Winning Space Race with Data Science

Clony Nunes de Abreu Júnior
23-05-2022

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- **Project background and context**

- As NASA's role has been established in developing technologies that can be used in space, the space transportation service has been lacking in those who could perform it, so this gap has been filled by companies like SpaceX, whose main role is to offer space transportation services, and to do this, SpaceX has sought to develop ways to cheapen the costs of this service. One of SpaceX's biggest gains is providing a cheap first stage launches of rockets such as the Falcon 9, which makes the cost relatively cheap. 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. 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.

- **Problems you want to find answers**

- Correlate each rocket feature with the successful landing rate

- Identify the conditions to obtain the best results and ensure the best successful landing rate

Section 1

Methodology

Methodology

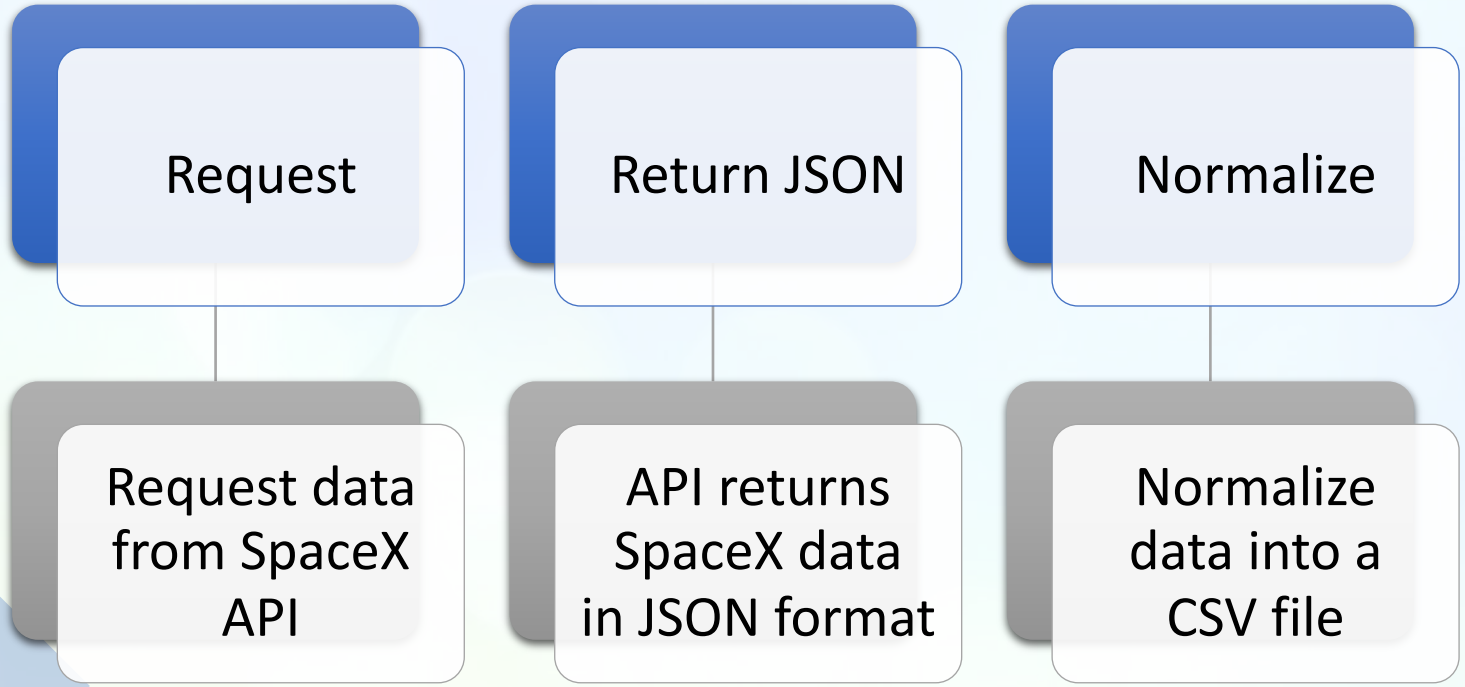
- Executive Summary
- Data collection methodology:
  - SpaceX API (https://docs.spacexdata.com/#intro);
  - Webscraping data on Wikipedia Falcon 9 page
- Perform data wrangling
  - Identify the outcomes with the booster landing successful or unsuccessful and convert them into training labels
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Identifying the best hyperparameters for SVM, Classification Trees and Logistic Regression algorithms

Data Collection

- The data collection process merges API requests from the SpaceX API and web scraping data from a table in the Wikipedia page of SpaceX. The SpaceX API Data Columns are as follow:
  - FlightNumber,
  - Date,
  - BoosterVersion,
  - PayloadMass,
  - Orbit,
  - LaunchSite,
  - Outcome,
  - Flights,
  - GridFins,
  - Reused,
  - Legs,
  - LandingPad,
  - Block,
  - ReusedCount,
  - Serial,
  - Longitude,
  - Latitude.
- The Wikipedia Web Scrape Data Columns are as follow:
  - Flight No.,
  - Launch site,
  - Payload,
  - PayloadMass,
  - Orbit,
  - Customer,
  - Launch outcome, Version Booster, Booster landing, Date, Time.

Data Collection

- SpaceX API process



Data Collection

- WebScraping process

Get Response	Extract Data	Normalize
• Get HTML Response from Wikipedia page	1. Use BeautifuSoup Library	• Normalize data into a CSV file

Data Collection – SpaceX API

1. Request to SpaceX API Rocket Launch Data

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

response = requests.get(spacex_url)
```

2. Convert response to JSON format

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

3. Cleaning data

```
# Call getBoosterVersio
getBoosterVersion(data)

# Call getLaunchSite
getLaunchSite(data)

# Call getPayloadData
getPayloadData(data)

# Call getCoreData
getCoreData(data)
```

4. Uniting columns into a dictionary and creating a dataframe

```
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}
```

```
# Create a data from launch_dict
launch_df = pd.DataFrame.from_dict(launch_dict)
```


GitHub Respository

Data Collection – SpaceX API

5. Filtering and exporting dataframe into a CSV file

```
# Hint data['BoosterVersion']!='Falcon 1'
df_falcon9 = launch_df[launch_df['BoosterVersion'] == 'Falcon 9']
```

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```


GitHub Respository

Data Collection - Scraping

1. Getting Response from webpage

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

2. Creating BeautifulSoup object from a response text

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

3. Creating a list with all tables elements

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

4. Extracting the column names

```
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)
```

5. Creating empty dictionary

```
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. Fill up launch_dict dictionary[1]

```
launch_dict = fill_launch_dict()
```

GitHub Respository

[1]Refer to jupyter notebook named “02_Capstone_Data Collection with Web Scraping.ipynb” to see fill_launch_dict function in details

Data Collection - Scraping

7. Creating and exporting dataframe to CSV file

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

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


GitHub Respository

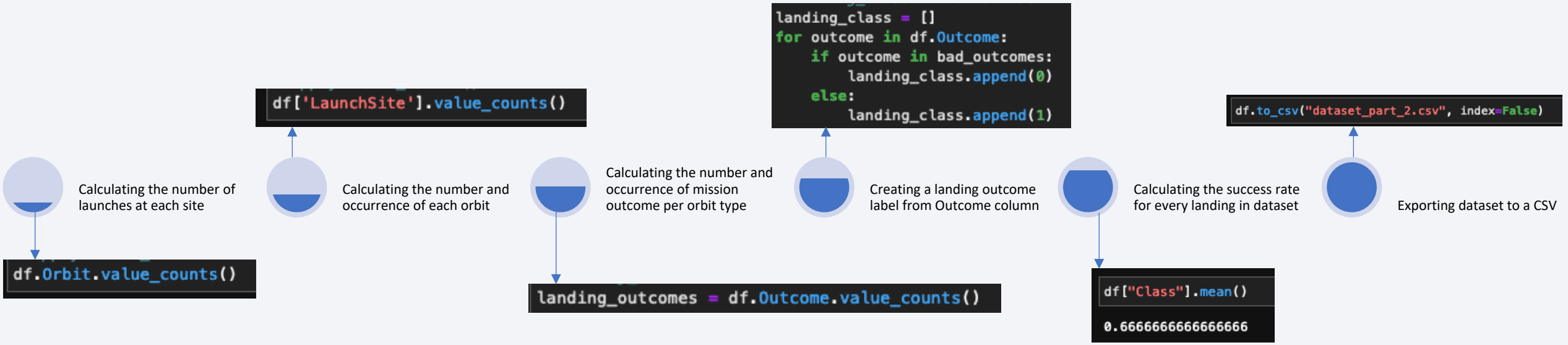
Data Wrangling

- The dataset presented several cases in which the booster failed to successfully land. These results were converted into training label of 1=successfully and 0=failure and are represented in the table below:

Description	Result (Outcome)	Landing Region	Training label
True Ocean	Success	Specifc region of the ocean	1
False Ocean	Unsuccess	Specifc region of the ocean	0
True RTLS	Success	Ground pad	1
False RTLS	Unsuccess	Ground pad	0
True ASDS	Success	Drone ship	1
False ASDS	Unsuccess	Drone ship	0

GitHub Respository

Data Wrangling



GitHub Respository

EDA with Data Visualization

- **Scatter chart:** 
  - Flight Number vs. Launch Site
  - Payload vs. Launch Site
  - Flight Number vs. Orbit Type
  - Payload vs. Orbit Type

A scatter plot shows the relationship between two variables and this is called a correlation. This kind of plot evidences how much one variable is affected by another.

