IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Presentation by Nick Brunswick
<Date>

Outline

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

Executive Summary

- Summary of methodologies
  - Using Python with supplementary libraries to collect and process data from the public SpaceX api.
  - Pandas was used primarily to modify and transform data and create the ‘Class’ column which was the target variable to predict whether a landing was successful or not.
  - MatplotLib and Seaborn were used to visualize data in order to perform exploratory data analysis (EDA). SQL was also used to this end.
  - Sklearn was used to build and train machine learning models.
  - Grid Search was used to determine best parameters for machine learning models.
- Summary of all results
  - The most relevant factor in landing success appears to be number of launches, as experience taught the Spacex team a lot.
  - Other factors such as the booster version and the payload mass can also be factored in to predict whether a landing will be successful, though these also go hand in hand with development and experience.

Introduction

- Project background and context
  - SpaceX is a aerospace engineering company that designs, manufactures, and launches rockets and spacecraft. The company was founded in 2002 with the mission to "revolutionize space technology"
  - If a competitor were to enter the market, they would want to learn from SpaceX's history. We analyze the data to determine what the best predictors for success are.
- Problems you want to find answers to
  - What are the major determiners of successful launches and landings?
  - What are the major determiners of failed launches and landings?
  - Can we predict a cost of operating such a business?
  - What unseen patterns exist in the data?

Section 1

Methodology

Methodology

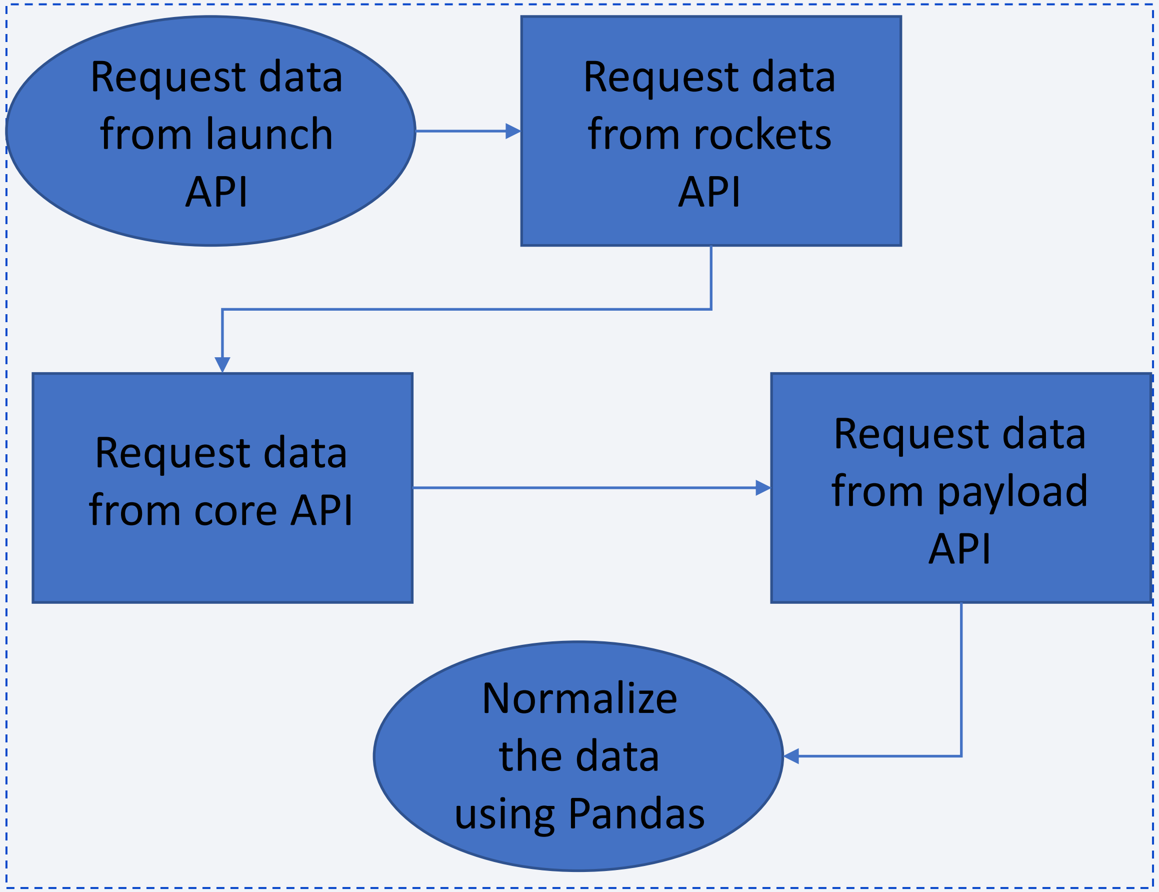
- Data collection methodology:
  - Data was collected from a json file available at api.spacexdata.com, and from web scraping in the Wikipedia Falcon 9 Heavy Launches article.
- Perform data wrangling
  - Using Pandas and NumPy libraries, data was converted into analyzable numeric values.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Logistic Regression, Decision Trees, Support Vector Machines (SVM), and K-Nearest Neighbor
  - All are predictive classifiers that use features of the data to predict which class a new entry with similar features will fall into.

Data Collection

- Requests REST API
  - Data was queried from the Spacex data website using the requests library and REST APIs
  - Data was requested from the Launch, Rockets, Payload, and Core APIs.
  - Data was then normalized using Pandas
- Web scraping
  - Use requests to get the HTML text from the web article
  - Use Beautiful Soup to parse the HTML
  - Extract column and variable names from the HTML table header
  - Create a data frame by parsing the HTML tables

Data Collection – SpaceX API

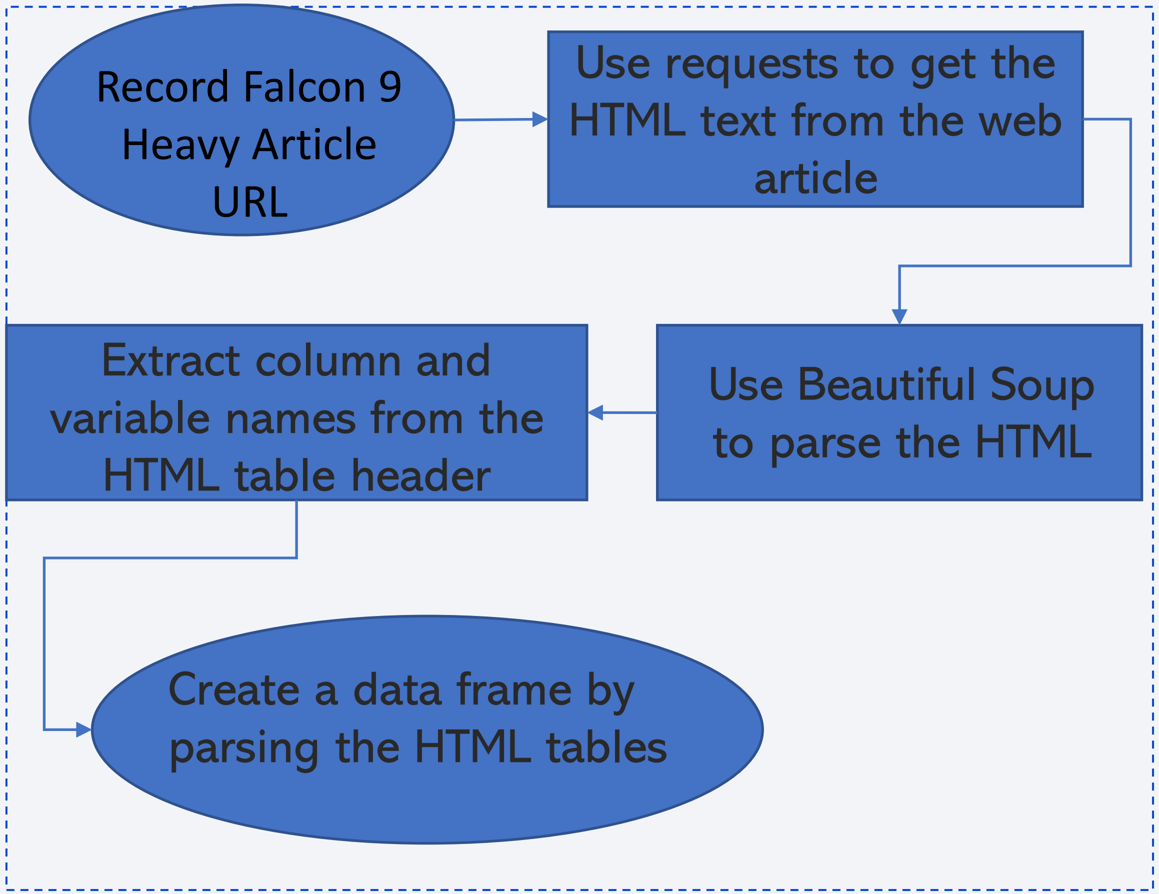
- Flowchart of the requests using REST APIs and the spacex data
- GitHub URL for completed notebook:
    - https://github.com/TheToastBones/DataScienceCapstone/blob/master/Lab1%20Collecting%20Data.ipynb



