IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Preston Moore
July 28, 2023

Outline

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualizations
  - Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

Github Repository Links: https://github.com/PrestonMoore3/IBM-Data-Science-Capstone.git

PrestonMoore3/IBM-Data-Science-Capstone: Completion of Tasks to Predict SpaceX Falcon9 Launch and Landing (github.com)

Executive Summary

Problem

- Will the First Stage of the Space X Falcon 9 Successfully Launch and Land?
  - Provide a predicted cost of a launch
- Potentially Save 103 million dollars or more.
- Provides information to see if alternative bids from other companies should be made.

Solution:

- I predicted the Space X Falcon 9 First Stage will successfully launch and land with a high level of certainty.
- 83% Chance of Success
- 94% Precision of the Chance of Success Result

Executive Summary

The outlook: bright!

<u>Bottom Line:</u>

Because we have a high level of certainty that the Stage One Space X Falcon 9 Rocket will launch and land successfully, we can use this information to provide competitive bids for which Space X will have significantly lower costs.

Introduction

Project Background and Context

- Space X Falcon 9 Rocket Launches Failures and Successes
- Looking at success rate based on a large variety of variables.

Project Goal

- The main goal was to find successful launches and landings versus unsuccessful launches/landings for each launch site.
  - The specific variables and parameters that led to successful landings were used to tune the Predictive Models.

Variables:

- Booster Version
- Payload Mass
- Orbit
- Launch Site
- Outcome
- Flights
- Grid Fins
- Reused
- Legs
- Landing Pad
- Block
- Reused Count
- Serial
- Longitude
- Latitude

In order to collect the data, extract
relevant information, then obtain valuable insights,
the following methods were utilized:

- Data Collection with APIs
- Data Wrangling
- Exploratory Data Analysis with Pandas
- Data Visualization
- Folium Maps
- Machine Learning Predictive Analysis
- Predictive Modeling
- SQL Database Queries
- The Confusion Matrix

Summary of Findings:

- With 4 Machine Learning Algorithms utilized for Predictive Modeling, the prediction of landing certainty for the Space X Falcon 9 First Stage was 83%,with a precision of 94% that this prediction was correct and that the first stage has a reasonably high certainty of landing with the tuned hyperparameters.

Section 1

Methodology

Methodology

Executive Summary

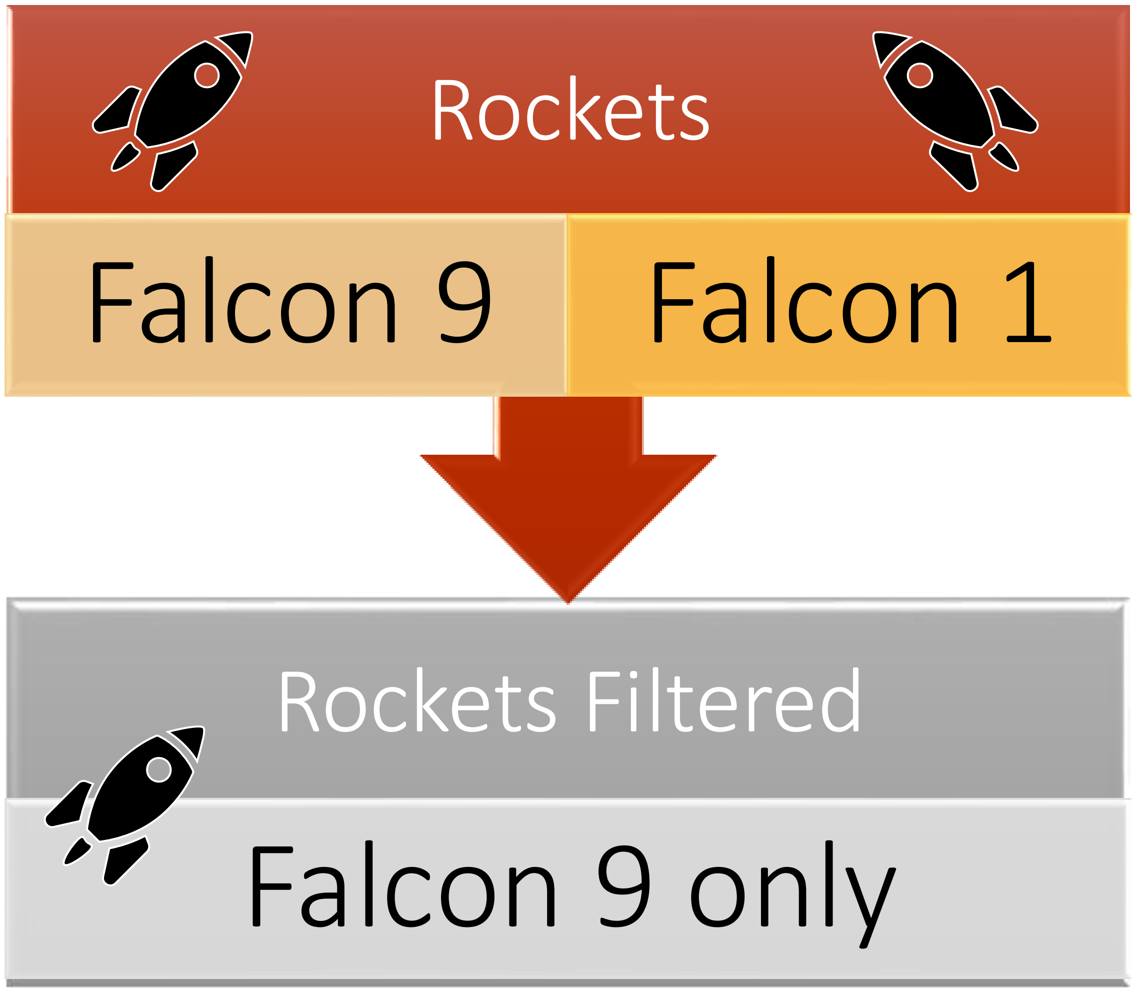
- Data collection methodology:
  - HTTP Requests were made to the SpaceX API.
  - Extracted specific data from the SpaceX API to append to a list for each specific variable.
- Performed data wrangling
  - Created a Pandas DataFrame, then cleaned data that was missing or irrelevant .
- Performed Exploratory Data Analysis (EDA) using visualization and SQL
- Performed interactive visual analytics using Folium and Plotly Dash
- Performed predictive analysis using classification models
  - Built, tuned, and produced Machine Learning Predictive Classification Models.

Data Collection

- Data Sets were collected with an HTTP Request to the SpaceX API.
    - Functions were defined to gather data into lists that would be used to create the columns of a pandas DataFrame.
    - The requests.get function was used to collect the data from SpaceX.
    - The response content was in the form of a JSON file, meaning that it needed to be converted to a pandas DataFrame with .json_normalize()
- The data from each request was prepared to be stored in the list to be used to create the DataFrame.
- After all data was ready, the user-defined functions were called to store the data in the lists.
- The columns were put into a dictionary called "launch_dict" and then was recreated into a pandas DataFrame called "data"

Data Collection

- Because Falcon 9 launches were the only desired pieces of data, the DataFrame filtered for Falcon 9 launches.
- The collected data at this point included the variables that had effect only on Falcon 9 launches.



Data Collection – SpaceX API

Requested Rocket Launch Data from URL

→ Requested and Parsed Information with JSON Static Response Object Request

→ Decoded response object as JSON file.

→ Got subset of Dataframe with features we wanted.

→ Cleaned 'cores' and 'payload' columns

→ Got datetime information and restricted the date of the launches.

→ Prepared all information from API calls to be stored in lists.

Data Collection – Web Scraping

Called Rocket function.

Called Payload function.

Called launchpad function.

Called core function.

Created dictionary of lists created with the API calls.

Created Pandas Dataframe.

Data Collection GitHub URL

https://github.com/PrestonMoore3/IBM-Data-Science-Capstone/blob/637f30df5329d691297f511df4d4ffaa46123b7f/Data%20Collection%20with%20APIs.ipynb

Jupyter Notebook Rendered URL

Jupyter Notebook Viewer (nbviewer.org)

Data Wrangling

- Imported required libraries: pandas and NumPy
- In order to get started wrangling data, the data was analyzed to check for percentages of missing values, which was quite high for the "LandingPad" column.
- Data Types were checked with df.dtypes
- The number of launches per site, per Orbit Type, and per landing outcome were all calculated using the value_counts() method.
- Looped through the data to create a list of unique key : value pairs. This helped show where the types of missions and their successes or failures could be found
- Created an outcomes list for where the stage did not land successfully.

Data Wrangling - Insight

- A Landing Class column was added by creation from the Outcome column.
- The success rate overall was checked by using the .mean() method.

The ratio of Successful landings to failed landings was 0.6666 repeating, meaning a higher rate of success than failure.

Data Wrangling Flow Chart

Imported required libraries: pandas and NumPy

In order to get started wrangling data, the data was analyzed to check for percentages of missing values, which was quite high for the "LandingPad" column.

Data Types were checked with df.dtypes

The number of launches per site, per Orbit Type, and per landing outcome were all calculated using the value_counts() method.

Looped through the data to create a list of unique key : value pairs. This helped show where the types of missions and their successes or failures could be found

Created an outcomes list for where the stage did not land successfully.

A Landing Class column was added by creation from the Outcome column.

The success rate overall was checked by using the .mean() me

Data Wrangling GitHub URL

https://github.com/PrestonMoore3/IBM-Data-Science-Capstone/blob/637f30df5329d691297f511df4d4ffaa46123b7f/Data%20Wrangling.ipynb

Jupyter Notebook Rendered URL

Please download the file to view locally as the notebook cannot be rendered.

EDA with Data Visualization Methodology

- Categorical Plots
  - These show the categories of class, representing whether or not a launch landing was successful or not clearly marked by the hue.
  - The best reason is to see how two variables will correlate together to produce the Success or Failure.
  - Categorical Plots easily visualize multiple pieces of specific data in relation to each axes.
- Bar Charts
  - Made the comparison between multiple variables and the Success Rate Easily Visualized.
- Line Chart
  - The Line Chart showed the yearly trend of Success Rate, because these changed from year to year making it easy to compare how Success Rate Changed over the years.

EDA with SQL Methodology

Summary of SQL Queries

- Found the names of the Unique Launch Sites
- Displayed Launch Sites with 'CCA' beginning the name.
- Selected the Sum of the Payload Mass (kg) from boosters launched by NASA
- Displayed the average payload mass carried by the booster version by averaging the Payload Mass (kg)
- Found the first successful landing date on a ground pad.
- Listed the names of the boosters by selecting the distinct booster with a Successful Landing using drone ships with a Payload Mass (kg) between 4000 and 6000.
- Counted the total of successful and failed outcomes respectively.
- Selected the Booser Versions that could carry the maximum Payload Mass (kg).
- Displayed the failed landings for drone ships, their booster versions, and the corresponding launch sites in 2015.
- Ranked the number of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order to show the specific ordered number of landings.

Folium Interactive Map of Launch Sites Methodology

- Map Objects Utilized on the Folium Map:
  - Map with initial centered location to be the NASA Johnson Space Center in Houston, TX.
  - Circle – showed each launch site area.
  - Marker – showed the location of each launch site.
  - Marker Cluster – helped to simplify the map with multiple sites in one small area, when zoomed out.
  - Popup – provided information about the launch site.
  - MousePosition – provided latitude and longitude coordinates to help add additional map markings.
  - PolyLine - delineated the distances from a launch site to particular map features.

Plotly Dash Dashboard Methodology

Pie Chart – shows the percentage of distribution of success rates for all sites and specific site success ratios.

Scatter Chart - provides clarity as to where clusters of successful launches were to dive deeper into further understanding.

Dropdown – provided easily responsive feedback to explore different launch sites, payloads, and success rates with booster versions.

