IBM Developer
SKILLS NETWORK

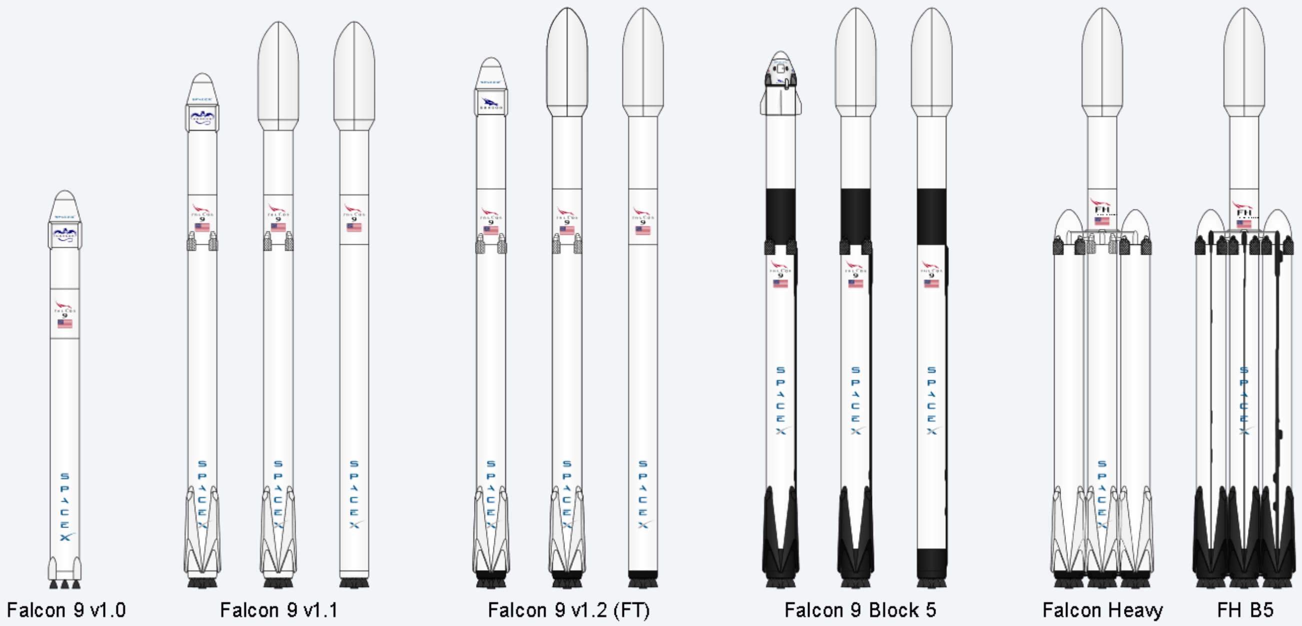
Winning Space Race with Data Science

Adel M. Al-Alawiyat

August 18, 2024

Outline

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



Executive Summary

- Summary of methodologies
  - SpaceX Data Collection using SpaceX API
  - SpaceX Data Collection with Web Scraping
  - SpaceX Data Wrangling
  - SpaceX Exploratory Data Analysis using Visualization and SQL
  - Space-X EDA DataViz Using Python Pandas and Matplotlib
  - Space-X Launch Sites Analysis with Folium-Interactive Visual Analytics and PlotyDash
  - SpaceX Machine Learning Landing Prediction
- Summary of all results
  - EDA results
  - Interactive Visual Analytics and Dashboards
  - Predictive Analysis(Classification)

Introduction

- Project background and context

  SpaceX advertises Falcon 9 rocket launches 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. If we can predict if the first stage landing is successful, then 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.

- Problems you want to find answers

  ✓ predict if the Falcon 9 first stage will land successfully using data from Falcon 9 rocket launches advertised on its website.

  ✓ Determine factors contributing to the rocket landing successfully

  ✓ Features that determine the success rate of a successful landing.

  ✓ Operating conditions to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected from SpaceX REST API and using web scraping wiki pages (launch tables)
- Perform data wrangling
  - The data collected as JSON object from HTML tables, then data was converted into Panda dataframe for visualization and analysis
  - One-hot encoding was applied to categorical features
- 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
  - Using machine learning model to determine if the first stage of Falcone 9 will land successfully

Data Collection

- Data was collected using SpaceX API (a RESTful API) by making a get request to the SpaceX API.
- Used a series of 5 helper functions to process web scraped HTML table to extract information using identification numbers in the launch data (API) and then requesting rocket launch data from the SpaceX API URL.
- The SpaceX launch data was requested and parsed using the GET request, decoded the response content as a JSON result to make it more consistent, then JSON was converted into a Pandas dataframe.
- Performed web scraping to collect Falcon 9 historical launches from a Wiki page stored in a HTML tables.
- Extracted the Falcon 9 launch HTML table records from the Wiki page using BeautifulSoup and request Libraries;
- parsed the table and converted it into a Pandas dataframe.

Data Collection

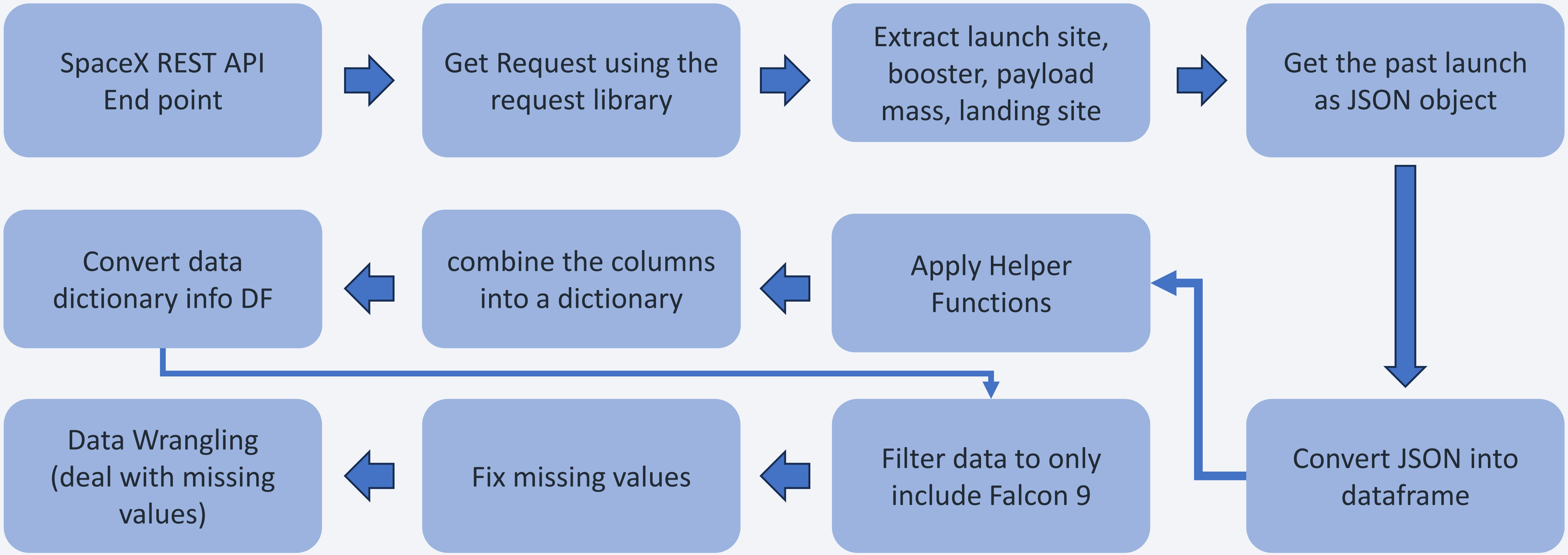
The data was collected from SpaceX REST API and launch tables from wiki pages using web scraping

SpaceX REST API → Get Request → Get Past Launch, JSON Object → Convert JSON to dataframe

Web Scraping → Web Scrap Wiki tables using BeautifulSoup → Parse Data from Table → Convert Tables into dataframe

Data Collection – SpaceX API

Collect data from SpaceX REST API and convert to right format



GitHub URL of the SpaceX API notebook: https://github.com/adelmka/Data-Science-Capstone/blob/main/LAB1_jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

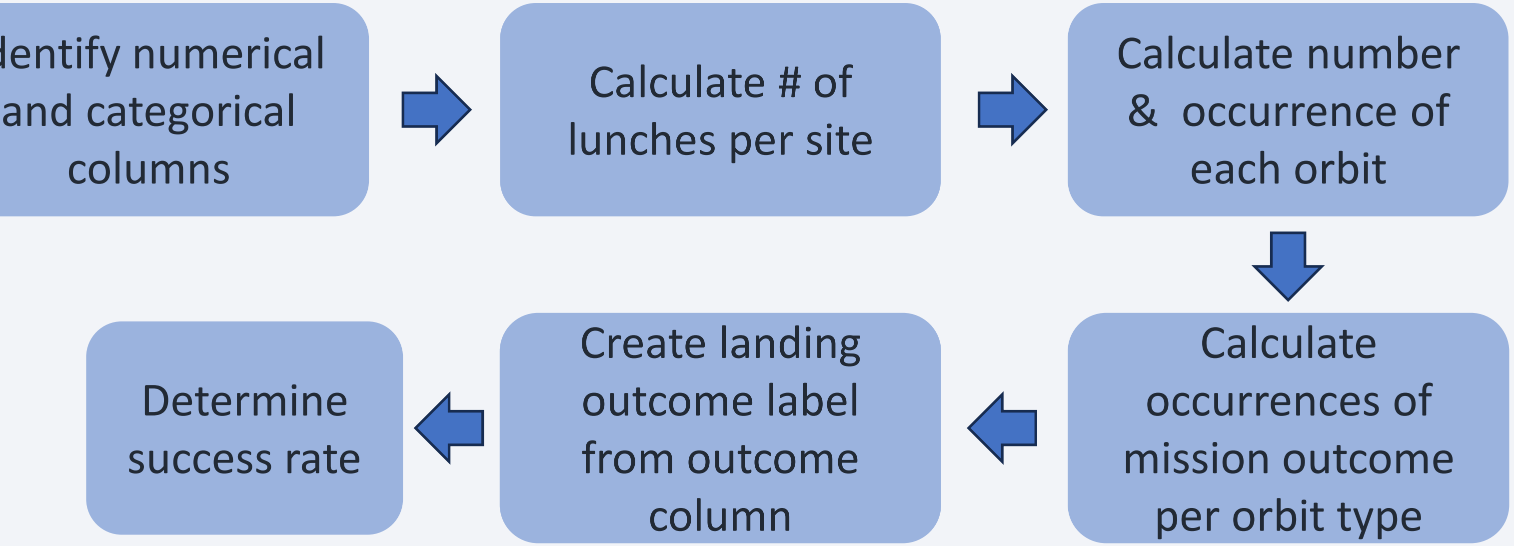
Perform web scarping to collect Falcon 9 launch data from wiki pages

Request the Falcon 9 lunch wiki URL → Create BeautifulSoup object from response → Extract all columns names from HTML table headers → Parse HTML tables and for past launches → Create dictionary with keys from the extracted columns → Create dataframe for past launches → Data Wrangling

GitHub URL of the Web Scraping notebook: https://github.com/adelmka/Data-Science-Capstone/blob/main/LAB2_jupyter-labs-webscraping.ipynb

Data Wrangling

- ✓ Identify missing data values and performed Exploratory Data Analysis (EDA) to find patterns in the data
- ✓ determine what would be the label for training supervised models
- ✓ obtain and create a Pandas DF from the collected data,
- ✓ filter data using the BoosterVersion column to only keep the Falcon 9 launches,
- ✓ dealt with the missing data values in the LandingPadand and PayloadMass columns.
- ✓ Replace PayloadMass missing data values using mean value.



