IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Vishwak M
24-09-2021

Outline

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

Executive Summary

- Data Collection – Space X API, WIKI web scraping
- Data Wrangling- Standardized and captured necessary data.
- Data Analysis – SQL using DB2
- Visualize Data with the seaborn library
- Data was mapped using Folium and Plotly Dash
- Predictive analysis was used to plot and score the best fit model
- Summary of all results

Introduction

- The New Space Race
    - Space X is ahead of the pack with reusable first stage rockets
- Questions
    - What will be the estimated price of each launch?
    - Will the first stage be reused?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Data was collected through the api.spacexdata.com/v4/launches/past.
    - Additionally data was collected through web scraping using Beautiful Soup and Pandas libraries.
- Perform data wrangling
    - Data was wrangled by standardizing the outputs and identifying which data will be further analyzed. Categorical data was restructured to aid in analysis.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - Tested using logistic regression, support vector machine, decision tree classifier, k and nearest neighbor

Data Collection

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

Data Collection – SpaceX API

- Get Request from SpaceX Rest endpoint API:
  - "https://api.spacexdata.com/v4//past"
- Pull Different Boosters, Launch Site , Payload, and Core Data from API
- Filter Data to Only Include Falcon 9 launches

- https://github.com/Vishwak-M/Data-Science-Capstone /blob/master/W1_DataCollectiong.ipynb

```
response = requests.get("https://api.spacexdata.com/v4//past")
```

```python
# Takes the dataset and uses the rocket column to call the API and append the data to the list
def getBoosterVersion(data):
    for x in data['rocket']:
        response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
        BoosterVersion.append(response['name'])
```

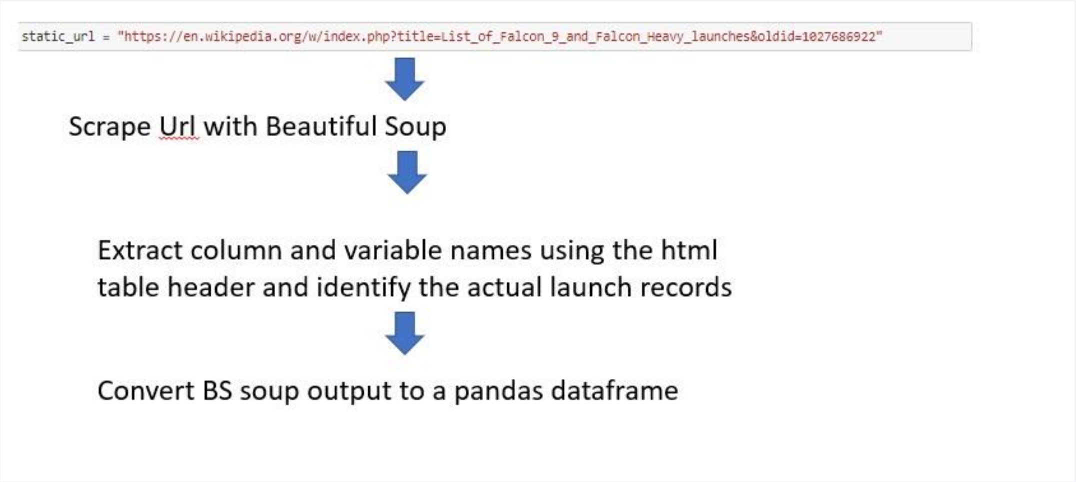
```python
# Takes the dataset and uses the launchpad column to call the API and append the data to the list
def getLaunchSite(data):
    for x in data['launchpad']:
        response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).json()
        Longitude.append(response['longitude'])
        Latitude.append(response['latitude'])
        LaunchSite.append(response['name'])
```

```python
# Takes the dataset and uses the payloads column to call the API and append the data to the lists
def getPayloadData(data):
    for load in data['payloads']:
        response = requests.get("https://api.spacexdata.com/v4/payloads/"+load).json()
        PayloadMass.append(response['mass_kg'])
        Orbit.append(response['orbit'])
```

```python
# Takes the dataset and uses the cores column to call the API and append the data to the lists
def getCoreData(data):
    for core in data['cores']:
            if core['core'] != None:
                response = requests.get("https://api.spacexdata.com/v4/cores/"+core['core']).json()
                Block.append(response['block'])
                ReusedCount.append(response['reuse_count'])
                Serial.append(response['serial'])
            else:
                Block.append(None)
                ReusedCount.append(None)
                Serial.append(None)
            Outcome.append(str(core['landing_success'])+' '+str(core['landing_type']))
            Flights.append(core['flight'])
            GridFins.append(core['gridfins'])
            Reused.append(core['reused'])
            Legs.append(core['legs'])
            LandingPad.append(core['landpad'])
```

Data Collection - Scraping

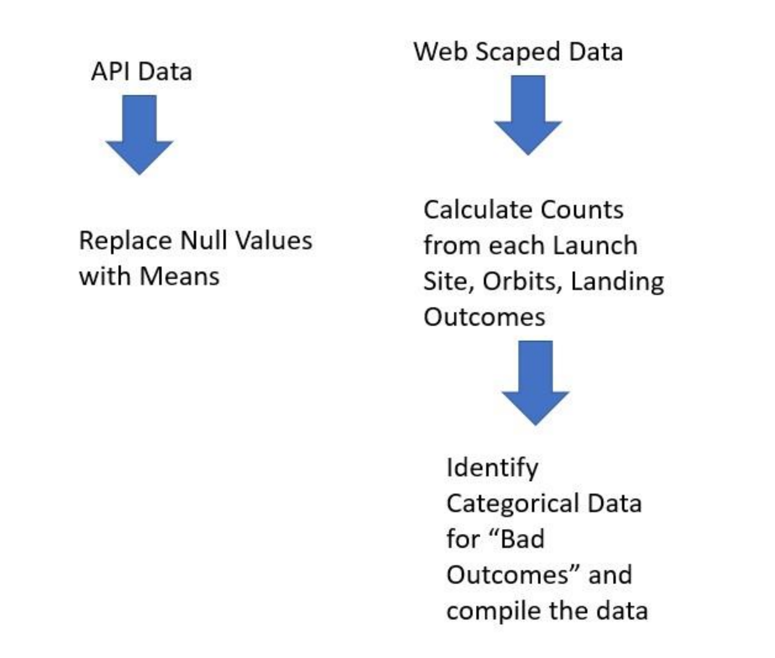
- Supplemental Data collected using the Beautiful Soup and Pandas libraries using web scraping from the following url:
  - "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"



