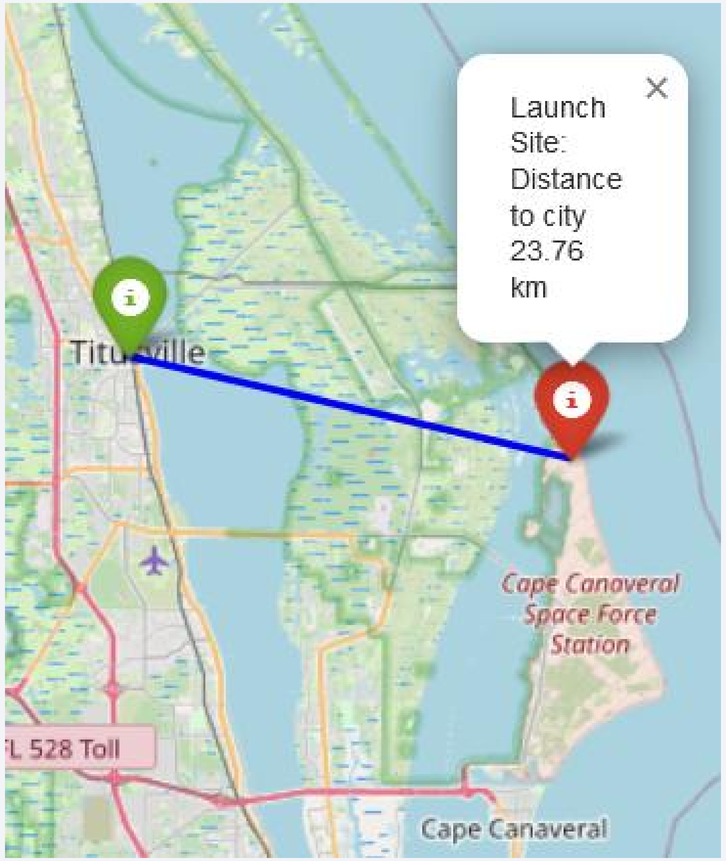
Launch Proximities to Rail, Coastline, Highway, and City