- **Bar chart:** 
- Orbit Type vs. Success Rate

A Bar chart makes it easy to compare datasets between multiple groups. One axis represents a category, and the other axis represents a discrete value. This chart indicates the relationship between the two axes.

- **Line chart:** 
  - Year vs. Success Rate

A Line chart shows data variables and trends very clearly and helps predict the results of data that has not yet been recorded.

GitHub Respository

EDA with SQL

Loading the dataset into the corresponding table in a Db2 database, and executing SQL queries to answer following questions:

- 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 Respository

Build an Interactive Map with Folium

- Objects created and added to a folium map:
  - Markers showing all launch sites on a map;
  - Markers showing the success/failed launches for each site on the map;
  - Lines showing the distances between a launch site to its proximities.
- These objects help find that launch sites are close in proximity to railways, highways, coastlines and keep certain distances away from cities.

GitHub Respository

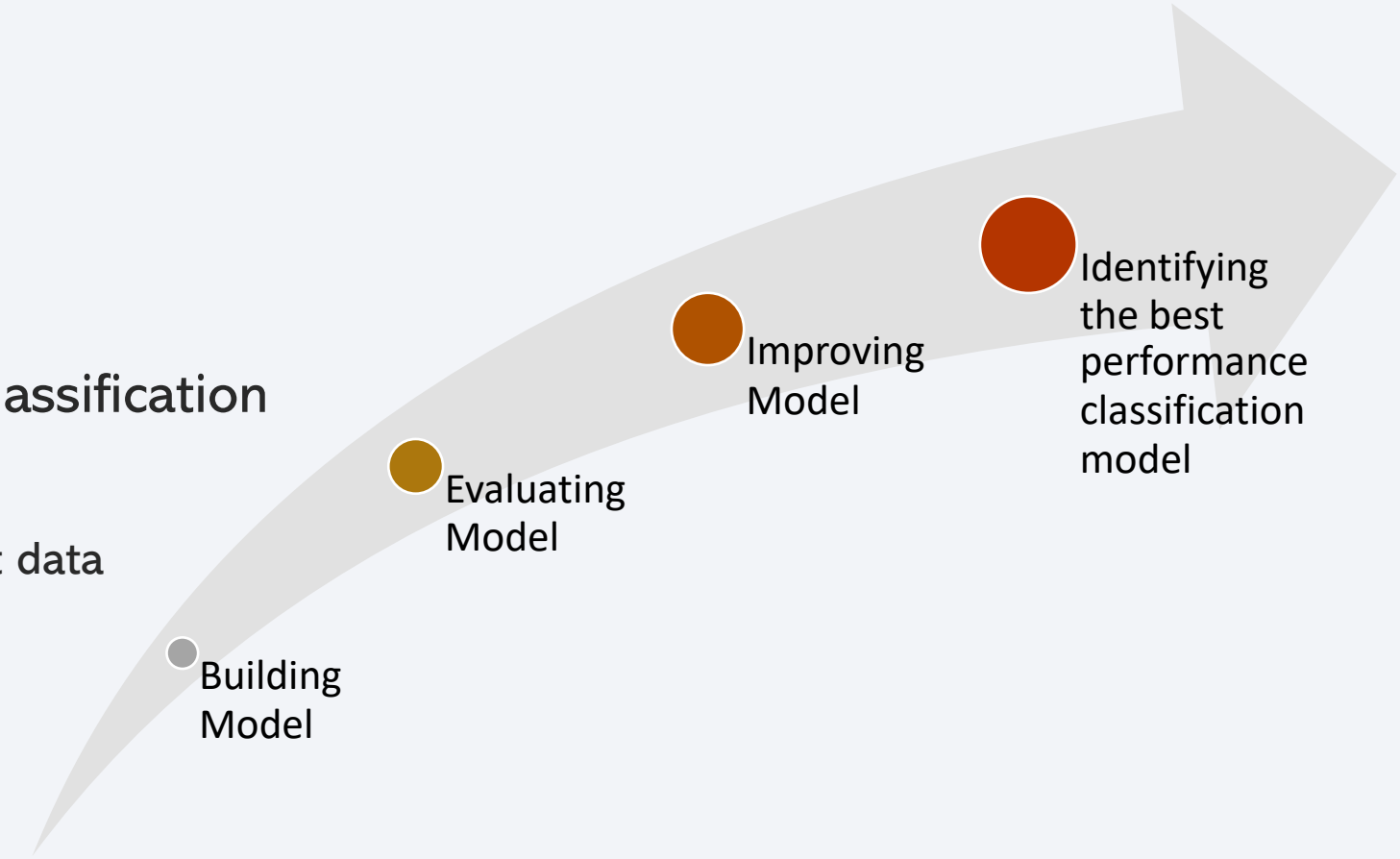
Build a Dashboard with Plotly Dash

- 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 Respository

Predictive Analysis (Classification)

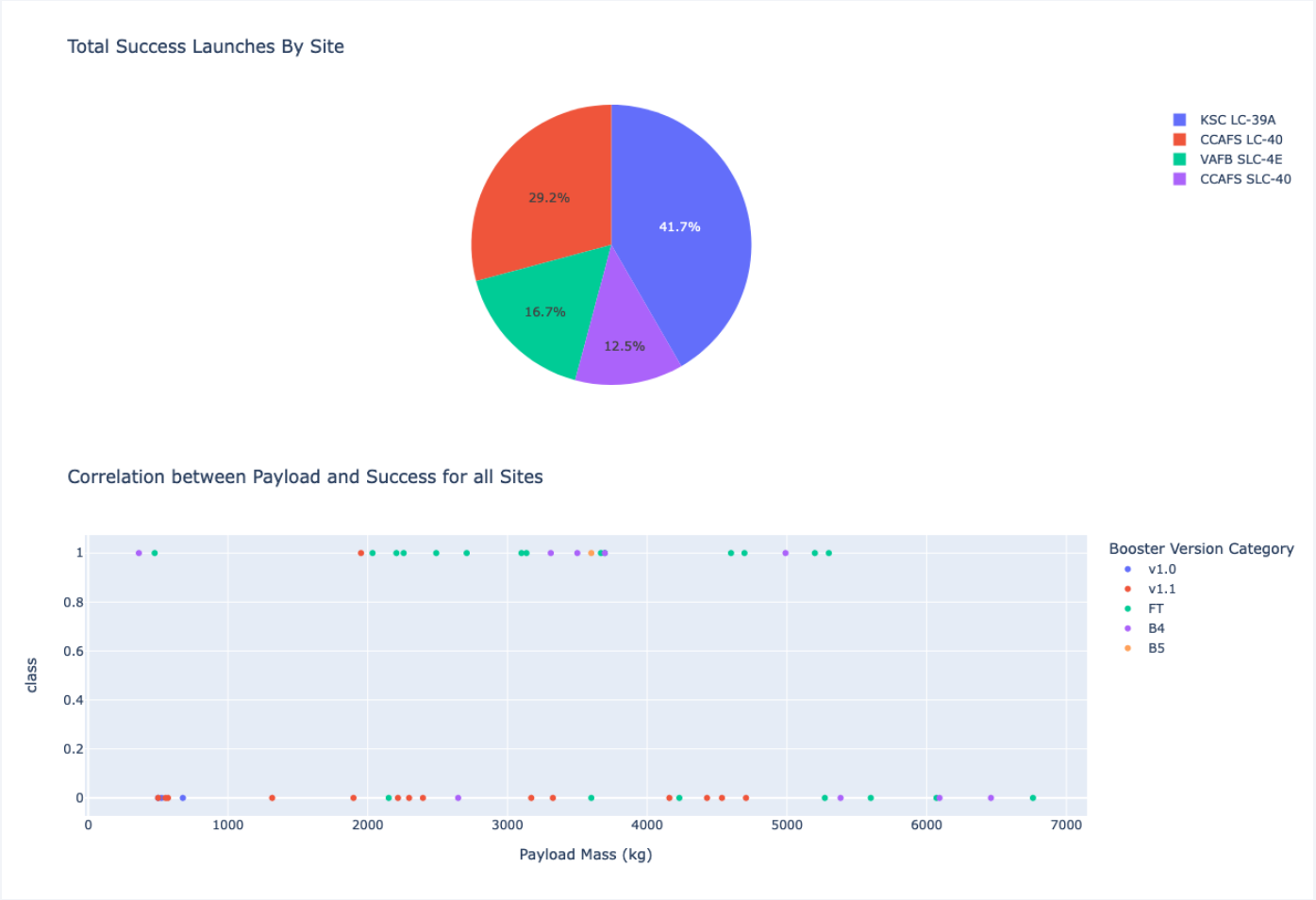
- Performing Exploratory Data Analysis and determining Training Labels to:
  - Create a column for the class
  - Standardize the data
  - Split into training data and test data
- Find best Hyperparameter for SVM, Classification Trees and Logistic Regression
  - Find the method performs best using test data



