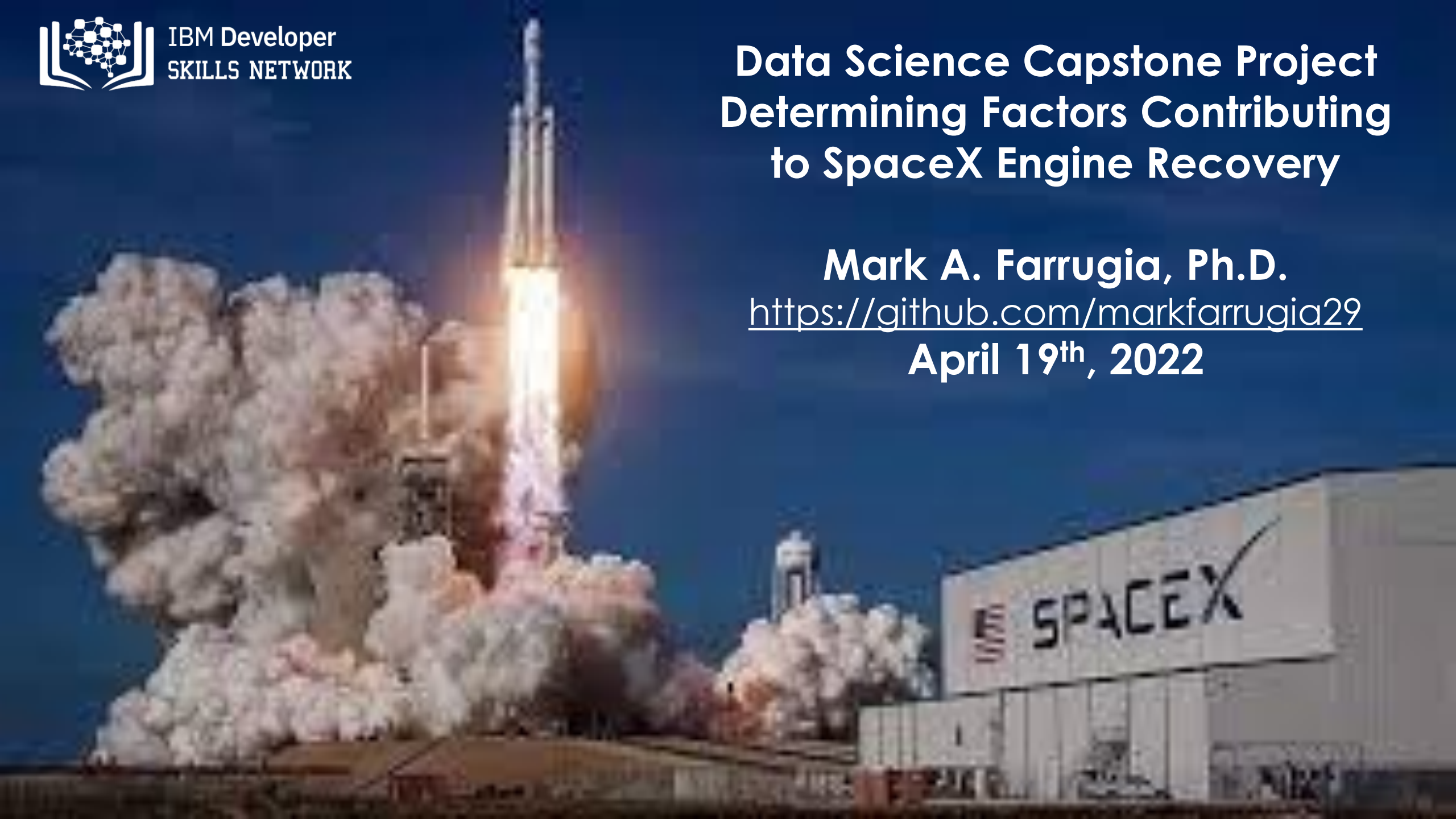


Data Science Capstone Project Determining Factors Contributing to SpaceX Engine Recovery

Mark A. Farrugia, Ph.D.
https://github.com/markfarrugia29

April 19th, 2022

Presentation Outline

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

Executive Summary

The Opportunity: Private space flights are a growing industry, with SpaceX having the best price per launch at $62 million compared to the next best price of $162 million. This is largely due to their ability to capture and reuse their stage 1 rocket.

The Task: SpaceY wishes to generate a machine learning model to predict variables contributing to successful rocket recoveries.

The data: Scrapped SpaceX API and Wikipedia data for launch information based on a variety of temporal and physical parameters.

The methodology: Extracted data using SQL and pandas, prepared preliminary data visualization with Plotly and Seaborn, generated location mapped data using Python libraries Folium and Plotly, predictive analysis of factors contributing to successful launches using Scikit-learn at an 80:20 train:test split.

The outcome: All models have the same accuracy of 83.3% when used on the test set, though Decision Tree had a higher predictive accuracy of 88.9% when used on the training set. This is largely due to the small test sample set, so more data will be needed.

Introduction

Background:

Private companies are vying for personal and government voyages to space

SpaceX has the best pricing ($62M vs $165M in 2022)

Their market advantage is the ability to recover stage 1 of the launch

Deliverable:

Using publicly available launch data from SpaceX, identify the most accurate machine learning model to predict successful stage 1 recovery

Section 1

Methodology

Overview of Data Collection, Wrangling, Visualization, Dashboarding and Modeling Methods

Methodology

- Data collection methodology:
  - Data of interest on launch time and location, as well as physical parameters of the launch, were extracted with SpaceX public API and Wikipedia page
- Performed data wrangling
  - OneHotEncoding was performed in preparation for model generation and launch success was classified as True Landings and failures as False Landings.
- Performed exploratory data analysis (EDA) using Seaborn visualization and SQL
- Performed interactive visual analytics using Folium and Plotly Dash
- Performed predictive analysis using classification models
  - GridSearchCV was utilized to tune and prepare KNN, DecisionTree, SVM and Logistic Regression

Data Collection

Data was extracted using a static dataset generated from the SpaceX API, which yielded a .json response which was converted to a Pandas dataframe for analysis. Supplimental data was scrapped from Wikipedia using BeautifulSoup

Source: (https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922)

Columns from the SpaceX API:

FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitutde

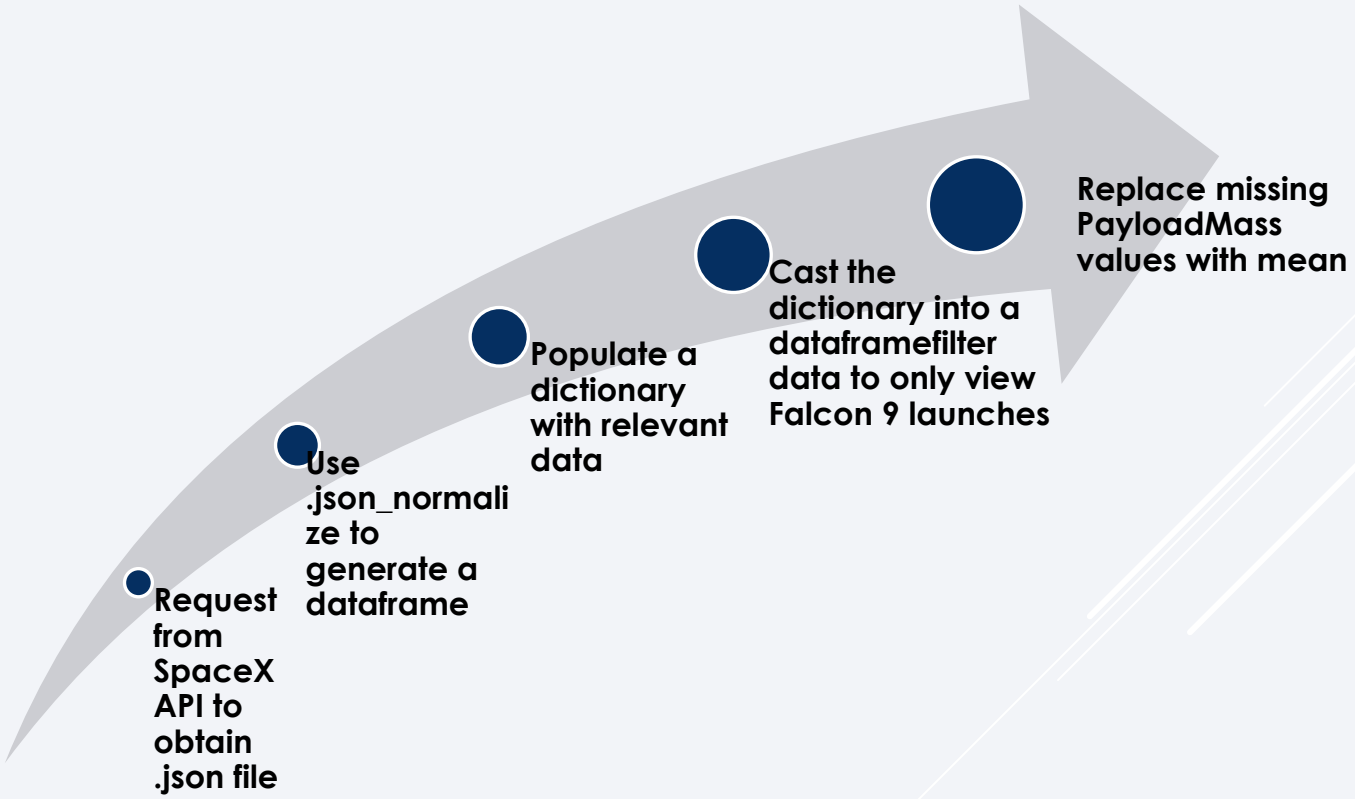
Columns from Wikipedia Webscraping (novel columns in red):

