IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Melissa Vales
11/26/2024

# Outline

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

# Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scrapping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis with Visualization Results
  - Exploratory Data Analysis with SQL Results
  - Interactive Map with Folium Results
  - Plotly Dash Dashboard Results
  - Predictive Analysis (Classification) Results

# Introduction

- Project background and context
  - ➢ In this capstone, a machine learning pipeline was used to predict if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.

- Problems you want to find answers
  - ➢ Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

Section 1

# Methodology

# Methodology

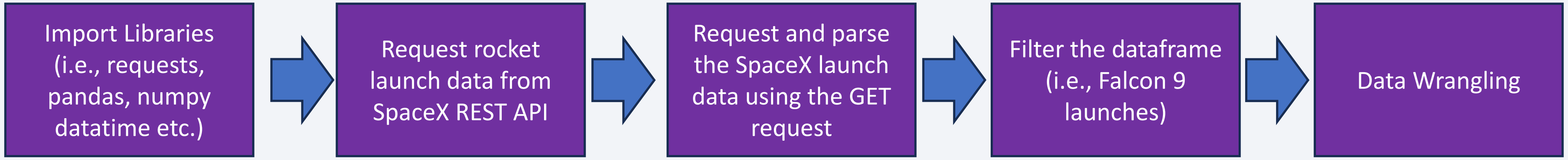
## Executive Summary

- Data collection methodology:
  - ➢ SpaceX launch data was gathered from an API, specifically the SpaceX REST API
  - ➢ Data was also obtained from Wikipedia via Web Scrapping
- Perform data wrangling
  - ➢ Space X Falcon 9 First Stage Landing Outcomes (Success or Failure) were converted into Training Labels (1or 0)
  - ➢ 1 = booster landed successfully, 0 = booster was unsuccessful
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - ➢ This includes preprocessing, standardizing data, using Train_test_split and performing a grid search that will find the hyperparameters that will show which algorithm will perform best. This can be done by Logistic Regression, Support Vector machines, Decision Tree Classifier, K-nearest neighbors and a confusion matrix.

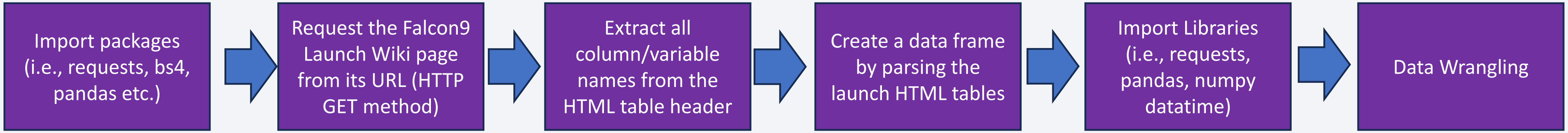
# Data Collection

- Data was collected via SpaceX API and web scrapped from Wikipedia

➢SpaceX API




➢Web Scrapping



# Data Collection – SpaceX API

- Data collection with SpaceX REST calls
- Link here: GitHub URL

1. Request rocket launch data from SpaceX API using a URL

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

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

2. Use the static response object to make the requested JSON results more consistent

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

3. Decode the response content as a Json and turn it into a Pandas dataframe

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

4. Use API to get information about the launches using the IDs given for each launch

```
# 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)]
```

# Data Collection - Scraping

- Web scraping process
- Link here: [GitHub URL]()