Slider – provided flexibility with interactive range to discover the best range for payload mass (kg)

All Components - top layer of where to begin further searching.

Individual Launch Site - provided the failure/success rate for each individual launch site.

Predictive Analysis (Classification) Methodology & Flow Chart

Data Standardization using SciKit Learn Preprocessing

Train-Test-Split to create a Training Set and Testing Set of Data.

Tuned hyperparameters to find the best for the machine learning algorithms.

Machine Learning Algorithms Utilized:

- K-Nearest Neighbors (KNN)
- Decision Tree Classifier
- Support Vector Machine
- Logistic Regression

Created model using the training and testing data, then scored the Predictive Model of each algorithm against the actual dataset to determine accuracy of trained model.

Scoring:

- F1 Score for Precision of the Results and the Accuracy of the model

Visualizations: Bar Chart, Confusion Matrix

Confusion Matrix:

- Displayed the effectiveness of the best performing model to predict a landing outcome.

Results Overview

Exploratory data analysis results

Interactive analytics demo in screenshots

Interactive Folium Maps

Predictive analysis results

Conclusions

Section 2

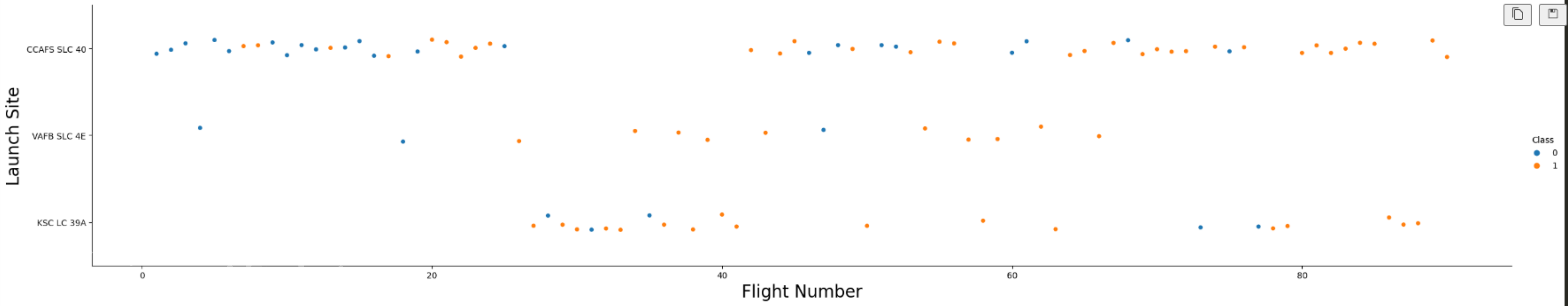
Insights drawn from EDA

Flight Number vs. Launch Site

- Looking at each flight number, for each Launch Site, we can see the classification of successes or failures based on flight number.
- The insights:
  - For All Sites, there were many more successes as flight numbers increased
  - CCAFS SLC-40 Had a larger number of launches, especially with the higher Flight Numbers and the Lower Flight Numbers. This means that the site was revisited after trying flights at the other launch sites.
  - Once the VAFB and KSC sites were utilized with high success, we see an increase in successes with CCAFS.
  - KSC LC-39A had a high level of success in all flight numbers overall.

```python
sns.catplot(x="FlightNumber", y='LaunchSite', hue = 'Class', data=df, aspect = 5)
plt.xlabel('Flight Number', fontsize = 20)
plt.ylabel('Launch Site', fontsize = 20)
plt.show()
```

Python

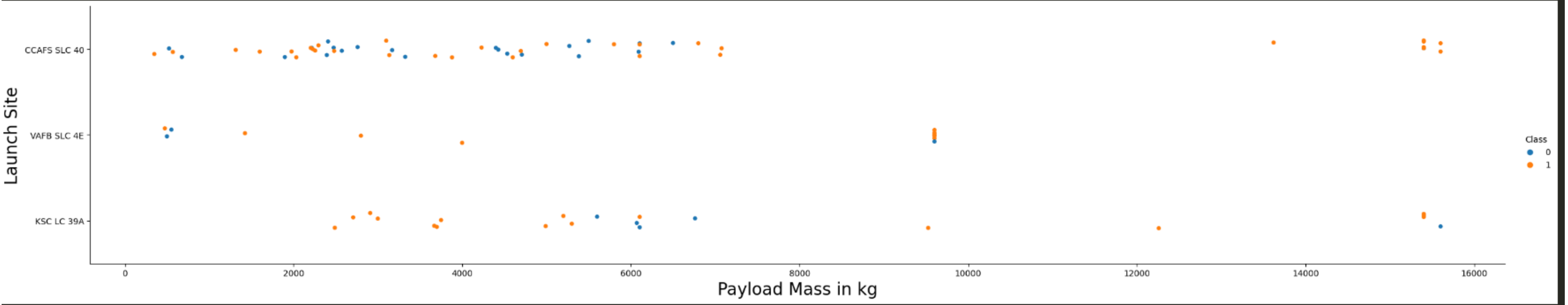


Payload vs. Launch Site

Displayed are the different Payload Mass in kg for each Launch Site, with classification of each launch Payload Mass per site as a success or failure.

- Payload Mass in kg between 2,000 and 6,000 at KSC LC-39A and VAFB SLC-4E provided a high level of success rates.
- Not many launches were completed with 8,000 to 15,000 kg Payload Mass.
- CCAFS SLC-40 had a 100% success rate in the 13,500 to 16,000 kg Payload Mass.
- CCAFS SLC-40 had mixed success with Payload Mass below 8,000 kg.

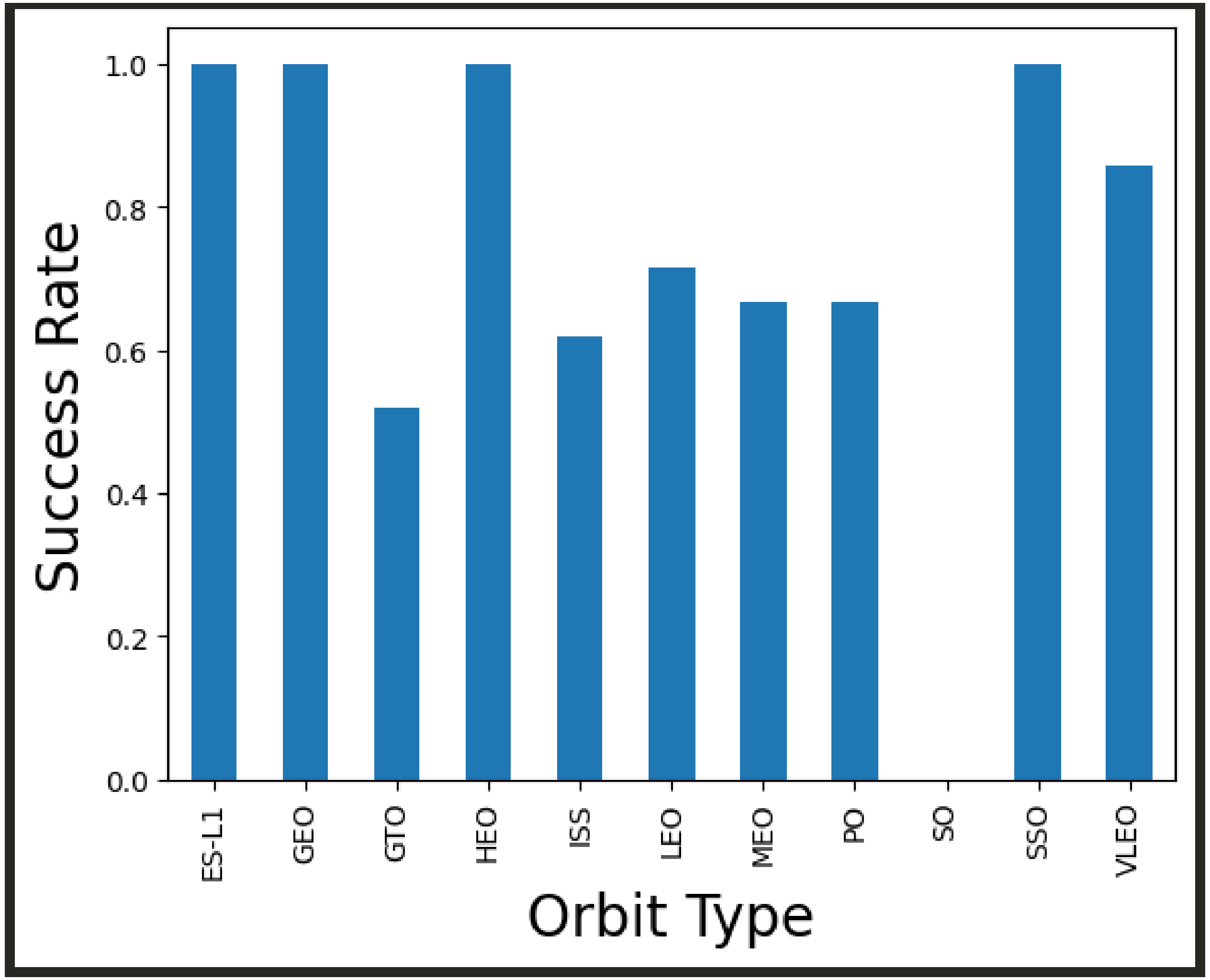
```
sns.catplot(x='PayloadMass', y= 'LaunchSite', data = df, hue = 'Class', aspect=5)
plt.xlabel('Payload Mass in kg', fontsize=20)
plt.ylabel('Launch Site', fontsize=20)
plt.show()
```

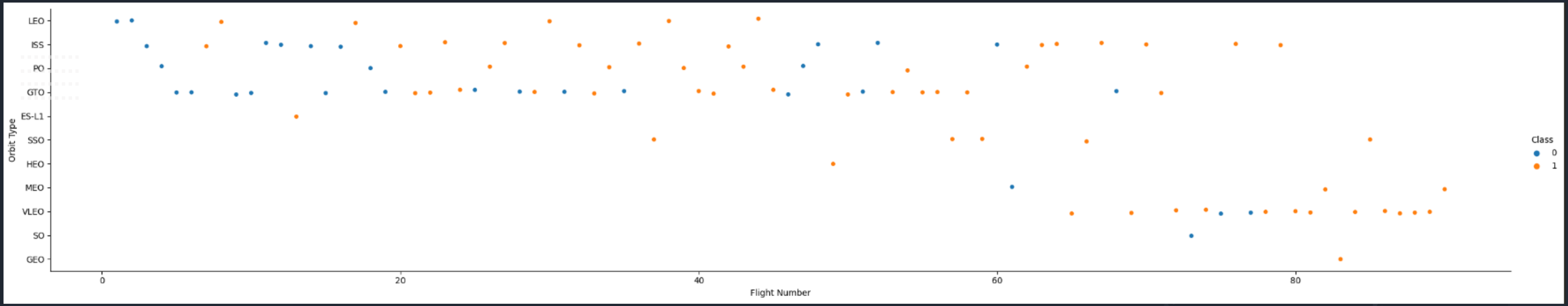


Success Rate vs. Orbit Type

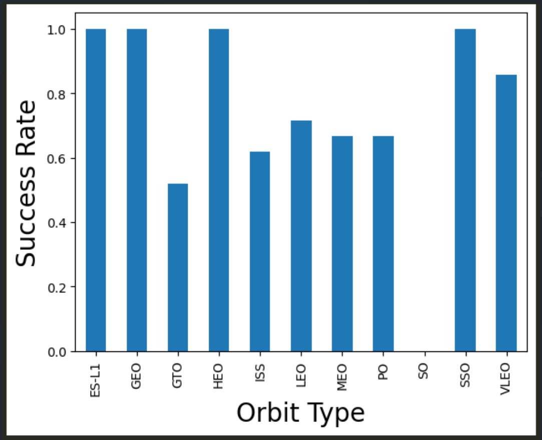
The success rate by orbit type is compared between each type of Orbit.