Flight No., Launch Site, Payload, PayloadMass, Orbit, Customer, Launch Outcome, Version Booster, Booster landing, Date, Time

Data Collection – SpaceX API

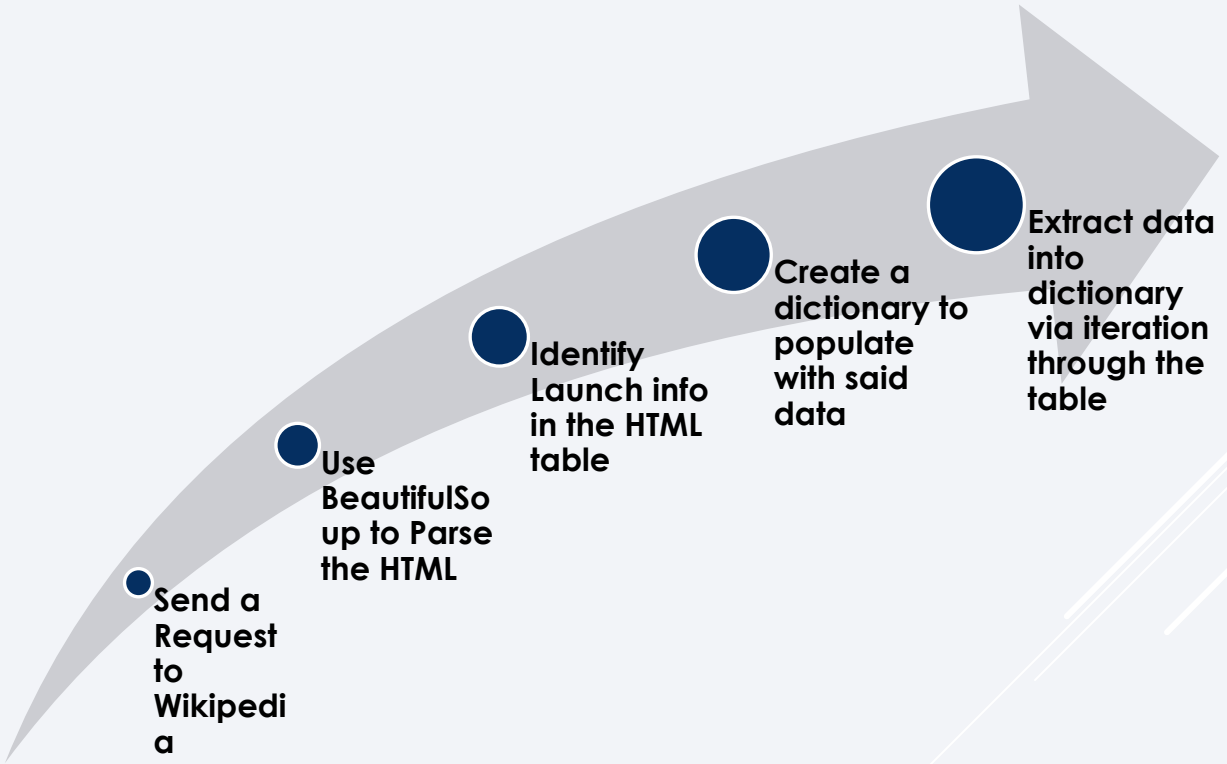
- A jupyter notebook of the following process can be found at the following URL:

- https://github.com/markfarrugia29/IBM-Data-Science-Capstone-Project-SpaceX/blob/main/Capstone%20Project%20SpaceX/Data%20Collection/Data%20Collection%20API%20Lab.ipynb



Data Collection - Scraping

- A jupyter notebook of the following process can be found at the following URL:
- https://github.com/markfarrugia29/IBM-Data-Science-Capstone-Project-SpaceX/blob/main/Capstone%20Project%20SpaceX/Data%20Collection/Data%20Collection%20with%20Webscraping.ipynb



Data Wrangling

Succesful and Unsuccessful recoveries were classed into binary as 1 and 0, Respectively.

Successful recoveries were keyed as follows:

True ASDS, True RTLS, or True Ocean

Unsuccessful recoveries:

None None, False ASDS, False Ocean, None ASDS, False RTLS

A mean success of 66% was identified on analysis of unsuccessful/successful recoveries.

GitHub URL for code:

- https://github.com/markfarrugia29/IBM-Data-Science-Capstone-Project-SpaceX/blob/main/Capstone%20Project%20SpaceX/Data%20Visualization/Data%20Visualization%20Data%20Wrangling%20.ipynb

EDA with Data Visualization

Scatter, bar and line plots were utilized to visualize potential factors contributing to stage 1 recovery rates.

Comparisons:

- Flight No. Vs. Payload Mass/Launch Site
- Payload Mass Vs. Launch site
- Orbit Vs. Success Rate
- Payload Mass Vs. Orbit
- Flight No. Vs. Orbit
- Success over time (years)

Link to github of EDA: **https://github.com/markfarrugia29/IBM-Data-Science-Capstone-Project-SpaceX/blob/main/Capstone%20Project%20SpaceX/Data%20Visualization/Data%20Visualization%20with%20plots.ipynb**

EDA with SQL

- A .CSV file containing SpaceX launch data from 2010-2017 was uploaded to IBM DB2 database
- Several SQL queries were performed to get a better understanding of the dataset
- Information on launch site names, mission outcomes, booster versions, landing outcomes and payload sizes were obtrained

Github URL: https://github.com/markfarrugia29/IBM-Data-Science-Capstone-Project-SpaceX/blob/main/Capstone%20Project%20SpaceX/Data%20Collection/Exploratory%20Data%20analysis%20with%20SQL.ipynb

Build an Interactive Map with Folium

Python was utilized to generate a folium map to mark successful/unsuccessful launch recoveries as well as to identify proximities to several potential success factors: railways, highways, coasts and cities.

These insights were used to help explain why some sites had higher recoveries relative to others.