1. Perform an HTTP GET method to request the Falcon9 Launch HTML pages, as an HTTP response

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

2. Create a BeautifulSoup object from HTML response

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

3. Create an empty dictionary with keys from the extracted column names

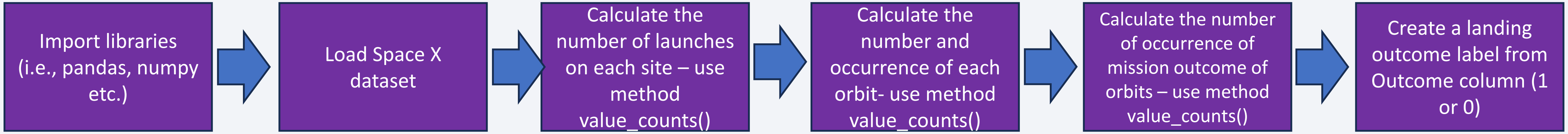
```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
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'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

# Data Wrangling

- Exploratory Data Analysis (EDA) was used to find patterns in the data and to determine the label for training supervised models. Outcomes were converted to Training Labels with 1 (successfully landed) and 0 (Unsuccessful landing).




- Link here: [GitHub URL]

# EDA with Data Visualization

Summary of chart that were plotted:

- FlightNumber vs Payload
  - ➢ As flight number increases, the first stage is more likely to land successfully. With more massive payloads, the first stage returns successfully. Catplot was used to visualize this relationship
- Flight Number vs Launch Site
  - ➢ Catplot was used to visualize the relationship between Flight Number and Launch Site. Some launch sites were more successful than others as flight numbers increased.
- Payload Mass vs Launch Site
  - ➢ A Catplot was used to visualize the relationship between payload mass and launch site. There were no rockets launched for heavypayload mass for launch site VAFB-SLC
- Relationship between success rate of each orbit type
  - A bar chart was used to visualize the relationship between success rate of each orbit type. Orbit types SSO, HEO, GEO and ES-L1 had the highest success rate.

# EDA with Data Visualization Cont.

- Relationship between FlightNumber and Orbit type
  - ➢ A Catplot was used to visualize the relationship between flight number and Orbit type. In the LEO orbit, success is related to the number of flights. For the GTO orbit, there is no relationship between flight numbers and success.
- Relationship between Payload Mass and Orbit type
  - ➢ A Catplot was used to visualize the relationship between payload mass and orbit type. For orbit types Polar, LEO and ISS, the heavy payloads had more successful landings
- Launch success yearly trend
  - ➢ A line chart was used to visualize the launch success yearly trend. The success rate increased from 2013 to 2020.

- Link Here: [GitHub URL]

# EDA with SQL

SQL queries performed:

1. Display the names of the unique launch sites in the space mission
   - %sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL
2. Display 5 records where launch sites begin with the string ‘CCA’
   - %sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE ‘CCA%’ LIMIT 5
3. Display the total payload mass carried by boosters launched by NASA (CRS)
   - %sql SELECT SUM(PAYLOAD_MASS_KG_) FROM SPACEXTBL WHERE CUSTOMER = ‘NASA(CRS)’
4. Display average payload mass carried by booster version F9 v1.1
   - %sql  SELECT AVE(PAYLOAD_MASS_KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION = ‘F9 v1.1’

# EDA with SQL

5. List the date when the first successful landing outcome in ground pad was achieved

   - %sql SELECT MIN(DATE) FROM SPACEXTBL WHERE LANDING_OUTCOME = 'SUCCESS (GROUND PAD)'

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

   - %sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING_OUTCOME = 'SUCCESS (DRONE SHIP)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000

7. List the total number of successful and failure mission outcomes

   - %sql SELECT COUNT(MISSION_OUTCOME) FROM SPACEXTBL WHERE MISSION_OUTCOME = 'SUCCESS' OR MISSION_OUTCOME = 'FAILURE (IN FLIGHT)'

8. List the names of the booster_versions which have carried the maximum payload mass.

   - %sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)

# EDA with SQL

9. List the record which will display the month names, failure landing_outcomes in drone ship, booster versions, launch_site for the months in year 2015

