Winning Space Race with Data Science

IBM Data Science Capstone Project

Jonathan Vowles
18 May 2021

Outline

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

Executive Summary

- **Summary of Methodologies**
    - Data Collection
    - Data Wrangling
    - EDA with Data Visualization
    - EDA with SQL
    - Building an Interactive Map with Folium
    - Building a Dashboard with Plotly Dash
    - Predictive Analysis (Classification)
- **Summary of all Results**
    - Exploratory Data Analysis of Results
    - Interactive Analysis Demonstration in Screenshots
    - Predictive Analysis Results

Introduction

- **Project Background and Context**

The commercial space age has arrived and companies are making space travel possible and affordable. Virgin Galactic, RocketLab, Blue Origin and SpaceX dominate the industry, providing suborbital spaceflights, satellites, satellite internet and reusable rockets. Possibly the most successful of these companies is SpaceX, with relatively inexpensive rocket launches, advertising Falcon 9 rocket launches at a cost of 62 million dollars. Compared to the competition charging 165 million dollars and upwards, SpaceX dominate the market by saving on the first stage of launch by recovering (or reusing) the first stage.

This crucial first step is vital to SpaceX's success and we will therefore purpose to predict the likelihood of the Flacon 9 landing successfully. If the success of the first stage can be predicted with a high-level of accuracy, we can predict the cost of a launch. This information can be utilized by competing companies seeking to contend in this field.

- **Problems we will Seek to Answer**
    - The price of each launch
    - Correlations between rocket attributes and successful landings
    - Conditions that result in successful landings

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API
  - Web Scraping from Falcon 9 and Falcon Heavy Launches Records - Wikipedia
- Perform data wrangling
  - Convert outcomes into Training Labels with the booster successfully/unsuccessfully landed
- 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 the mot effective hyperparameters for SVM (Support Vector Machines), Classification Trees and Logistic Regression

Data Collection

- The data collection process is a conglomeration of both API (Application Programming Interface) requests from the SpaceX API and web scraping from the SpaceX Wikipedia page Falcon 9 and Falcon Heavy Launches Records
  - SpaceX API Data Columns:
    - Flight Number, Date, Booster Version, Payload Mass, Orbit, Launch Site, Outcome, Flights, Grid Fins, Reused, Legs, Landing Pad, Block, Reused Count, Serial, Longitude, Latitude
  - Web Scraping:
    - Wikipedia Web Scrape Data Columns: Flight No., Launch Site, Payload, Payload Mass, Orbit, Customer, Launch Outcome, Version Booster, Booster Landing, Date, Time

Data Collection – SpaceX API

- Request Rocket Launch Data from Space X:

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

response = requests.get(spacex_url)
```

- Normalize response to a JSON. file and convert the JSON result into a data frame:

```
data = pd.json_normalize(response.json())
```

- Employ *get* functions to clean data:

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

- GITHUB URL

- Construct a dataset using the data obtained and combine the columns into a dictionary:

```
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}
```

```
launch_df = pd.DataFrame.from_dict(launch_dict)
```

- Filter the data frame and export as a CSV

```
data_falcon9 = launch_df[launch_df ['BoosterVersion'] == 'Falcon 9']
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

- Retrieve response from HTML:

```
html_data = requests.get(static_url).text
```

- Create a BeautifulSoup object:

```
soup = BeautifulSoup(html_data, 'html5lib')
```

- Find all tables and assign the result to a List:

```
html_tables = soup.find_all('table')
```

- Extract Individual column name

```
column_names = []

for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if(name != None and len(name) > 0):
        column_names.append(name)
```

- Create an Empty Dictionary with Keys

```
launch_dict= dict.fromkeys(column_names)

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

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']=[]
```

- Create a Data Frame by parsing the Launch HTML Tables
- Create a Data Frame and Export it as a CSV

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

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

- GITHUB URL

Data Wrangling

- **There are several different cases where the booster did not land successfully. Other times, a landing was attempted but failed due to an accident:**
  - True Ocean: The mission result has successfully landed in a specific area of the ocean
  - False Ocean: The mission outcome has not resulted in a successful landing in an area of the ocean
  - True RTLS: The mission result successfully landed on the ground pad
  - False RTLS: The mission outcome has not resulted in a successful landing on the ground ship
  - True ASDS: The mission result has successfully landed on the drone ship
  - False ASDS: The mission outcome has not resulted in a successful landing on the drone ship

- **Converting the Results into Training Labels**
  - 1 = Successful
  - 0 = Failure

- GITHUB URL

Data Wrangling

- Calculating the number of launches at each site

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

- Calculating the number and occurrence of each orbit

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

- Calculating the number and occurrence of mission outcome per orbit type

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

- Creating a landing outcome from the outcome column

```
landing_class = []
for outcome in df.Outcome:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

- [GITHUB URL]

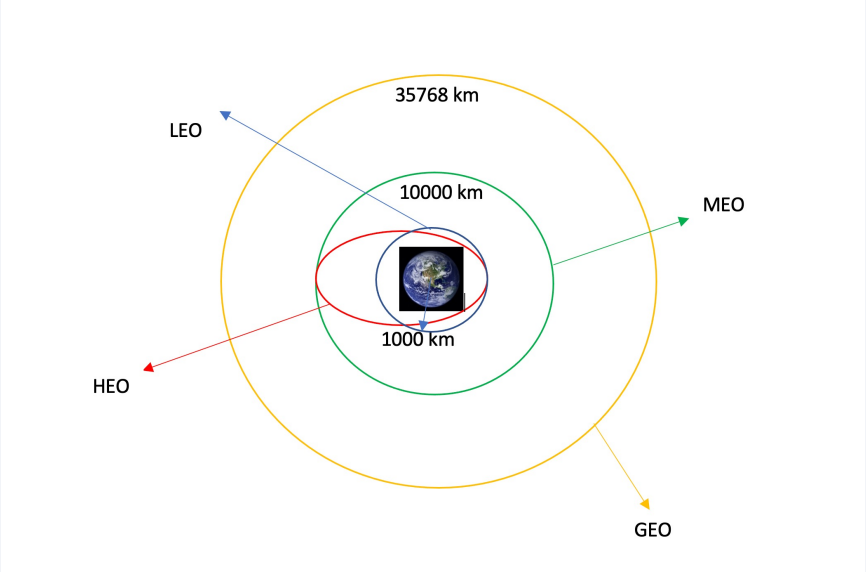
- Calculating the success rate for every landing in the dataset

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