Very close to a Railway and the Coastline



Nearest Highway is relatively far

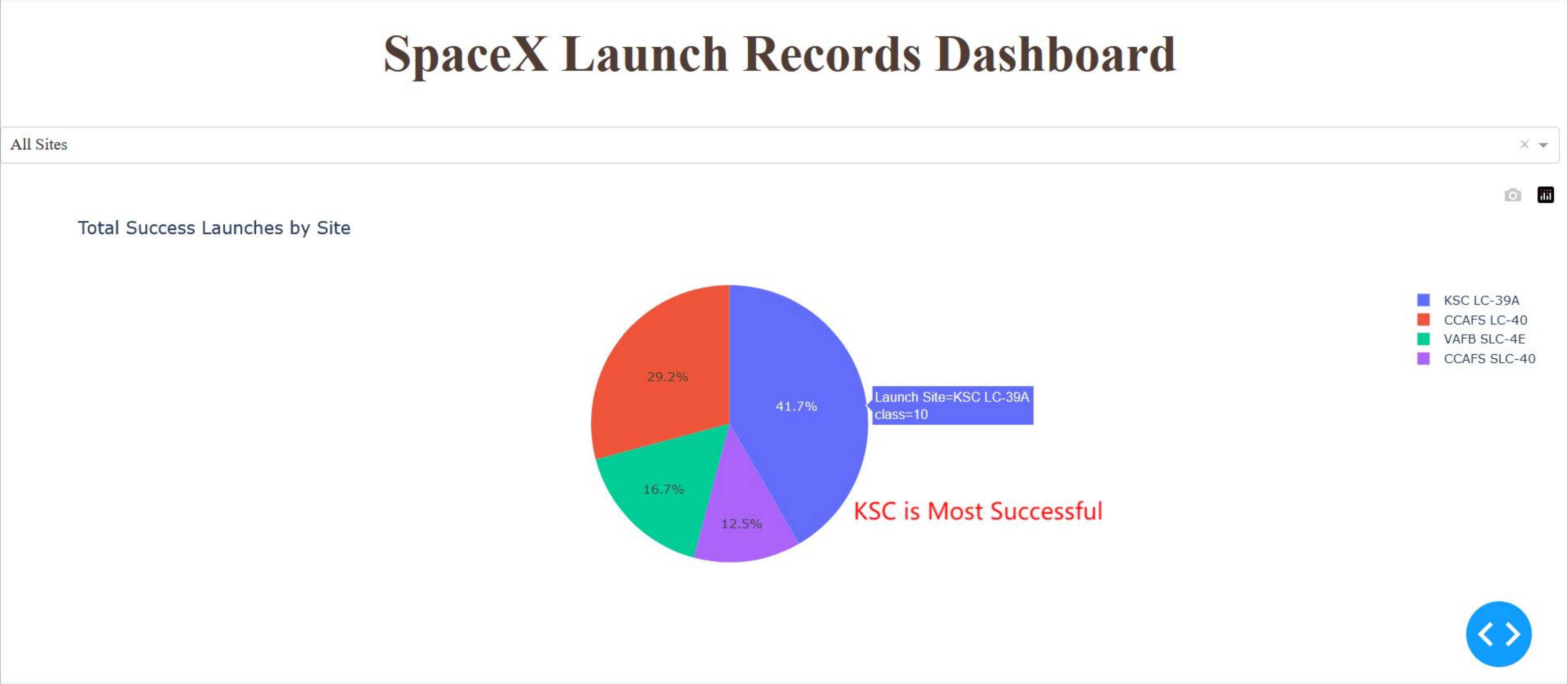


Pretty far from nearest City

Section 4

Build a Dashboard with Plotly Dash

Making a Real-Time Data Dashboard

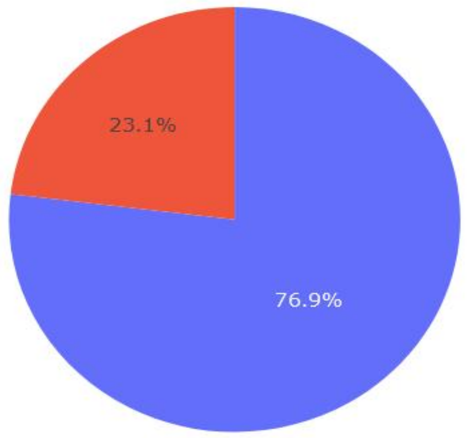