- https://github.com/Vishwak-M/Data-Science-Capstone/blob/master/W1_web scraping.ipynb

Data Wrangling

- Data from the API output was wrangled by normalizing the missing values by replacing null payload mass with mean values.
- Web scraped data was categorically compiled to identify the data required to analyze. This include identifying data that could be considered "bad" i.e., where the first stage rocket was not able to be reused. This data could then be related to orbital stage and launch sites which all have different percentages of "bad" outcomes.
- https://github.com/Vishwak-M/Data-Science-Capstone /blob/master/W1_data%20wrangling.ipynb



EDA with Data Visualization

- Plots
    - Scatter Plot for Payload Mass vs Flight Number
        - The heavier the payload, the lest likely the first stage will be recovered.
    - Scatter Plot for Launch Site vs Flight Number with a comparison between classes
        - To compare trends between booster recovery across flight number and launch sites
    - Scatter Plot for Launch Site vs Payload Mass with a comparison between classes
        - To compare trends between booster recovery across Payload Mass and launch sites
    - Bar Plot for Orbit vs Average Success Rate
        - To compare trends in orbit type and success rate of booster recovery
    - Scatterplot for Orbit vs Flight Number
        - Find trends in the Orbit type over the course of different flights
    - Scatterplot for Orbit vs Payload Mass
        - Find trends in Orbit type and mass of the payload
    - Lineplot for Success Rate over Year
        - See if there is a trend for success over time
- https://github.com/Vishwak-M/Data-Science-Capstone/blob/master/W2_Data%20Visualization.ipynb

EDA with SQL

- Queries
  - select distinct launch_site from spacextbl
  - select * from spacextbl where launch_site like 'CCA%' limit 5
  - select sum(payload_mass kg_) from spacextbl where customer = 'NASA (CRS)'
  - select avg(payload_mass kg_) from spacextbl where booster_version like 'F9 v1.1%'
  - select min(date) from spacextbl where landing outcome = 'Success (ground pad)'
  - select Booster_Version from spacextbl where Landing Outcome Like '%drone%' and payload_mass kg_> 4000 and payload_mass kg_<6000
  - select mission_outcome, count(mission_outcome) from spacextbl group by mission_outcome
  - with cte as
    (select booster_version, payload_mass_kg_ mass from spacextbl),
    m as
    (select max(payload_mass_kg_) mass from spacextbl)
    select cte.booster_version from cte, m where cte.mass = m.mass
  - select landing outcome, booster_version, launch_site from spacextbl where landing outcome = 'Failure (drone ship)' and date like '2015%'
  - select landing outcome, count(landing outcome) c from spacextbl where Date between '2010-06-04' and '2017-03-20'
    group by landing outcome order by c desc
- https://github.com/Vishwak-M/Data-Science-Capstone
  /blob/master/W2_EDAwithSQL.ipynb

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Map Identifying the launch locations with a count cluster marker for number of launches.
  - The cluster market location allows us to easily visualize on a map where and how many launches occurred (marker and circles).
  - I added a mouse point marker to easily identify mouse coordinates for user convenience.
  - Created a marker, line, and label to identify the Atlantic coastline from the mouse point (launch location). This shows the line, and the distance to the second point.
  - Create a second marker, line, and label to identify the closest highway from the mouse point (launch location). This shows the line, and the distance to the second point

- https://github.com/Vishwak-M/Data-Science-Capstone /blob/master/W3_folium_maps.ipynb

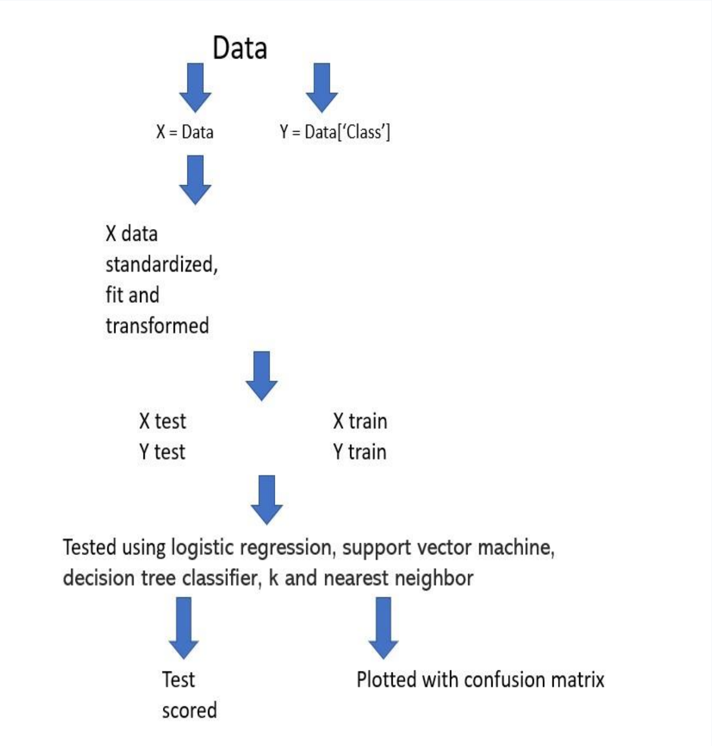
Build a Dashboard with Plotly Dash

- Explain why you added those plots and interactions
- Dropdown
  - Used select specific launch sites or all of them combined
- Pie Chart
  - If individual launch sites chosen, visualize success vs failures for launches
  - If All is selected, show the launches at each site
- Payload Range Slider
  - Select the payload range for the scatter plot
- Scatter Plot
  - Visualize the success or failure of a launch across payload size with Booster Version as a variable
- https://github.com/Vishwak-M/Data-Science-Capstone/blob/master/spacex_dash_app.py

Predictive Analysis (Classification)

- Predictive Analysis was used using pandas, seaborn, numpy, matplotlib, and sklearn libraries.
- Data was loaded into a pandas dataframe and the launch outcome class was loaded into the a numpy array.
- Data was standardized and transformed using “preprocessing.StandardScaler()” to standardize the data.
- scaler = preprocessing.StandardScaler().fit(X) and X=scaler.transform(X) was used to fit and transform the data.
- The data was split into train and test arrays with 18 test samples.
- A logistic regression, support vector machine, decision tree classifier, k and nearest neighbor were used to find the best fitting parameters.
- The method “score” was used to calculate the accuracy and plot with a confusion matrix.

