Winning Space Race with Data Science

<Sviat AJ Navrotskyi>

<07/14/2022>

Outlines

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

Executive Summary

- Project methodologies overview
- Summary of results

Project methodologies overview

- The intention of this project is to predict whether SpaceX's Falcon 9 will land successfully on first launch stage – ultimately saving millions in dollars and resources

- Data was obtained from SpaceX's API and web scraping. The data is normalized so that it can be transformed into a dataframe which enables analysts to clean missing data, format data into a more meaningful format, and transform & visualize it so it can be accurately conclusion could be drawn

- Once meaningful data are extracted, visualized, and analyzed; the data is trained so that predictive analysis can be performed and choose the best method with the highest-level of accuracy. A confusion matrix is generated to measure the accuracy & precision of the models

Summary of results

- Results are obtained by performing various models to predict landings, trans data to predict landing outcome and supporting the results by analyzing predicted model accuracy

- Several factors such as weight of the payload, launch site's location & proximity to natural environment, and performing multiple launches played a role to the improvement of the outcome of the landings

- Based on the models, we can accurately predict that the landings will be successful 83% of the time – meaning there is a high confidence that first stage will land back to Earth successfully without any issues. This may be improved overtime as there is an upward trend in the success rate.

Introduction

- Space exploration is a highly resource-intensive industry costing billions of dollars to build only to be dumped in space/orbit or back to Earth unsalvageable. This has direct economic and environmental impact

- SpaceX built Falcon 9 rocket launches at a cost of 62 to 165 million US dollars. The mission is to dock back to Earth without sustaining (significant) damage so that it can be reused for future explorations

- This project intends to predict whether Falcon 9 will land successfully on its first stage – ultimately saving millions in dollars and resources. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch

Section 1

Methodology

Methodology

- Data is collected in two ways: A)API calls from SpaceX B) Web scraping from wikipedia. Data wrangling is performed by A) analyzing the columns and checking their data types B) identifying fields with null/missing values and C) exploring data patters in terms of unique launch sites and calculating the outcome occurrence per mission in each orbit
- Data is also analyzed using SQL to explore distinct launch sites, total payload mass carried by specific launchers, average payload carried by a specific booster version, date when first successful landing outcome, and landing outcome given a particular year. There queries were performed to check if there are factors that may affect a successful landing
- Once data is explored and ready for analysis, it is transformed into various forms of visualizations. The first visualization is through Folium map which allows the analysts to see the locations of the launch sites and how these locations were selected based on proximities to natural and man-made structures. Markers were added to access the landing outcome in each site
- Data is also visualized using Plotly Dash that enables interactive presentation of all the launch sites. This includes percentage of successful outcomes per site and successful & failed boosters in each payload range per selected launch site
- Predictive analysis and evaluation using various models are performed to predict successful landings. Models include as Logistic Regression, Support Vector, Decision Tree, and K Nearest Neighbor. The best classification model is selected based on the highest level of accuracy. A confusion matrix is generated to validate and support the findings

Data Collection – API Call

- Data is collected by requesting data from API call from https://api.spacexdata.com/v4/launches/past
- Once connection to the API is successful, a response is sent back from SpaceX API. This contains launch data from SpaceX
- The data is normalized into a Panda dataframe which allows analysts to clean the data by removing/ replacing missing values and selecting relevant columns only

For more information see Appendix and Notebook:
https://github.com/DrNavrotskyi/SpaceY/blob/main/Data%20collection%20API.ipynb

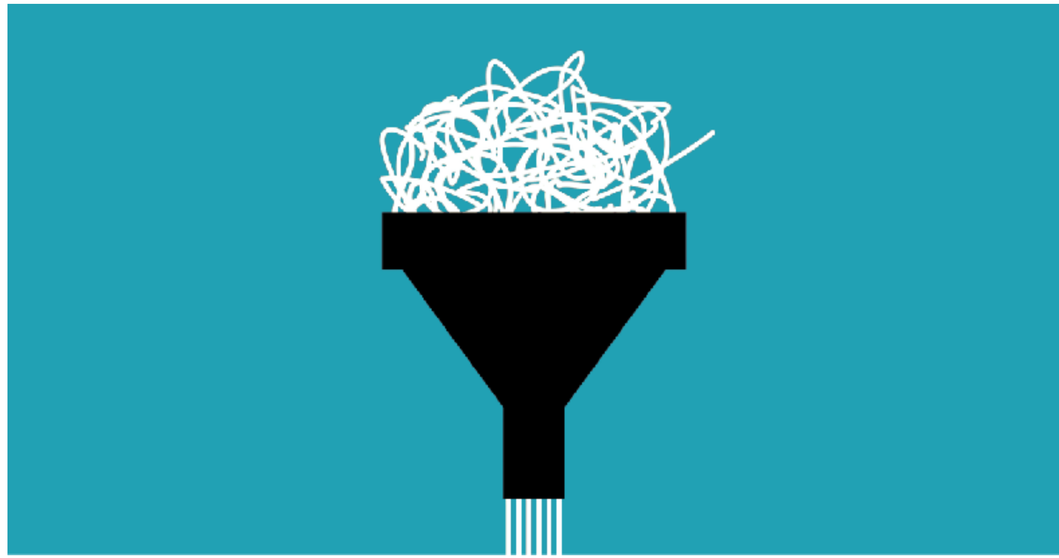
Data Collection – Web Scraping

- Data is also collected by scraping relevant HTML table from Wikipedia using BeautifulSoup. Once data is parsed, only relevant section(s) of the page are taken so that it can be converted into a Panda dataframe which can be used for analysis.

For more information see Appendix and Notebook:
https://github.com/DrNavrotskyi/SpaceY/blob/main/Webscraping.ipynb

Data Wrangling

- Exploratory Data Analysis is performed to find data patterns to determine the label for training supervised models.
- Analysts counted for the number of launches per specific launch sites as various flights with different booster versions were launched from different sites. The number of occurrences of mission outcome per orbit type is also determined.
- An outcome label is created from Outcome column to distinguish between good and bad outcomes.
- This resulted to **0.66 success rate**



- For more information see Notebook:

https://github.com/DrNavrotskyi/SpaceY/blob/main/Data%20wrangling.ipynb

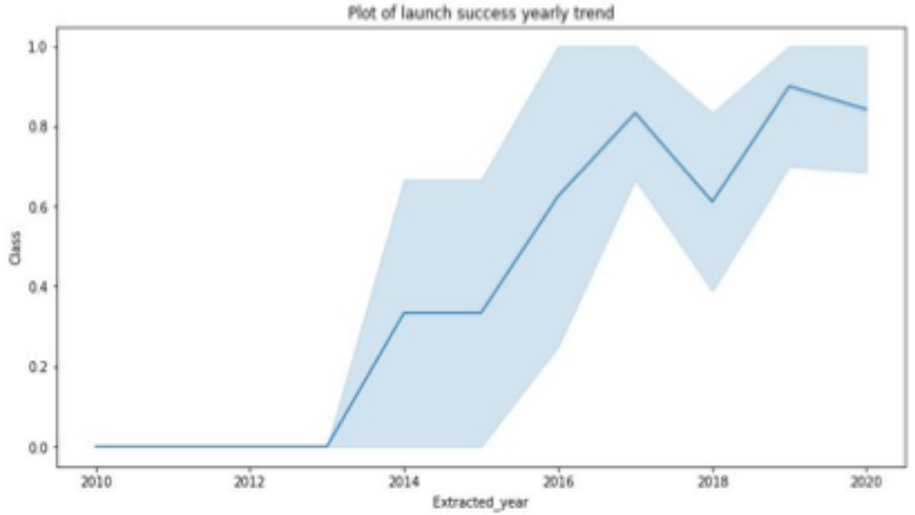
EDA with Data Visualization

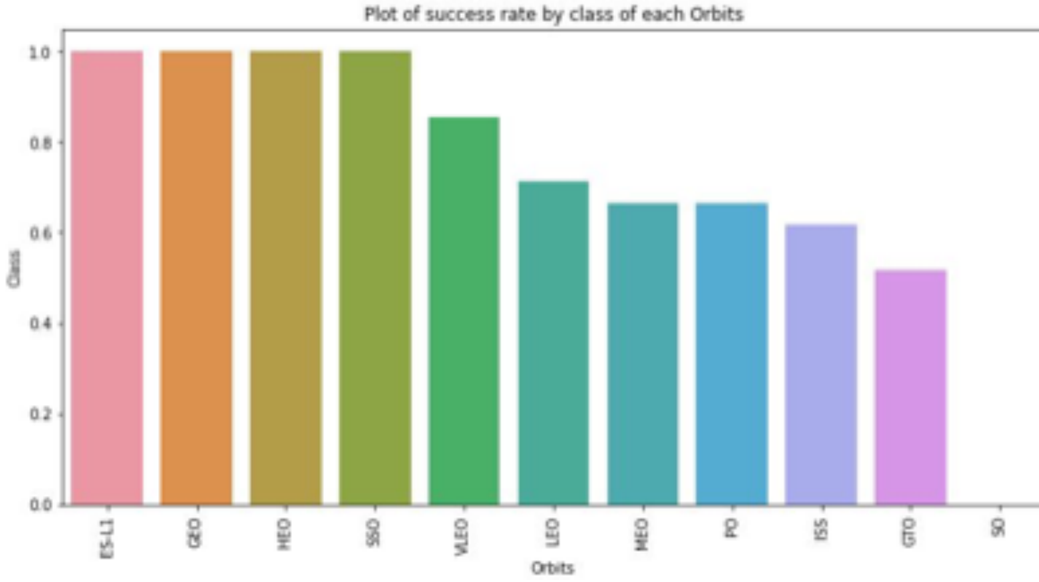
- Various visualization tools are created to analyze the success of landings given various variables such as payload, launch site, and orbit type