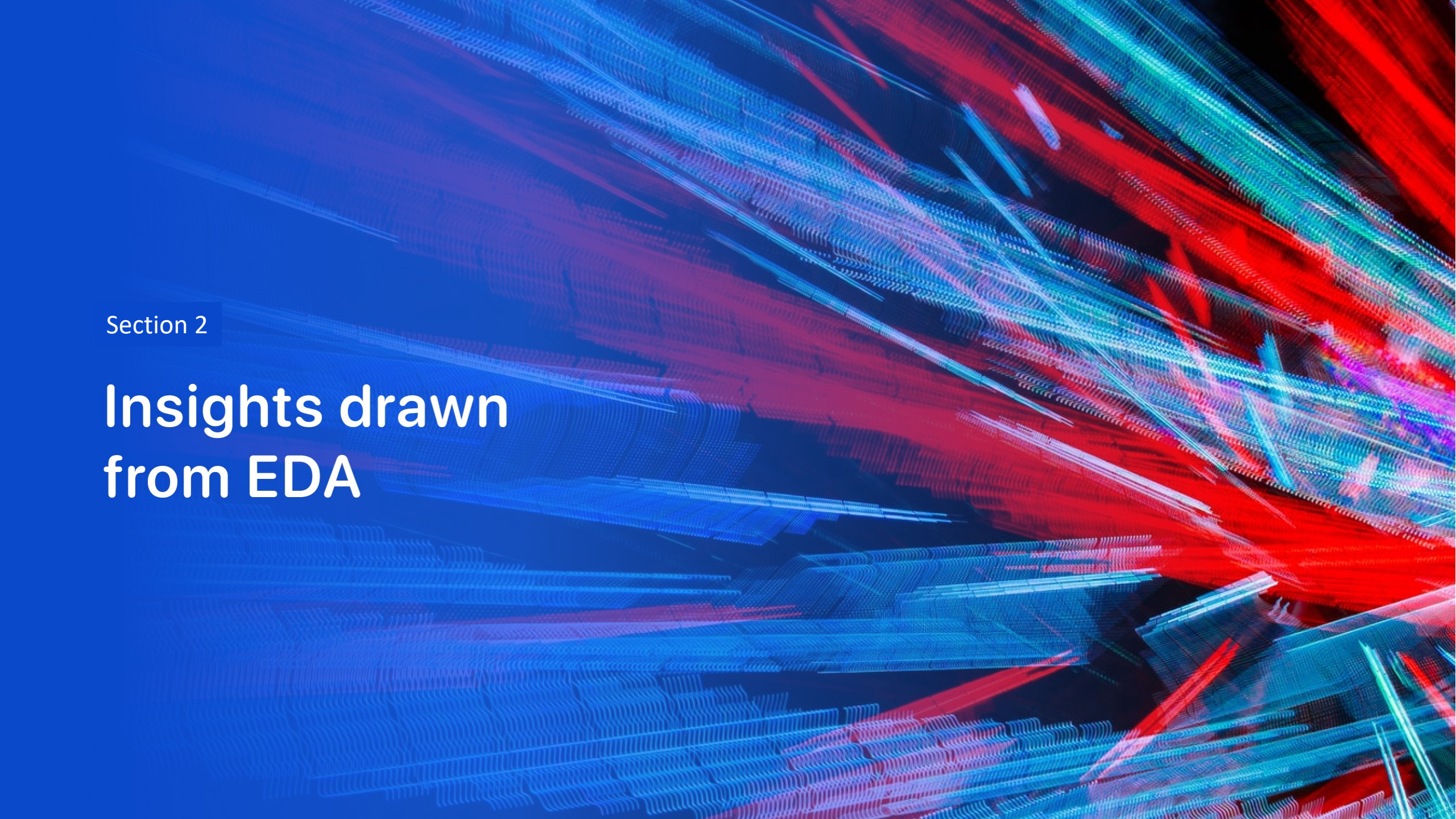
GitHub Respository

Results

- The image shows the Interactive analytics demo in screenshots;
- Other results will be shown in other slides
- The Predictive analysis results shows the same accuracy for all 4 methods i.e 83%

