IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Aaron Resnick
3/1/22

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 results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

Introduction

- Project background and context

The age of commercial space is here and multiple companies are making space travel affordable for everyone. The most notable/successful of them is SpaceX, and one reason for this is that their rocket launches are relatively inexpensive.

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 state.

Therefore, we'll predict if the Falcon 9 first stage will land successfully. If we can determine if the first stage will land, we can determine the cost of a launch. This information could be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems you want to find answers

Correlations between each rocket variables and successful landing rate

Conditions to get the best results and ensure the top successful landing rate

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API & Web Scraping Falcon 9 and Falcon heavy Launches Records from Wikipedia
- Perform data wrangling
  - Convert outcomes into Training Labels with the booster successfully/unsuccessful 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 best Hyperparameter for SVM, Classification Trees and Logistic Regression

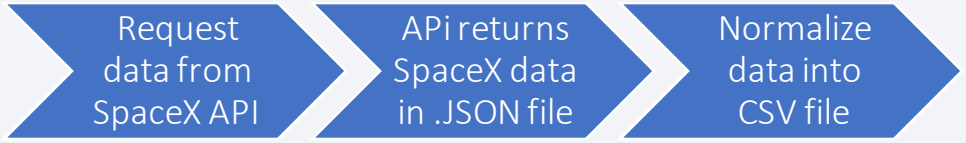
Data Collection

- The data collection process includes a combination of API requests from the SpaceX API and web scraping data from a table in the Wikipedia page of SpaceX, Falcon 9 and Falcon Heavy Launches Records.

  - SpaceX API Data Columns: `FlightNumber, Date, BoosterVision, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude`

  - Wikipedia Web Scrape Data Columns: `Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time`

SpaceX API



Web Scraping

Data Collection – SpaceX API

1. Requesting rocket launch data from SpaceX API

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

response = requests.get(spacex_url)
```

2. Converting response to a JSON file

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

3. Using custom functions to clean data

```
# Call getBoosterVersion
getBoosterVersion(data)
```

```
# Call getLaunchSite
getLaunchSite(data)

# Call getPayloadData
getPayloadData(data)

# Call getCoreData
getCoreData(data)
```

4. Combining the columns into a dictionary to create data frame

```
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)
```

5. Filtering dataframe and exporting to a CSV

```
data_falcon9 = launch_df[launch_df['BoosterVersion']=="Falcon 9"]

data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```

GitHub URL

Data Collection - Scraping

1. Getting response from HTML

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

2. Creating a BeautifulSoup Object

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

3. Finding all tables and assigning the result to a list

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

4. Extracting column name one by one

```
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)
```

GitHub URL

5. Creating an empty dictionary with keys

```
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']=[]
```

6. Filling up the launch_dict with launch records

(To long to insert, please refer to GitHub link)

7. Creating a dataframe and exporting it to a CSV

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

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

Data Wrangling

- There are several cases where the booster didn't land successfully in the dataset. Sometimes it attempted to land 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 result has not successfully landed in a specific area of the ocean
  - True RTLS: the mission result successfully landed on the ground pad
  - False RTLS: the mission result has not successfully landed on the ground pad
  - True ASDS: the mission result has successfully landed on the drone ship
  - False ASDS: the mission result has not landed on the drone ship
- Converting these results into training labels:
  - 1 = successful/0 = failure

GitHub URL

Data Wrangling

1. Calculating the number of launches at each site

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

2. Calculating the number and occurrence of each orbit

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

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

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

4. Creating a landing outcome label from Outcome column

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

5. Calculating the success rate for every landing in dataset

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

```
0.6666666666666666
```

6. Exporting dataset to a CSV

```
df.to_csv("dataset_part\_2.csv", index=False)
```

GitHub URL

EDA with Data Visualization

- Scatter Chart:
  - Flight Number vs. Launch Site
  - PayLoad vs. Launch Site
  - Flight Number vs. Orbit Type
  - PayLoad vs. Orbit Type
  - Scatter plot shows how much one variable is affected by another. A relationship between two variables is called a correlation. These plots are generally composed of large data bodies.
- Bar Chart:
  - Orbit Type vs Success Rate
  - A Bar chart is an easy way to compare datasets between multiple groups at a glance. One axis represents a category and the other represents a discrete value. Purpose of this chart is to indicate the relationship between the two axes.
- Line Chart:
  - Year vs. Success Rate
  - A Line chart very clearly shows data variables and trends while helping predict the results of data that hasn't been recorded.

- GitHub URL

EDA with SQL

- After loading the dataset into a corresponding table in a Db2 database, we answered the following questions by executing SQL queries:
  - Displaying the names of the unique launch sites in the space mission
  - Displaying 5 records 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
  - Listing the date when the first successful landing outcome in ground pad was achieved
  - Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - Listing the total number of successful and failure mission outcomes
  - Listing the names of the booster_versions which have carried the maximum payload mass
  - Listing the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - Ranking 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

- 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 success/failed launches for each site on the map
- Lines that show the distances between a launch site to its proximities
- By adding these objects, following geographical patterns about launch sites are found:
- Are launch sites in close proximity to railways? *Yes*
- Are launch sites in close proximity to highways? *Yes*
- Are launch sites in close proximity to coastline? *Yes*
- Do launch sites keep certain distance away from cities? *Yes*

- GitHub URL/ IBM Cloud URL (for interactive maps)

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 can be selected to indicate a successful landing distribution across all launch sites or to indicate the success rate of individual launch sites

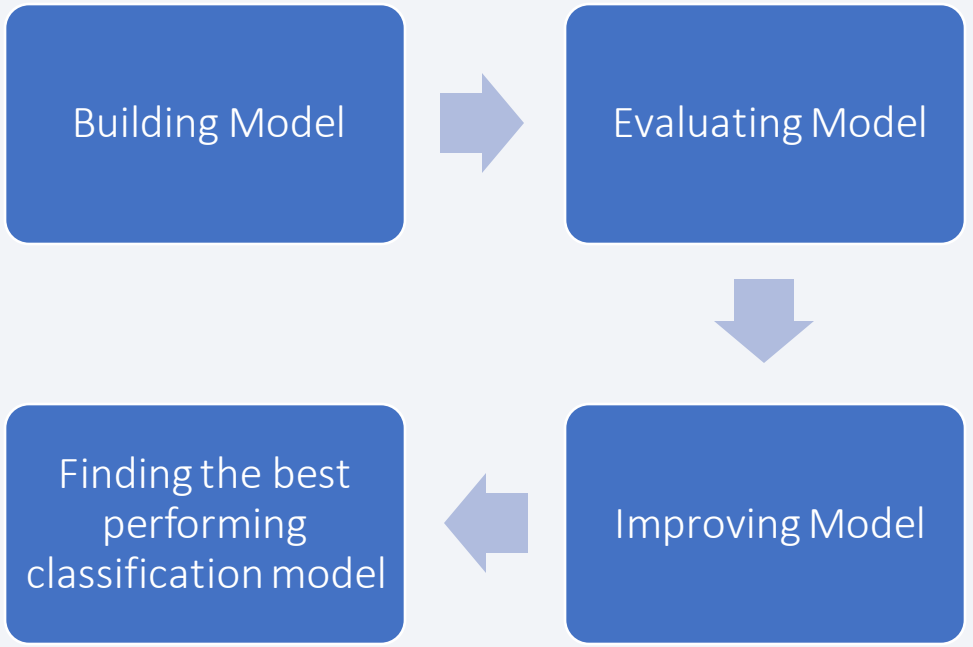
  - Scatter chart

    - For showing the relationship between Outcomes and Payload mass (Kg) by different boosters
    - Has 2 inputs: All sites/individual site & Payload mass on a slider between 0 and 10000 kg
    - This chart helps determine how success depends on the launch point, payload mass, and booster version categories.