Launch Classification Outcome:

1 – 1st stage landed successfully

0 – 1st stage didn't land successfully

GitHub URL of the Data Wrangling notebook: https://github.com/adelmka/Data-Science-Capstone/blob/main/LAB3_labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Performed data Analysis and Data Feature Engineering using Pandas and Matplotlib for Exploratory Data Analysis
- Use catplot the FlightNumber vs. PayloadMass and overlay the outcome of the launch.
- Use catplot to visualize the relationship between Flight Number and Launch Site, Payload and Launch Site, FlightNumber and Orbit type, Payload and Orbit type.
- Used Bar chart to visualize the relationship between success rate of each Orbit type.
- Line plot to Visualize the launch success yearly trend.

GitHub URL of the EDA with Data Visualization notebook: https://github.com/adelmka/Data-Science-Capstone/blob/main/LAB5_edadataviz.ipynb

EDA with SQL

Performed SQL queries and EDA to get insight from the data:

- ❑ The names of unique launch sites in the space mission.

  "SELECT DISTINCT Launch_Site FROM SPACEXTABLE"

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

  "SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5"

- ❑ The total payload mass carried by boosters launched by NASA (CRS)

  "SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)'"

- ❑ The average payload mass carried by booster version F9 v1.1

  "SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version like 'F9 v1.1%'"

- ❑ List the date when the first successful landing outcome in ground pad was achieved.

  "SELECT Min(Date) FROM SPACEXTABLE WHERE Mission_Outcome like 'Success%'"

- ❑ List the total number of successful and failure mission outcomes

  "SELECT TRIM(Mission_Outcome), COUNT(*) FROM SPACEXTABLE GROUP BY TRIM(Mission_Outcome)"

GitHub URL of the EDA with SQL notebook: https://github.com/adelmka/Data-Science-Capstone/blob/main/LAB4_jupyter-labs-eda-sql-coursera_sqllite.ipynb

EDA with SQL – continues…

Performed SQL queries and EDA to get insight from the data:

- ❑ List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

  “SELECT Booster_Version FROM SPACEXTABLE WHERE Mission_Outcome = 'Success'
  AND Landing_Outcome = 'Success (drone ship)’ AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000”

- ❑ List the names of the booster_versions which have carried the maximum payload mass.

  “SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ =
  (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)”

- ❑ The failed landing outcomes in drone ship, their booster version and launch site names.

  “SELECT substr(Date, 6, 2) AS Month, Booster_Version, Launch_Site, Landing_Outcome FROM SPACEXTABLE
  WHERE substr(Date, 0, 5) = '2015’ AND Landing_Outcome = 'Failure (drone ship)’”

- ❑ 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.

  “SELECT Landing_Outcome, COUNT(*) FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
  GROUP BYLanding_Outcome ORDER BY COUNT(*) DESC”

GitHub URL of the EDA with SQL notebook: https://github.com/adelmka/Data-Science-Capstone/blob/main/LAB4_jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Created folium map to marked all the launch sites with an initial center location to be NASA Johnson Space Center at Houston, Texas.
- Added each site's location on a map using site's latitude and longitude coordinates.
- Created map objects such as markers, circles, lines to mark launches for each launch site:
- Adde highlighted circle area with text label using folium.Circle & folium.Marker
- Added a MousePosition on the map to get coordinate for a mouse over a point on the map
- Drew a line between a launch site to proximities (e.g., city, railway, highway) using foliumPolyLine object
- Mark the success/failed launches for each site on the map by assigning the feature launch outcomes (success=1, failure=0).
- Identified which launch sites have high success rate with color labeled marker clusters.
- Calculated the distances between a launch site to its proximities.

GitHub URL of the interactive map Folium map notebook: https://github.com/adelmka/Data-Science-Capstone/blob/main/LAB6_lab_jupyter_launch_site_location.ipynb

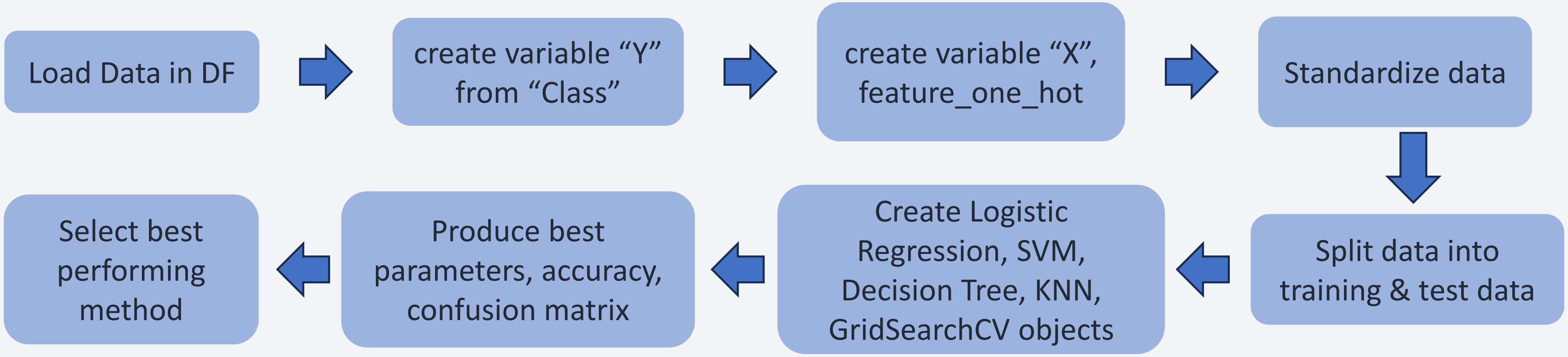
Build a Dashboard with Plotly Dash

- Created an interactive dashboard to analyze SpaceX launch data in real-time.
- The dashboard provides a user-friendly interface for exploring SpaceX launch data.
- Users can interactively investigate the relationship between launch site and success rates, as well as the impact of payload weight on launch outcomes.
- **Components and Interactions:**
  - **Dropdown Menu**
    - Allows users to select different launch sites.
    - Linked to a callback function that updates a pie chart.
  - **Pie Chart**
    - Visualizes launch success counts for the selected launch site.
    - Callback function updates the Pie Chart based on site selection.
  - **Range Slider**
    - Enables users to filter data based on payload range.
    - Linked to a callback function that updates scatter plot
  - **Scatter Plot:**
    - Visualizes the relationship between payload and mission outcome for selected launch sites.
    - Callback function updates the plot based on site selection.

GitHub URL of the interactive Plotly Dash lab: https://github.com/adelmka/Data-Science-Capstone/blob/main/LAB7_spacex_dash_app.py

Predictive Analysis (Classification)

- Loaded the data using Numpy and Pandas, transformed the data, split the data into training and testing.
- Built different machine learning models and tune different hyperparameters using GridSearchCV.
- Used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- found the best performing classification model.



GitHub URL of the interactive Plotly Dash lab: [https://github.com/adelmka/Data-Science-Capstone/blob/main/LAB8_SpaceX_Machine Learning_Prediction_Part_5.ipynb](https://github.com/adelmka/Data-Science-Capstone/blob/main/LAB8_SpaceX_Machine_Learning_Prediction_Part_5.ipynb)

Results

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

- Are launch sites in close proximity to railways? *No*
- Are launch sites in close proximity to highways? *No*
- Are launch sites in close proximity to coastline? *Yes*
- Do launch sites keep certain distance away from cities? *Yes*

- Average payload mass carried by booster version F9 v1.1: *2534.66 KG*

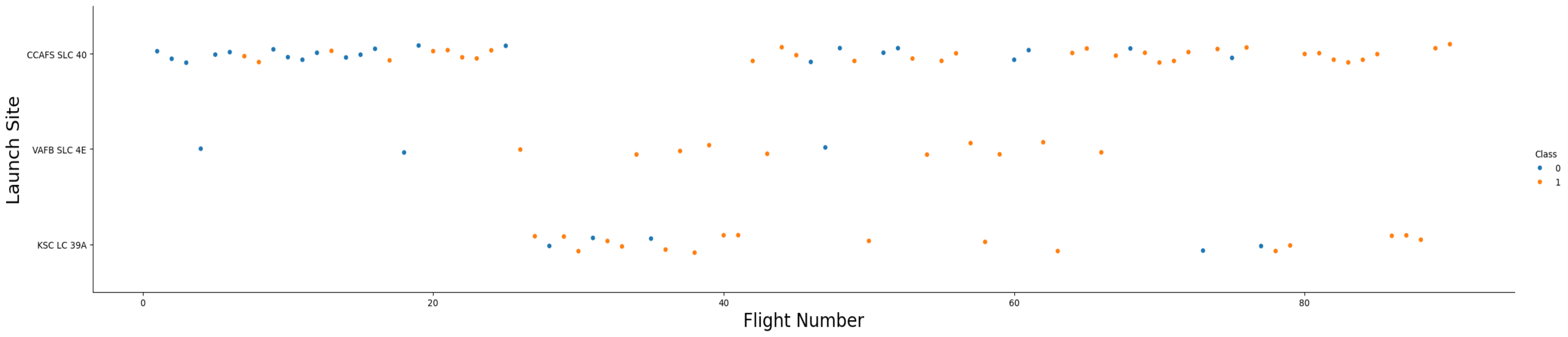
- Total payload mass carried by boosters launched by NASA (CRS): *45596 KG*

- Date when the first successful landing outcome in ground pad was achieved: *2010-06-04*

Section 2

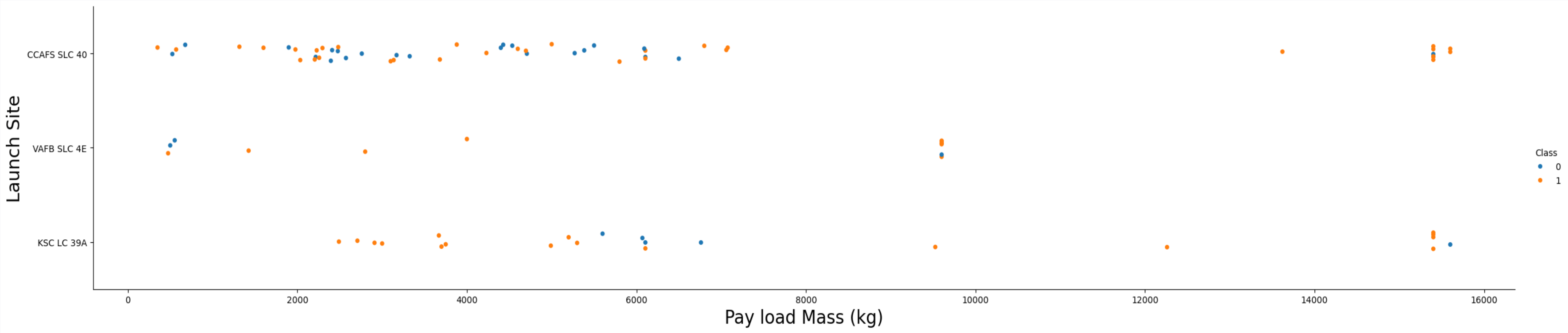
Insights drawn from EDA

Flight Number vs. Launch Site



- As the number of flights increased, the success rate increased for all sites.
- Site “CCAFS SLC40” has majority of flights, with no failures after the 80th flight.
- Site “KSC LC 39C” is #2 in terms of number of flights, with no failure after 80th flight.
- Site “VAFB SLC 4E” has the least flights, with 100% success after the 50th flight.