Data Collection - Scraping

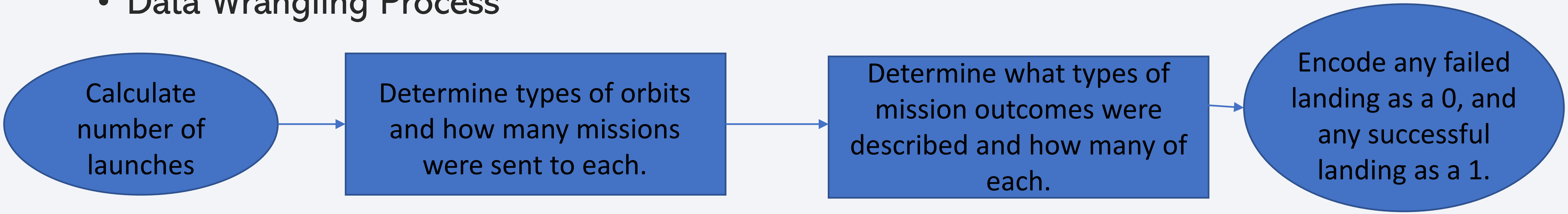
- Flowchart for webs craping process

- GitHub link for completed notebook:
    - https://github.com/TheToastBones/DataScienceCapstone/blob/master/Lab%202%20Web%20Scraping.ipynb



Data Wrangling

- Data Wrangling Process

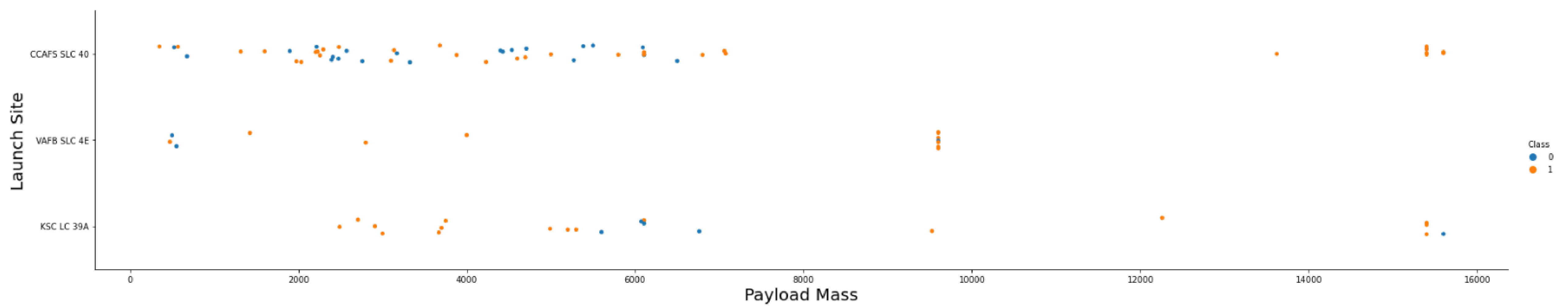


- Tools used for wrangling
  - Pandas and Numpy were used to extract data, perform calculations, and create the data frames.
- GitHub URL for completed Notebook:
  - https://github.com/TheToastBones/DataScienceCapstone/blob/master/Lab%203%20Data%20Wrangling.ipynb

EDA with Data Visualization

- The CatPlot
  - Seaborn's Categorical Plot function (catplot) was used to show relationships between variables such as launch site, payload mass, and whether the landing was successful or not.
  - The described chart is shown in the next slide.
- GitHub URL of completed notebook:
  - https://github.com/TheToastBones/DataScienceCapstone/blob/master/Lab%205%20EDA%20With%20Visualization.ipynb

Seaborn Categorical Plot



In the plot depicted above, blue dots indicate failed landings and orange dots indicate successful landings. The x axis is describing payload mass in kilograms, and the vertical axis indicates the different launch sites.

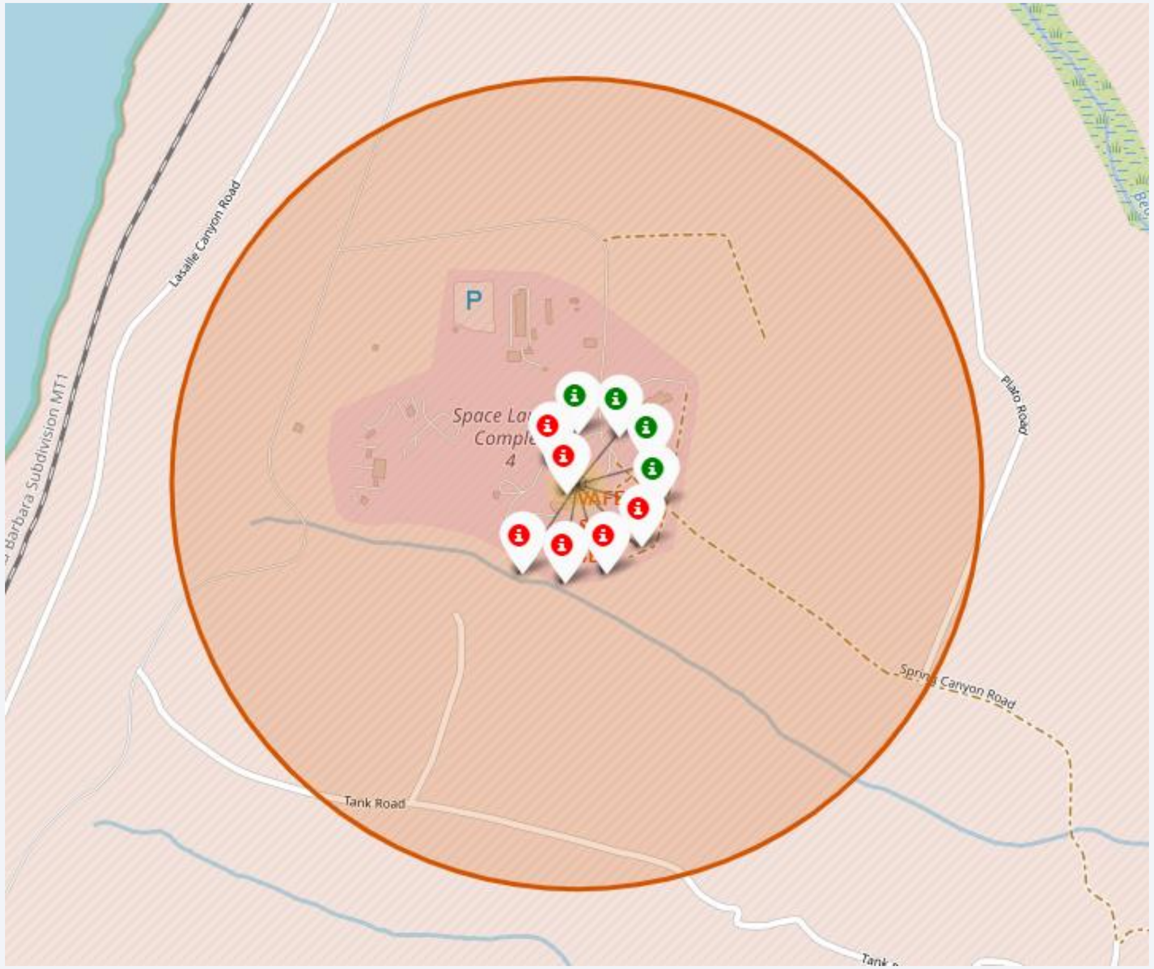
EDA with SQL

- SQL queries were written to locate and present data with specified attributes
  - A list of distinct launch sites
  - The top five launch records where the launch site began with “CCA”
  - The payload mass of each booster launched for NASA (CRS)
  - The average payload mass carried by the F9 v1.1 booster
  - The date of the first successful ground pad landing (December 22nd, 2015)
- The names of boosters that carried the maximum payload (15,600 kg)
- All failed landings involving drone ships, the booster versions, and the launch sites they were launched form in 2015
- A ranked list of landing outcomes by type (highest being “Success” with a count of 38).
- The names of boosters that successfully landed on drone ships with payload mass between 4,000 kg and 6,000 kg
- The total number of successful and failed mission outcomes (not landing outcomes)

GitHub URL of completed notebook:

https://github.com/TheToastBones/DataScienceCapstone/blob/master/Lab%204%20EDA%20with%20SQL.ipynb

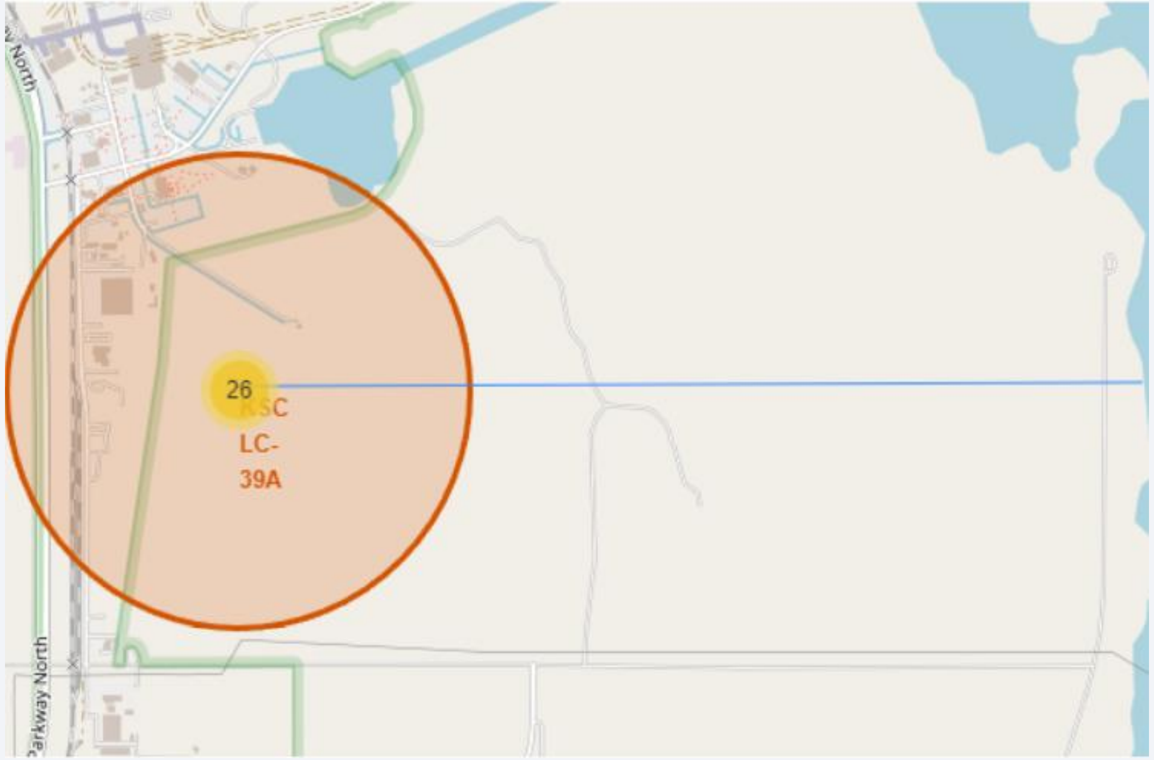
Build an Interactive Map with Folium



- Launch sites were marked with circles and marker clusters were used to indicate where launches took place and whether they succeeded or failed to land.

