Winning Space Race with Data Science

Michael James Thompson
15/03/2022

Outline

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

Executive Summary

- Summary of Methodologies
  - Data Collection
  - Data Wrangling
  - EDA with Data Visualisation
  - EDA with SQL
  - Folium Interactive Map
  - Plotly Dashboard
  - Predictive Analysis
- Summary of All Results
  - Exploratory Data Analysis Results
  - Analytics Screenshots
  - Predictive Analysis Results

Introduction

- Project Background and Context:

A new era of the space race is upon us! However unlike the previous era, the private sector has become the driving force behind this race. Several companies are trying to make space travel more affordable. SpaceX is chief among those companies.

SpaceX’s Falcon 9, a two stage rocket with a reusable first stage, costs $62 million dollars per launch. This is significantly cheaper then what rival space transportation companies offer; generally prices start a $165 million per launch.

As an up-and-coming space transportation SpaceY, we wish to learn from SpaceX’s success. If we can determine the probability that the first stage of a Falcon9 rocket will land successfully, we can determine the overall cost of the launch. This information can used by us to develop our own rival rocket platform.

- Problems you want to find answers:
  - Is there a correlation between a rockets data variables and the likelihood of a successful launch?
  - What are the best conditions to ensure the best results and rate of successful landing?

Section 1

Methodology

Methodology

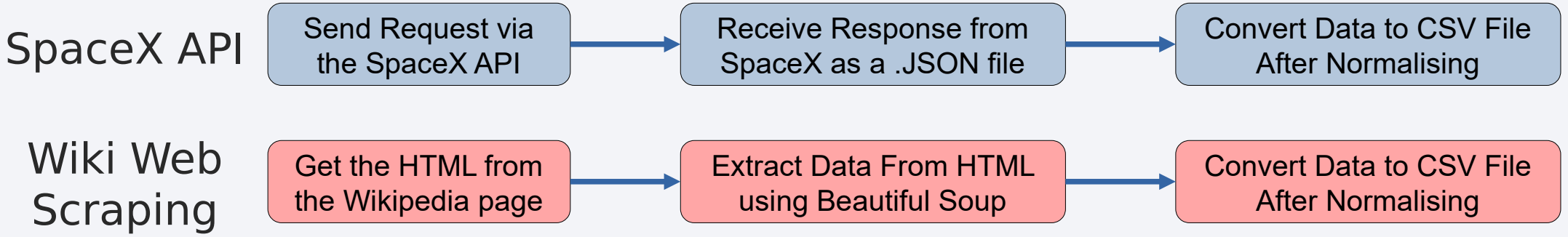
Executive Summary

- Data collection methodology:
  - Data was gathered from the following sources:
    - SpaceX API
    - Web scraping the Falcon 9 and Falcon Heavy Launches Wikipedia Page
- Perform data wrangling
  - Converted the outcomes into training labels for successful and unsuccessful landings
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Find best parameter for Decision Tree, Logistic Regression, K Nearest Neighbour and SVM

Data Collection

For this project data was collected from two sources/ via to methods:

- SpaceX API response. The data columns include: FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude.
- Wikipedia Web Scraping. The data columns include: Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time.



Data Collection – SpaceX API

1) Send a request for rocket launch data from the Space API and receive a response

```
spacex_url="https://api.spacexdata.com/v4/launches/past"

response = requests.get(spacex_url)
```

2) Convert response into a JSON format

```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

3) Clean the data (using the pre-designed functions)

```
getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

4) Create a dictionary from the JSON formatted data, then convert dictionary to Pandas Dataframe

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

5) Filter Data and export to CSV file

```
data_falcon9 = launchData[launchData['BoosterVersion'] != 'Falcon 1']

data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

https://github.com/MichaelMJTH/DataScienceCapstoneRepo/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

1) Send a request for rocket launch data from the Space API and receive a HTML response

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
response = requests.get(static_url)
```

2) Create a Beautiful Soup object from the HTML response

```
beautifulSoup = BeautifulSoup(response.text, 'html5lib')
```

3) Extracting tables from Beautiful Soup object

```
html_tables = beautifulSoup.find_all(name='table')
```

4) Getting Columns names and creating empty dictionary

```
column_names = []

for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    print(name)
    column_names.append(name)
```

```
launch_dict= dict.fromkeys(column_names)
```

5) Assign every dictionary key a list and then populate with launch records data

```
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6) Convert dictionary to a Pandas Dataframe

```
df=pd.DataFrame(launch_dict)
```

7) Export Dataframe as a CSV file

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

https://github.com/MichaelMJTH/DataScienceCapstoneRepo/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

1) Calculate the number of launches on each site.

```
df["LaunchSite"].value_counts()
```

2) Calculate the number and occurrence of each orbit.

```
df.Orbit.value_counts()
```

3) Calculate the number and occurrence of mission outcome per orbit type.

```
landing_outcomes = df.Outcome.value_counts()
```

4) Create a new column within the data to sort the landing outcomes into successes and failures. This new column is labelled “Class”. (More details on next slide)

```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
landing_class = []
for outcome in df['Outcome']:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
df['Class']=landing_class
```

5) Calculate the mean of the landing class in order to find success rate.

```
df["Class"].mean()
```

6) Convert dataframe/ export data as CSV file.