- https://github.com/Vishwak-M/Data-Science-Capstone/blob/master/W4_machi ne%20learning.ipynb



Results

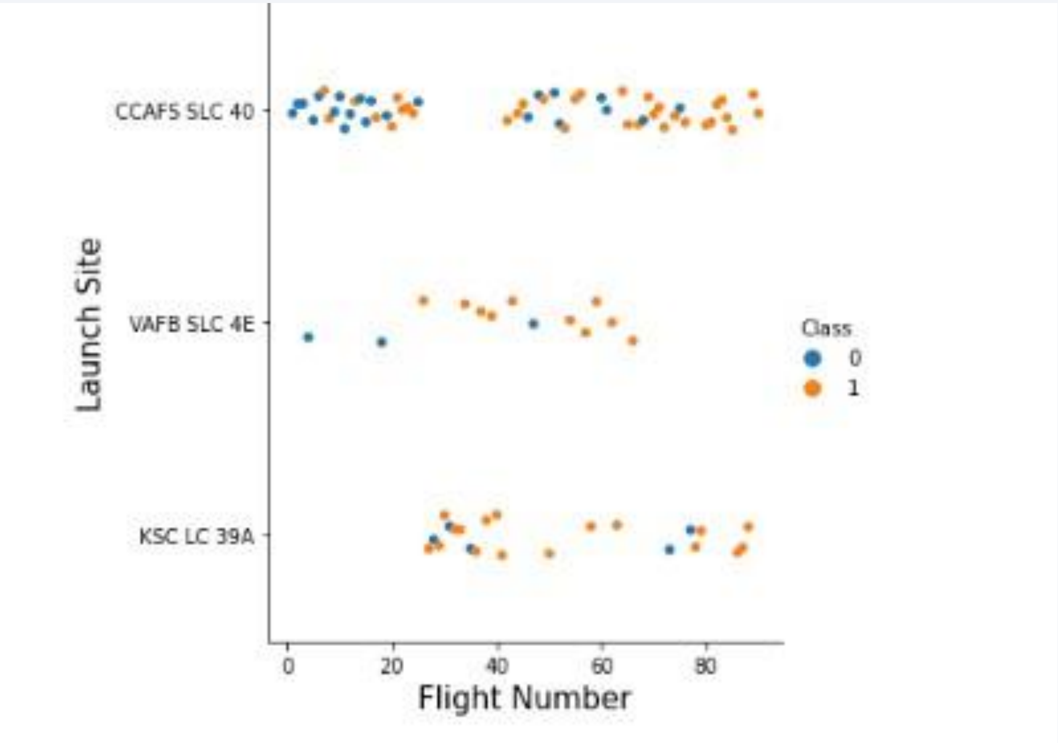
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

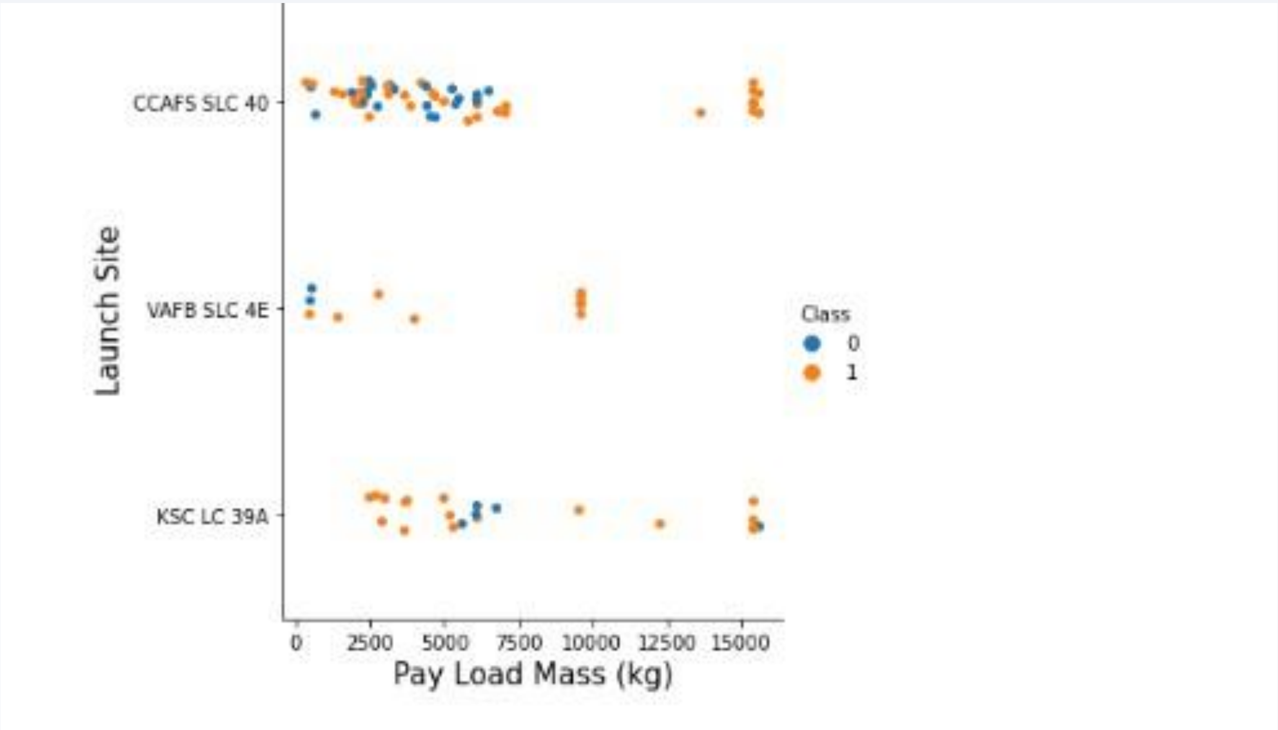
Flight Number vs. Launch Site

- Highest Number of Launches Occurred at CCAFS SLC 40
- From this figure, it would be difficult to identify success vs failure ratios.



