IBM Developer
SKILLS NETWORK

Winning the Space Race with Data Science

AMANDA ANDERSON

JUNE 2023

Outline

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

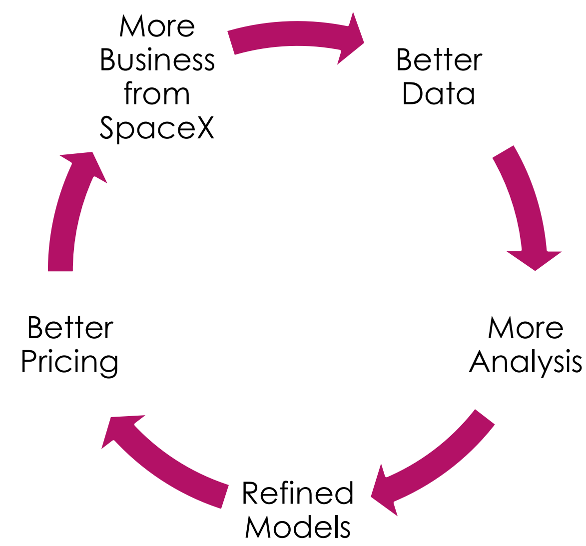
Executive Summary

Summary of Methodology

- All data collected was made available through public reference via websites
- A combination of web scraping and REST APIs were used to gather the data
- Many Python libraries were utilized in the wrangling, analysis, and visualization of the data including: seaborn, Matlotlib, pandas, and SQL
- Testing was used to pick the best performing prediction model with the greatest accuracy

Summary of Results

The success of flight landings has steadily improved over time. There are certain Orbit types that stand out as better performers, and after our data analysis it's safe to say to more launches = better results. This means as a rival company, we found significant relationships in the data we analyzed and can make predictions on landing outcomes.



Introduction

Can you believe we are living in the age of commercial space flight?! Companies are busy making **space travel** affordable for everyone, SpaceX being not only one of those companies – but a revolutionary one.

SpaceX's accomplishments include:

- Sending spacecraft to the International Space Station.
- Starlink, a satellite internet constellation providing satellite Internet access.
- Sending manned missions to Space.

SpaceX's rocket launch costs are well below industry average - $62 million versus others costs of $165 million

- The reason is because SpaceX can **reuse the first stage** by re-landing the rocket for another mission

In this project, we are working as data scientists for a rival company. We want to bid on projects in which we'll be competing with SpaceX.

It's crucial we correctly predict the outcome of the first landing in order to know the right price to bid against SpaceX. This includes:

- Identifying the factors that influence the outcome of the landing
  - The relationship between those variables and the effect on the outcome
- The conditions to produce the best possible outcome of the landing

Methodology

SECTION 1

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected via web scraping from Wikipedia and using the SpaceX API
- Perform data wrangling
  - Data was processed utilizing one-hot encoding and dropping columns not needed.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

The datasets were collected in 2 ways:

- SpaceX API: https://api.spacexdata.com/v4/rockets/
  - Use a get request
  - Normalize the data and convert to pandas dataframe
  - Clean data; check for missing values, drop columns, etc.
- Web Scraping from Wikipedia: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
  - Use BeautifulSoup library to extract HTML table
  - Parse table and convert to pandas dataframe

Data Collection – SpaceX API

- **Use SpaceX API for launch data**
  - get request
- **Normalize data**
  - Convert to pandas df
- **Filter Data**
  - Falcon 9 data only
- **Clean data**
  - Fill in missing values
  - Filter by date

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

```python
response = requests.get(spacex_url)
```

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

```python
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that have multiple
#payloads in a single rocket.
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

GitHub Link:
https://github.com/amanda303/DataScience_Capstone/blob/main/week%201_jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

Request the Falcon 9 Launch data from Wiki page	• Requests.get()
Extract data using BeautifulSoup	• Find tables
Get Column Names	• Column and variable names from HTML header
Convert to Pandas df	• Parse HTML tables

```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url).text
```

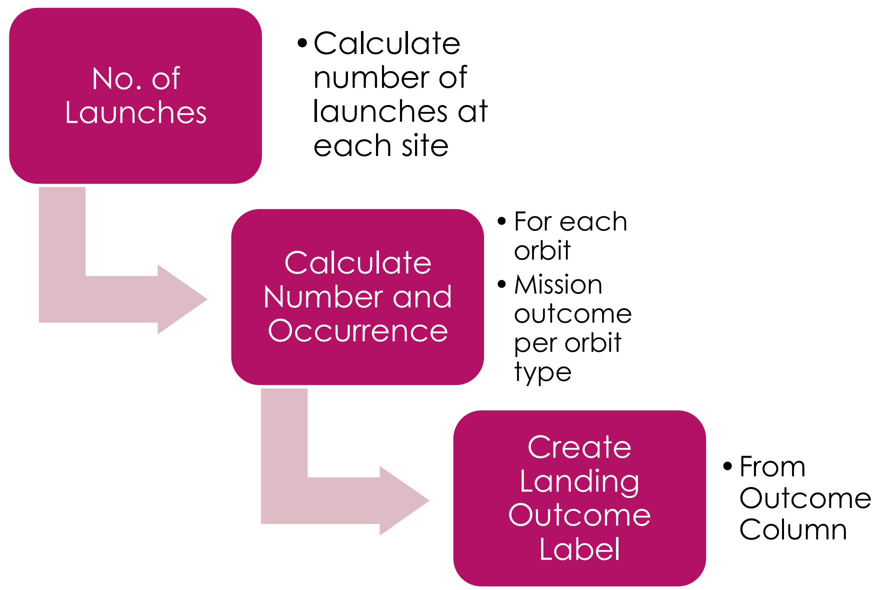
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response, 'html.parser')
```