- Visualizations include relationship between Flight number & launch site, relationship between payload & launch site, relationship between success rate of each orbit type, relationship between flight number and orbit type, relationship between payload & orbit type, and launch success yearly trenc

- These visualizations help identify relationship between two significant variables that may affect the success of landing e.g. the higher the payload, the less likely first stage will return

- For more information see Notebook:





https://github.com/DrNavrotskyi/SpaceY/blob/main/EDA-DATA_%20visualization.ipynb

EDA with SQL

- Information during exploratory data analysis using SQL can be used if an alternate company wants to bid against SpaceX for a rocket launch

- Queries include:
  - Selection of unique launch sites / Selection of launch sites that starts with “CCA”
  - Return the payload mass carried by boosters launched by NASA (CRS)
  - Return average payload carried by F9 v1.1
  - List total number of successful and failure mission outcomes

- With all these information in mind, we are able to understand SpaceX dataset and validate important variables that may affect successful landing of first stage

- For more information see Notebook:

https://github.com/DrNavrotskyi/SpaceY/blob/main/EDA_SQL.ipynb

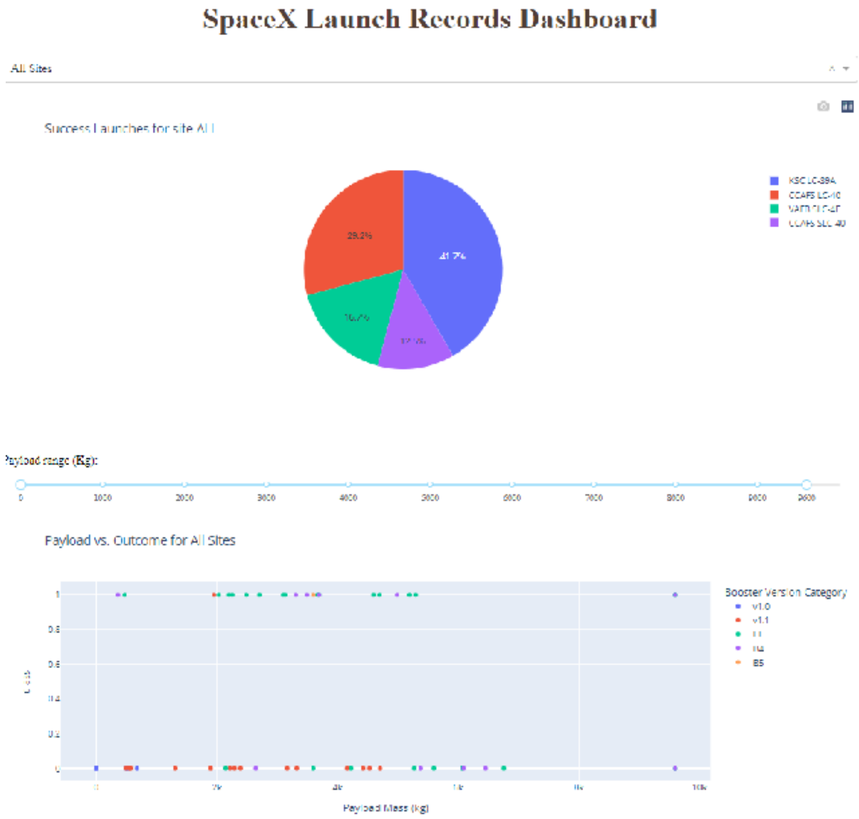
Build an Interactive Map with Folium

- Launch success rate may depend on location and proximities of launch site so it is paramount to find the best location by visualizing launch sites through Folium maps
- Launch sites are identified in the map through markers & circles. The circles are placed on the map by taking the location coordinates (latitude & longitude) of each site and giving a color and radius so that it would be visible in the map
- To identify circle's identity, a marker is added by taking the name of the launch site
- Colored markers are also created to help visually distinguish between successful and failed launch at a given site
- Polylines are created to determine the proximity between railways and coastline
- For more information see Notebook:

https://github.com/DrNavrotskyi/SpaceY/blob/main/Interactive_Visual_Analytic_with_Folium.ipynb

Build a Dashboard with Plotly Dash

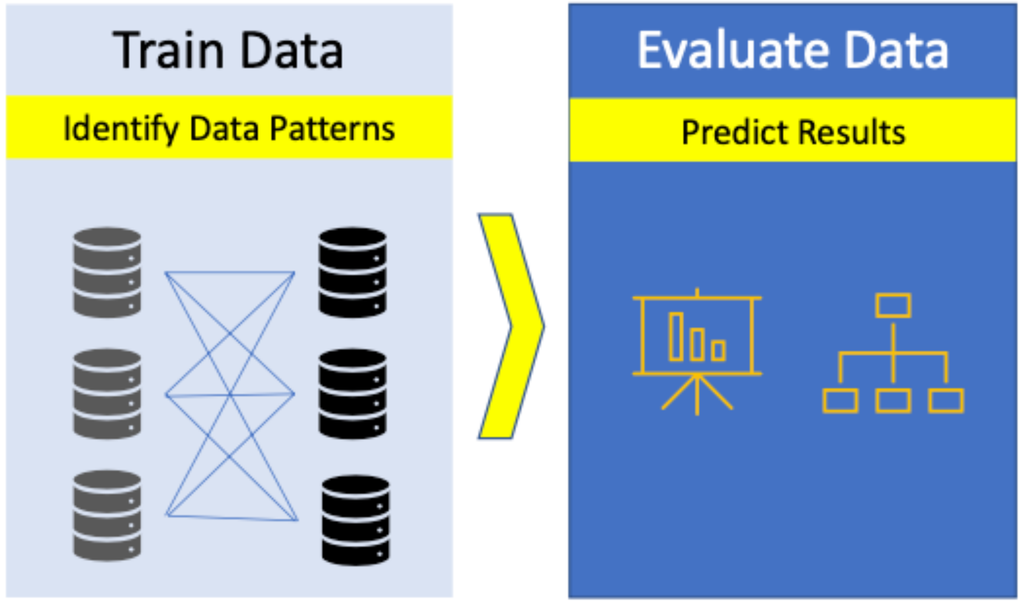
- Interactive visual analytics help user visualize SpaceX launch data in real-time
- Interactions include dropdown list of unique Launch Sites and a range slider to manage the payload range (KG) per given launch site
- Each interaction updates related pie chart and scatter plot chart



- For more information see Notebook:

https://github.com/DrNavrotskyi/SpaceY/blob/main/Dashboard%20Application%20with%20Plotly%20Dash.ipynb

Predictive Analysis (Classification)



- Success of first launch stage is a focal point of this project. Therefore machine learning pipeline is created to predict outcome of first stage
- To perform predictive analysis, previously cleaned and transformed data is used and normalized. Training data & test data are assigned with parameter test size set to 0.2 and random state set to 2
- Various data trainings are performed in order to identify solution with the highest level of accuracy. This involves Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K Nearest Neighbours. Each method is calculated for score to determine accuracy
- The best method with highest level of accuracy at **0.93** is Decision Tree Classifier
- For more information see Notebook:

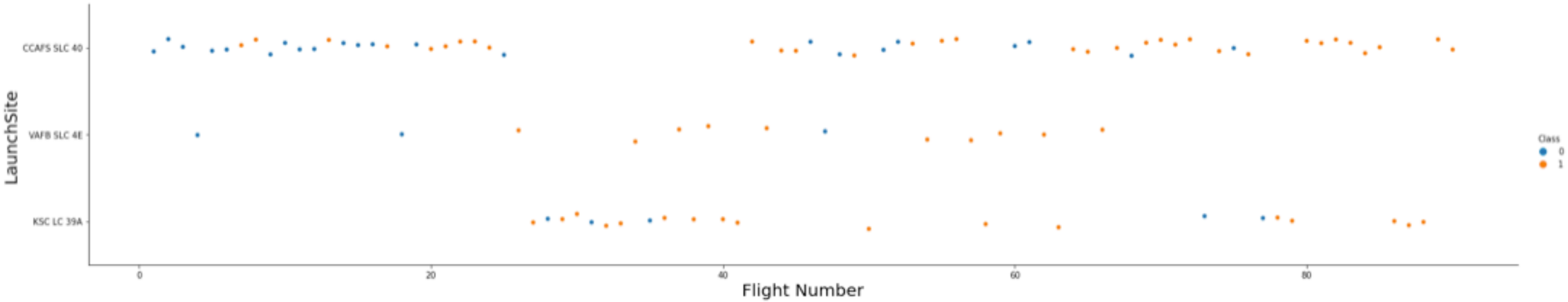
https://github.com/DrNavrotskyi/SpaceY/blob/main/Machine%20Learning.ipynb

Results

- Exploratory data analysis showed the most successful orbits and launch sites
- Interactive map was build to depict launch sites
- Predictive model was build to predict launch success with 93% accuracy