```
0.6666666666666666
```

- Export the data to CSV

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



EDA with Data Visualization

- **Scatter Chart**
  - Flight Number vs. Launch Site
  - Payload vs. Launch Site
  - Flight Number vs. Orbit Type
  - Pay Load vs. Orbit Type
  - A scatter plot demonstrates how one variable is affected by another variable. This interrelationship is known as correlation and is generally composed of large data sets.

- **Bar Chart**
  - Orbit Type vs. Success Rate
  - A bar chart allows us to compare datasets and allows for simple visual evaluation. The first axis represents a data category, while the other axis represents a discrete value. The purpose of the bar chart is to display the relationship between the two axis

- **Line Chart**
  - Years vs. Success Rate
  - A line chart displays data variables and trends, helping us predict the results of data may have not yet been recorded.

- GITHUB URL

EDA with SQL

- **Loading the Data Set into the Corresponding Table in the Db2 Database, Executing SQL Queries to Answer the Following Questions:**
  - Displaying the names of the unique launch sites in the space mission
  - Displaying 5 recordings where launch sites begin with the string ‘CCA’
  - Displaying the total payload mass carried by boosters launched by NASA (CRS)
  - Displaying average payload mass carried by booster version F9 v1.1
  - List the date when the first successful landing outcome on the ground pad was achieved
  - List the names of the boosters which have success on the drone ship and have a pay load mass greater than 4000 and less than 6000
  - List the total number of successful and unsuccessful mission outcomes