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            #print(flight_number)
            datatimelist=date_time(row[0])
```

GitHub Link:
https://github.com/amanda303/DataScience_Capstone/blob/main/week%201_jupyter-labs-webscraping.ipynb

Data Wrangling

Clean/unify data and perform initial Exploratory Data Analysis (EDA)

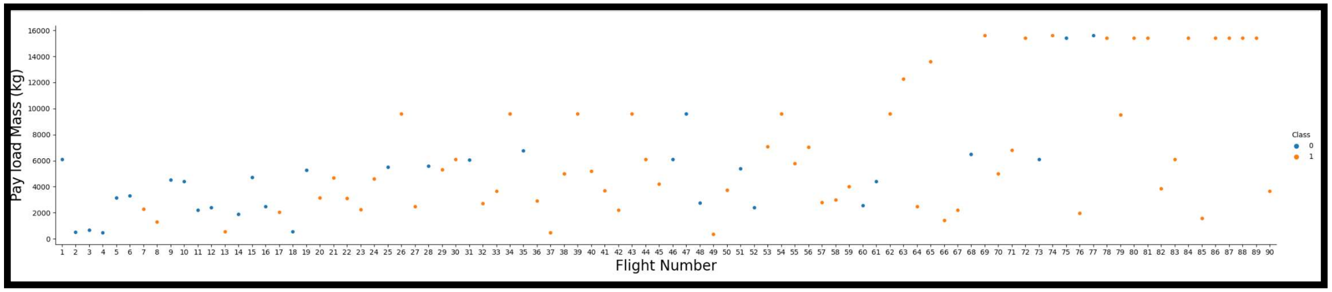


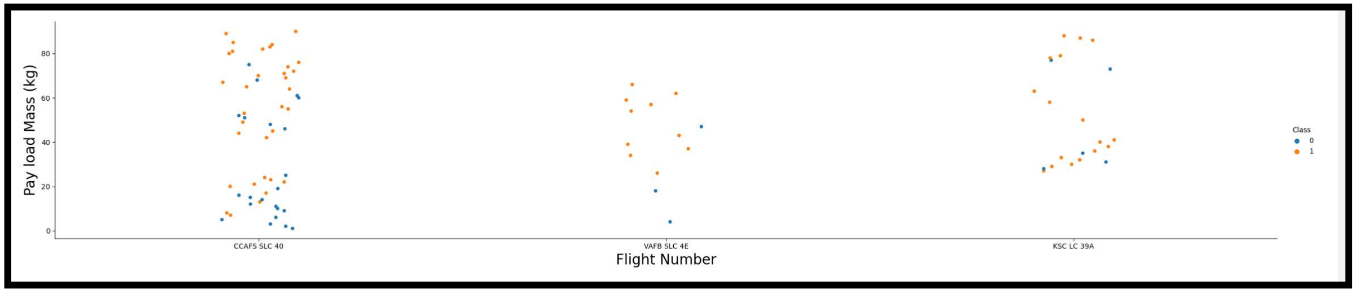
GitHub Link:
https://github.com/amanda303/DataScience_Capstone/blob/main/week%201_jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

EDA with Data Visualization

We examined the relationship between different variables using scatterplots

- Such as Flight Number vs. Payload Mass
- Flight Number by Launch Site





GitHub Link:
https://github.com/amanda303/DataScience_Capstone/blob/main/week%202_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

Further analysis was performed with SQL queries:

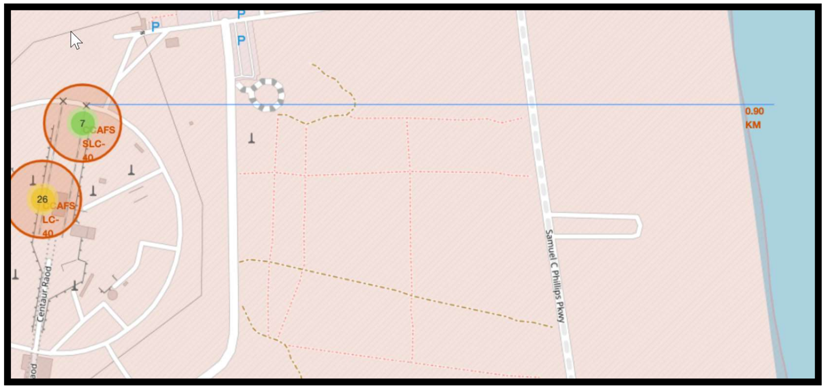
- Names of the launch sites
- 5 Launch site with names that being with 'CCA'
- Total Payload Mass carried by Boosters launched by NASA
- Average Payload Mass carried by booster F9 v1.1.
- Date of first successful landing outcome achieved at a ground pad
- Boosters with success in drone ship with a payload mass between 4000 and 6000
- Total number of Success vs Failure Outcomes
- Booster version with the maximum payload mass
- Failed landing booster information includmg outsomes in drone ship, their booster versions, and launch site names from year 2015
- Rank the count of landing outcomes between specific dates in descending order

GitHub Link:
https://github.com/amanda303/DataScience_Capstone/blob/main/week%202_jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

We used the Folium Python library to visualize the sites with a map

- Derived from latitude/longitude coordinates
- Circle markers were added to the sites and Marker Clusters for launches at each site
- Green and Red formatting were added to indicate success or failure of the launch
- Distance calculations to determine proximity to cities, coastlines, etc.