- Insights:
  - ES-L1, GEO, HEO, and SSO had the highest success Rate of 100%
    - One of these Orbit Types should be picked in order to guarantee the highest level of Launch Success Rate.
    - More insight can be gained by looking at the number of launches within this group, to show the most reliable Orbit type in terms of number of launches with 100% Success.
  - GTO, ISS, LEO, MEO, and PO show moderate success rates.
  - SO had 0% success.
  - VLEO had a high level of success but not 100%





Flight Number vs. Orbit Type



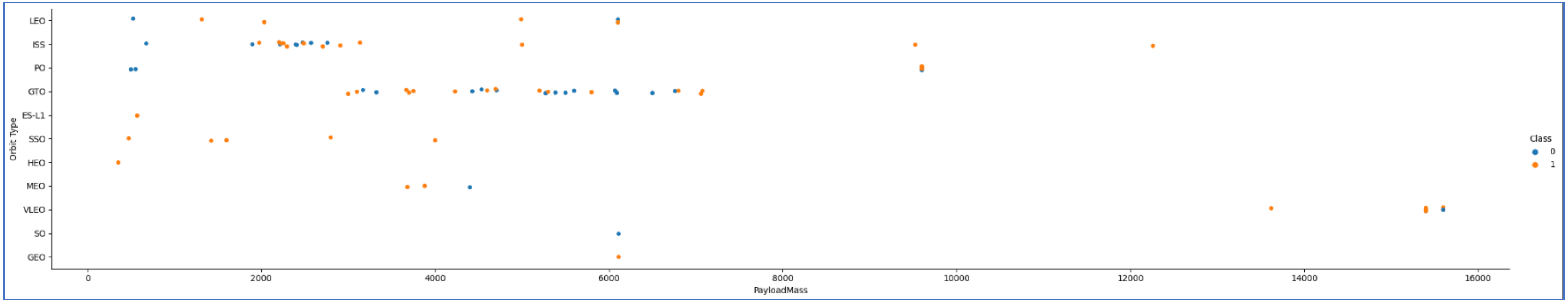
*Reference for success rate of orbit type

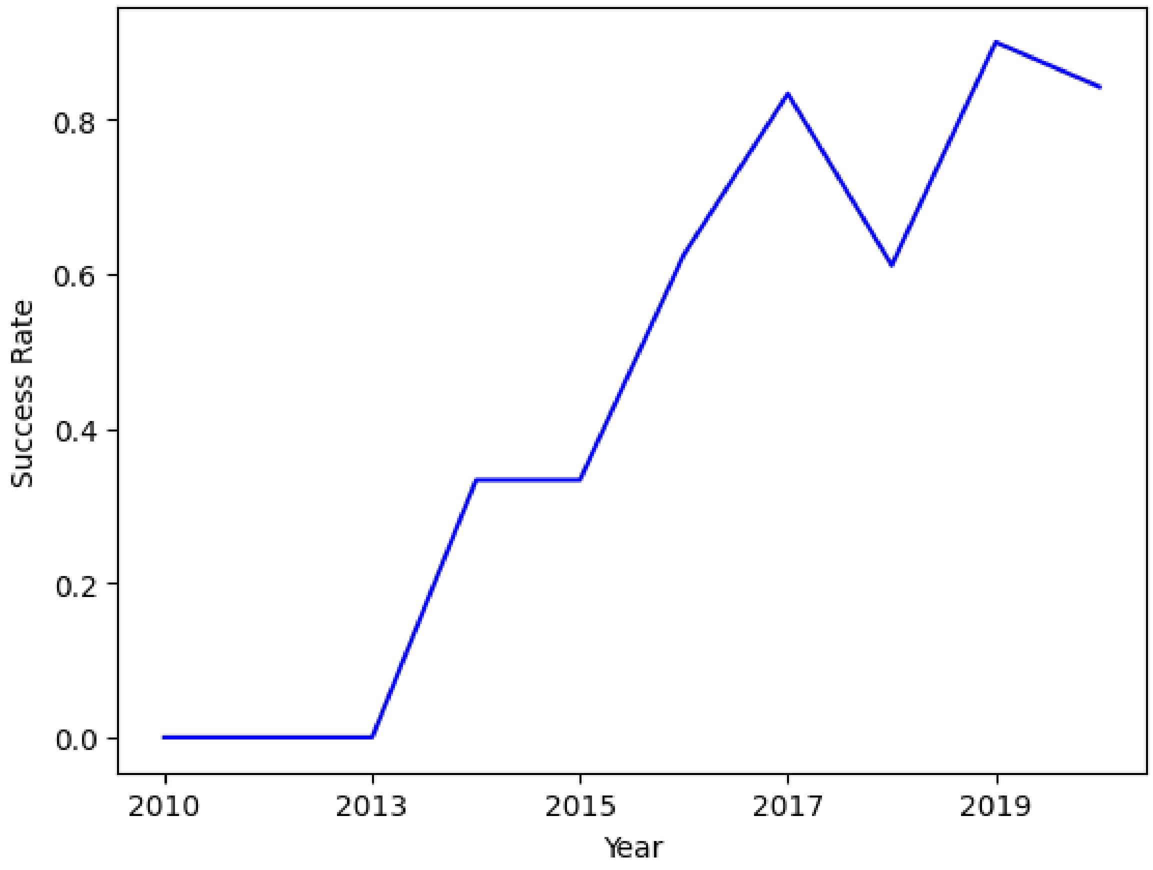
Insights:

- SSO had 100% success with five launches, making it a good candidate for Orbit Type Choice.
- VLEO had a high number of successes and was tested multiple times with the most recent flight numbers – this is where in previous testing, results were refined when looking at Flight Number vs. Launch Site. This site would make an excellent choice for Orbit type because of the high amount of data in agreement with 100% success in those later launches: even more so than SSO.
- GEO, ES-L1 and HEO had 100% success but only one launch, which does not provide enough data.
- ISS Had many failures at the beginning of the flights, but after learning from other launches, this Orbit type then began to see high levels of success from flights 60-80.
  - With it's medium level of success in Flights 20-40 and 60-80 this would also make a good secondary choice to VLEO.

Payload vs. Orbit Type

- SSO showed 5 launches 100% successful within the low Payload range of 0kg-4000kg.
- ISS had mixed results in lower Payload Mass but increased its success with Payload Mass increased passed 3500kg.
- VLEO was only tested at a high payload mass above 13,000 kg, whereas the highest level of successful launches were in lower Payload Mass ranges between 2,000 and 8,000kg.





Launch Success Yearly Trend

- We see the success rate climbing as the years go on.
- 2019 Had the highest success rate.
- 2010 to 2013 had the lowest rates of success.
- The 2017-2019 range showed a dip in success but still a high success rate.
    - **We should look at what variables changed to determine what lowered the success rate.**

Data Visualization GitHub URL

https://github.com/PrestonMoore3/IBM-Data-Science-Capstone/blob/637f30df5329d691297f511df4d4ffaa46123b7f/Exploratory%20Data%20Analysis%20and%20Data%20Visualization.ipynb

Jupyter Notebook Rendered URL

https://nbviewer.org/github/PrestonMoore3/IBM-Data-Science-Capstone/blob/5c48e5f4c5f513c381153b18240b3162e15e6426/Exploratory%20Data%20Analysis%20and%20Data%20Visualization.ipynb

All Launch Site Names

- Unique Launch Sites Displayed: 4
- Explanation: names of the four unique launch sites.

History

Results

Result set 1

Details

Filter table

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

DATE	TIME__UTC_	BOOSTER_VERSION	LAUNCH_SITE	PAYLOAD	PAYLOAD_MASS__KG_	ORBIT	CUSTOMER	MISSION_OUTCOME	LANDING_OUTCOME
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

5 Launch Site Names that Begin with 'CCA'

The CCAFS LC-40 Launch Site shows 5 different Payloads, with 5 different Payload Masses (in kg).

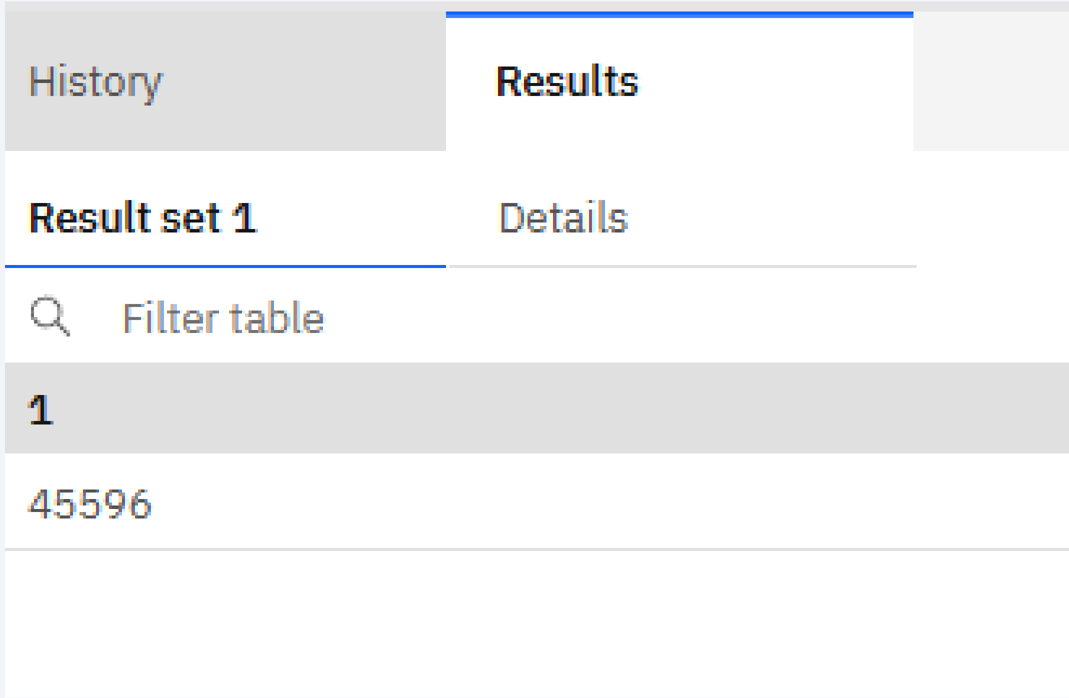
The Orbit Type was LEO for all 5 launches.

The Booster Version was F9 v.1.0 for all 5 launches.