  - List the names of the booster versions that have carried the maximum payload mass
  - List the failed landing outcomes on the drone ship, their booster versions and launch site names for the year 2015
  - Ranking the count of landing outcomes (drone ship failure or ground pad success) between the dates 04/06/2010 and 20/03/2017, in descending order

- GITHUB URL

Build an Interactive Map with Folium

- **Objects Created and Added to a Folium Map:**
  - Markers that show all launch sites on a map
  - Markers that show the successful and failed launches for each site on the map
  - Lines that show the proximities between launch sites

- **By Adding these Objects the Following Geographic Patterns about Launch Sites are Found:**
  - Launch sites are in close proximity to railways
  - Launch sites are in close proximity to highways
  - Launch sites are in close proximity to coastlines
  - Launch sites keep specific distance from cities

- GITHUB URL

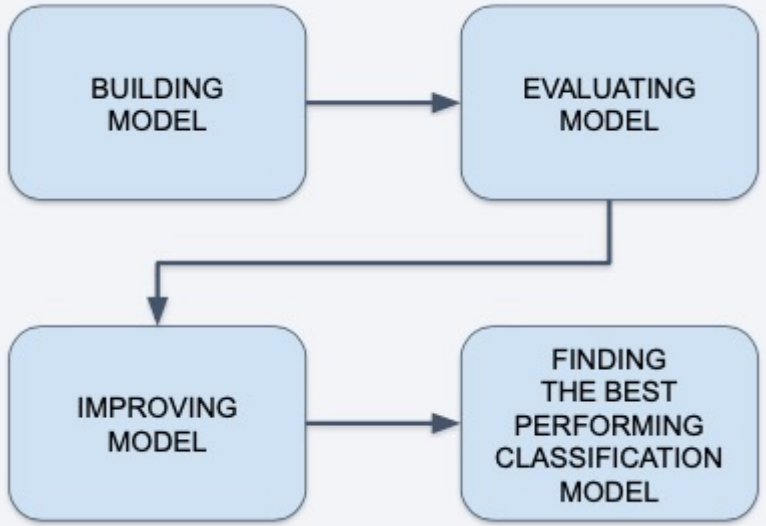
Build a Dashboard with Plotly Dash

- **The Dashboard Application Contains a Pie Chart and a Scatter Point Chart:**
  - Pie Chart
    - For showing total success launches by sites
    - This chart is useful to indicate a successful landing distribution across all launch sites or to indicate the success rate of individual launch sites
  - Scatter Chart
    - For showing the relationship between Outcomes and Payload Mass (Kg) by different boosters
    - The scatter chart has 2 inputs: All sites/individual sites and payload mass on a slider between 0 and 10000 kg
    - The scatter chart helps estimate how each launch site, payload mass and booster version categories determine a successful launch