- GitHub URL (code only)

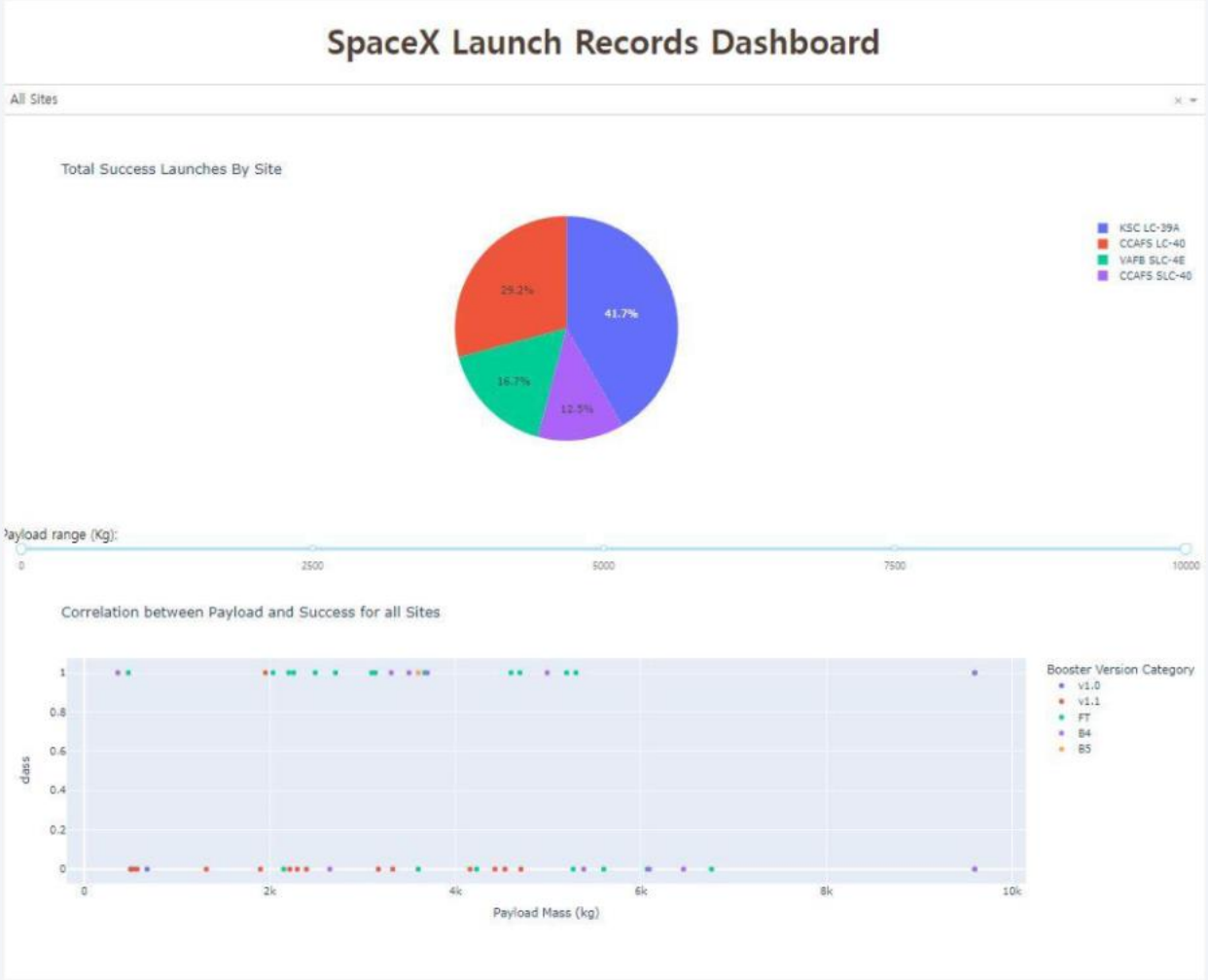
Predictive Analysis (Classification)

- Perform exploratory Data Analysis and determine Training Labels
  - Create a column for the class
  - Standardize the data
  - Split into training data and test data
- Find best Hyperparameter for SVM, Classification Trees and Logistic Regression
  - Find the method performs best using test data
- GitHub URL



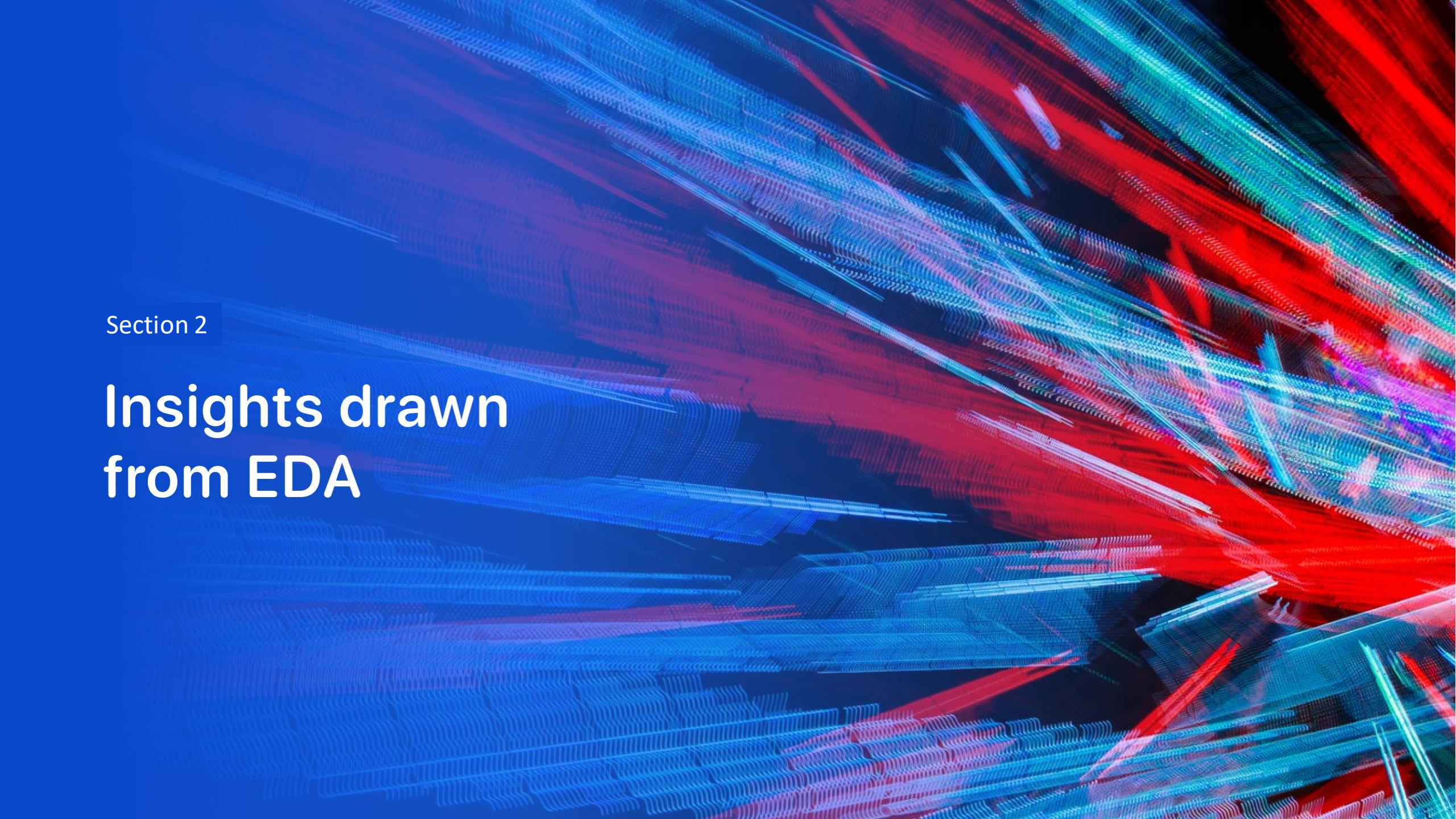
Results

- The screenshot shown is a preview of the Dashboard with Plotly Dash
- The results of EDA with visualization, EDA with SQL, Interactive Map with Folium and Interactive Dashboard will be show in upcoming slides
- Comparing the accuracy of the four methods, all return the same accuracy of about 83% for test data



