



IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Bruno Lessa>
<06/14/2022>



Outline

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

Executive Summary

Summary of methodologies

- Data Collection through API
- Data Collection with Web Scraping
- Data Wrangling
- Exploratory Data Analysis with SQL
- Exploratory Data Analysis with Data Visualization
- Interactive Visual Analytics with Folium
- Machine Learning Prediction

Summary of all results

- Exploratory Data Analysis result
- Interactive analytics in screenshots
- Predictive analytics result from Machine Learning Lab

Introduction

SpaceX is a revolutionary company who has disrupted the space industry by offering a rocket launch specifically Falcon 9 as low as 62 million dollars; while other providers cost upward of 165 million dollars each. Most of this saving thanks to SpaceX's astounding idea to reuse the first stage of the launch by re-land the rocket to be used on the next mission. Repeating this process will make the price even further down. As a data scientist of a startup rivaling SpaceX, the goal of this project is to create the machine learning pipeline to predict the landing outcome of the first stage in the future. This project is crucial in identifying the right price to bid against SpaceX for a rocket launch.

The problems included:

- Identifying all factors that influence the landing outcome.
- The relationship between each variable and how it is affecting the outcome.
 - The best condition needed to increase the probability of successful landing.
- Identifying all factors that influence the landing outcome.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
 - Data was collected using SpaceX REST API and web scrapping from Wikipedia
- Perform data wrangling
 - Data was processed using one-hot encoding for categorical features
 - Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
 - How to build, tune, evaluate classification models

Data Collection

Data collection is the process of gathering and measuring information on targeted variables in an established system, which then enables one to answer relevant questions and evaluate outcomes. As mentioned, the dataset was collected by REST API and Web Scrapping from Wikipedia

For REST API, its started by using the get request. Then, we decoded the response content as Json and turn it into a pandas dataframe using `json_normalize()`. We then cleaned the data, checked for missing values and fill with whatever needed.

For web scrapping, we will use the BeautifulSoup to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for further analysis

Data Collection – SpaceX API

Get request for rocket launch data using API

Use `json_normalize` method to convert json result to dataframe

Performed data cleaning and filling the missing value

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

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

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

```
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.  
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]
```

```
# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that have multiple payloads in a single rocket.  
data = data[data['cores'].map(len)==1]  
data = data[data['payloads'].map(len)==1]
```

<https://github.com/lessabruno/Data-Science-Capstone/blob/main/Data-Collection.ipynb>