   - %sql SELECT MISSION_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE EXTRACT (YEAR FROM DATE) = ‘2015’ AND LANDING_OUTCOME LIKE ‘%FAILURE%’

10. 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.

   - %sql SELECT LANDING_OUTCOME, COUNT(*) AS COUNT_LANDING_OUTCOME FROM SPACEXTABLE WHERE (DATE BETWEEN ‘2010-06-04’ AND ‘2017-03-20’) GROUP BY LANDING_OUTCOME ORDER BY COUNT_LANDING_OUTCOME DESC

- Link Here: GitHub URL

# Build an Interactive Map with Folium

Summary of map objects such as markers, circles, lines created and added to a folium map

- folium.Circle and folium.Marker
  - ➢ Used to add a highlighted circle area with a text label on a specific coordinate
- MarkerCluster
  - ➢ Helps combine markers of close proximity into clusters. This will simplify the display of markers on the map.
- MousePosition
  - ➢ This will get a coordinate for a mouse over a point on the map
- Folium.PolyLine
  - ➢ This will draw a line between launch sites to the selected coastline point

- Link Here: [GitHub URL]

# Build a Dashboard with Plotly Dash

Summary of plots/graphs and interactions added to the dashboard

- Launch Site Drop-down Input Component
  - ➢ Will allow selection of different launch sites
- callback function 1
  - ➢ Will render success-pie-chart based on selected site dropdown
- Range slider
  - ➢ Will Select Payload
- callback function 2
  - ➢ Will render success-payload-scatter-chart scatter plot
- Link here: GitHub URL

# Predictive Analysis (Classification)

Summary of how the model was built, evaluated, improved, and best performing classification model

1. Import libraries (i.e., pandas, numpy etc.)
2. Load the dataframe
3. Create NumPy array (assign to variable Y) and standardize the data in X
4. Use the function train_test_split to split data X and Y into training and test data
5. Create logistic regression and GridSearchCV object
6. Calculate the accuracy on the test data using the method score
7. Calculate the accuracy on the test data using the method score
8. Create a decision tree classifier object and create a GridSearchCV object
9. Calculate the accuracy of tree_cv on the test data using the method score
10. Create a k nearest neighbors object and then create a GridSearchCV object knn_cv
11. Calculate the accuracy of knn_cv on the test data using the method score
12. Find the method that performs best
13. Create a support vector machine object and create a GridSearchCV object

Link here: [GitHub URL]

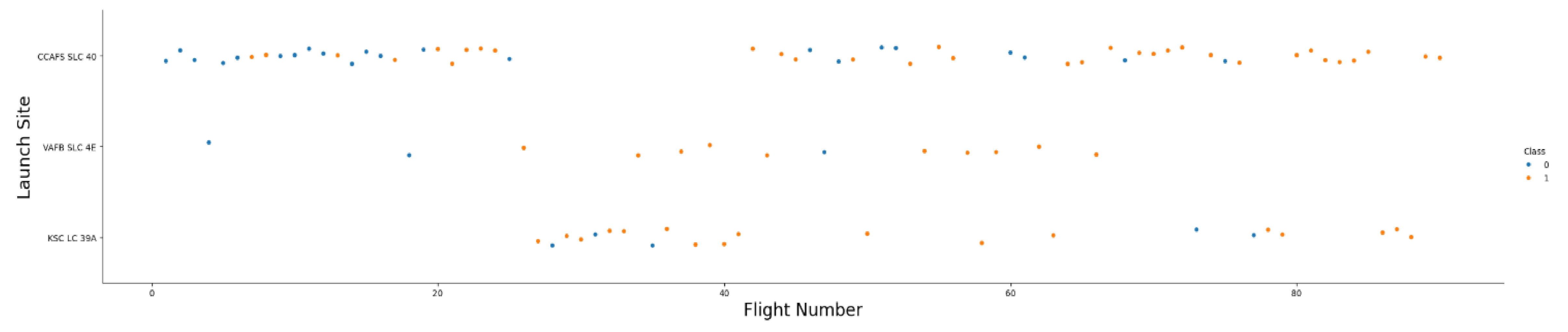
# Results

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

Section 2

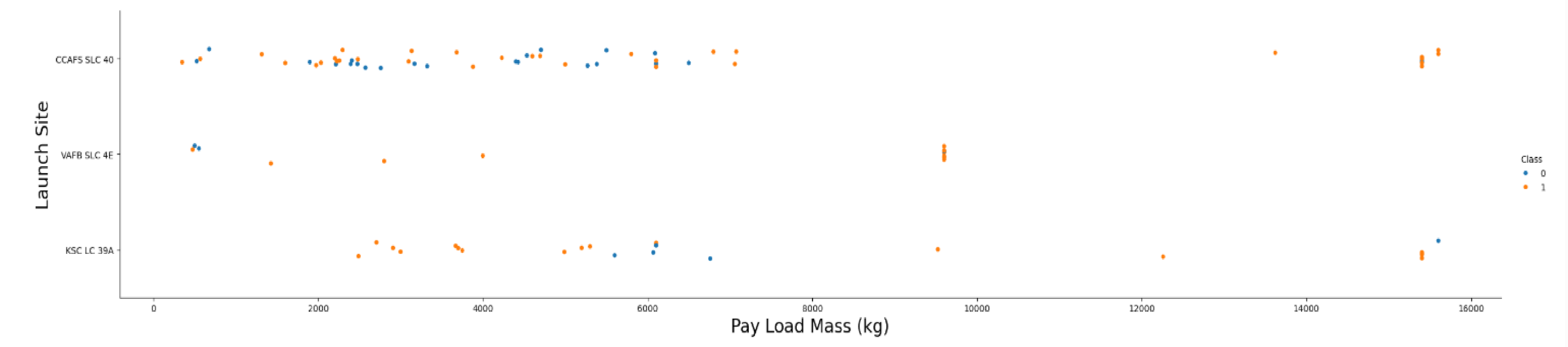
# Insights drawn from EDA

# Flight Number vs. Launch Site



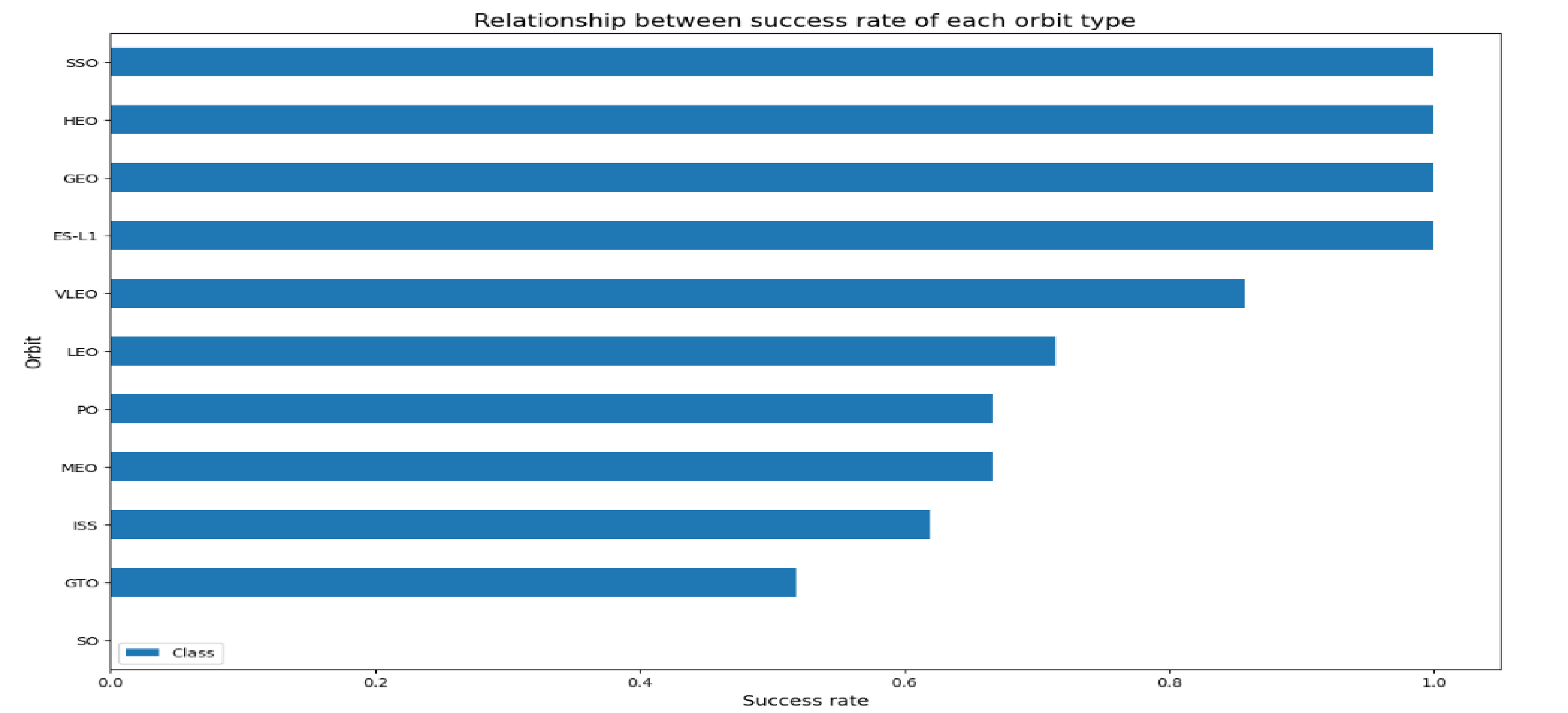

- As the Flight Number increases there are more successful landings at each of the Launch Sites.
- CCAFS SLC 40 had the most successful landings as the Flight Number increased

# Payload vs. Launch Site



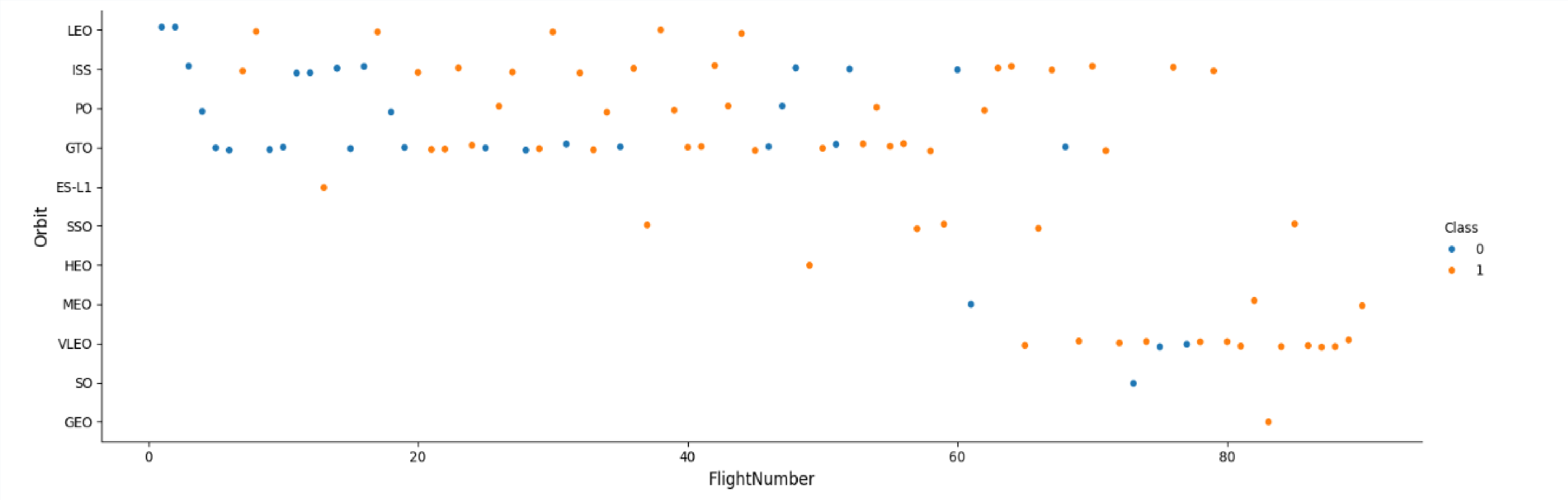

- VAFB-SLC launch site has no rockets launched for heavypayload mass(greater than 10000).
- CCAFS SLC 40 and KSC LC 39A had successful landings when Pay Load Mass was greater then 14000 kg.

# Success Rate vs. Orbit Type



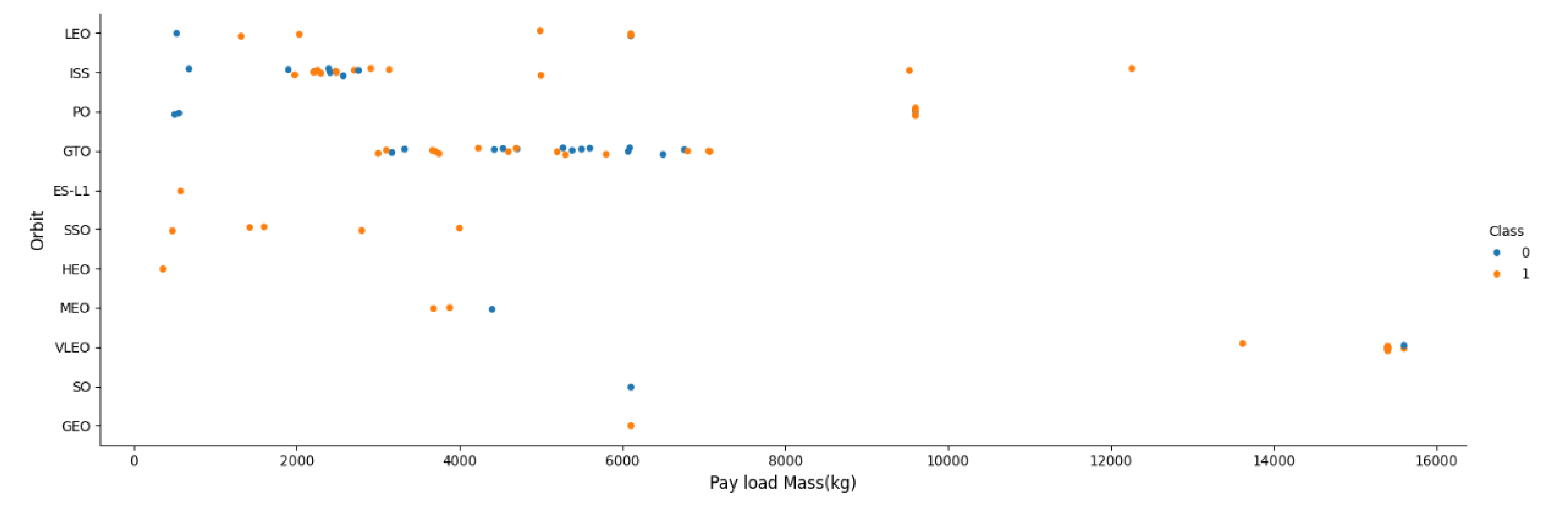

- Orbit types: SSO, HEO, GEO, and ES-L1 has the greatest success rate.
- Orbit type: SO had no success rate.

# Flight Number vs. Orbit Type



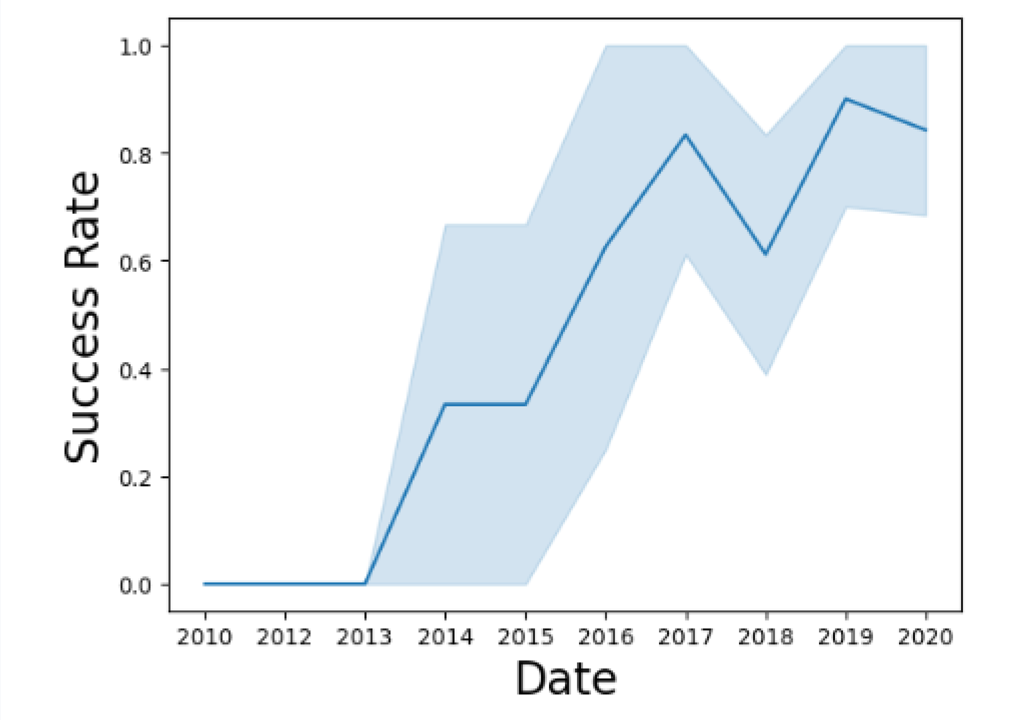

- In the LEO orbit, success is related to the number of flights.
- In the GTO orbit, there is no relationship between flight number and success.

# Payload vs. Orbit Type




- Orbit types: Polar, LEO, and ISS had more successful landings with heavy payloads
- GTO had both successful and unsuccessful landings, therefore a relationship between payload and orbit type is not easy to discern.

# Launch Success Yearly Trend




- The success rate increased from 2013-2020.

# All Launch Site Names