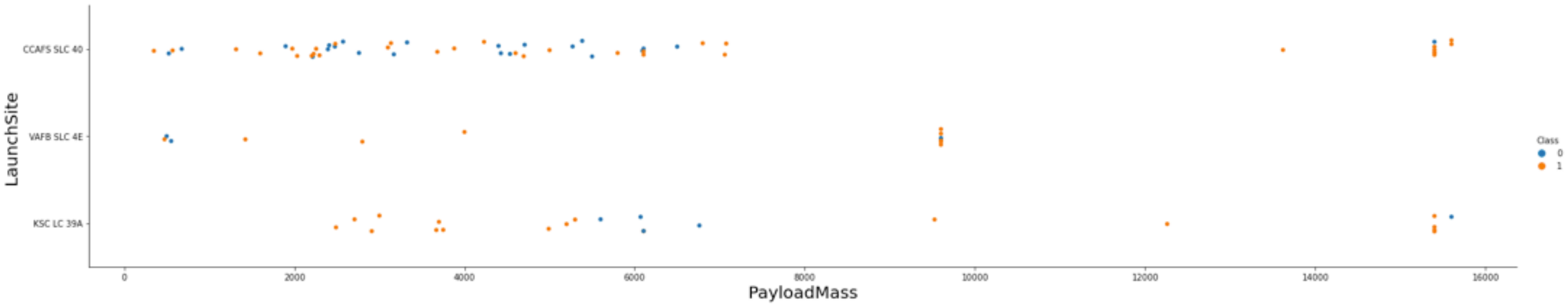
Section 2

Insights drawn from EDA



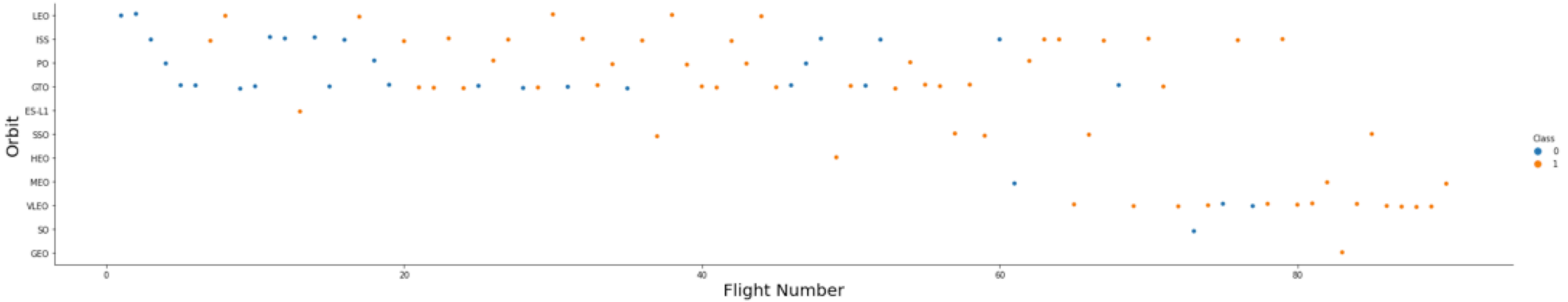
Flight Number vs Launch Site

- The graph shows that as more flights were launched, the landings become more successful. This tells us that initial landings are unsuccessful and as more improvements are done to fix the initial failures, the more successful the landings have become. CCAFS SLC 40 site has the most launches.



Payload vs. Launch Site

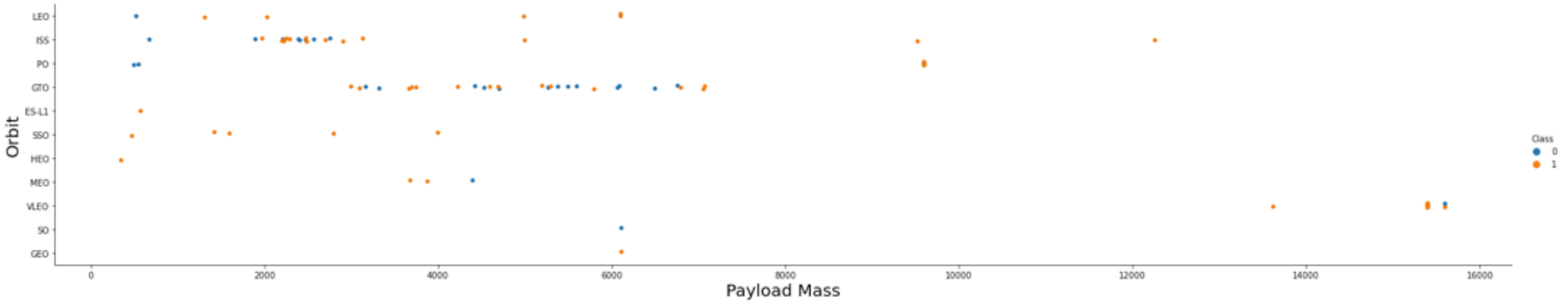
- The graph shows that the landings are likely to succeed with a lighter payload than a heavier payload. CCAFS SLC 40 has carried lighter payloads than other sites but has more unsuccessful landings too. On the other hand VAFB SLC 4E has carried out less flights with varying payloads but with more success



Flight Number vs Orbit Type

- The graph shows that there are more flights happen between Low Earth Orbit (LEO) and Geographical Transfer Orbit (GTO) than the orbits between ES-L1 and GEO (4 fails vs 22 successes). This means that not all flights make it to a higher altitude/orbit. However, higher orbit has positive impact to the outcome as there are less failed outcomes than in the lower orbit –due to less gravitational pull from the Earth

- Several failed outcomes happened during the initial flights, but success rate has immensely improved later



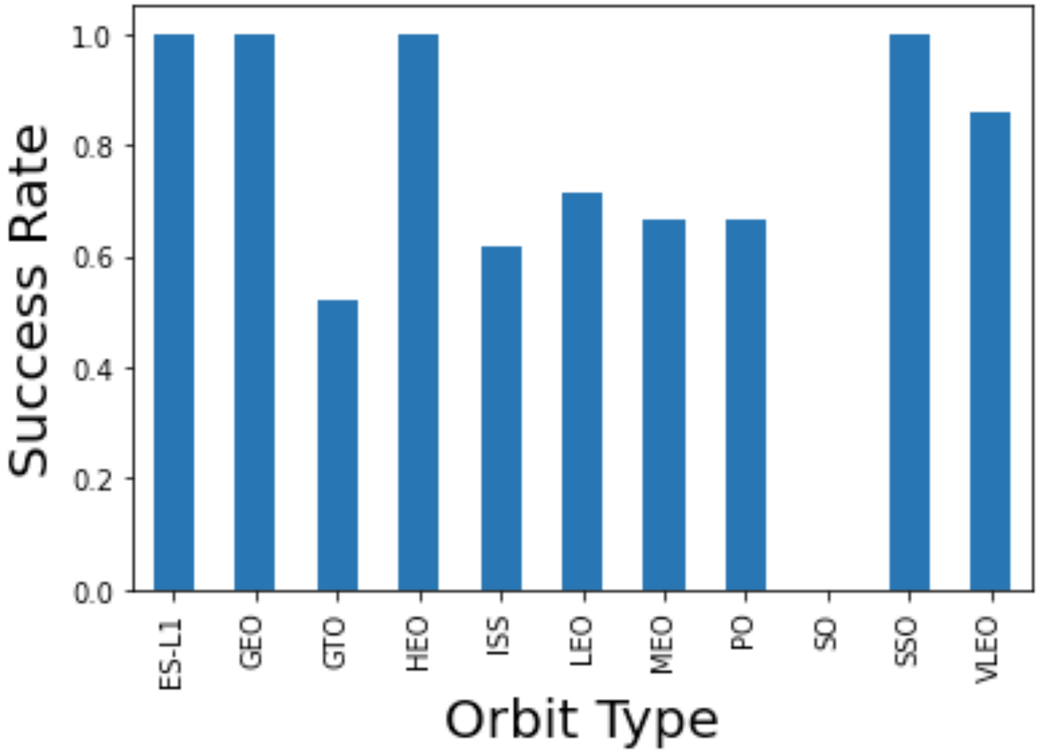
Payload vs Orbit Type

- Heavy payload negatively impacts VLEO & MEO orbits but positively impacts ISS and PO orbits. SSO has never had a bad up to 4,000 kg
- At 6000kg, GEO has a successful outcome when SO has failed
- The heaviest payload (>15,000kg) made it to VLEO with 50% success rate

Success Rate vs Orbit Type

- The graph shows that the orbits ES-L1, GEO, HEO, and SSO have consistently provided successful outcomes at 1.0 mean while SO never had a successful outcome at 0.0 mean