- GitHub URL for completed notebook:
  - https://github.com/TheToastBones/DataScienceCapstone/blob/master/Lab%206%20Locations%20Analysis%20with%20Folium.ipynb

Build an Interactive Map with Folium



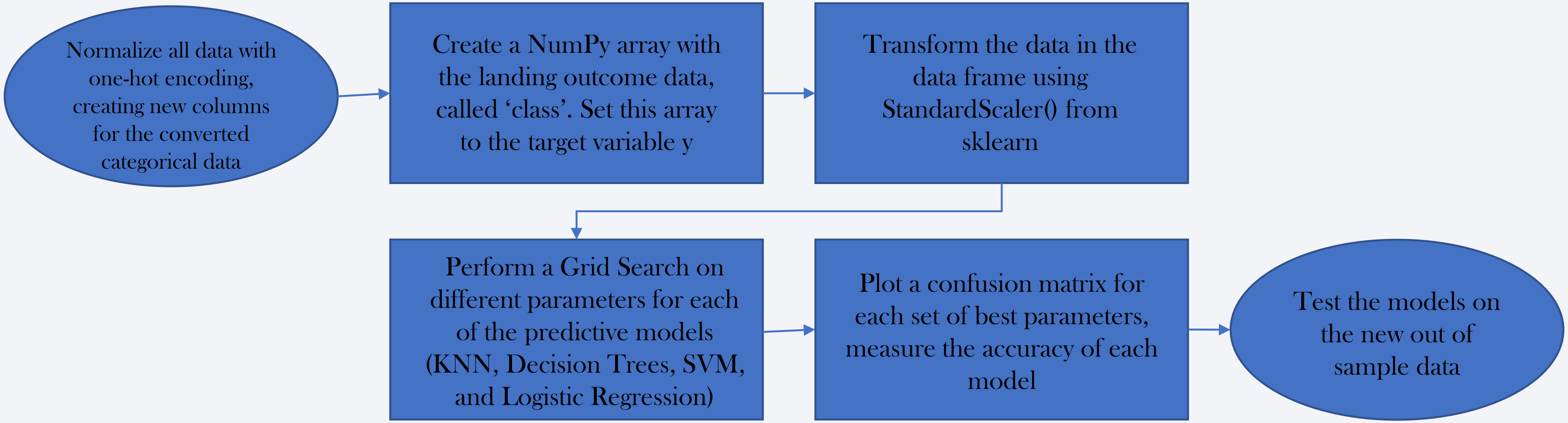
- Distance from Launch sites to nearby features were also marked on the map. This information could be used to detect patterns between nearby geographic features and landing success/failure.
  - In the image shown, a line is drawn from a launch site in Florida to a nearby coastline.
- GitHub URL for completed notebook:
  - https://github.com/TheToastBones/DataScienceCapstone/blob/master/Lab%206%20Locations%20Analysis%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- An interactive pie chart that shows both the overall success rates as a proportion of all successful flights as well as success/failure rates for each site individually.
  - This is shown to indicate which launch sites are most successful both compared with each other and individually
- A Categorical plot that shows the success rate of each payload mass versus the booster version. The drop-down menu allows for different comparisons.
  - This is shown to indicate which boosters and payload masses are associated with successful landings.
- An image of the Dash application is shown on the next slide.
- GitHub URL for interactive Dash app code:
  - https://github.com/TheToastBones/DataScienceCapstone/blob/master/spacex_dash_app.py

Predictive Analysis (Classification)

- Predictive Analysis Process

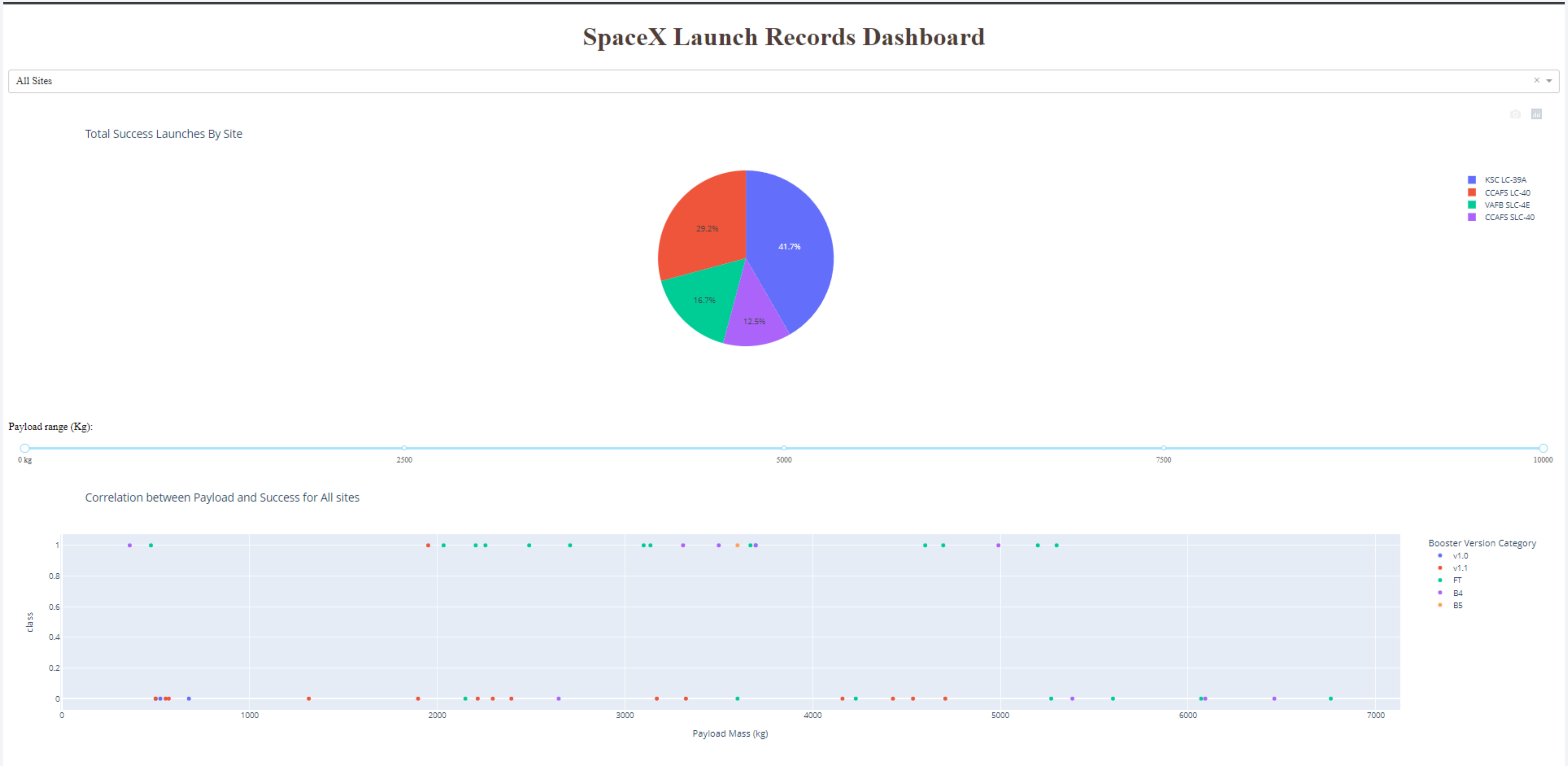


- GitHub URL of completed notebook
  - https://github.com/TheToastBones/DataScienceCapstone/blob/master/Lab%207%20SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- Exploratory data analysis results
  - It was found that as Spacex performed more flights, they had more successful landings. This is likely due to the experience they gained from analyzing previous launches.
  - It also seems that payload mass and the particular launch site have something to do with the success rate. CCAFS LC-40 and KSC LC-39A had the highest success rates.
- Interactive analytics demo in screenshots
  - Refer to the screenshot on the next slide.
- Predictive analysis results
  - The accuracy of each of the models was around 85%, with the most accurate being the decision tree model by a small margin.