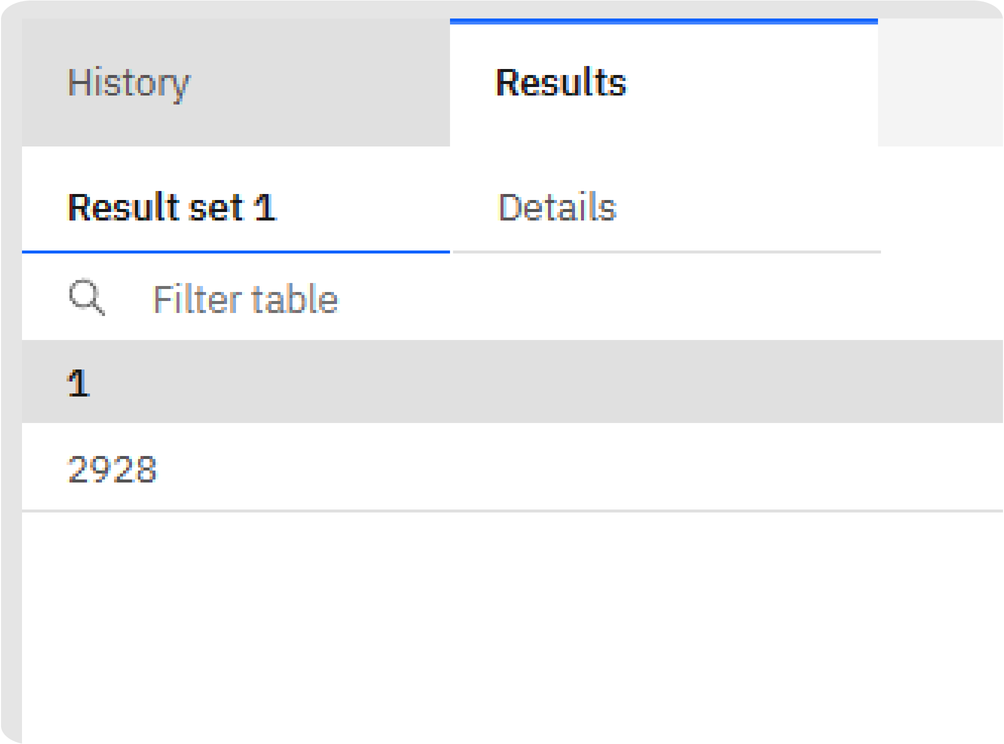
All 5 launches were a success but the Landings were failures or not attempted, meaning the Stage 1 was not recovered.

Total Payload Mass

- The total payload carried by boosters from NASA was 45,596 kg
- This combined all booster versions and payloads for NASA Specifically.



Average Payload Mass by F9 v1.1



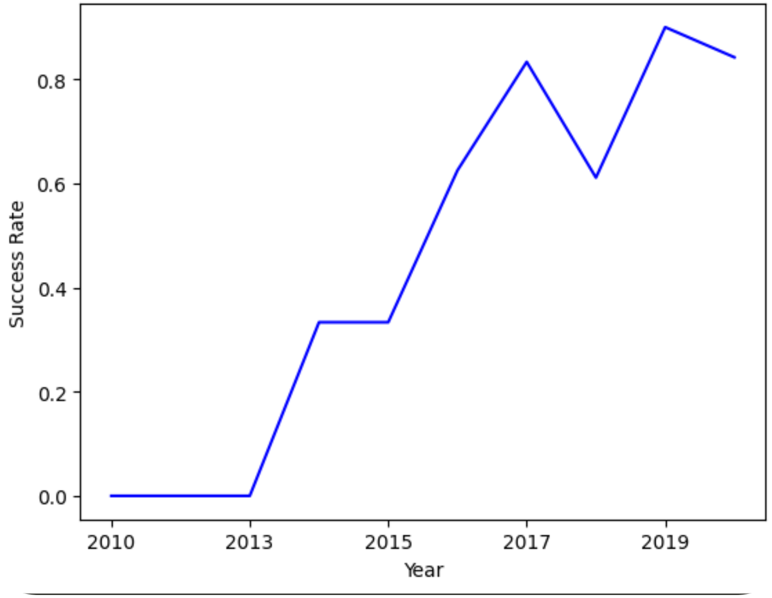
- The average payload mass carried by booster version F9 v1.1: 2,928 kg
- This showed the Booster Version's Average Payload Mass, which was within the range of where the most successful launches occurred overall.

History Results

Result set 1 Details

Filter table

1
2015-12-22



First Successful Ground Landing Date

- The Date of the first successful landing outcome on ground pad: 12/22/2015.
- Insight: From 2015 and onward, on average the success rate increased, meaning ground pad could have been a factor in that increase.
  - *Reference: Launch Success Yearly Trend

Successful Drone Ship Landing with Payload between 4000kg and 6000 kg

- The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 is displayed.

- The Booster Version was FT B1 in all four cases of Drone Ship Landing with Payload between 4000kg and 6000kg

History Results

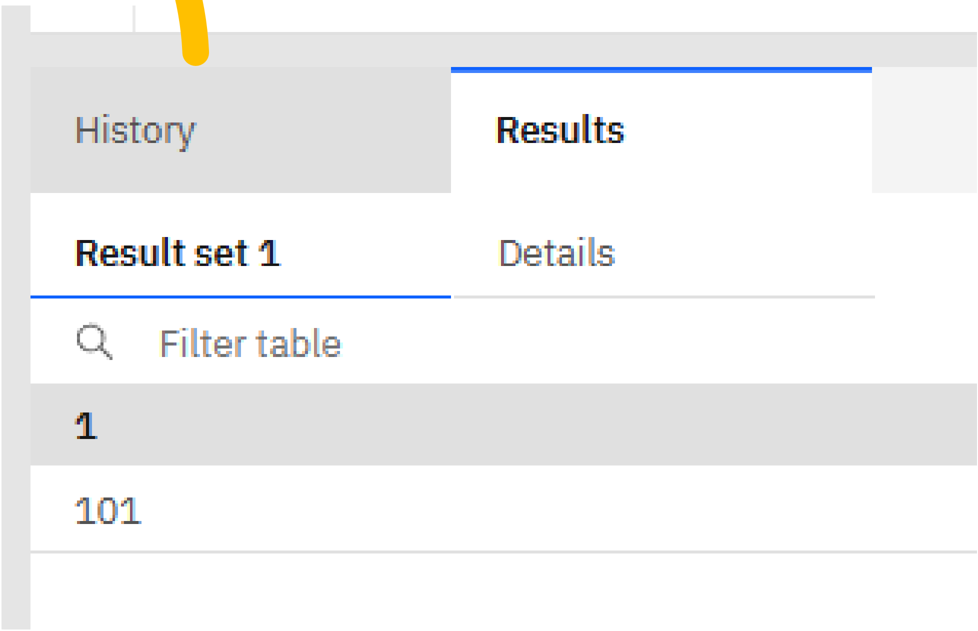
Result set 1 Details

Filter table

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

Total Number of Successful and Failure Mission Outcomes

- The result shows the total number of successful and failure mission outcomes
- The 101 mission outcomes had a success ratio of 0.6666 with a high number of missions.
    - *Reference: Data Wrangling



History | Results

Result set 1 | Details

Filter table

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

Boosters that Carried Maximum Payload

- The listed names of the booster which have carried the maximum payload mass is displayed.
- The Booster Version was B5 with slight variations in all cases.
- Booster Version B5 had a low rate of success because the Maximum Payload in Kg also had a low rate of success.
    - *Reference: Payload Mass (kg) Dashboard Visualization, and Scatter Plots of Payload Mass (kg) success rates.

LANDING_OUTCOME	BOOSTER_VERSION	LAUNCH_SITE	DATE
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015-10-01
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015-04-14

2015 Launch Records

- Displayed are the Failed Landings in drone ship, the booster versions, and launch site names for the year 2015.

- The booster version and launch site were consistent: Booster Version v.1.1 and CCAFS LC-40.
  - Drone ship failed landings in 2015 had a correlation with Booster Version v 1.1 and the CCAFS LC-40 Launch Site.
  - For future success, the launch site and/or Booster Version would need to be changed to see if there could be success with drone ship landings.

History	Results

Result set 1	Details

Filter table Total:8

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

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

- Displayed are the ranked counts of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order.
- The highest landing outcome counts of failure and success were both drone ship, with a 50/50 split between success and failure.
- The highest success rate with **conclusive evidence is ground pad**, where the count was 5 successes with 0 failures displayed.

EDA with SQL GitHub URL

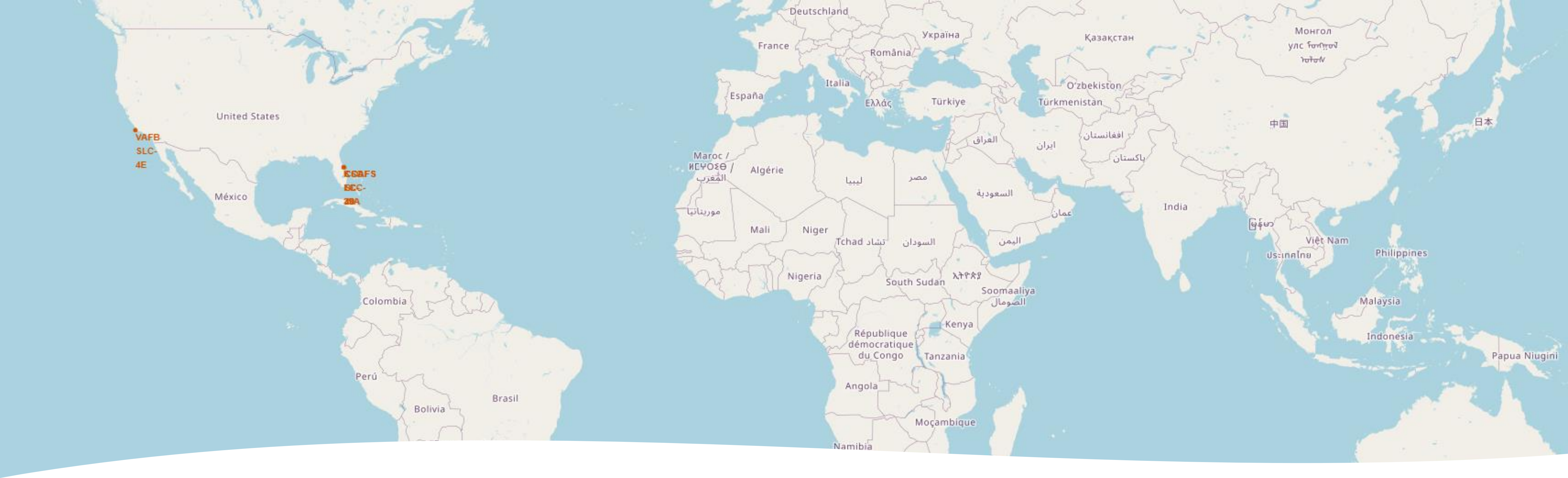
https://github.com/PrestonMoore3/IBM-Data-Science-Capstone/blob/85f3347b82e2b01010b1fb320fe703619b771523/jupyter-labs-eda-sql-coursera_sqllite%20-%202nd%20version.ipynb

Jupyter Notebook Rendered URL

Please download locally to view.