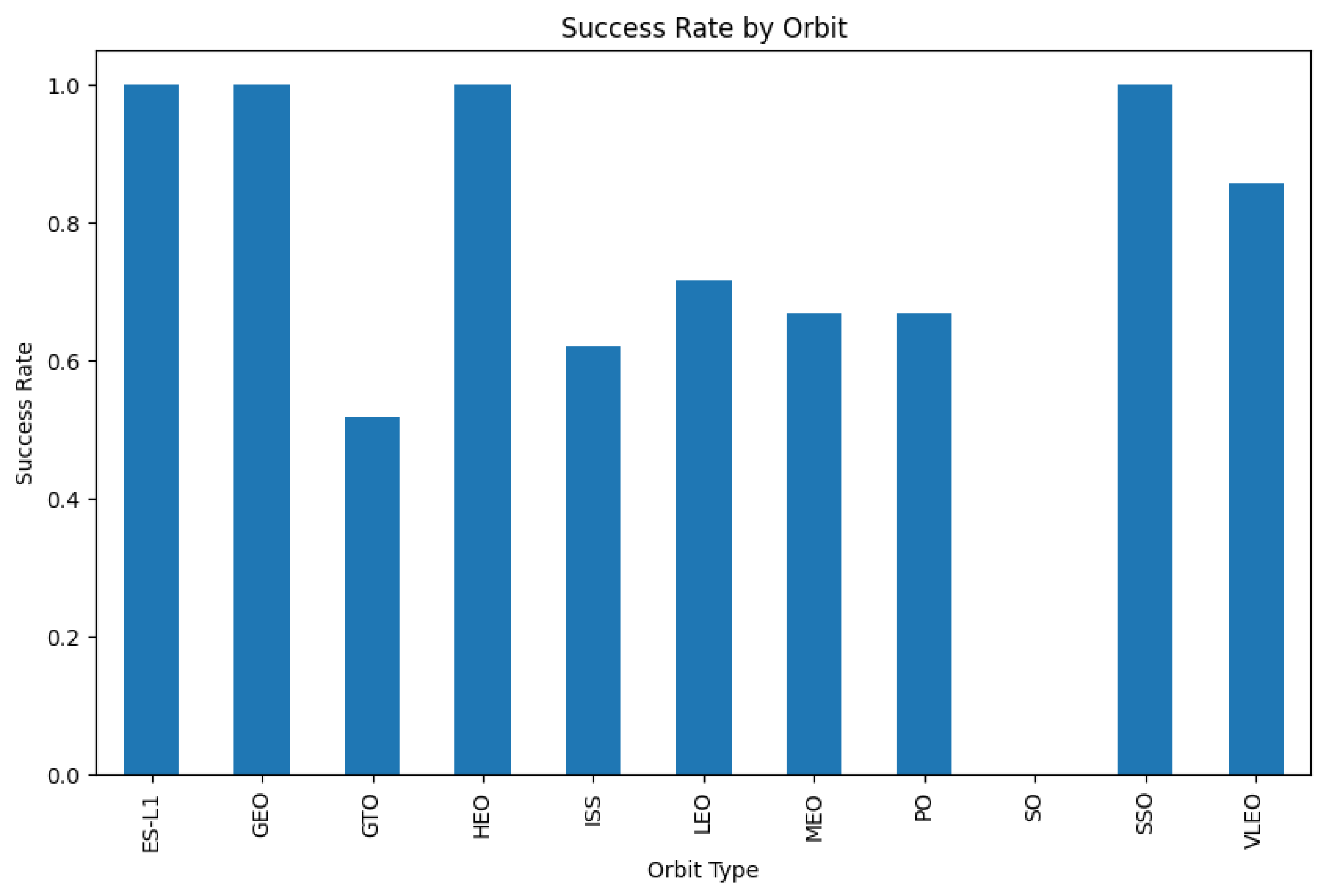
Payload vs. Launch Site



- Launch site “VAFB SLC 4E” has no flight with heavier than 10,000 kg
- Launch site “KSC LC 39C” has no flights with payload less than 2,500 kg
- Launch site had no flights with payload between 7,500 kg and 13,000 kg

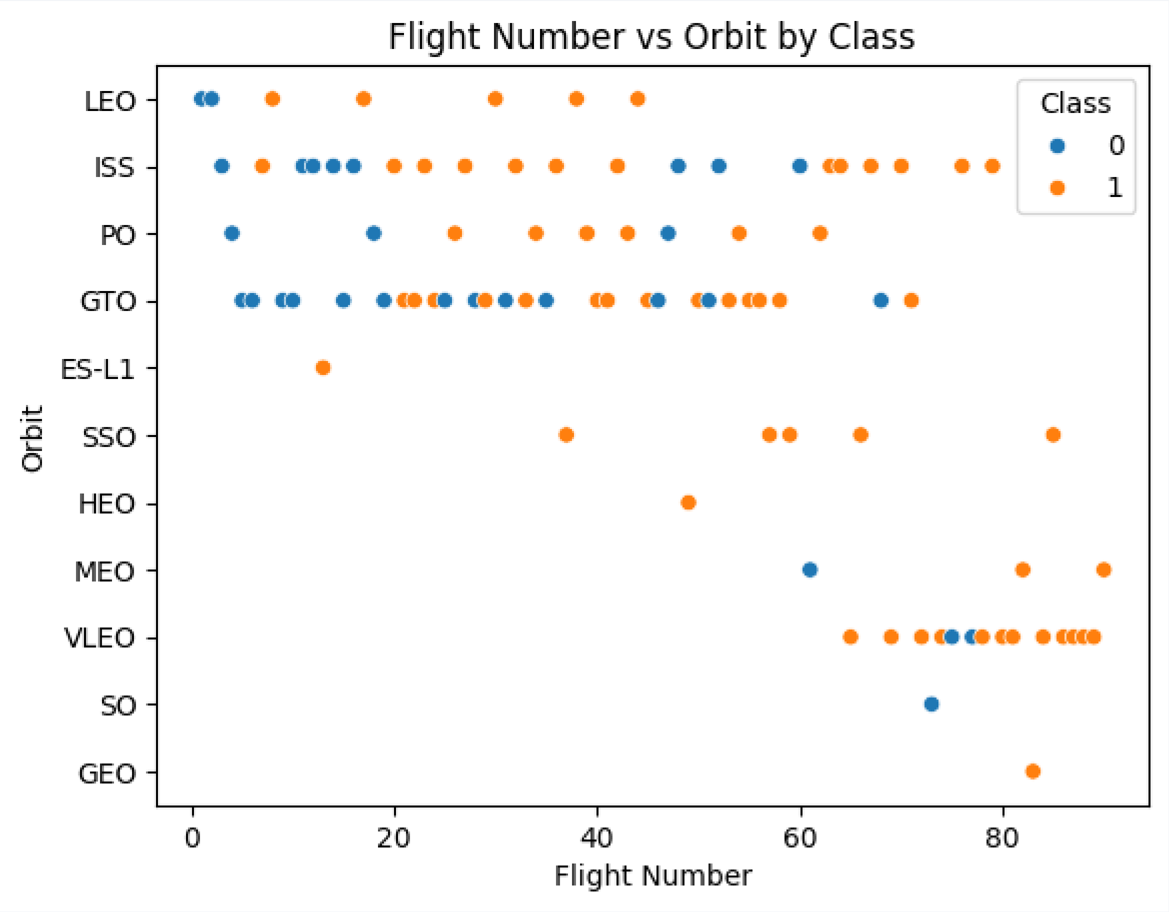
Success Rate vs. Orbit Type

- Orbit Types ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- Orbit Type VLEO had the next beast success rate (after the above 4).
- Orbit Type SO had 0 success rate
- We need to analyze the # of flights of each Orbit Type by Class (success/failure)

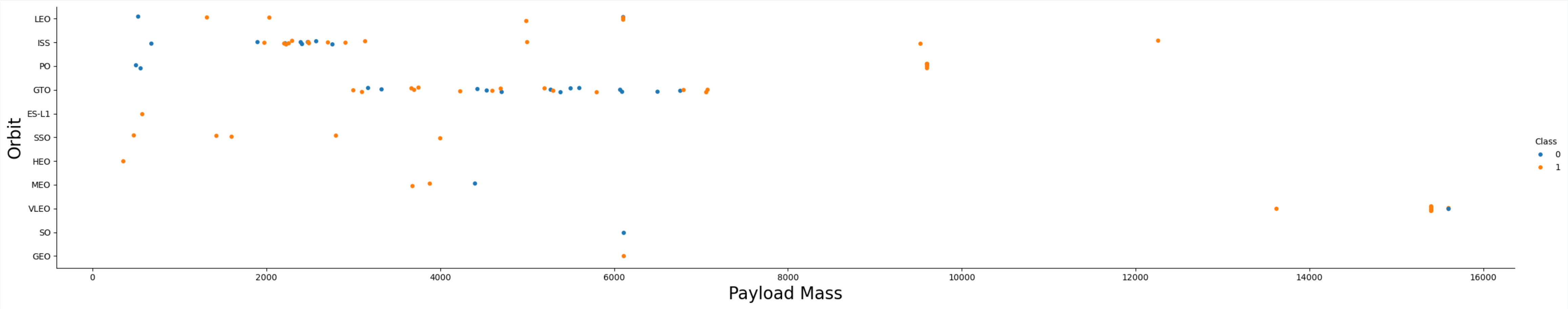


Flight Number vs. Orbit Type

- Orbit Type vs. Flight Number for GTO, it seems that the Class is split 50/50 (no relation)
- Orbits Types ISS, GTO with more flight number yet with the most failures
- Orbit Types with highest success rate have the least launches