- GITHUB URL

Predictive Analysis (Classification)

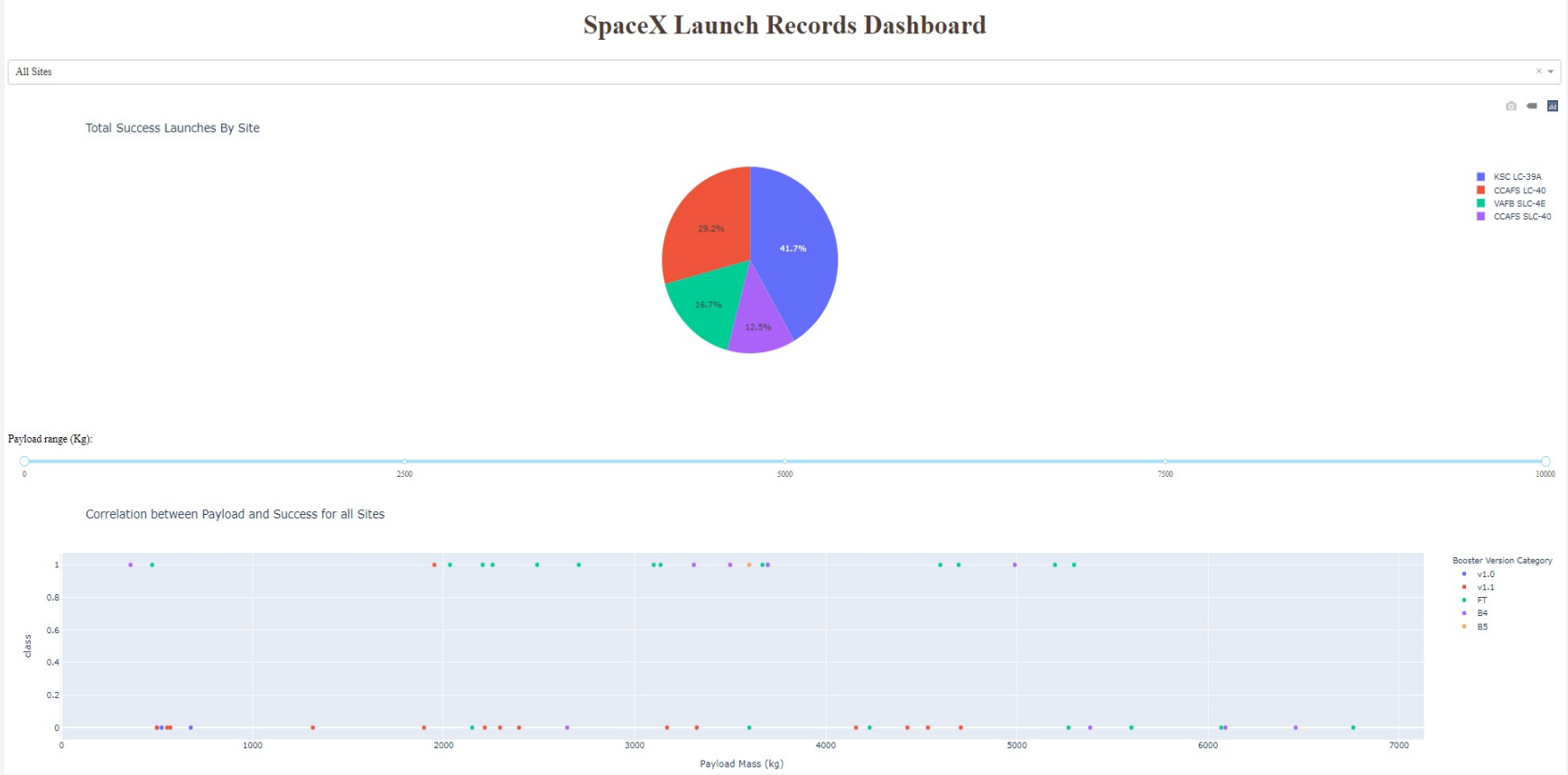
- **First, We Explore Data Analysis and Determine Training Labels**
  - Create a column for class
  - Standardize the data
  - Split into training data set and test data set
- **Find the Best Hyperparameter for SVM, Classification**
  - Trees and Logistic Regression
  - Find the Method that performs best using test data

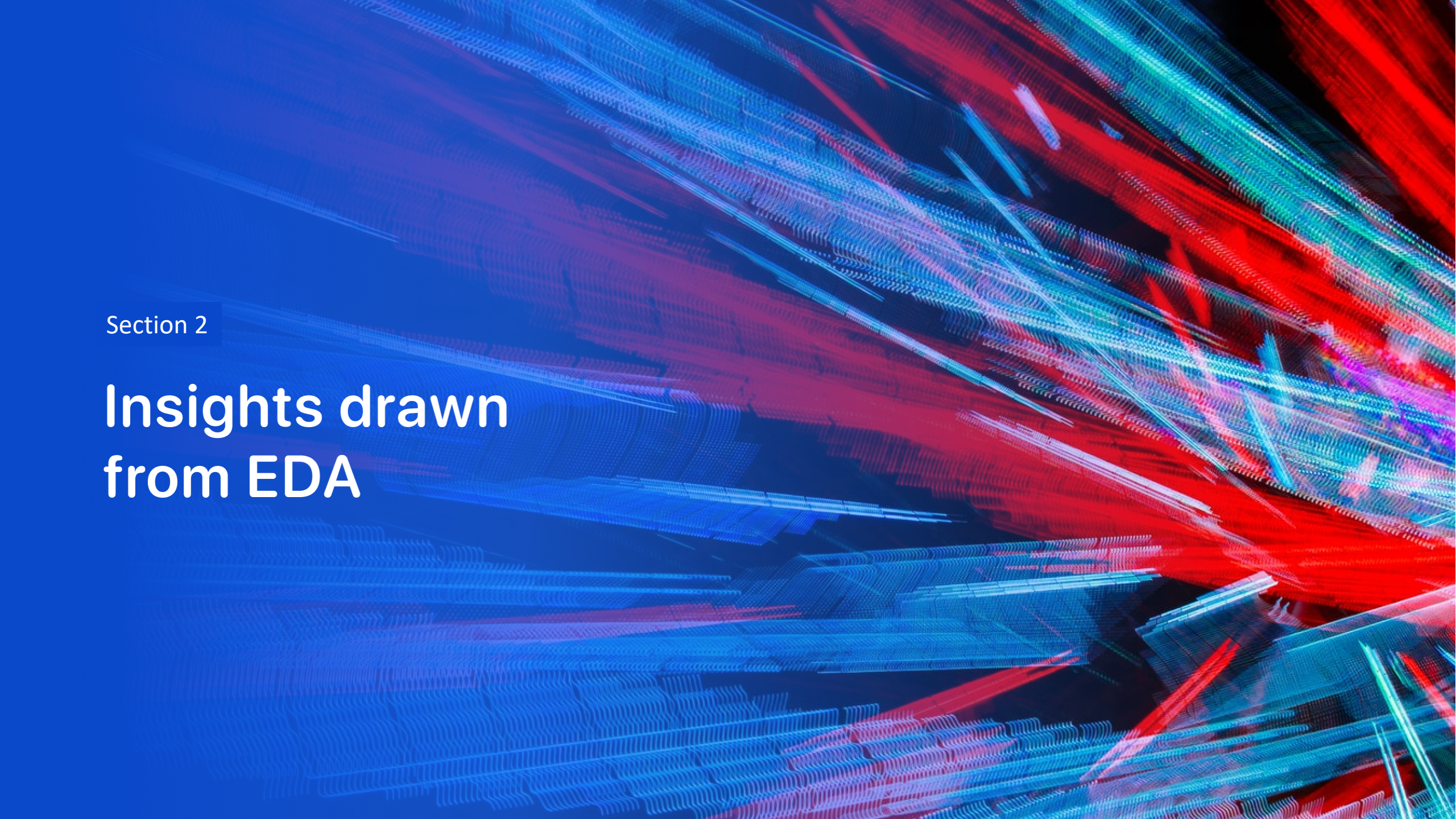


- GITHUB URL

Results

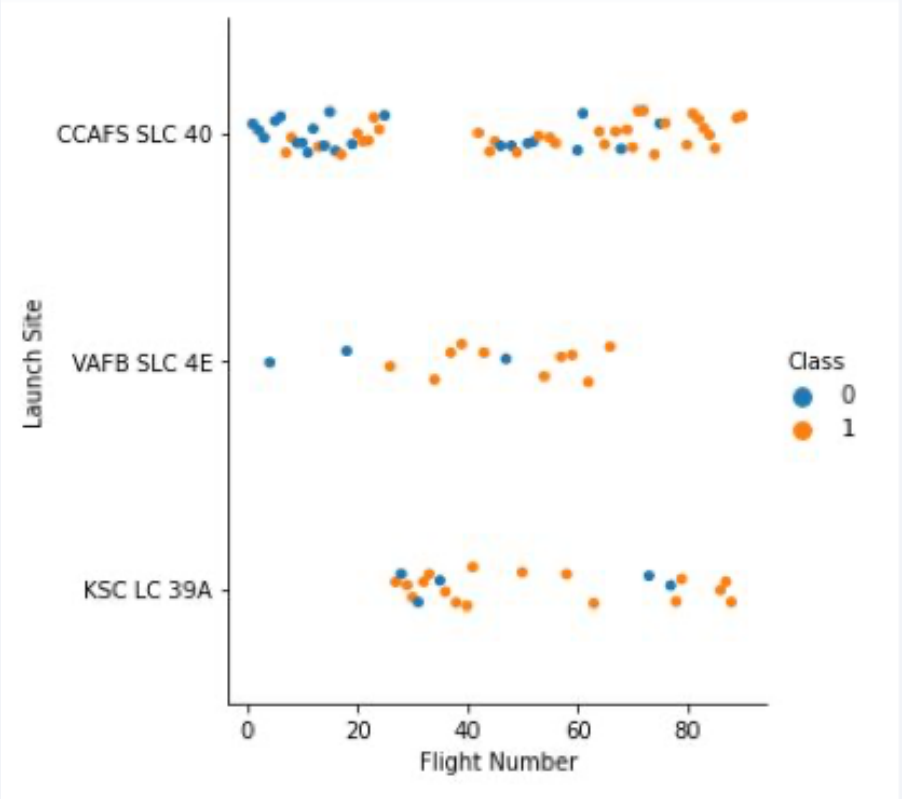
- The Image Shows a Preview of the Dashboard made with Plotly Dash
- The results of EDA with Visualization, EDA with SQL, the Interactive Map with Folium will be Included in the Upcoming Slides
- The Four Methods Mentioned Above return an Accuracy of 83% for the Test Data




Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

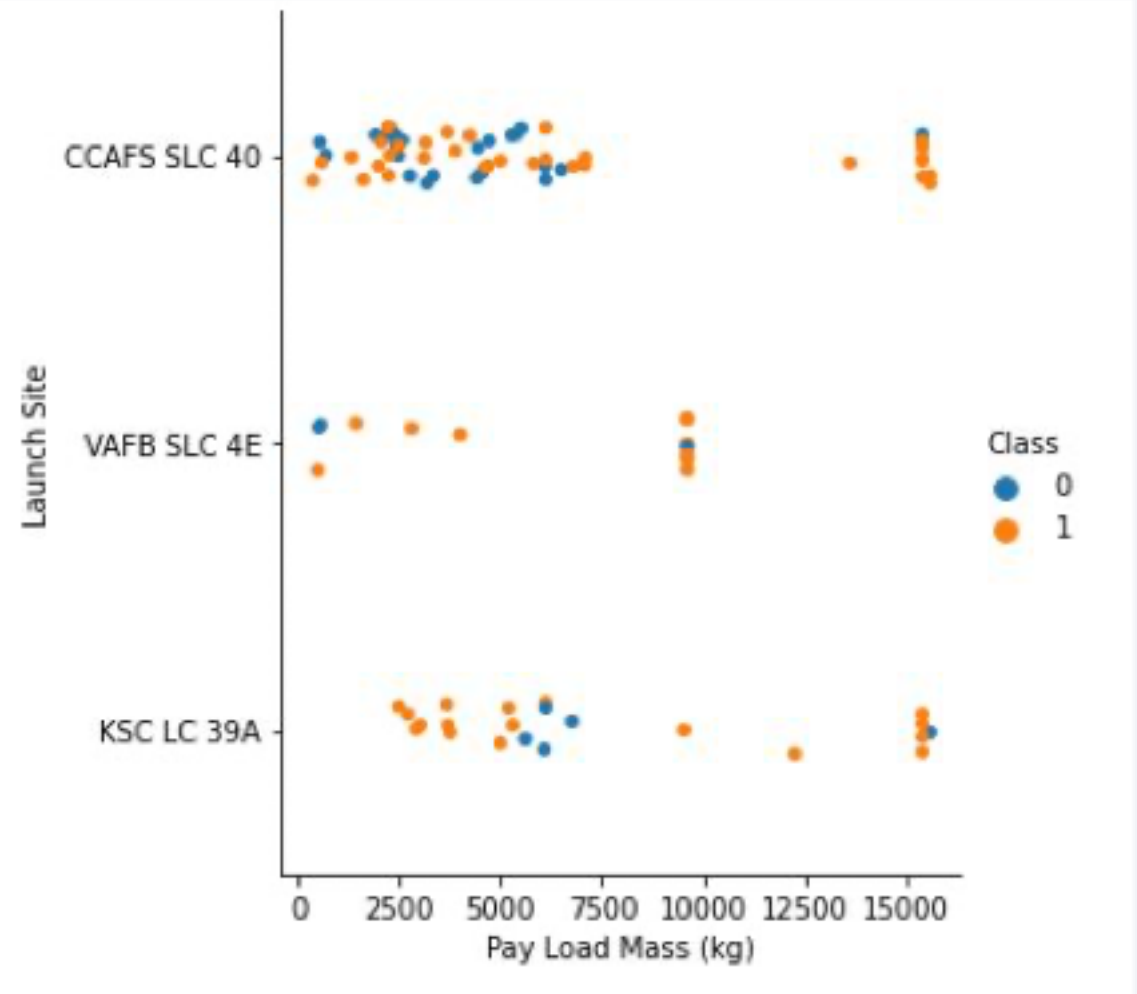
- **Class 0 (Blue):**
  - Represents an unsuccessful launch
- **Class 1 (Orange):**
  - Represents a successful launch
- The figure presented demonstrates that the success rate increased as the number of flights increased
- The success rate increases drastically at the 20th flight



Payload vs. Launch Site

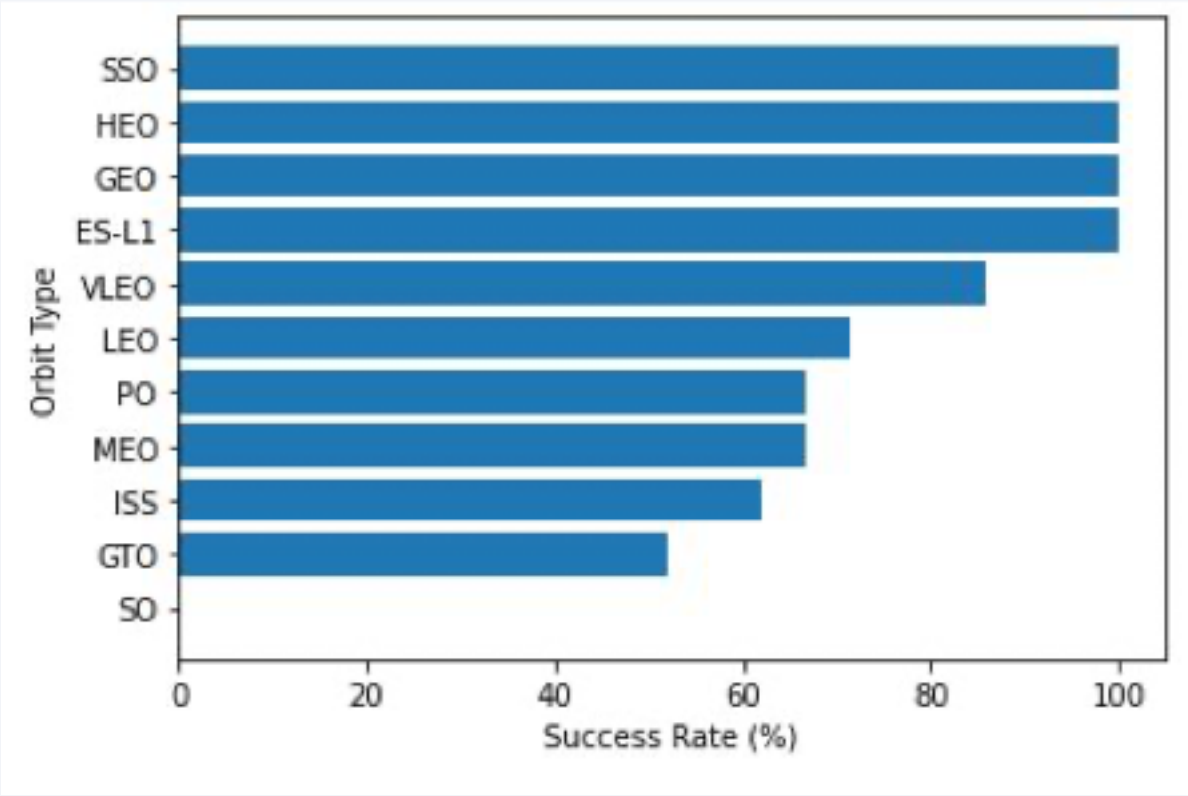
- **Class 0 (Blue):**
  - Represents and unsuccessful Launch
- **Class 1 (Orange):**
  - Represents a successful launch

While the larger pay load mass would seem to indicate a higher rocket success rate there is no clear correlation between a successful launch and the pay load mass (kg)



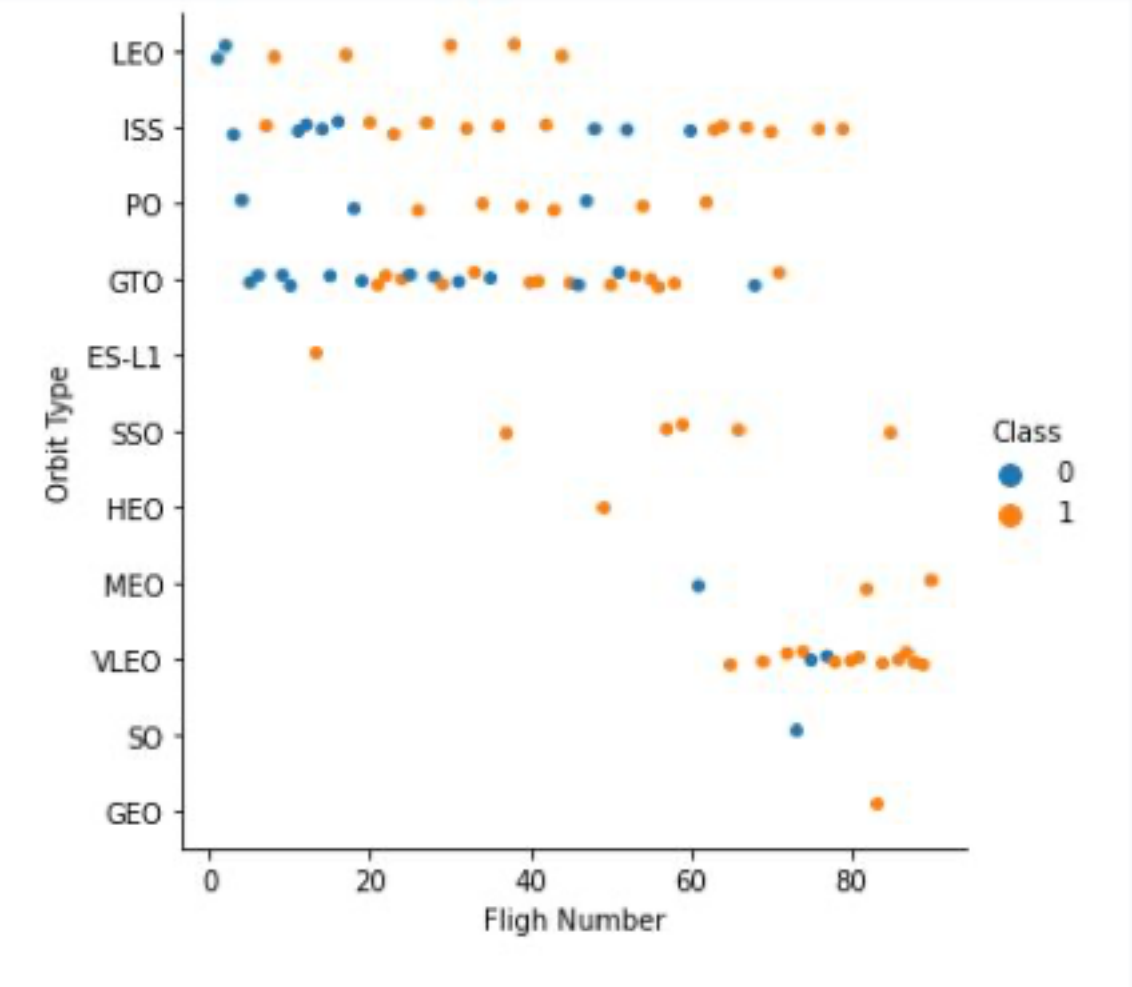
Success Rate vs. Orbit Type

- **Orbit Types:**