KSC LC-39A Launch Site

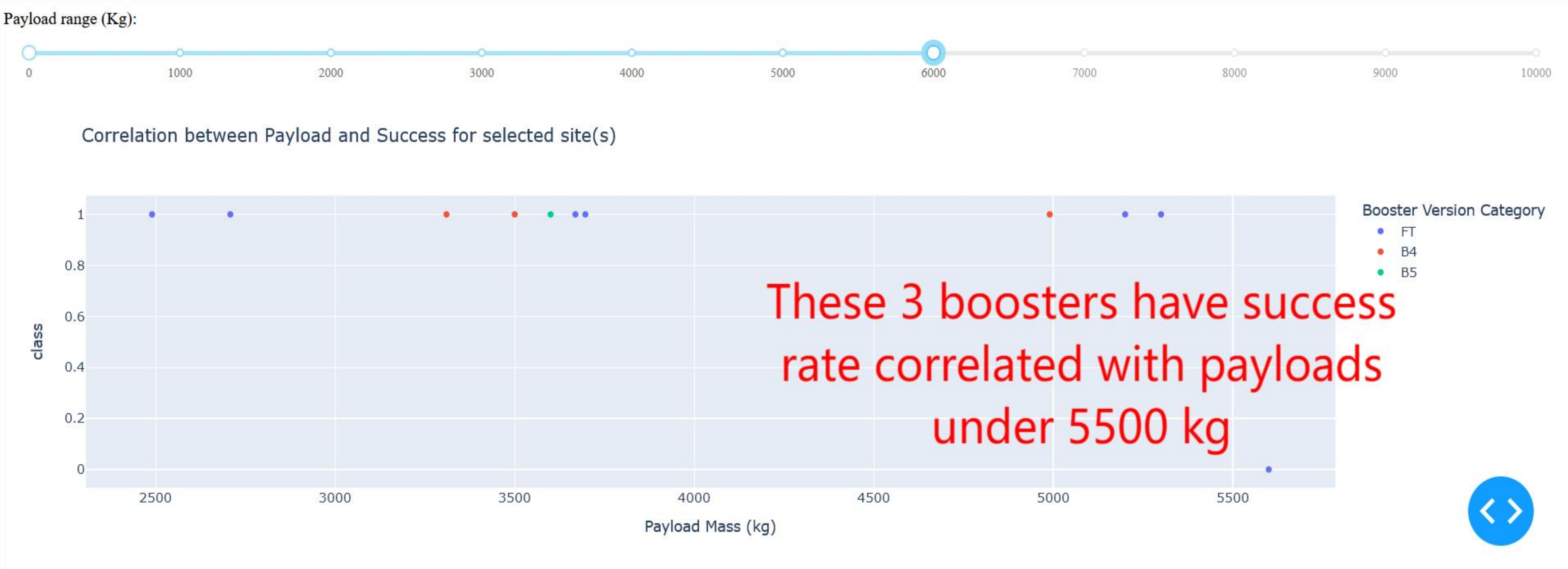
SpaceX Launch Records Dashboard

KSC LC-39A

Total Success and Fail Launches for KSC LC-39A



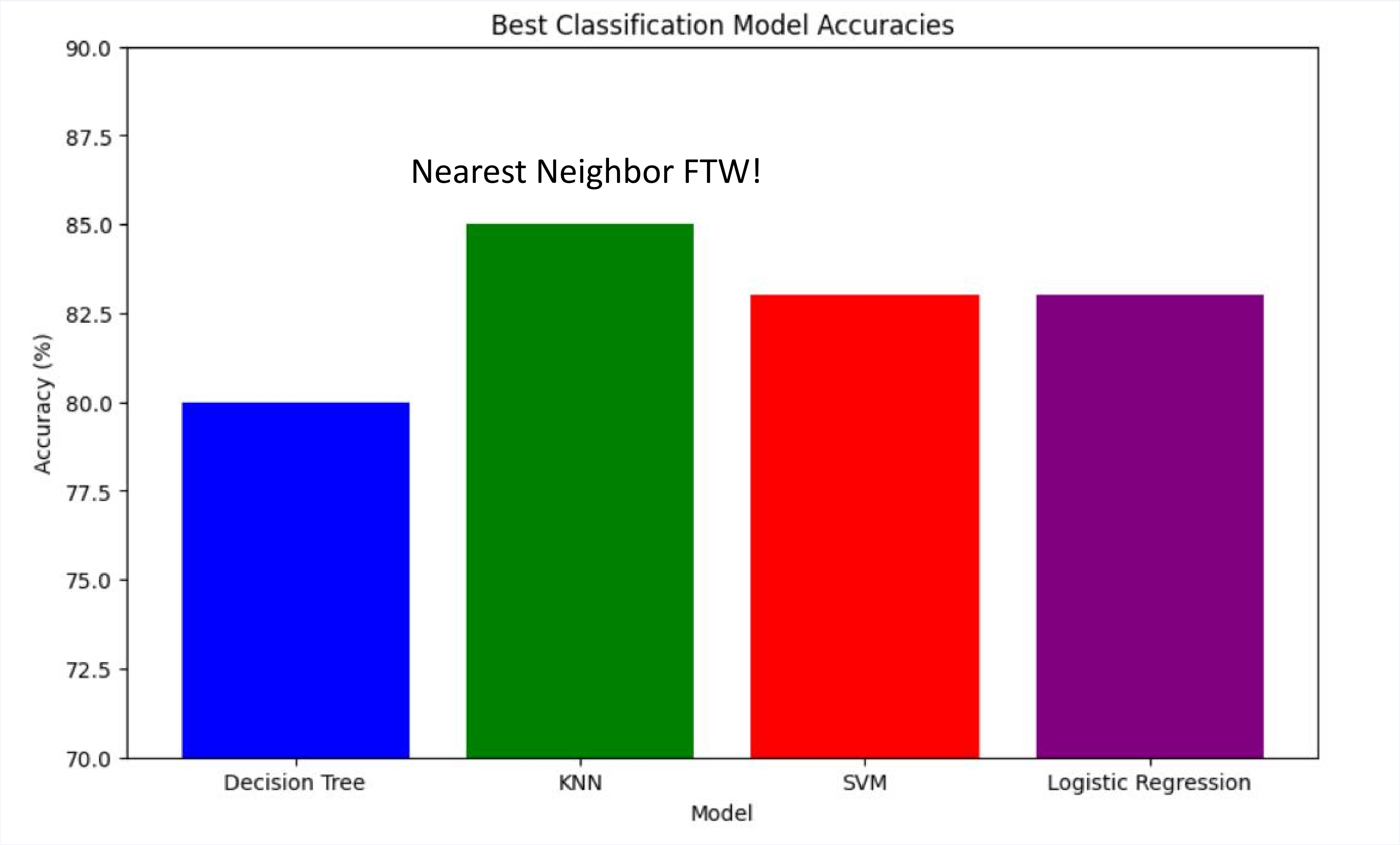
Slider Bar to Toggle Between Payload Sizes