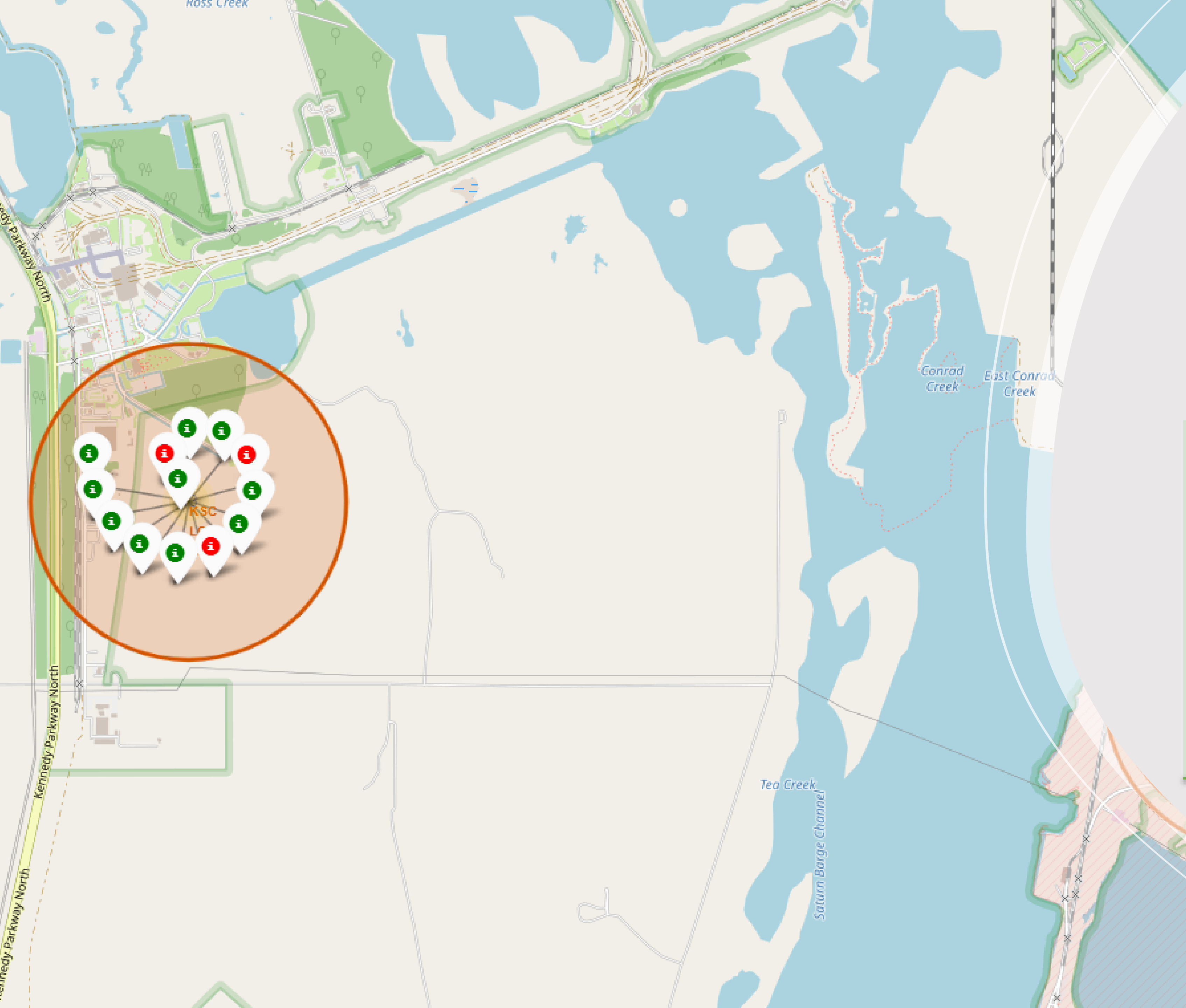
Section 3

Launch Sites Proximities Analysis



Falcon 9 Unique Launch Sites

- All of the Launch Sites were located in the United States of America.
- All Launches were near a coastline.
- All launches were in the Southern United States.
- All launch sites were somewhat near the equator, north of the equator about the same amount.



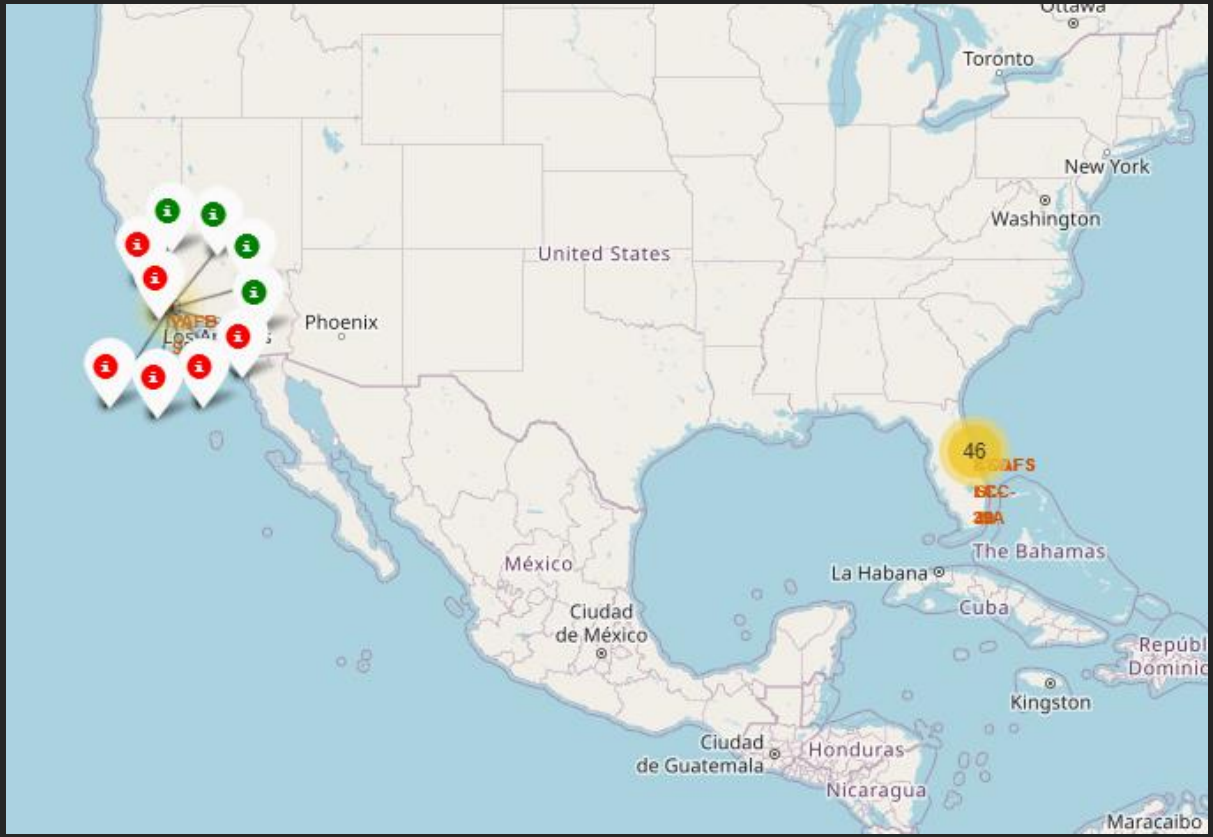
Launch Outcomes by Launch Site

1

Site: KSC LCA-39A

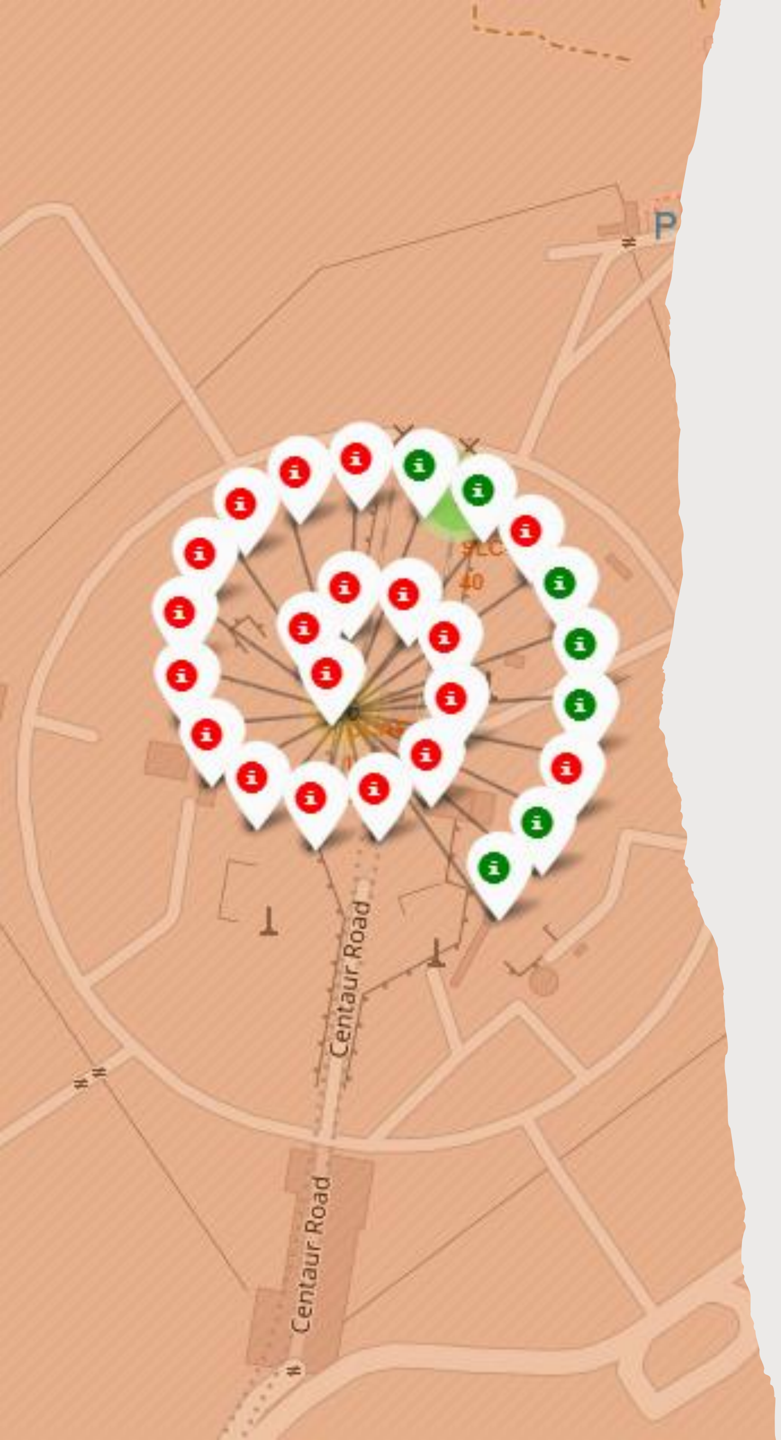
2

We see a <u>high</u> percentage of success ratio to failure at this site.



Launch Outcomes by Launch Site

1. Launch Site: VAFB **SLC-4E**
2. A **Low** success ratio



Launch Outcomes by Launch Site

1

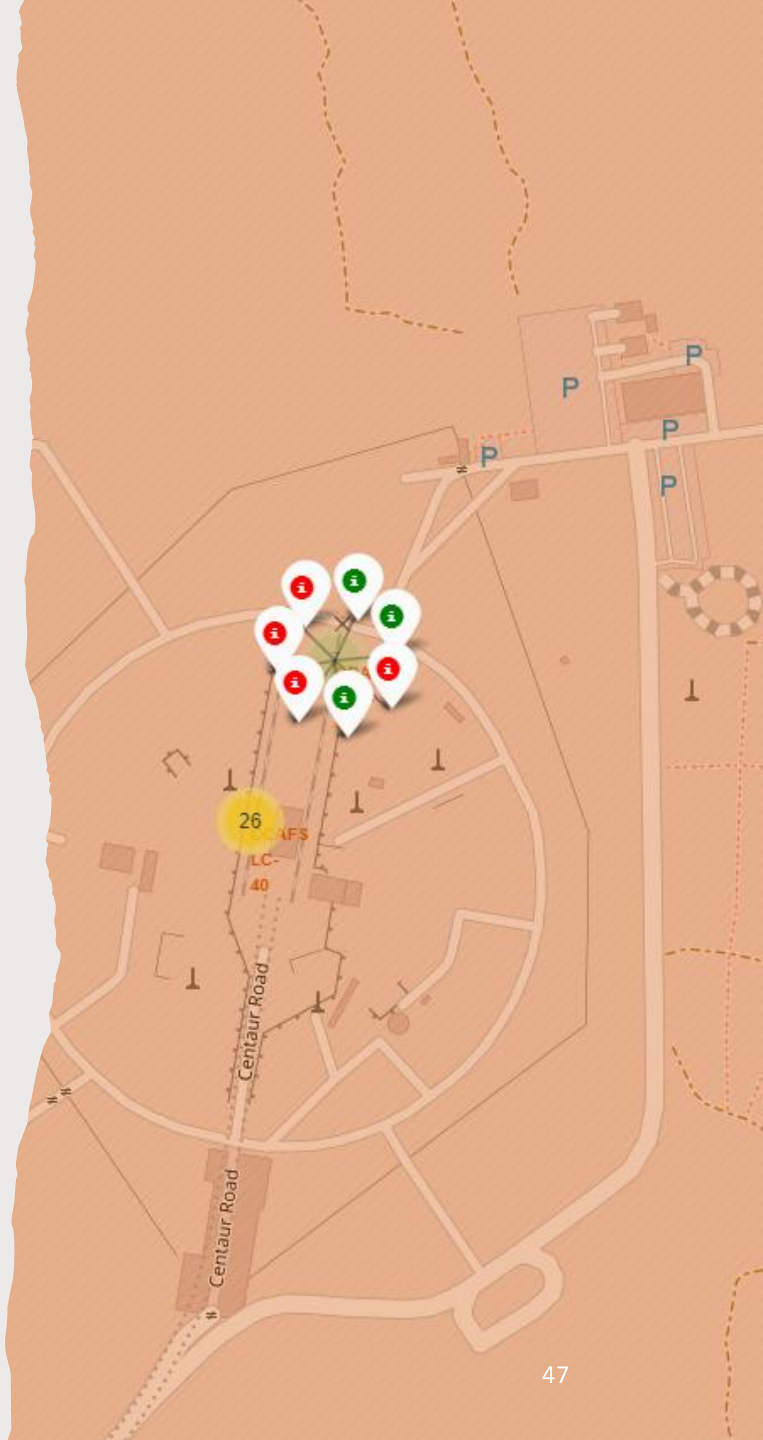
CCAFS SLC-40

- Low Success Ratio

2

CCAFS LC-40

- Moderately Low Success Ratio
- Lower number of launches



CCAFS LC-40 Launch Site in Proximity to a Close Railway, Highway, Coastline, and City