- 1.0 Success Rate = success, 0.0 Success Rate = fail

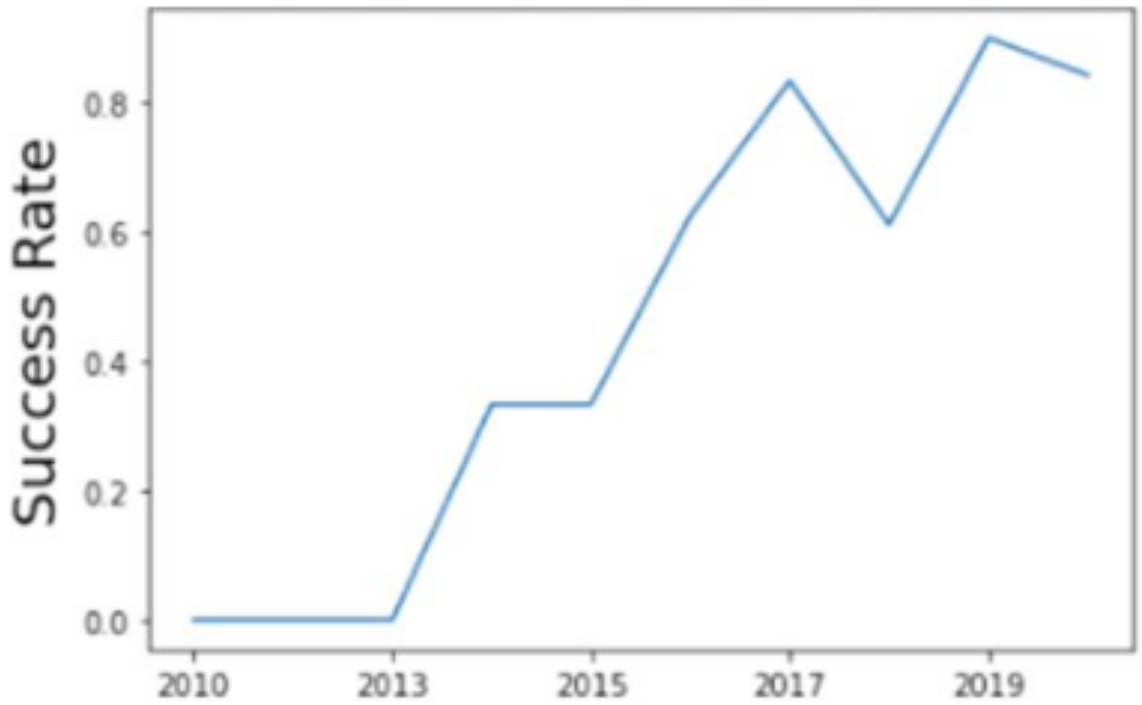


Launch Success Yearly Trend

- A yearly trend from 2010-2020 shows the mean success rate each year

- 2019 onwards have a promising outcome with a success rate >80%. This means that higher success rate is possible in the future

- Overall success rate since 2013 kept increasing till 2020



Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'

In [11]:
```
task_2 = '''
        SELECT *
        FROM SpaceX
        WHERE LaunchSite LIKE 'CCA%'
        LIMIT 5
        '''
create_pandas_df(task_2, database=conn)
```

Out[11]:

	date	time	boosterversion	launchsite	payload	payloadmasskg	orbit	customer	missionoutcome	landingoutcome
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- Operators **WHERE** and **LIKE** were used to display 5 records where launch sites begin with ‘CCA’

All Launch Site Names

- Operator **DISTINCT** was used to show only unique launch sites

Display the names of the unique launch sites in the space mission

In [10]:
```
task_1 = '''
        SELECT DISTINCT LaunchSite
        FROM SpaceX
'''
create_pandas_df(task_1, database=conn)
```

Out[10]:

	launchsite
0	KSC LC-39A
1	CCAFS LC-40
2	CCAFS SLC-40
3	VAFB SLC-4E

Total Payload Mass

- Operators **SUM**, **WHERE** and **LIKE** used to calculated the total payload carried by boosters from NASA (45,596 kg)

Display the total payload mass carried by boosters launched by NASA (CRS)

In [12]:
```
task_3 = '''
        SELECT SUM(PayloadMassKG) AS Total_PayloadMass
        FROM SpaceX
        WHERE Customer LIKE 'NASA (CRS)'
        '''
create_pandas_df(task_3, database=conn)
```

Out[12]:

	total_payloadmass
0	45596

Average Payload Mass by F9 v1.1

- Operators **AVG**, **AS** and **WHERE** used to calculate average payload mass carried by booster version F9 v1.1 (2,928.4 kg)

Display average payload mass carried by booster version F9 v1.1

In [13]:
```
task_4 = '''
        SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
        FROM SpaceX
        WHERE BoosterVersion = 'F9 v1.1'
        '''
create_pandas_df(task_4, database=conn)
```

Out[13]:

	avg_payloadmass
0	2928.4

First Successful Ground Landing Date

- Operators **MIN**, **AS, WHERE** and **LIKE** used to pull date of the first successful landing outcome on ground pad (22nd December 2015).

In [14]:

```
task_5 = '''
        SELECT MIN(Date) AS FirstSuccessfull_landing_date
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Success (ground pad)'
        '''
create_pandas_df(task_5, database=conn)
```

Out[14]:

	firstsuccessfull_landing_date
0	2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Operators **WHERE** used to filter for boosters which have successfully landed on drone ship and applied the **AND** condition to determine successful landing with payload mass greater than 4000 but less than 6000

In [15]:

```
task_6 = '''
        SELECT BoosterVersion
        FROM SpaceX
        WHERE LandingOutcome = 'Success (drone ship)'
            AND PayloadMassKG > 4000
            AND PayloadMassKG < 6000
        '''
create_pandas_df(task_6, database=conn)
```

Out[15]:

	boosterversion
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Operators **WHERE, LIKE** and **COUNT** used to count total number of success and failed outcomes

List the total number of successful and failure mission outcomes

In [16]:
```
task_7a = '''
        SELECT COUNT(MissionOutcome) AS SuccessOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Success%'
        '''

task_7b = '''
        SELECT COUNT(MissionOutcome) AS FailureOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Failure%'
        '''
print('The total number of successful mission outcome is:')
display(create_pandas_df(task_7a, database=conn))
print()
print('The total number of failed mission outcome is:')
create_pandas_df(task_7b, database=conn)
```

```
The total number of successful mission outcome is:
```

	successoutcome
0	100

```
The total number of failed mission outcome is:
```

Out[16]:

	failureoutcome
0	1

Boosters Carried Maximum Payload

- Operators **WHERE** and **MAX** with subquery were used to determine booster that have carried the maximum payload

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [17]:

```
task_8 = '''
        SELECT BoosterVersion, PayloadMassKG
        FROM SpaceX
        WHERE PayloadMassKG = (
                                SELECT MAX(PayloadMassKG)
                                FROM SpaceX
                                )
        ORDER BY BoosterVersion
        '''
create_pandas_df(task_8, database=conn)
```

Out[17]:

	boosterversion	payloadmasskg
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
5	F9 B5 B1051.3	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1051.6	15600
8	F9 B5 B1056.4	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1060.2	15600
11	F9 B5 B1060.3	15600

2015 Launch Records

- Operators **WHERE, LIKE**, **AND**, and **BETWEEN** used to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

In [18]:
```
task_9 = '''
        SELECT BoosterVersion, LaunchSite, LandingOutcome
        FROM SpaceX
        WHERE LandingOutcome LIKE 'Failure (drone ship)'
            AND Date BETWEEN '2015-01-01' AND '2015-12-31'
        '''
create_pandas_df(task_9, database=conn)
```

Out[18]:

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- Operators **COUNT, WHERE, BETWEEN, GROUP BY** and **ORDER** used to rank landing outcomes between 06-04-2010 and 03-20-2017

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad))

In [19]:
```
task_10 = '''
        SELECT LandingOutcome, COUNT(LandingOutcome)
        FROM SpaceX
        WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY LandingOutcome
        ORDER BY COUNT(LandingOutcome) DESC
        '''
create_pandas_df(task_10, database=conn)
```

Out[19]:

	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

Section 3

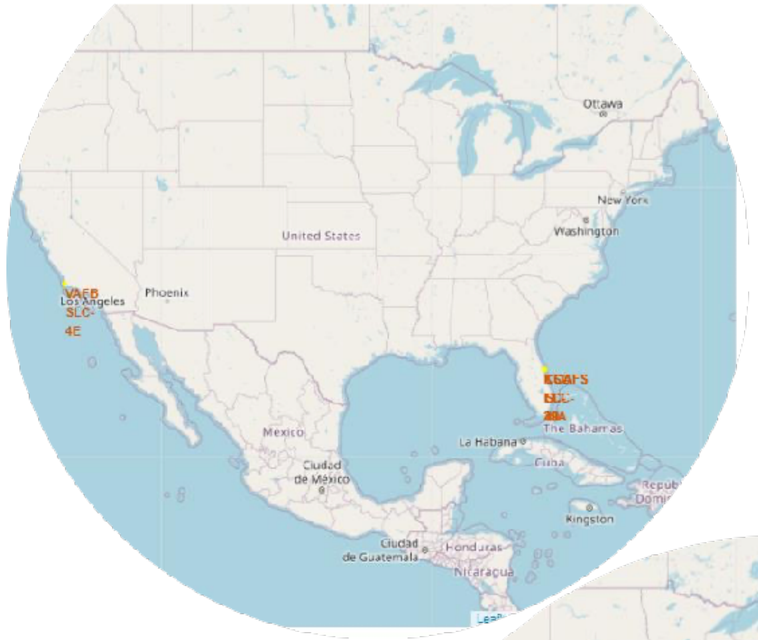
Launch Sites Proximities Analysis

Visualization of launch sites

- There are 4 unique launch sites that are marked in the Folium map. The three of them located on the east coast and one on the west coast of the United States