```
%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 |

- There are four unique launch sites: CCAFS LC-40, VAFB SLC-4E, KSC LC-39A, CCAFS SLC-40
- A “Distinct” statement was used to obtain these four launch sites.

# Launch Site Names Begin with 'CCA'

```
%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 |
|---|---|---|---|---|---|---|---|---|---|
| 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 | 7:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | No attempt |
| 2012-10-08 | 0:35:00 | 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 |

- There are 5 records where launch sites begin with `CCA`
- “Like” was used to check for “CCA” and “Limit” was used to obtain only 5 records.

# Total Payload Mass

```
%sql SELECT SUM(Payload_Mass__KG_) FROM SPACEXTBL WHERE customer ='NASA (CRS)';
```

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

| SUM(Payload_Mass__KG_) |
|---|
| 45596 |

- The total payload carried by boosters from NASA was 45,596 KG

# Average Payload Mass by F9 v1.1

```
%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 |

- The average payload mass carried by booster version F9 v1.1was 2,928.4 KG

# First Successful Ground Landing Date

```
%sql select min(DATE) from SPACEXTBL where Landing_Outcome = 'Success (ground pad)';
```

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

| min(DATE) |
| --- |
| 2015-12-22 |

- The dates of the first successful landing outcome on ground pad was 2015-12-22

# Successful Drone Ship Landing with Payload between 4000 and 6000

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

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

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

- The following boosters have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 : F9 FT B1022, F9 FT B1026, F9 FT B1021.2, F9 FT B1031.2

# Total Number of Successful and Failure Mission Outcomes

```
%sql select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure (in flight)'
```

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

| count(MISSION_OUTCOME) |
|---|
| 99 |

- The total number of successful and failure mission outcomes was 99

# Boosters Carried Maximum Payload

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

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

| 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 |

- The above SQL output lists the  names of the booster which have carried the maximum payload mass

# 2015 Launch Records

```
%sql SELECT  substr(Date, 6,2) as Month, substr(Date,0,5) as Year, Landing_Outcome, Booster_Version,  Launch_Site FROM SPACEXTABLE WHERE Date Like '%2015%' and Landing_Outcome LIKE '%Failure (drone ship)%'
```

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

| Month | Year | Landing_Outcome | Booster_Version | Launch_Site |
|---|---|---|---|---|
| 01 | 2015 | Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| 04 | 2015 | Failure (drone ship) | F9 v1.1 B1015 | CCAFS LC-40 |

- The above SQL graph lists the  failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

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

```
%sql SELECT Landing_Outcome, count(*) as Count_Landing_Outcome from SPACEXTABLE WHERE (Date BETWEEN  '2010-06-04' and '2017-03-20') GROUP BY Landing_Outcome ORDER BY Count_Landing_Outcome DESC;
```