Github url:

https://github.com/markfarrugia29/IBM-Data-Science-Capstone-Project-SpaceX/blob/main/Capstone%20Project%20SpaceX/Data%20Visualization/Data%20Visualization%20dashboard%20with%20folium%20and%20ploylt%20Dash.ipynb

Build a Dashboard with Plotly Dash

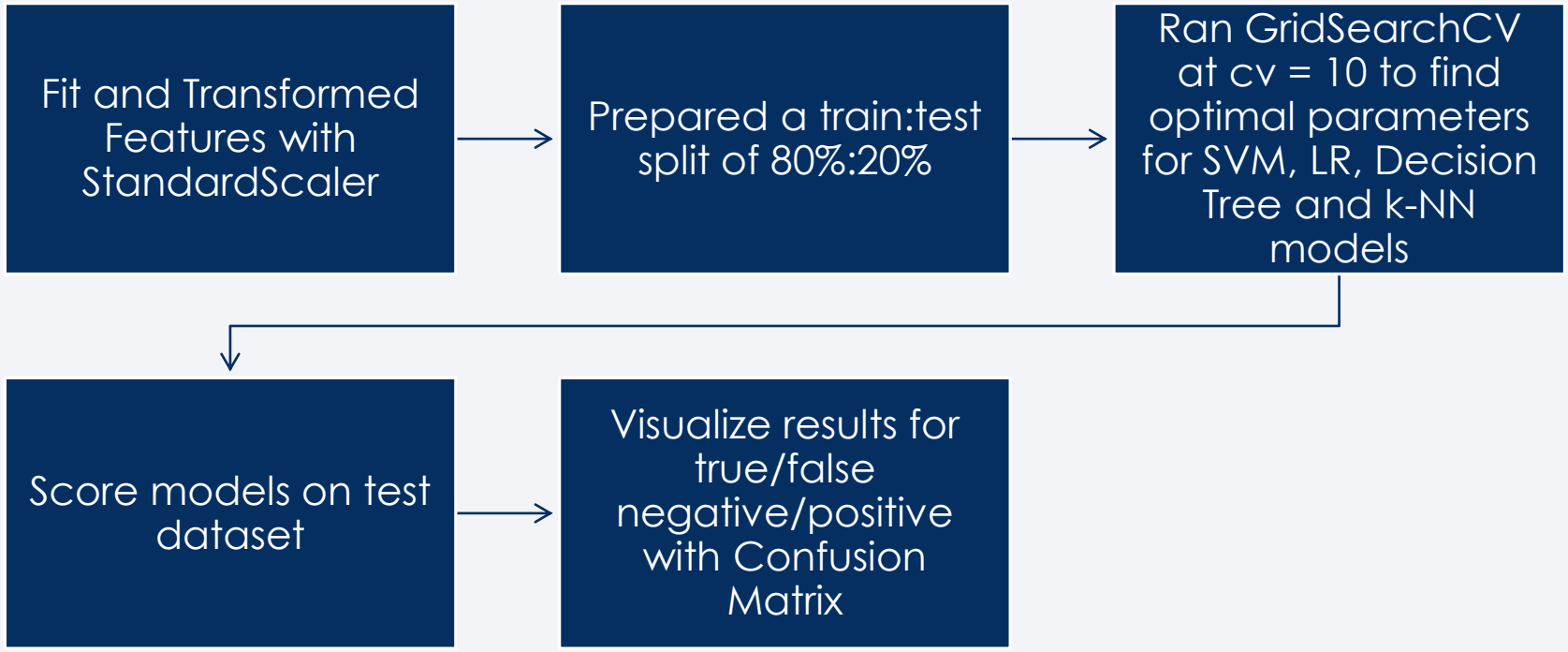
Dash and Plotly Python libraries were used to generate an interactive dashboard containing two specific visualizations.

First, a pie chart was prepared to show how landings are distributed across all launch sites. This chart can be selected to show individual launch site success rates.

A scatter plot was prepared to take either all sites or individual sites, as well as a selectable payload mass between 0 and 10,000 kg. This lets us determine how success could vary depending on launch site, payload mass and booster version.

Github URL: https://github.com/markfarrugia29/IBM-Data-Science-Capstone-Project-SpaceX/blob/main/Capstone%20Project%20SpaceX/Data%20Visualization/SpaceX_app.py

Predictive Analysis (Classification)



Github URL: https://github.com/markfarrugia29/IBM-Data-Science-Capstone-Project-SpaceX/blob/main/Capstone%20Project%20SpaceX/Model%20generation%20and%20evaluation/Capstone%20Machine%20Learning.ipynb

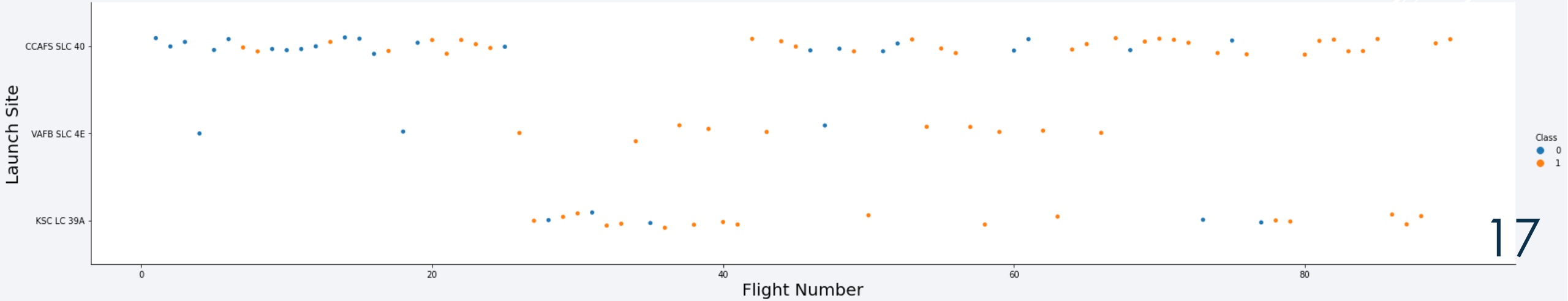
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Below, we see a scatter plot of the flight number of the launch versus the three launch sites. Successful recoveries are shown in **orange** and unsuccessful in **blue**.

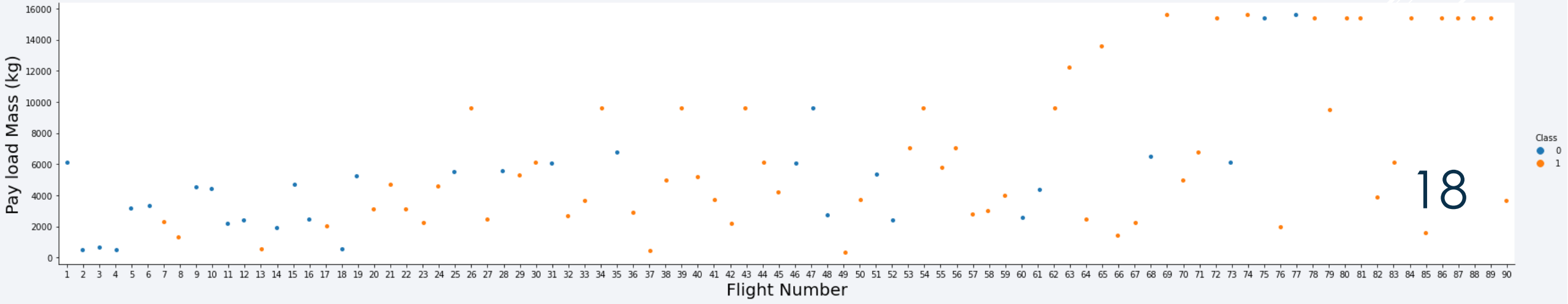
- The bulk of the failures occurred with early launches (<35), which clustered in site CCAFS SLC 40.