Data Collection - Scraping

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

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data, 'html.parser')
```

```
extracted_row = 0
#Extract each table
for table_number, table in enumerate(soup.find_all('table', "wikitable plainrowheaders collapsible")):
    # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
            else:
                flag=False
```

Request the Falcon9 Launch Wiki page from url

Create a BeautifulSoup from the HTML response

Extract all column/variable names from the HTML header

https://github.com/lessabruno/Data-Science-Capstone/blob/main/Data_Collection_with_Web_Scraping.ipynb

Data Wrangling

Data Wrangling is the process of cleaning and unifying messy and complex data sets for easy access and Exploratory Data Analysis (EDA). We will first calculate the number of launches on each site, then calculate the number and occurrence of mission outcome per orbit type. We then create a landing outcome label from the outcome column. This will make it easier for further analysis, visualization, and ML. Lastly, we will export the result to a CSV

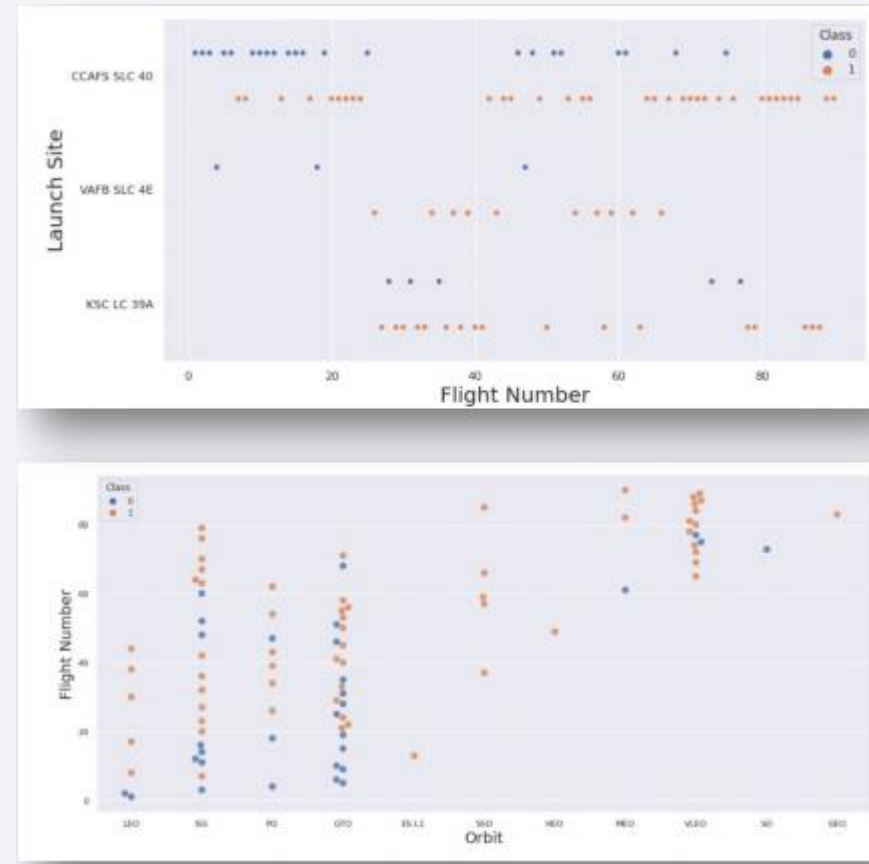
<https://github.com/lessabruno/Data-Science-Capstone/blob/main/DataWrangling.ipynb>

EDA with Data Visualization

We first started by using scatter graph to find the relationship between the attributes such as between:

- Payload and Flight Number.
- Flight Number and Launch Site.
- Payload and Launch Site.
- Flight Number and Orbit Type.