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

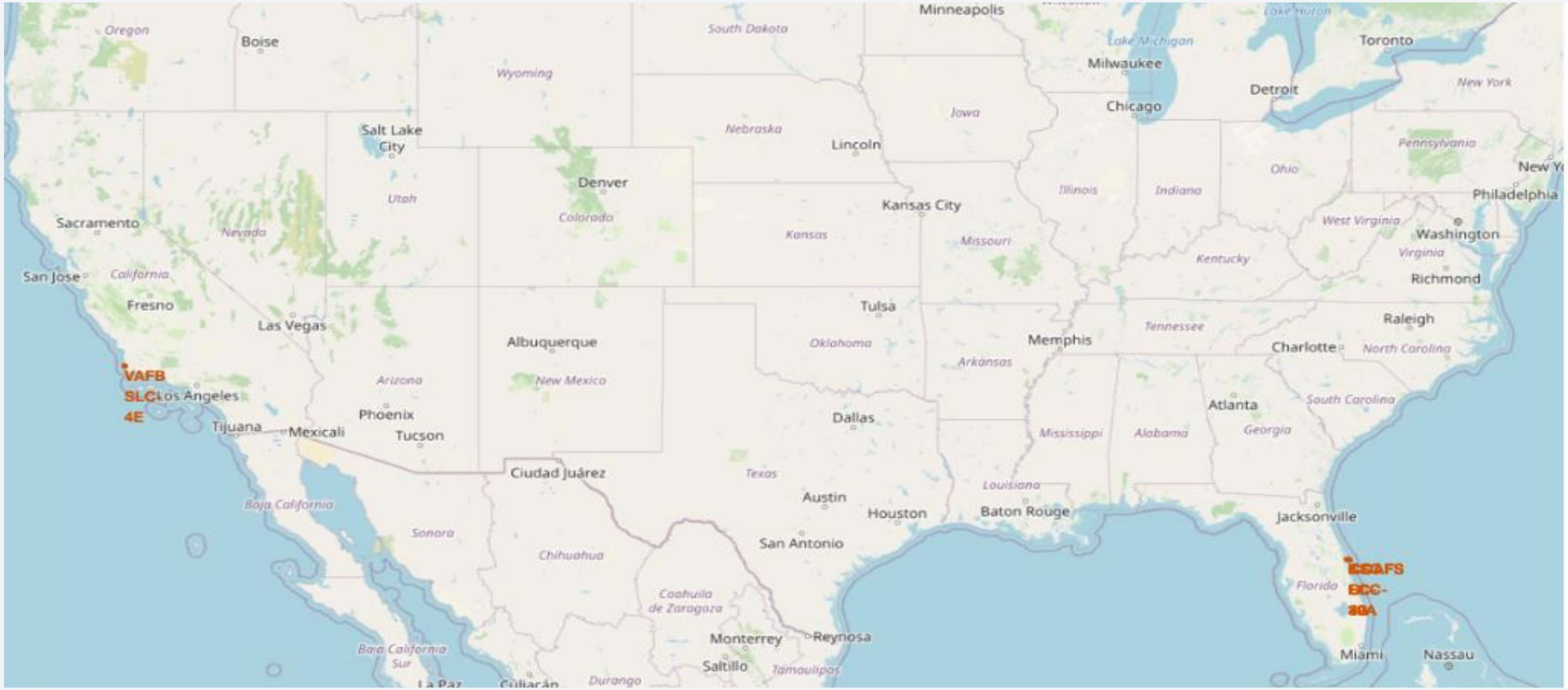
| Landing_Outcome | Count_Landing_Outcome |
|---|---|
| No attempt | 10 |
| Success (drone ship) | 5 |
| Failure (drone ship) | 5 |
| Success (ground pad) | 3 |
| Controlled (ocean) | 3 |
| Uncontrolled (ocean) | 2 |
| Failure (parachute) | 2 |
| Precluded (drone ship) | 1 |

- The above SQL graph shows 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

Section 3

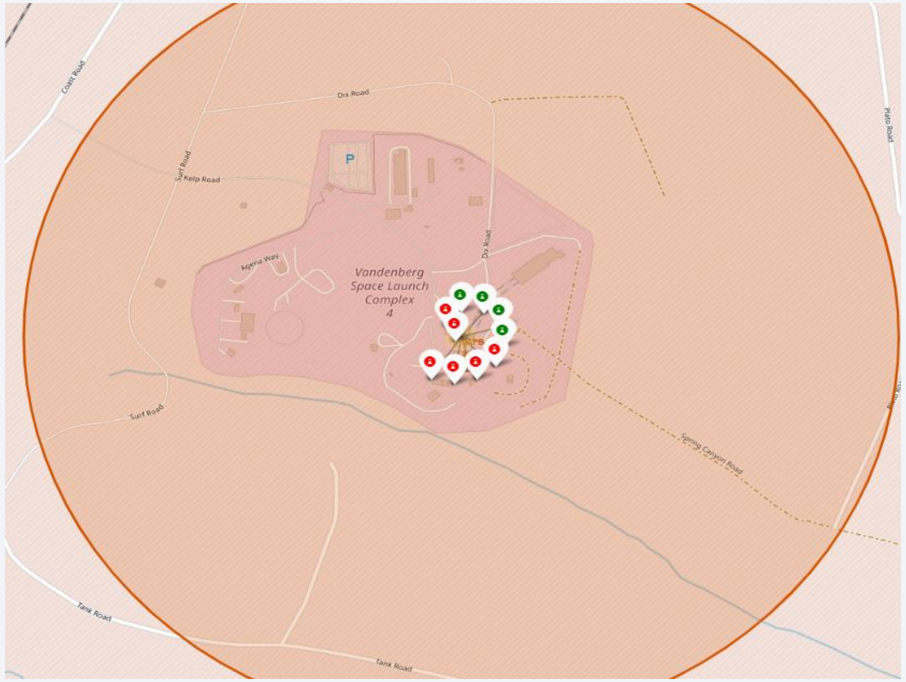
# Launch Sites Proximities Analysis

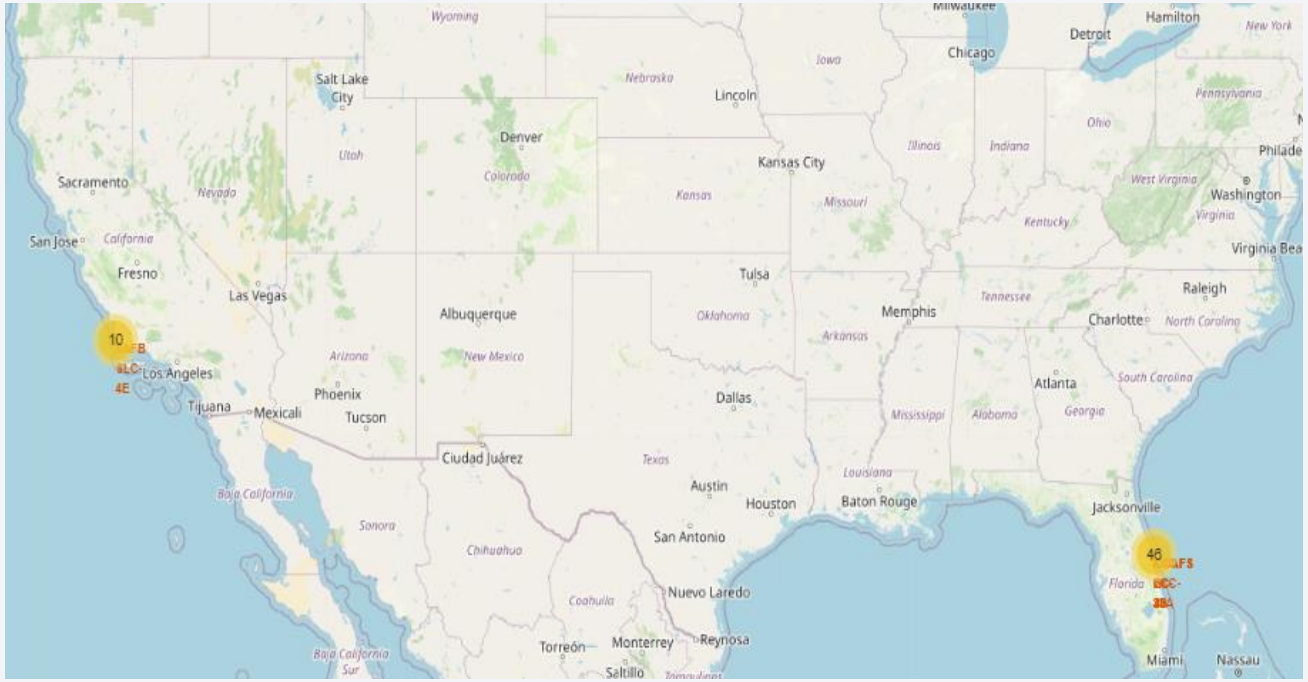
# All Launch Sites




- Launch sites are located on both the east and west coast. This is zoomed out area, however, Florida has 3 launch sites (not shown).