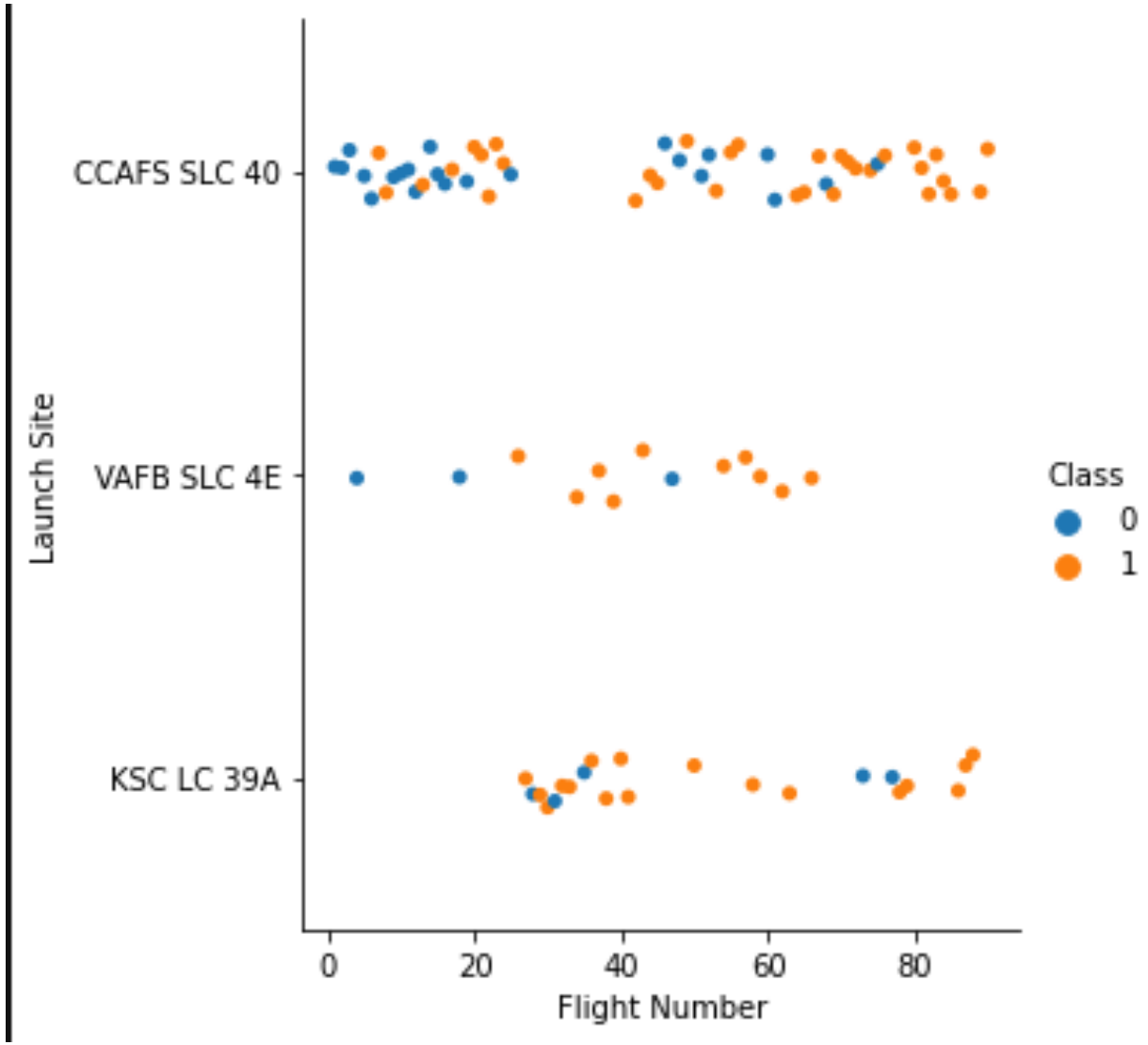


GitHub Respository


Section 2
Insights drawn from EDA

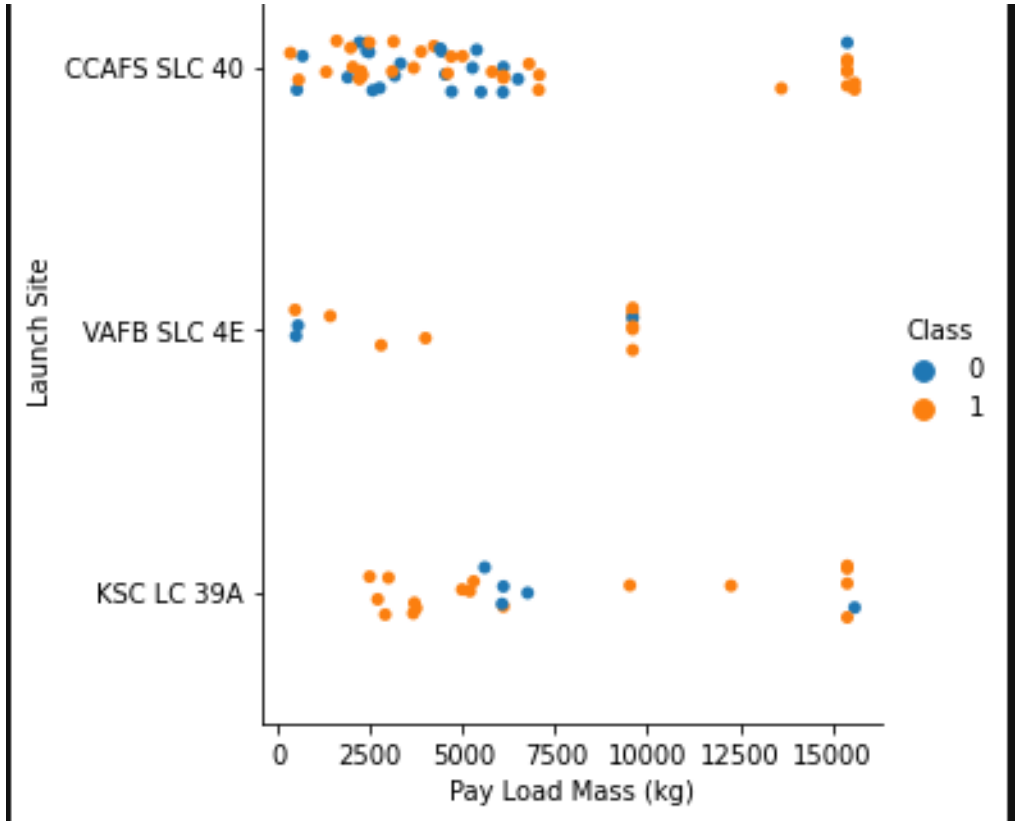
Flight Number vs. Launch Site

- As the nº of flights increase the success rate (orange dots / class = 1) raises as well
- The VAFB SLC 4E launch site had fewer flights then the other two launch sites for some reason that data can't show
- There were a considerable progress on success rate around 20 flights