Payload vs. Launch Site

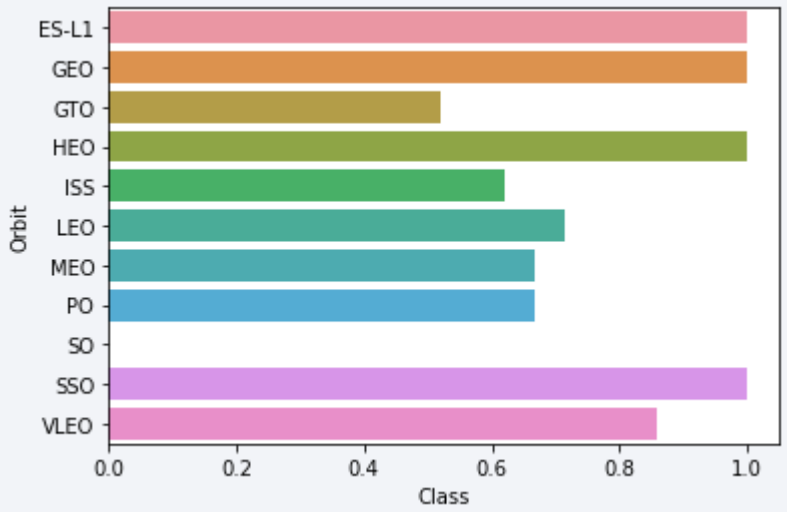
- Below, we see a scatter plot of the flight number of the launch versus the total payload mass (kg). Successful recoveries are shown in **orange** and unsuccessful in **blue**.

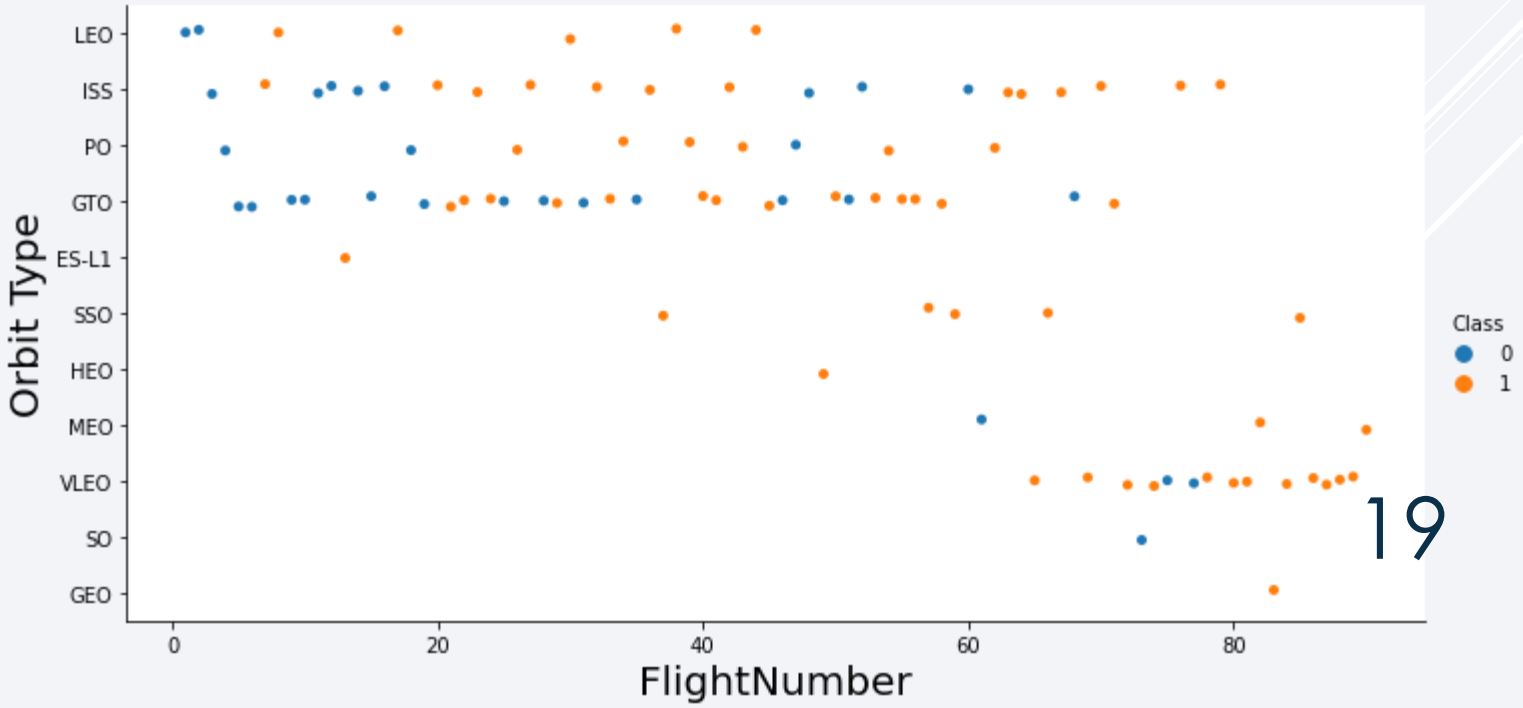
- Again, the bulk of the failures occur with earlier launches, but a bulk of the failed recoveries are with payloads less than 10,000 kg.



Success Rate/Flight Number vs. Orbit Type

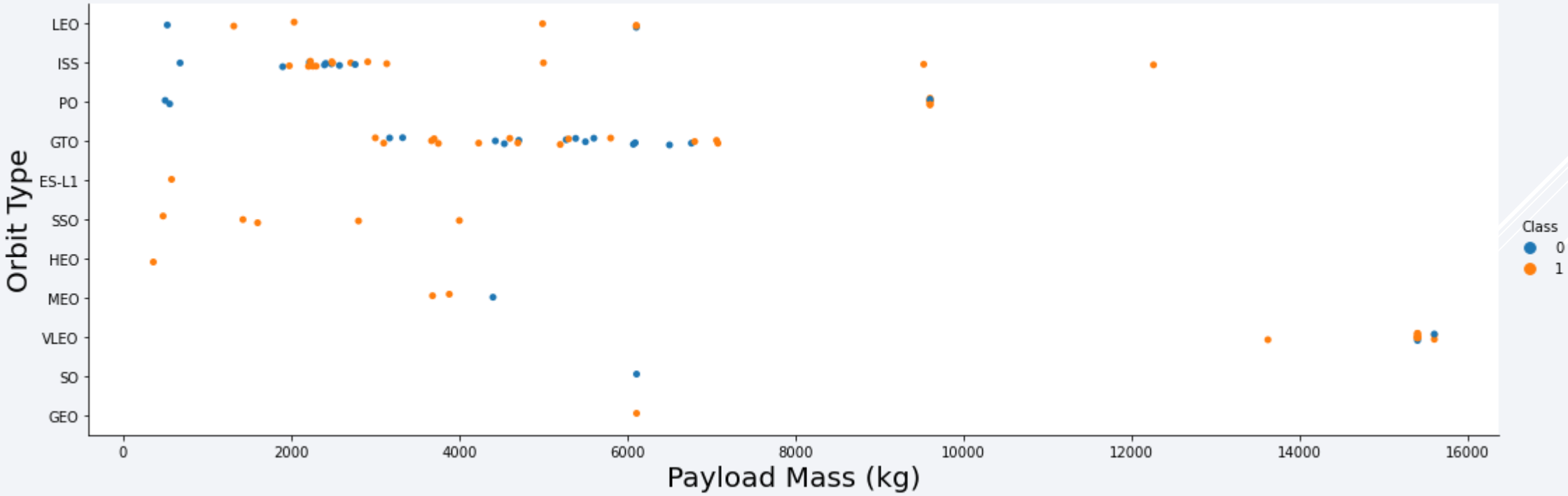
Here we look at flight number versus orbit type. Several orbit types had perfect success rate, but **low total launches** (ES-L1, HEO, GEO). SSO had a perfect success rate, but relatively low total launches. VLEO had a far higher total launch number with a stellar near 80% recovery rate. SO stands out as a complete failure, but it only had a single launch so no conclusions can be drawn. The remaining launch orbits had middling recovery rates around 50-60%.