Plotly Dash Application Screenshot

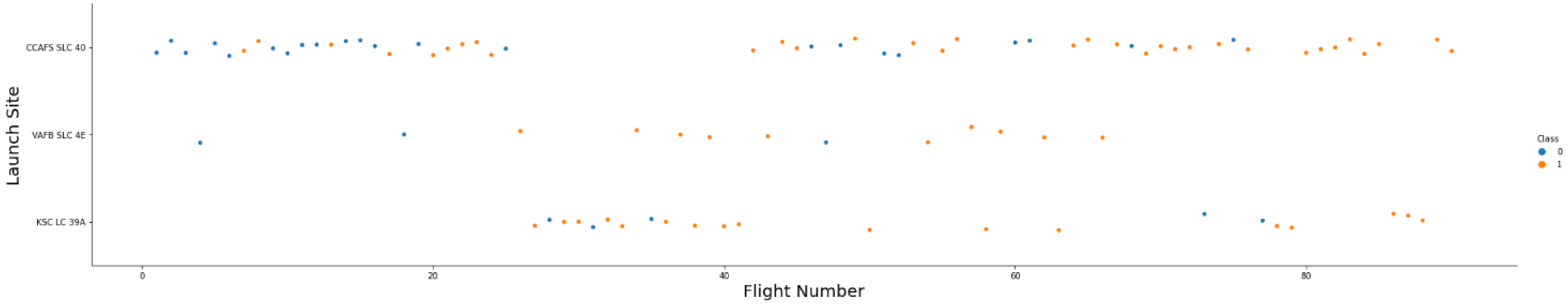


Section 2

Insights drawn from EDA

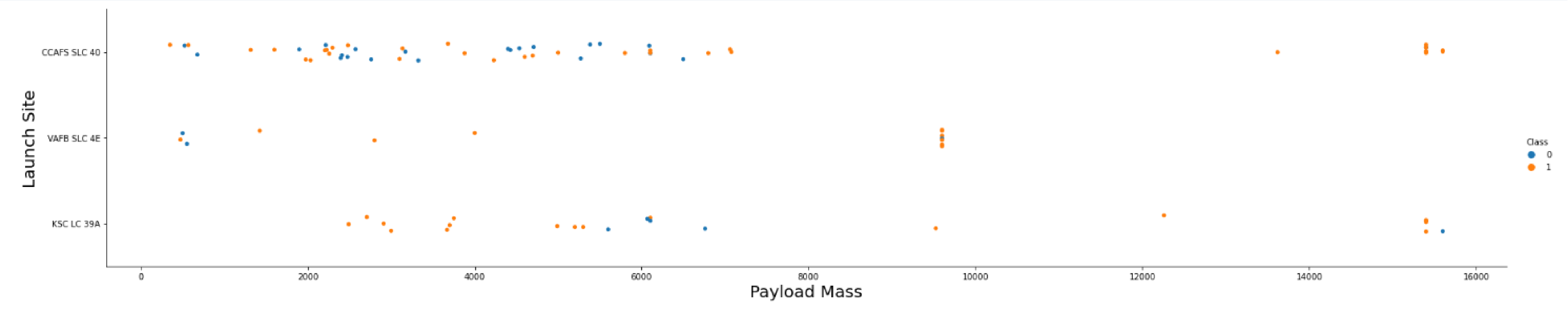
Flight Number vs. Launch Site

- Below is a scatter plot of Flight Number vs. Launch Site
  - Orange dots show successful landings, while blue dots show failed landings.
  - Success seems to increase with flight number
  - Highest percentage of successful landings occurred at KSC LC 39A with 76% success



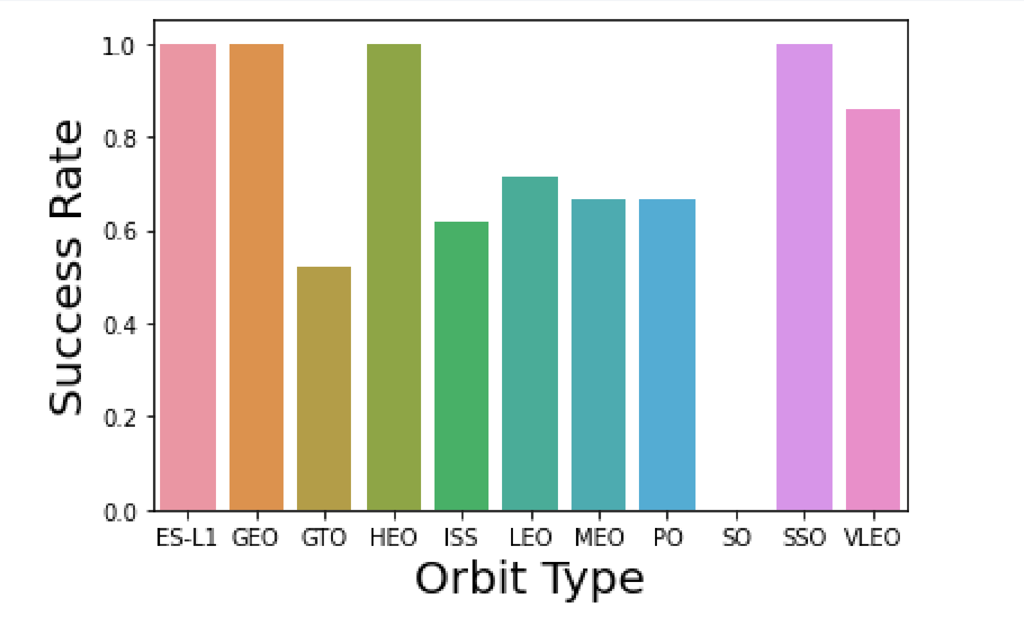
Payload vs. Launch Site

- Below is a scatter plot of Payload vs. Launch Site
    - Orange indicates successful landing, blue indicates failed landing
    - CCAFS SLC 40 had no launches that carried between 8,000 kg and 13,000 kg
    - Most of the heaviest payloads had successful landings, whereas there were many failed landings with lighter payloads



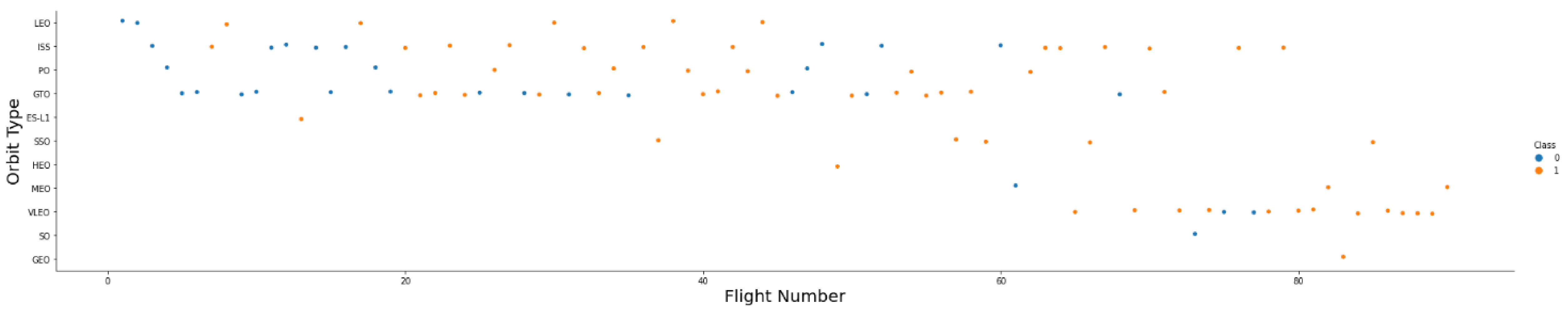
Success Rate vs. Orbit Type

- Below is a bar chart for the success rate of each orbit type
  - ES-L1, GEO, HEO, and SSO had 100% success rate with booster landings. SSO had several launches, whereas the others had 2 or less.
  - The GTO had the lowest positive success rate with dozens of flights.
  - SO had one flight which failed to land.



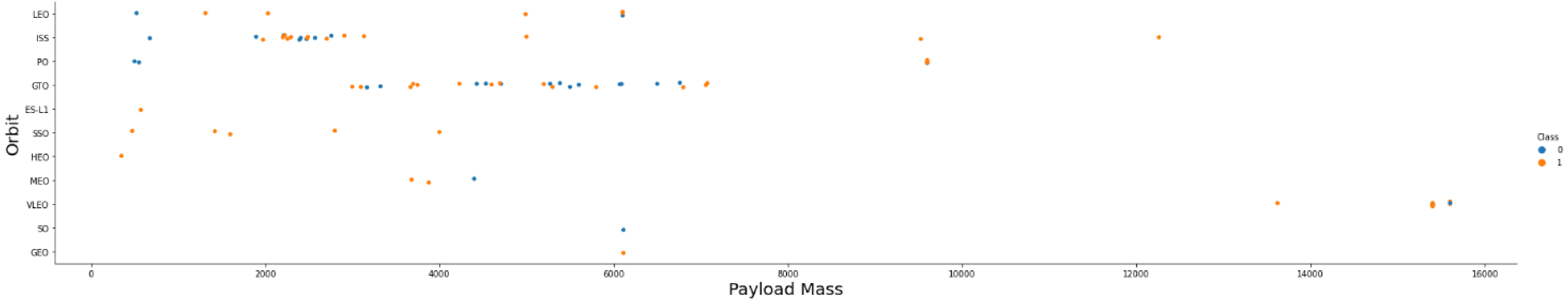
Flight Number vs. Orbit Type

- Below is a scatter plot of Flight number vs. Orbit type
  - Orange indicates success, blue indicates failure.
  - As SpaceX performed more flights, they changed the types of orbits they were launching missions to (for example, LEO orbit missions stopped around flight number 45)
  - As said in the previous slide, it can be seen that the SSO orbit type has several flights with 100% success rate.