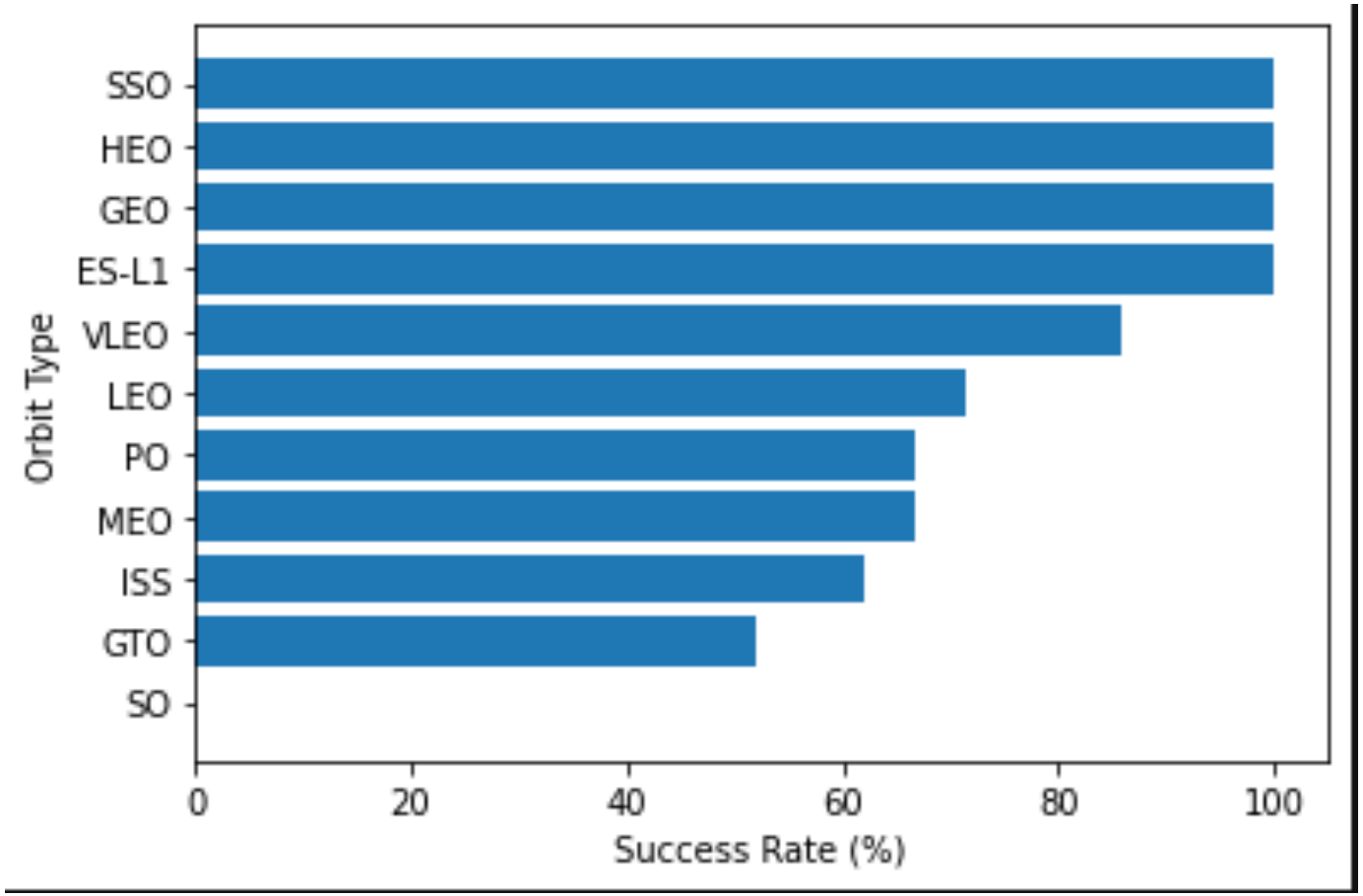
Payload vs. Launch Site

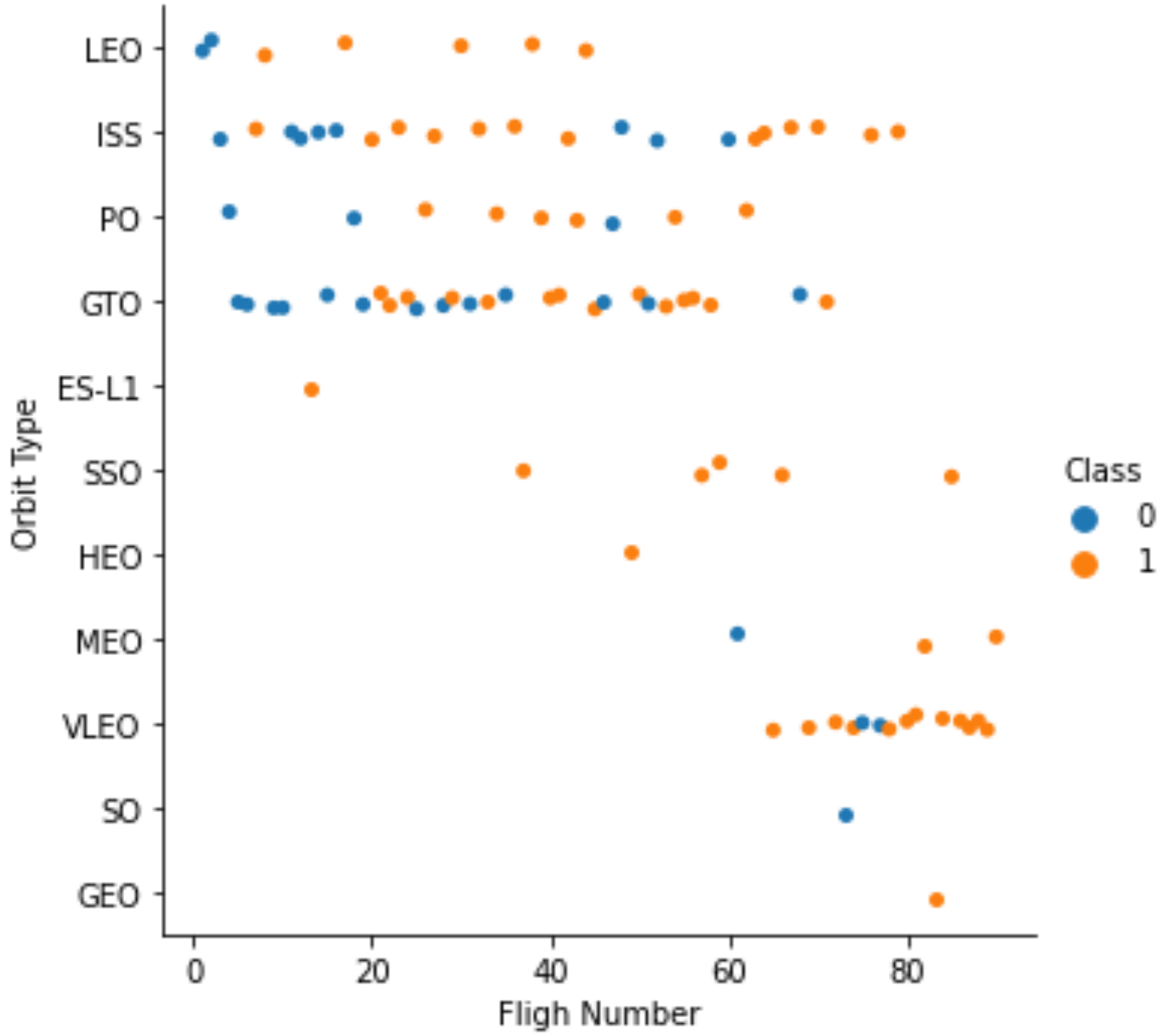
- There are no launches from the VAFB-SLC launch site with payloads of mass greater than 10000 kg
- From the graph it is not possible to identify any visible pattern relating the success of the launch and to the mass of payload.



Success Rate vs. Orbit Type

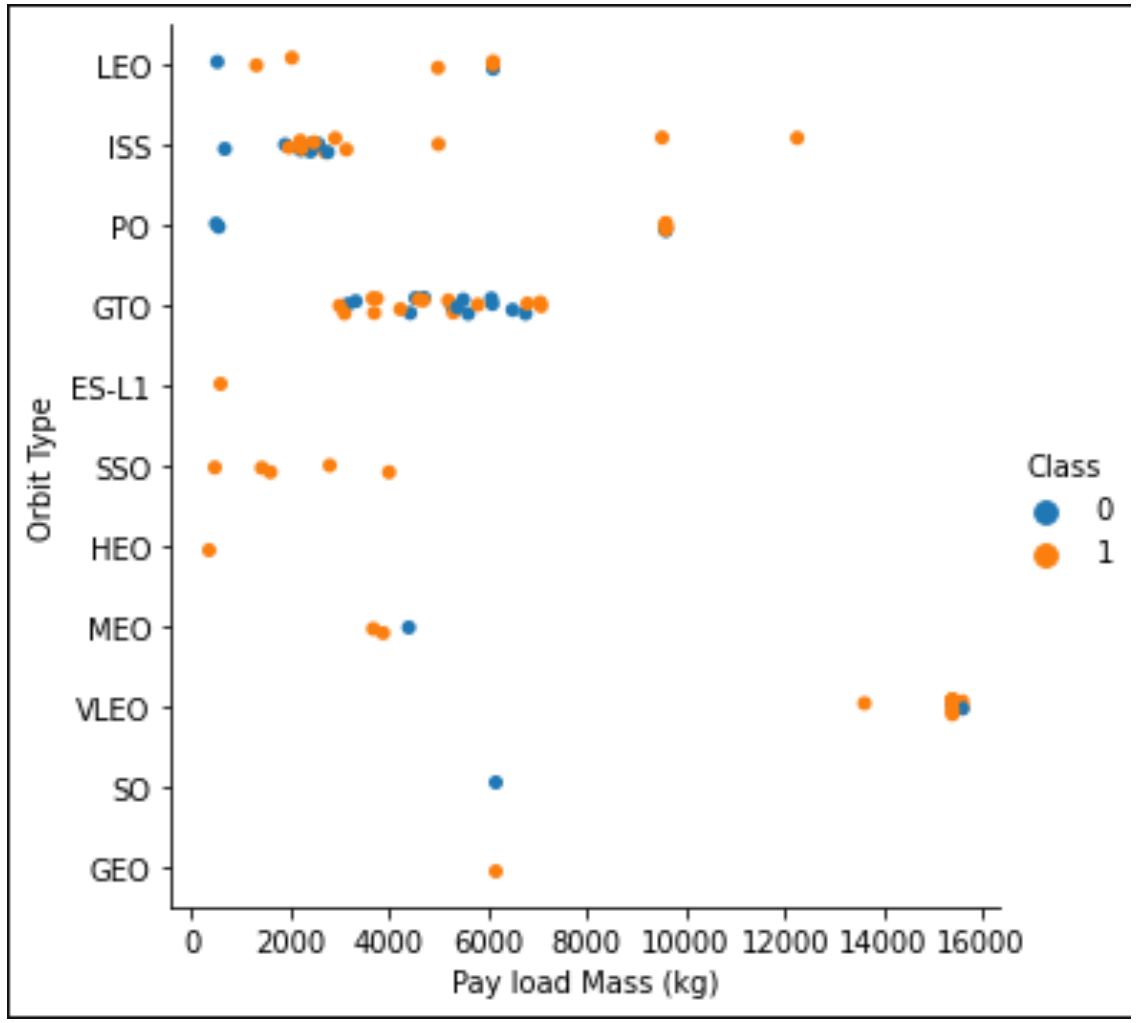
- The SO orbit registered failure in its only attempt being the orbit that registered the worst performance among all;
- Unfortunately, the success rate of orbit type GTO is only 50%, and it is the lowest among those which had at least one attempt with success;
- Orbit types SSO, HEO, GEO, and ES-L1 have the highest success rates (100%).





Flight Number vs. Orbit Type

- In the scatter plot, the LEO orbit type, evidences a relationship with the number of flights, since the orange dots (class =1) representing success, predominate as the number of flights increases;
- The GEO orbit type does not evidence the same pattern seen in LEO, since the number of failures (blue dots and class=0) is quite frequent independent of the number of flights.

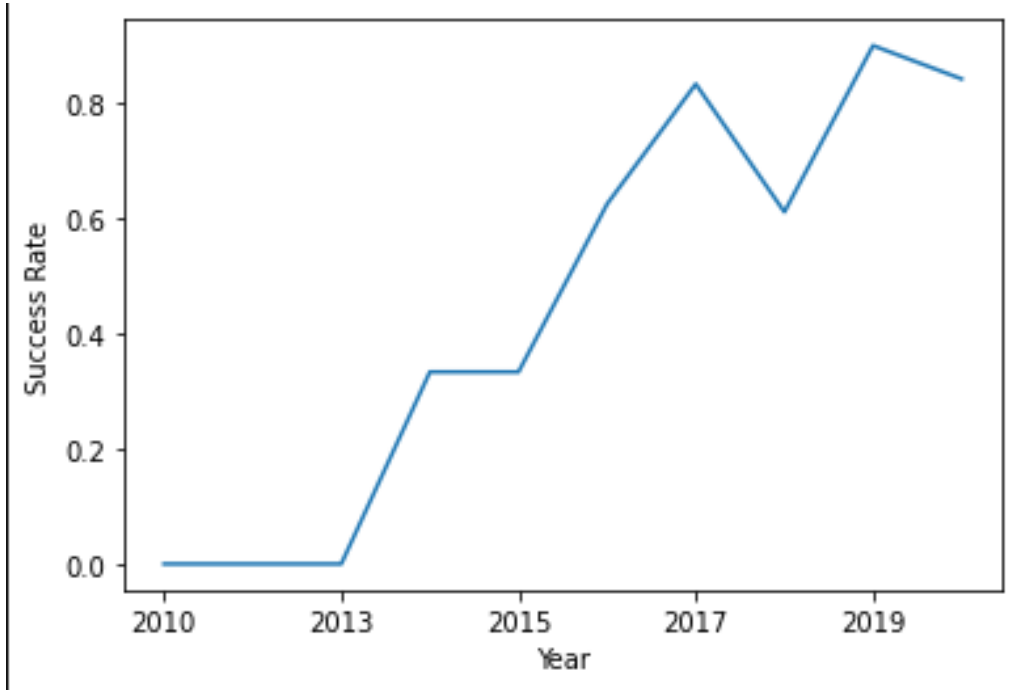


Payload vs. Orbit Type

- Orbit type GTO, it is hard to distinguish between the positive landing rate (orange dots and class=1) and the negative landing (blue dots and class = 0) because they are all very close to each other;
- LEO and ISS orbit types, evidence a positive landing rate (orange dots and class=1) at larger payloads.