- Payload and Orbit Type.

Scatter plots show dependency of attributes on each other. Once a pattern is determined from the graphs. It's very easy to see which factors affecting the most to the success of the landing outcomes.



<https://github.com/lessabruno/Data-Science-Capstone/blob/main/ExploratoryAnalysisVisualisationLabj.ipynb>

EDA with SQL

- Displaying the names of the launch sites.
- Displaying 5 records where launch sites begin with the string 'CCA'. –
- Displaying the total payload mass carried by booster launched by NASA (CRS). –
- Displaying the 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 sites names for in year 2015.
- - Rank the count of landing outcomes or success between the date 2010-06-04 and 2017-03-20, in descending order.

<https://github.com/lessabruno/Data-Science-Capstone/blob/main/ExploratoryDataAnalysisSQL.ipynb>

Build an Interactive Map with Folium

To visualize the launch data into an interactive map. We took the latitude and longitude coordinates at each launch site and added a circle marker around each launch site with a label of the name of the launch site.

We then assigned the dataframe `launch_outcomes(failure,success)` to classes 0 and 1 with Red and Green markers on the map in `MarkerCluster()`.

We then used the Haversine's formula to calculate the distance of the launch sites to various landmarks to find answers to the questions of:

- How close the launch sites with railways, highways and coastlines?