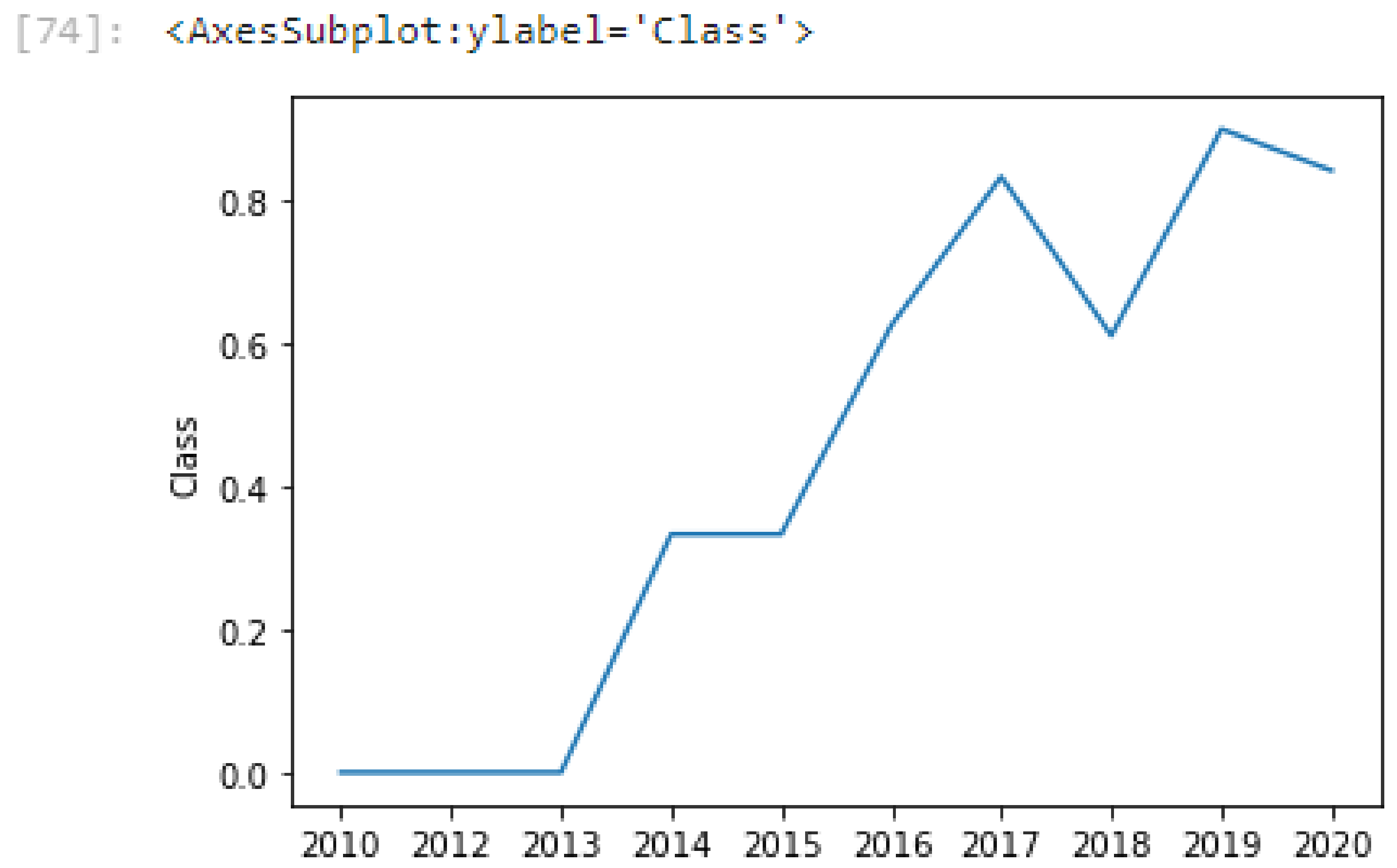
Payload vs. Orbit Type

- Below is a scatter plot of payload vs. orbit type
    - Orange is success, blue is failure
    - Shows which payloads were sent to which orbits
    - Only payloads with mass of 4,000 kg or less were sent to the SSO orbit, which had a 100% success rate
    - Payloads between 2,500 kg and 7,500 kg were sent to the GTO orbit, with mixed success on landing attempts.



Launch Success Yearly Trend

- Below is a line chart of yearly average success rate



- Success was low in the early 2010s, only beginning to climb in 2013.
- Success plateaued around 40% in 2014.
- Large growth in success occurred in 2015 and 2016, with a sudden sharp fall in 2017
- Success recovered and grew even more in 2018 and 2019
- Data indicates a downward trend as of 2020.

All Launch Site Names

- Using SQL, the names of the unique launch sites were found:

```
[3]: %sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL
```

```
 * db2://zfs82611:***@fbd88901-ebdb-4a4f-a32e-9
Done.
```

[3]:

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

The DISTINCT keyword in SQL finds all unique entries that are specified. Here, we see four different, unique launch sites are resulted.

Launch Site Names Begin with 'CCA'

- Pictured is the SQL query and output for selecting the top 5 records where launch sites begin with `CCA

```
[7]: %sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
 * db2://zfs82611:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

[7]:

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

We have resulted the first five such results since the LIMIT function starts at the top. We can find others using OFFSET and LIMIT in conjunction.

Total Payload Mass

- Pictured is the SQL query where the total payload carried by boosters from NASA (CRS) was calculated.

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

```
[21]: %sql SELECT SUM(PAYLOAD_MASS__KG_), CUSTOMER AS "Total Mass" FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)' GROUP BY CUSTOMER
```

```
 * db2://zfs82611:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

[21]:

1	Total Mass
45596	NASA (CRS)

This query adds together all the payload masses of all the NASA (CRS) flights. The total mass comes out to 45,596 kg.

Average Payload Mass by F9 v1.1

- Pictured below on the left is the average payloads of all boosters with F9 v1.1 in their title, and the average of the original one indicated by a red

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

```
[22]: %sql SELECT AVG(PAYLOAD_MASS__KG_), BOOSTER_VERSION AS "F9_v1.1_AVG_PAYLOAD_MASS__KG" FROM SPACEXTBL WHERE BOOSTER_VERSION LIKE '%F9 v1.1%' GROUP BY BOOSTER_VERSION
 * db2://zfs82611:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

[22]:

1	F9_v1.1_AVG_PAYLOAD_MASS__KG
2928	F9 v1.1
500	F9 v1.1 B1003
2216	F9 v1.1 B1010
4428	F9 v1.1 B1011
2395	F9 v1.1 B1012
570	F9 v1.1 B1013
4159	F9 v1.1 B1014
1898	F9 v1.1 B1015
4707	F9 v1.1 B1016
553	F9 v1.1 B1017
1952	F9 v1.1 B1018

Pictured below on the right is the aggregate average mass of all boosters with the F9 v1.1 designation.

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

```
24]: %sql SELECT AVG(PAYLOAD_MASS__KG_) AS "F9_v1.1_AVG_PAYLOAD_MASS__KG" FROM SPACEXTBL WHERE BOOSTER_VERSION LIKE '%F9 v1.1%'
 * db2://zfs82611:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

24]:

F9_v1.1_AVG_PAYLOAD_MASS__KG
2534

First Successful Ground Landing Date

- Below is a query for the first successful landing outcome on ground pad

```
[49]: %sql SELECT LANDING__OUTCOME, MIN("DATE") AS "Date of First Successful Landing"FROM SPACEXTBL WHERE LANDING__OUTCOME LIKE '%Success%' GROUP BY LANDING__OUTCOME
 * db2://zfs82611:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32731/bludb
Done.
```

[49]:

landing__outcome	Date of First Successful Landing
Success	2018-07-22
Success (drone ship)	2016-04-08
Success (ground pad)	2015-12-22

As you can see, the first successful ground pad landing was on December 22nd, 2015.

Successful Drone Ship Landing with Payload between 4000 and 6000

- Here, the names of boosters which have successfully landed on drone ship and had payload mass greater than 4,000 kg but less than 6,000 kg
  - Here is the query:

```
%%sql
SELECT BOOSTER_VERSION, PAYLOAD_MASS__KG_, LANDING__OUTCOME
FROM SPACEXTBL
WHERE (LANDING__OUTCOME = 'Success (drone ship)') AND (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000)
```

  - And the output:

[25]:

booster_version	payload_mass__kg_	landing__outcome
F9 FT B1022	4696	Success (drone ship)
F9 FT B1026	4600	Success (drone ship)
F9 FT B1021.2	5300	Success (drone ship)
F9 FT B1031.2	5200	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- We can calculate the total number of successful and failure mission outcomes with the following query:

```
[68]: %%sql
SELECT COUNT(MISSION_OUTCOME) AS "Count", MISSION_OUTCOME FROM SPACEXTBL GROUP BY MISSION_OUTCOME

 * db2://zfs82611:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomair
Done.
```

[68]:

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

As you can see, the vast majority of the missions were successful. However, mission success and landing success are not the same thing, which is why there is a lower success rate pictured in previous graphics.

Boosters Carried Maximum Payload

- Boosters which carried the maximum payload mass of 15,600 kg.
  - This was the highest payload value listed in the SpaceX data. We have no information as to whether this is maximum possible payload or just the highest that SpaceX launched.
  - Here's the query (below) and the

output (right)

```
%%sql
SELECT PAYLOAD_MASS__KG_, BOOSTER_VERSION FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
ORDER BY BOOSTER_VERSION
```

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

2015 Launch Records

- Failed landings in 2015 that attempted to land on a drone ship and the launch sites they were launched from.

```
%%sql
SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE, "DATE" FROM SPACEXTBL
WHERE (LANDING__OUTCOME = 'Failure (drone ship)') AND (YEAR("DATE")=2015)
```

```
 * db2://zfs82611:***@fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde0
Done.
```

landing__outcome	booster_version	launch_site	DATE
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015-01-10
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015-04-14

Only failed launches in 2015 are shown. To show more, we can remove the year constraint.