Launch Success Yearly Trend

- The annual success trend is increasing from 2013 onwards;
- There is a slight drop in the success rate in the year 2018;



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, VAFB SLC-4E

Launch Site Names Begin with 'CCA'

QUERY

```
SELECT LAUNCH_SITE, PAYLOAD
FROM SPACEXTBL WHERE
LAUNCH_SITE LIKE ‘CCA%’ LIMIT 5
```

RESULT

LAUNCH_SITE	PAYLOAD
CCAFS LC-40	Dragon Spacecraft Qualification Unit
CCAFS LC-40	Dragon demo flight C1, two CubeSats,
CCAFS LC-40	Dragon demo flight C2
CCAFS LC-40	SpaceX CRS-1
CCAFS LC-40	SpaceX CRS-2

The clause 'LIMIT 5' at the end of the query assures that only 5 records will return from the database

The command like withe the % sign athe the end of the letters indicates that any launch site beginning with these letters, will return from the query, if exists

Total Payload Mass

QUERY

```
SELECT
sum(PAYLOAD_MASS__KG_)
as Total_Payload_Mass from
SPACEXTBL WHERE
CUSTOMER = 'NASA (CRS)' ;
```

RESULT

TOTAL_PAYLOAD_MASS
45596

Using the sum function to calculate the total of payload mass

The where clause assures that sum function will be applied only for 'NSA (CRS)' customers

Average Payload Mass by F9 v1.1

QUERY

```
SELECT avg(PAYLOAD_MASS__KG_)
as Avg_Payload_Mass from
SPACEXTBL
WHERE booster_version = 'F9 v1.1';
```

RESULT

AVG_PAYLOAD_MASS
2928

Using the avg function to calculate the average of payload mass

THE where clause assures that avg function will be applied only for booster version = ‘f9 v1.1’

First Successful Ground Landing Date

QUERY

```
SELECT MIN(DATE)
AS first_successful_grounded_landing_date
FROM SPACEXTBL
WHERE LANDING__OUTCOME
= 'Success (ground pad)';
```

RESULT

FIRST_SUCCESSFUL_GROUNDED_LANDING_DATE
2015-12-22

- Using the min function to discover the earliest date
- The where clause assures that the 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

- The where clause assures that the landing_outcome is ‘Success (drone ship)’
- The and clause assures that the payload mass is between 4000 and 6000 kg

Total Number of Successful and Failure Mission Outcomes

QUERY

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

RESULT

MISSION_OUTCOME	TOTAL
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Uses the count function to count each row of the query

The group by clause assures that the function count will be grouped by mission_outcome, counting each value of this columns

The query resul shows that SpaceX had 99% of success

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)
```

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

Used 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.

The query result shows that version F9 B5 B10xx.x boosters carried the maximum payload

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
Failure (drone ship)	F9 v1.1 B1012
Failure (drone ship)	F9 v1.1 B1015

The where clause, filter the dataset to perform a search if Landing__outcome is Failure (drone ship). The and operator is used to display a record if additional condition year is equals to 2015.

The query result shows that 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
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING__OUTCOME
ORDER BY total DESC;
```

RESULT

LANDING__OUTCOME	TOTAL
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3

- The where clauseclause, filter the dataset to perform a search if the date is between 2010-06-04 and 2017-03-20.
- The order by keyword sort the records by total of landing, and desc keyword sort the records in descending order.
- The query result shows that the number of successes and failures between 2010-06-04 and 2017-03-20 was similar.