```
df.to_csv("dataset_part_2.csv", index=False)
```

https://github.com/MichaelMJTH/DataScienceCapstoneRepo/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

Data Wrangling

At this point most opaque column of data in need of wrangling was the “Outcome” column. It had eight options depending on whether the outcome was successful, the landing was successful, what type of landing:

- True Ocean: The mission outcome successfully landed in a specific region of the ocean.
- False Ocean: The mission outcome unsuccessfully landed in a specific region of the ocean.
- True RTLS: The mission outcome was successfully landed on a ground pad.
- False RTLS: The mission outcome was unsuccessfully landed to a ground pad.
- True ASDS: The mission outcome was successfully landed to a drone ship.
- False ASDS: means the mission outcome was unsuccessfully landed to a drone ship.
- None ASDS and None None: These represent a failure to land.

In order to use these results in machine learning we need to convert these results into a binary column of result. Success = 1. Failure = 0.

https://github.com/MichaelMJTH/DataScienceCapstoneRepo/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualisation

We used a number of different chart types in order to visualise the data whilst exploring it:

- Scatter Plots: Used to observe the relationship between variables, also known as the correlation. These are useful when trying to display a large dataset.
  - Flight Number vs Payload Mass (KG)
  - Flight Number vs Launch Site
  - Launch Site vs Payload Mass (KG)
  - Flight Number vs Orbit Type
  - Payload Mass (KG) vs Orbit Type
- Bar Chart: Used for presenting categorical data as bars, where the length of each bar is proportional to the value of the bar.
  - Orbit Type vs Success Rate (%)
- Line Chart: Used to display trends in data. Lines are used to connect individual data points. Usually used to convey quantitative data over a specified continuous time period.
  - Years vs Success Rate (%)

https://github.com/MichaelMJTH/DataScienceCapstoneRepo/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

After loading the dataset in a Db2 database table, the following questions were answered by creating SQL queries:

- Display the names of the unique launch sites in the space mission.
- Display 5 records where launch sites begin with the string 'CCA'.
- Display the total payload mass carried by boosters launched by NASA (CRS).
- Display average payload mass carried by booster version F9 v1.1.
- List the date when the first successful landing outcome in ground pad was acheived.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- List the total number of successful and failure mission outcomes.