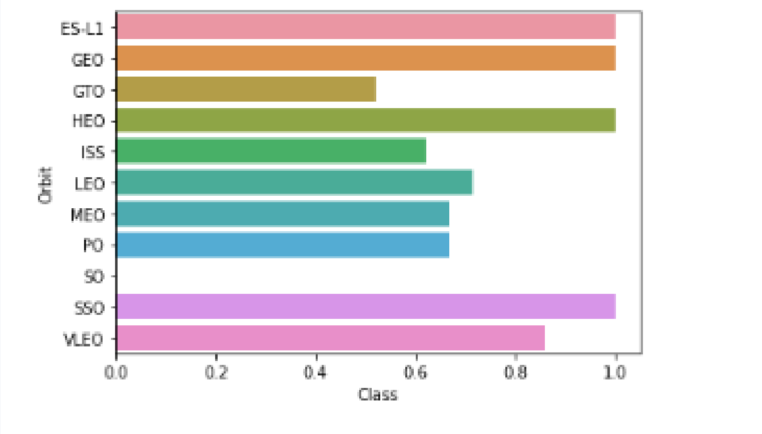
Payload vs. Launch Site

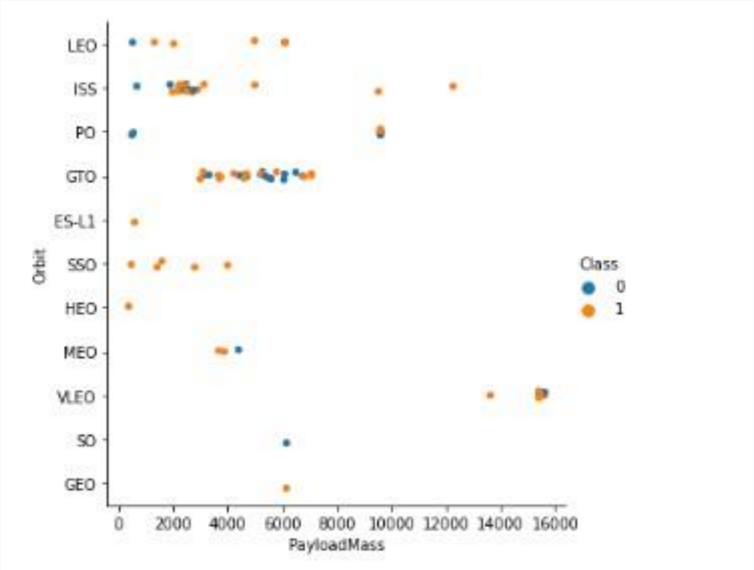
- Most payloads are < 7500 kg.
- Payloads >7500 were not as successful



Success Rate vs. Orbit Type

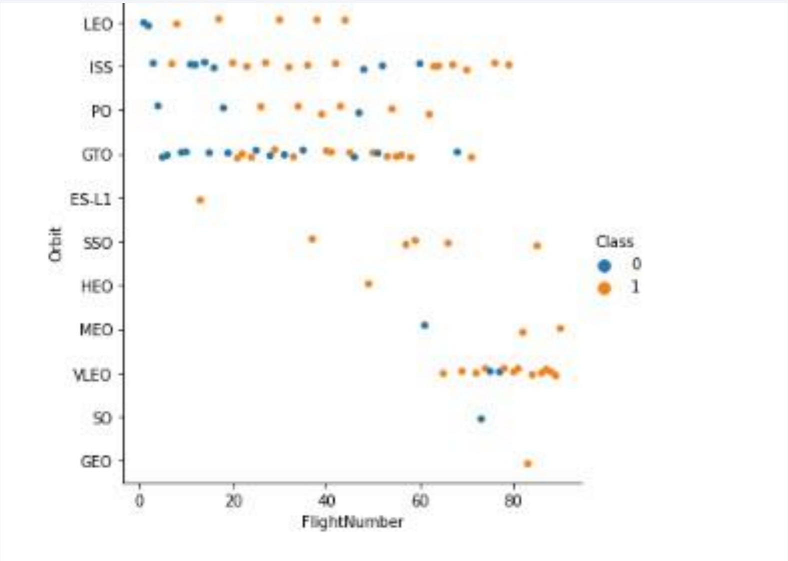
- GTO and ISS orbits appear to be the most common.
- There appears to be a possible correlation to Payload mass ranges and orbit type
- There appears to be a correlation between successful launches and orbit type





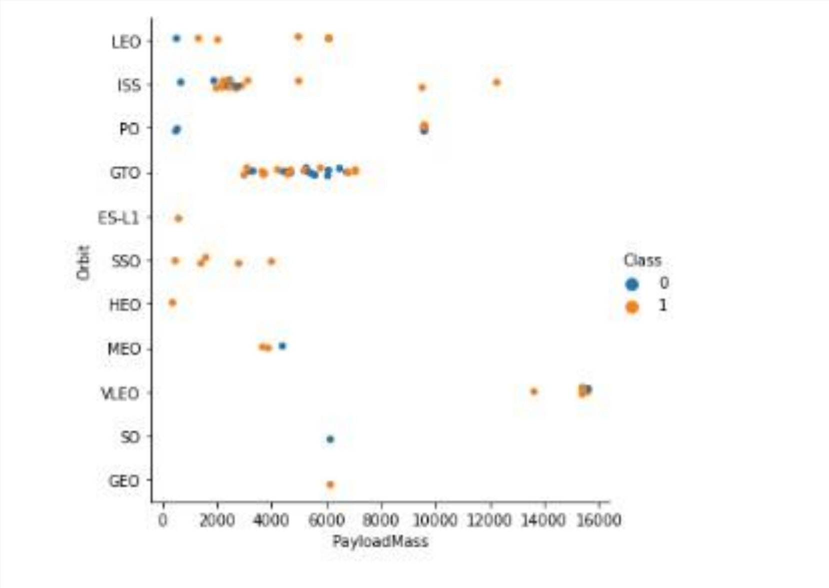
Flight Number vs. Orbit Type

- There appears to be a correlation to flight number and Orbit, however certain orbit types appear continuous for large ranges of flight numbers.