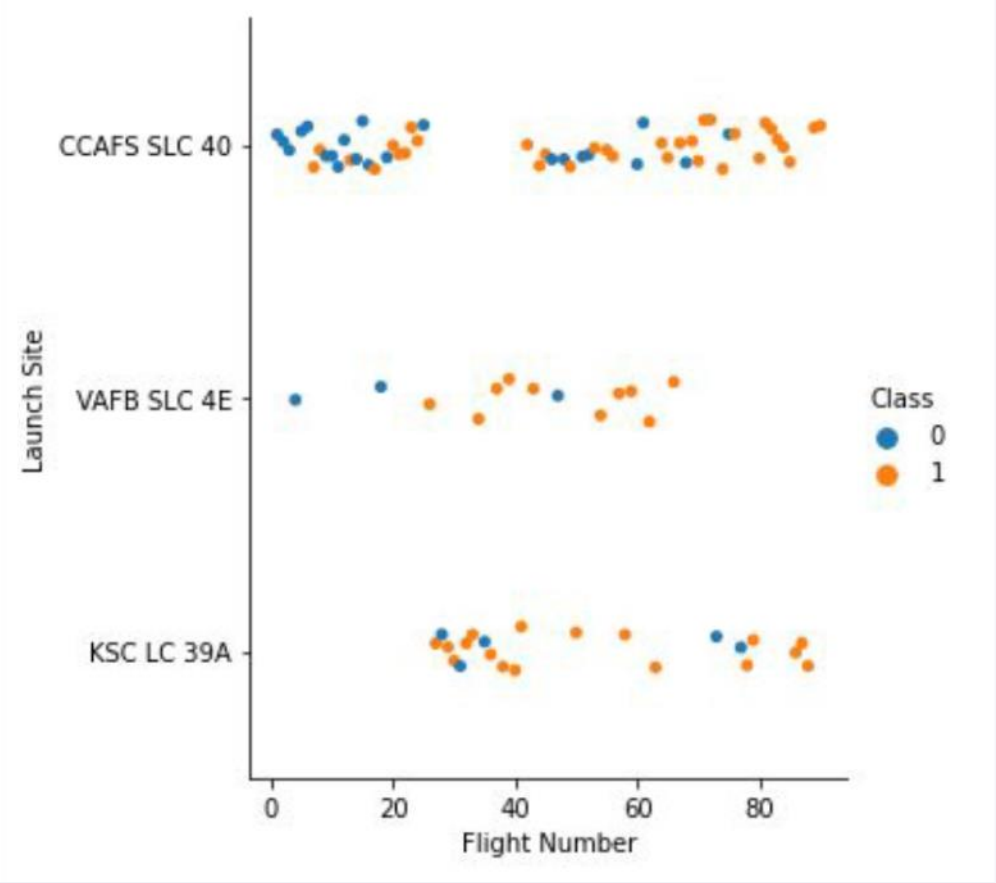
Section 2

Insights drawn from EDA



Flight Number vs. Launch Site

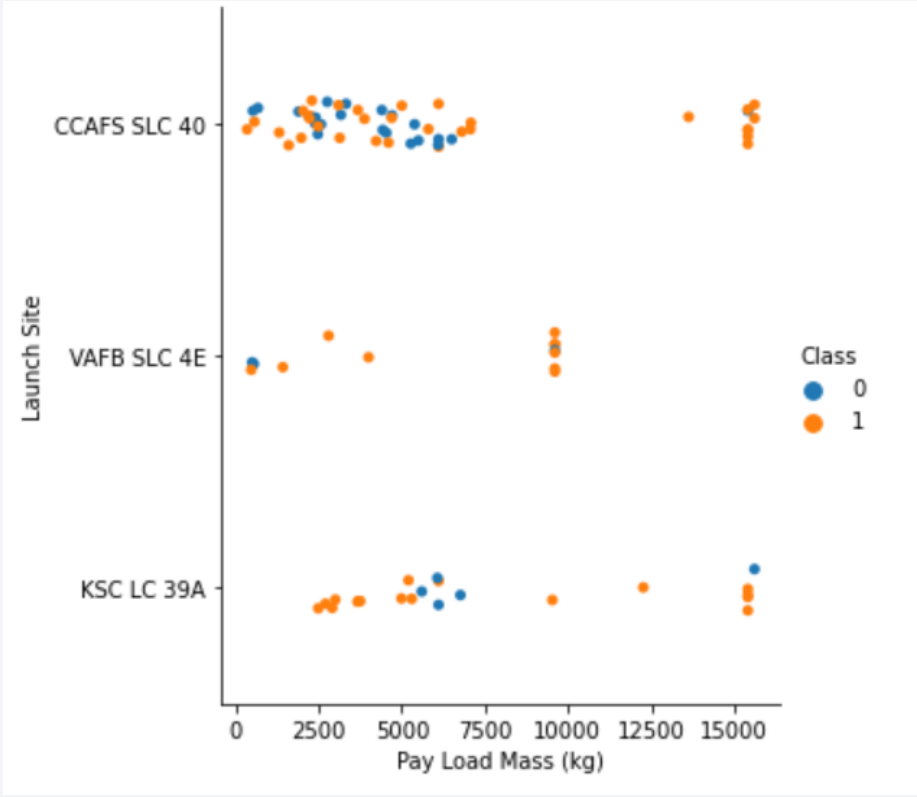
- Class 0 (blue) represents unsuccessful launch, and Class 1 (orange) represents successful launch
- This figure shows that **the success rate increased as the number of flights increased**
- This appears to be a breakthrough as the success rate has increased considerably since the 20th flight



Payload vs. Launch Site

- Class 0 (blue) represents unsuccessful launch, and Class 1 (orange) represents successful launch