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
  - The query (below) and the output (right). As you can see, the most common outcome was "Success" with 38 occurrences.
  - Apparently there was "no attempt" to land 22 of the boosters, so they don't count as failures.

```
%%sql
SELECT LANDING__OUTCOME, COUNT(LANDING__OUTCOME) AS "Count" FROM SPACEXTBL
GROUP BY LANDING__OUTCOME
ORDER BY COUNT(LANDING__OUTCOME) DESC
```

landing__outcome	Count
Success	38
No attempt	22
Success (drone ship)	14
Success (ground pad)	9
Controlled (ocean)	5
Failure (drone ship)	5
Failure	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

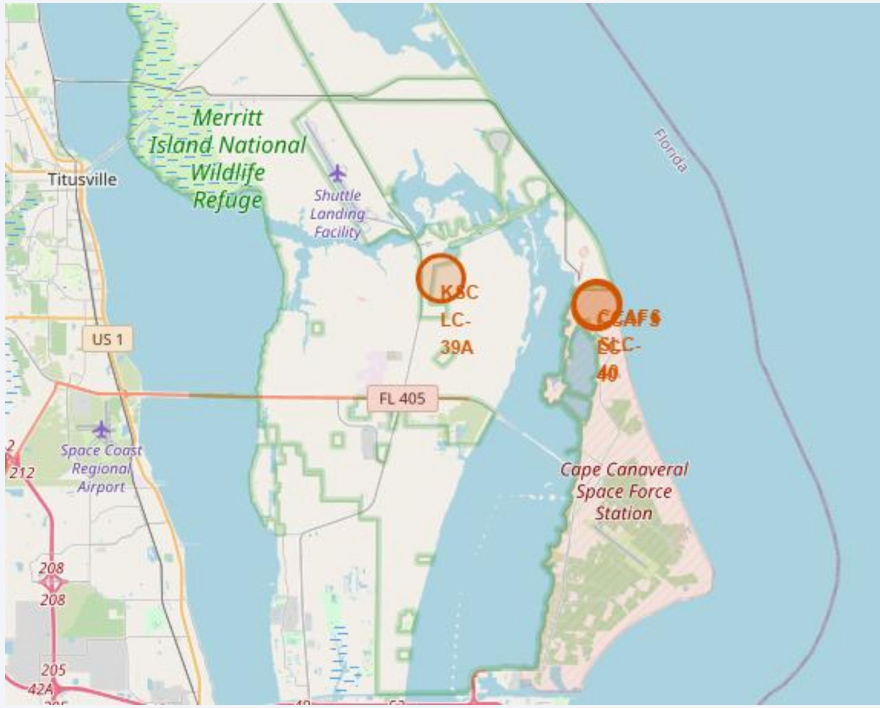
Section 3

Launch Sites Proximities Analysis

Folium Map of All Launch Sites

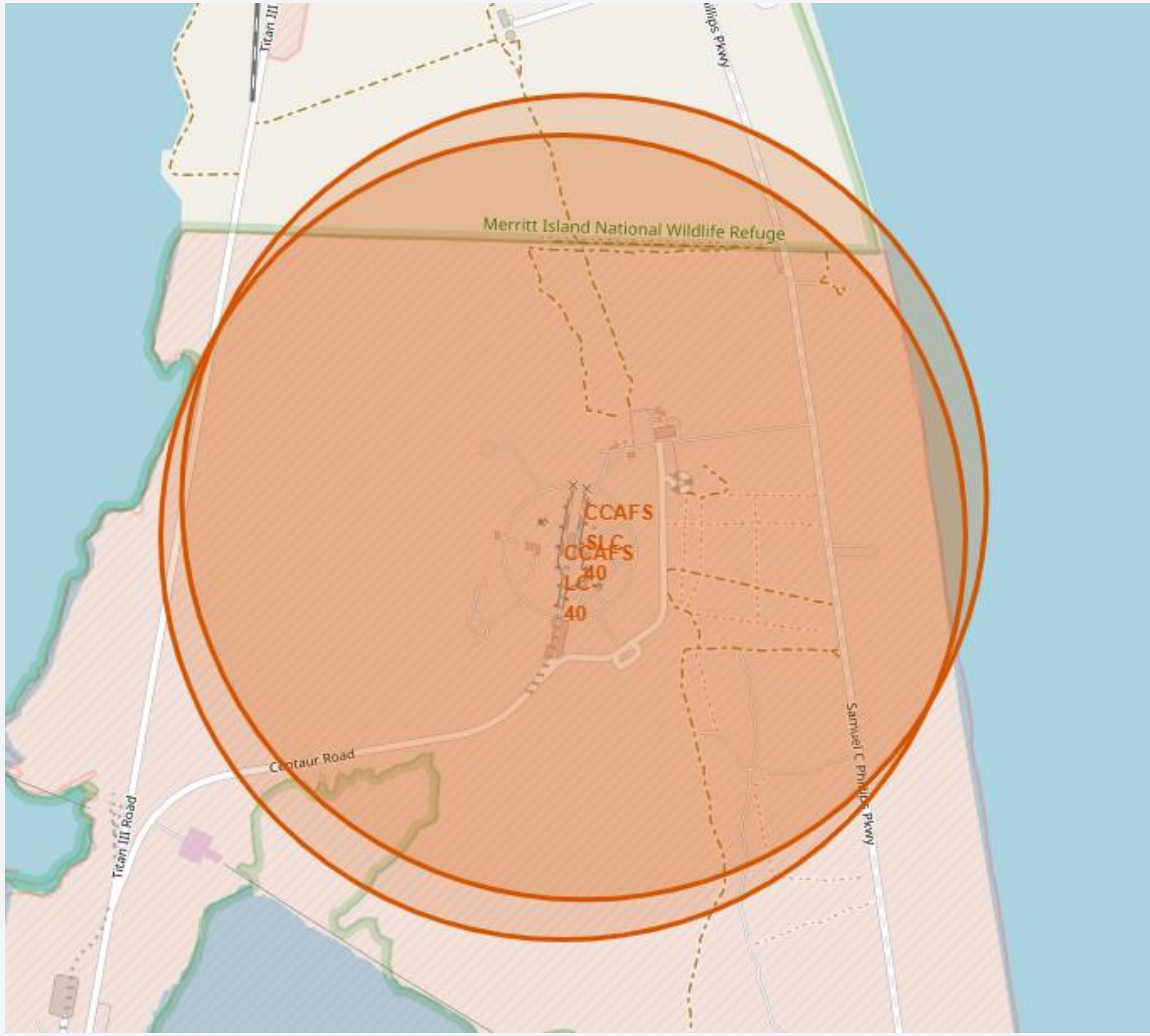
- This screenshots show an interactive map generated by the Folium package for python. The orange markers and circles indicate the Launch sites, and the names of the sires are displayed.

The image on the right shows two sites very close together with similar names, which is why it looks blurred. A more precise screenshot is shown on the next slide.



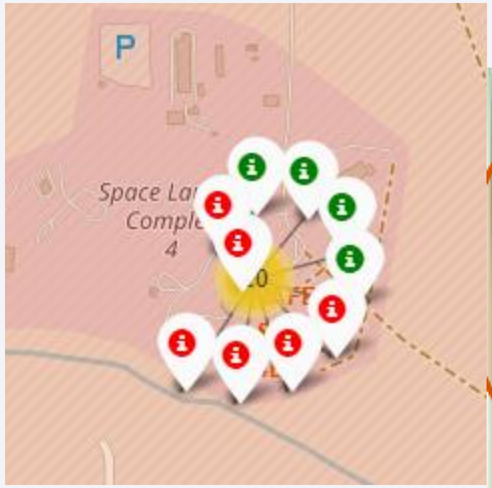
Zoomed-In Screenshot of CCAFS SLC and LC 40

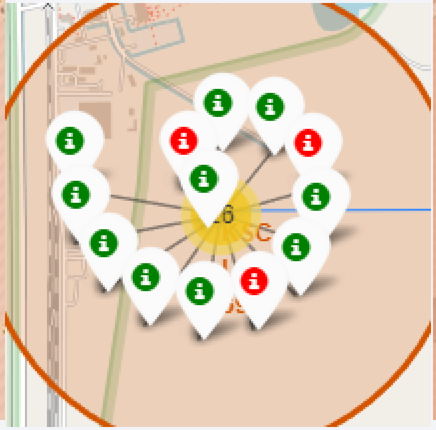
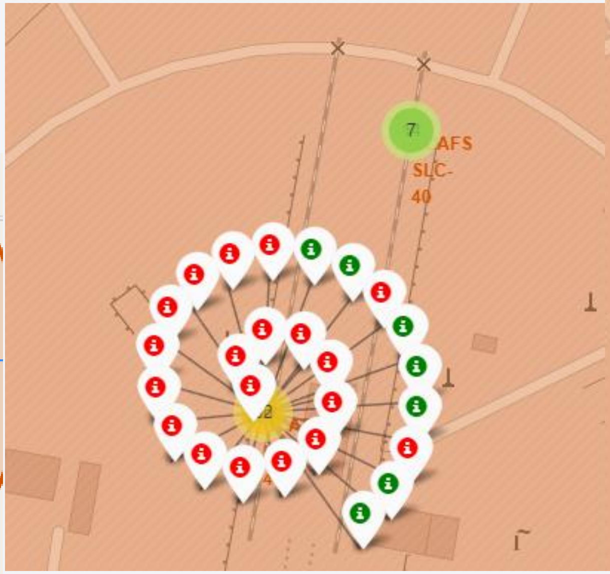
As you can see, these sites are very close to one another and the circle markers overlap.