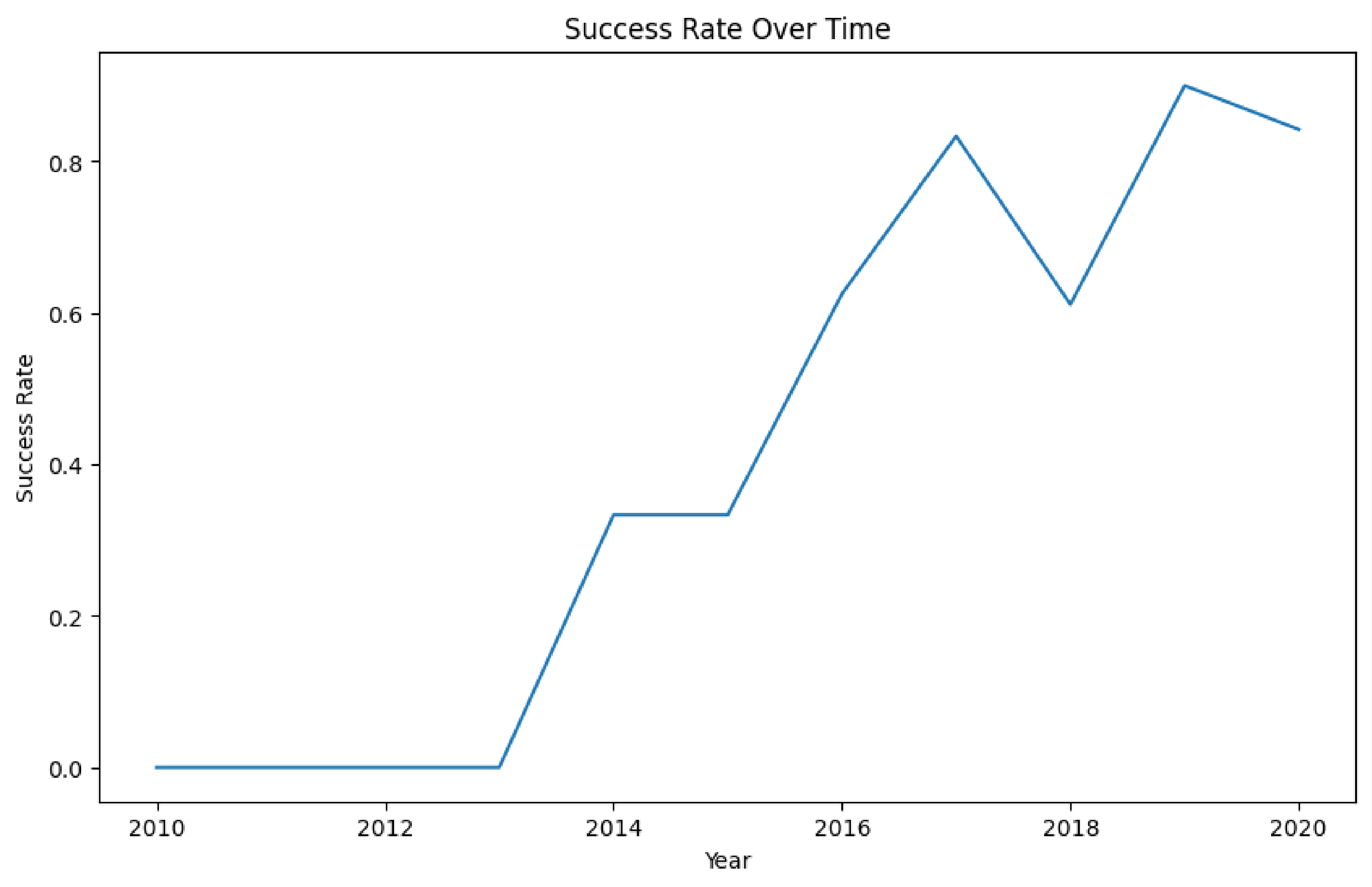
Payload vs. Orbit Type



- Orbit Types LEO, ISS and PO have the higher success rate when payload is heavier.
- For Orbit Type GTO, it seems that the Class is split 50/50 (no relation with payload)

Launch Success Yearly Trend

- Starting 2013, Launch success had upward trend until 2020
- Noticeable dip (~0.20) in 2018



All Launch Site Names

- Use “DISTINCT” to find the Launch Site names

%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE;

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

```
[17]: # added to check alternative SQL
      %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE;

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

[17]:

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`
- Used LIKE ‘CCA%’ to specify sites names that begin with ‘CCA’ and LIMT 5 to limit the result to 5 records

```
%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
2010-06-04	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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Used function SUM to find the total payload for NASA
- Specified the ‘Customer’ to ‘NASA (CRS)’
- The result - 45596

```
%sql SELECT SUM(PAYLOAD_MASS__KG_), Customer FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)'
```

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

SUM(PAYLOAD_MASS__KG_)	Customer
45596	NASA (CRS)

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Used function ‘AVG’ to calculate the payload mass for ‘Booster_Version’ that starts with ‘F9 v1.1’

the result – 2534.66

```
%sql SELECT AVG(PAYLOAD_MASS__KG_), customer,Booster_Version  FROM SPACEXTABLE WHERE Booster_Version like 'F9 v1.1%'
```

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

AVG(PAYLOAD_MASS__KG_)	Customer	Booster_Version
2534.6666666666665	MDA	F9 v1.1 B1003

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Used function ‘Min’ to find the minimum dates for ‘Mission_outcome’ is ‘Success’
- the result - 2010-06-04

```
%sql SELECT Min(Date) FROM SPACEXTABLE WHERE Mission_Outcome like 'Success%'
```

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

Min(Date)
2010-06-04

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- Used SQL to get the 'Booster_Version'
- Qualified the 'Mission_Outcome' is 'Success' and 'Landing_Outcome' is 'Success(drone ship)' and 'PAYLOAD_MASS__KG_' between 4000 and 6000.

```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE Mission_Outcome = 'Success' AND 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

- Calculate the total number of successful and failure mission outcomes
- Used function 'COUNT(*)' to get the count and 'GROUP BY' to group the result by mission outcome.
- Assigned the fetched records to 'results' and put the outcome in list and printed.

```
%sql SELECT TRIM(Mission_Outcome), COUNT(*) FROM SPACEXTABLE GROUP BY TRIM(Mission_Outcome)
```

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

TRIM(Mission_Outcome)	COUNT(*)
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Used nested ‘SELECT’ and function ‘MAX’ to get the maximum payload mass

```
%sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

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

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Used function ‘substr()’ in the SELECT to get the month and ‘function ‘substr()’ in the WHERE clause to limit the year to 2015 from the ‘Date’ and limit the ‘Landing_outcome’ to ‘Failure (drone ship)’
- printed the resulted list

```
%sql SELECT substr(Date, 6, 2) AS Month, Booster_Version, Launch_Site, Landing_Outcome FROM SPACEXTABLE
      WHERE substr(Date, 0, 5) = '2015' AND Landing_Outcome = 'Failure (drone ship)'
```

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

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

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

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (drone ship)) between the date 2010-06-04 and 2017-03-20, in descending order
- Used function 'COUNT(*)' and used 'WHERE Date between' clause to limit the date and used 'GROUP BY' to aggregate the result by the 'Landing_Outcome'.

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

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

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

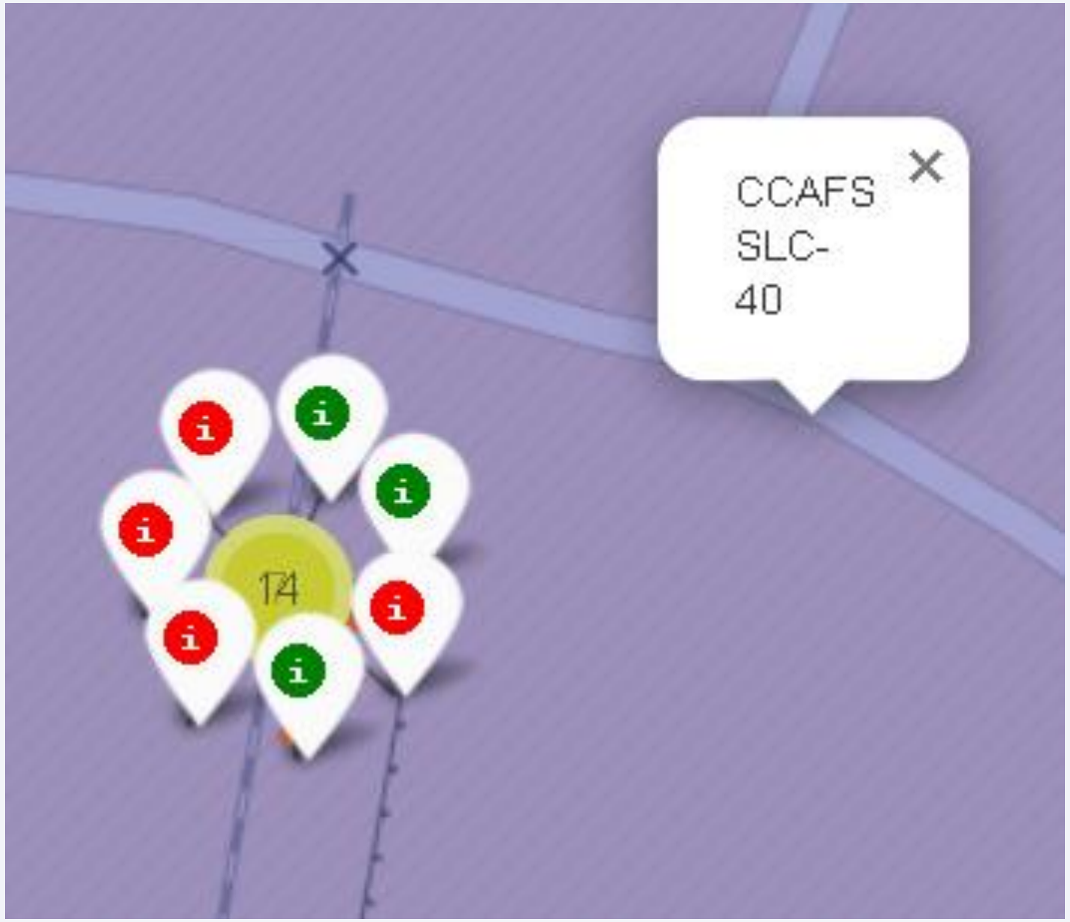
Section 3

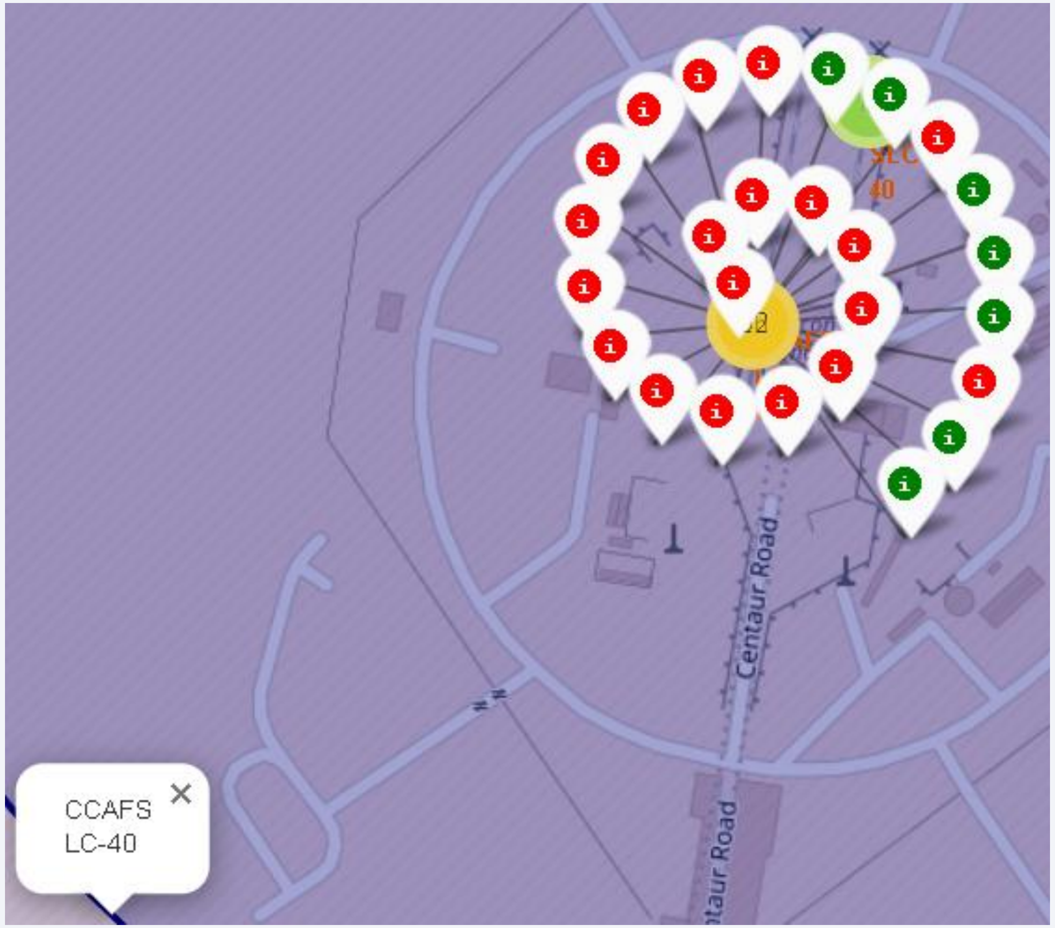
Launch Sites Proximities Analysis

All Launch Sites on the Map

- All Lunch Sites are in the United States (California, Florida) in proximity to the coast and the Equator.