  - SEO (Sun-Synchronous Orbit), HEO (Highly Elliptical Orbit), GEO (Geostationary Orbit), ES-L1 (Lagrangian) have 100% success rate
  - The success rate of the GTO (Geostationary Transfer orbit) type has a 50% success rate
  - The lowest success rate is the type SO Orbit



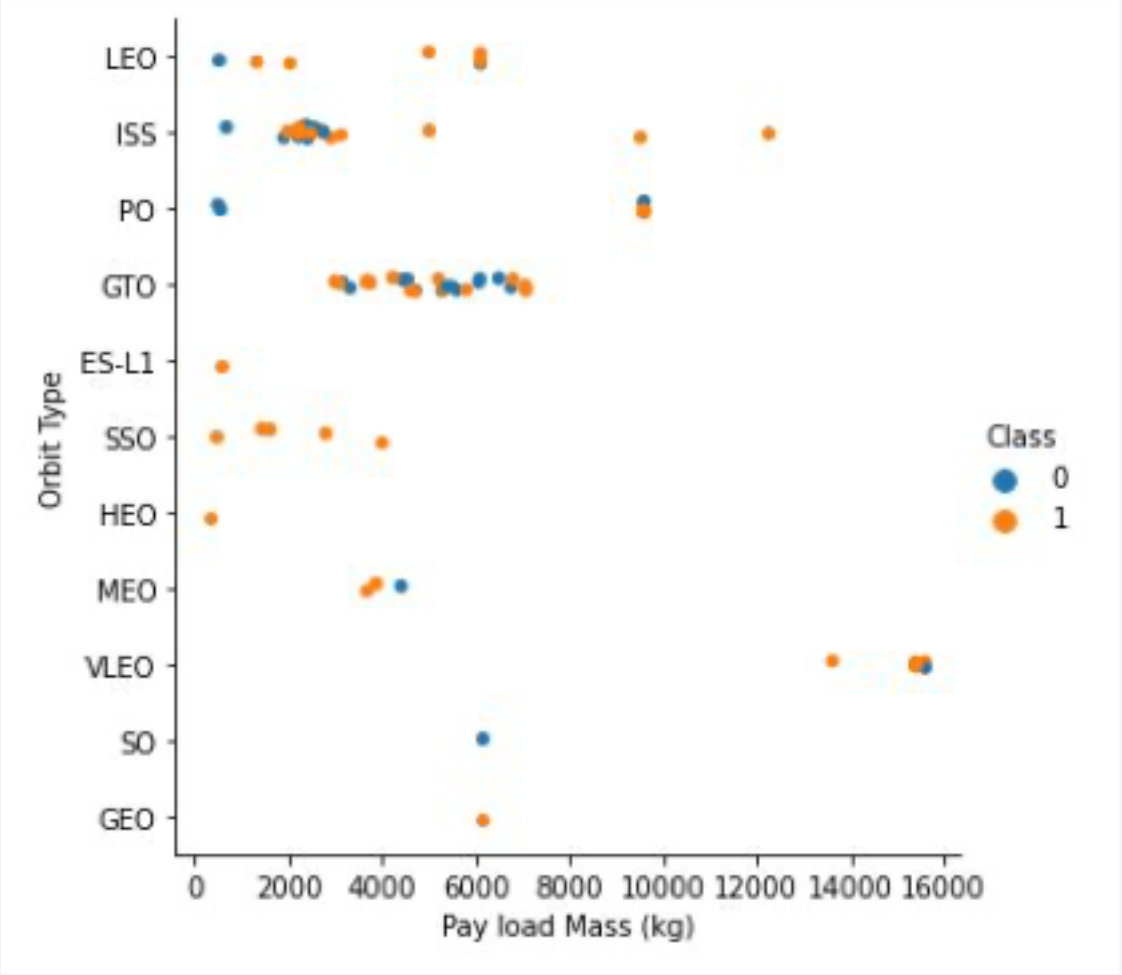
Flight Number vs. Orbit Type

- **Class 0 (Blue):**
  - Represents an unsuccessful launch
- **Class 1 (Orange):**
  - Represents a successful launch
- There appears to be a correlation between launch outcome and flight number
- GTO (Geosynchronous Transfer Orbit) is an exception, with no demonstrable correlation between flight number and success rate
- VLEO (Very Low Earth Orbit) demonstrates the highest success rate and is used in the most recent launches



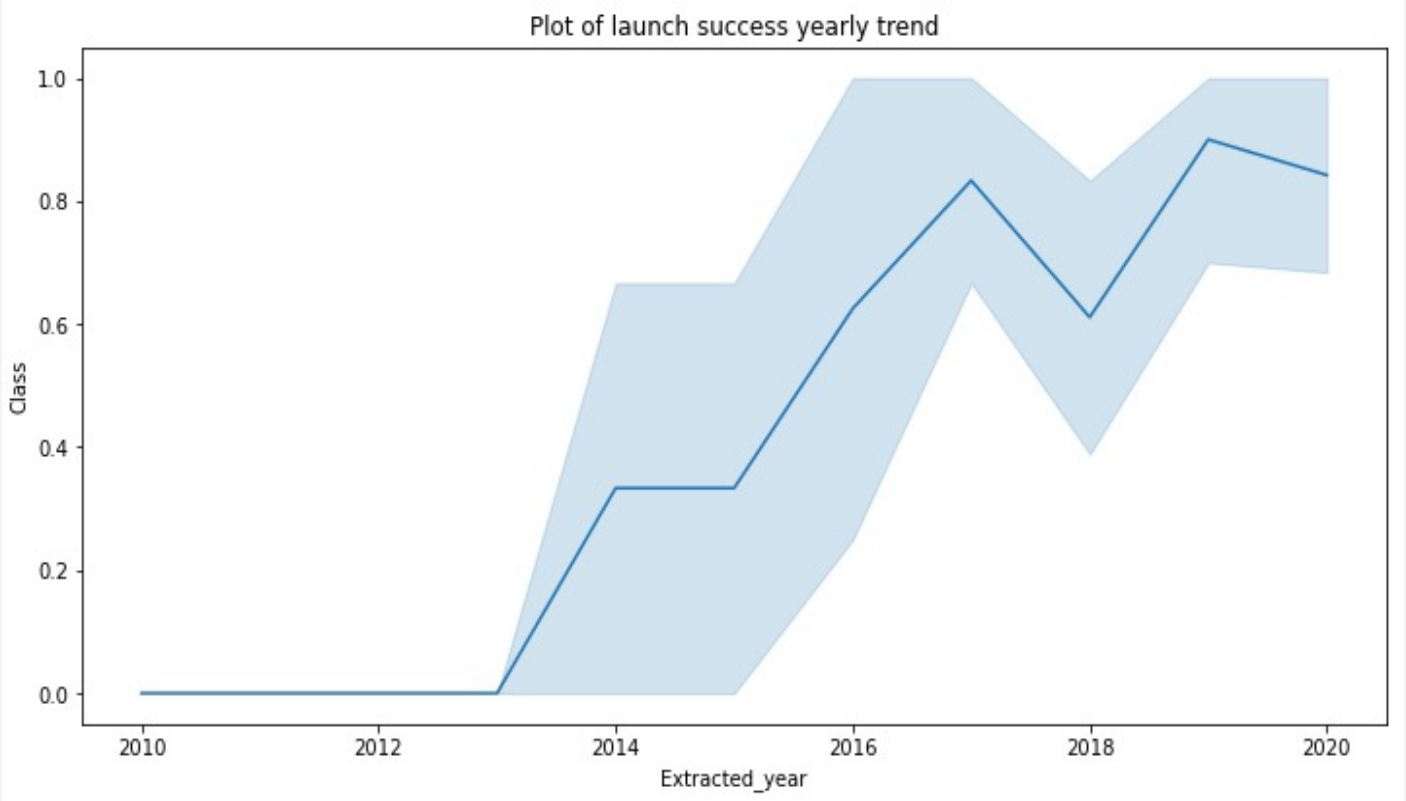
Payload vs. Orbit Type

- **Class 0 (Blue):**
  - Represents an unsuccessful launch
- **Class 1 (Orange):**
  - Represents a successful launch
- LEO (Low Earth Orbit) and ISS (International Space Station Orbit) successful landings are associated with lighter Pay Load Mass (Kg)
- GTO (Geosynchronous Transfer Orbit) is difficult to distinguish as the data is evenly spread between blue and orange



Launch Success Yearly Trend

- The success rate of rocket launches has continued to increase from 2013 till 2017
- A sharp decline in success rate is observed from 2017 to 2018
- The success rate is then seen increasing to an all time high in 2019 and declines steadily from then
- To date, a success rate of roughly 80% is observable



All Launch Site Names

- The SQL DISTINCT clause is used for this query to extract unique values pertaining to the Launch Sites from the SpaceX table