Findings:

- Proximity to Coastline, Railway, and Highway were all very close.
- Proximity to a City was the farthest distance.
- There are multiple coastlines in each direction very close to the Launch Site.

Folium Maps GitHub URL

https://github.com/PrestonMoore3/IBM-Data-Science-Capstone/blob/637f30df5329d691297f511df4d4ffaa46123b7f/Launch%20Sites%20Location%20Analysis%20with%20Folium.ipynb

Jupyter Notebook Rendered URL

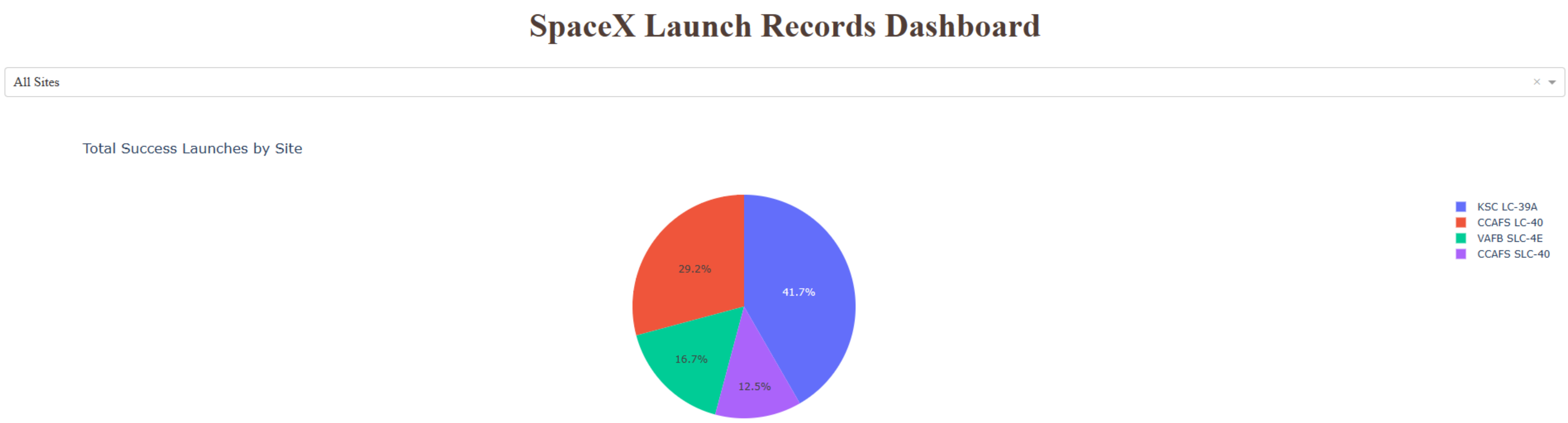
https://nbviewer.org/github/PrestonMoore3/IBM-Data-Science-Capstone/blob/99cab735536b0dee6616ebc8344d70e18dcf8368/Launch%20Sites%20Location%20Analysis%20with%20Folium.ipynb

Section 4

Build a Dashboard with Plotly Dash

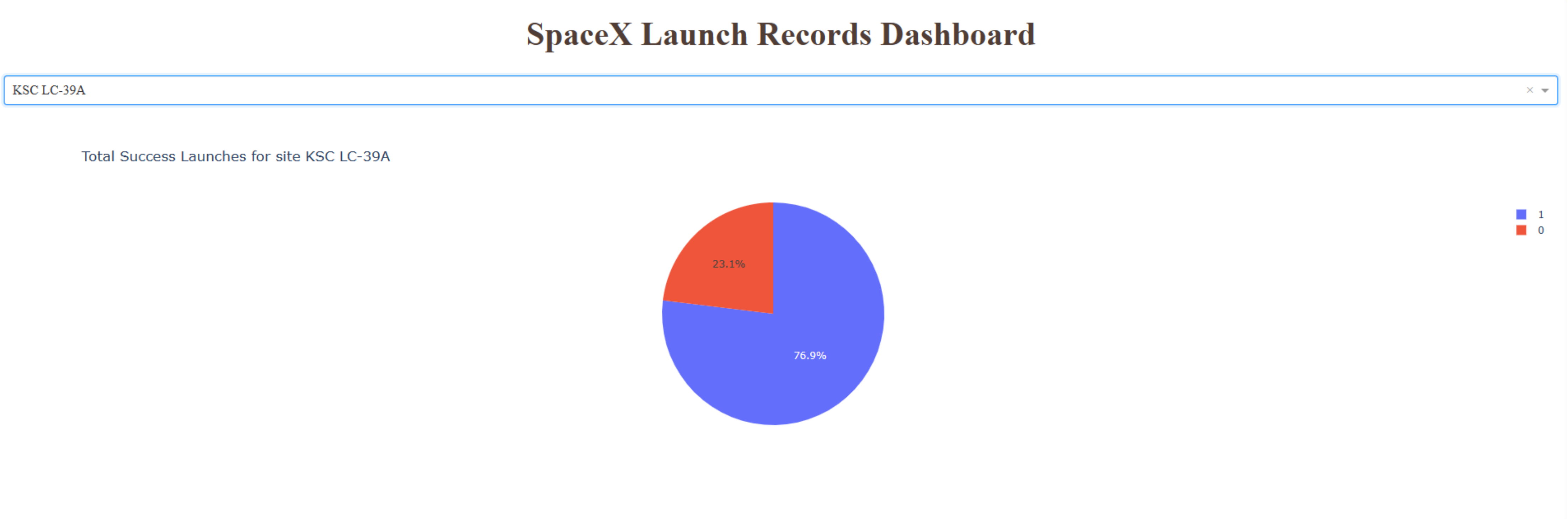
Total Successful Launches by Site

- Looking at all sites, we see the highest success rate based on Launch Site was KSC- LC-39A with 41.7% of the successful landings in comparison to lower percentages of overall success rate between each of the other 3 Launch Sites.
- This highlights KSC LC-39A as the launch site promoting the most amount of success and will be targeted for the suggested launch site due to the success rate.
- This is a preliminary finding that requires further discovery since it only shows the number of successes, not the ratio of successes within the launch site itself.



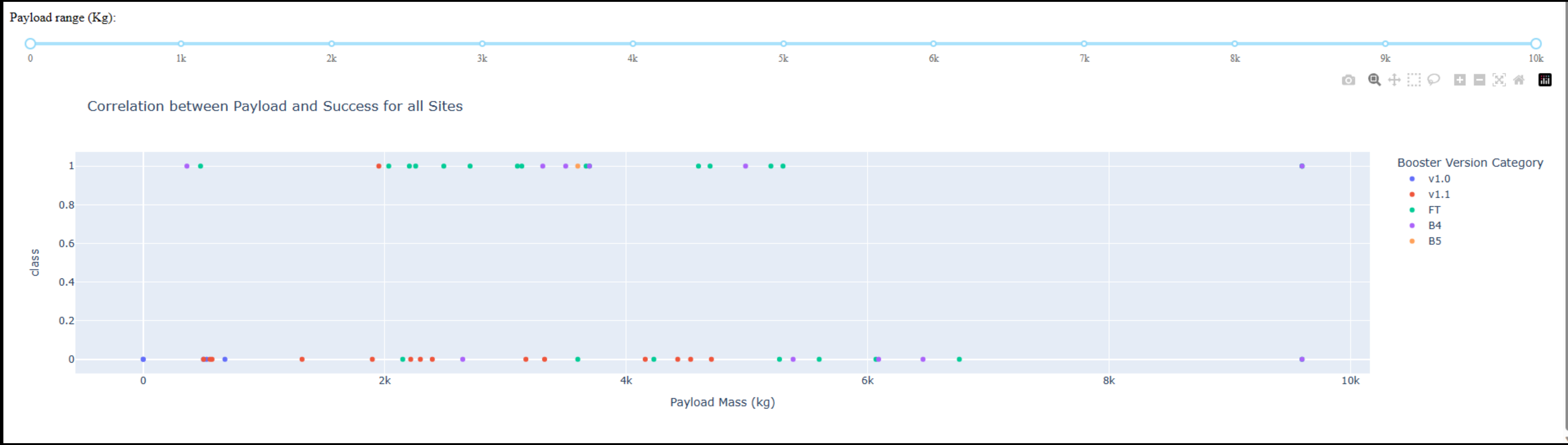
Lauch Site with Highest Launch Success Ratio

- Since KSC LC-39A has the highest number of successes, we want to focus on its percentage of success versus failure ratio, rather than sole number of successes.
- When looking at the total success launch ratio, we see a 76.9% success rate for all launches at KSC LC-39A.
- We looked at all percentages of launch site successes, with KSC LC-39A proving the highest rate of successful landings versus unsuccessful landings.