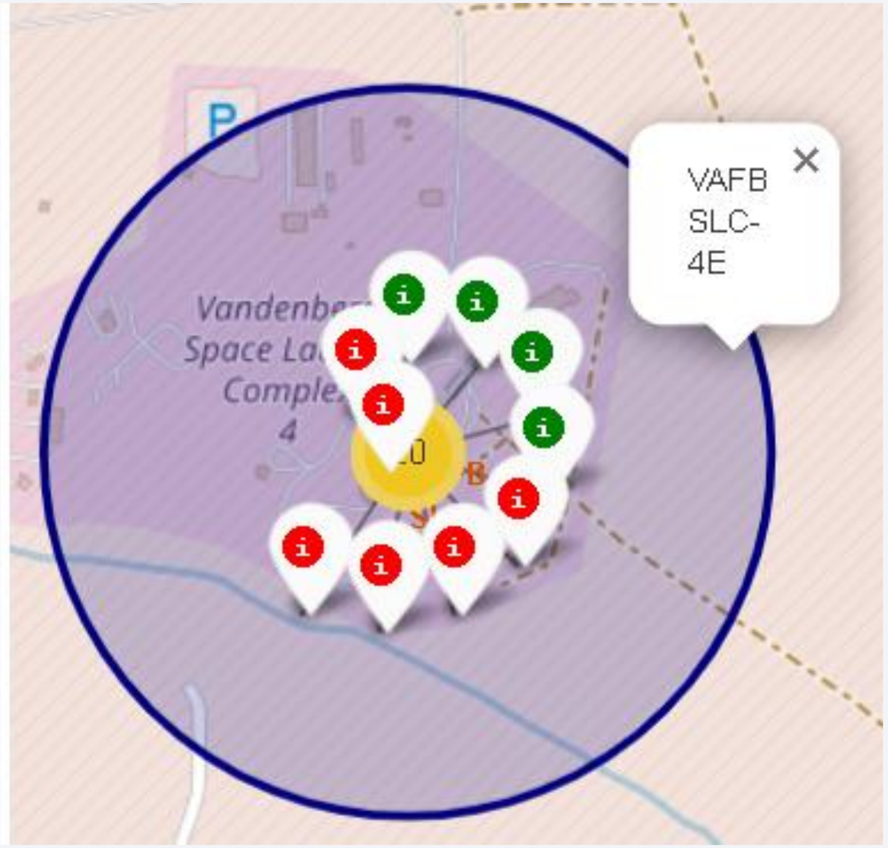
The Success/Failed Launches for Each Site Marked on The Map

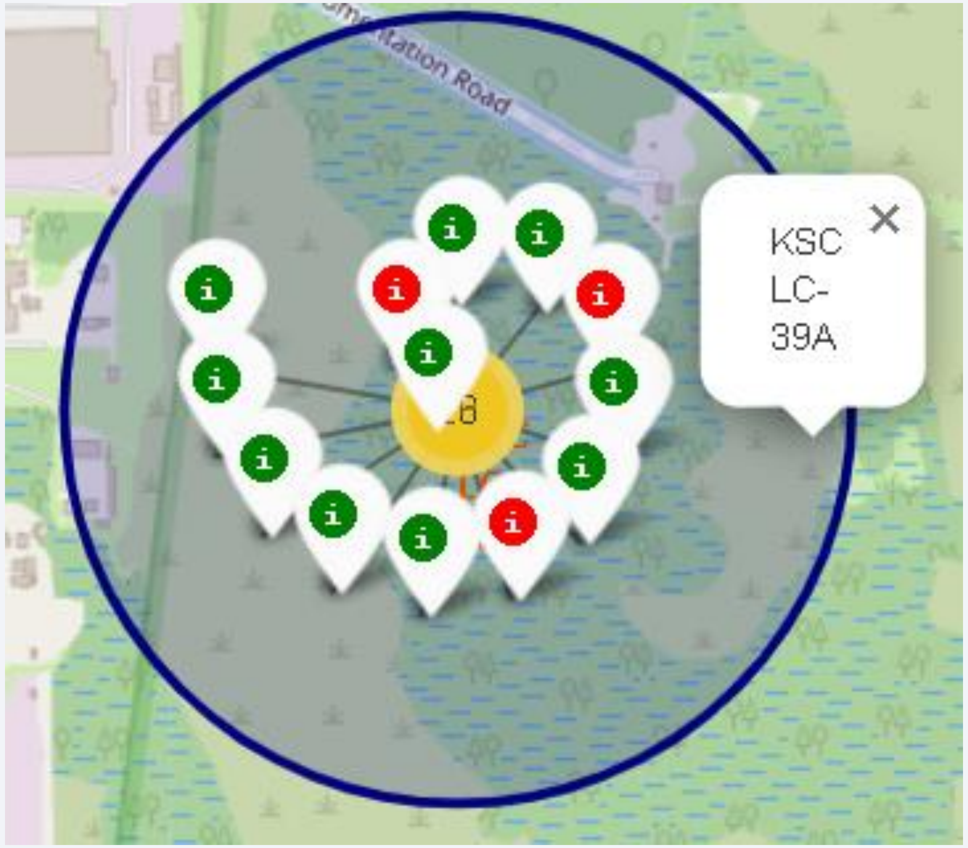




- Launch site KSC LC-39A has relatively high success rates compared to CCAFS SLC-40 & CCAFS LC-40.

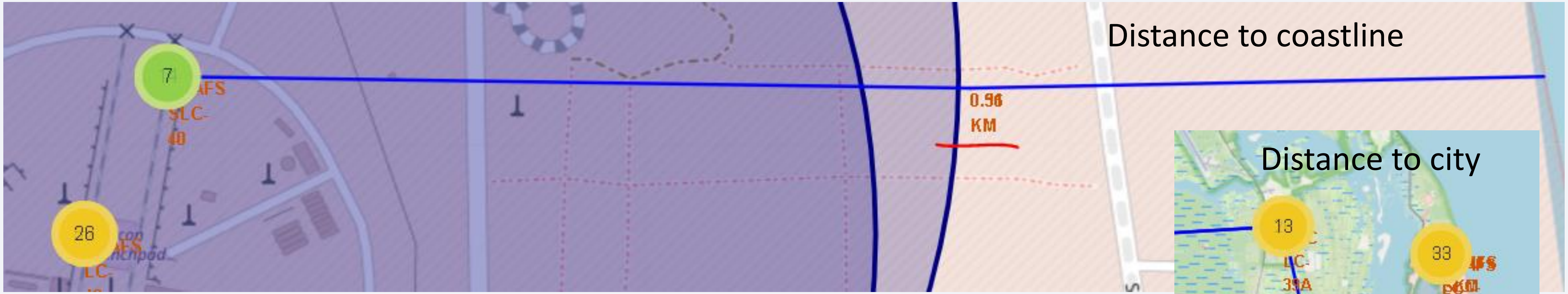
The Success/Failed Launches for Each Site Marked on The Map



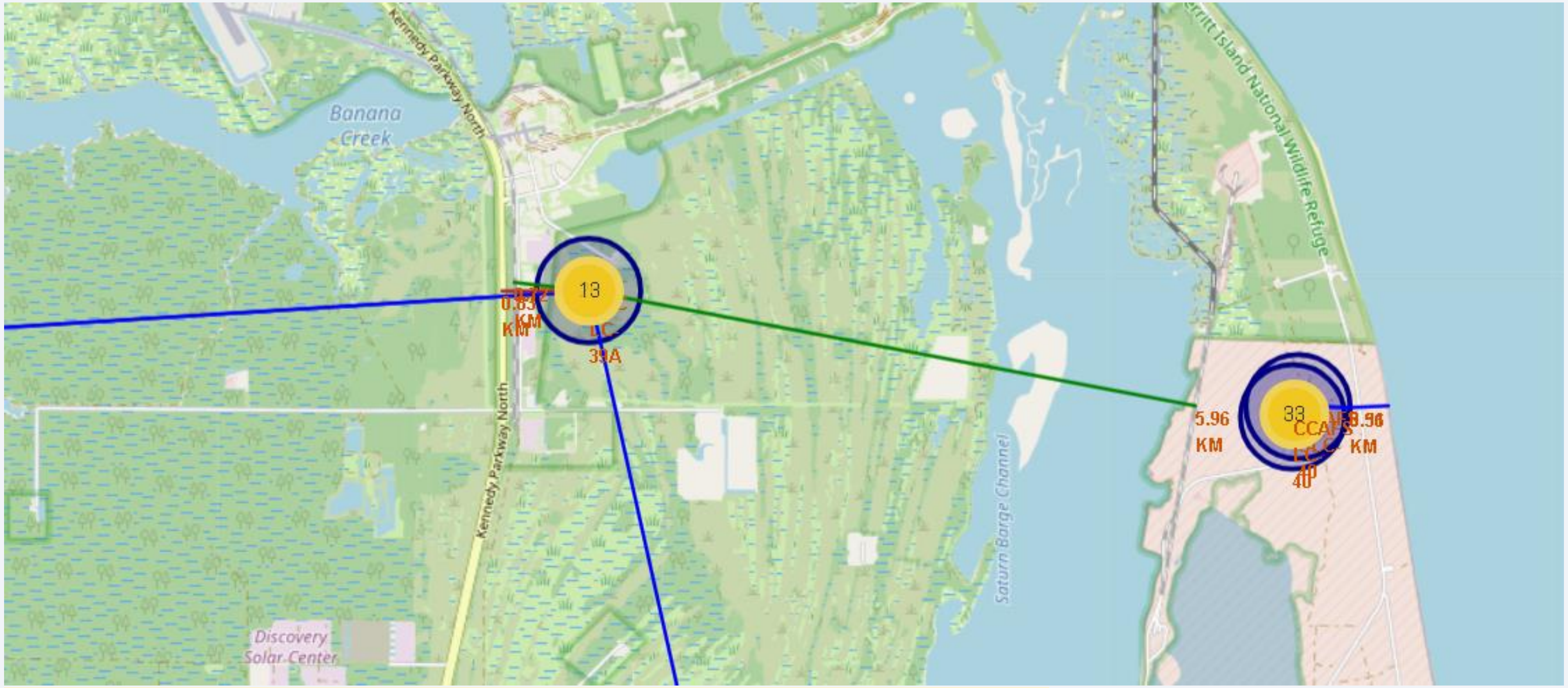


- Launch site VAFB SLC-4E has relatively lower success rates compared to KSC LC-39A.