```
%%sql
SELECT DISTINCT LAUNCH_SITE
FROM SPACEXTBL
```

- This query produces four unique launch sites:
  - CCAFS LC-40
  - CCAFS SLC-40
  - KSC LC-39A
  - VAFB SLC-4E

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

Launch Site Names Begin with 'CCA'

- SELECT * SQL query is used to display the site names from the SpaceX records
    - LIMIT 5 is employed to present the first five launch sites
- The LIKE operator and ‘CCA%’ are called to present the launch sites

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

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Employing the SUM() function to calculate the sum of column PAYLOAD_MASS_KG from the SpaceX table

- The WHERE clause will filter the dataset to perform calculations where the customer is NASA (CRS)

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

- RESULT:

total_payload_mass_kg
45596

Average Payload Mass by F9 v1.1

- Employing the AVG() function we will calculate the average value of the column PAYLOAD_MASS_KG
- The WHERE clause will filter the dataset to perform calculations only if Booster_version is F9 v1.1

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

- RESULT:

avg_payload_mass_kg
2928

First Successful Ground Landing Date

- Employing the MIN() function to calculate the earliest date in the column DATE
- The WHERE clause will filter the dataset to perform calculations where the Landing_outcome is successful with regards to the ground pad

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

- RESULT:

first_successful_landing_date
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

- Employing the SELECT() clause to filter results specifically to BOOSTER_VERSION
- The WHERE clause will filter the dataset to produce results where the Landing_outcome is ‘success’ according to the drone ship
    - The AND operator will specify results where the condition is the PAYLOAD_MASS_KG is between 4000 and 6000

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

- RESULT:

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

Total Number of Successful and Failure Mission Outcomes

- Employing the COUNT() function to calculate the total number of columns
- The GROUP BY statement will filter the dataset into summary rows for rows that contain the same values in order to find the total sum in each Mission_outcome