- How close the launch sites with nearby cities?

<https://github.com/lessabruno/Data-Science-Capstone/blob/main/InteractiveAnalyticsFolium.ipynb>

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash which allowing the user to play around with the data as they need.
- We plotted pie charts showing the total launches by a certain sites.
- We then plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

<https://github.com/lessabruno/Data-Science-Capstone/blob/main/spaceYDashApp.py>

Predictive Analysis (Classification)

Build Model

- Load Dataset
- Transform data
- Decide which MK use

Evaluate Model

- Check accuracy
- Tune parameters
- Plot confusion matrix

Improve Model

- Use Feature Engineering

Find the best model

- Best accuracy will be best model

Results

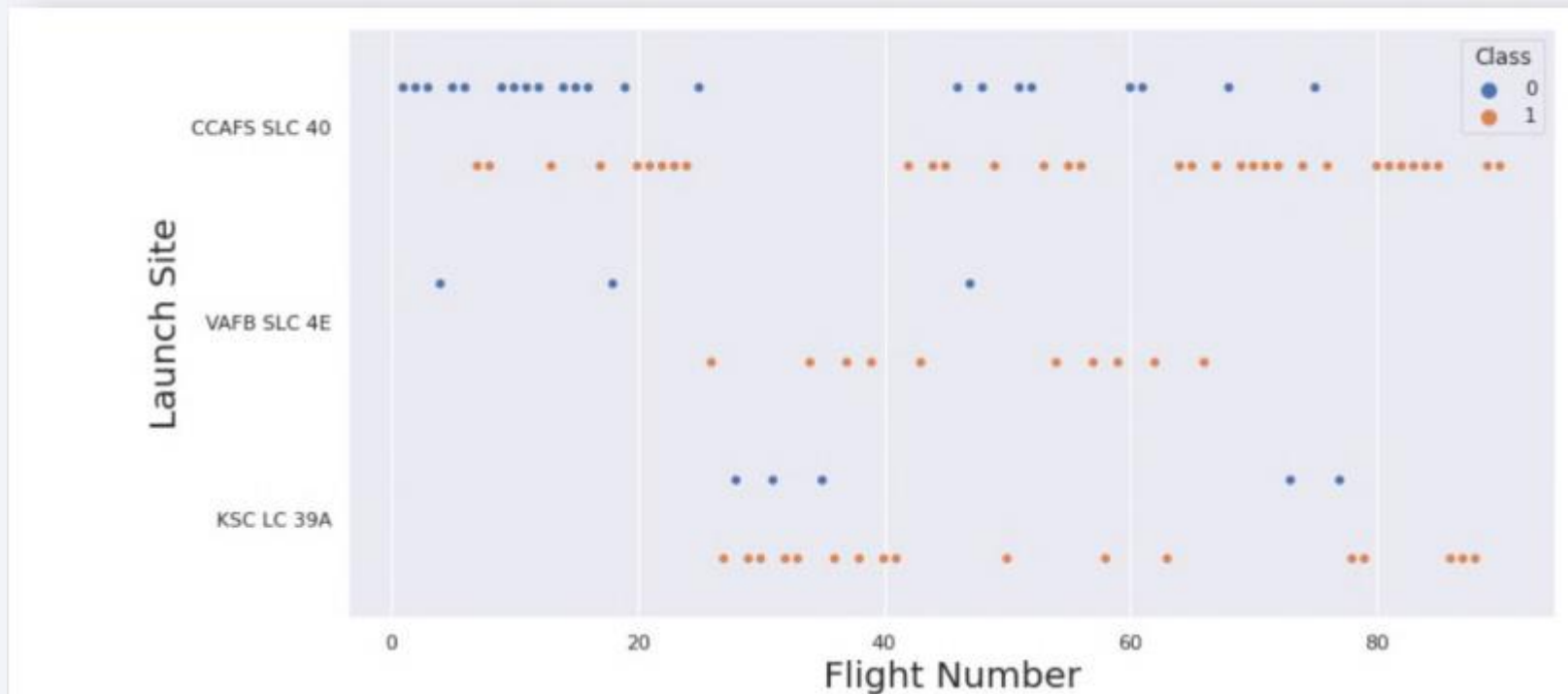
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results



Section 2

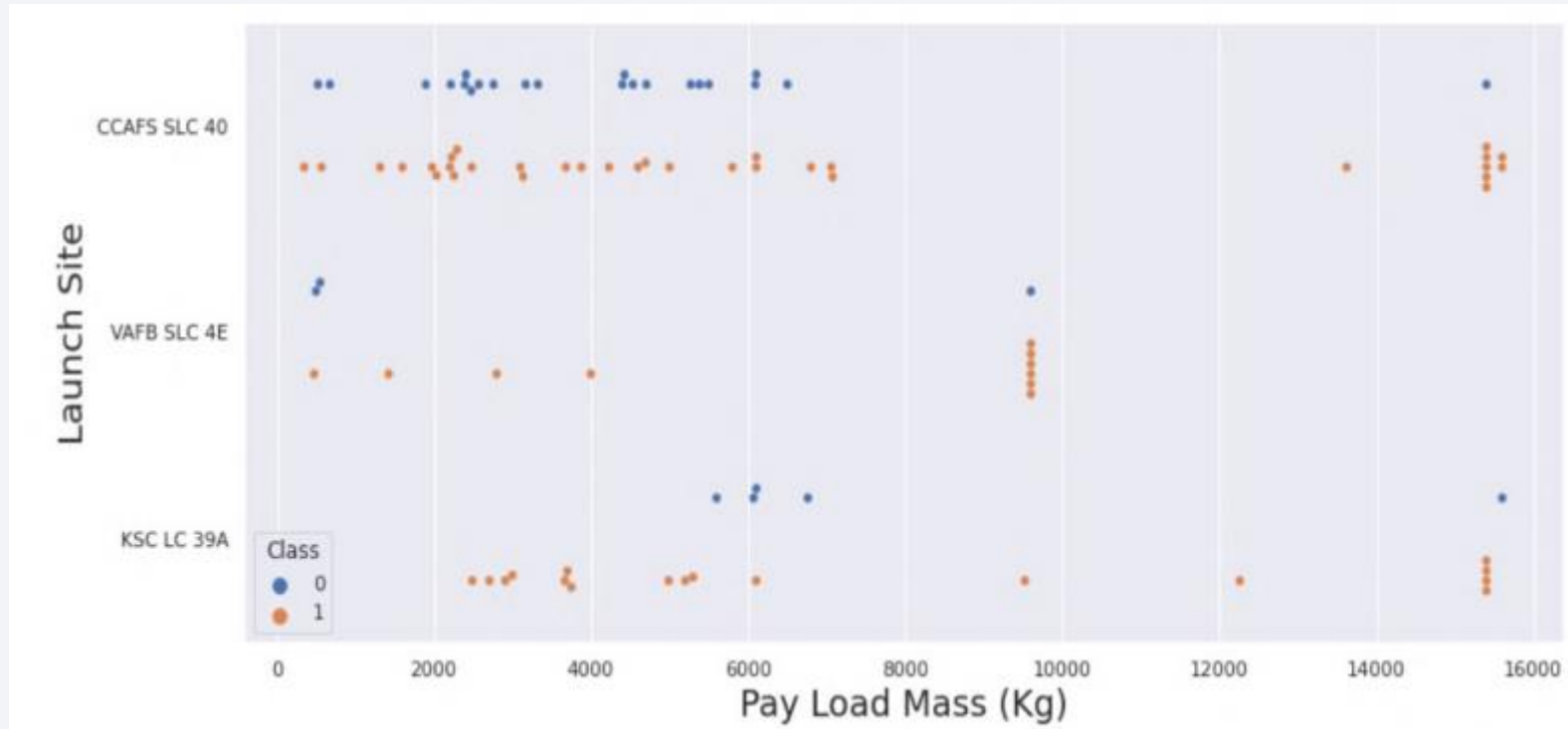
Insights drawn from EDA

Flight Number vs. Launch Site



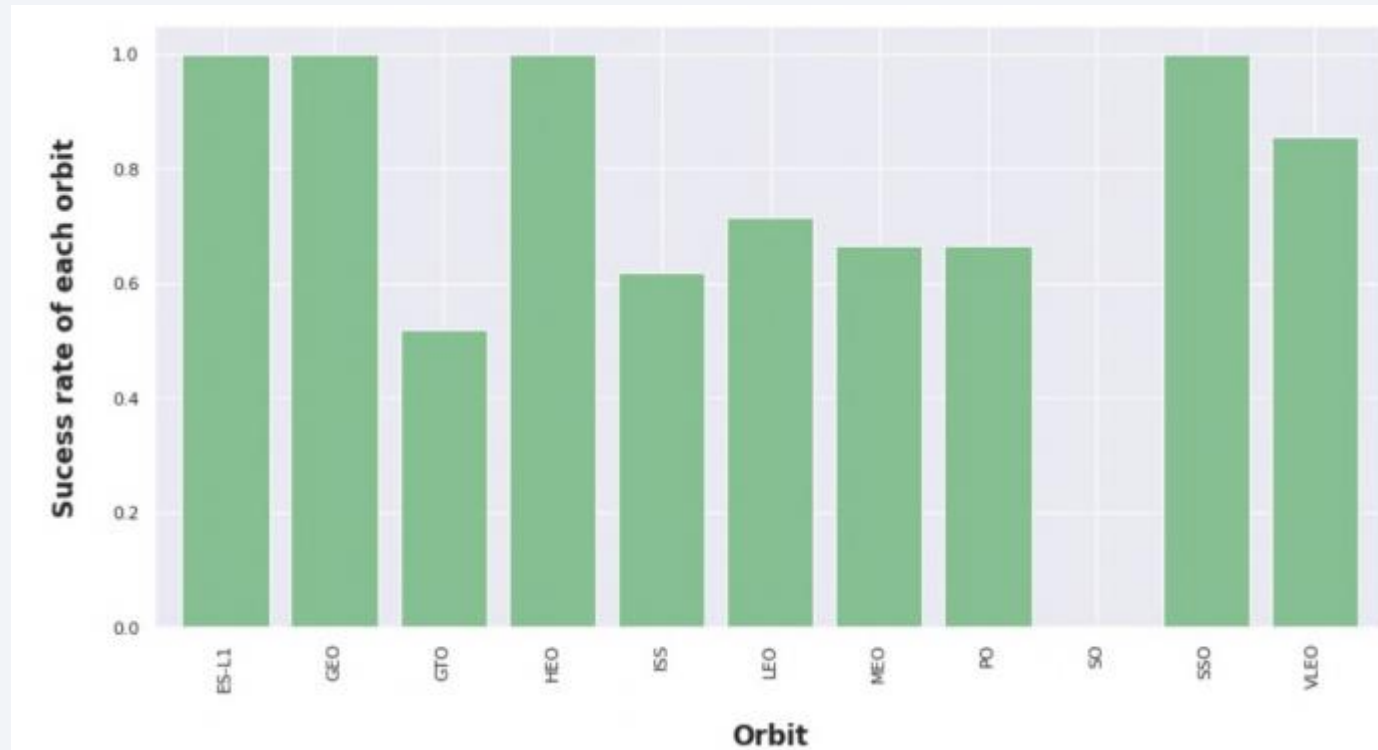
the larger the flights amount of the launch site, the greater the the success rate will be.

Payload vs. Launch Site

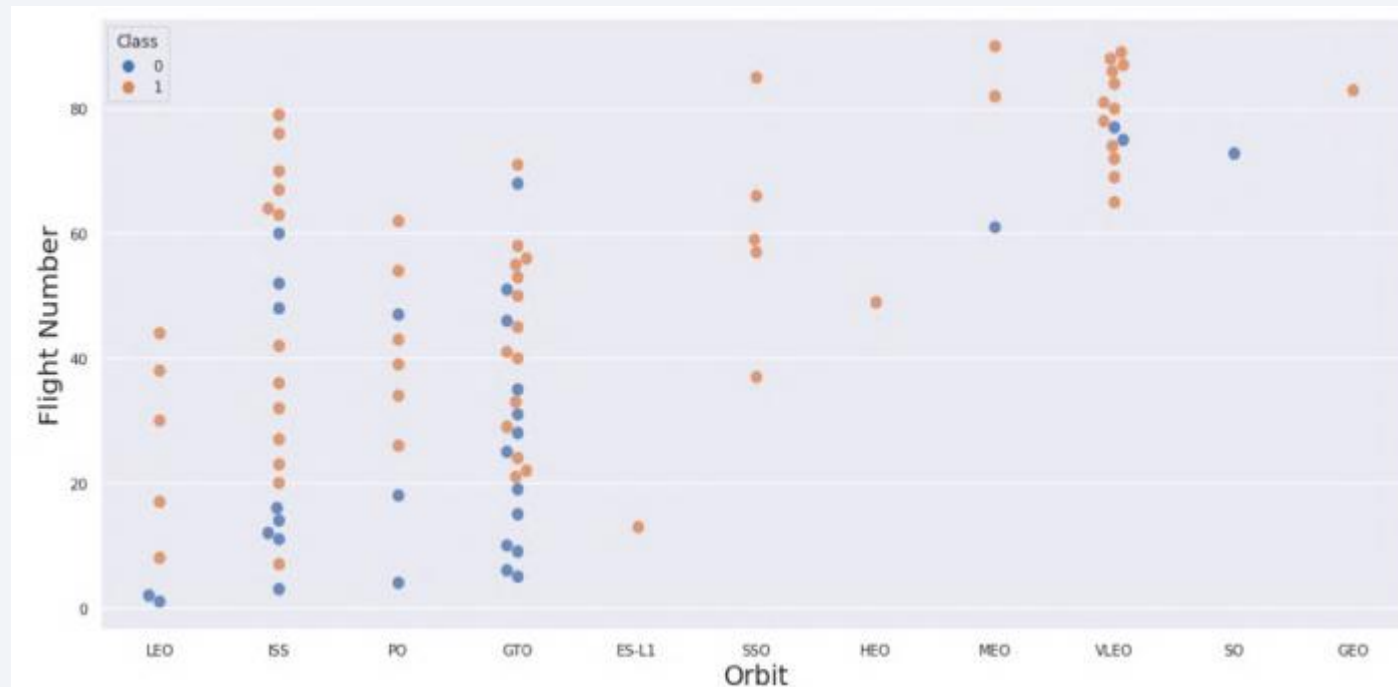


once the pay load mass is greater than 7000kg, the probability of the success rate will be highly increased.

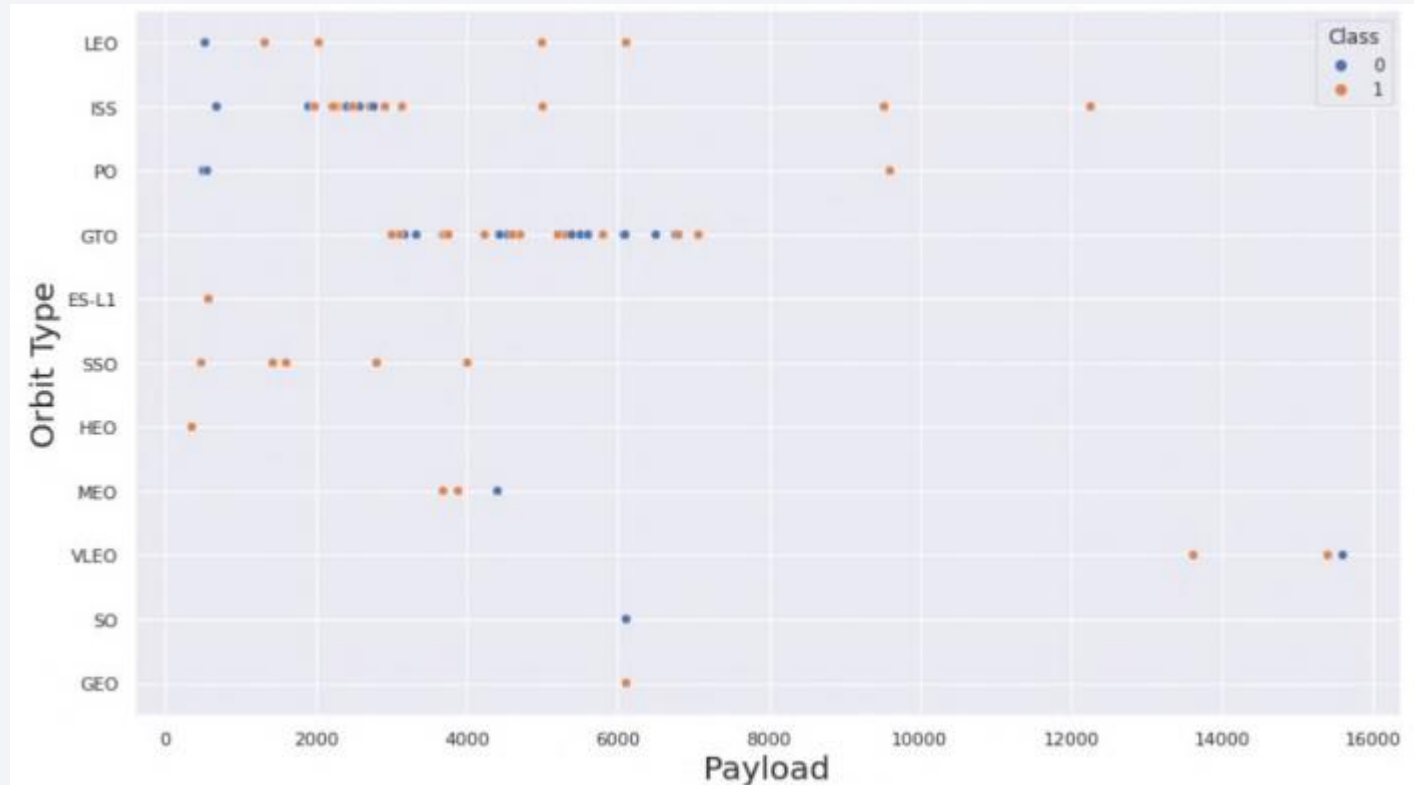
Success Rate vs. Orbit Type



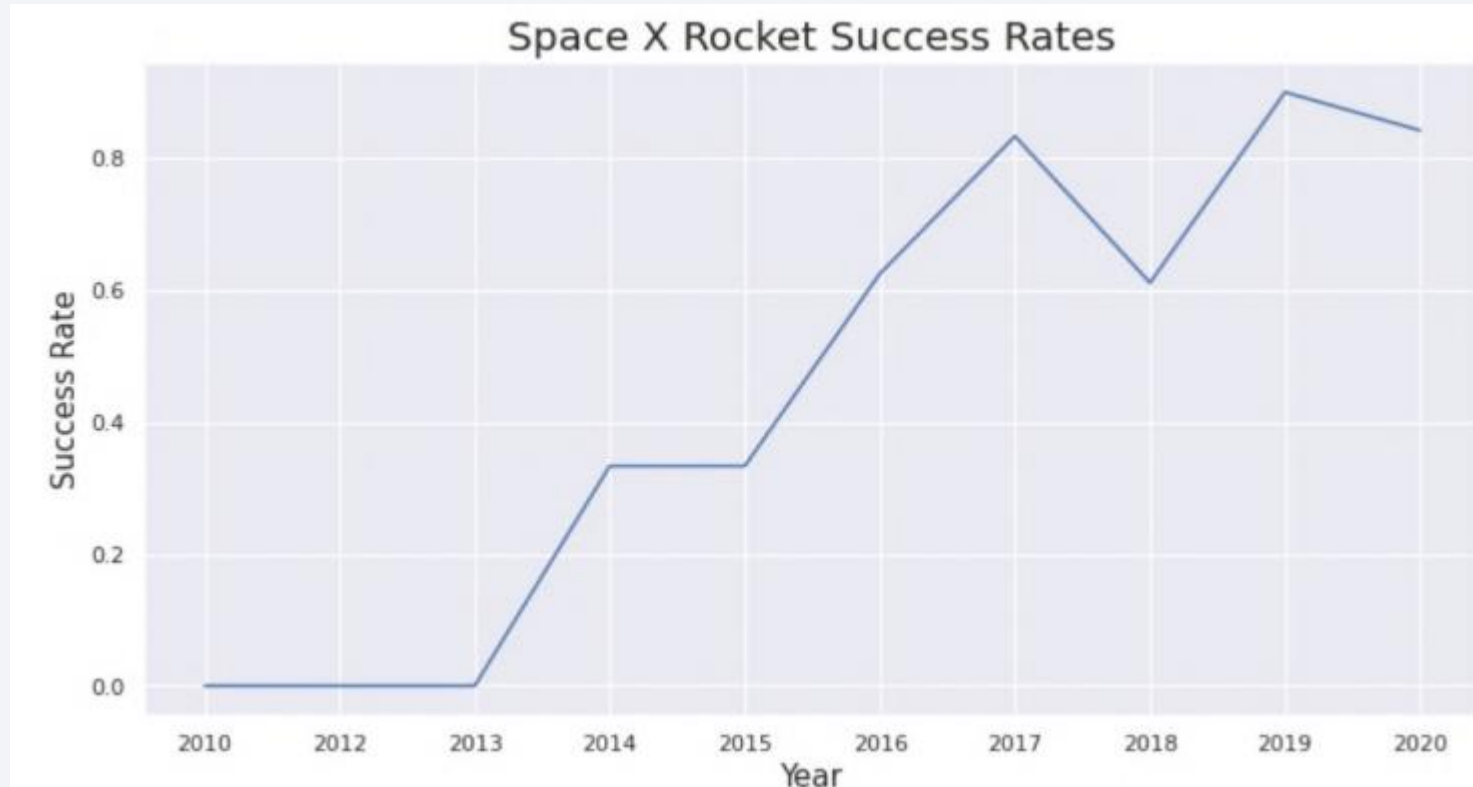
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

SELECT DISTINCT , to show us unique launch sites.