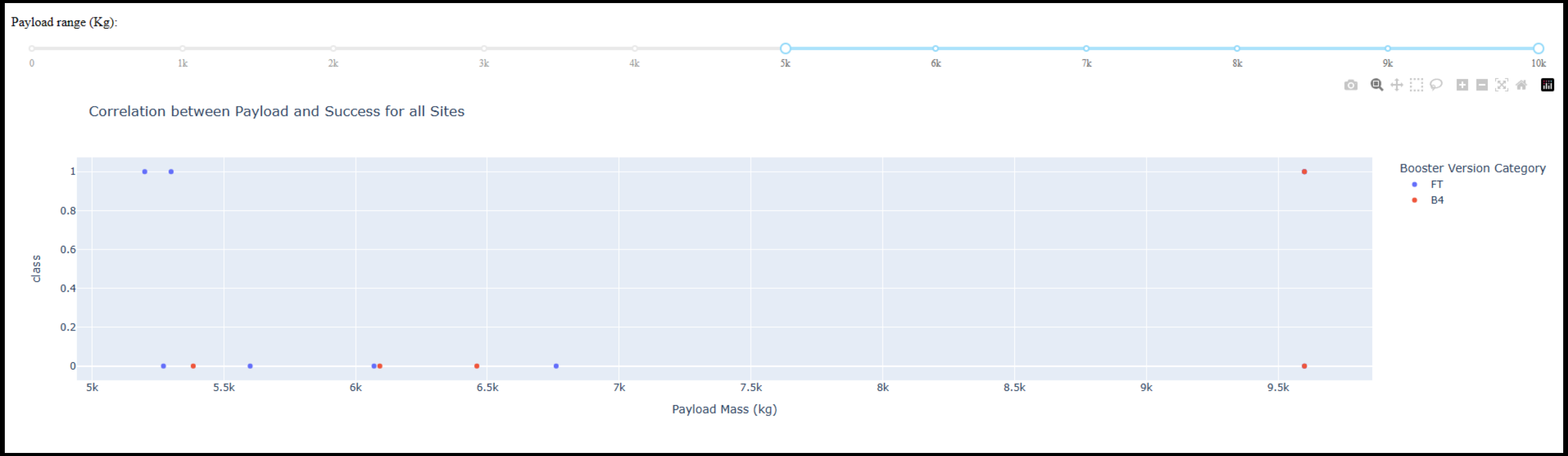
Payload vs. Launch Outcomes for All Sites

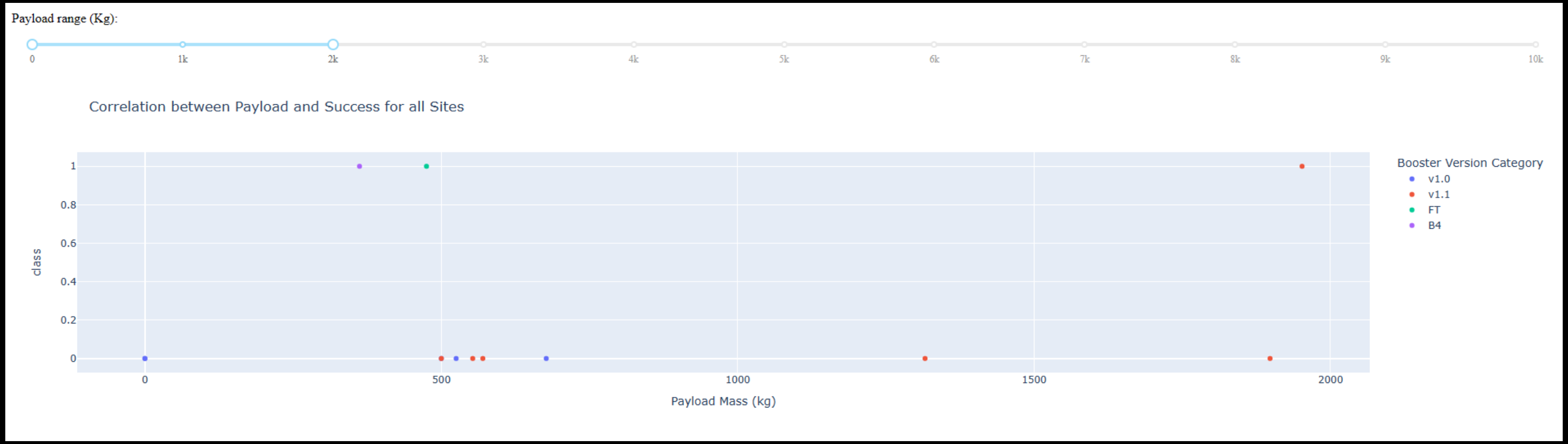
- High Success Rates Overall
  - Booster Version Category: FT
  - Booster Version Category Runner-Up: B4
  - Payload Mass (kg) within the 1k-6k (kg) range
- LOW
  - v. 1.1 in all Payload Mass ranges.
  - High Payload Mass (kg) with Booster Version B4 and FT above 6000kg



Payload vs. Launch Outcomes for All Sites

The slider here is used to focus in solely on high Payload Mass (5000kg to 10,000kg) and low Payload Mass (0 kg to 2000kg). We see low levels of success for launch outcomes in both of these ranges.

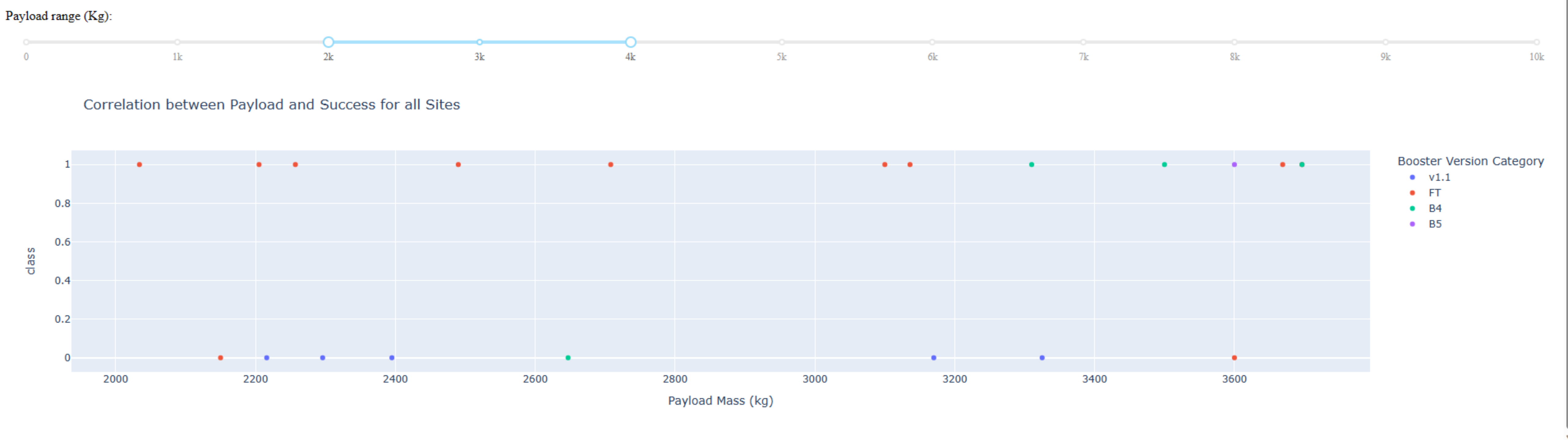




Payload vs. Launch Outcomes for All Sites

Insight:

This Payload Mass range of 2000kg to 4000kg showed the **ideal range** of successful launches with the highest successful Booster Version Category of FT.

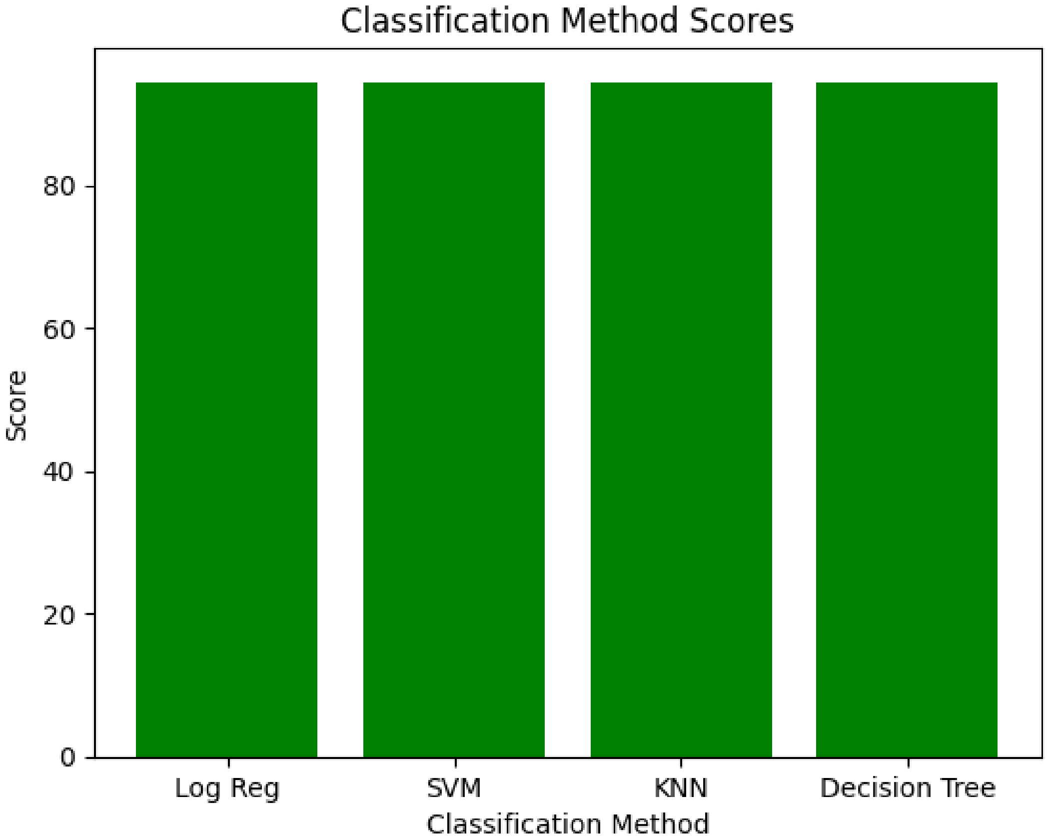


Plotly Dashboard GitHub URL

https://github.com/PrestonMoore3/IBM-Data-Science-Capstone/blob/637f30df5329d691297f511df4d4ffaa46123b7f/Spacex_Data%20Plotly%20Dashboard.py

Section 5

Predictive Analysis (Classification)



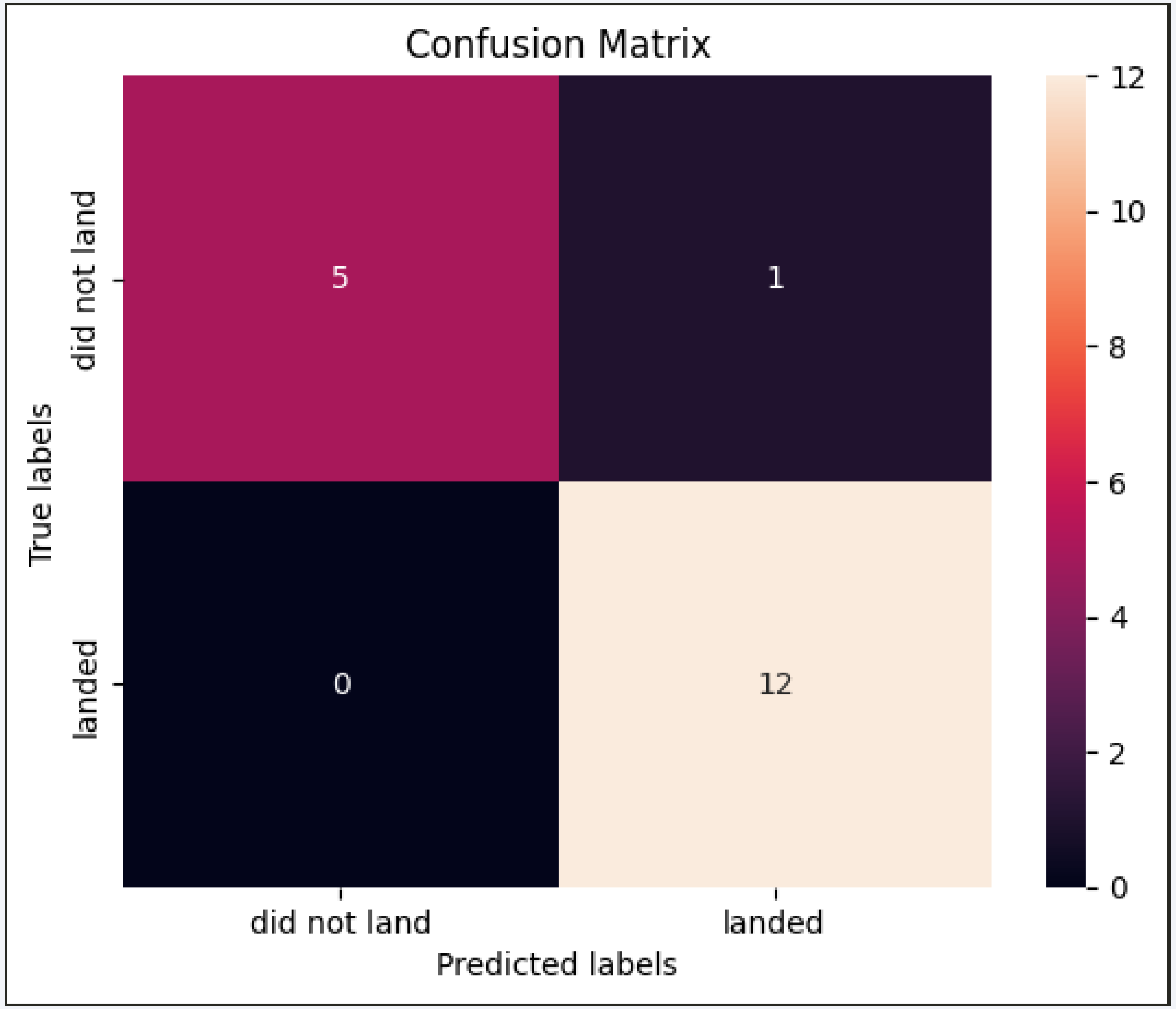
	precision	recall	f1-score	support
0	1.00	0.83	0.91	6
1	0.92	1.00	0.96	12
accuracy			0.94	18
macro avg	0.96	0.92	0.93	18
weighted avg	0.95	0.94	0.94	18