Distances between launch sites to proximities
(Coastline, Railroad, City, Highway)



Distances between launch sites to proximities (Coastline, Railroad, City, Highway)

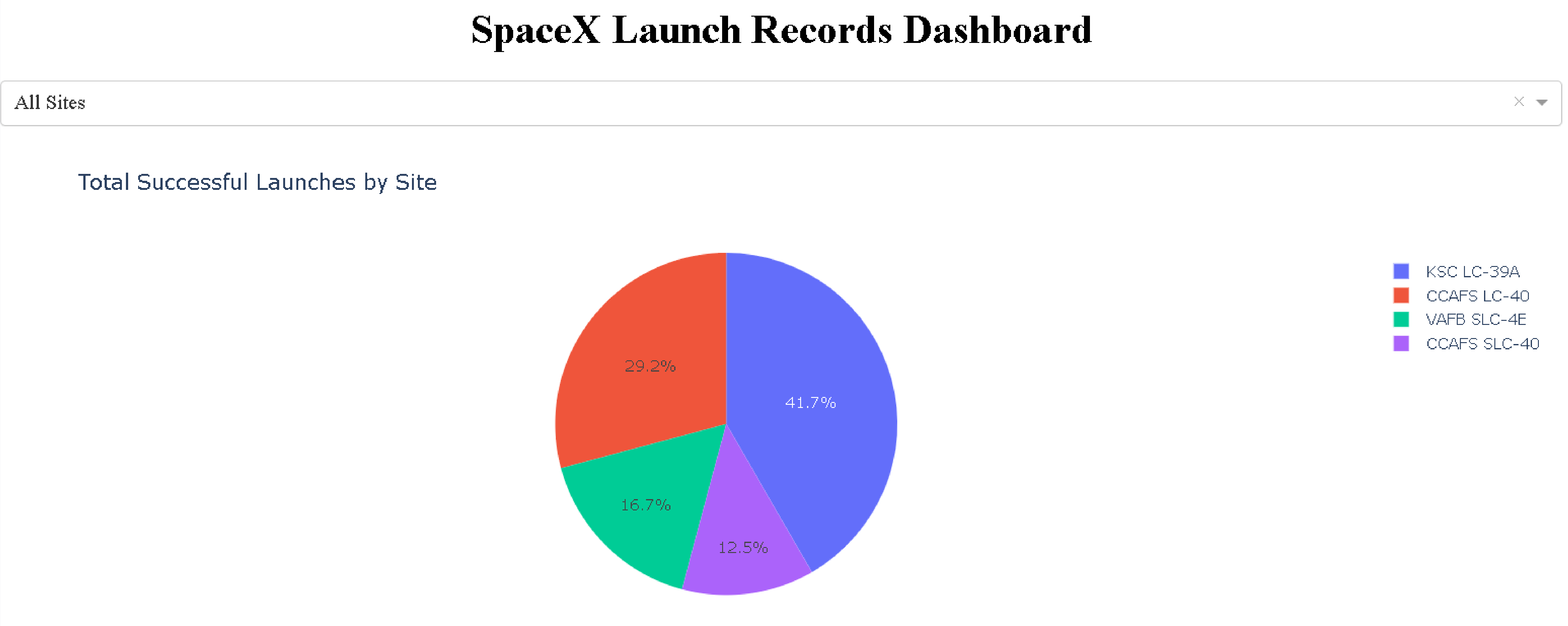


- Are launch sites in close proximity to railways? *No*
- Are launch sites in close proximity to highways? *No*
- Are launch sites in close proximity to coastline? *Yes*
- Do launch sites keep certain distance away from cities? *Yes*

Section 4

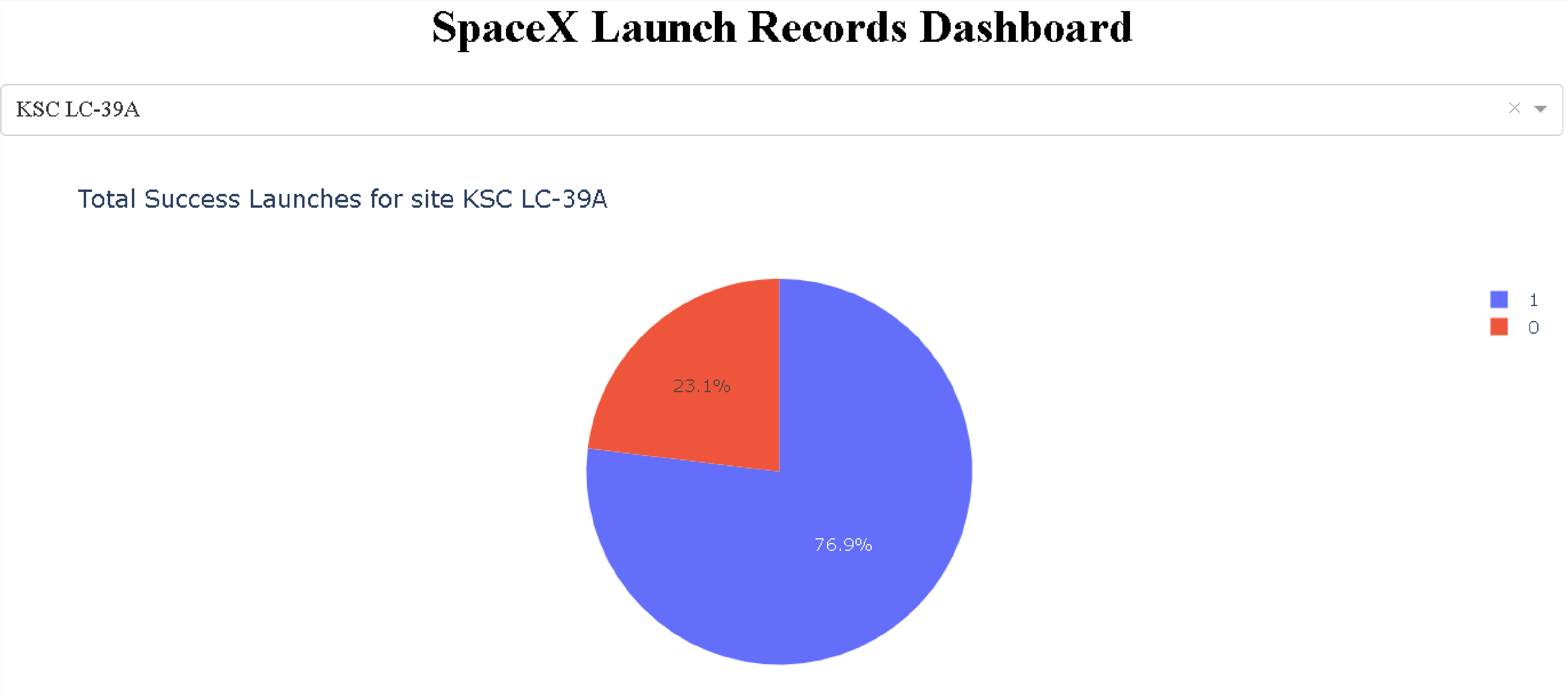
Build a Dashboard with Plotly Dash

Total Successful Launches for all Sites



- KSC LC-39A has the highest success rate (41.7%) followed by CCAFS LC-40 (29.2%), VAFB SLC-4E (16.7%), CCAFS SLC-40 (12.5%)

Successful Launches Rate for KSC LC-39A



- Launch site KSC LC-39A achieved ~77% successful launches.

Payload vs. Launch Outcome for all Sites

- Booster FT has the highest success rates (between 2000kg – 6000kg).

Payload vs. Launch Outcome for all Sites

- The heavier the payload, the higher the success rate (between 0 – 4000kg).

Payload vs. Launch Outcome for all Sites

- Between 5500 kg and 7000 kg payload, majority of outcomes are failures

Section 5

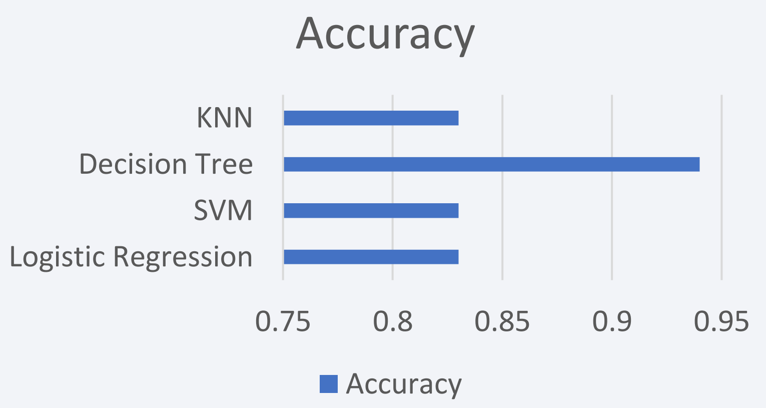
Predictive Analysis (Classification)

Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart

- *Best Performing Model: Decision Tree with an accuracy of 0.94*

Model	Accuracy
Logistic Regression	0.83
SVM	0.83
Decision Tree	0.94
KNN	0.83

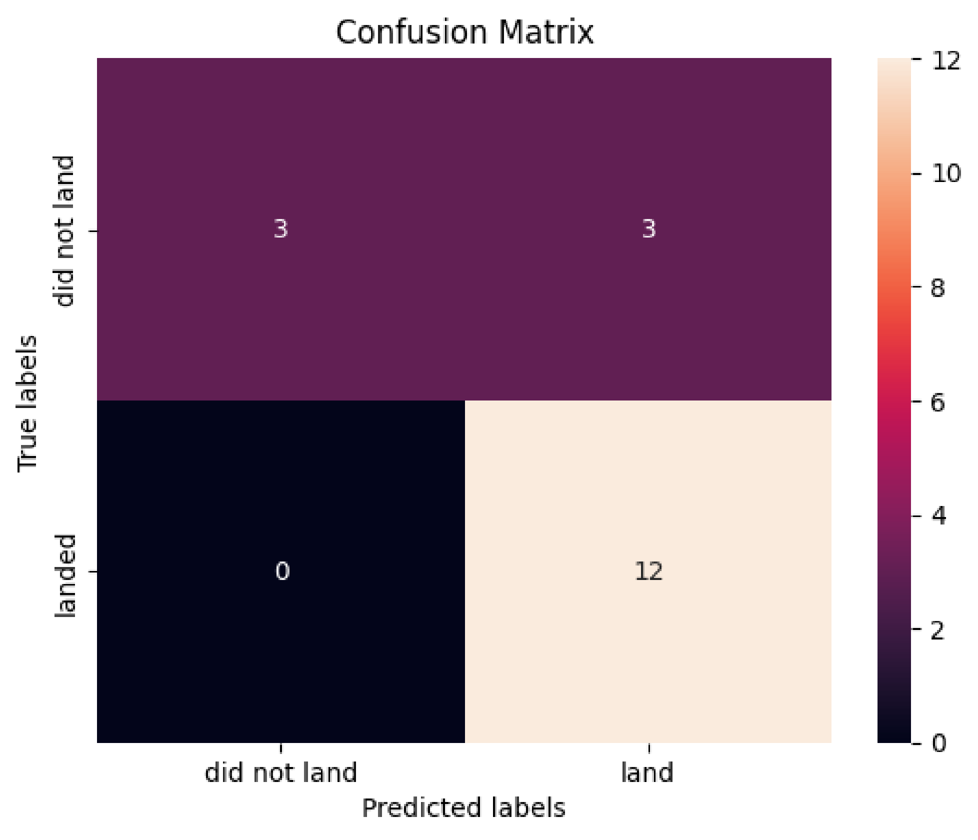


Method Performing Best:
Logistic Regression Test Accuracy: 0.8333333333333334
SVM Test Accuracy: 0.8333333333333334
Decision Tree Test Accuracy: 0.9444444444444444
KNN Test Accuracy: 0.8333333333333334

Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation
- 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.