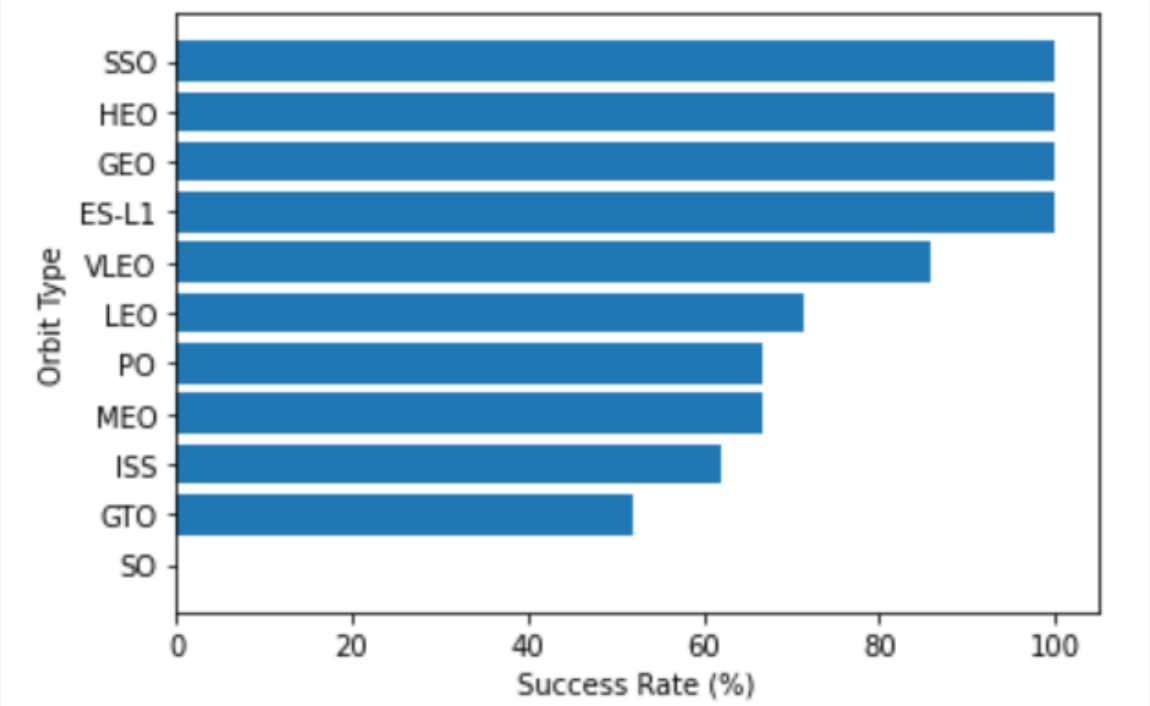
- It initially appears the larger the pay load mass, the higher the rocket's success rate. However, it seems difficult to use that to make decisions. That's because **no clear pattern can be found between successful launch and Pay Load Mass**.



Success Rate vs. Orbit Type

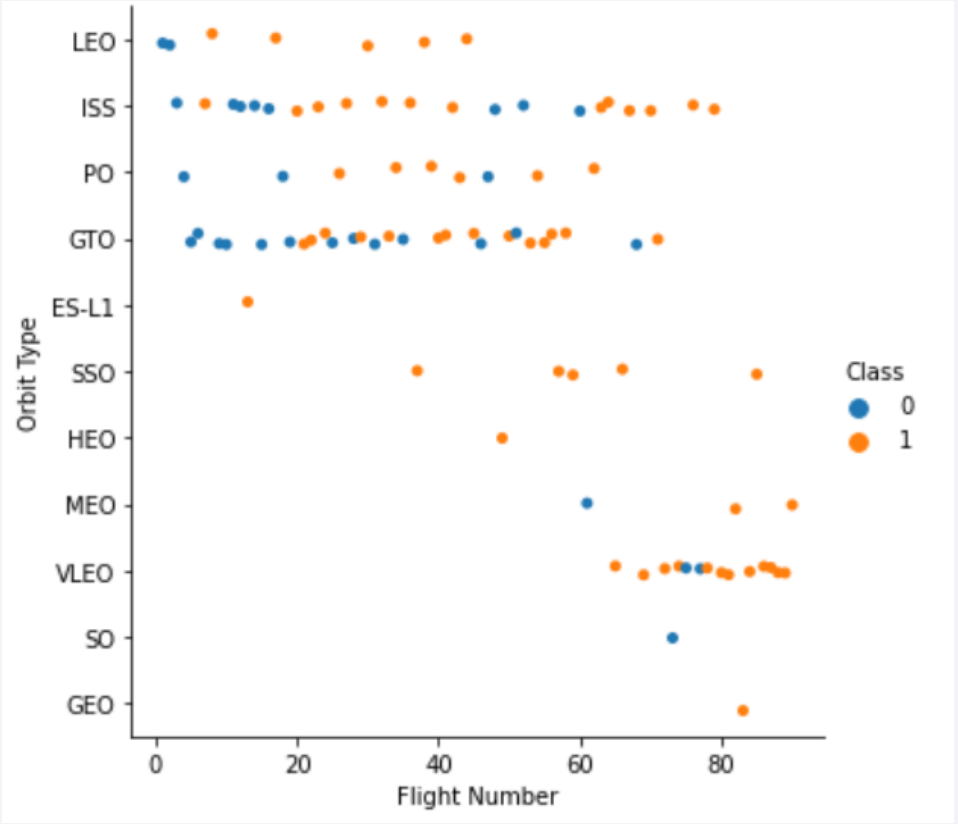
- Orbit types **SSO, HEO, GEO and ES-L1 have the highest success rates at 100%**

- The success rate of orbit type **GTO** is only 50% which is the **lowest** except for type SO which recorded failure in a single attempt



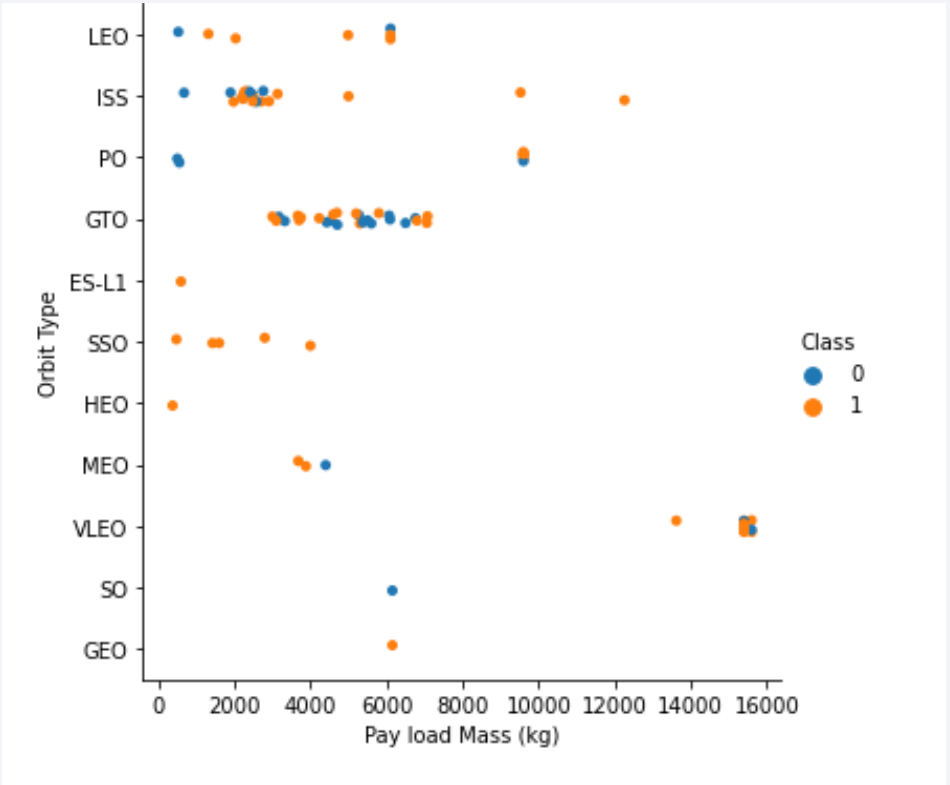
Flight Number vs. Orbit Type

- Class 0 (blue) represents unsuccessful launch, and Class 1 (orange) represents successful launch
- In most cases, the launch outcome seems to be correlated with the flight number
- Though in GTO orbit, there seems to be no relationship between flight numbers and success rate
- SpaceX stars with LEO with a moderate success rate. It appears VLEO which has a high success rate has been used in the most recent launches