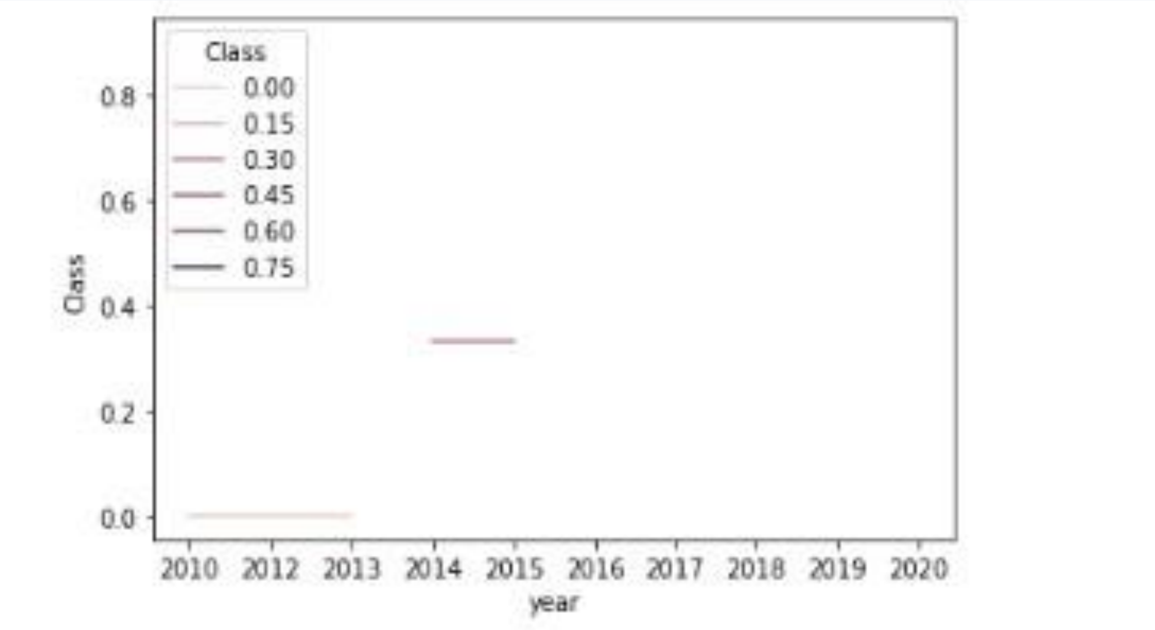
Payload vs. Orbit Type

- There appears to be a possible correlation to Payload mass ranges and orbit type
- There appears to be a correlation between successful launches and orbit type



Launch Success Yearly Trend

- We a clear line in a higher count of successful launches at later dates.



All Launch Site Names

- The following are the unique launch sites recorded from the data.

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

Launch Site Names Begin with 'CCA'

- The following are first 5 results (LIMIT 5) for Launch Sites beginning with CCA
  - This method allows for us to quickly validate query results

```
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)
2012-08-10	00:35:00	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
2013-03-12	22:41:00	F9 v1.1 CCAFS LC-40	SES-8	3170	GTO	SES	Success No attempt
```

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
    - This value presents the entire payload for each launch aggregated.

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- This value is the average payload mass (kg) carried by the F9 v1.1 booster.

```
3226
```

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- The following date was the first successful recovery of a phase 1 booster from the ground launch pad.

```
2017-01-05
```

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
- The following boosters have had successful landings on a drone ship within the payload range of 4000 and 6000

```
F9 FT B1020
F9 FT B1022
F9 FT B1031.2
```

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- The following represents the successful vs failure mission outcomes –
  - Not necessarily dependent on the landing outcome

```
Success 44
Success (payload status unclear)        1
```

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- The following Booster versions have all carried the maximum payload mass

```
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
```

2015 Launch Records

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

Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40

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

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

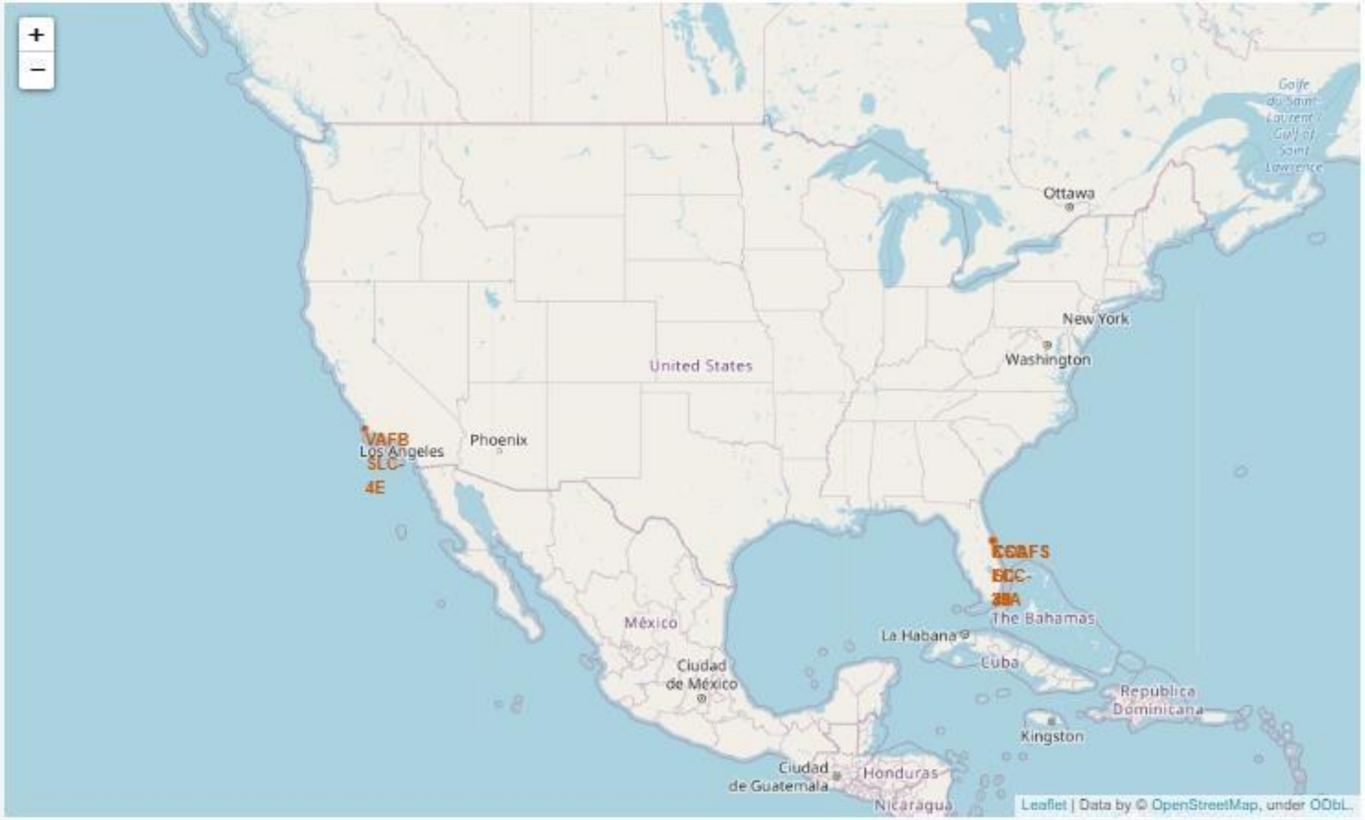
No attempt	7
Failure (drone ship)	2
Success (drone ship)	2
Success (ground pad)	2
Controlled (ocean)	1