```
%%sql
SELECT MISSION_OUTCOME, COUNT(*) AS total_number
FROM SPACEXTBL
GROUP BY MISSION_OUTCOME
```

- RESULT:
  - SpaceX has successfully completed close to 99% of its missions

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

Boosters Carried Maximum Payload

- In this section we will employ a subquery to determine the maximum value of the payload. This will be done using the MAX() function
- Secondly, we will limit the dataset to search if PAYLOAD_MASS_KG is the maximum value of the payload
- RESULT:
    - Version F9 B5 B1048.4 – F9 B5 B1060.3 boosters are capable of carrying the maximum payload

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

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

2015 Launch Records

- Using the WHERE clause, we will filter that dataset to perform a search if the Landing_outcome is Failure with regards to the drone ship
    - The AND operator will display a record if the additional year is 2015
- In 2015, there was one landing failure on drone ships

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

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

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

- The COUNT function selects the landing outcomes, while the WHERE clause will filter the dataset to conduct a search between 04/06/2010 and 20/03/2017
- The ORDER BY clause was selected to sort the total records by number of landings – sorted in descending order

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

- There is minimal deviation in results between the number of successful and unsuccessful launches between 04/06/2010 and 20/03/2017

landing__outcome	total_number
Success	20
No attempt	11
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4

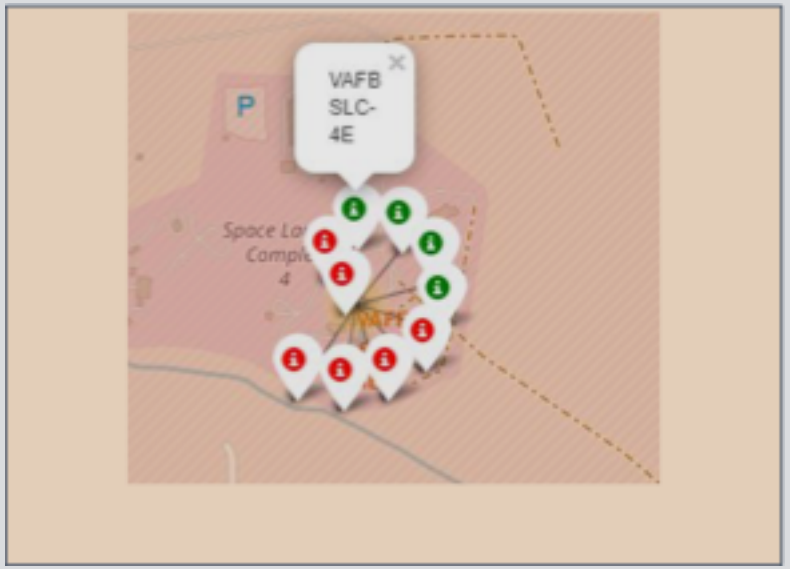

Section 3
Launch Sites
Proximities Analysis

SpaceX Launch Site Locations

- The Folium map displays all SpaceX launch sites
- It is evident that all launch sites are coast-bound

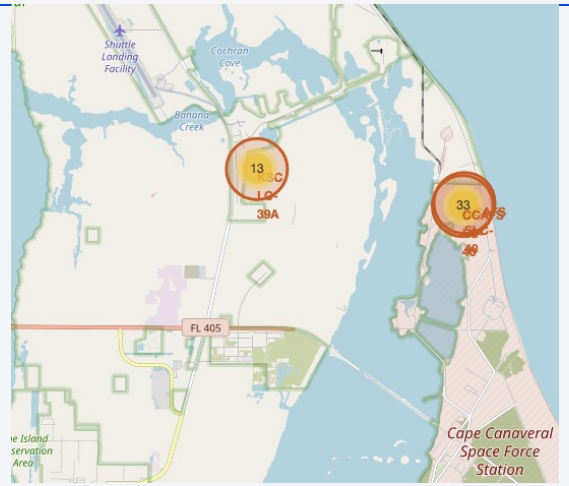
Launch Site Outcomes

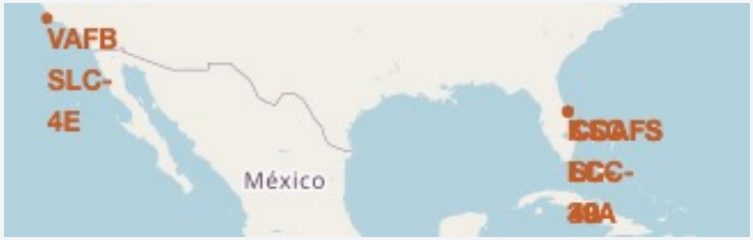
- **Successful landings (green):**
- **Failed landings (red):**

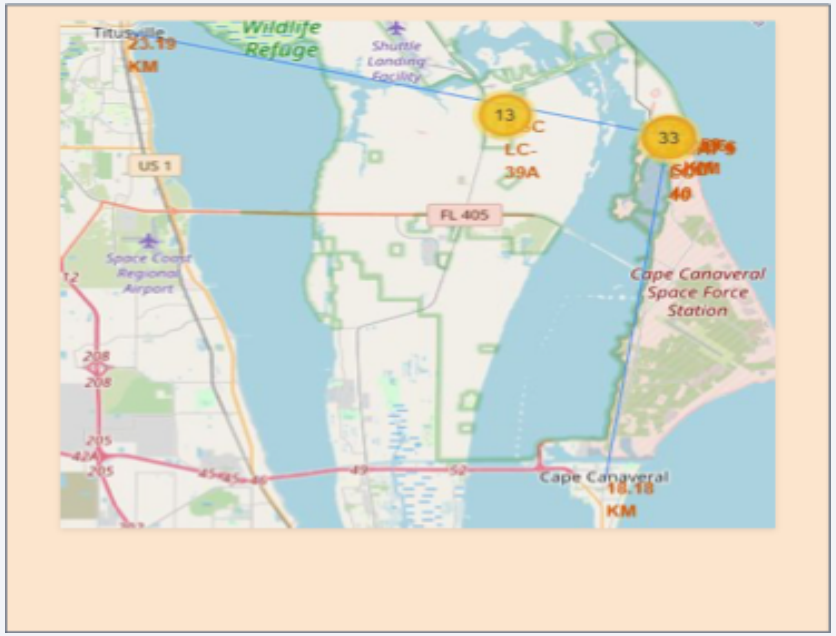


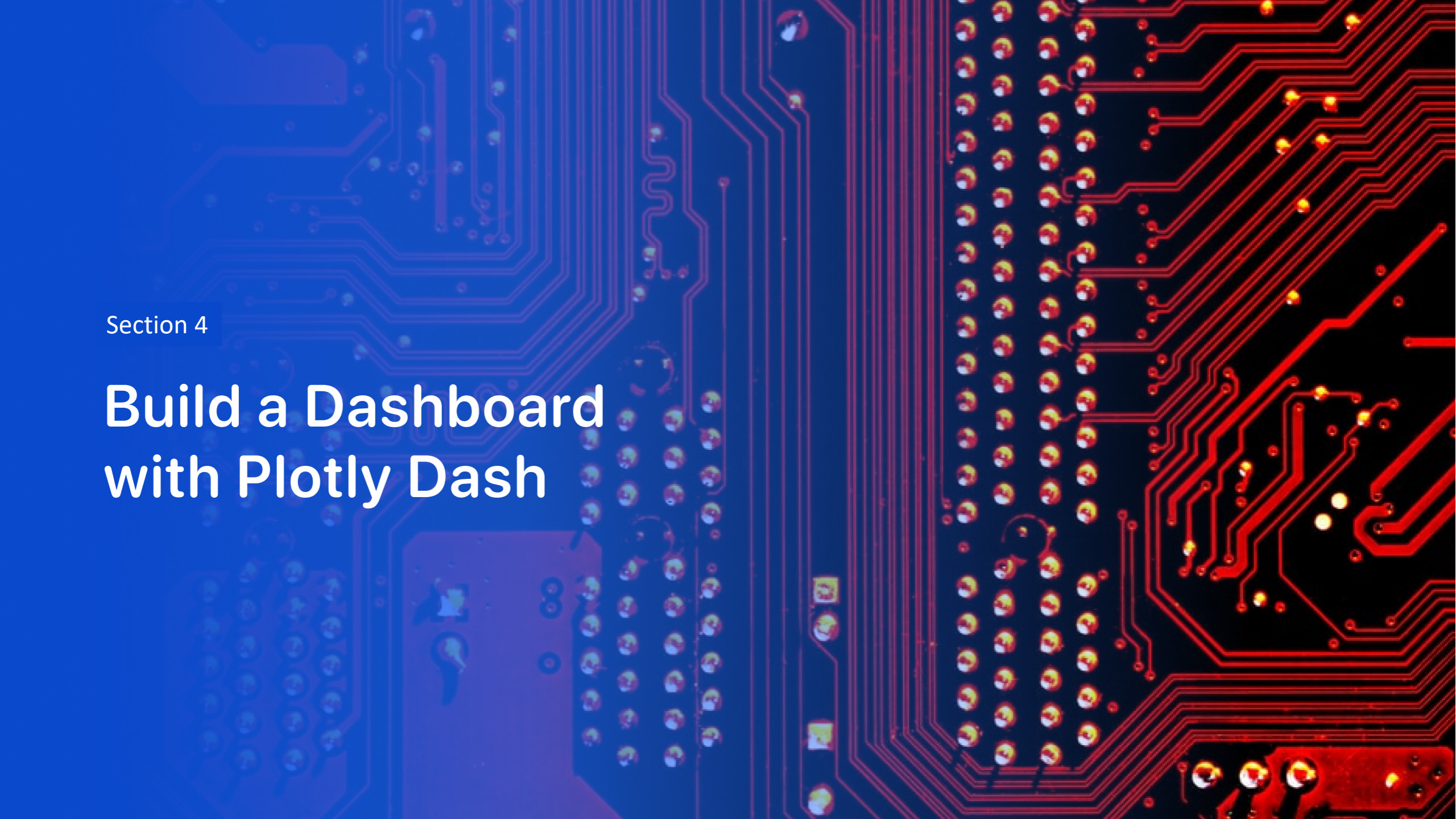
Launch Site Proximity

- Launch sites are predominantly close to railways and highways. It may be hypothesized that this is due to ease of transportation for equipment and personnel
- Proximity to the coastline and a safe distance from cities allows for safe launch without compromising the safety of citizens located in the nearby towns and cities.