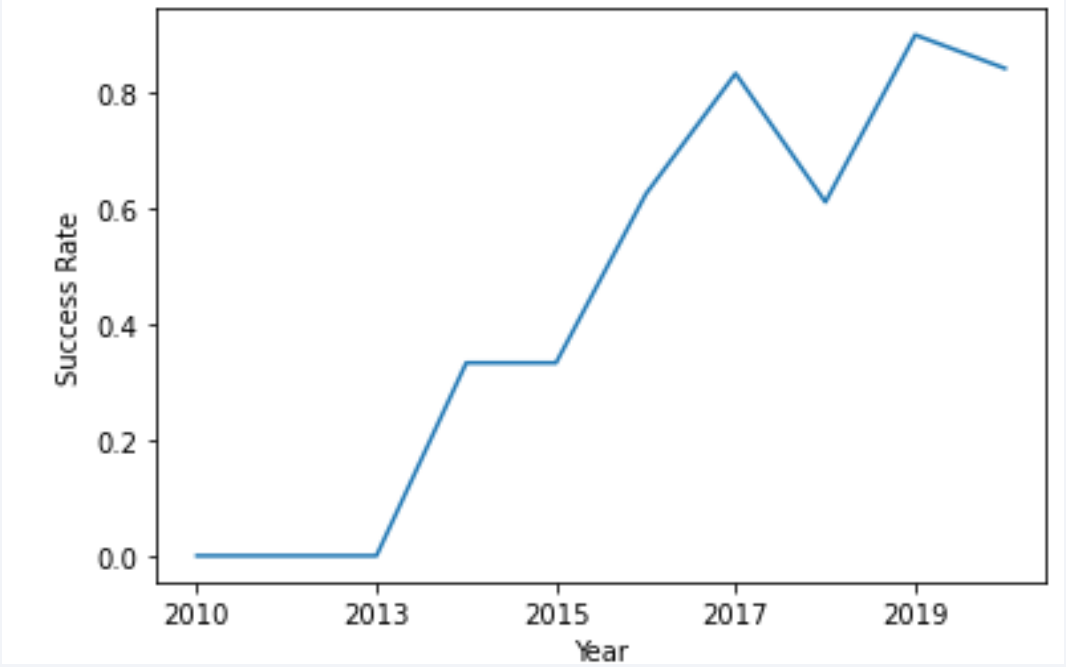
Payload vs. Orbit Type

- Class 0 (blue) represents unsuccessful launch, and Class 1 (orange) represents successful launch
- With heavy payloads the successful landing or positive landing rates are higher for LEO and ISS
- In the case of GTO it's hard to distinguish between the positive and negative landing rates because they're all clustered together



Launch Success Yearly Trend

- Since 2013, the success rate increased until 2017
- That changed as it decreased in 2018
- Recently, there has been a success rate of about 80%



All Launch Site Names

- Query

```
SELECT DISTINCT LAUNCH_SITE
FROM SPACEXTBL
```

- Result

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

- When the SQL Distinct clause is used in the query, only unique values are displayed in the Launch_Site column from the SpaceX table

- There are four unique launch sites: CCAFS LC-40, CCAFS SLC-40, KSC LC-39A and VAFB SLC-4E

Launch Site Names Begin with 'CCA'

- Query:

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

- Result:

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	None	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	None	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	None	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	None	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	None	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- Only five records of the SpaceX table were displayed using LIMIT 5 clause in the query
- Using the LIKE operator and the percent sign(%) together, the Launch_Site name starting with CAA could be called.

Total Payload Mass

- Query:

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

- Result:

total_payload_mass_kg
45596

- Using the SUM() function to calculate the sum of column PAYLOAD_MASS__KG_
- In the WHERE clause, filter the dataset to perform calculations on if Customer is NASA (CRS)

Average Payload Mass by F9 v1.1

- Query:

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

- Result:

avg_payload_mass_kg
2928

- Using the AVG() function to calculate the average value of column PAYLOAD_MASS_KG_
- In the WHERE clause, filter the dataset to perform calculations only if Booster_version is F9 v1.1

First Successful Ground Landing Date

- Query:

```
SELECT MIN(DATE) AS first_successful_landing_date
FROM SPACEXTBL
WHERE LANDING  OUTCOME = 'Success (ground pad)'
```

- Result:

first_successful_landing_date
2015-12-22

- Using the MIN () function to find out the earliest date in the column DATE
- In the WHERE clause, filter the dataset to perform a search only if Landing_outcome is Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

- Query:

```
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

- In the WHERE clause, filter the dataset to perform a search if Landing_outcome is Success (drone ship)
- Using the AND operator to display a record if additional condition PAYLOAD_MASS__KG_ is between 4000 and 6000

Total Number of Successful and Failure Mission Outcomes

- Query:

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

- Result:

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

- Using the COUNT() function to calculate the total number of columns
- Using the GROUP BY statement, group rows that have the same values into summary rows to find the total number in each Mission_outcome.
- According to the results, SpaceX seems to have **successfully completed nearly 99% of their missions.**

Boosters Carried Maximum Payload

- Query:

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

- Using a subquery, first find the maximum value of the payload by using MAX() function and second, filter the dataset to perform a search if PAYLOAD_MASS__KG_ is the maximum value of the payload
- According to the result, version F9 B5 B10xx.x boosters could carry the maximum payload.

- Result:

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

- Query:

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

- Result:

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

- In the WHERE clause, filter the dataset to perform a search if Landing__outcome is Failure (drone ship)
  - Using the AND operator to display a record if additional condition YEAR is 2015
- In 2015 there were two landing failures on drone ships

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

- Query:

```
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
```

- In the WHERE clause, filter the dataset to perform a search if the date is between 2010-06-04 and 2017-03-20
- Using the ORDER By keyword to sort the records by total number of landings and using DESC keyword to sort the records in descending order.
- According to the results, the number of successes and failures between 2010-06-04 and 2017-03-20 were similar

- Result:

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

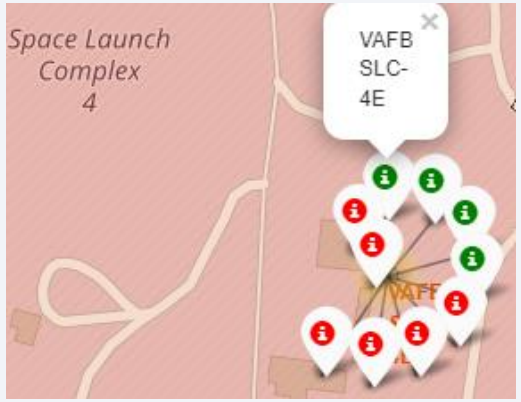

Section 3
Launch Sites
Proximities Analysis

SpaceX Launch Site Locations

- The larger map shows all SpaceX launch sites, while the smaller one shows that all their launch sites are in the United States

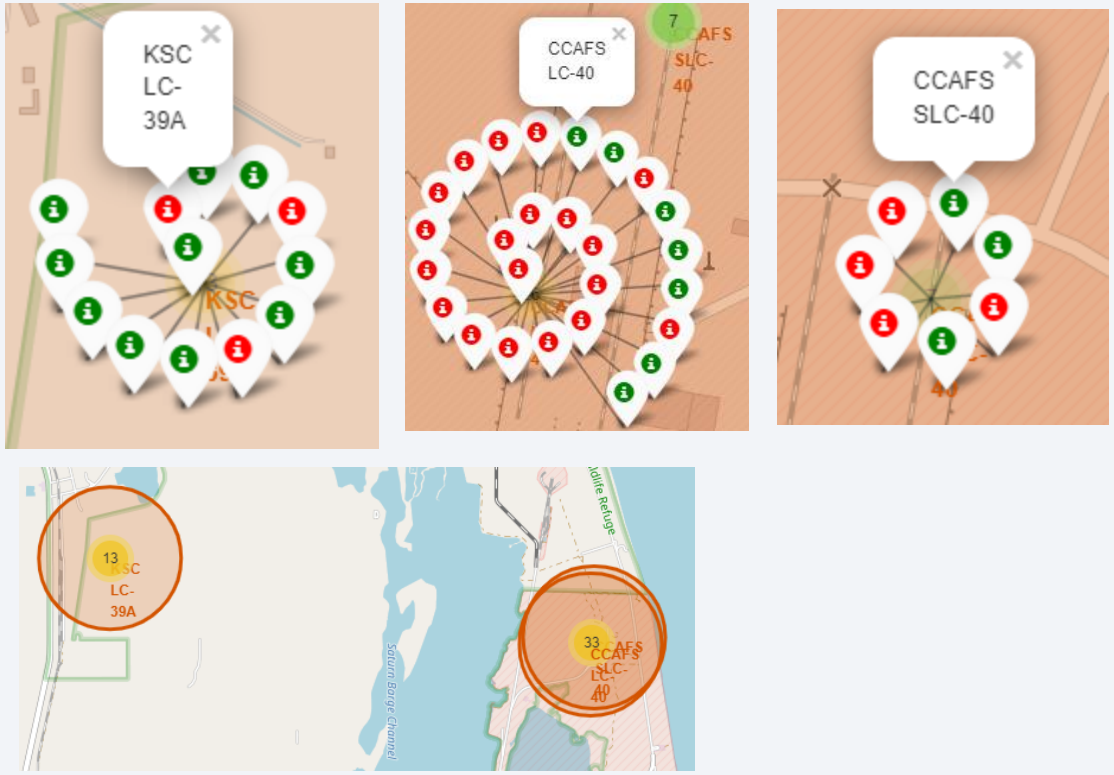
- All of SpaceX's launch sites are near the coast

Color-labeled Launch Outcomes



Launch site in California

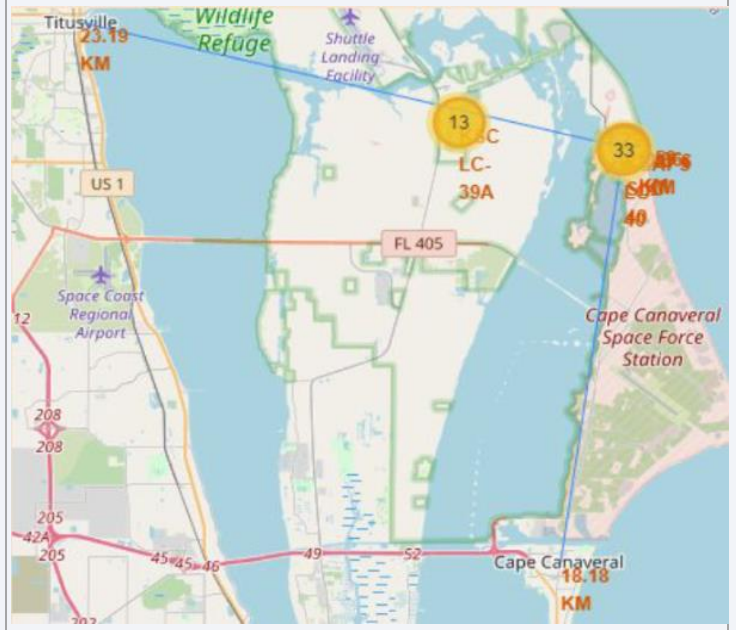
- By clicking on the marker clusters, successful landings (green) and failed landings (red) are displayed



Launch Sites in Florida

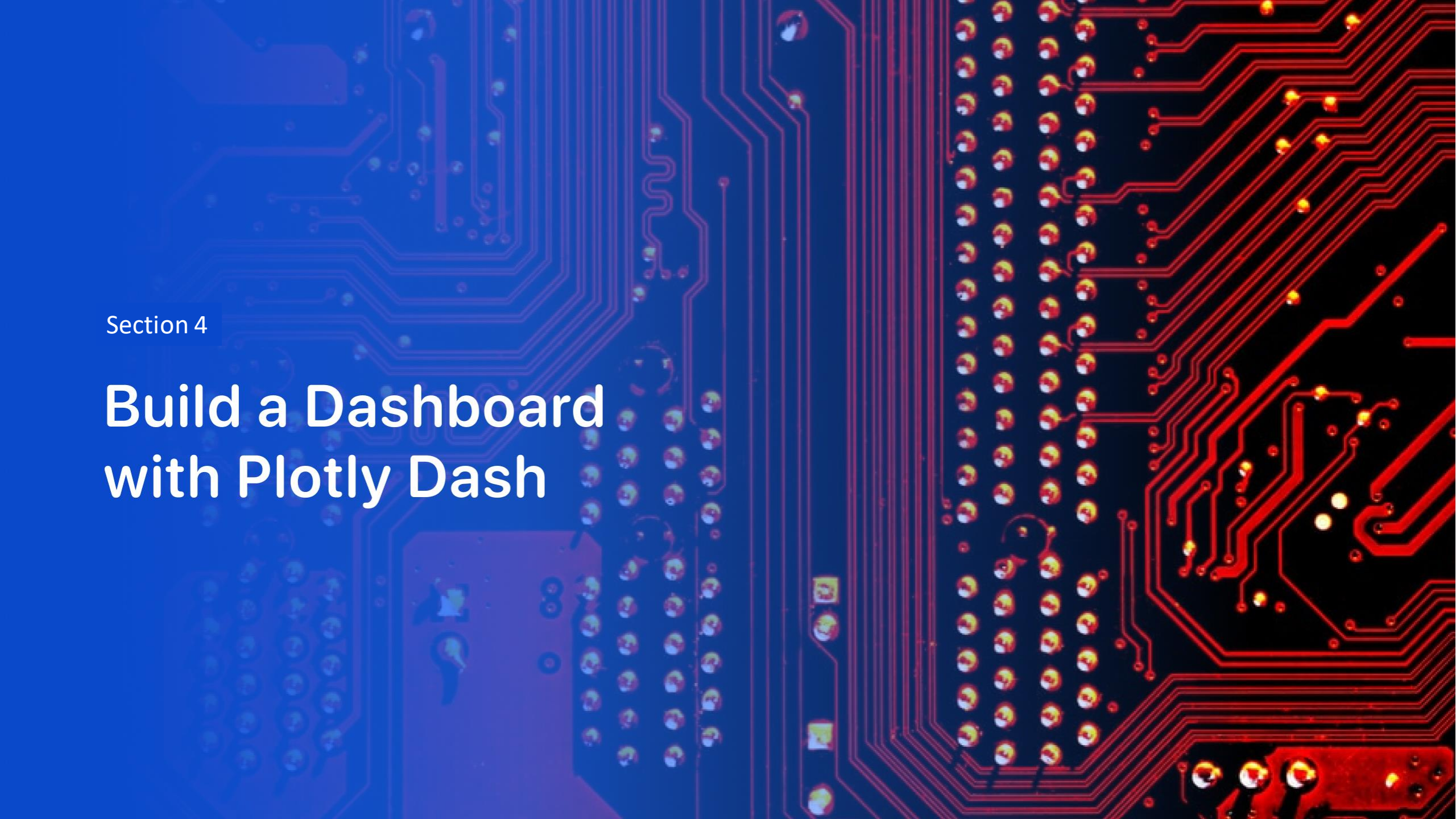
Proximities of Launch Sites

Are launch sites in close proximity to railways? **Yes**
Are launch sites in close proximity to highways? **Yes**
Are launch sites in close proximity to coastline? **Yes**

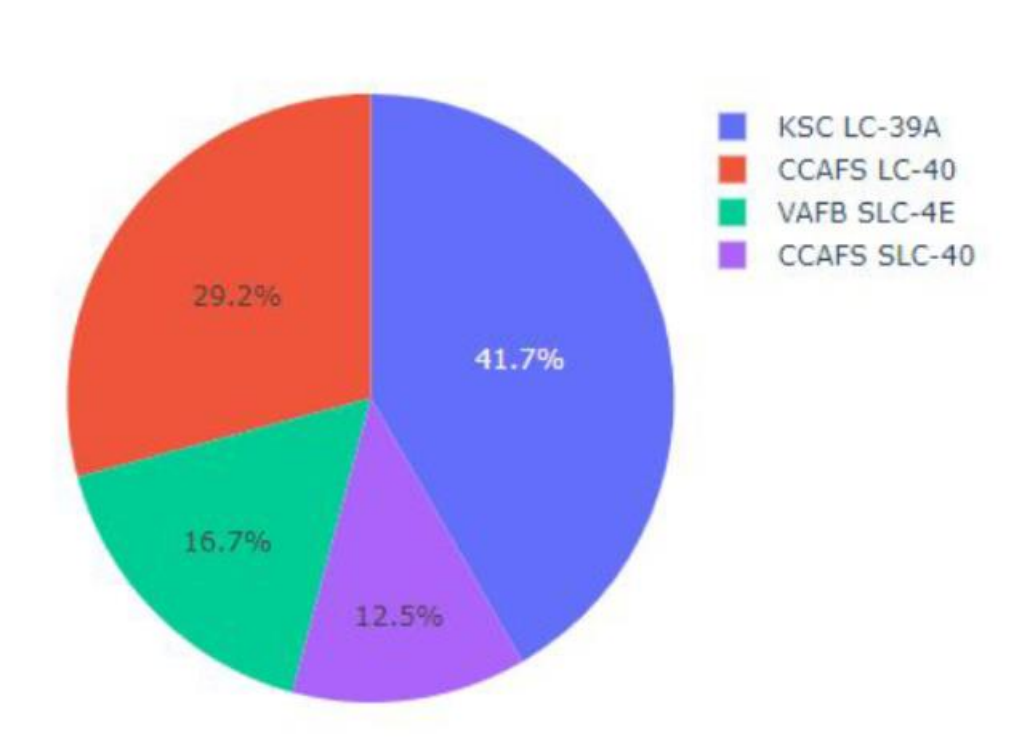


Do launch sites keep certain distances away from cities? **Yes**