- List the names of the booster_versions which have carried the maximum payload mass.
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015.
- 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.

https://github.com/MichaelMJTH/DataScienceCapstoneRepo/blob/main/jupyter-labs-eda-sql-coursera.ipynb

Build an Interactive Map with Folium

The following object were added to folium map depicting the US:

- Markers that show the location of all launch sites on the map.
- Markers that show successful and unsuccessful launches at each launch site.
- Lines showing the distance between a launch site and neighbouring places of interest. These include:
  - Proximity to railway lines
  - Proximity to motorways/ highways
  - Proximity to coastlines
  - Proximity to the nearest city

https://github.com/MichaelMJTH/DataScienceCapstoneRepo/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

An interactive dashboard has been created and includes two visualisations:

Pie Chart:

- This pie chart is used to show the total number of successful launches split by the different launch sites.
- The interactive element of this chart allows us to see either:
  - The total successful launches with each slice proportional to number of success in each site.
  - Show the success rate of an individual launch site.

Scatter Plot:

- The scatter plot shows the correlation between landing outcomes and payload mass (KG).
- It can show how success depends on different factors such as, launch point, payload mass and booster version categories.
- The interactive element of this chart includes:
  - All sites/ individual sites drop down.
  - Payload Mass slider with two inputs.

https://github.com/MichaelMJTH/DataScienceCapstoneRepo/blob/main/spacex_dash_app.py

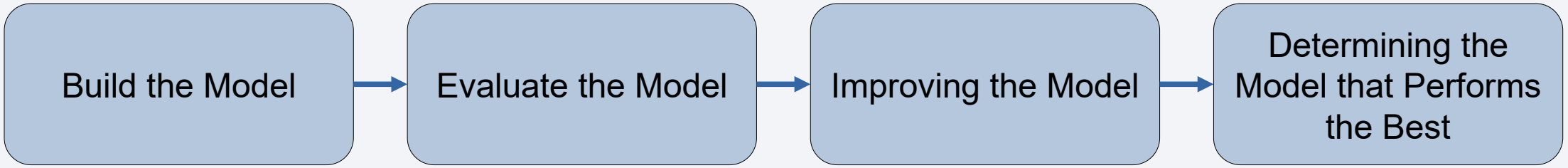
Predictive Analysis (Classification)

Performing Exploratory Data Analysis (EDA) and Determining Training Labels.

- We must create a column for the class
- We must standardize the data
- We must split the data into training data and test data

Then we need to find the best Hyperparameter for SVM, Classification Trees and Logistic Regression.

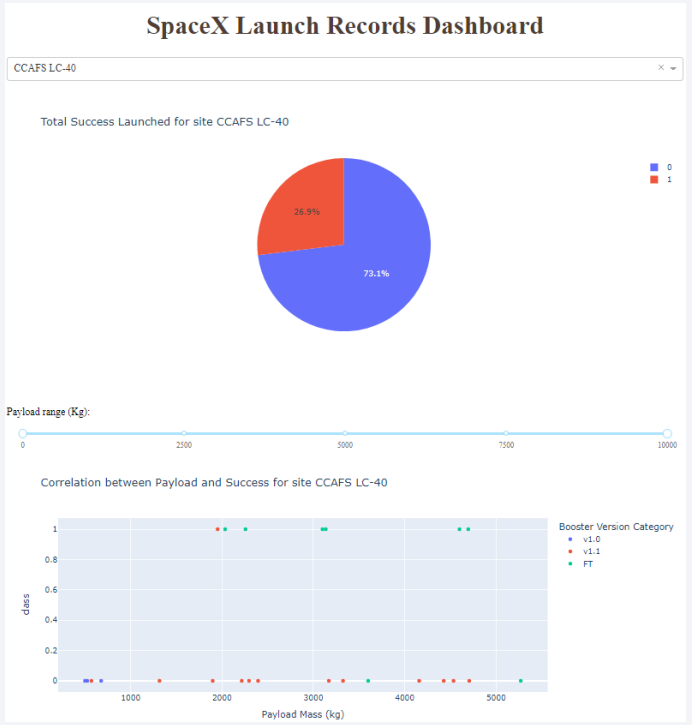
- Finding out which method performs the best using test data

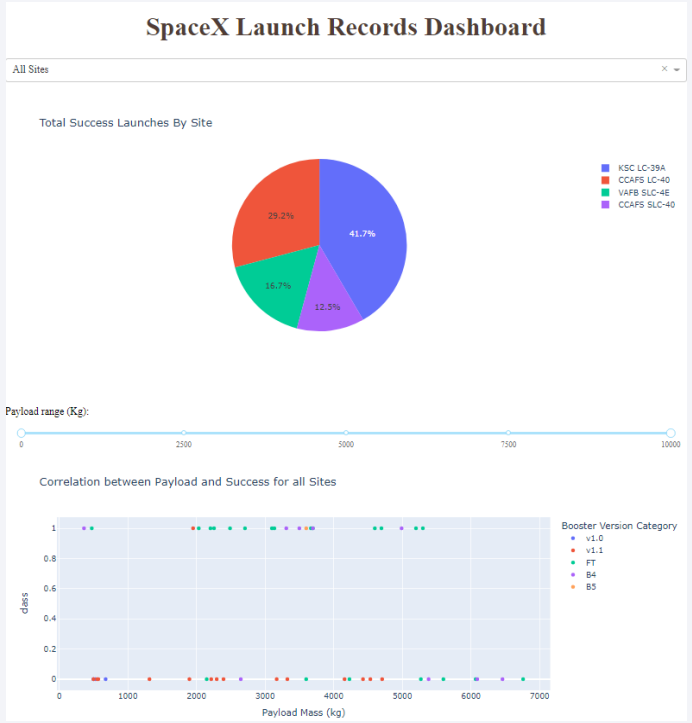


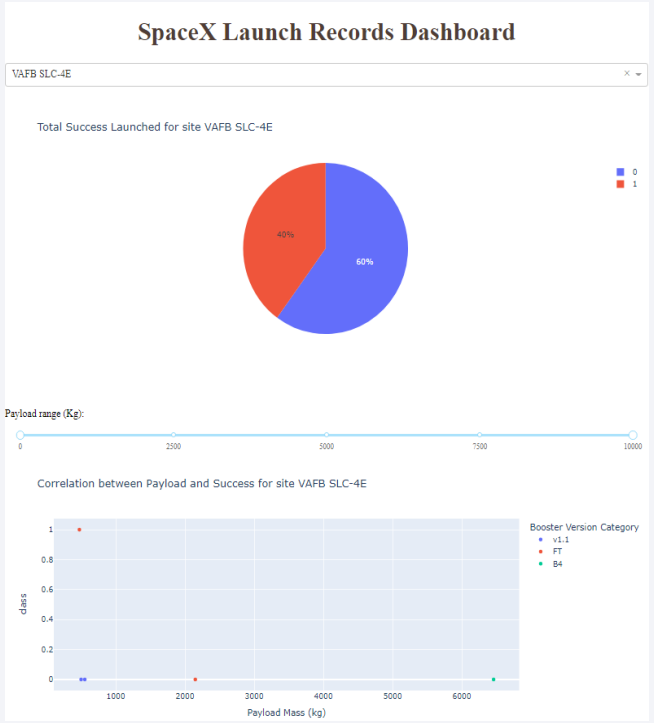
https://github.com/MichaelMJTH/DataScienceCapstoneRepo/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- The results from our predictive data analysis has shown that all methods (Decisions Trees, SVM, K-Nearest Neighbour, Logisitic Regression) all return accuracy scores of 83%, within decimal places of each other.
- The results of the exploratory data analysis (EDA), including visualisations, SQL, and Folium maps will be discussed in the following section.