GitHub Link:
https://github.com/amanda303/DataScience_Capstone/blob/main/week%203_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Created a slider to explore data by payload range
- Created charts to look at percentage of success to launch sites
- Created scatter plots to analyze success with booster version category

plotly

GitHub Link:
https://github.com/amanda303/DataScience_Capstone/blob/main/week%203_jupyter_launch_site_location.jupyterlite.ipynb

Predictive Analysis (Classification)

We built 4 different models to compare results:

1. Logistic Regression
2. Decision Tree
3. Support Vector Machine
4. KNN

Build
- Use NumPy Arrays
- Transform Data
- Choose ML Algorithm

Evaluate
- Check accuracy of model
- Inspect Confusion Matrix

Improve & Best Fit
- Fine tune model if needed
- Look for best accuracy score

GitHub Link:
https://github.com/amanda303/DataScience_Capstone/blob/main/week%204_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

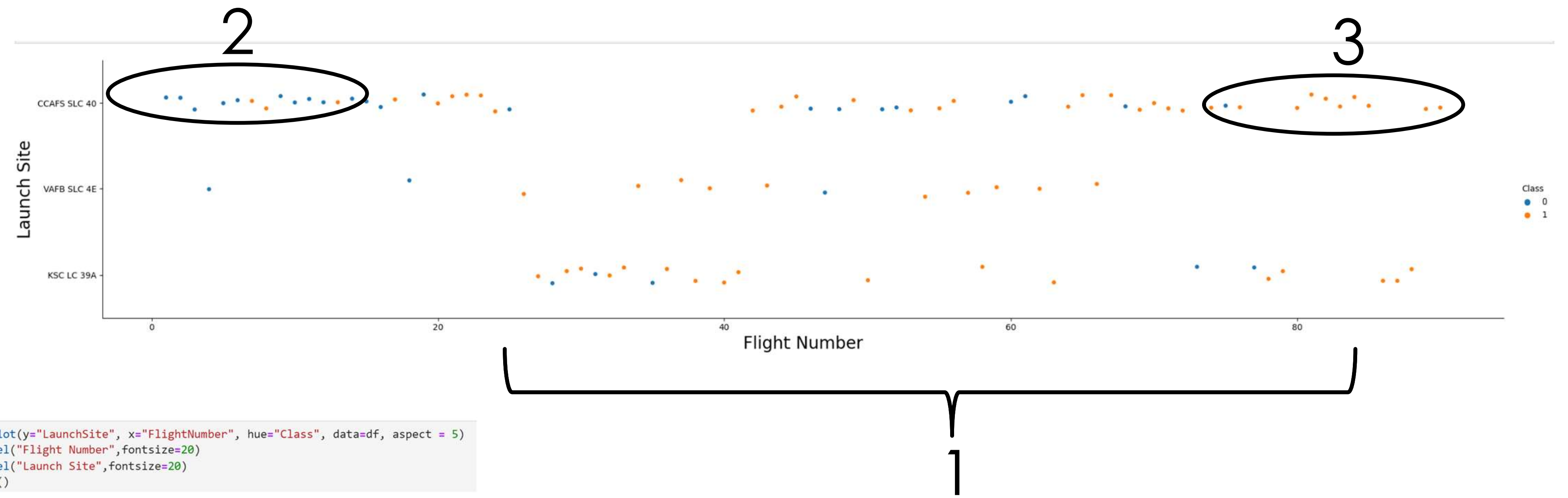
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

EDA Visualization and Insights

SECTION 2

Flight Number vs. Launch Site

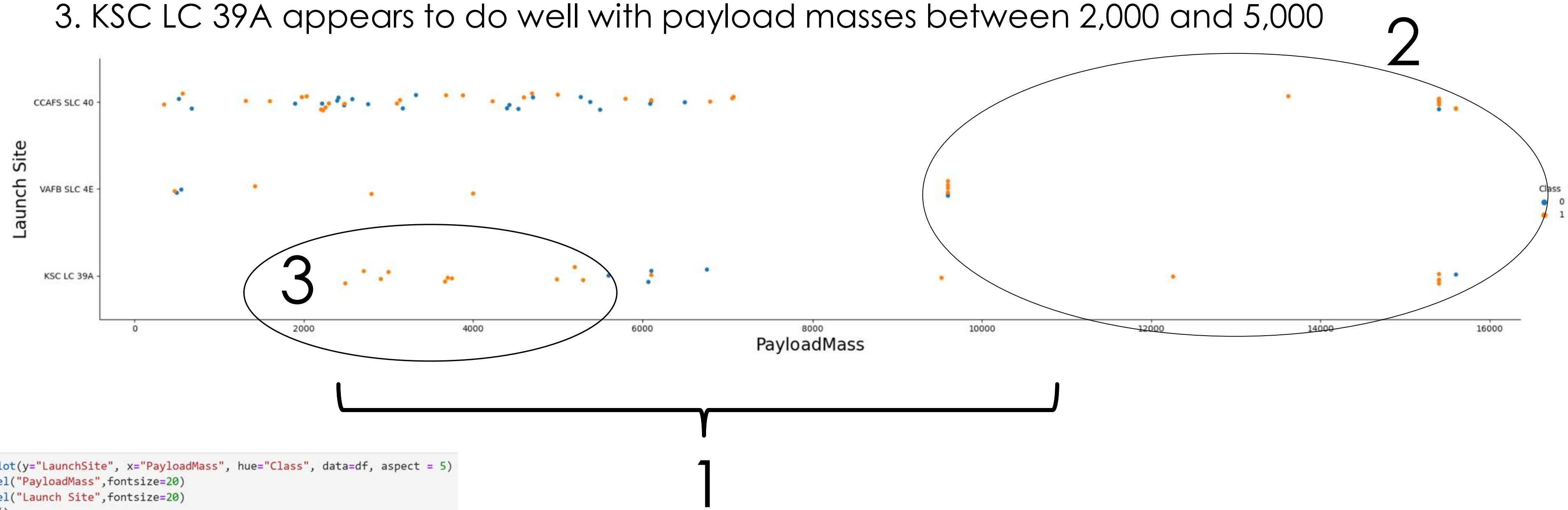
1. There looks like a positive relationship between number of launches and success rates
2. CCAFS SLC 40 appears to improving success rates over time and with more flights
3. CCAFS SLC 40 has had the greatest number of successful launches most recently




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

Payload vs. Launch Site

1. No strong correlation to make a definitive conclusion
2. However it does appear that higher payload weight leads to more successful launches in general
3. KSC LC 39A appears to do well with payload masses between 2,000 and 5,000




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

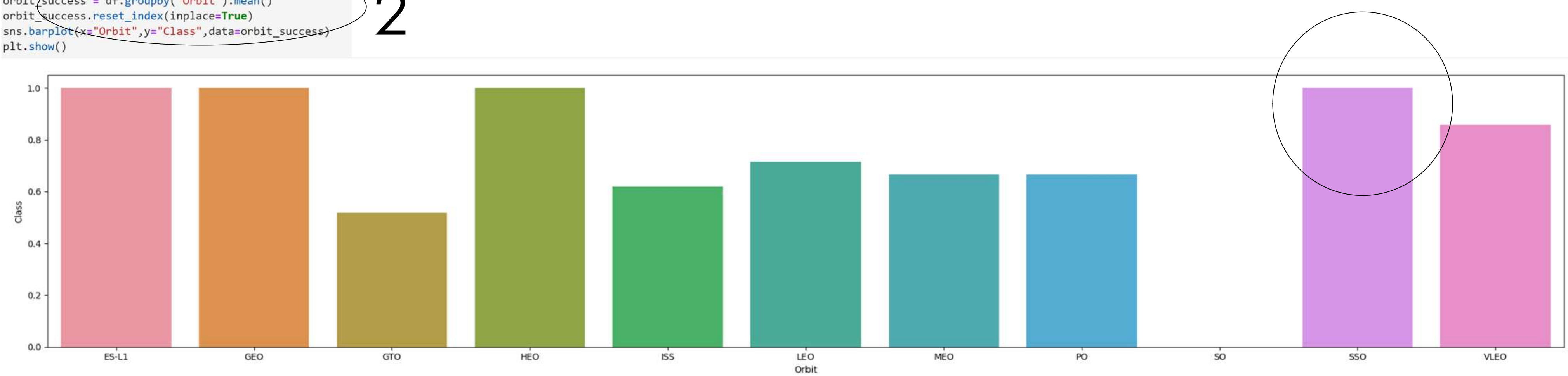
Success Rate vs. Orbit Type

1. Some had 100% success rates, but this isn't accounting for the number of launches for each orbit type
2. Using groupby() method on the Orbit column gives us the Success Rate mean