Marker Clusters with Color-Coded Outcomes

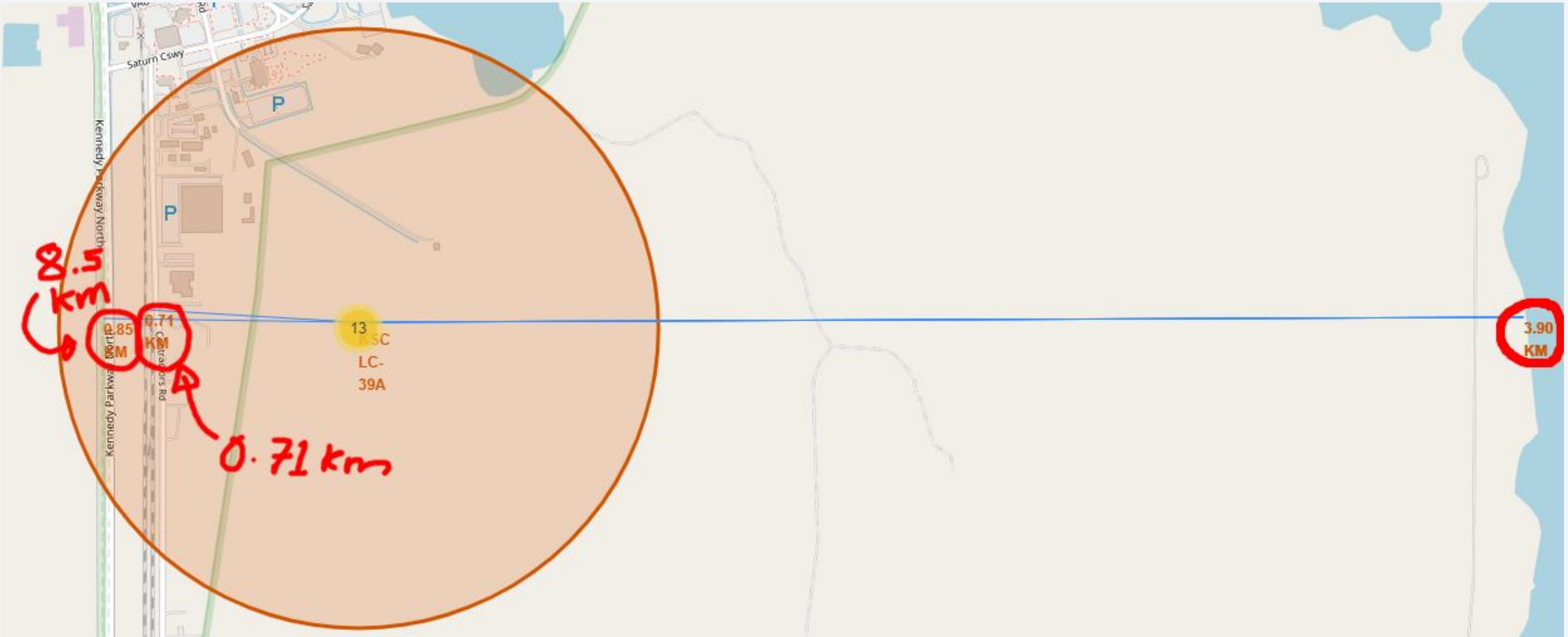
- Green indicates a successful landing and red indicates a failed landing.
    - We use marker clusters since the geographic locations for the sites of many launches are identical.



Folium Map Markers for Distance and Lines

- Below is a screenshot where lines are rendered to local features of interest, in this case a coast line, a railway, and a highway. The distances are also rendered, and highlighted with red circles in the image below.

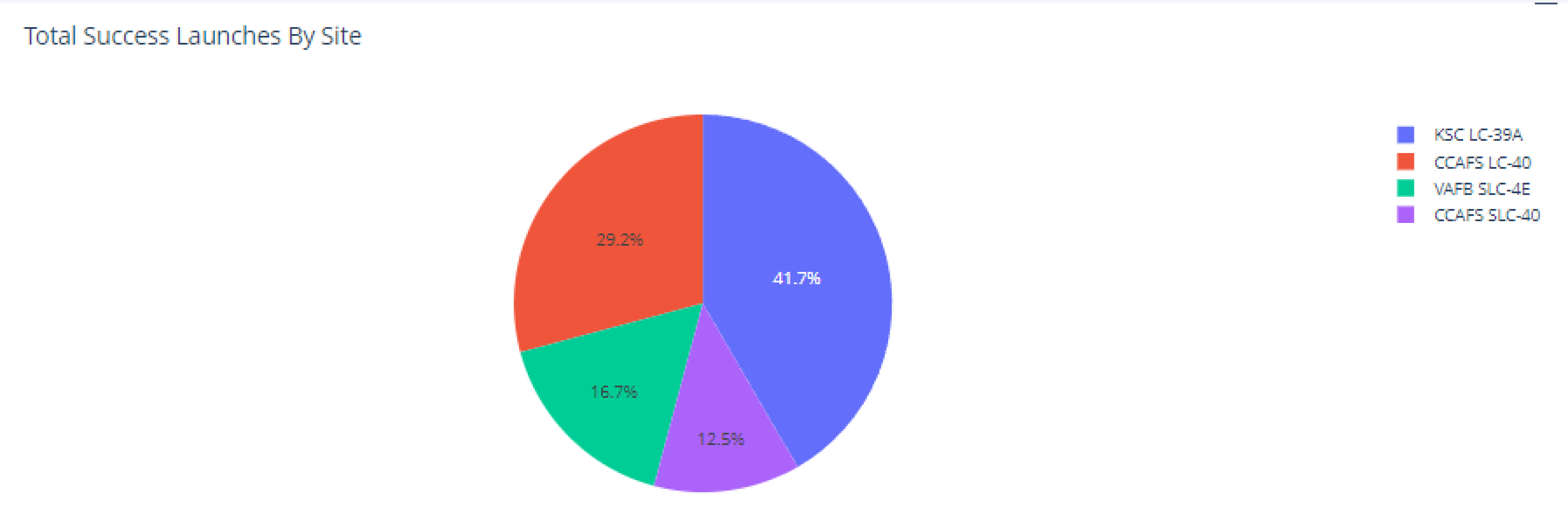


Section 4

Build a Dashboard with Plotly Dash

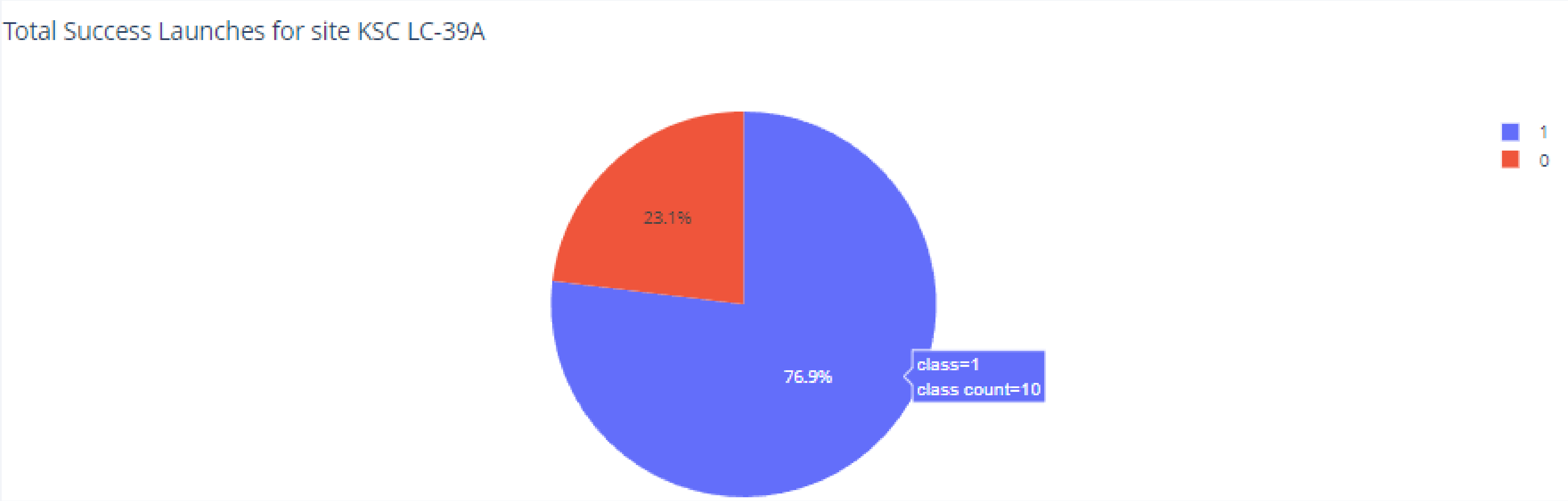
Portion of Successful landings from All Sites

- Pictured is a screenshot of the interactive dash application that shows the percentage proportion of successful landings for all launch sites.
  - Most of the successful landings were launched from KSC LC-39A