- Below are some screenshots of to act as a preview of the interactive visualisation as shown with the Ploty Dash dashboard.





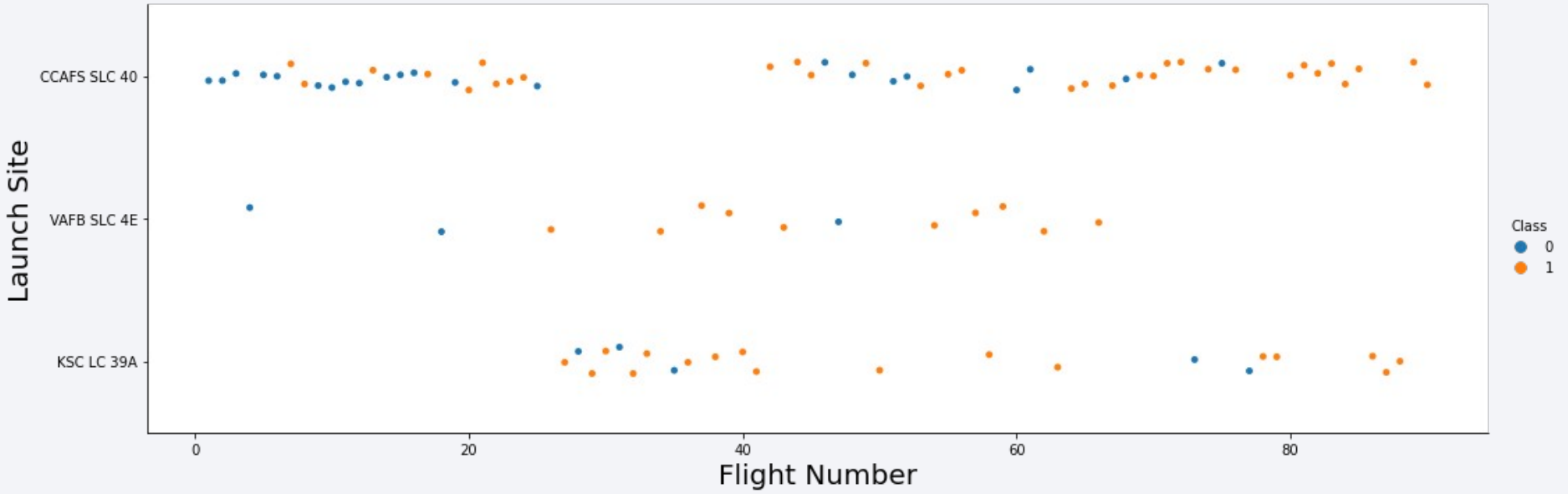


Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

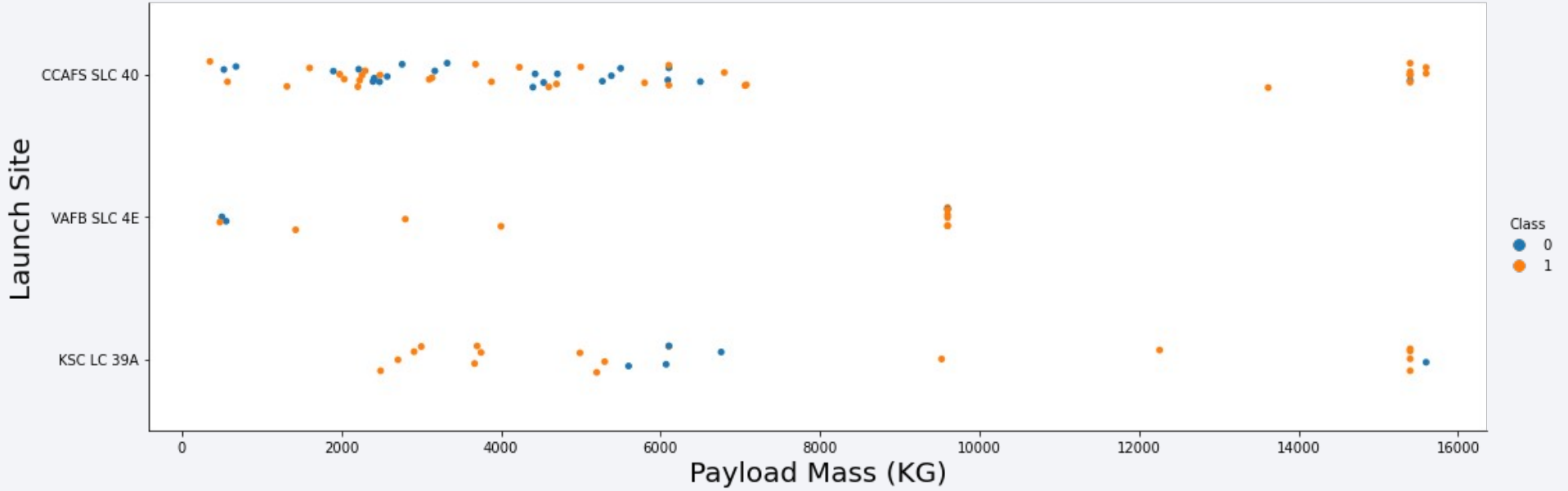
- The graph below is a scatter plot of Flight Number vs Launch Site. Class 0 blue points represent unsuccessful landings. Class 1orange points represent successful landings.
- As the flight number is linearly progressive with time, this plot shows that as the number of landings increase, the rate of successful landings increase. In particular the success rate seems to improve a lot after flight 20.



Payload vs. Launch Site

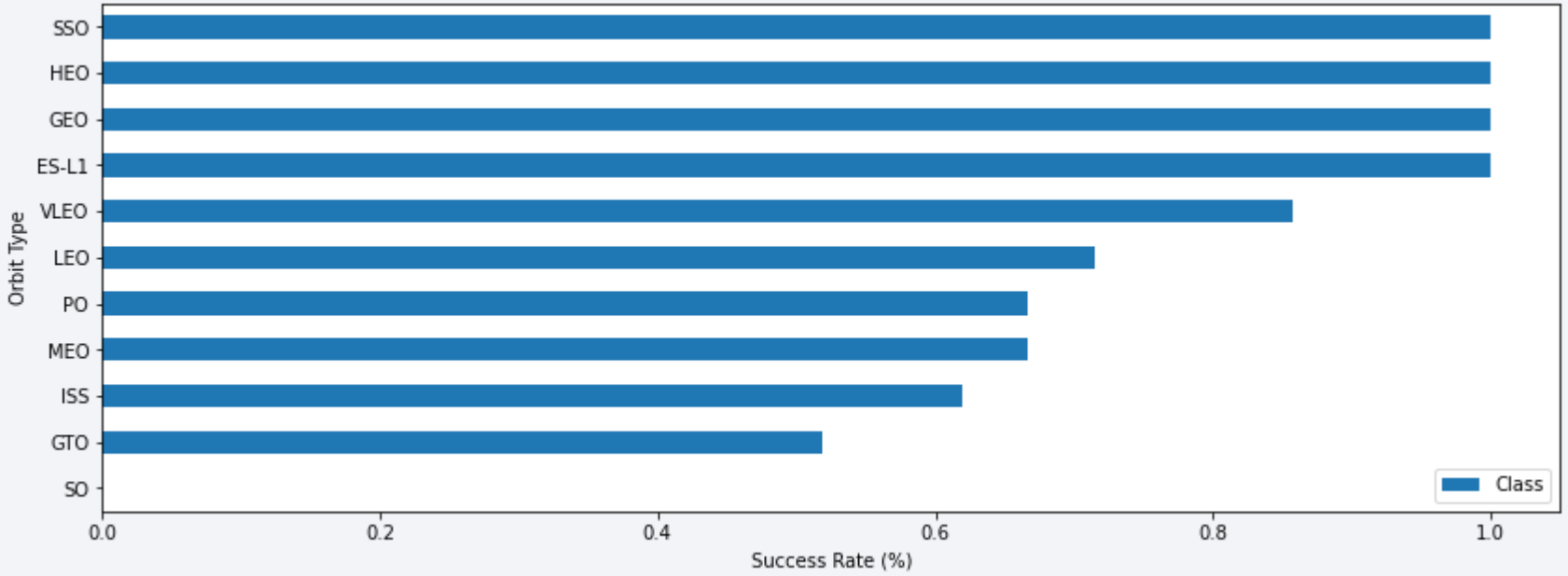
The graph below is a scatter plot of Payload Mass (KG) vs Launch Sites. Class 0 blue points represent unsuccessful landings. Class 1orange points represent successful landing.

- It initially looks like that there is a higher chance of a successful landing with higher mass payloads. However there is less data with less spread at higher masses then lower ones, so it’s hard to say if there is a proper correlation.



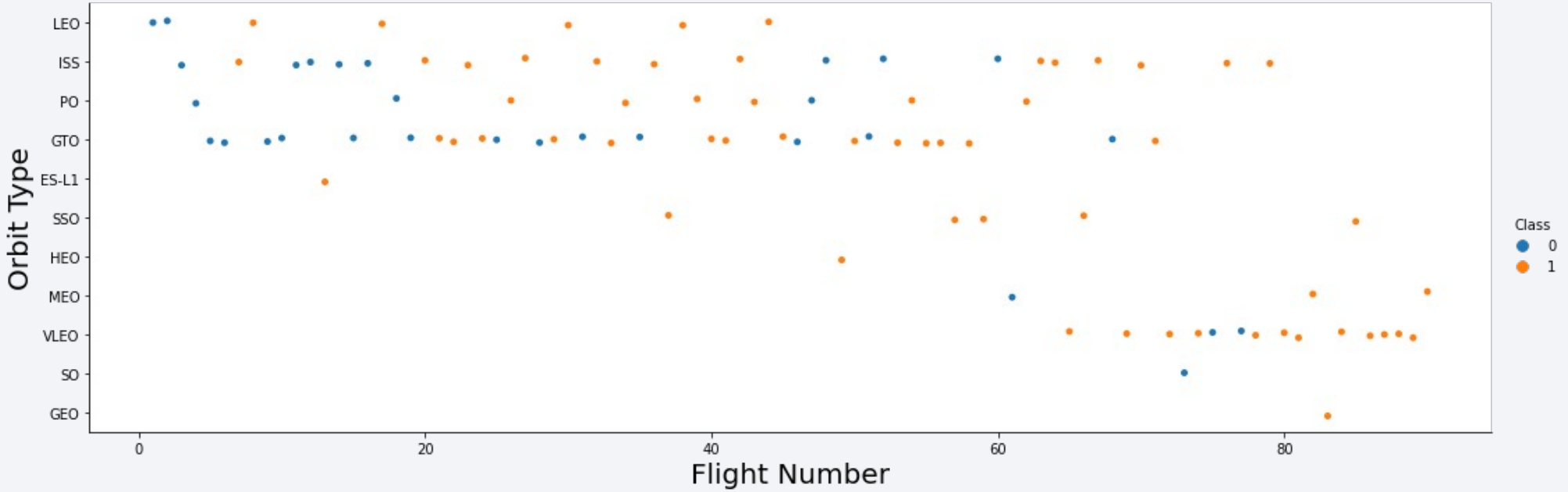
Success Rate vs. Orbit Type

- The bar chart shows Success rate vs Orbit Type.
- The most successful orbit types appear to be SSO, HEO, GEO, and ES-L1. Each of these have a success rate of 100%.
- The least successful orbit type was GTO with only a 50% success rate. I consider the SO launch to be an outlier as there was only one failed landing.



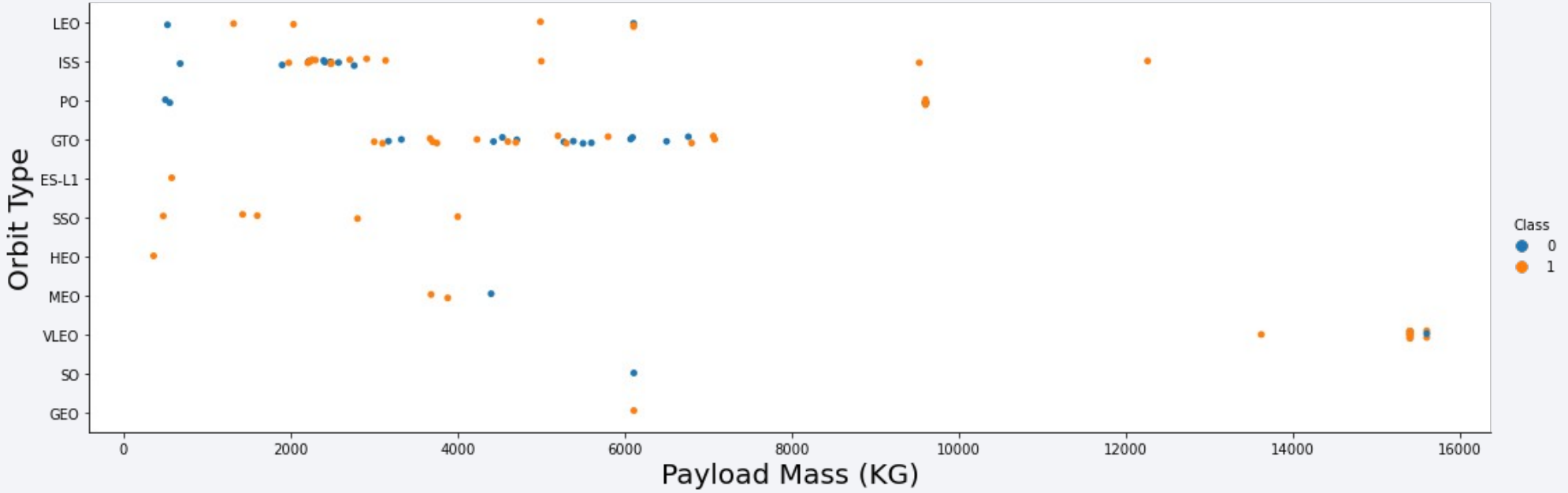
Flight Number vs. Orbit Type

- The graph below is a scatter plot of Flight Number vs Orbit Type. Class 0 blue points represent unsuccessful landings. Class 1orange points represent successful landings.
- You should see that for the LEO orbit the success rate improves with number of flights. However, there seems to be no relationship between flight number and success rate for orbits like GTO and VLEO.



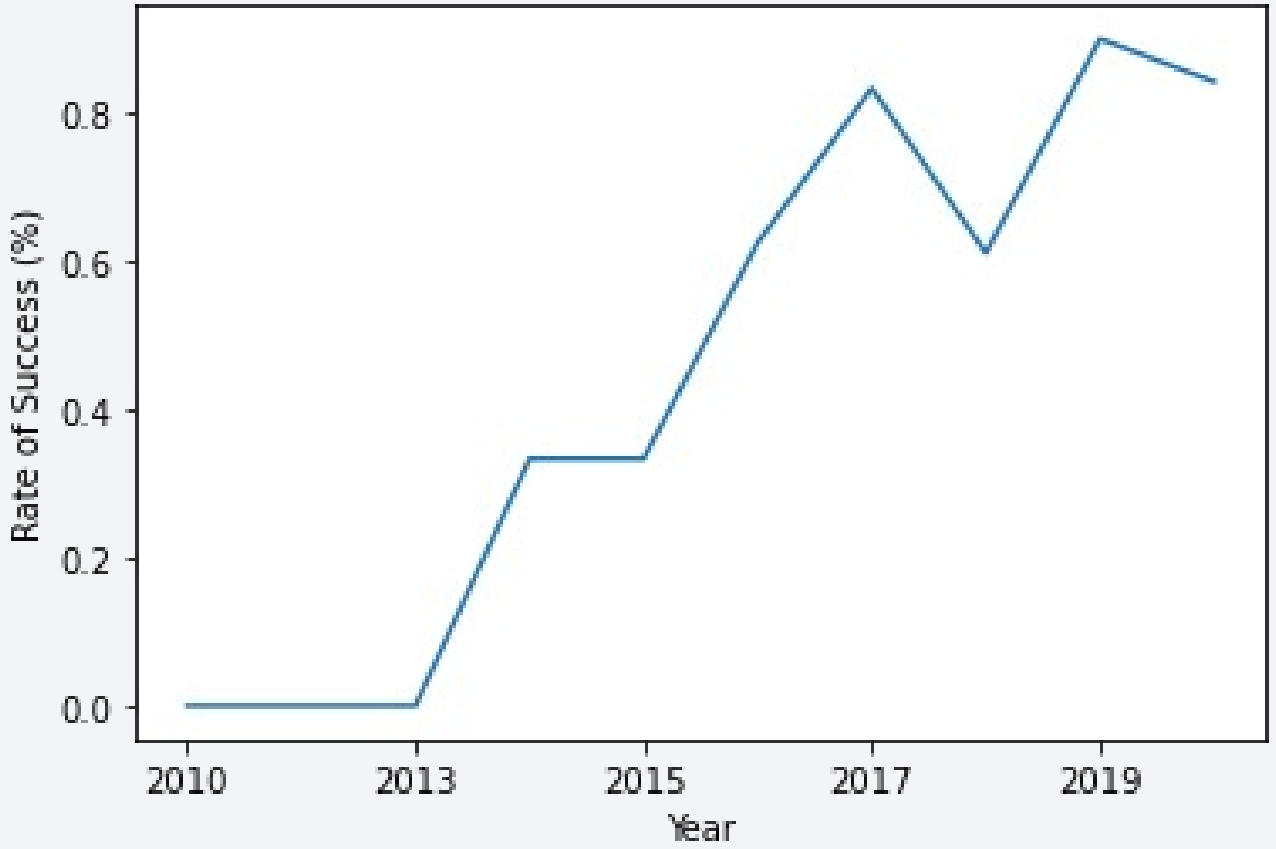
Payload vs. Orbit Type

- The graph below is a scatter plot of Payload Mass (KG) vs Orbit Type. Class 0 blue points represent unsuccessful landings. Class 1orange points represent successful landings.
- For heavier payloads the successful landing rate are higher for PO, LEO and ISS. However for GTO they appears to be no correlation for mass and success rate.



Launch Success Yearly Trend

- This line graph displays the yearly continuous trend of the success rate of landings.
- As is shown there is a increasing trend of success starting in 2013 and continuing on to the most recent figure 80% in 2020.
- The only year to have a major dip was 2018.



All Launch Site Names