- As visualized, launch sites are near coastal areas (Pacific Ocean & Atlantic Ocean) to avoid any negative impact to the lives of the residents nearby

- Locations are successfully pinned to the map using Launch Site's coordinates – longitude and latitude



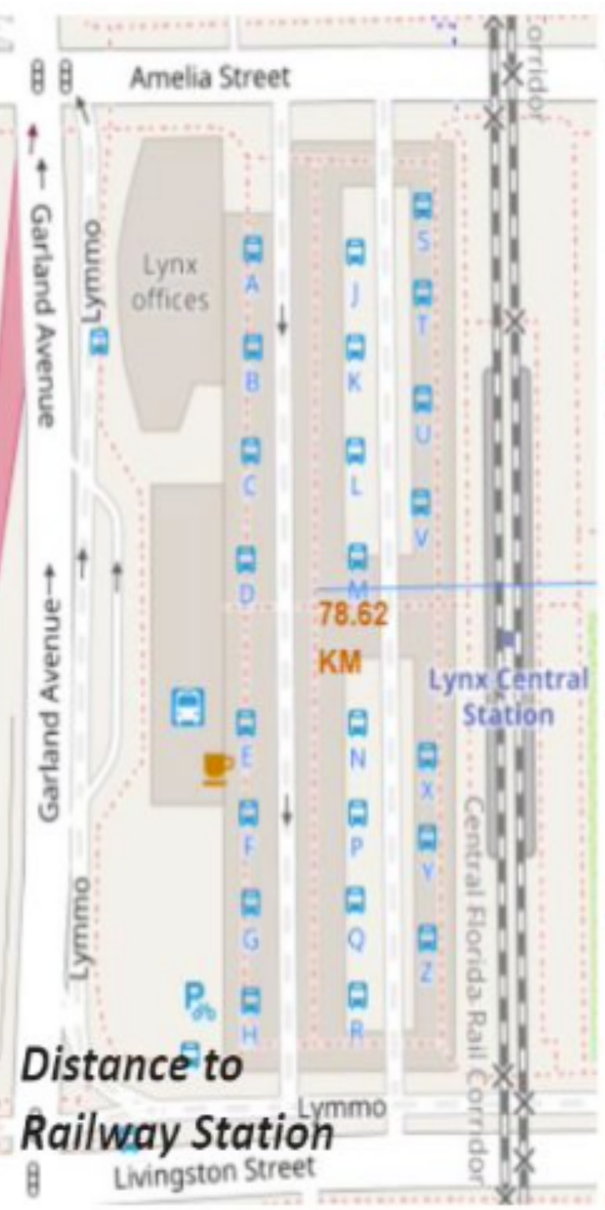
Visualizing Landing Outcomes of each Launch Site

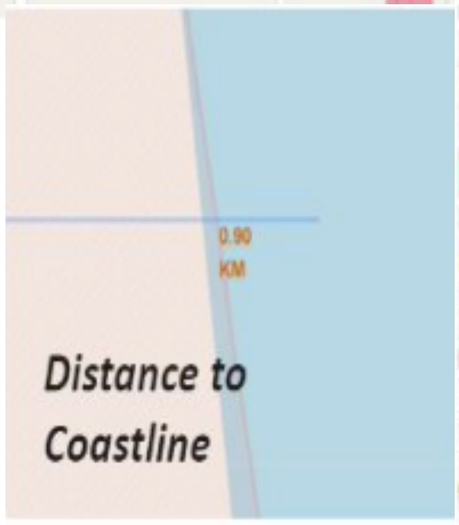
• It is important to visualize the landing outcome for each launch site as it helps analysts to see which launch site has a good track record in providing successful outcomes

• A colored marker is created for each site to determine the landing outcome – green for successful outcome and red for failed outcome. These markers are visible once the analyst zooms in and clicks each site

• The pictures provide quick visualization on how many successful/ failed outcomes there are in each site. As you can see there are more red markers than green (4/7) so it means that CCFS SLC-40 (first photo) has more failed outcomes than success. Launch Site CCFS LC-40 (second photo) also has more failed outcomes as visualized by red markers

Visualizing proximity of Launch Sites to Key Natural & Man-made elements





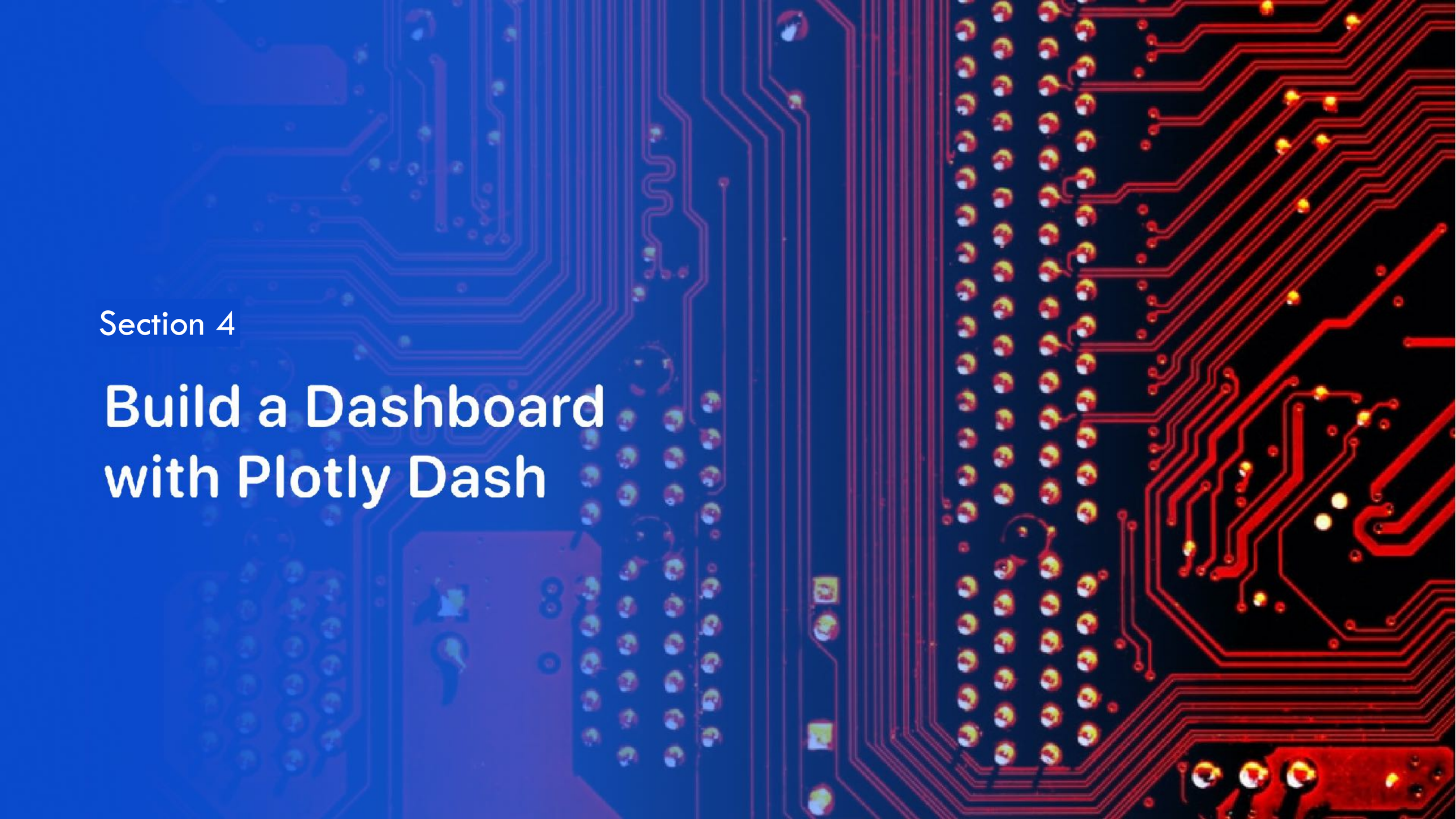
- Launch site CCAFS SLC-40 on East Coast was selected to visualize it proximity to nearest landmarks