- The map shows launch sites are **close to railways and highways** to transport equipment or personnel while also **close to the coastline** and relatively **far from the cities** so potential launch failure doesn't pose a threat

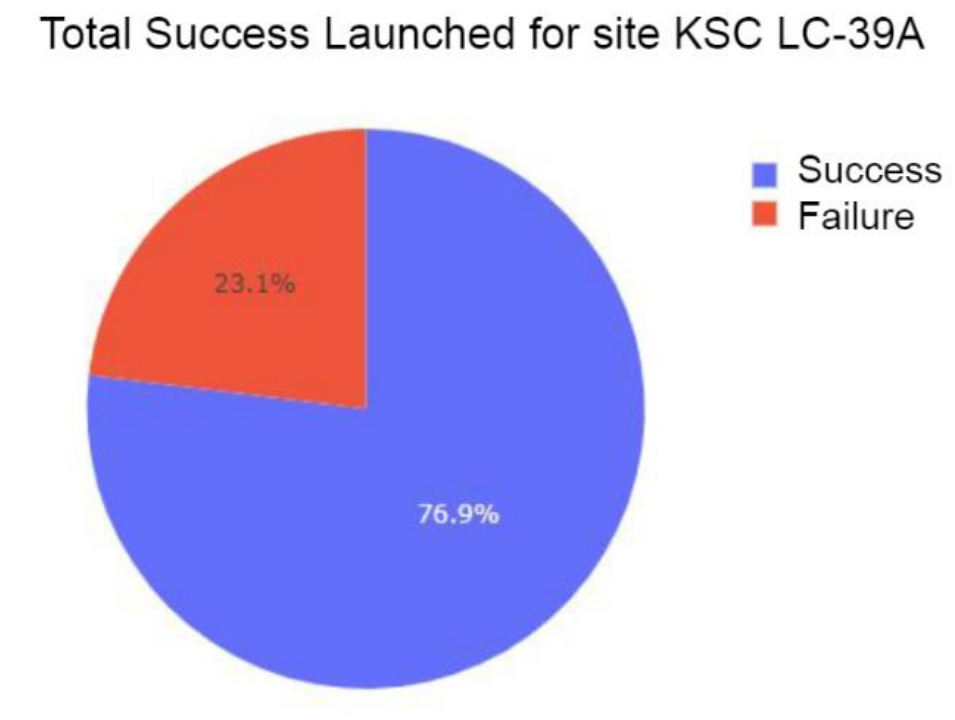

Section 4
Build a Dashboard with Plotly Dash

Launch Success Count For All Sites



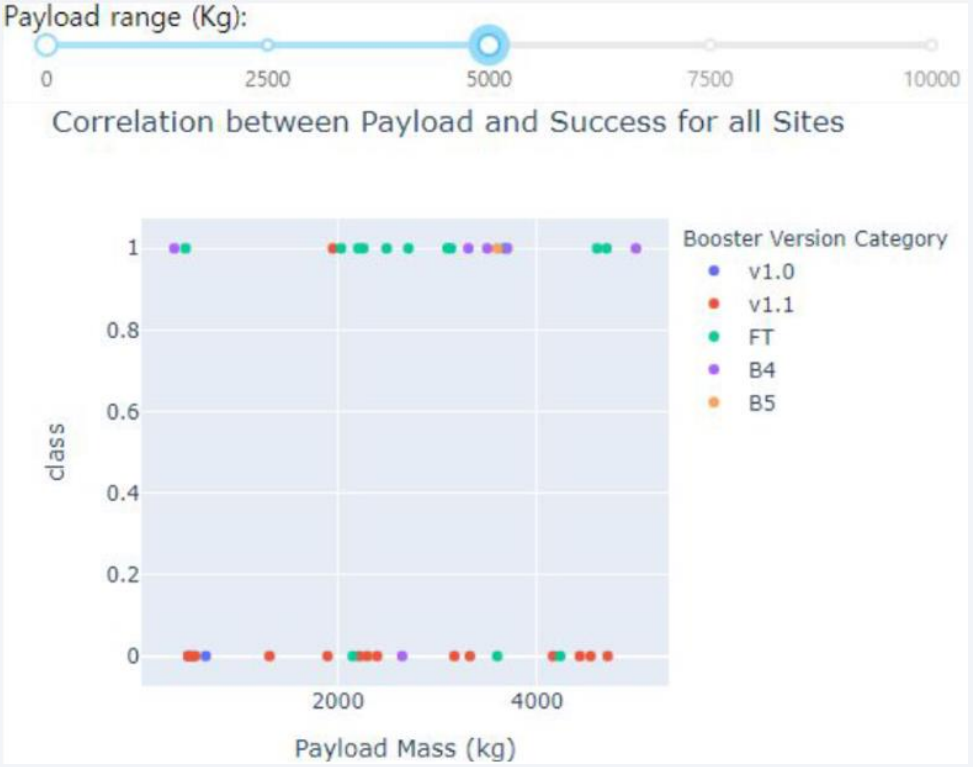
- KSLC LC-39A records the most launch success among all sites
- VAFB SLCE-4E has the third highest success, but the smallest sample of launches
  - It's also the only site located in California, so there could be more difficulty in performing launches

Launch Site with Highest Launch Success Ratio



- KSLC-39A has the highest success rate with 10 of their 13 launches being successes which is a success rate of 76.9%

Payload vs Launch Outcome Scatter Plot for all sites

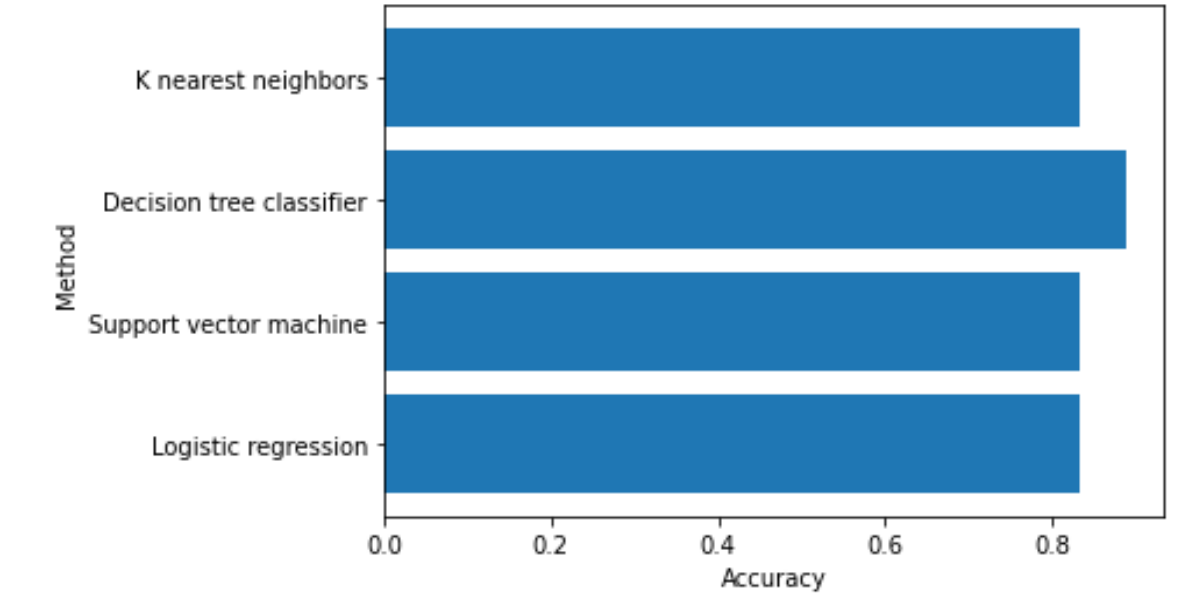


- These figures show that the launch success rate (class 1) for low weighted payloads (0-5000 kg) is higher than that of heavy weighted payloads (5000-1000 kg)

Section 5

Predictive Analysis (Classification)

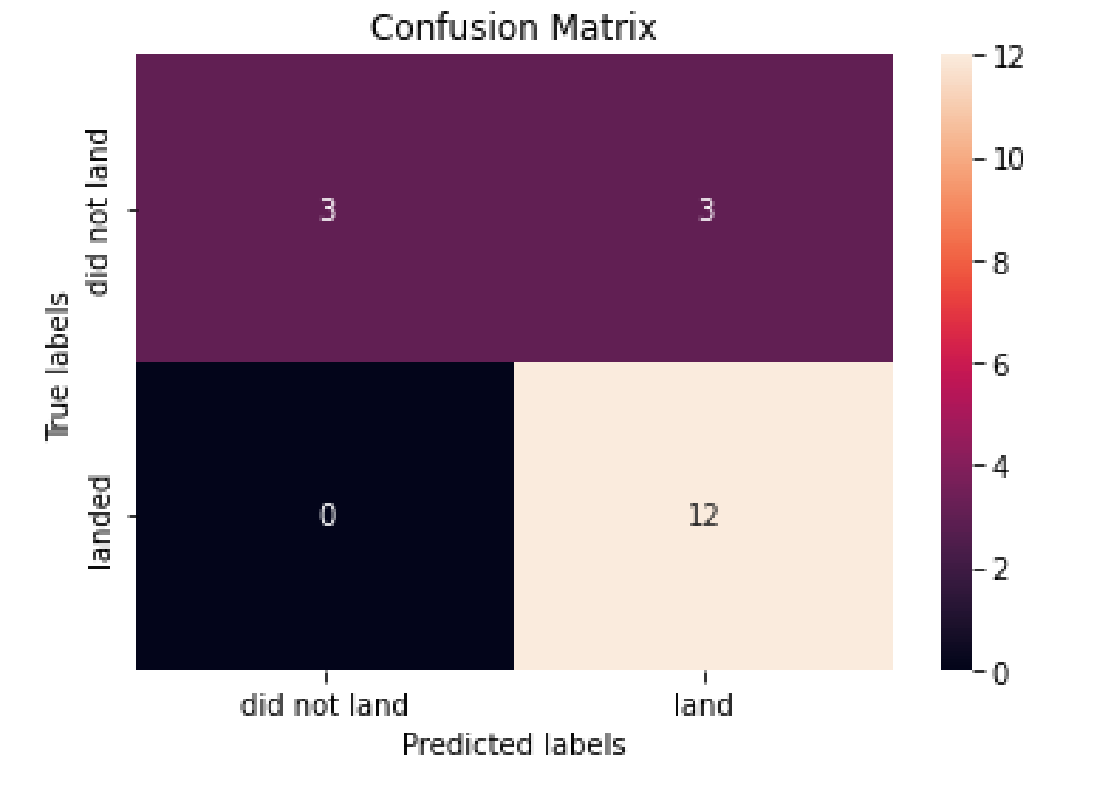
Classification Accuracy



Method	Accuracy
Logistic Regression	0.833333
Support vector machine	0.833333
Decision tree classifier	0.888888
K nearest neighbors	0.833333

- In the test set, all models had an accuracy of at least 83.33%
- Decision tree classifier was the highest at 88.88%
- It should be noted that at 18, the test size was small
- As a result, more data is needed to determine the optimal model

Confusion Matrix



- Despite the decision tree classifier being slightly more accurate, the confusion matrix was the same for all the models. This could be due to the limited sample and all four performing similarly
- The models predicted 12 landings when the true label was successful and 3 failed landings when the true label was failure. However, there were also 3 predictions that said successful landings when the true label was failure (false positive)
- Overall, **the models predicted successful landings**

Conclusions

- As the number of flights increased, so did the success rate and it's recently exceeded 80%
- Orbital types SSO, HEO, GEO, and ES-L1 have the highest success rate (100%)
- The launch sites are close to railways, highways, and coastline but far from cities
- KSLC-39A has the highest number of launch successes and the highest success rate among all sites
- The launch success rate of low weighted payloads is higher than that of heavy weighted payloads
- In this dataset all models had accuracy of at least 83.33%, but due to small sample size, more data is likely needed to determine the optimal model

Appendix

- GitHub Repository URL

- Coursera Applied Data Science Capstone Course URL

Thank you!