```
SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL
```

The query shown above returns a list of all the unique launch site names. The SQL operator “DISTINCT” is used to achieve this. Below is the returned table of results.

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

Launch Site Names Begin with 'CCA'

```
SELECT * FROM SPACEXTBL  WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

The query shown above returns a table of results showing the first 5 records where a launch site name begins with ‘CCA’. The SQL operator “LIKE” is used to achieve this find Lauch site names similar to ‘CCA’. The ‘LIMIT’ operator in this cases limits results to 5. Below is the returned table of results.

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

```
SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)'
```

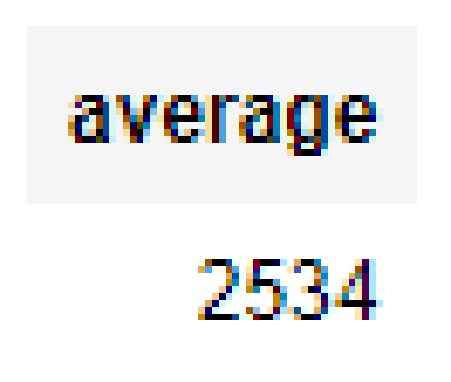
The query shown above returns the summation mass of all payloads where the customer was ‘NASA (CRS). The SQL operator “SUM” is used to achieve this. Below is the returned result.

total
45596

Average Payload Mass by F9 v1.1

```
SELECT AVG(PAYLOAD_MASS__KG_) AS AVERAGE FROM SPACEXTBL WHERE BOOSTER_VERSION LIKE 'F9 v1.1%'
```

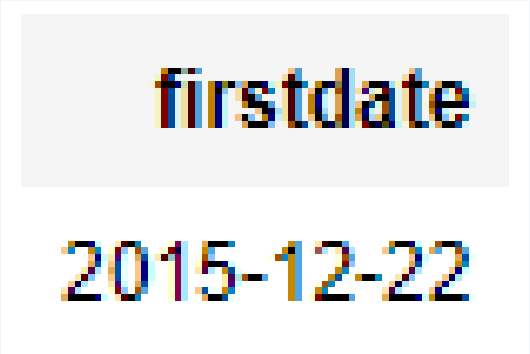
The query shown above returns the mean average payload mass of the ‘F9 v1.1’ version of the booster. The SQL operator “AVG” is used to achieve the average value. The ‘LIKE’ operator is used to find the correct booster version. Below is the returned result.



First Successful Ground Landing Date

```
SELECT MIN(DATE) AS FirstDate FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (ground pad)'
```

The query shown above returns the earliest date on which there was a successful ground pad landing. The SQL operator “MIN” is used to find the first date. Below is the returned result.



Successful Drone Ship Landing with Payload between 4000 and 6000

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

The query shown above returns a list of all unique booster versions that had a successful landing via drone ship. The ‘DISTINCT’ operator find the unique values. The SQL operator ‘BETWEEN’ allows us to have the correct result between two values. In this case between a payload mass of 4000 and 6000. Below is the returned result.

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

Total Number of Successful and Failure Mission Outcomes

```
SELECT MISSION_OUTCOME, COUNT(*) AS TOTALS FROM SPACEXTBL GROUP BY MISSION_OUTCOME
```

The query shown above returns a table showing the number of mission per mission outcome value. The ‘GROUP BY’ operator groups the results by mission outcome value. The ‘COUNT’ operator counts the number of values per group. Below is the returned result. 99 out of 101 launches had successful missions.

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

Boosters Carried Maximum Payload

```
SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

The query shown above returns a list of names of the names of the booster versions which have carried the maximum payload mass. A sub query is used to find the maximum value of the payload mass using the ‘MAX’ operator. Then the data set is filtered to show booster that have flown with this payload mass. The table to the right is the result of this query. ‘F9 B5 B10…’ Seemed to be used to carry max payloads.

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

2015 Launch Records

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

The query shown above returns a table showing failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015. The 'YEAR' operator is used to filter for the year 2015. The table below is the results of this query. Two failed landing this year.

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