Distance to coast – 0.9 km

Distance to city (Orlando) – 78.45 km

Distance to train station – 78.62

Distance to highway – 20.21 km

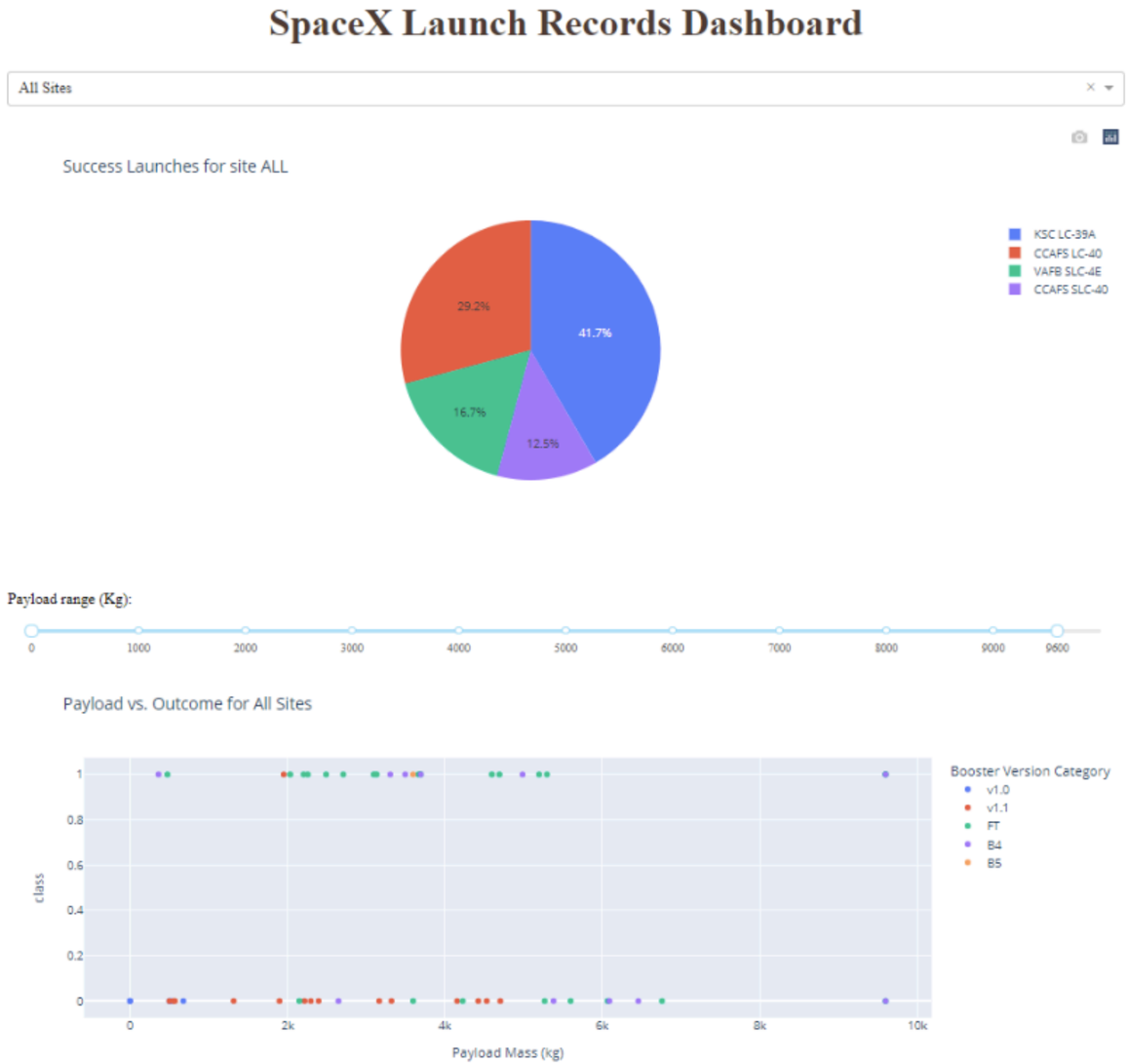

Section 4
Build a Dashboard with Plotly Dash

Interactive Dashboard: SpaceX Launch Records

- An interactive dashboard is created to show visual analytics in real-time which is created via Plotly Dash

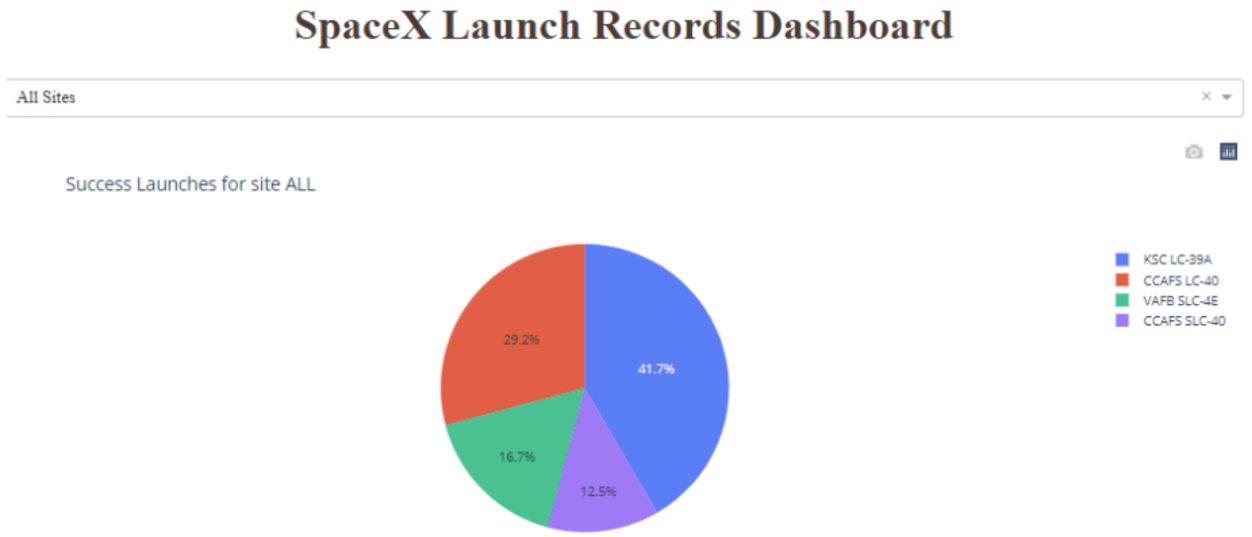
- Interactive functionalities include drop-down input component to select a specific or all launch sites, pie chart showing landing outcome once launch site(s) is chosen, range of payload, and a scatter plot that updates in real-time given a specific payload range and/or launch site

- These visual interactions are important to show quick visual reference on how payload and launch sites impacts the outcome of the landings

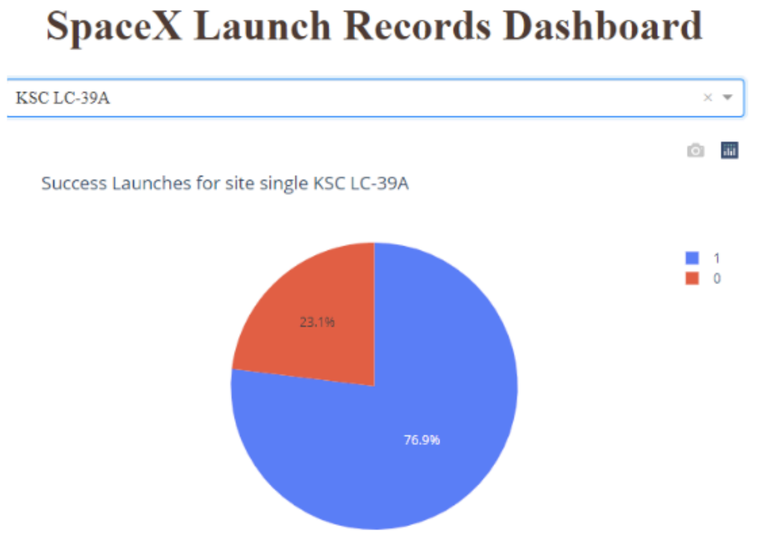


Interactive Dashboard: Pie chart based on Launch Site

• By default, “All Sites” are chosen. It shows all four launch sites and share of launches that happen within that site e.g. Most of the launches happened in KSC LC-39A at 41.7% while least launches happened in CCAFS SLC-40 at 12.5%