Section 4

Launch Sites Proximities Analysis

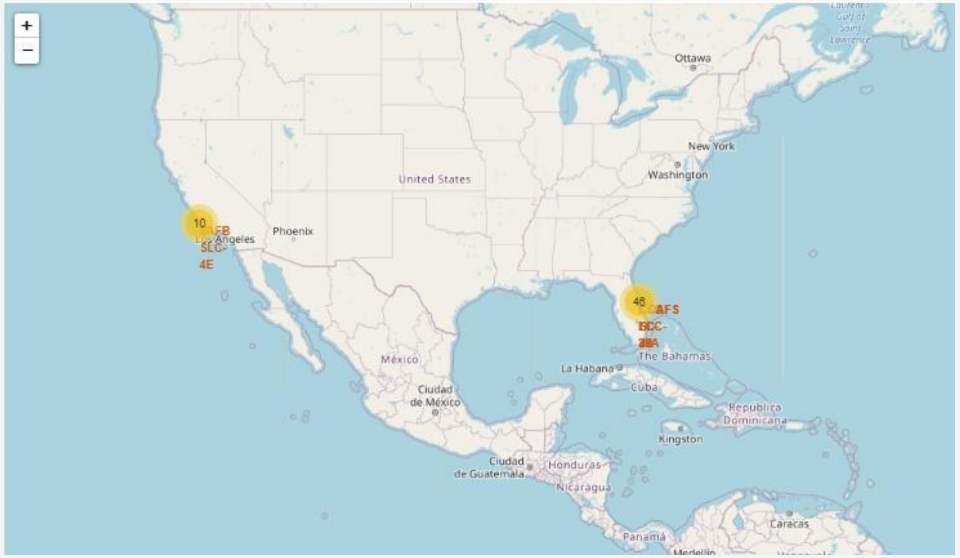
Location of All Launch Sites

Circle Markers and Labels Identifying the Location of the Launch Sites



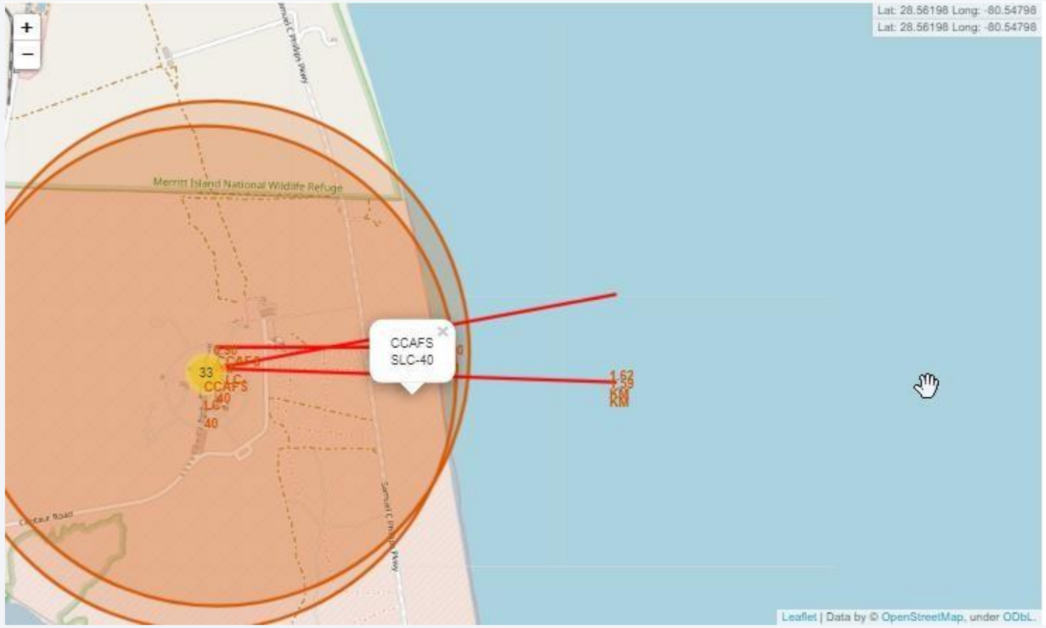
Successful Launch Outcomes from Each Site

- Cluster markers added to identify the number of successful launches of each site