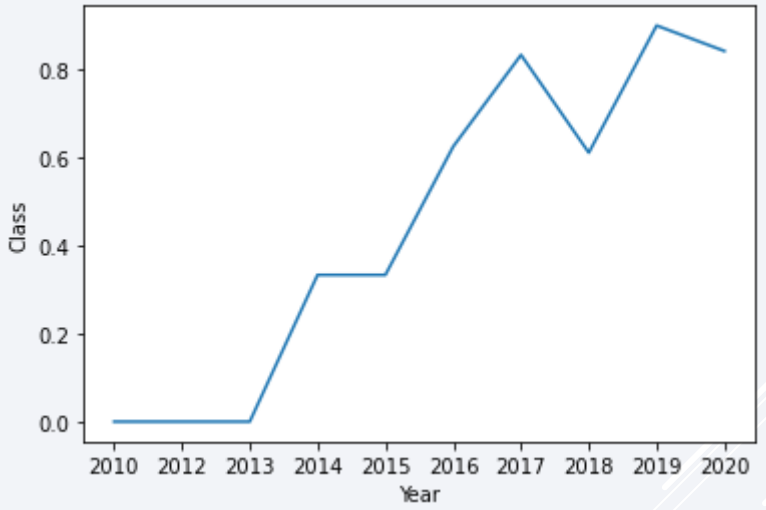
Payload vs. Orbit Type

- With heavy payloads, the successful landing are more for Polar orbit, LEO and ISS. However, for GTO we cannot distinguish this well as both positive landing rate and negative landing are both observed.



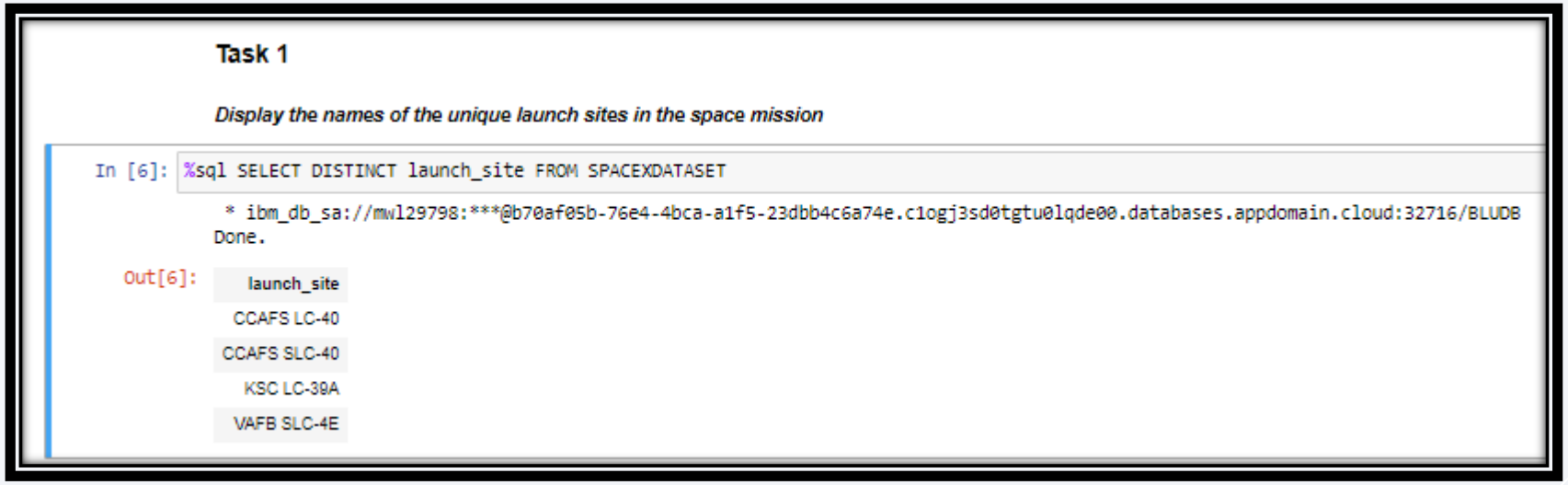
Launch Success Yearly Trend

- As a testament to SpaceX's iterative process, we clearly see an increasing mean success rate (with a perfect rate being 1.0) observed with increasing time.



All Launch Site Names

- SpaceX utilizes four distinct launch sites. To identify them, we can simply query them with SQL using the DISTINCT command in the SELECT statement. Here we see that CAAFS SLC-40/LC-40, KSC LC-39A and VAFB SLC-4E are the sites.



Task 1

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

```
In [6]: %sql SELECT DISTINCT launch_site FROM SPACEXDATASET
```

```
 * ibm_db_sa://mwl29798:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/BLUDB
Done.
```

Out[6]:

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

Launch Site Names Begin with 'CCA'

- We can explore the records with limits on finding attributes based on launch site, such as searching for launch sites starting with CCA.

```
In [15]: %sql SELECT * FROM SPACEXDATASET WHERE launch_site LIKE 'CCA%' LIMIT 5
 * ibm_db_sa://mwl29798:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/BLUDB
Done.
```

Out[15]:

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

Total Payload Mass

- Impressively, we can determine the overall payload mass launched by SpaceX on behalf of NASA. This total payload is 45596 kg as of the writing of this.

```
In [16]: %sql SELECT SUM(payload_mass__kg_) FROM SPACEXDATASET WHERE customer = 'NASA (CRS)'
          * ibm_db_sa://mwl29798:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/BLUDB
         Done.
Out[16]:
| 1 |
|---|
| 45596 |
```

Average Payload Mass by F9 v1.1

- We can also drill down into the data and find average launch masses for specific booster versions. Here we can see the average payload mass carried by booster version F9 v1.1 is 2928 kg.

Task 4

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

```
In [17]: %sql SELECT AVG(payload_mass__kg_) FROM SPACEXDATASET WHERE booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://mwl29798:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/BLUDB
Done.
```

Out[17]:

1
2928

First Successful Ground Landing Date

- SpaceX has had successful recoveries since 2014, but the earliest evidence of a successful ground pad launch was on Dec 22nd, 2015.

```
In [18]: %sql SELECT min(DATE) FROM SPACEXDATASET where landing__outcome = 'Success (ground pad)'
 * ibm_db_sa://mwl29798:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/BLUDB
Done.
Out[18]:
1
2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- Taking a look at specific successful landings on drone ships, we can see how many occurred with a payload between 4000 and 6000 kg.

Task 6

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

In [19]:
```
%sql select distinct booster_version FROM SPACEXDATASET where landing__outcome = 'Success (drone ship)' and payload_mass__kg_ < 6000 AND payload_mass__kg_ > 4000
```

```
 * ibm_db_sa://mwl29798:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/BLUDB
Done.
```

Out[19]:

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

Total Number of Successful and Failure Mission Outcomes

- We can also get a sense of the total number of successful vs unsuccessful mission outcomes. Note this is different than recovered stage 1 rockets. Impressively, the success rate of missions is over 99% with SpaceX! This would mean that most recovered launch failures were planned.

List the total number of successful and failure mission outcomes

```
In [23]: %sql select mission_outcome, count(mission_outcome) from SPACEXDATASET group by mission_outcome
 * ibm_db_sa://mwl29798:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/BLUDB
Done.
```

Out[23]:

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- Thinking that payload mass has an effect on success rates, we can identify all of the booster versions that carried the maximum payload amount. All boosters are of the F9 B5 B10**.* type.

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [30]:
```
#%sql select max(booster_version) from spacexdataset
%sql select booster_version from SPACEXDATASET where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXDATASET)
```

```
 * ibm_db_sa://mwl29798:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/BLUDB
Done.
```

Out[30]:

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

2015 Launch Records

- We can also identified failed drone ship landings in the year 2015 and display some features of the events. Only two such events were documented.

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

```
In [13]: %sql Select booster_version, launch_site, date from spacexdataset WHERE landing__outcome = 'Failure (drone ship)' AND date like '2015%'
```

```
 * ibm_db_sa://mwl29798:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/BLUDB
Done.
```

Out[13]:

booster_version	launch_site	DATE
F9 v1.1 B1012	CCAFS LC-40	2015-01-10
F9 v1.1 B1015	CCAFS LC-40	2015-04-14

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

- Finally, we can explore all landing outcomes between certain timeframes, such as 2010-2017.

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

In [73]:
```
%sql SELECT landing__outcome, count(landing__outcome) AS total_events FROM spacexdataset WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY landing__outcome ORDER BY total_events DESC
```