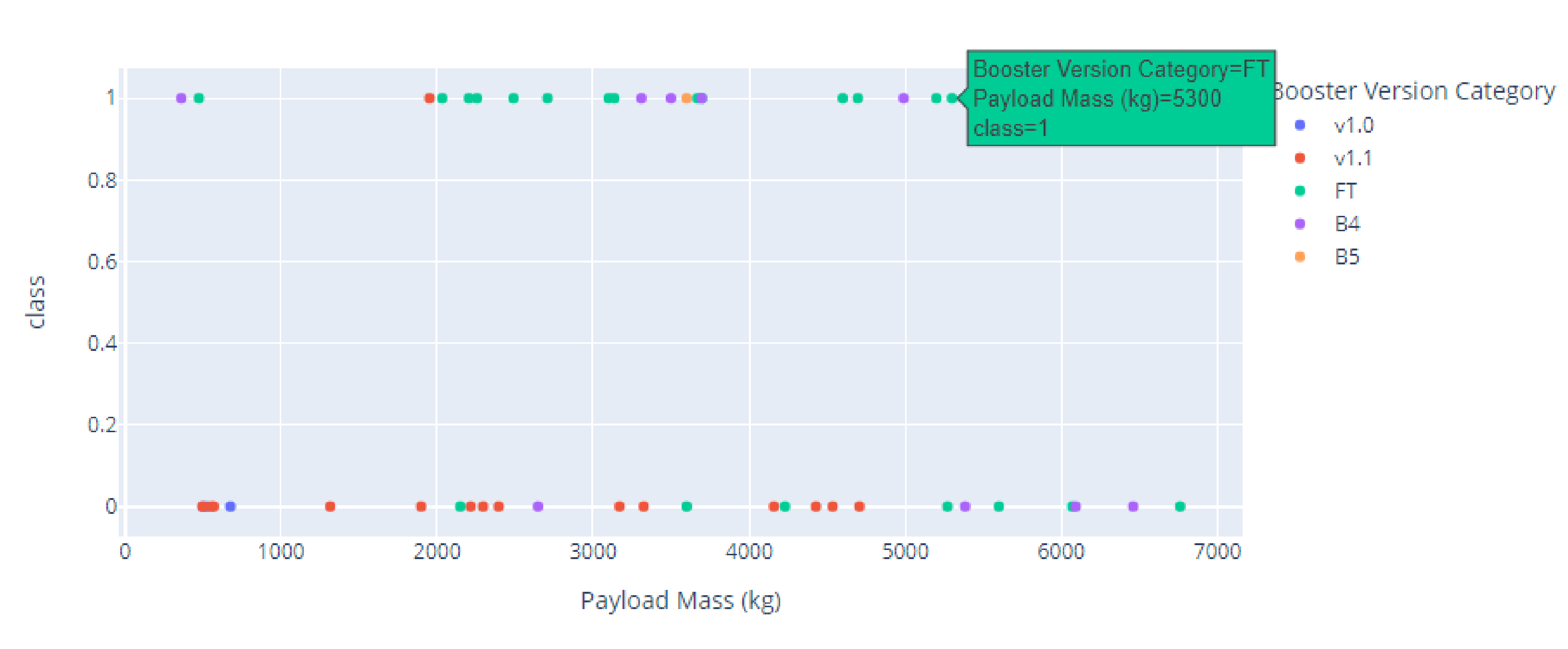
Highest Success Rate Site: KSC LC-39A

- The highest success rate was achieved by the KSC LC-39A launch site.
  - As indicated, nearly 77% of the launches from KSC LC-39A were successfully landed, for a total of 10 landings.



Payload Mass vs Success by Booster Category

- Below is a plot depicting the success rate of each booster type as they relate to payload mass.
  - The highlighted point gives an example, with a successful landing accomplished by the FT booster with a payload mass of 5,300 kg. You can hover your cursor over any point and get a similar summary.



Section 5

Predictive Analysis (Classification)

Classification Accuracy

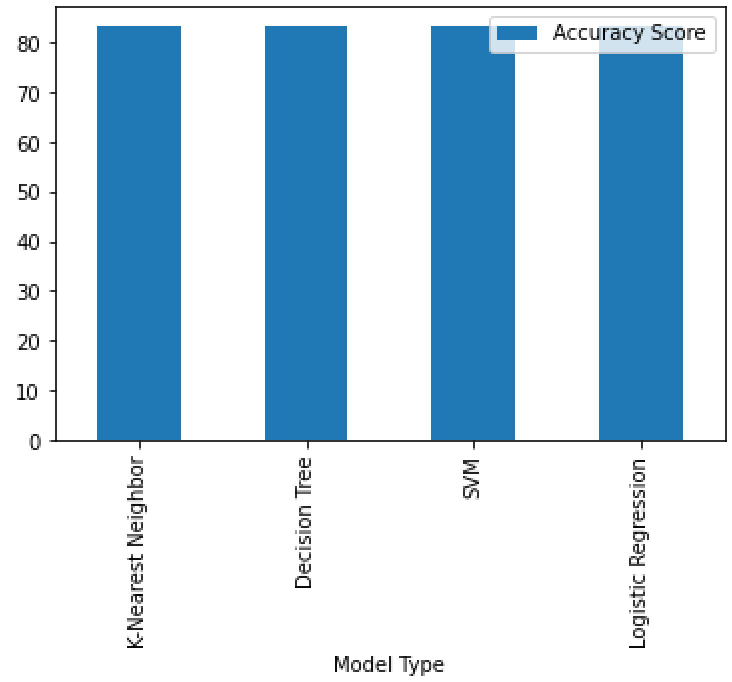
Each model we built had the same accuracy. The code to create the data frame and the output are shown on the left, and the bar chart is shown on the right.

The models likely have the same accuracy because the data set is small and any of these algorithms can perform well. With larger data sets, we'd likely see more variance.

```
accuracy_data = {'Model Type': ["K-Nearest Neighbor","Decision Tree","SVM","Logistic Regression"],
                 'Accuracy Score': [(KNN_cv.score(X_test, Y_test)*100),
                 (tree_cv.score(X_test, Y_test)*100),
                  (svm_cv.score(X_test, Y_test)*100),
                  (logreg_cv.score(X_test, Y_test)*100)]}
accuracy_df = pd.DataFrame(data=accuracy_data)
accuracy_df
```

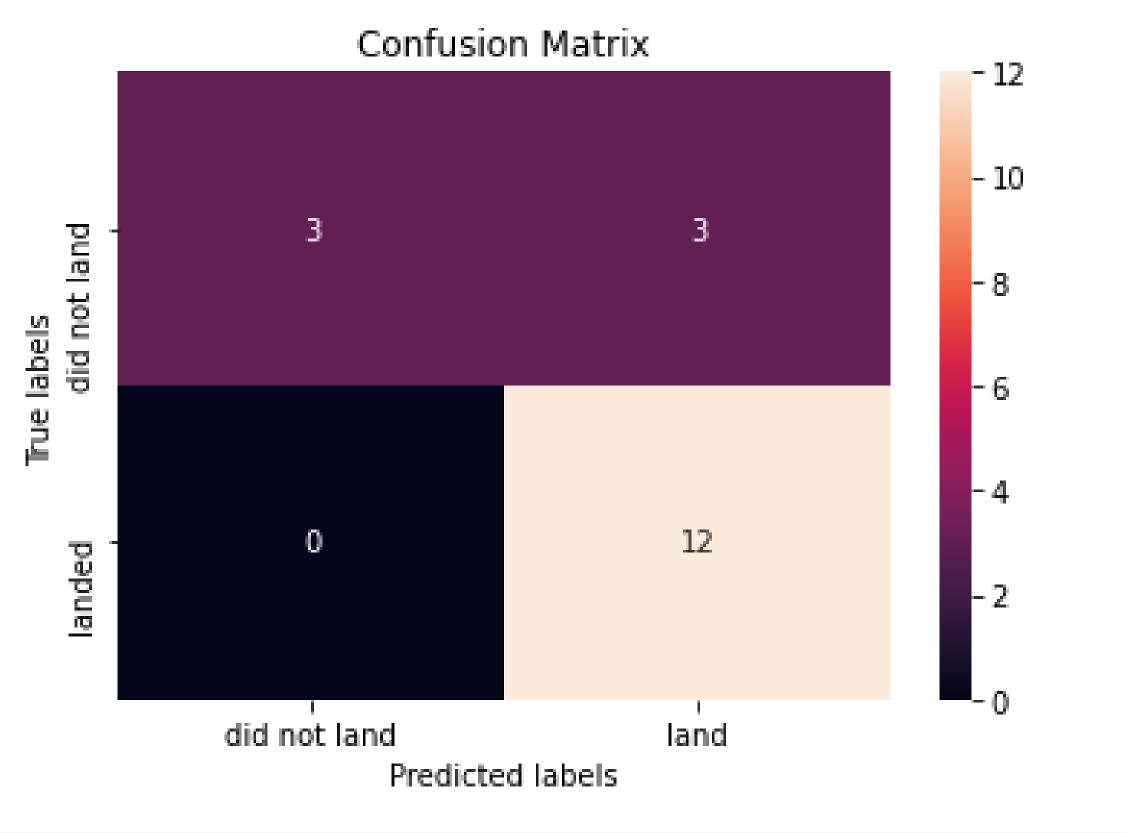
	Model Type	Accuracy Score
0	K-Nearest Neighbor	83.333333
1	Decision Tree	83.333333
2	SVM	83.333333
3	Logistic Regression	83.333333

```
28]: accuracy_df.plot.bar(x='Model Type')
28]: <AxesSubplot:xlabel='Model Type'>
```



Confusion Matrix

- Since all the models perform with the same accuracy, the confusion matrices are identical.



The models appear to have trouble predicting whether a booster will not land with 50% false positives, but it was completely accurate on whether the booster did land, with all 12 samples predicted correctly.

Conclusions

- Early launches had more failures
  - The SpaceX team was trying something new and faced difficulties early on. They collected data and performed analysis to determine how to improve.
- There appears to be some correlation with orbit type and success
  - Particularly, the GTO orbit seemed to have a lot of difficulty.
- The SpaceX team improved with experience
  - Higher flight numbers are correlated with higher success. Without detailed technical knowledge, we can't tell why the flights improved, as the other parameters such as payload, location, and booster version are all influenced by what the SpaceX team learned and not some mysterious pattern.

Appendix

- Primary Data Source:
  - https://api.spacexdata.com is a REST API. In this project, the following extensions were used: /v4/rockets, /v4/launchpads, /v4/payloads, and /v4/cores
- GitHub URL for the complete repository of all files related to this project:
  - https://github.com/TheToastBones/DataScienceCapstone
- Python libraries and packages used in this project:
  - Pandas, MatplotLib, Seaborn. Sklearn, BeautifulSoup, NumPy, Folium, wget, math, sqlalchemy, ibm_db_sa, ipython-sql, bs4, re, unicodedata, requests, datetime
- Wikipedia Article used for web scraping:
  - https://en.wikipedia.org/wiki/Falcon_Heavy

Thank you!