```
orbit_success = df.groupby('Orbit').mean()
orbit_success.reset_index(inplace=True)
sns.barplot(x="Orbit",y="Class",data=orbit_success)
plt.show()
```

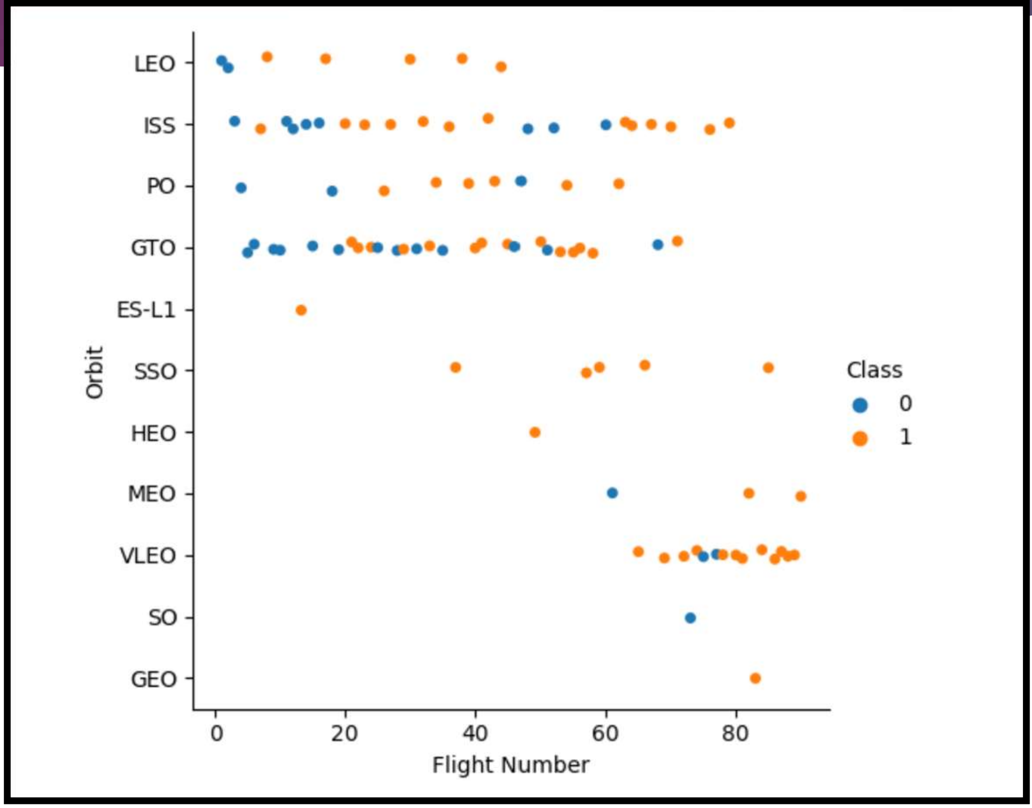
2

1



Flight Number vs. Orbit Type

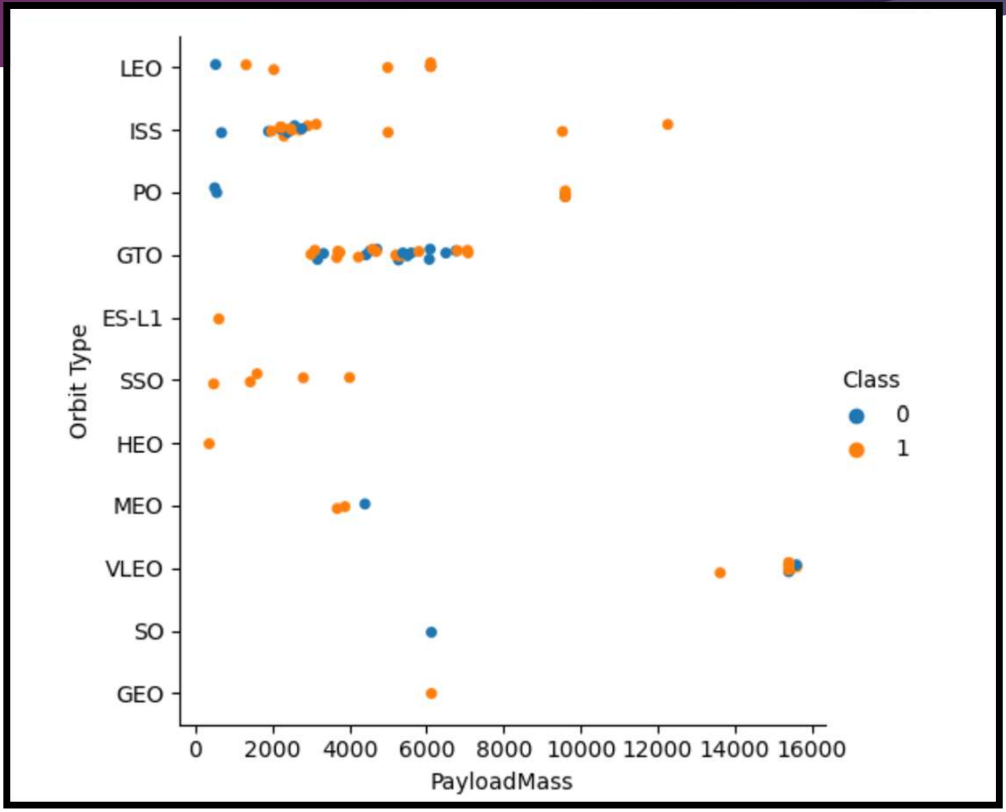
- VLEO has taken the most number of flights
- There appears to be an indirect correlation between failed launches and the number of them, however some outliers prevent this from having a strong relationship



Payload vs. Orbit Type

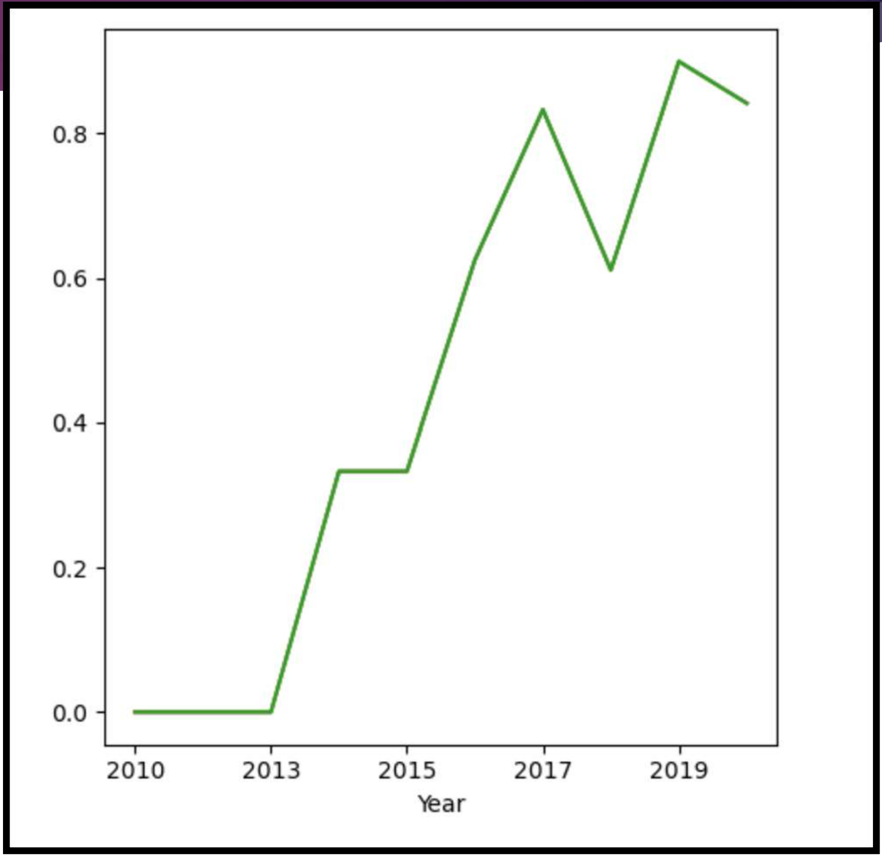
- Larger Payload Masses have a positive effect on some orbits and a negative on others; meanwhile orbit GTO launch success shows no correlation to Payload weight

- We can determine the success rate of launches can not be considered based on payload masses alone



Launch Success Yearly Trend

- There's been incredible growth of successful launches since 2013.
- There was a bit of a downturn around 2018-2019, but looks to have corrected itself by 2020.



All Launch Site Names

According to the data, using DISTINCT shows us there are 4 launch sites in the SpaceX data