```
 * ibm_db_sa://mwl29798:***@b70af05b-76e4-4bca-a1f5-23dbb4c6a74e.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32716/BLUDB
Done.
```

Out[73]:

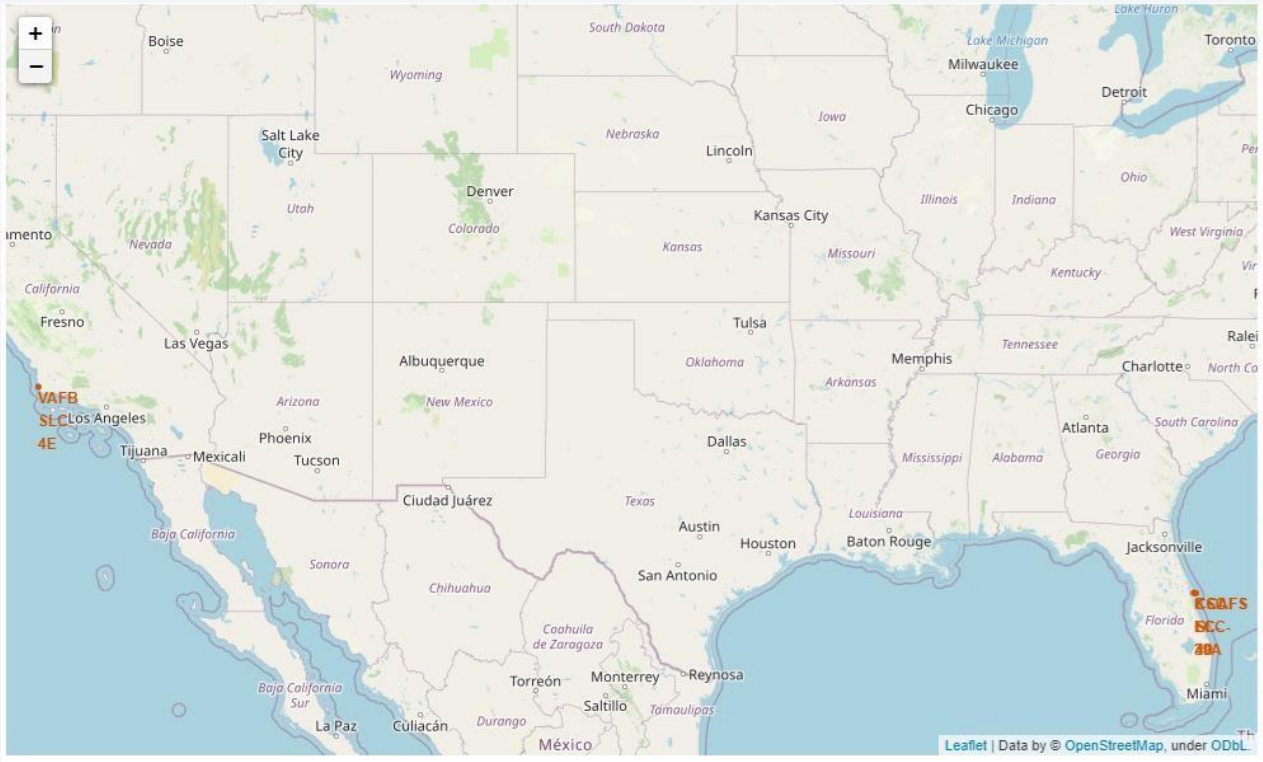
landing__outcome	total_events
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

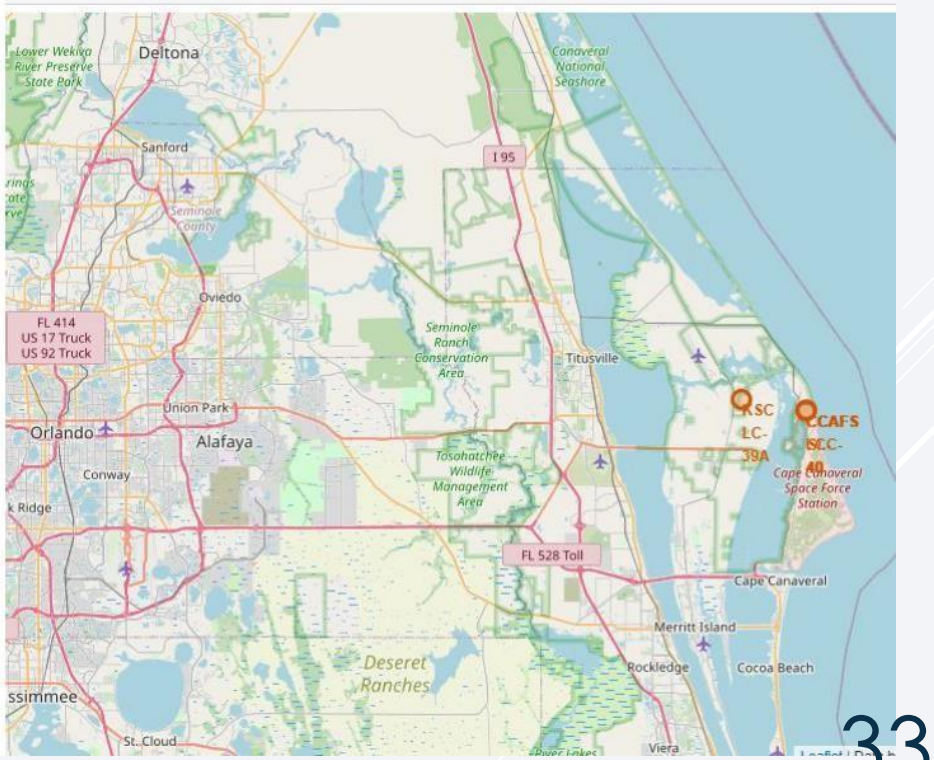
Section 3

Launch Sites Proximities Analysis

Locations of SpaceX Launch Sites

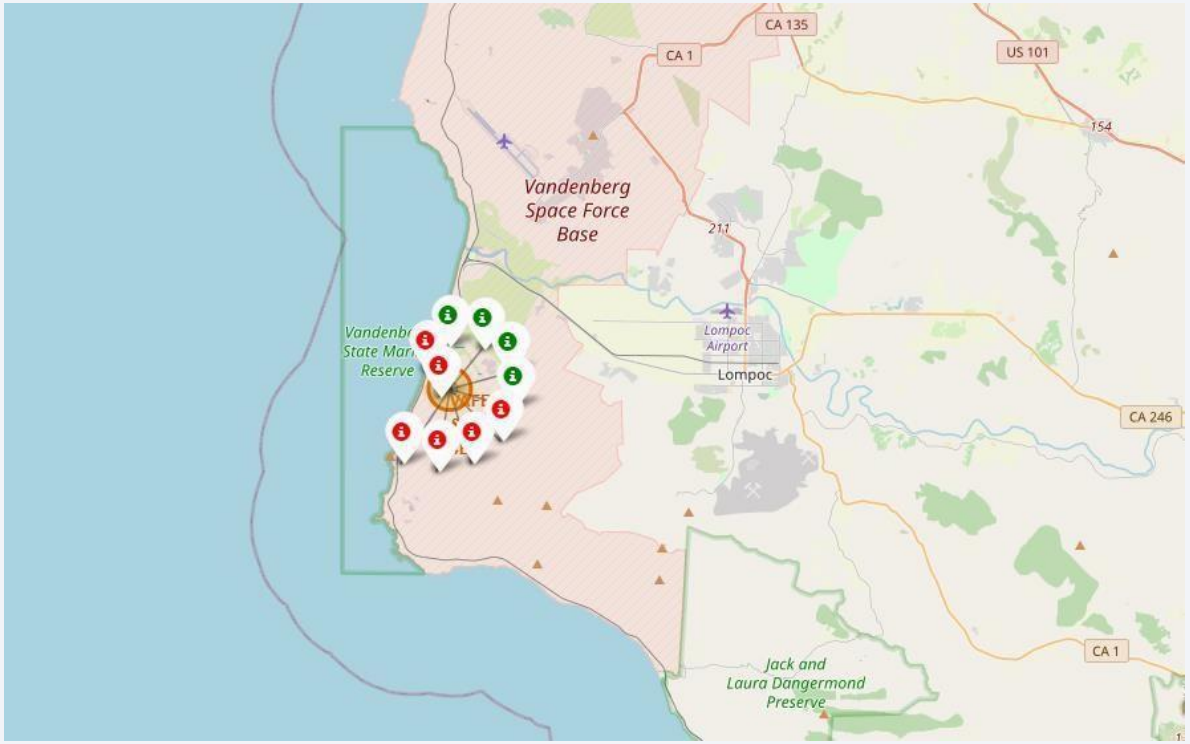
- Using the folium python library, we can plot the locations of each launch site on a map (below)





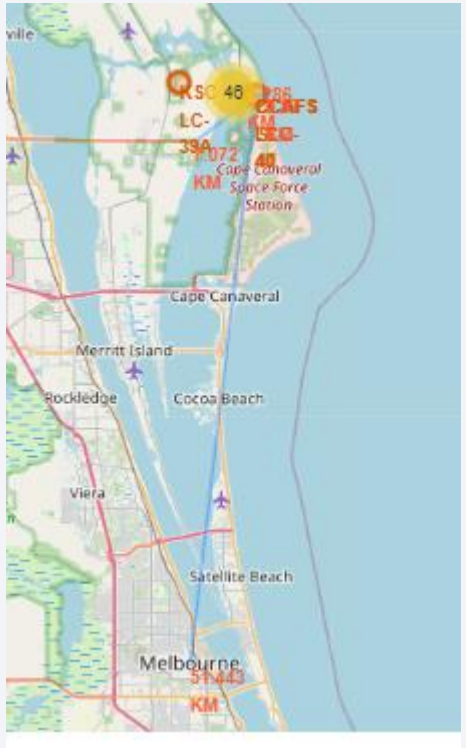
Identifying Successful/Failed Recoveries Per Launch Site

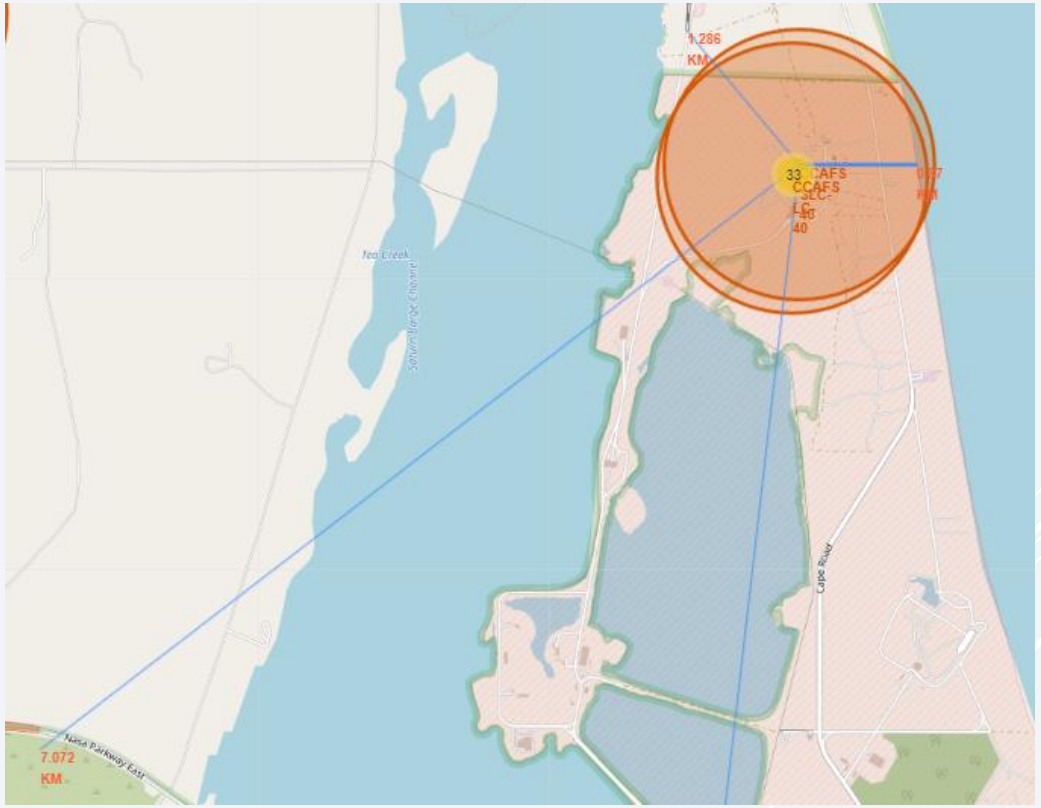
- We can also add identifiers to each site marker by plotting successes (green) and failures (red) on top of the markers. Here, we are looking at VAFB SLC-4E and observing 6 failed landings to 4 successful ones.



Proximities to Critical Locations/Infrastructure

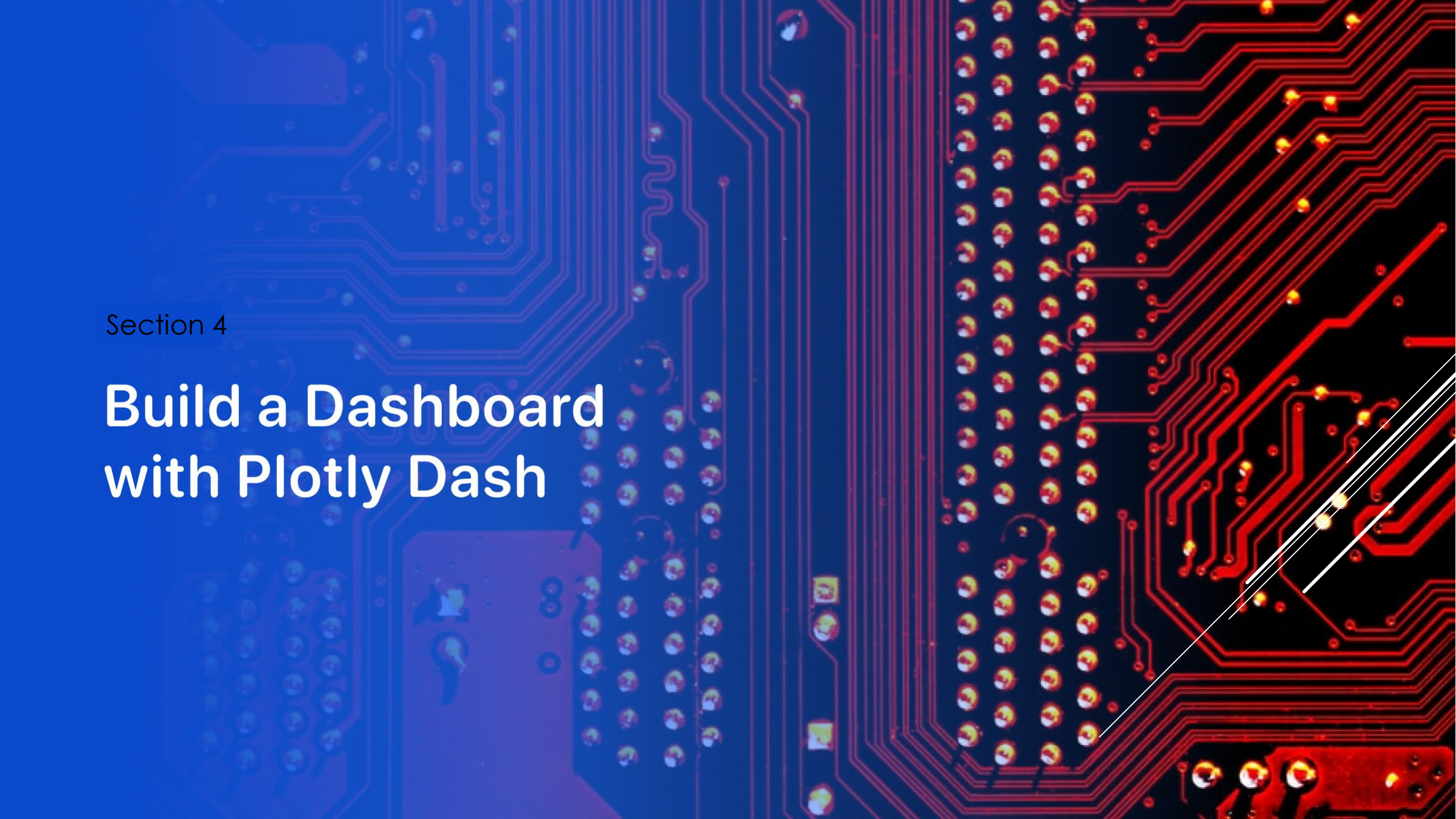
We can look at the distances from CCAFS SLC-40 to major infrastructure, such as cities, rail lines, highways and coastlines.





Section 4

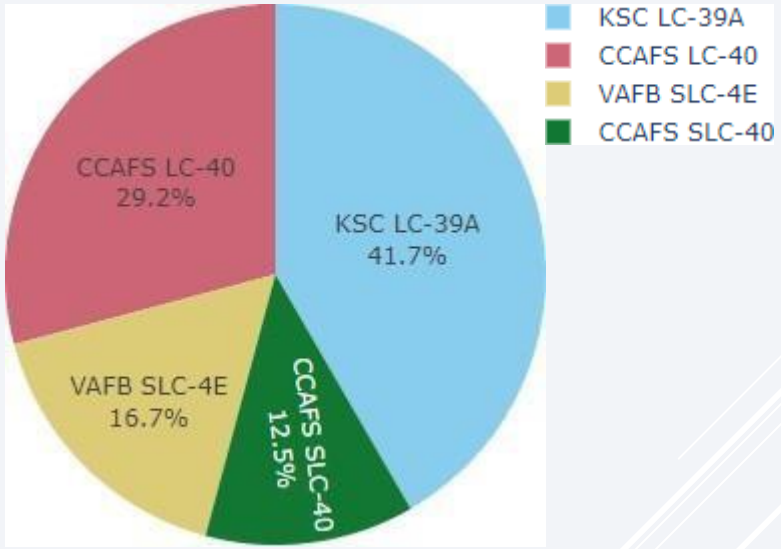
Build a Dashboard with Plotly Dash



Pie Chart of Successful Landings For All Launch Sites