- *Best Performing Model: Decision Tree with an accuracy of 0.94*



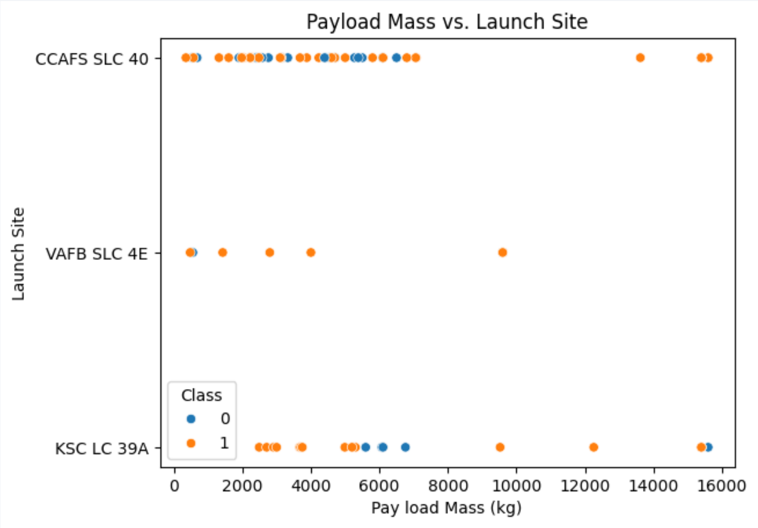
Conclusions

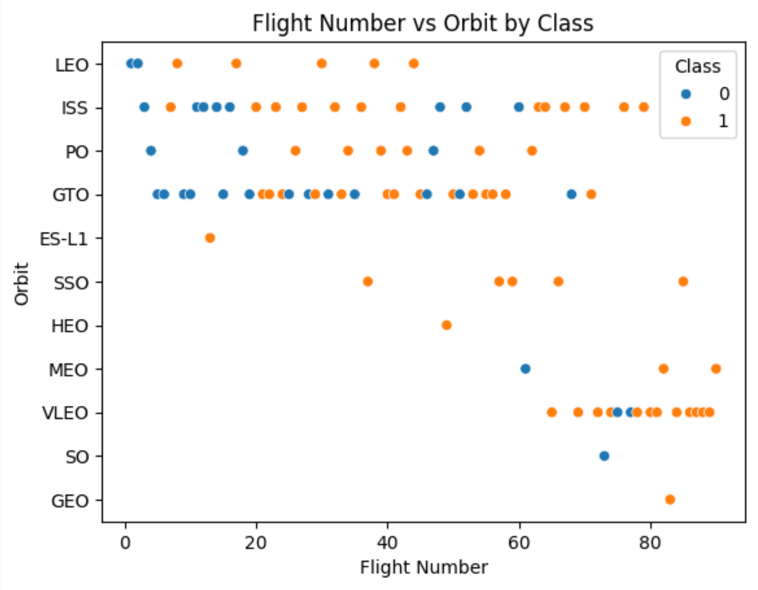
- The larger the flight at a launch site, the greater the success rate.
- Launch success rate started to increase in 2013 till 2020.
- KSC LC-39A had the most successful launches of any sites.
- Launch site KSC LC-39A has the highest launch success rate (42%) followed by CCAFS LC-40 (29%), VAFB SLC-4E (17%), CCAFS SLC-40 (13%)
- Launch site CCAFS LC-40 had the 2nd highest success ratio of 73% success against 27% failed launches
- For launch site CCAFS LC-40, the booster version FT has the largest success rate for payload mass >2000kg
- Booster FT has the highest success rates (between 2000kg – 6000kg).
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate with heavier payload.
- The Decision Tree classifier is the best machine learning algorithm with 94% accuracy.

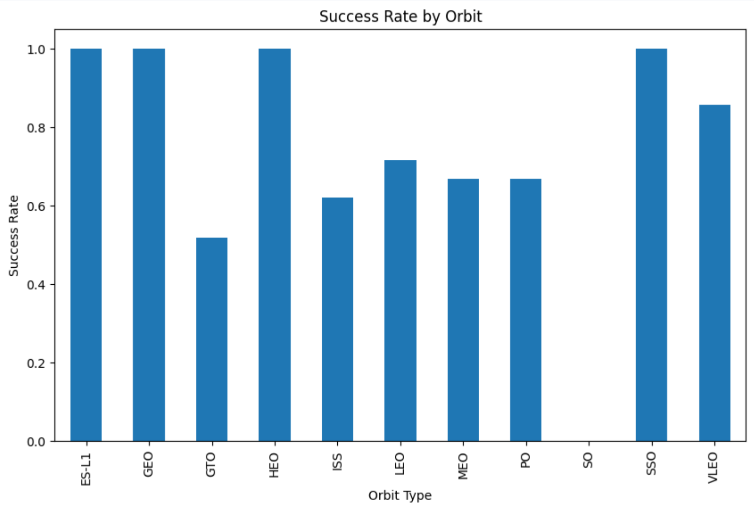
Appendix

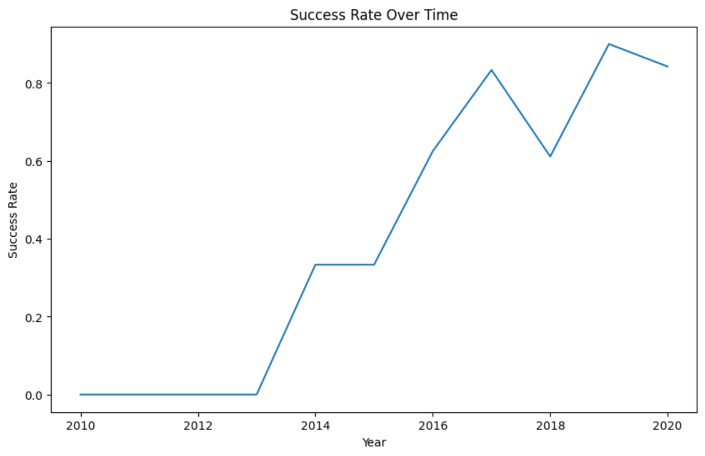
- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Results: Exploratory data analysis results