Markers to Specific GPS Points

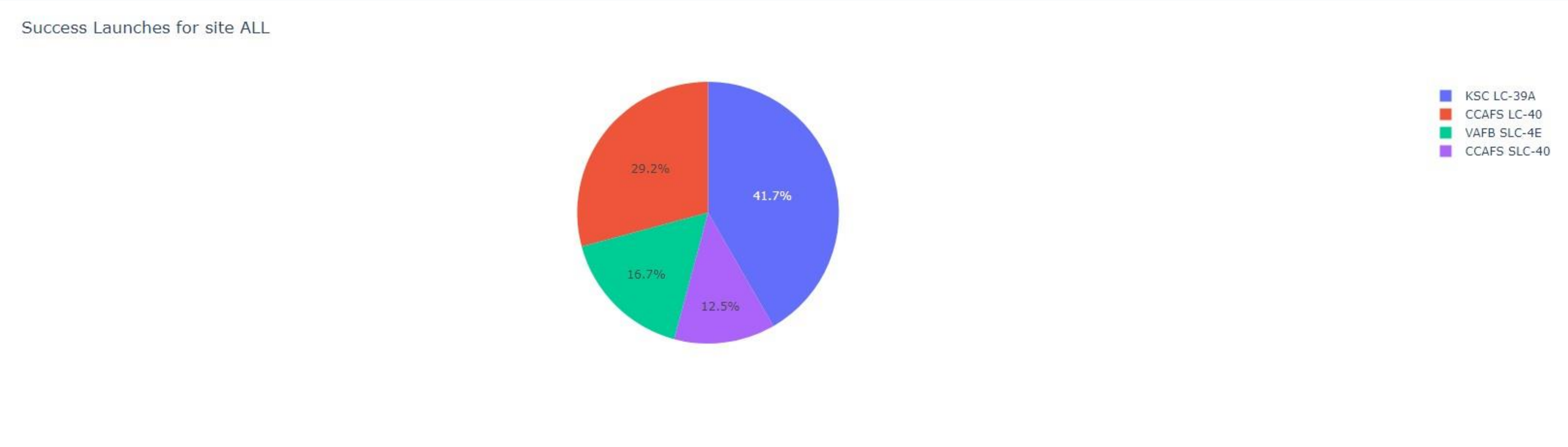
- Markers Generated to identify a second GPS point location.
- A line was created to that second point, along with a distance measurement.



Section 5

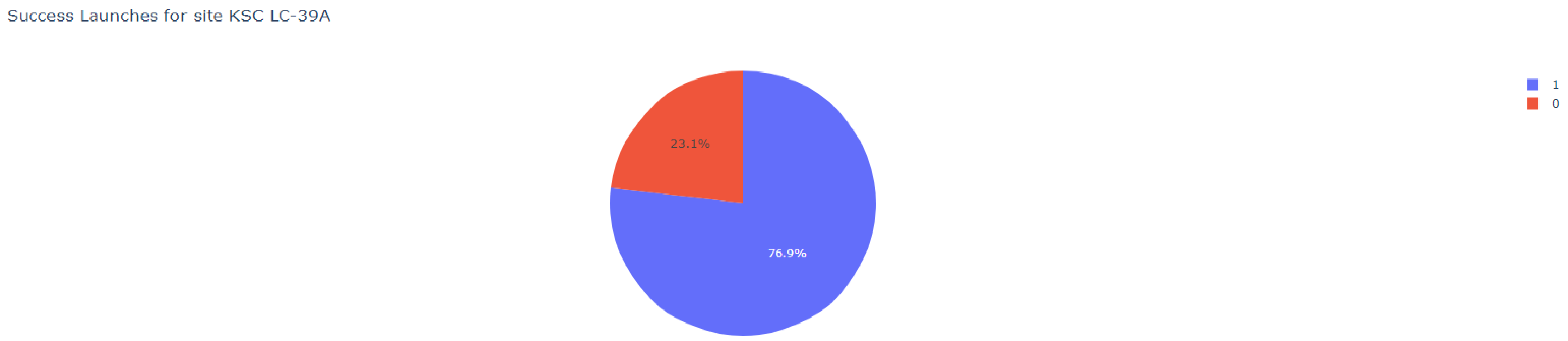
Build a Dashboard with Plotly Dash

Success Launches Across All Sites



- Majority of Successful Launches were from KSC LC39A
- Least number of Successful Launches were from CCAFS SL-40