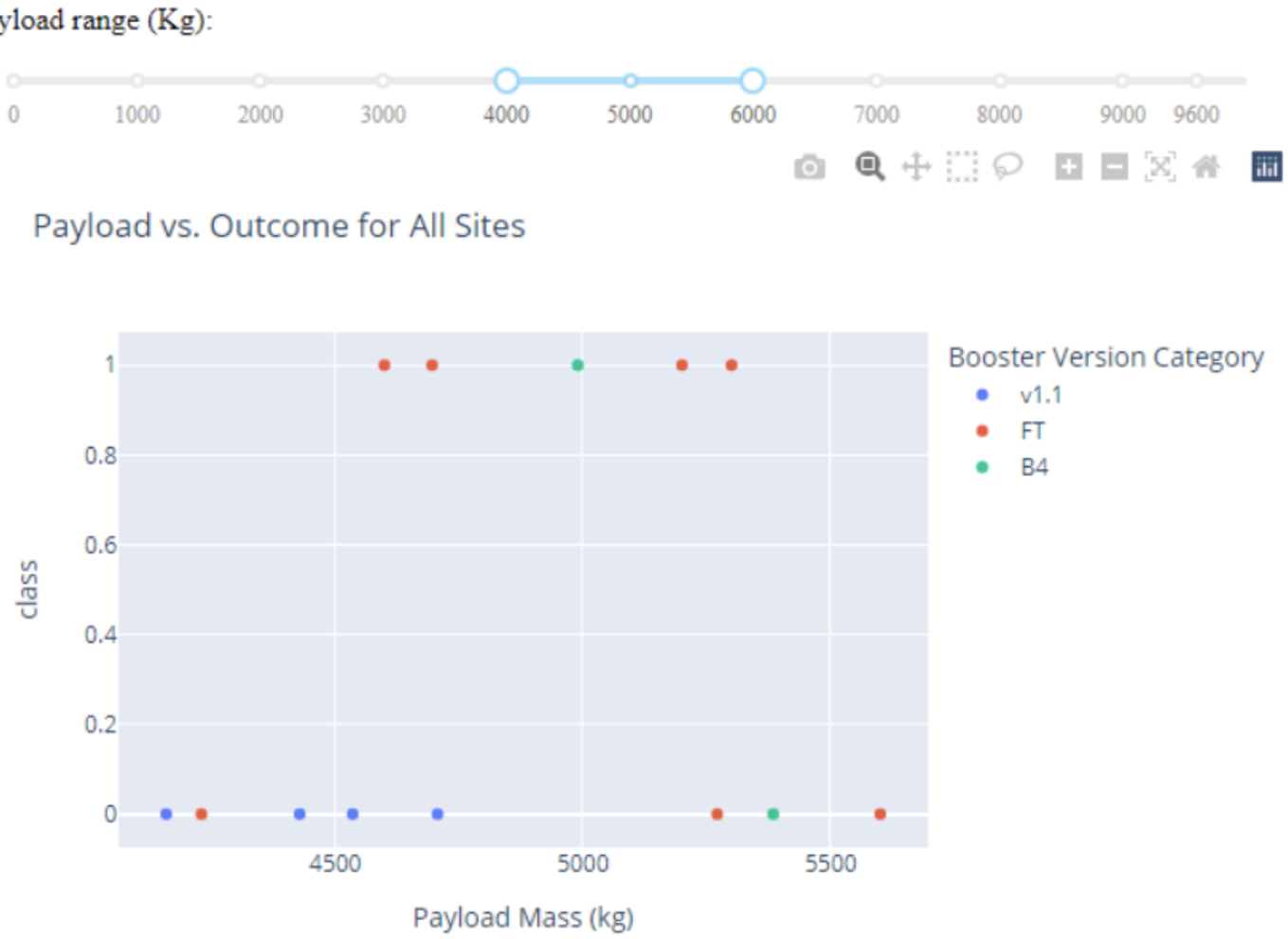


• It is possible to drill down to specific launch by selecting the launch site in the drop-down menu. For example KSC LC-39A site was selected in figure below. Graph shows the percentage of successful & failed launches (76.9% successful)



Interactive Dashboard: Scatter Plot based on Launch Site & Payload

- The most successful booster version category is FT, which is especially effective in a payload range 2-4 k
- All sites and the min/max payloads are chosen by default. Dashboard user may opt to change the payload range slider to see the scatter plot change which contains the landing outcome and related booster versions
- Snip shows that a payload 4,000 – 6,000 kg. Three boosters version present – it has 5 successful (outcomes class=1) and 8 failed (outcomes class=0). Booster version FT has 57% success rate, B4 has 50% success rate, and v1.1 has 0% for the selected payload range



Section 5

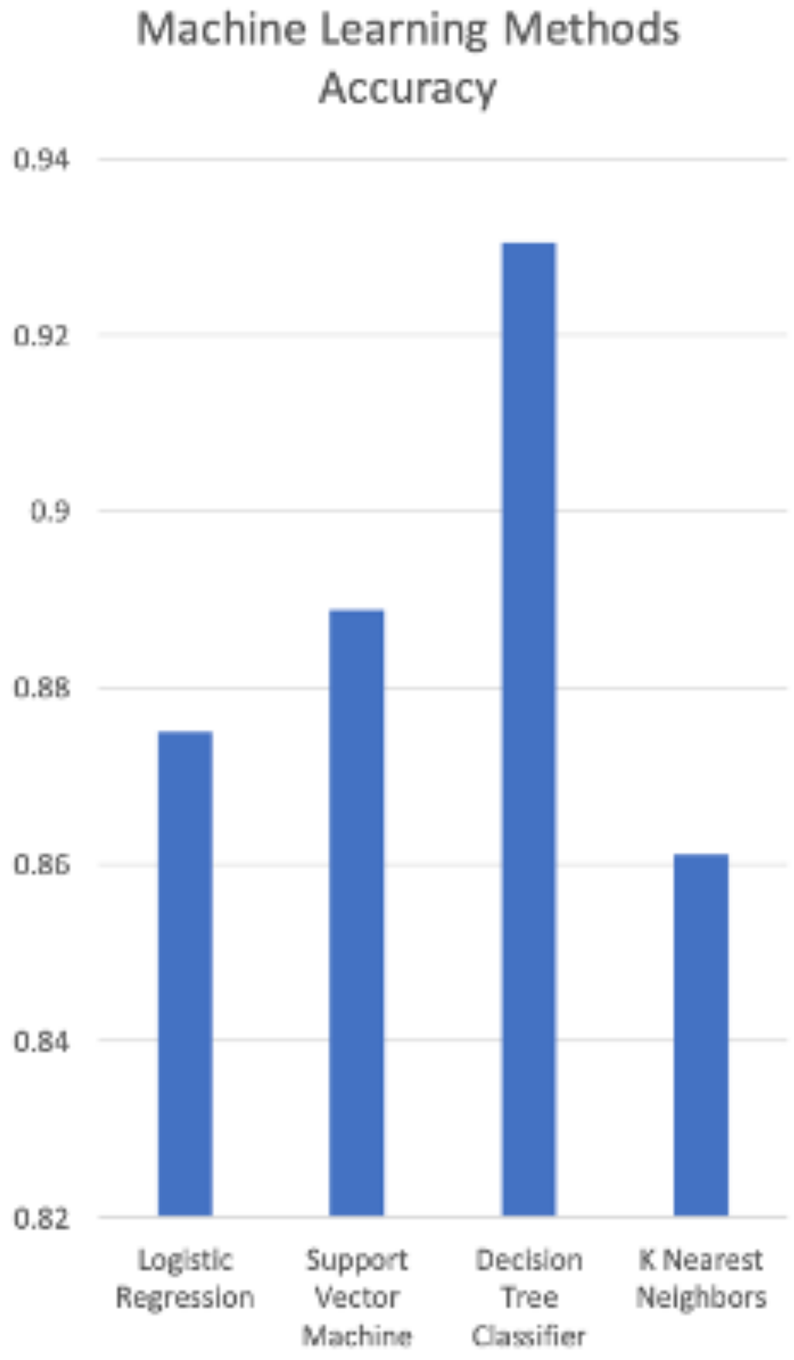
Predictive Analysis (Classification)

Classification Accuracy

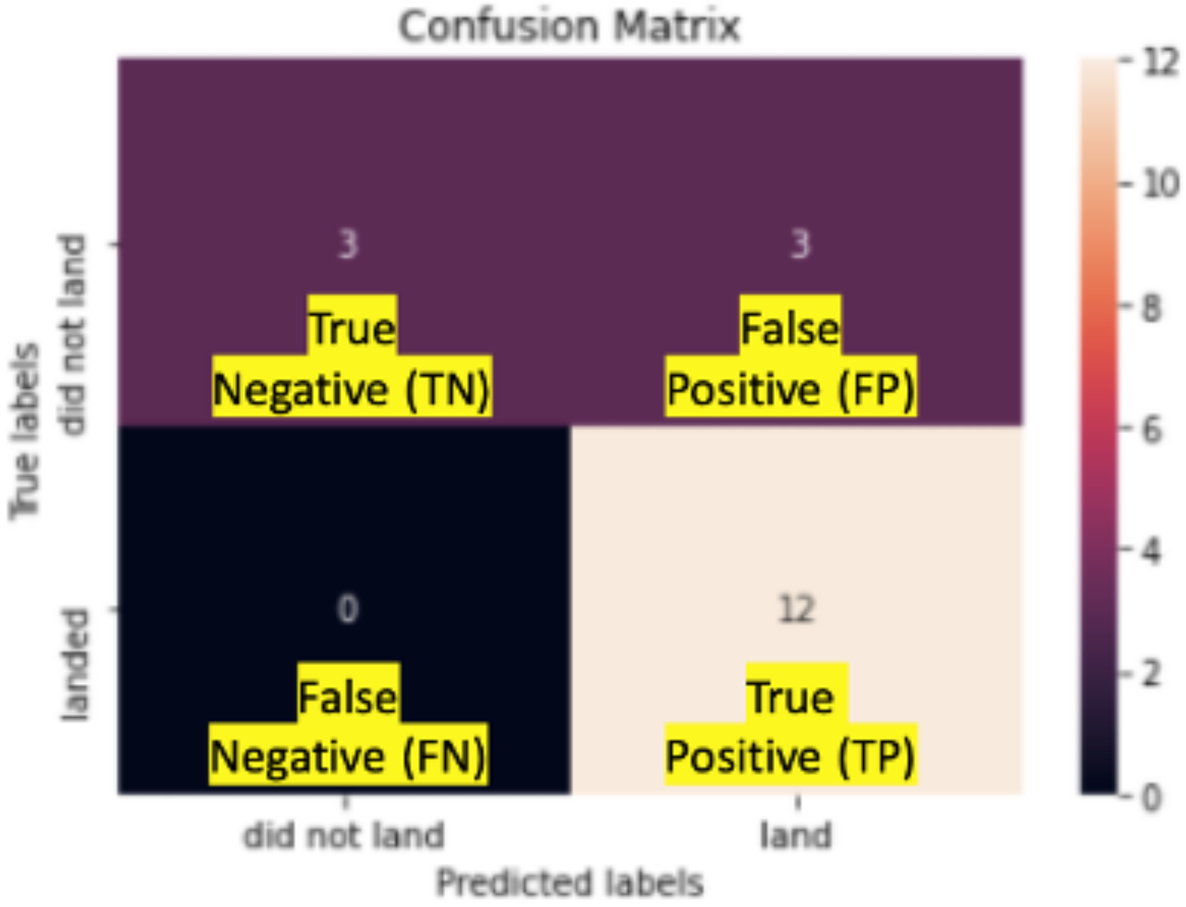
• Classification models were build to predict the outcome of first stage landings