```
%sql SELECT LANDING__OUTCOME, COUNT(*) AS TOTALS FROM SPACEXTBL \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY LANDING__OUTCOME ORDER BY TOTALS DESC
```

The query shown above returns the count of landing outcomes between the dates 2010-06-04 and 2017-03-20, ranked in descending order. The ‘BETWEEN’ operator is used to filter the dates. The ‘GROUP BY’ and ‘COUNT’ operators groups the numbers of each outcome, and ‘DESC’ ranks in descending order. The table to the right is the result.

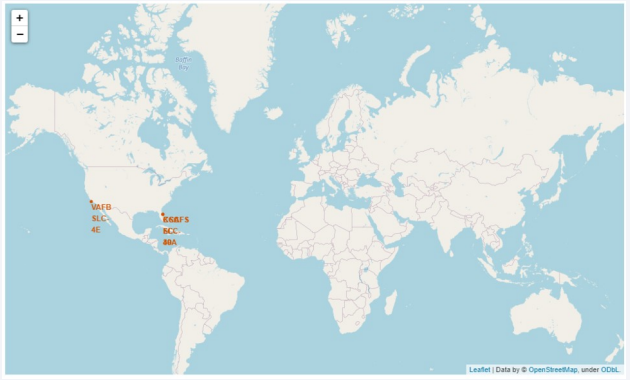
landing__outcome	totals
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

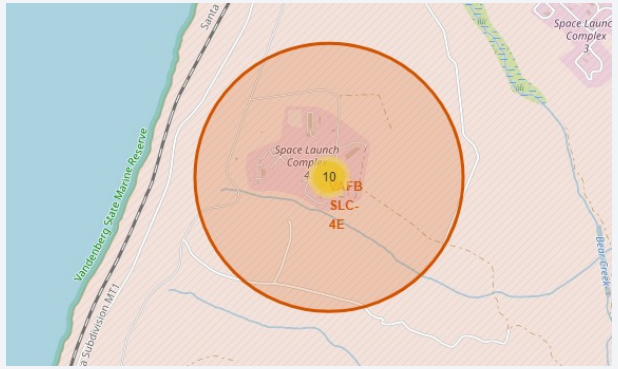
Launch Site Locations

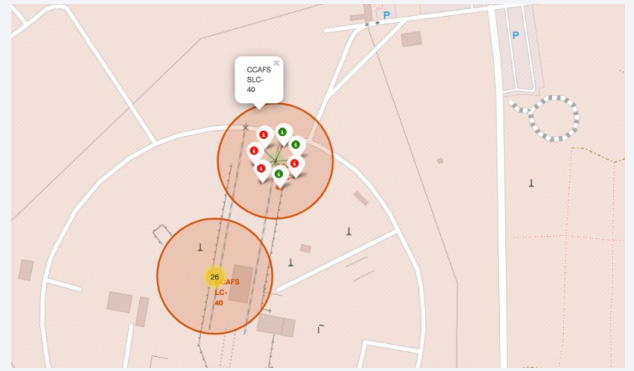
- The first map to the right shows the launch locations on a map of the main land United States. Non-mainland US and Alaska do not have any launch locations so were not included in this map. All launch locations appear to be coastal.
- The second map is a world map view.



Labeled Launch Sites

The screenshots to the right show examples of the of the labeling. Large Red circles represent the clusters areas where launches took place with Yellow circled number. Red labels are failed landings. Green labels are successful landings.

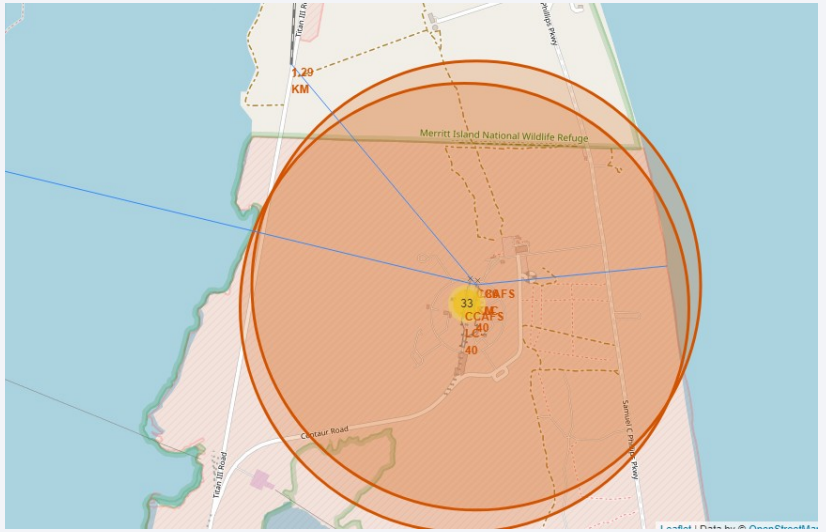


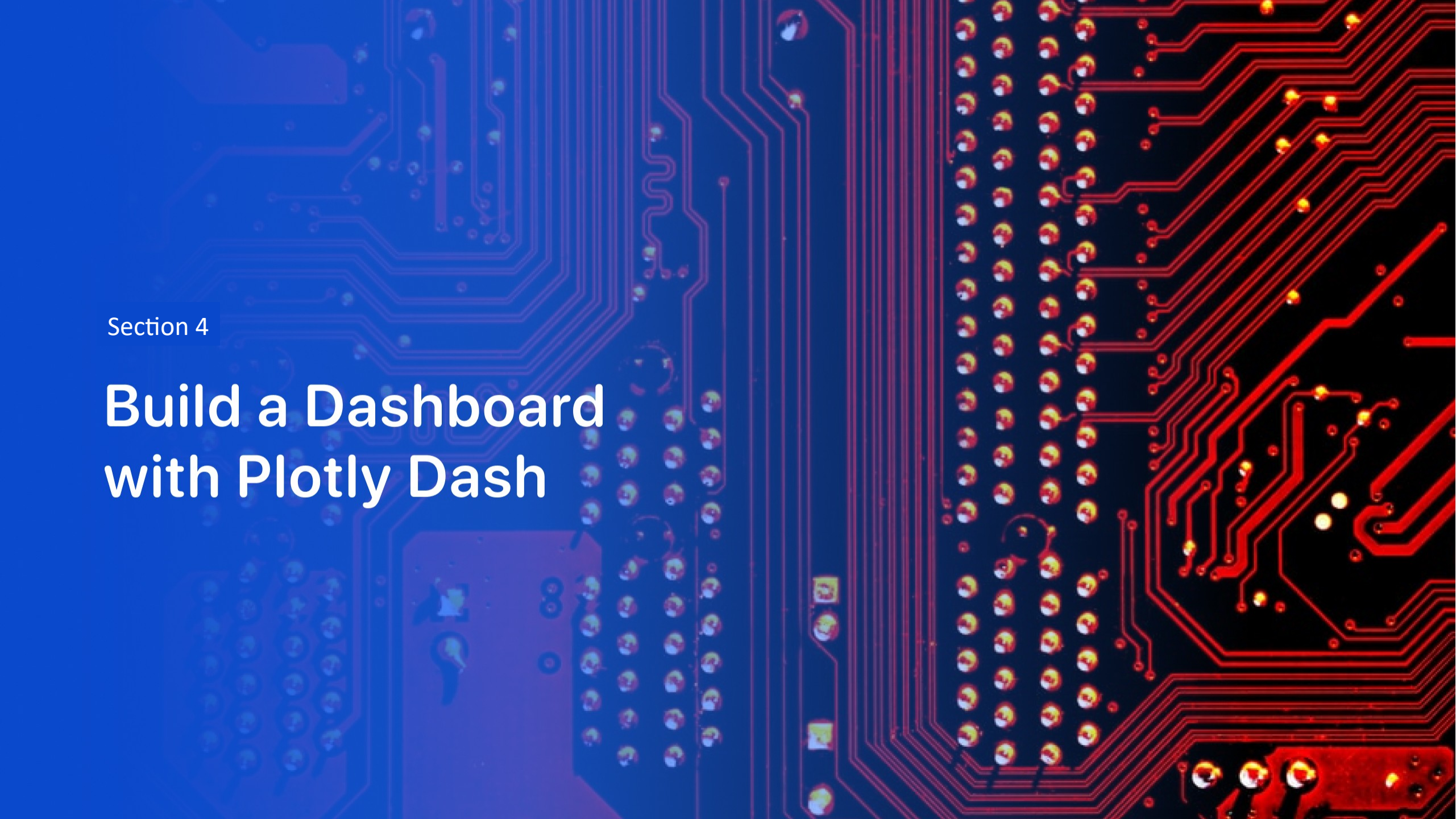


Neighbouring Place of Note To Launch Sites

The launch sites in Florida are relatively close to a railway line and highway, making for good transportation links for pieces, materials and equipment. The are also very close to the coast and ocean.

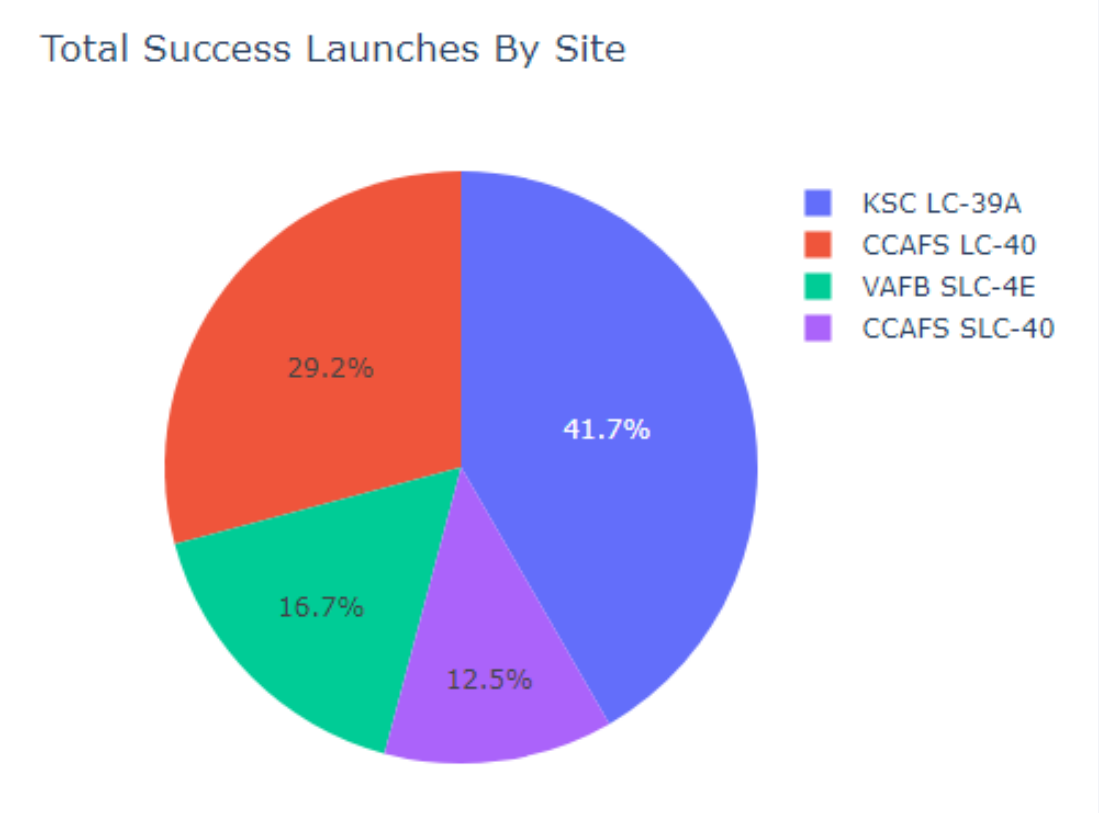
They also appear to be a reasonable distance from the nearest town/ city so a failed launch or landing poses minimal threat to the nearby civilian population.