Launch Success Percentage at KSC LC-39A



- 76.9% of Launches were successful at KSC LC-39A

All Site – Successful Launches Across Booster Version and Payload

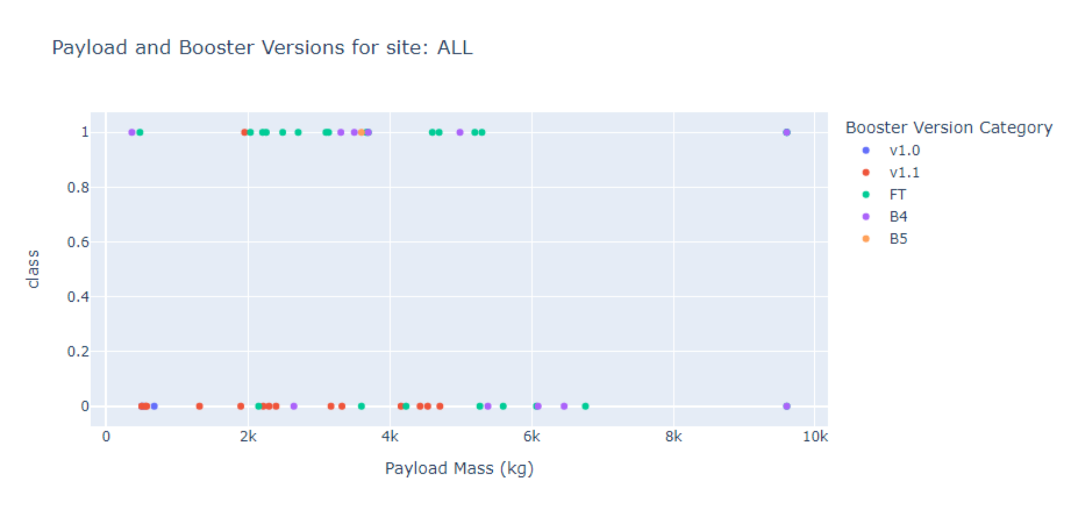


Fig 1

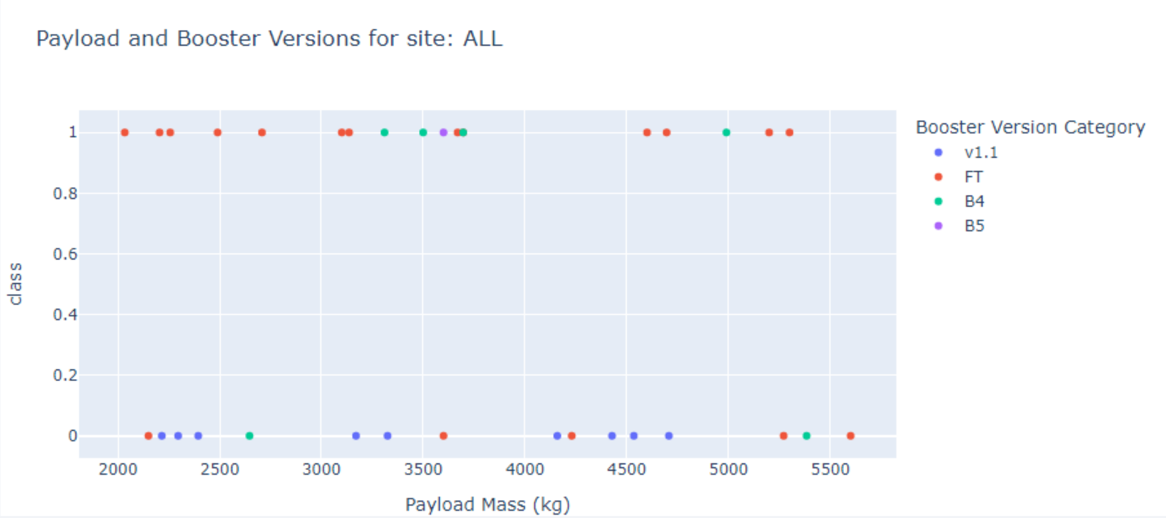


Fig 2

- Fig1 Shows us that the larger the Payload, the least amount of successful launches were seen
- Fig 2 Focuses on the most common payload amounts, with the most successful launches occurring sub 4000 kg. The FT Booster also appears to be the most successful version.

Section 6

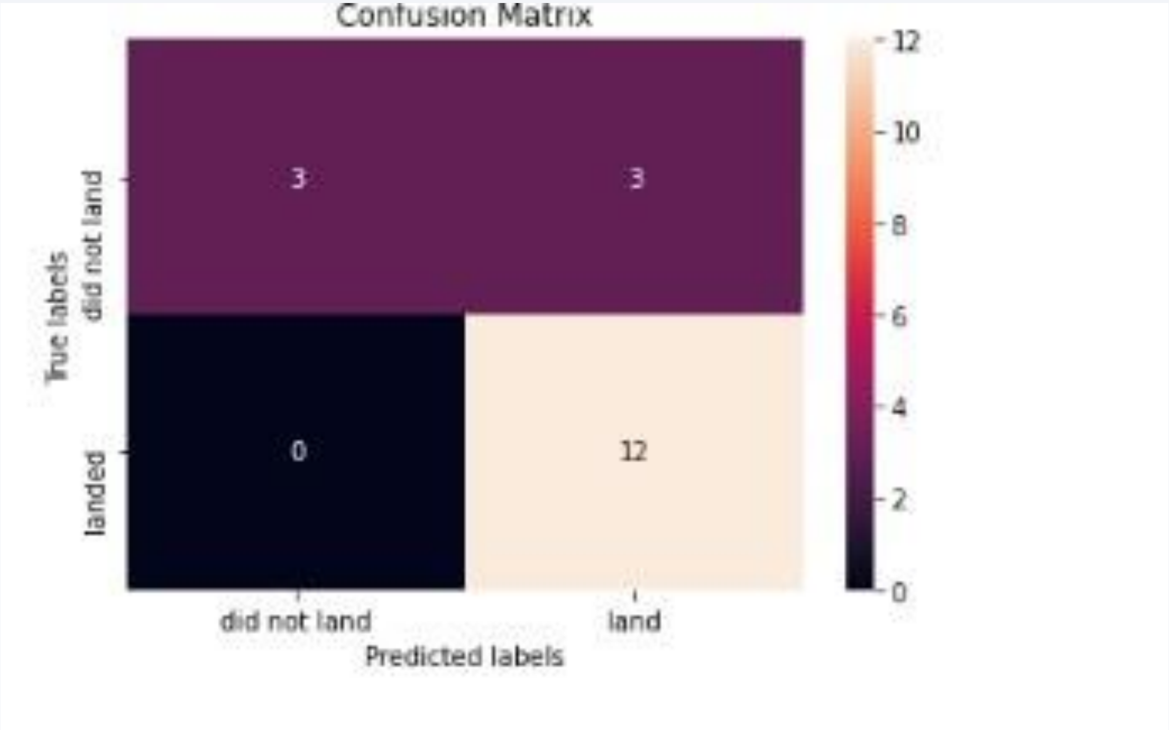
Predictive Analysis (Classification)

Classification Accuracy

- All models had even output scores

Confusion Matrix

- With these predictive models, we had the following:
  - 3 True Negatives
  - 3 False Positives
  - 0 False Negatives
  - 12 True Positives



Conclusions

- Exploratory Analysis
  - Highest number of launches occurred at CCAFS SLC 40
  - Most payloads are < 7500 kg.
  - Payloads >7500 were not as successful
  - GTO and ISS orbits appear to be the most common.
  - There appears to be a possible correlation to Payload mass ranges and orbit type
  - There appears to be a correlation between successful launches and orbit type
- The most successful boosters are the F9 boosters
- Dashboards and maps provide and an excellent user interface to visualize and understand data concepts and relationships.
- Predictive Analysis
  - All predictive models resulted in similar results. As such, any of the 4 models tested could be used to predict successful mission outcomes.

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

Thank you!