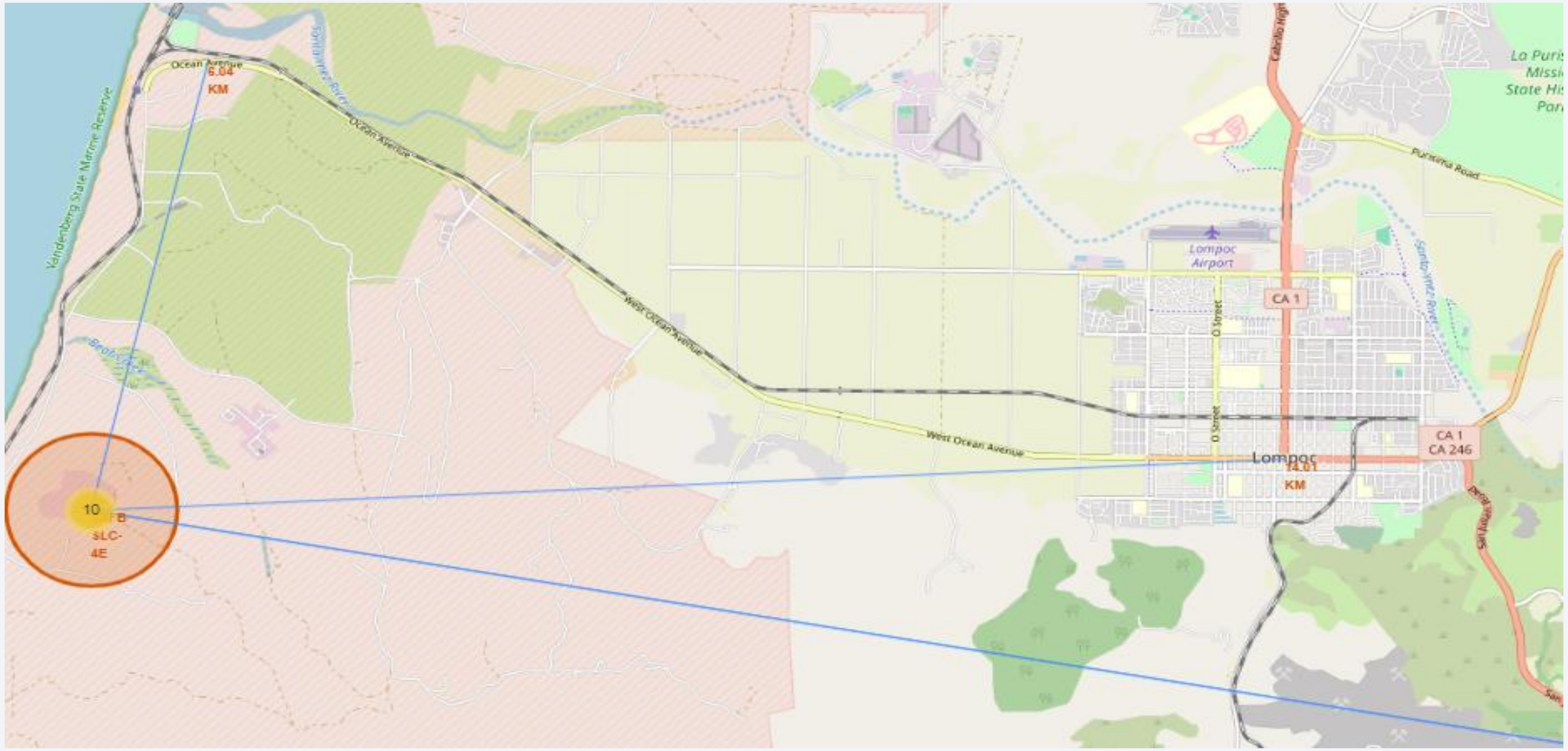
# Successful and Failed Launches per Site




- Zoom in – Map on left
  - The Vandenberg Space Launch Complete 4 had 6 failures (red) and 4 successes (green)
- Zoom Out – Map on Right
  - Shows high-level of sites: 10  (west coast) and 46 (east coast) success/failure launches.

# Launch Site to its proximities (Railway, Highway, Coastline)

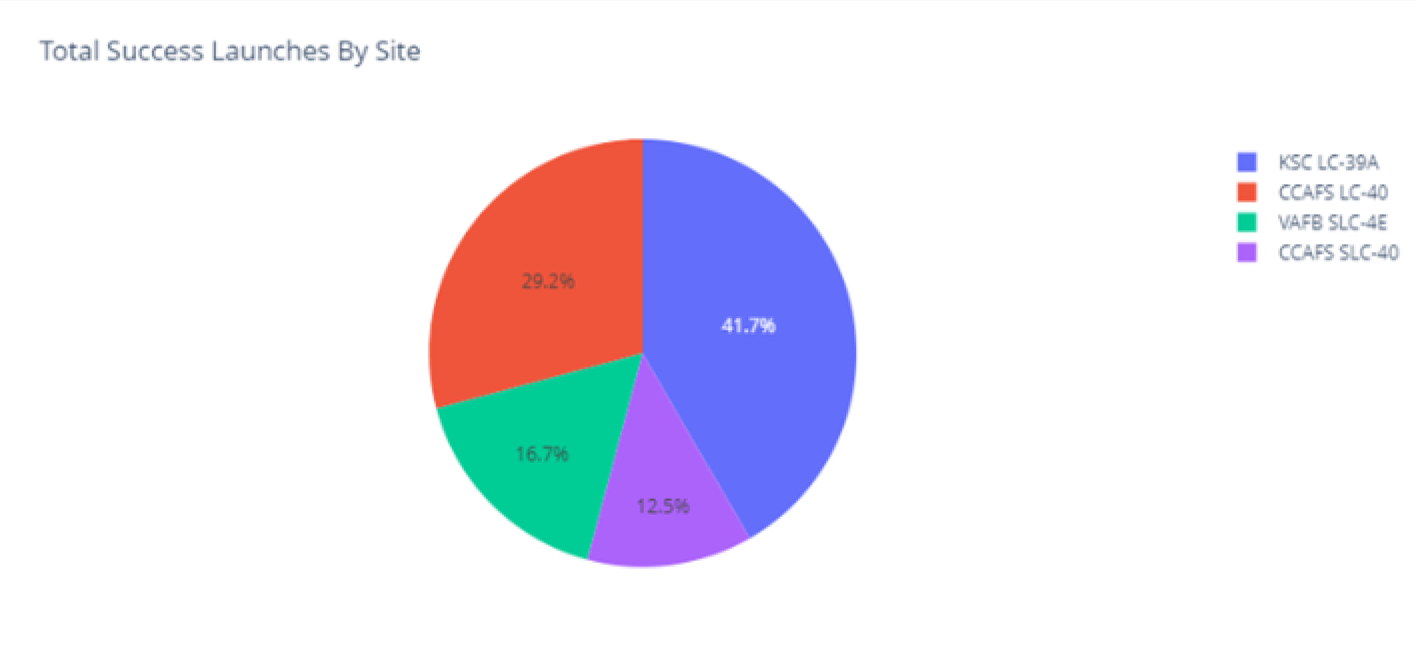



- Looking at the VAFB SLC- 4E, it can be shown that this is near major highways in multiple directions of the launch site.