```
%sql SELECT Distinct LAUNCH_SITE FROM SPACEXTBL
```

```
 * sqlite:///my_data1.db
Done.
```

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Launch Site Names Begin with 'CCA'

Using LIKE ‘CCA%’ together with LIMIT gets us 5 records with ‘CCA’ Launch Sites:

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

```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No attempt
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attempt
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

Total Payload Mass for NASA is calculated by finding the sum and restricting rows with the WHERE clause

Our total Payload Mass = 45,596 kg

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER='NASA (CRS)'

 * sqlite:///my_data1.db
Done.
```

SUM(PAYLOAD_MASS__KG_)
45596.0

Average Payload Mass by F9 v1.1

- Use the Average function and an additional WHERE clause to limit records to Booster Version F9 v1.1
- The average payload mass for F9 v1.1 is 2,928.4 kg

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION='F9 v1.1'
```

```
 * sqlite:///my_data1.db
Done.
```

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- The first successful landing outcome in a ground pad is Jan 8, 2018
- This was calculated using the min(DATE) with the WHERE clause to filter by the correct landing outcome

```
%sql SELECT min(DATE) FROM SPACEXTBL WHERE LANDING_OUTCOME='Success (ground pad)'
```

```
 * sqlite:///my_data1.db
Done.
```

min(DATE)
01/08/2018

Successful Drone Ship Landing with Payload between 4000 and 6000

These landings were calculated with the following query:

%sql **SELECT** BOOSTER_VERSION **FROM** SPACEXTBL

WHERE LANDING_OUTCOME = 'Success (drone ship)' **AND** PAYLOAD_MASS__KG_ > 4000 **AND** PAYLOAD_MASS__KG_ < 6000;

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

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (drone ship)' AND
PAYLOAD_MASS__KG_ > 4000 AND  PAYLOAD_MASS__KG_ < 6000;
```

Total Number of Successful & Failure Mission Outcomes

Utilizing Count together with the LIKE clause yields counts for both successful and failed missions

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Successful" FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Success%'
```

```
 * sqlite:///my_data1.db
Done.
```

Successful
100

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Failed" FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE 'Failure%'
```

```
 * sqlite:///my_data1.db
Done.
```

Failed
1

Boosters Carried Maximum Payload

These are the names of the Booster Versions carrying the maximum payload

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

Here is the query that produced the results

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