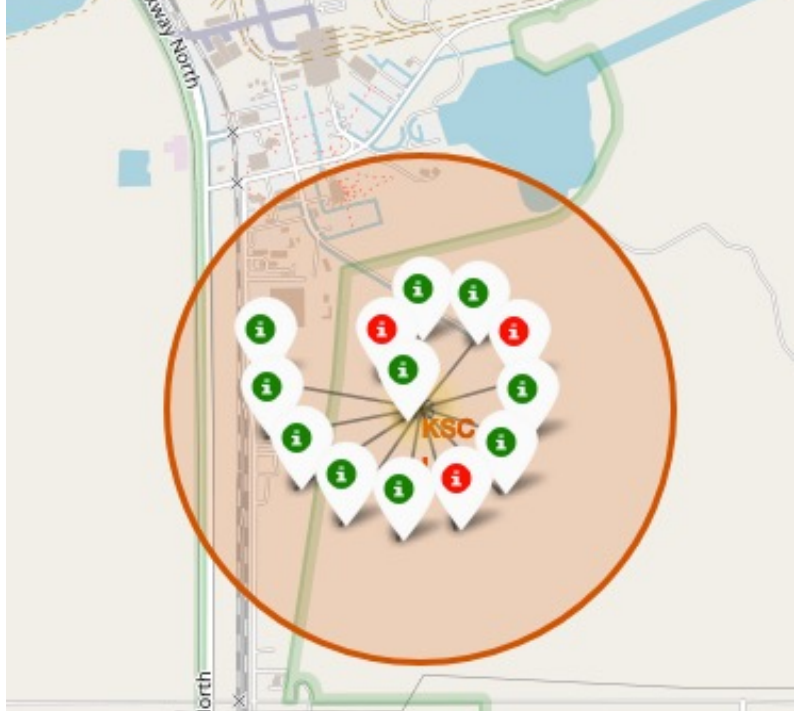

Section 3
Launch Sites
Proximities Analysis

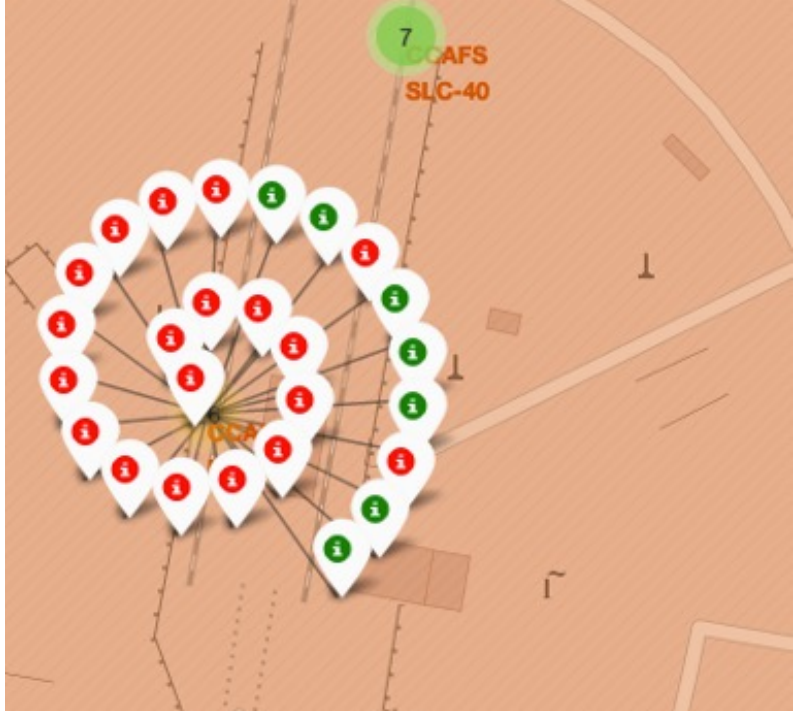
Launch Sites Location

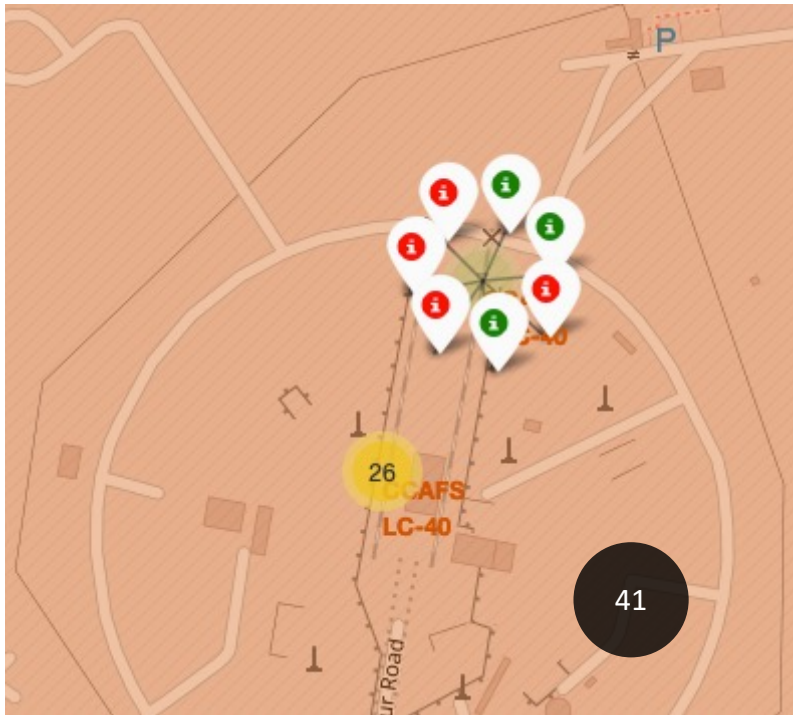
- The map shows all SpaceX launch sites, also shows that all launch sites are in the United States;
- It is possible to note that all launch sites are located on the coast;

Color-labeled launch outcomes

- Clicking on the marker clusters, successful landing (green) or failed landing (red) are displayed;
- The lower left maps shows the 10 California launch sites;
- All other maps (upper left and right, and down right) display Florida launch sites



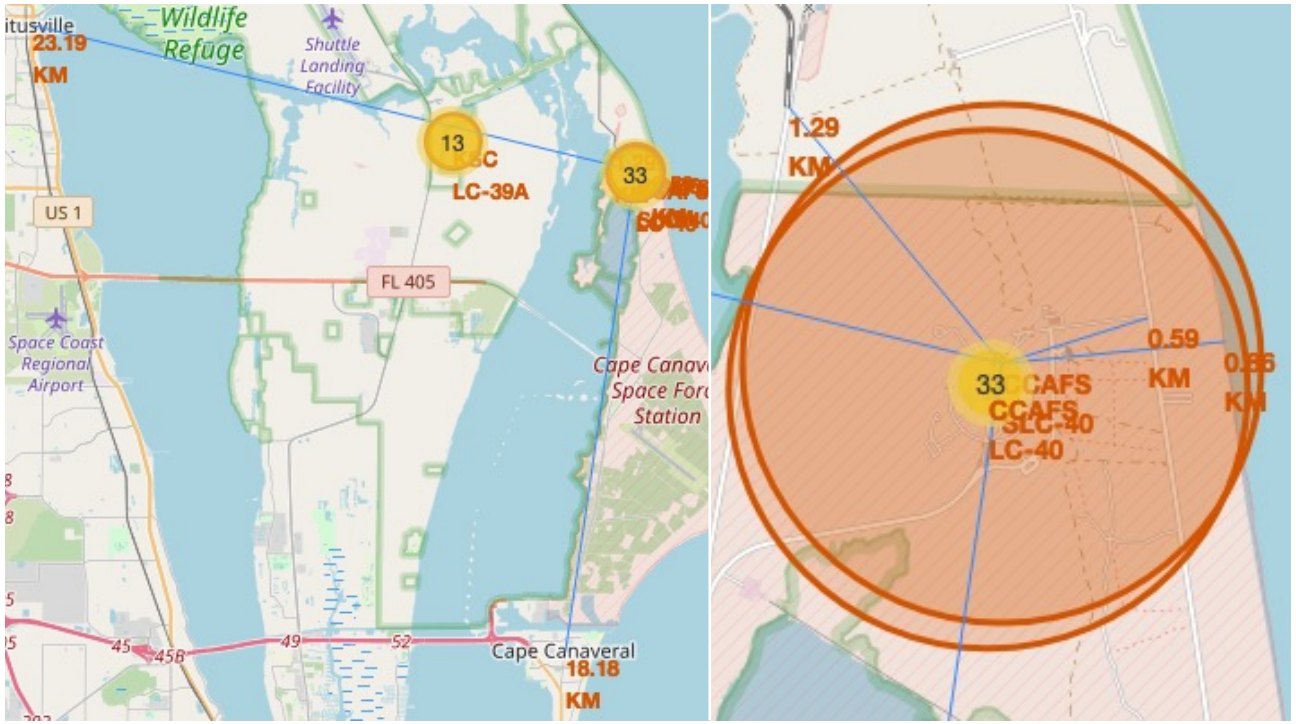


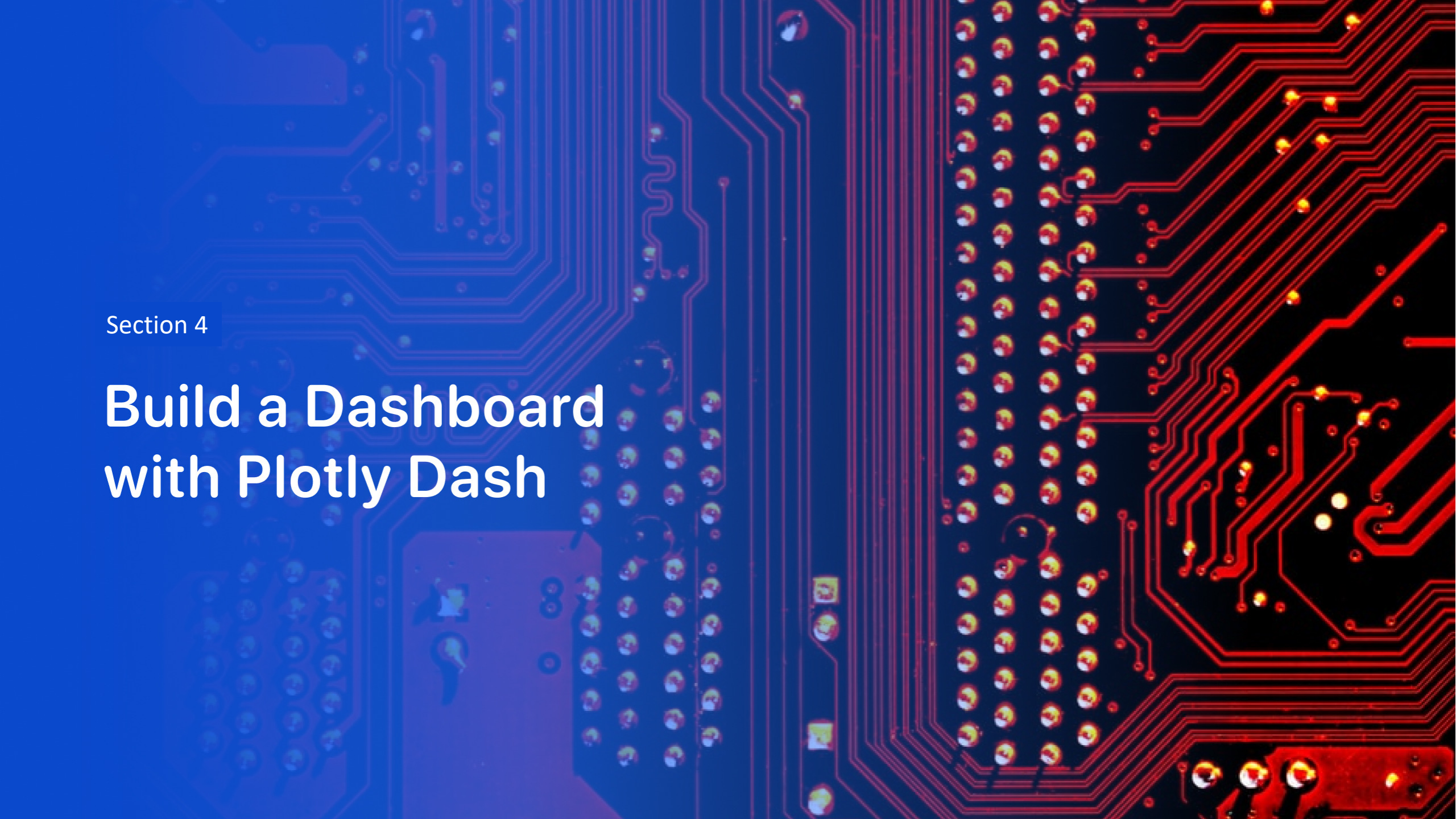


Launch Sites Proximities

- The right map shows that the launch sites are in close proximity to railways, highways and to coastline. The distance can be see in the map.