Let's take a look at the total successful stage 1 rocket landings for all launch sites. We can see that KSC LC-39A has the highest share of successful landings, with CCAFS LC-40/CCAFS SLC-40 being 2nd. Please note, CCAFS LC-40 is a name change of CCAFS SLC-40, so they are the same location in reality.

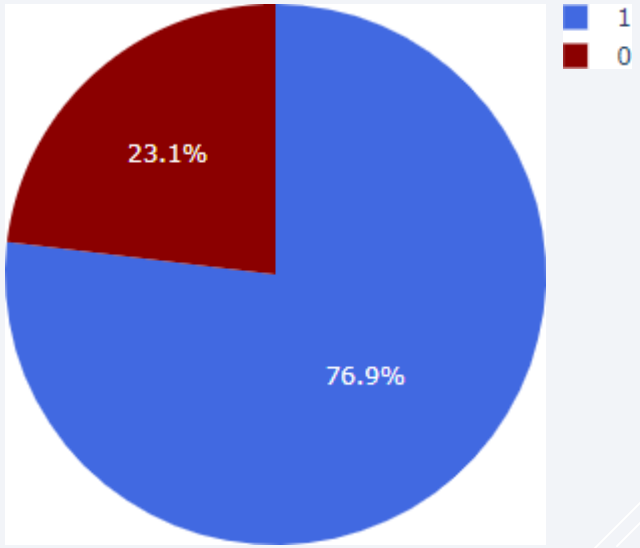
The west coast launch site, VAFB SLC-4E, has the lowest share of successful landings.



Visualization of Successful/Unsuccessful Landings by Individual Launch Site

Drilling down to the individual site with the highest share of successful launches, we can see that KSC LC-39A has an exceptionally high success rate with 10 successful landings and 3 failed landings.

Landing History for
Launch Site KSC LC-39A

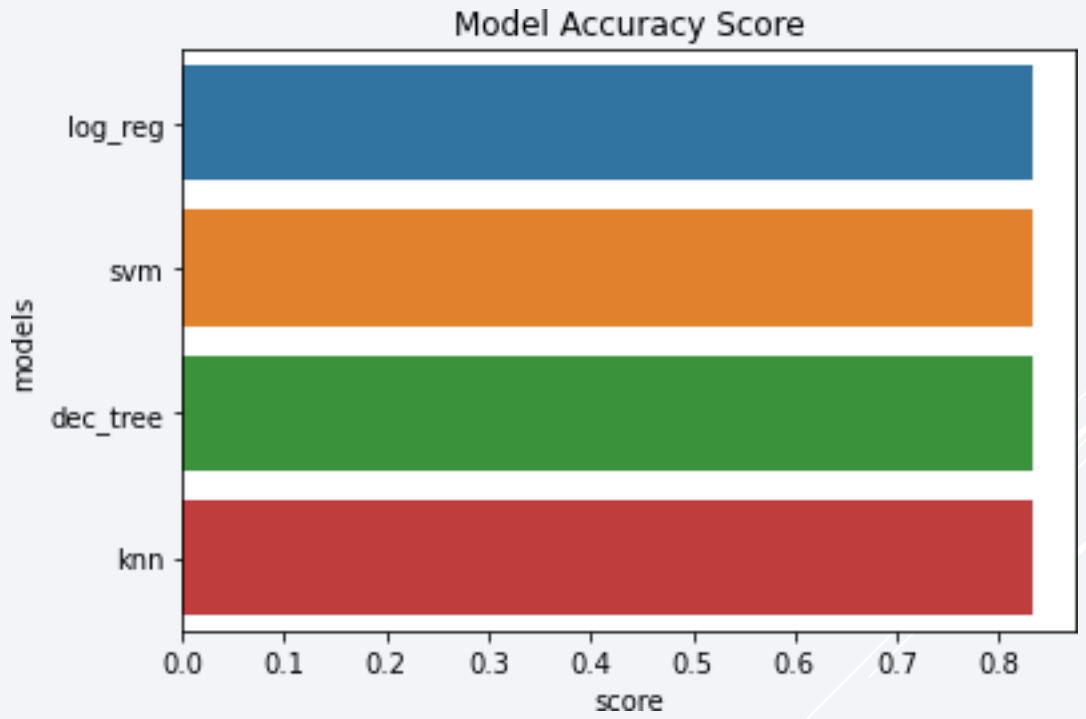


Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Several models were generated to predict factors contributing to landing success. However, all four models prepared had the same accuracy when tested against the test data. This is likely due to the small test sample set of 18, but could be due to no clear reason found within the dataset (such as lack of information on sensor data or material used in rocket composition).



Confusion Matrix for Machine Learning Models on Test Data

- In addition, all 4 models had the same confusion matrix results when used on the test data. To the right is an example, with 12 true positives, 3 true negatives, 3 false positives and 0 false negatives. Therefore, all models overpredict successful landings and should be tuned with a penalty for selecting false positives.
- Having a model that **overpredicts successful landings** (underpredicts failures) in this case is a **liability**, as it would lead to launches with higher likelihoods of failed recovery and, thus, increased cost due to increased rocket loss.

Conclusions

The challenge: Generate a machine learning model for Allon Mask of SpaceY to predict stage 1 rocket launch successes to reduce launch costs.

The method used: Obtained data from SpaceX's public API and web scraping the SpaceX Wikipedia page. Wrangle data and generate dashboards and folium maps to visualize the data .Train several machine learning models to predict successes/failures.

The result: Prepared several machine learning models with accuracies of 83% on test data and recall of 100%. However, the precision sits at 80% due to overestimating successful launches.

The next step: Additional data is required to generate machine learning models with higher accuracy. In addition, model parameters should be adjusted to de-incentivize false positives and risk launch losses.

Thank you!