Section 4
Build a Dashboard with Plotly Dash

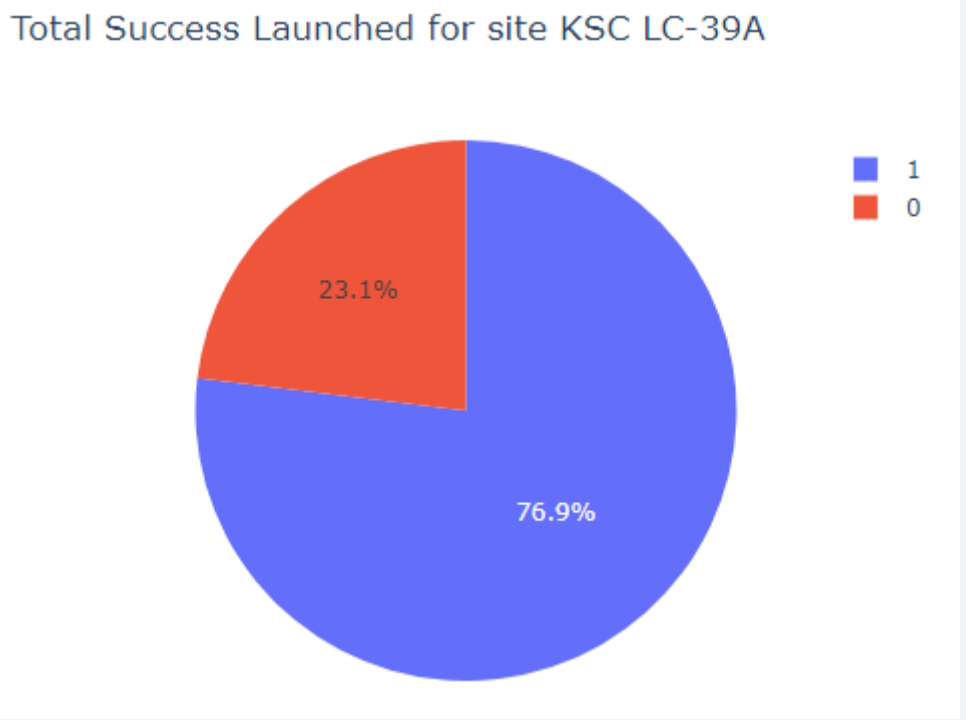
Total Successful Launches Split by Sites

- The pie chart to the right shows the proportion of successful launches per site.
- KSC LC-39A is the most successful launch site with 41.7% of all successes.
- VAFB SLC-4E is the least successful launch site with 12.5% of successes.
- What is not shown by this graph is that the data is somewhat small, so it is hard to make a conclusion as to whether these are trends.



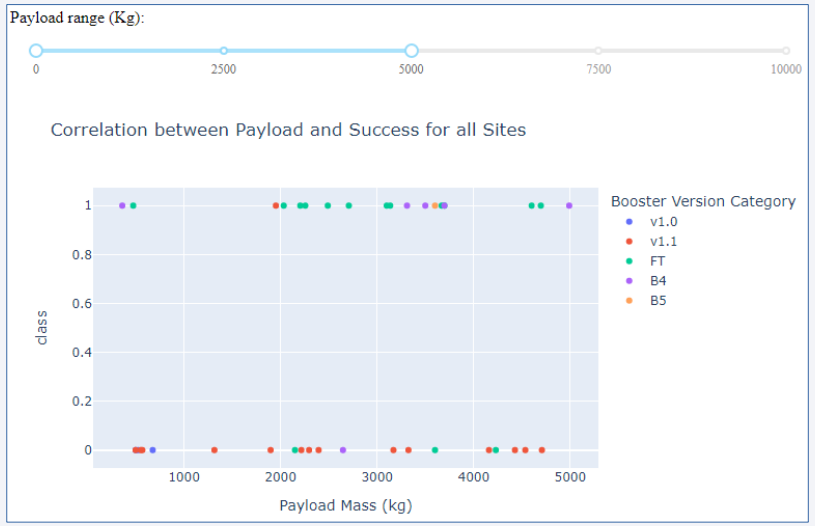
Ratio of Successes at Most Successful Site

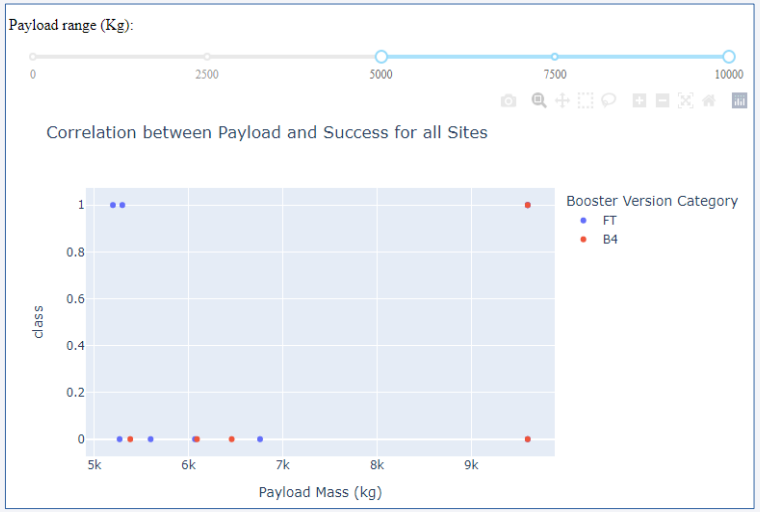
KSLC-39A was the most successful launch site. As displayed by this pie chart 76.9% of it’s landings were successful (10 total) whilst 23.1% (3) were failures.



Payload vs Launch Outcome Plots for All Sites

- The first scatter plot shows all landings with all payloads masses plotted. Given that class 0 is successful landings and class 1 was failed attempts, this plot shows there were a higher number of successes then failures.
- The 2nd and 3rd plots show launches at the 0-5000kg and 5000-1000Kg ranges respectively. These plots show two things:
  - More launches took place at lower payloads masses
  - There was a higher proportion of successful launches at higher payloads.



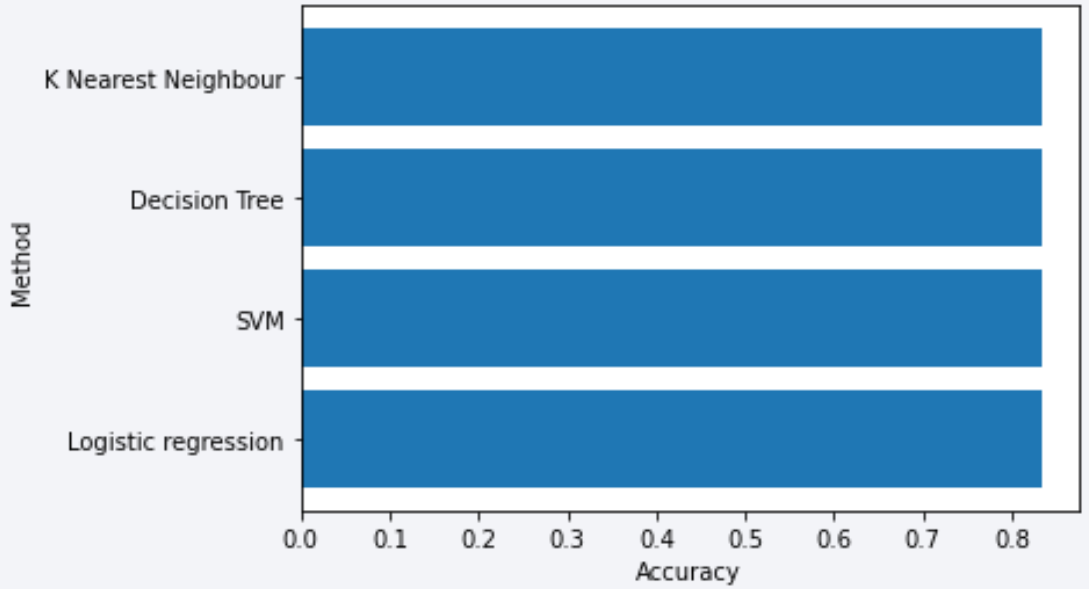


Section 5

Predictive Analysis (Classification)

Classification Accuracy

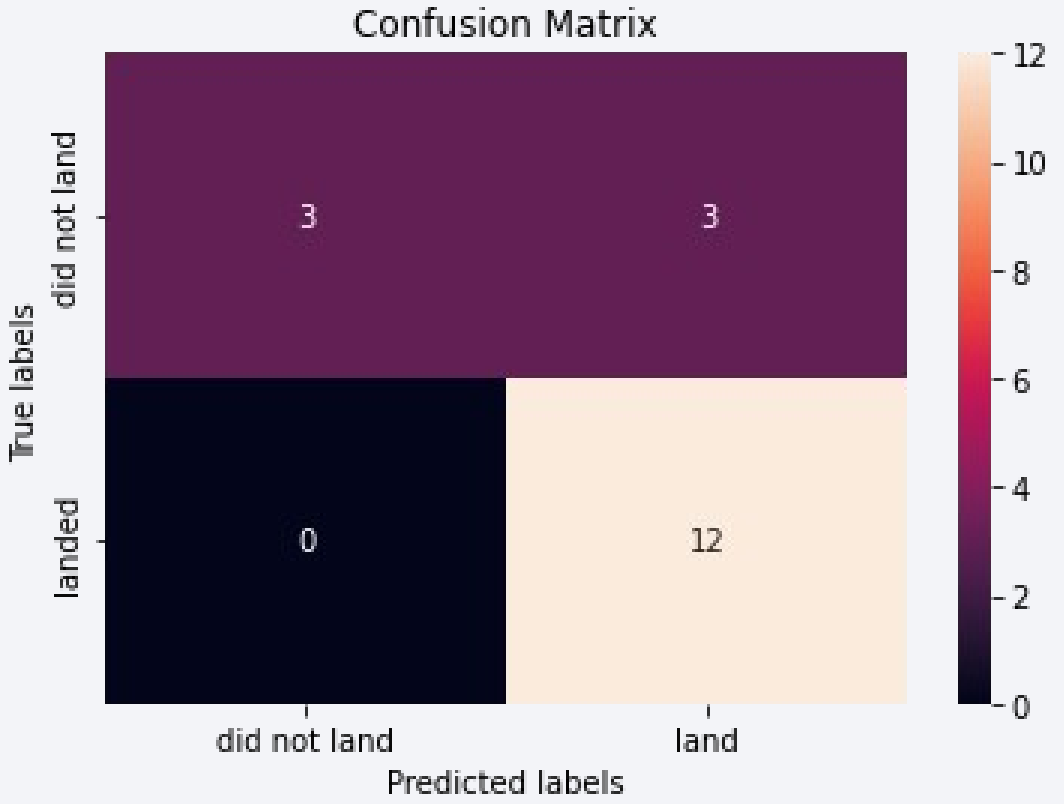
- After training four different models, K-Nearest Neighbour, Decision Tree, SVM and Logistic Regression, the accuracy of each model was found to be the same to 6 decimal places.
- The test size was defined as only 20% of the dataset. In this case this was only a small 18 values.