Section 4

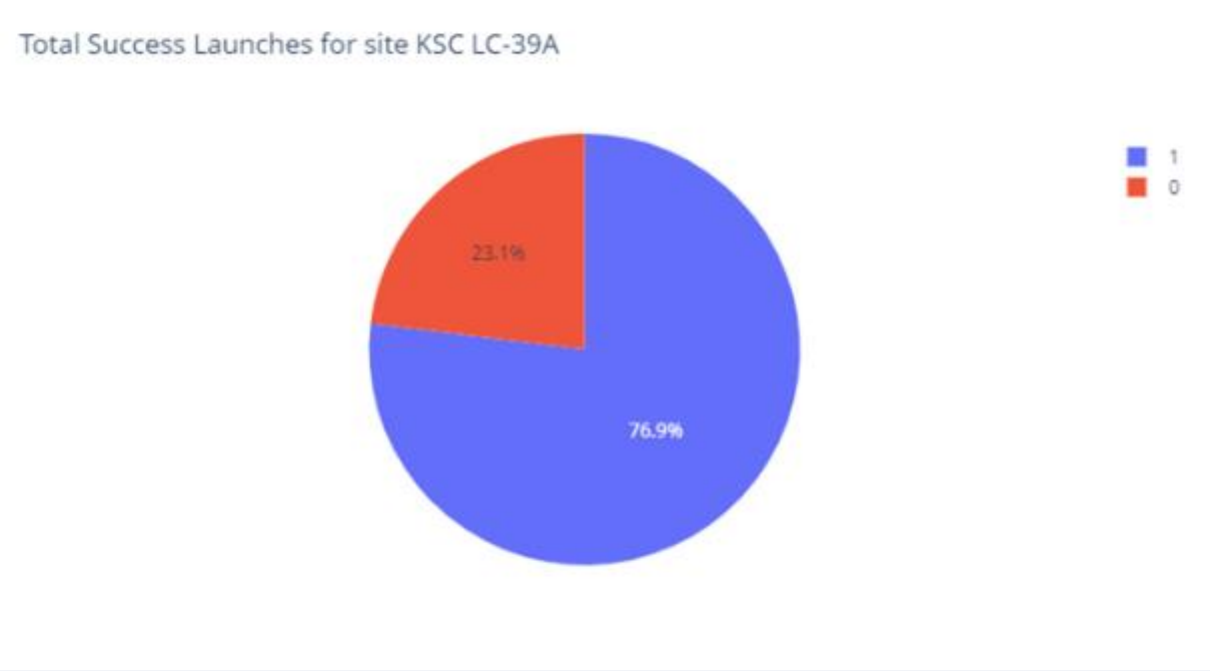
# Build a Dashboard with Plotly Dash

# Total Success Launches By All Sites



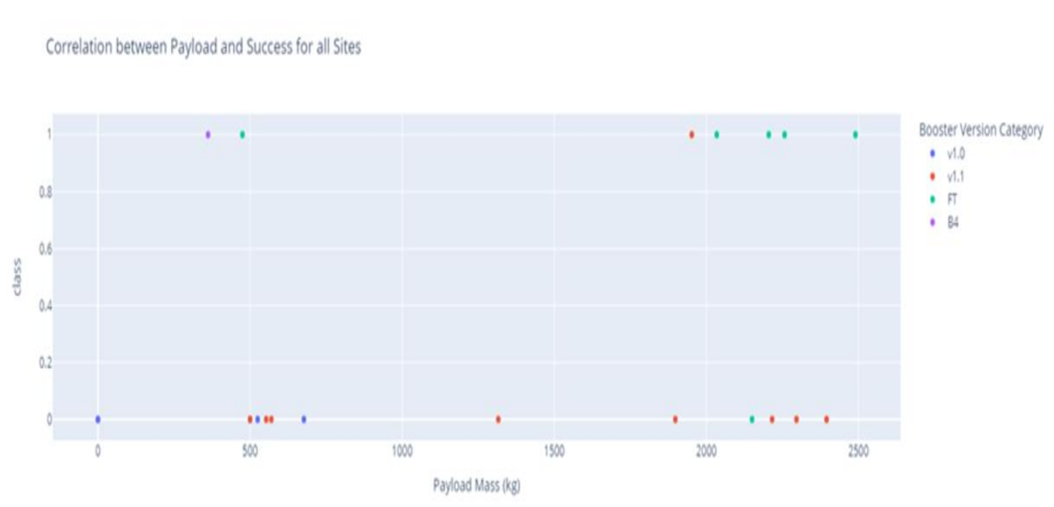

- KSC LC-39A had the greatest success at 41.7% while CCAFS SLC-40 had the lowest successful launch at 12.5%

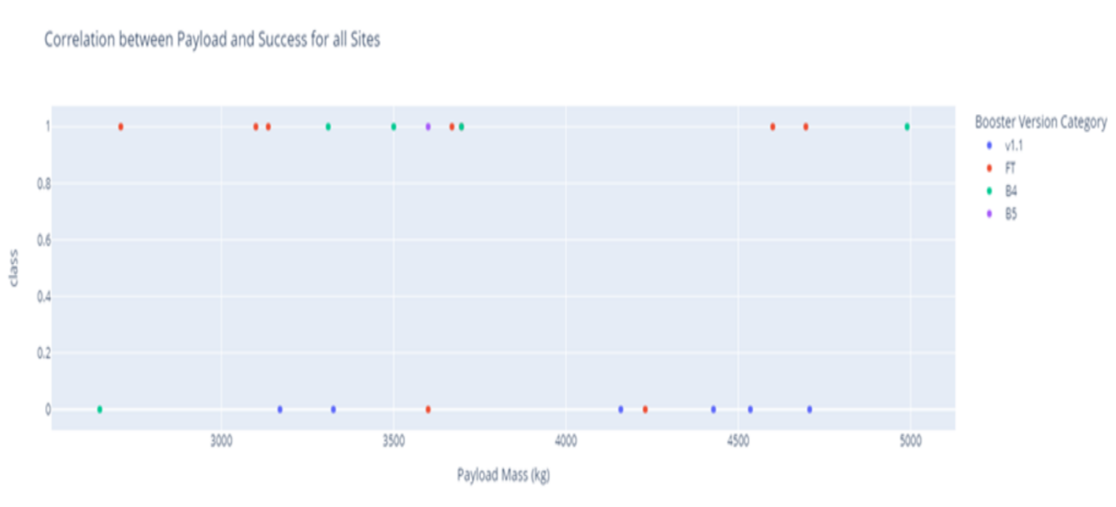
# Total Success Launches for Site KSC LC-39A