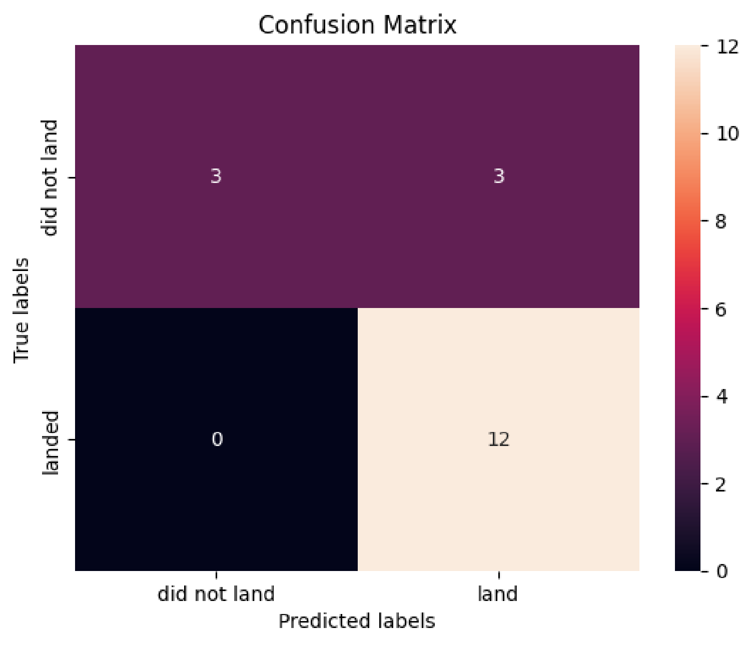


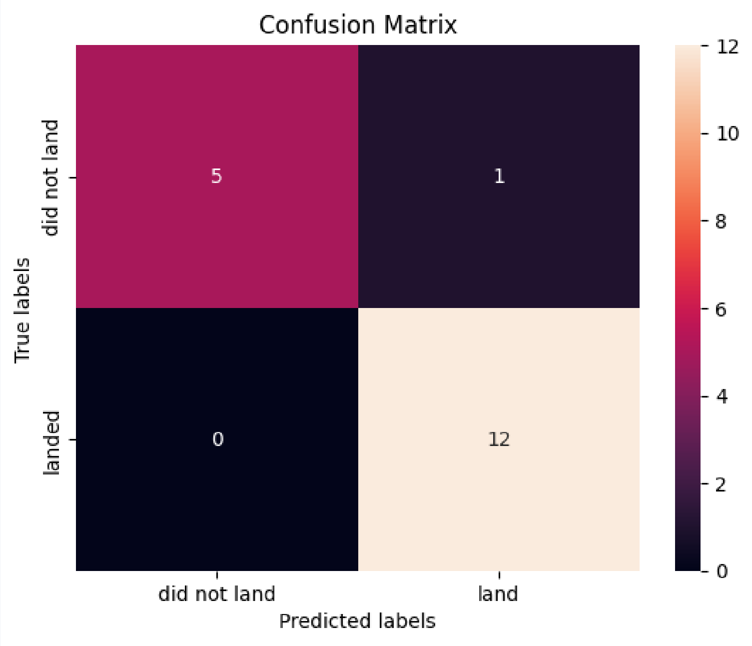


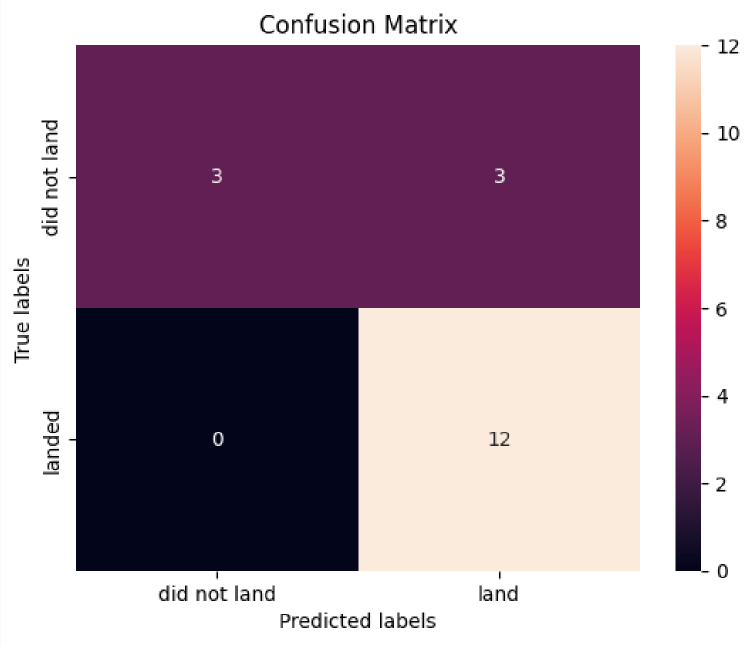




Results: Predictive analysis results





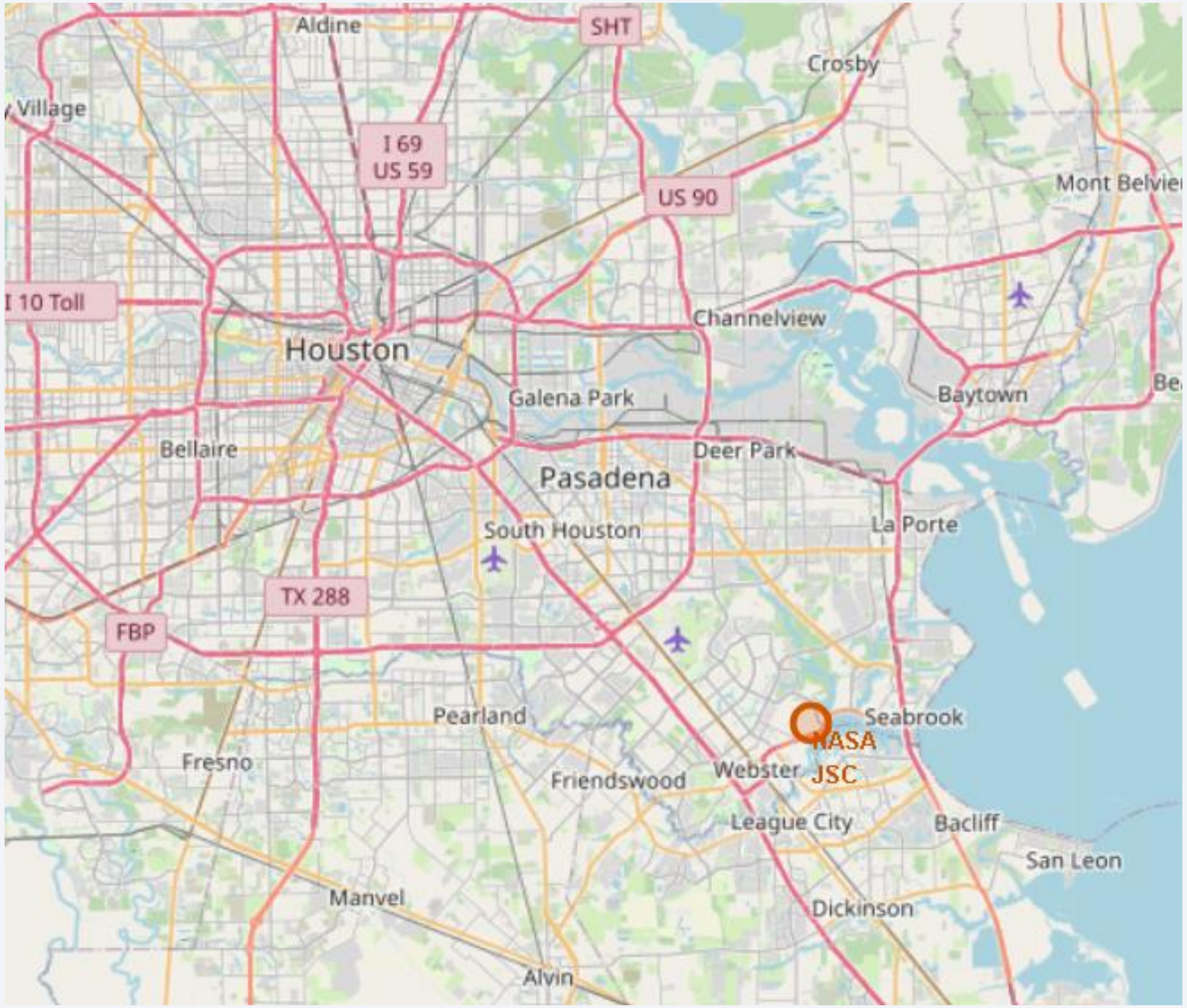


<u>Method Performing Best:</u>

Logistic Regression Test Accuracy: 0.8333333333333334
SVM Test Accuracy: 0.8333333333333334
Decision Tree Test Accuracy: 0.9444444444444444
KNN Test Accuracy: 0.8333333333333334

Best Performing Model:
Decision Tree with an accuracy of 0.94

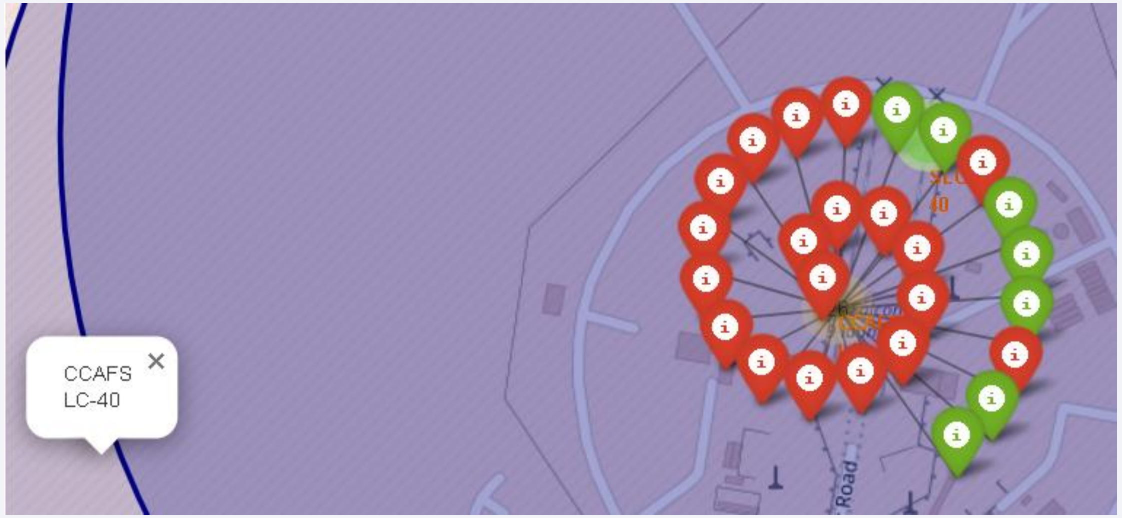
NASA Johnson Space Center – NASA JSC

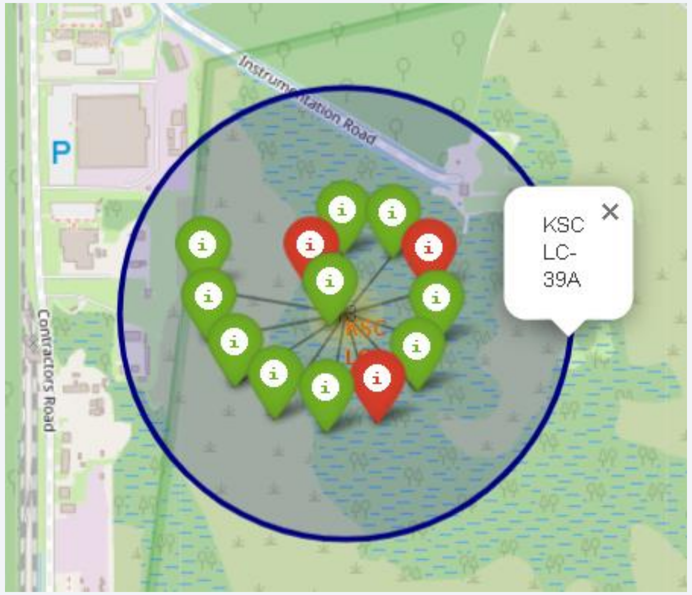


Launch Sites (Florida - CCAFS SLC-40, CCAFS SLC-40, KSC LC-39A)

Launch Site (California - VAFB SLC-4E)

Launch Sites – Success, Failures Markers





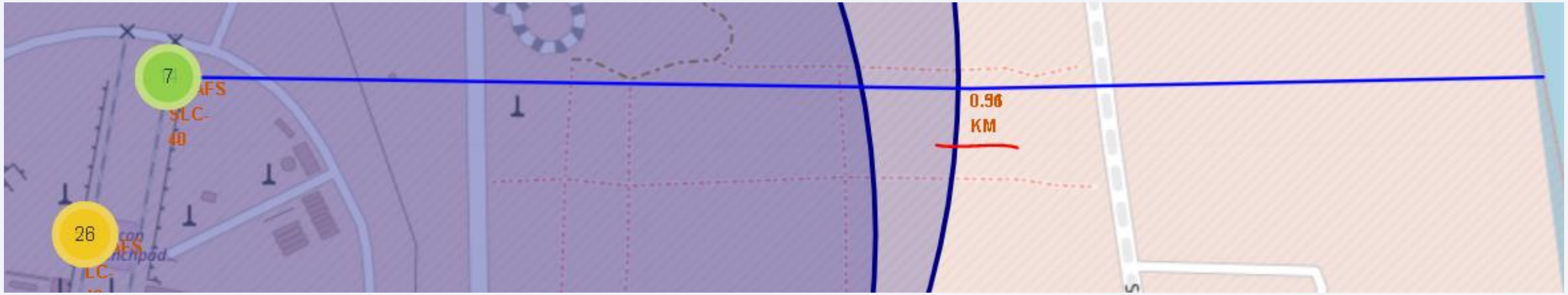
Adding Mouse Position on the Map

Launch Site Distance to Coastline

```
# find coordinate of the closet coastline
# e.g.,: Lat: 28.56367  Lon: -80.57163
# distance_coastline = calculate_distance(launch_site_lat, launch_site_lon, coastline_lat, coastline_lon)
# 49    CCAFS SLC-40    28.563197   -80.576820  1
#Lat:28.56313 Lon:-80.56792
launch_site_lat = 28.563197
launch_site_lon = -80.576820
# Example usage:
coastline_lat = 28.56313  # Replace with actual coastline latitude
coastline_lon = -80.57163  # Replace with actual coastline longitude
distance_to_coast = calculate_distance(launch_site_lat, launch_site_lon, coastline_lat, coastline_lon)
print("Distance to coastline:", distance_to_coast)
```

```
Distance to coastline: 0.5070768263679413
```

Launch site CCAFS SLC-40 proximity to coastline is 0.50km



Launch Sites Coordinates

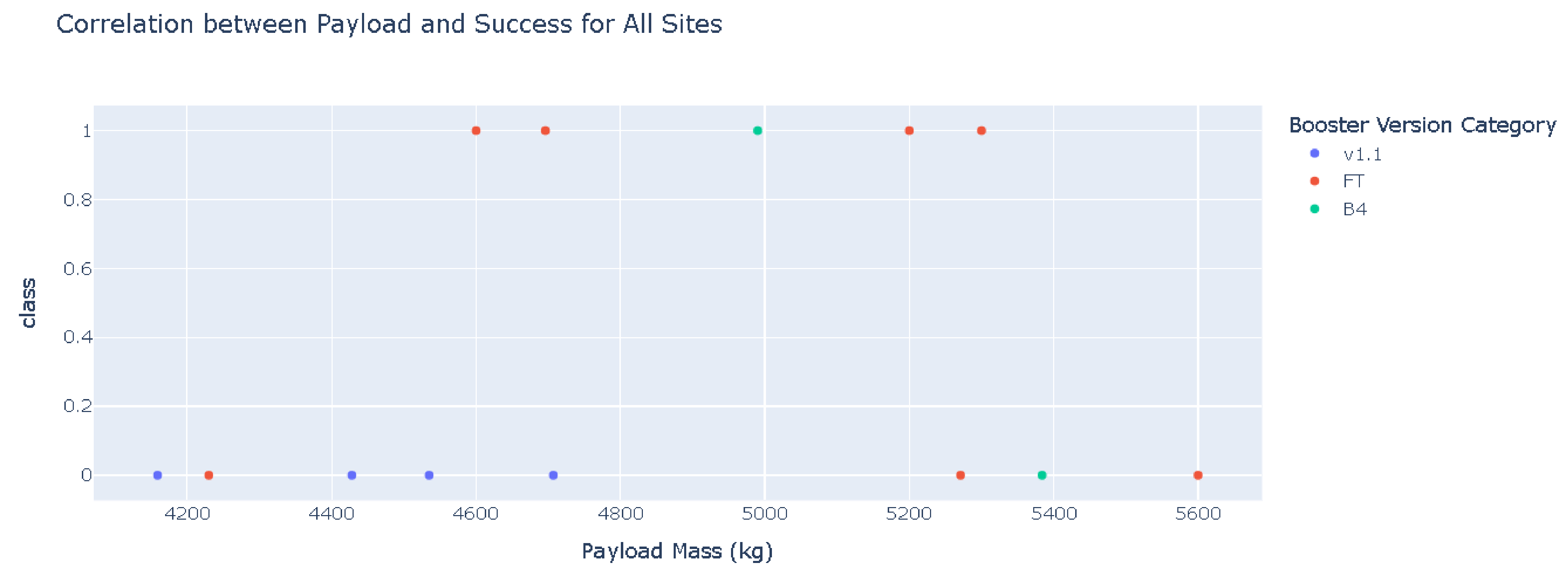
```
# Select relevant sub-columns: `Launch Site`, `Lat(Latitude)`, `Long(Longitude)`, `class`
spacex_df = spacex_df[['Launch Site', 'Lat', 'Long', 'class']]
launch_sites_df = spacex_df.groupby(['Launch Site'], as_index=False).first()
launch_sites_df = launch_sites_df[['Launch Site', 'Lat', 'Long']]
launch_sites_df
```

	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610745

	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610745

Correlation Between Payload vs Success

Correlation Between Payload vs Success



- boosters which have success/failures with payload mass between 4000 to 6000 kg.

Thank you!