```
 * sqlite:///my_data1.db
Done.
```

2015 Launch Records

Failed landing outcomes in drone ship by booster version and Launch site

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

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

The query included WHERE plus the AND clause to get failed missions

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

Landing Outcomes in Descending Order

Landing Outcome	Total Count
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

COUNT, DATE BETWEEN, and ORDER BY were used to create the list

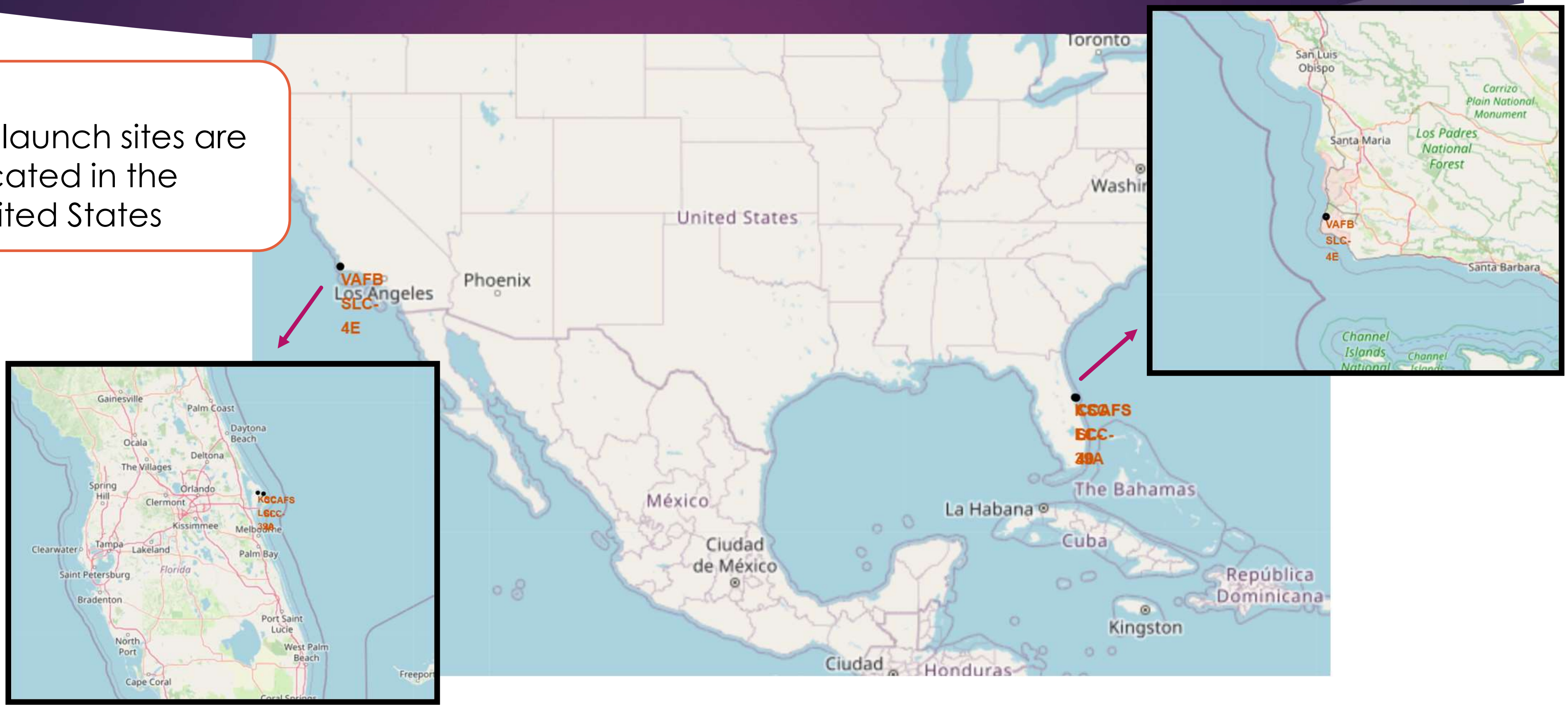
```
%%sql
SELECT LANDING_OUTCOME, COUNT(LANDING_OUTCOME) AS TOTAL_NUMBER
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING_OUTCOME
ORDER BY TOTAL_NUMBER DESC
```

Launch Sites Proximity Analysis

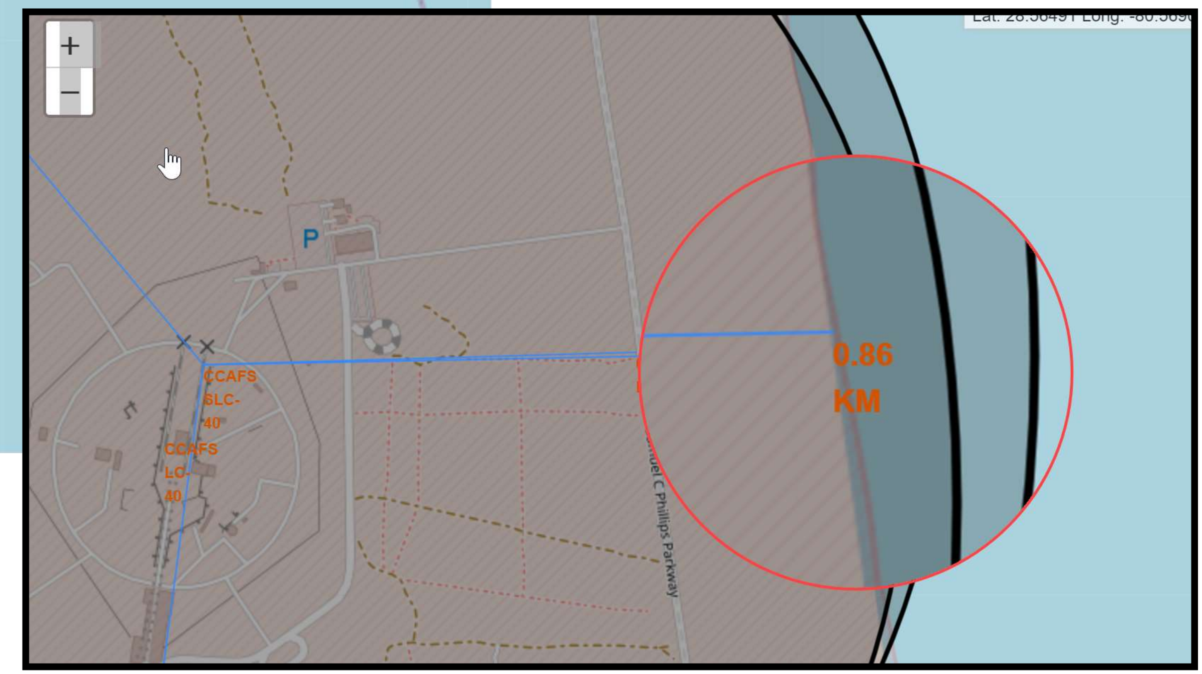
SECTION 3

Launch Site Locations

All launch sites are located in the United States



Launch Site Distances to Landmarks of Concern



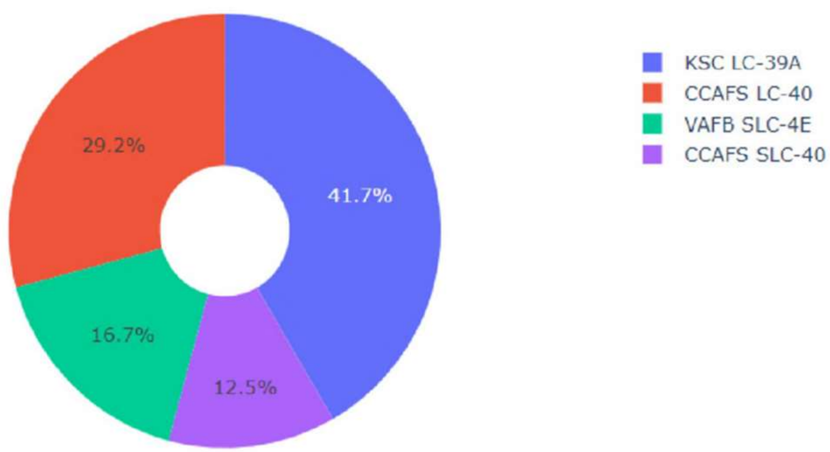
Launch sites are close to coastlines, but stay away from highways and railways