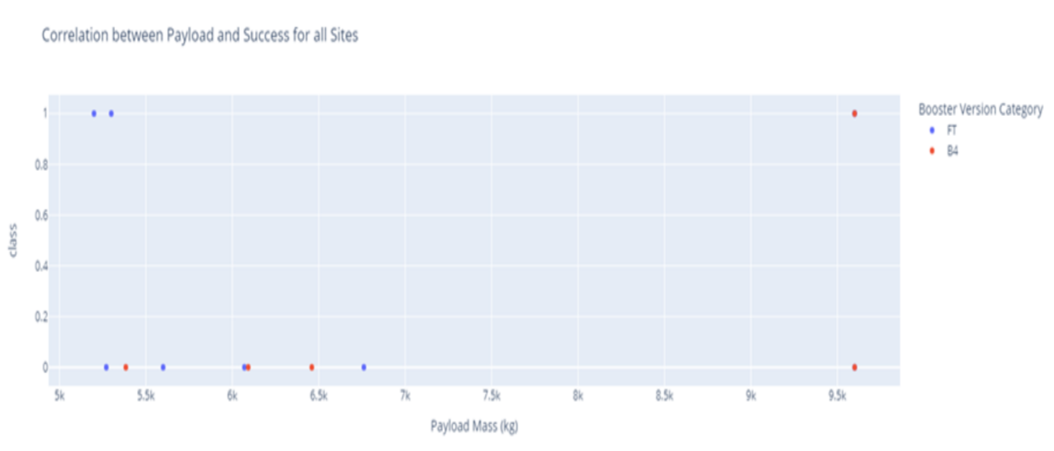



- There was a 76.9% success rate for KSC LC-39A

# Payload vs. Launch Outcome scatter plot for all sites



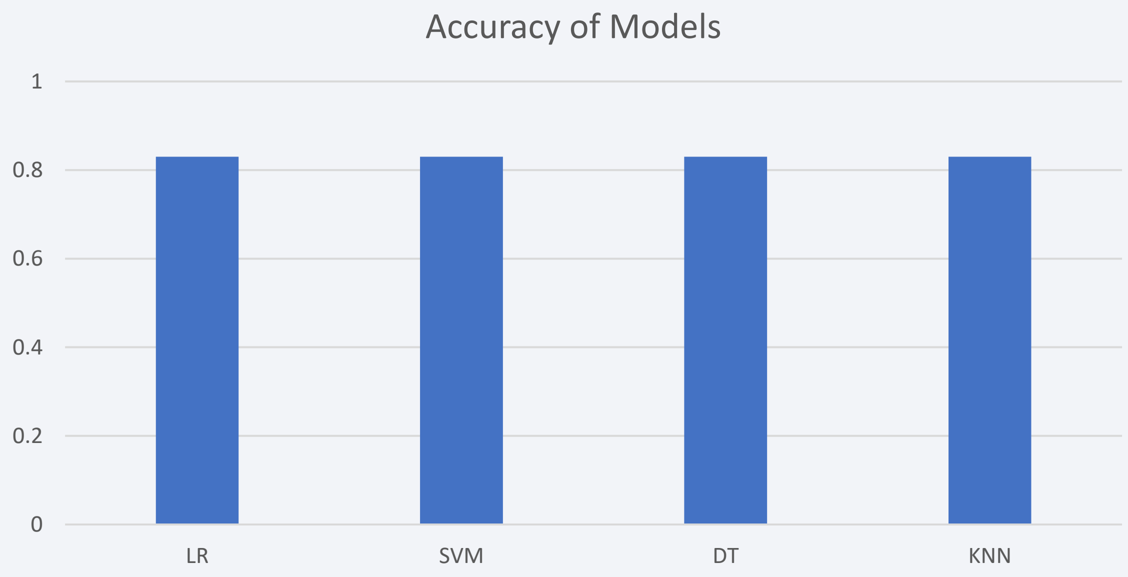

Low weighted payloads have a higher success rate then high weighted payloads. The top left graph depicts this with payloads between 0 – 2500 kg

Section 5

# Predictive Analysis (Classification)

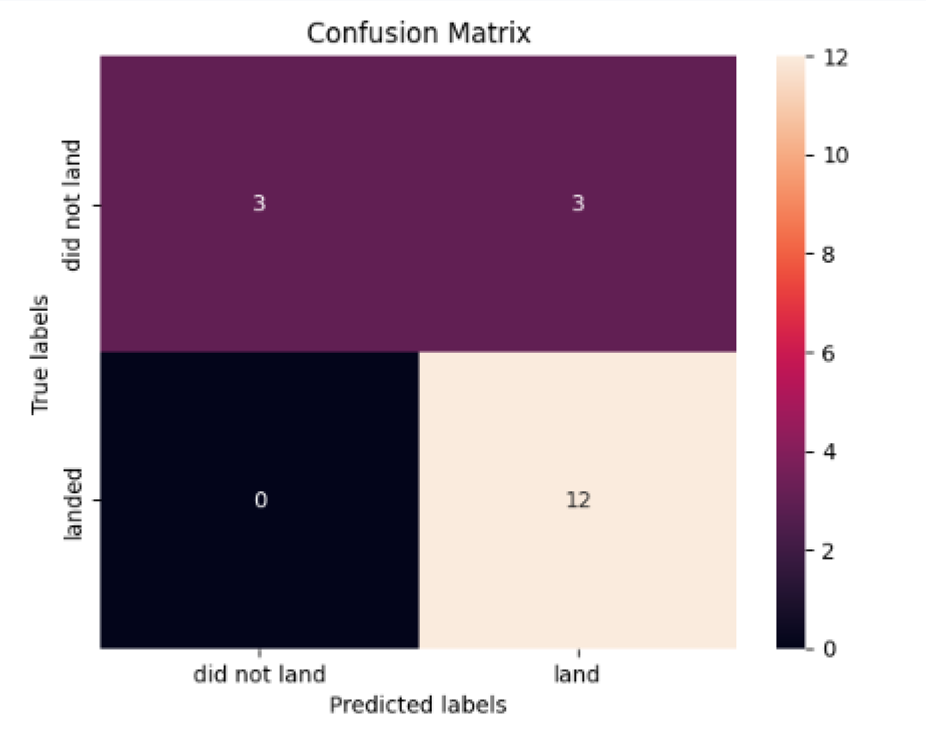
# Classification Accuracy

- All models performed at the same accuracy of 83.33%



# Confusion Matrix

- All models (4)  performed very similar. Most of the models had high false positives. These represent unsuccessful landings that are incorrectly marked as successful landing.



# Conclusions

- As the Flight Number increases there are more successful landings at each of the Launch Sites
- The success rate increased from 2013-2020
- Orbit types: Polar, LEO, and ISS had more successful landings with heavy payloads
- Orbits ES-L1, GEO, HEO, SSO, and VLEO had the highest success rate
- There was a 76.9% success rate for KSC LC-39A
- All models (4)  performed very similar. Most of the models had high false positives.

# Appendix

- Notebooks and scripts can be found in the [GitHub repository]

# Thank you!