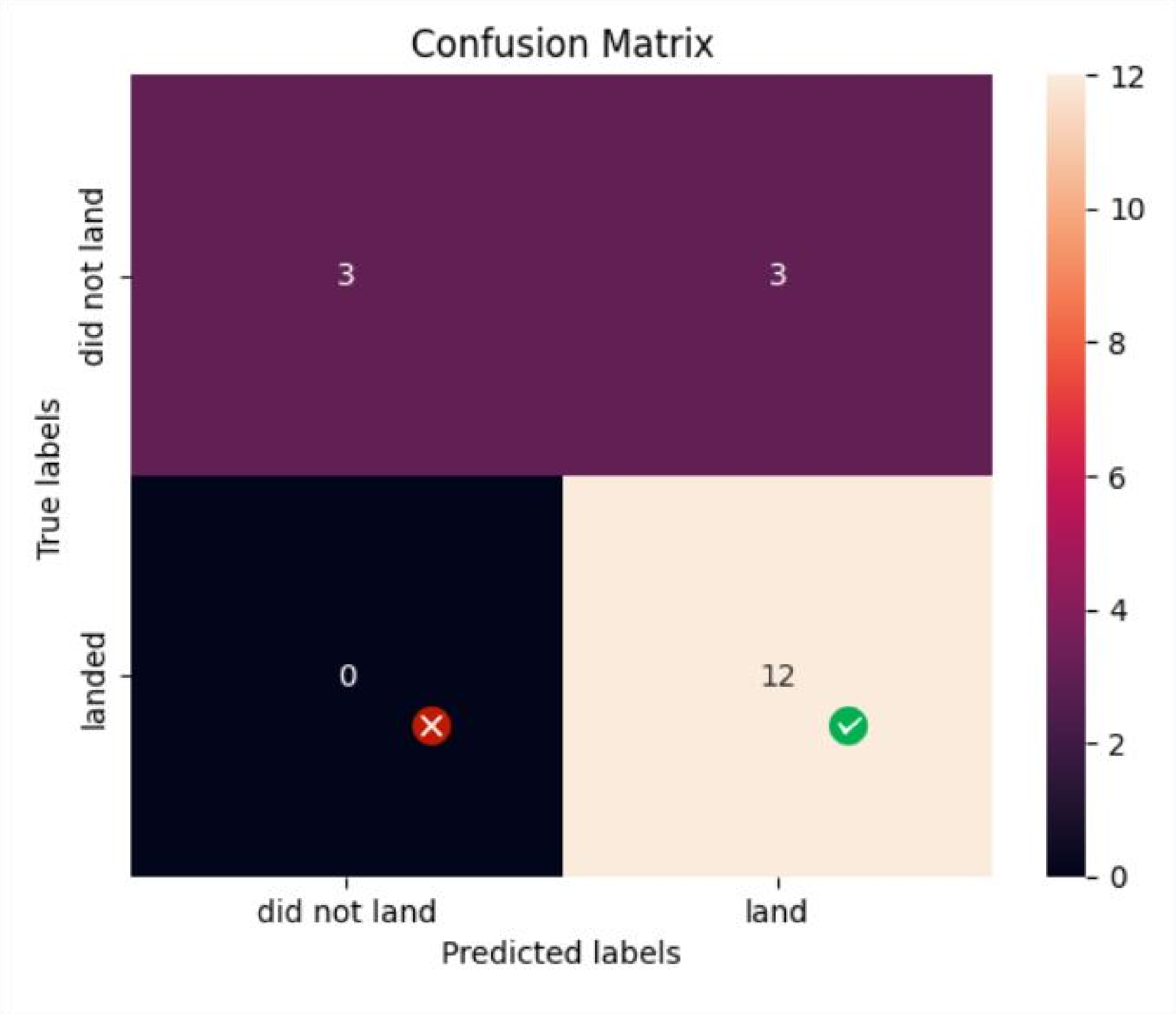
Section 5

Predictive Analysis (Classification)

Classification Accuracy



Confusion Matrix of Nearest Neighbor Method



- The model was great at predicting landings (12/15) but didn't predict any non-landings (0/3)

Conclusions

- For best Falcon 9 success:
  - Use one of these 4 orbits with best success rates:
    - 1ES-L1, GEO, HEO, SSO
  - K-Nearest Neighbor Model should be used for predicting outcomes with an 85% accuracy
  - Average Payload is 3000kg
  - Use site KSC LC-39 for best success rate
  - Use Booster FT for best success rate

Appendix

[GitHub Account]

[IBM Data Science Professional Certificate Capstone]


Thank you!
No birds were harmed in the making of this presentaion (that I’m aware of)