Build a Dashboard with Plotly Dash

SECTION 4

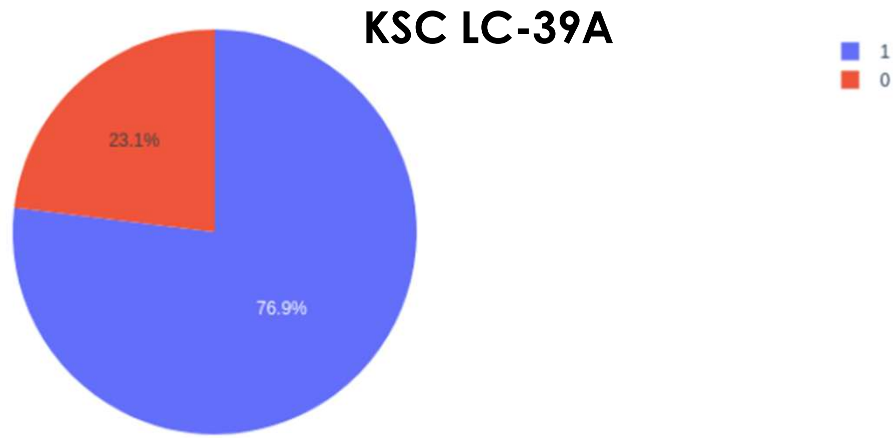
Success Percentage of All Launch Sites

KSC LC-39A has the most successful launches by far; CCAFS LC-40 is a distant second



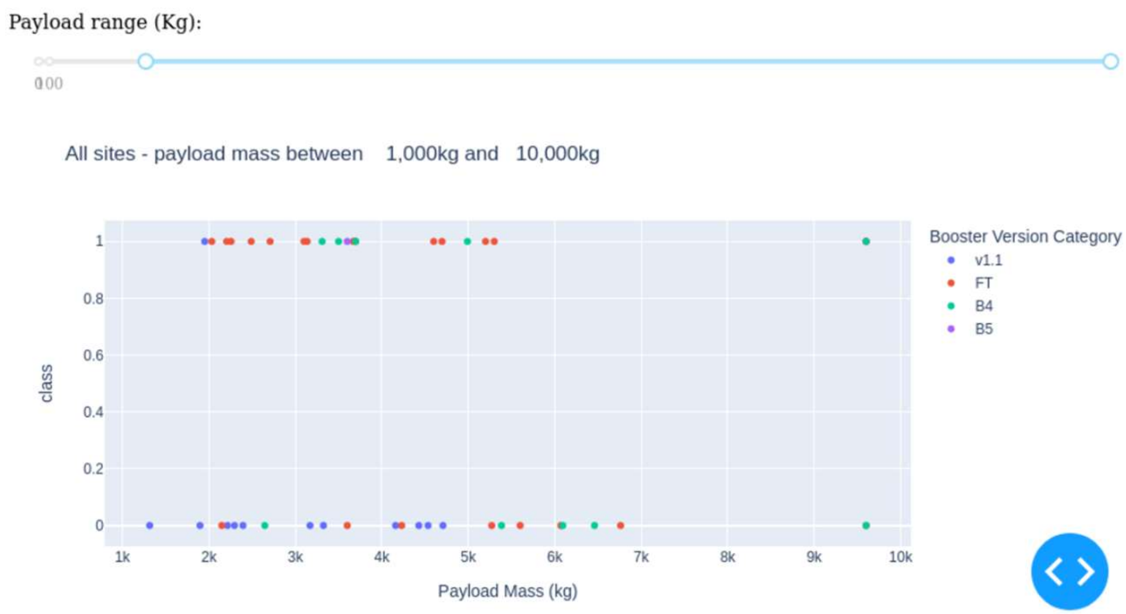
Launch Success Ratio for Highest Success

KSC LC-39A achieved an impressive 76.9% success rate and only 23.1% failure.



Payload vs Launch Outcome Scatter Plot

Low weighted boosters, particularly FT Boosters, are the most likely to have a successful launch outcome



Predictive Analysis (Classification)

SECTION 5

Classification Accuracy

Decision Tree Method appears to be the worst performing model, but there's no variance in accuracy among the remaining 3 methods

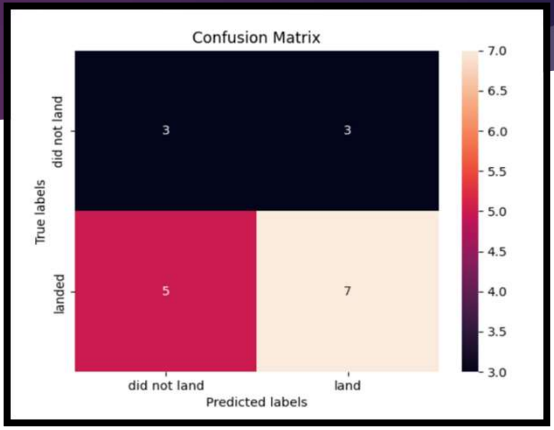
```
print('Accuracy for Logistics Regression method:', logreg_cv.score(X_test, Y_test))
print( 'Accuracy for Support Vector Machine method:', svm_cv.score(X_test, Y_test))
print('Accuracy for Decision tree method:', tree_cv.score(X_test, Y_test))
print('Accuracy for K nearsdt neighbors method:', knn_cv.score(X_test, Y_test))
```