•This is done by normalizing data and splitting it into training & test data, setting the parameters as test_size=0.2 and random_state=2, and performing various methods to take highest level of accuracy score. Once best algorithm is chosen, a confusion matrix is created to take the number of correct and incorrect predictions and measure accuracy, precision, recall, and prevalence

•The best algorithm with the highest accuracy is Decision Tree Classifier based on the method *score*

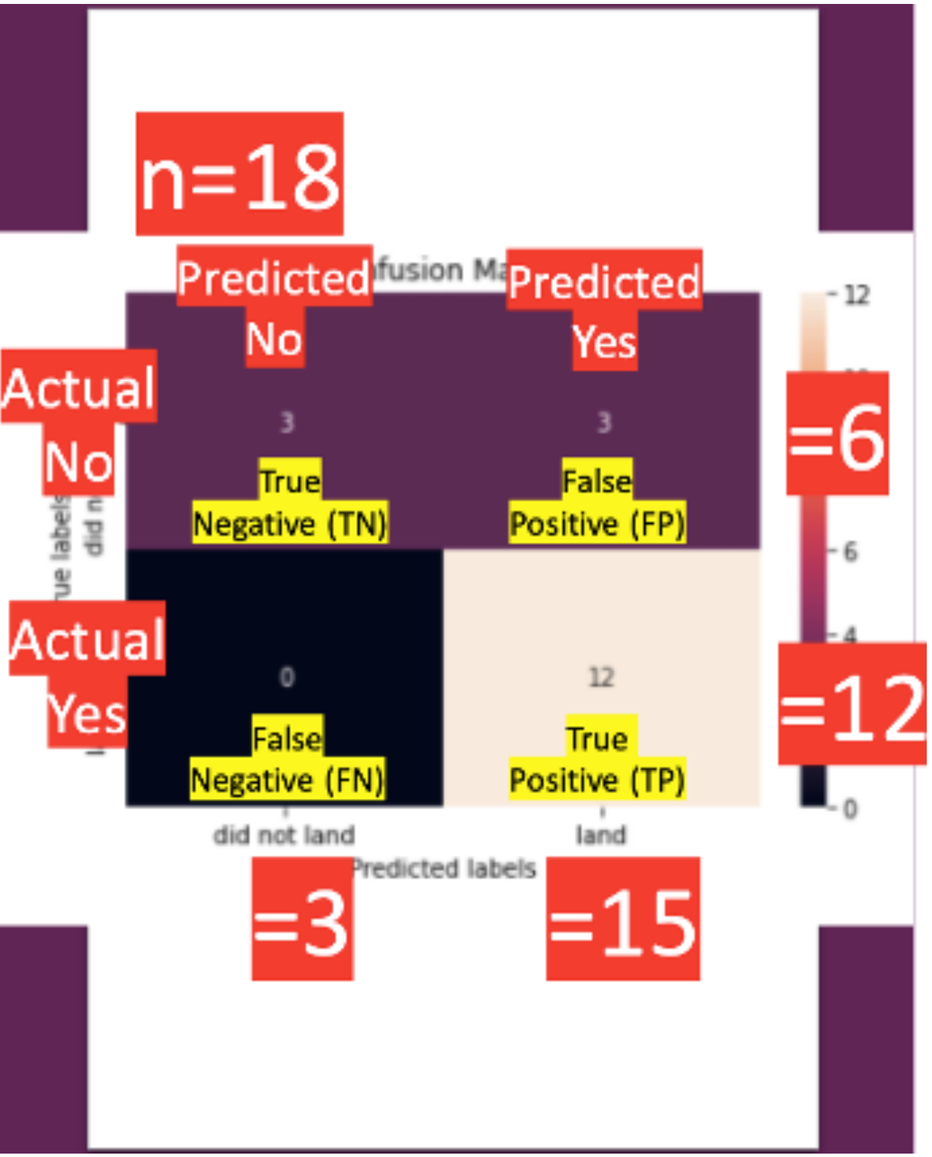


Confusion Matrix



- Decision Tree Classifier made 18 predictions at 0.93 accuracy score.
- Out of **18 predictions** identified, the classifier predicted 15 lands and 3 did-not-lands. In reality, 12 landed and 6 did not land.
- There are **3 TNs** – predicted not to land and did not land
- There are **3 FPs (Type 1 Err)** – predicted to land but did
- not land
- There is **0 FN (Type 2 Err)** – predicted not to land but landed
- There are **12 TPs** – predicted to land and landed

Confusion Matrix



• **Accuracy:** Overall, how often is the classifier correct? • (TP+TN)/total = (12+3)/18 = 0.833

•**Misclassification Rate:** Overall, how often is it wrong?

•(FP+FN)/total = (3+0)/18 = 0.167

•equivalent to 1 minus Accuracy

•also known as **"Error Rate"**

•**True Positive Rate:** When it's actually yes, how often does it predict yes?

•TP/actual yes = 12/12 = 1.00

•also known as "**Sensitivity" or "Recall"**

•**False Positive Rate:** When it's actually no, how often does it predict yes?

• FP/actual no = 3/6 = 0.50

•**True Negative Rate:** When it's actually no, how often does it predict no?

•TN/actual no = 3/6 = 0.50

•equivalent to 1 minus False Positive Rate

•also known as **"Specificity"**

•**Precision:** When it predicts yes, how often is it correct?

• TP/predicted yes = 12/15 = 0.80

• **Prevalence:** How often does the yes condition actually occur in our sample? • actual yes/total = 12/18 = 0.67

Conclusions

- Whilst Decision Tree model has the highest level of accuracy, all models show the same confusion matrix which gives us more confidence with the prediction models performed

- Based on the model accuracy, it can predict that the first stage will land successfully 83.3% of the time

- To test for the reliability of this model, we calculated precision and recall. Precision is 80% which tells us how many of the correctly predicted landings turned out to successfully land. Recall is 100% which tell us how many actual landings we were able to correctly predict with our model

- Therefore, it is concluded that SpaceX first stage landing will be successful based on 80% correct predictions on landings that turned out to successfully land; moreover, 100% of the actual landings were successfully predicted giving a high confidence with the chosen model

- With the success rate trend and improvement to technology, it is possible that landings can achieve >80% success rate – ultimately improving the utilization of resources and savings in the future of space exploration

APPENDIX

API CALL

1. Get request for rocket launch data using API

In [6]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

In [7]:
```
response = requests.get(spacex_url)
```

2. Use json_normalize method to convert json result to dataframe

In [12]:
```
# Use json_normalize method to convert the json result into a dataframe

# decode response content as json
static_json_df = res.json()
```

In [13]:
```
# apply json_normalize
data = pd.json_normalize(static_json_df)
```

3. We then performed data cleaning and filling in the missing values

In [30]:
```
rows = data_falcon9['PayloadMass'].values.tolist()[0]

df_rows = pd.DataFrame(rows)
df_rows = df_rows.replace(np.nan, PayloadMass)

data_falcon9['PayloadMass'][0] = df_rows.values
data_falcon9
```

Web Scrapping

1. Apply HTTP Get method to request the Falcon 9 rocket launch page

In [4]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

In [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
html_data = requests.get(static_url)
html_data.status_code
```

Out[5]:
```
200
```

2. Create a BeautifulSoup object from the HTML response

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html_data.text, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

In [7]:
```
# Use soup.title attribute
soup.title
```

Out[7]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

3. Extract all column names from the HTML table header

In [10]:
```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

element = soup.find_all('th')
for row in range(len(element)):
    try:
        name = extract_column_from_header(element[row])
        if (name is not None and len(name) > 0):
            column_names.append(name)
    except:
        pass
```

4. Create a dataframe by parsing the launch HTML tables
5. Export data to csv

APPENDIX

Notebooks

- Data Collection API: https://github.com/DrNavrotskyi/SpaceY/blob/main/Data%20collection%20API.ipynb
- Data Collection – Web Scraping: https://github.com/DrNavrotskyi/SpaceY/blob/main/Webscraping.ipynb
- Data Wrangling: https://github.com/DrNavrotskyi/SpaceY/blob/main/Data%20wrangling.ipynb
- EDA Data Visualization: https://github.com/DrNavrotskyi/SpaceY/blob/main/EDA-DATA_%20visualization.ipynb
- EDA with SQL: https://github.com/DrNavrotskyi/SpaceY/blob/main/EDA_SQL.ipynb
- Interactive Map: https://github.com/DrNavrotskyi/SpaceY/blob/main/Interactive_Visual_Analytic_with_Folium.ipynb
- Plotly Dashboard: https://github.com/DrNavrotskyi/SpaceY/blob/main/Dashboard%20Application%20with%20Plotly%20Dash.ipynb
- Predictive Analysis: https://github.com/DrNavrotskyi/SpaceY/blob/main/Machine%20Learning.ipynb

Thank you!