- This proximity makes ease for transportation of equipment or personnel. Also is relatively far from the cities so that launch failure does not pose a threat.



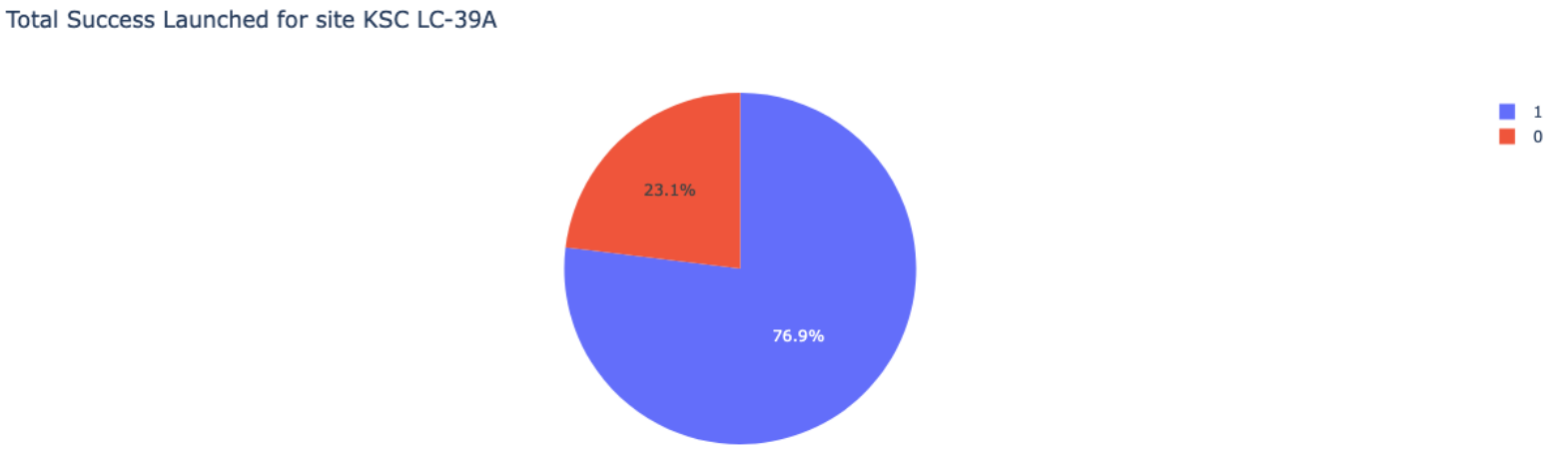

Section 4
Build a Dashboard with Plotly Dash

Total Success Launches by Site

- KSLC-39A has the most launch success among all sites.
- The VAFB SLC-4E has the fewest launch success, possibly because the data sample is small, or because it is the only site located in California, so the launch difficulty on the west coast may be higher than on the east coast.

Highest Launch Success Ratio

- Label 1 means success launch, label 0 means unsuccessful launch
- KSLC-39A has the highest success rate with 10 landing successes (76.9%) and 3 landing failures (23.1%).



Payload X Launch outcome for all sites (scatter plot)

- These scatter plots show that the launch success rate (class 1) for low weight payloads(0-5000 kg) is higher than the heavy weight payloads(5000-10000 kg).

Section 5

Predictive Analysis (Classification)

Classification Accuracy

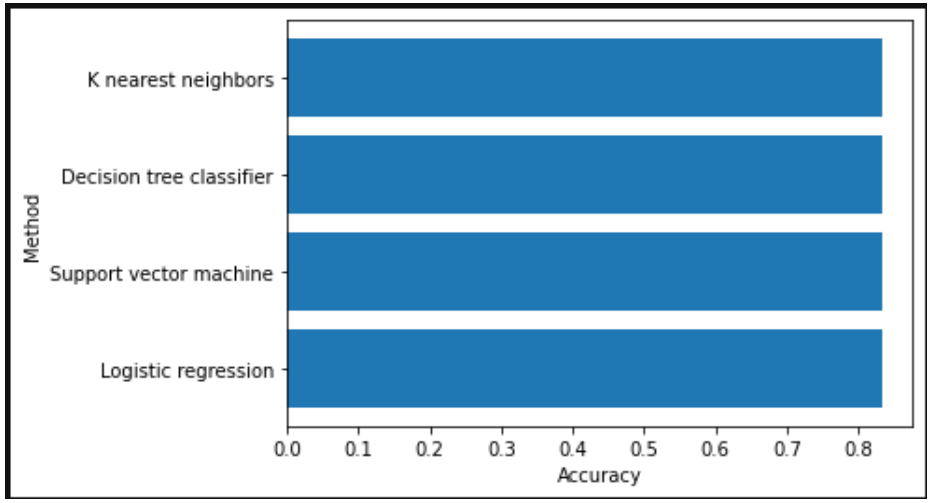
The accuracy achieved of all models, in the test set, was pratically the same at 83.33%.

A small test size was provided at 18 and this may be a problem;

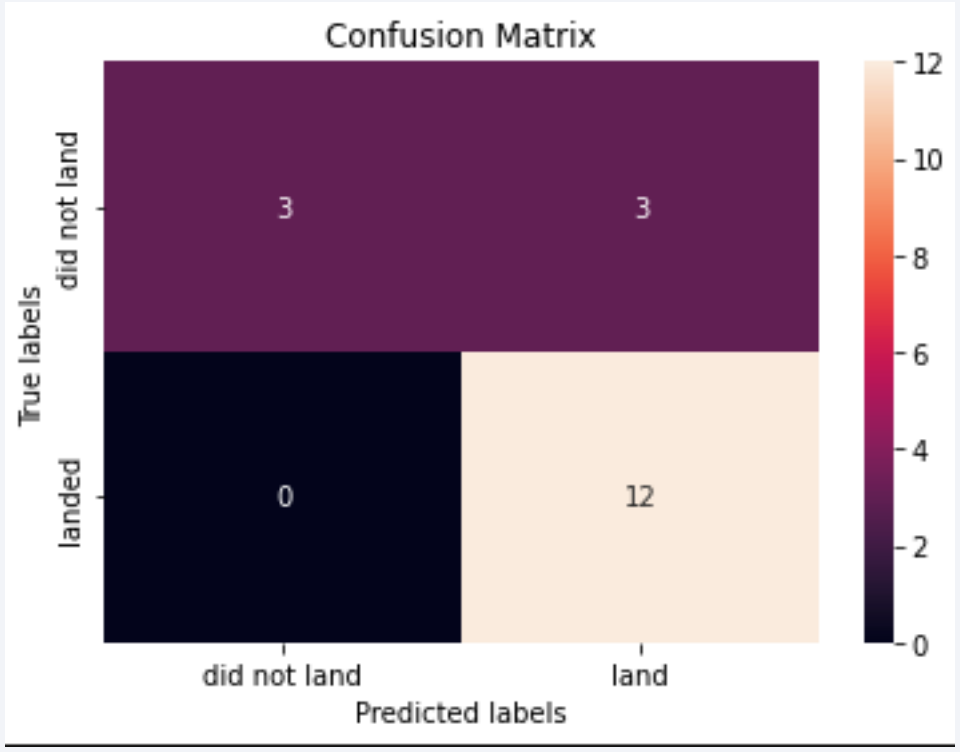
More data is needed to determine an optimal model;

```
d = {'Algorithm': methods, 'Accuracy':accuracy}
df_accuracy = pd.DataFrame(d, columns=['Algorithm','Accuracy'])
df_accuracy
```

	Algorithm	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



- The confusion matrix created was the same for all models. This occurred because the models had the same performance for the same test set;
- These models predicted 12 successful landings when the true label was successful and 3 failed landings when the true label was failure. There were also 3 successful landings predictions when the true label was failure revealing a false positive.

Conclusions

- As the number of flights increase, the success rate increases as well;
- Orbital types SSO, HEO, GEO, and ES-L1 show the highest success rate (100%).
- The launch site is close to railways, highways, and coastline, but with a certain distance 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 weight payloads is higher than that of heavy weight payloads.
- In this dataset, all models have the same accuracy (83.33%), however, more data is needed to determine the optimal model due to the small data size.

Appendix

- Coursera IBM Certification Course;
- Github Repository;

Thank you!