Classification Accuracy

- The classification method scores highlight that each machine learning algorithm reflected a predicted accuracy of 94.4% of the success ratio being 83.3%.
- Each algorithm produced almost the exact same F1 Score, showing equally that looking at the predicted landing, we can predict with high accuracy that the First Stage will likely land.

Confusion Matrix

- This Confusion matrix is taken from the K-Nearest Neighbors Method, which also is equal to the other three methods.
- The model's prediction was highly accurate, especially when predicting 12/13 landings correctly, and 5/5 non-landings correctly.
- We see that the model will accurately predict a classification of landing or not landing based on the data supplied.



Machine Learning GitHub URL

https://github.com/PrestonMoore3/IBM-Data-Science-Capstone/blob/bf964f516931d524f4bafefc273526fd1a1b2f2f/SpaceX%20Machine%20Learning%20Prediction%20Part%205.ipynb

Jupyter Notebook Rendered URL

Please download locally to view notebook.

Conclusions

- The most **successful** launch site was **KSC LC-39A.**
- The ideal launch site has proximity to multiple coastlines on the Atlantic Ocean with a latitude/long of approximately 28.57 and –80.6 respectively, with a larger distance between the launch site and a city.
- The Payload Mass in kg should be **2,000kg to 6,000kg** to **maximize** the success rate.
- The Orbit Type **SSO** had 100% Successful Launches, all within the Payload Mass (kg) maximal success rate range, making it the suggested Orbit Type to implement.
- The Booster Version within the Payload Mass range was **FT** with the highest level of successful Launch Outcomes making the **suggested Booster Version FT.**
- The highest **success rate** with conclusive evidence for Landing Outcomes **is ground pad.**
- **There is a high level of certainty using the Predictive Modeling that the ideal conditions will create a successful landing.**

Appendix

Code Snippet

- The code here put together a dictionary using pre-defined user functions to gather all of the information from the SpaceX Dataset, cleaned, to then make a dataframe of all of the launches, then filtered for only Falcon 9 Launches.

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

Then, we need to create a Pandas data frame from the dictionary launch_dict.

```
# Create a data from launch_dict
data = pd.DataFrame(launch_dict)
```

Show the summary of the dataframe

```
# Show the head of the dataframe
data.head()
```

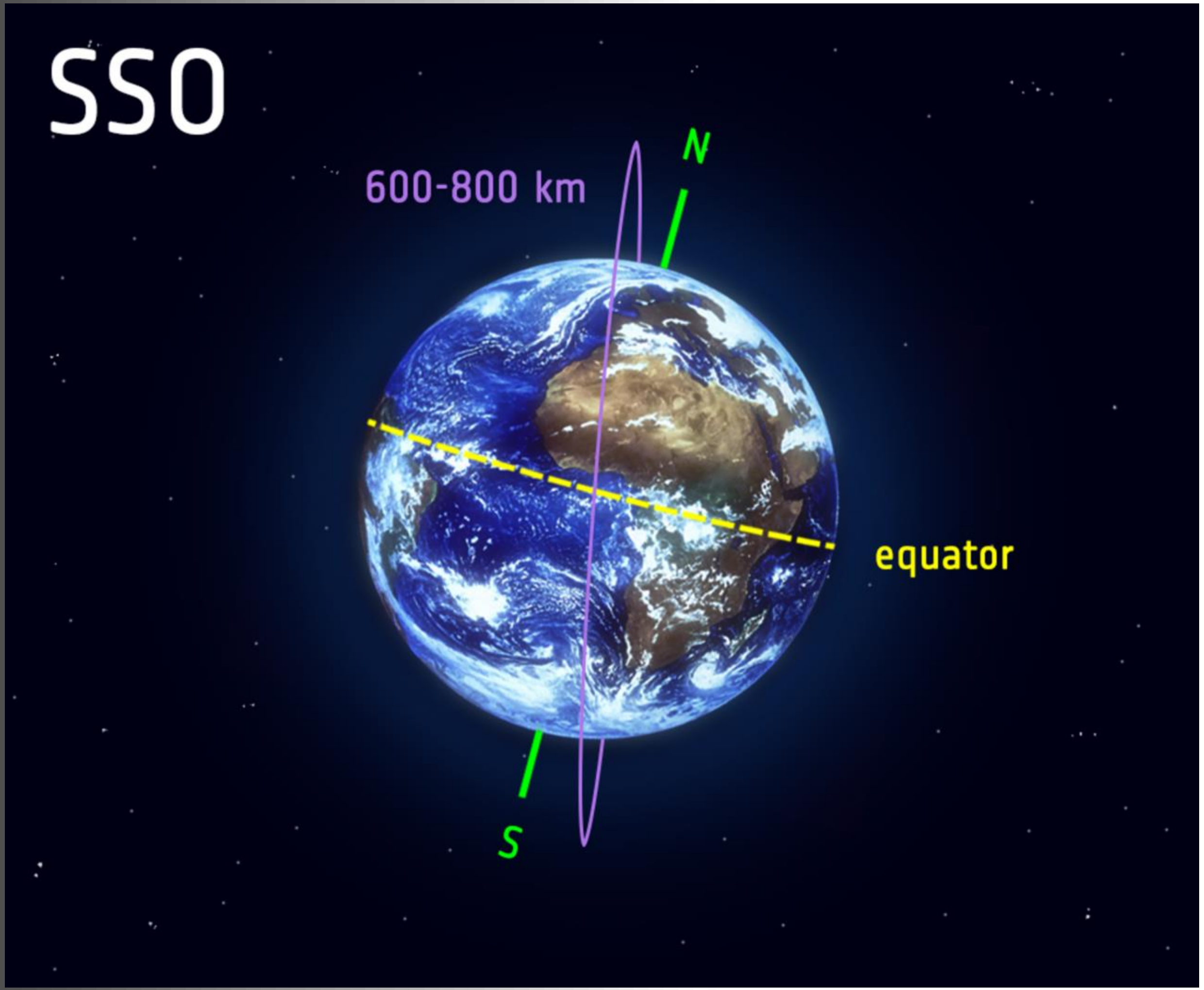
```
data_falcon9 = data[data.BoosterVersion != 'Falcon 1']
data_falcon9
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	
5	8	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	
6	10	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	
7	11	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	
8	12	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	
...	...	...	...	...	...	...	...	...	...	...	...	
89	102	2020-09-03	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	2	True	True	True	5e9e3032383ecl
90	103	2020-10-06	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	3	True	True	True	5e9e3032383ecl
91	104	2020-10-18	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	6	True	True	True	5e9e3032383ecl
92	105	2020-10-24	Falcon 9	15600.0	VLEO	CCSFS SLC 40	True ASDS	3	True	True	True	5e9e3033383ec
93	106	2020-11-05	Falcon 9	3681.0	MEO	CCSFS SLC 40	True ASDS	1	True	False	True	5e9e3032383ecl

90 rows × 17 columns

Suggested Orbit Type

SSO (or SO): It is a Sun-synchronous orbit--also called a helio-synchronous orbit--is a nearly polar orbit around a planet, in which the satellite passes over any given point of the planet's surface at the same local mean solar time [4] .



© ESA – L. Boldt-Christmas

Creating the Landing Classifications for Successful or Failed Landings

```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```

```
0 True ASDS
1 None None
2 True RTLS
3 False ASDS
4 True Ocean
5 False Ocean
6 None ASDS
7 False RTLS
```

We create a set of outcomes where the second stage did not land successfully:

```
# look at the data we just pulled: see the false landings or None: 1, 3, 5, 6, 7

bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])

# call the variable bad_outcomes, which is now a set of unique values based on calling the index alias for landing_outcomes,
bad_outcomes
```

```
{'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
```

```
landing_class=list()

for outcome in df.Outcome:
    if outcome in bad_outcomes:
        #print(i, outcome)
        landing_class.append(0)
    else:
        #print(i, outcome)
        landing_class.append(1)

landing_class
```

Added the Class Column to the Dataframe

```
df['Class']=landing_class
df[['Class']].head(8)
```

	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

Feature Engineering

```
features = df[['FlightNumber', 'PayloadMass', 'Orbit', 'LaunchSite', 'Flights', 'GridFins', 'Reused', 'Legs', 'LandingPad',
features.head()
```

	FlightNumber	PayloadMass	Orbit	LaunchSite	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial
0	1	6104.959412	LEO	CCAFS SLC 40	1	False	False	False	NaN	1.0	0	B0003
1	2	525.000000	LEO	CCAFS SLC 40	1	False	False	False	NaN	1.0	0	B0005
2	3	677.000000	ISS	CCAFS SLC 40	1	False	False	False	NaN	1.0	0	B0007
3	4	500.000000	PO	VAFB SLC 4E	1	False	False	False	NaN	1.0	0	B1003
4	5	3170.000000	GTO	CCAFS SLC 40	1	False	False	False	NaN	1.0	0	B1004

```
features_one_hot=pd.get_dummies(features, prefix=None, columns = ['Orbit', 'LaunchSite', 'LandingPad', 'Serial'] )

features_one_hot.head()
```

	FlightNumber	PayloadMass	Flights	GridFins	Reused	Legs	Block	ReusedCount	Orbit_ES-L1	Orbit_GEO	...	Serial_B1048	Serial_B10
0	1	6104.959412	1	False	False	False	1.0	0	0	0	...	0	
1	2	525.000000	1	False	False	False	1.0	0	0	0	...	0	
2	3	677.000000	1	False	False	False	1.0	0	0	0	...	0	
3	4	500.000000	1	False	False	False	1.0	0	0	0	...	0	
4	5	3170.000000	1	False	False	False	1.0	0	0	0	...	0	

5 rows × 80 columns

Now that our `features_one_hot` dataframe only contains numbers cast the entire dataframe to variable type `float64`

```
# HINT: use astype function

features_one_hot=features_one_hot.astype('float64')
features_one_hot.dtypes
```

```
FlightNumber    float64
PayloadMass     float64
Flights         float64
GridFins        float64
Reused          float64
                 ...
Serial_B1056    float64
Serial_B1058    float64
Serial_B1059    float64
Serial_B1060    float64
Serial_B1062    float64
Length: 80, dtype: object
```

Thank you!