```
Accuracy for Logistics Regression method: 0.8333333333333334
Accuracy for Support Vector Machine method: 0.8333333333333334
Accuracy for Decision tree method: 0.5555555555555556
Accuracy for K nearsdt neighbors method: 0.8333333333333334
```

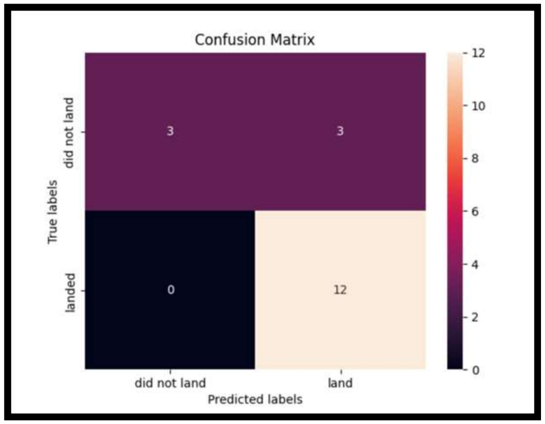
Confusion Matrix

Here we see the results of the Confusion Matrices indicate that all methods except Decision Tree can distinguish between the different classes. We see that the major problem is false positives.

Decision Tree Method



Logistic Regression
Support Vector Machine
K Nearest Neighbor



Conclusions

- It’s more probable to get a successful launch outcome the more launches are performed, and success rates have been increasing steadily with time since 2013
- Payload mass is a variable to pay attention to with far too little or too much weight, but not a factor to consider as a strong correlation to success or failed launches
- KSC LC-39A is the best launch site for probability of a successful launch
- The SSO Orbit has the highest success rate with 100% and more than 1 launch
- The Logistic Regression, KNN, and Vector Machine methods can be used to predict successful landings

Thank you!!