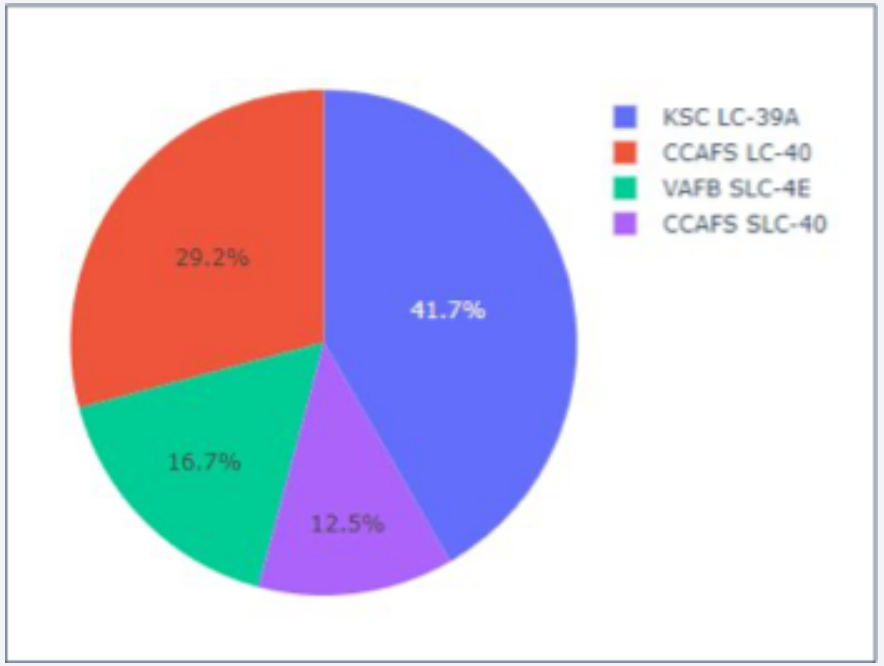





Section 4
Build a Dashboard with Plotly Dash

Successful Launches – All Sites

- KSLC - 39A has the highest success rate among all sites
- VAFB SLC-4E has the least success. This is likely due to:
    - A small data sample
    - It is the only site located in California, hence the West Coast may pose more difficulty than the East Coast, due to unknown factors

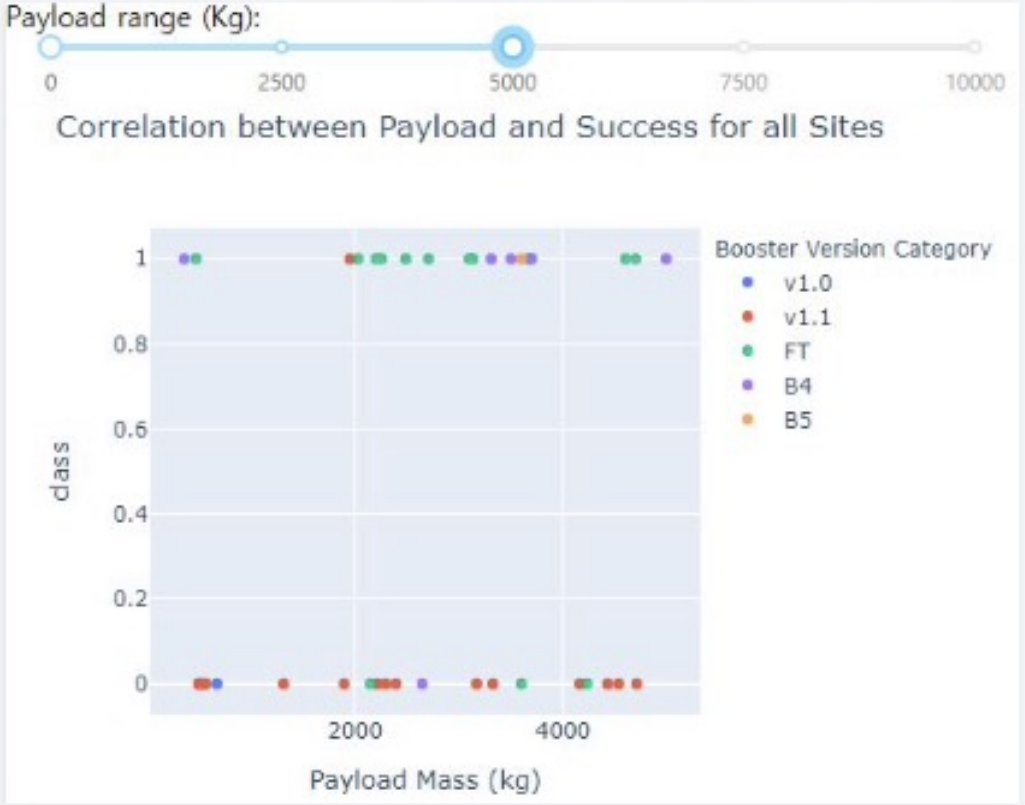


Laugh Site – Highest Success Rate

- KSLC – 39A holds the highest success rate:
  - 10 Successful Landing (76.9%)
  - 3 Landing Failures (23.1%)

Payload and Launch Outcome Scatter Plot

- The following graphs illustrate that the launch success rate (class 1) – for low weighted payloads (0 – 5000kg) – is higher than that of heavy weighted payloads (5000 – 10000 kg)

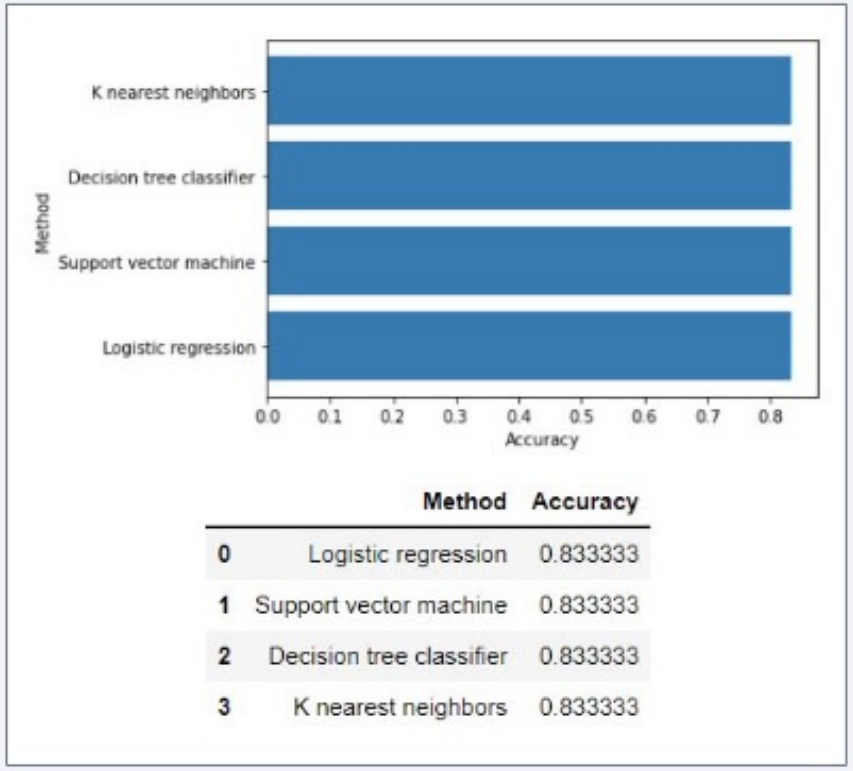


Section 5

Predictive Analysis (Classification)

Classification Accuracy

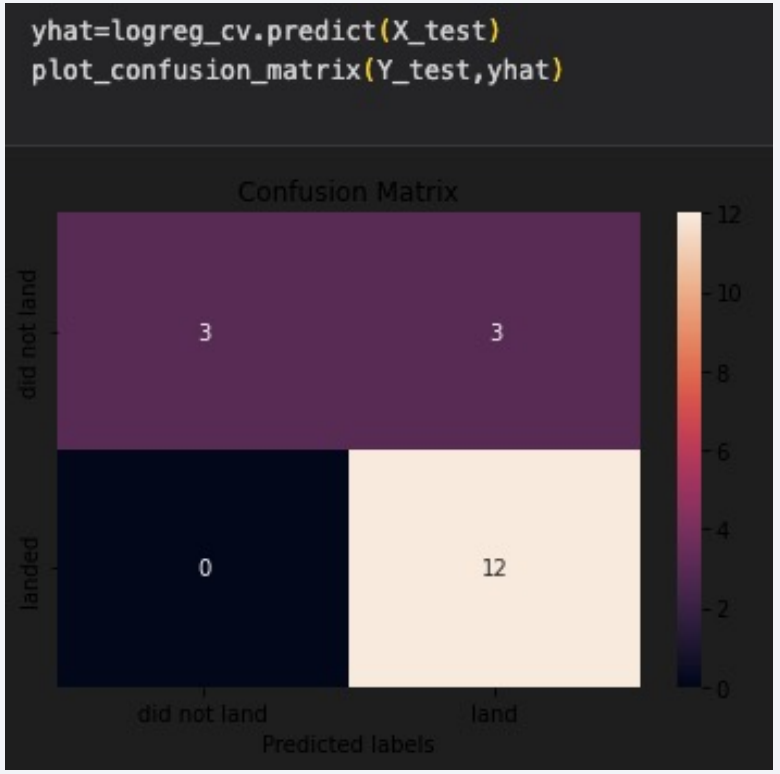
- The test set produced as accuracy of approximately 83.33%
- Due to the small test set (18) more data is needed to determine the ideal model



	Method	Accuracy
0	Logistic regression	0.833333
1	Support vector machine	0.833333
2	Decision tree classifier	0.833333
3	K nearest neighbors	0.833333

Confusion Matrix

- As all models were used for the same test set, the confusion matrix is the same for all models
- However, the confusion matrix for the decision tree demonstrates that the classifier can distinguish the different classes
- False positives remain a concern as unsuccessful landings may be marked as successful by the classifier
- Ultimately, these models predict successful landings



Conclusions

- As the number of flights increase, the success rate increased
    - Recently this number has exceeded 80%
- Orbital types SSO, HWO, GEO and ES-L1 have the highest success rate
- Launch success rate started to increase in 2013 till 2020
- KSLC-39A has the highest number of successful launches and the predominant success rate among all sites
- The decision tree classifier has proved to be the most successful model algorithm for this task
- The success rate of low weighted payloads is higher than heavy weighted payloads
- All models have the accuracy of 83.33%, though it would appear that more data is needed to determine the optimal model as a small data size was utilized

Appendix

- GITHUB URL
- Coursera Applied Data Science Cap Stone Course URL
- Visual Studio Code
- PyCharm CE

Thank you!