- It is recommended that more data is used in order to truly determine the best model choice.



	Methods Names	Accuracy Scores
0	Logistic regression	0.833333
1	SVM	0.833333
2	Decision Tree	0.833333
3	K Nearest Neighbour	0.833333

Confusion Matrix

- All models performed just as well as each other. This has lead to the confusion matrix of all 4 models looking the same.
- The models predict 12 successful landing where the true label was the same as the predicted. And 3 failed landings where the landing failed.
- The models also predicted 3 false positives. (Predicted landing, for was actually a failure)
- These model predominantly predict successful landings.



Conclusions

- Over time the number of successful landings increased, as shown in the pattern in flight numbers.
- The most recent years success rate was 80%.
- Four orbit types had a 100% success rates; SSO, HEO, GEO and ES-L1.
- The launch site with the highest number of successful landings and the highest proportion of successful landing was KSLC-39A.
- The launch site was close to a railway and highway for good transportation links, but also close the ocean and far enough away from city to pose minimal threat to civilians.
- The launch success rate of higher mass payload is higher then lower mass payload, but there needs to be more data for conclusive proof.
- All predictive models trained for this project had an accuracy of 83.3333%. However since they were trained/ tested with a limited amount of data, more would recommended to choose a definitive predictive model.

Appendix

- Link To the Applied Data Science Capstone Course
- Link To GitHub Repository For Code And Jupyter Notebooks
- Link To Falcon 9 and Falcon Heavy Wikipedia Article (used for web scraping data)

Thank you!