```
Out[5]: Launch_Sites
```

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

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

Total Payload Mass

Used SUM(PAYLOAD_MASS_KG) , to show the total Mass carried.

Total Payload Mass by NASA (CRS)
45596

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass by Booster  
WHERE BOOSTER_VERSION = 'F9 v1.1';
```

Average Payload Mass by Booster Version F9 v1.1

2928

First Successful Ground Landing Date

```
%sql SELECT MIN(DATE) AS "First Successful Landing Outcome in Ground Pad"  
WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

First Successful Landing Outcome in Ground Pad
--

2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT BOOSTER_VERSION FROM SPACEX WHERE LANDING__OUTCOME = 'Success (drone ship)' \
AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

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

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Successful Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Success%';
```

Successful Mission
100

```
%sql SELECT COUNT(MISSION_OUTCOME) AS "Failure Mission" FROM SPACEX WHERE MISSION_OUTCOME LIKE 'Failure%';
```

Failure Mission
1

Boosters Carried Maximum Payload

```
❖sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEX
WHERE PAYLOAD_MASS_KG_ =(SELECT MAX(PAYLOAD_MASS_KG_) FROM SPACEX);
```

Booster Versions which carried the Maximum Payload Mass
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX WHERE DATE LIKE '2015-%' AND \
LANDING__OUTCOME = 'Failure (drone ship)';
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

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

```
%sql SELECT LANDING__OUTCOME as "Landing Outcome", COUNT(LANDING__OUTCOME) AS "Total Count" FROM SPACEX \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY LANDING__OUTCOME \
ORDER BY COUNT(LANDING__OUTCOME) DESC ;
```

Landing Outcome	Total Count
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

A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The background is a deep blue gradient.

Section 3

Launch Sites Proximities Analysis

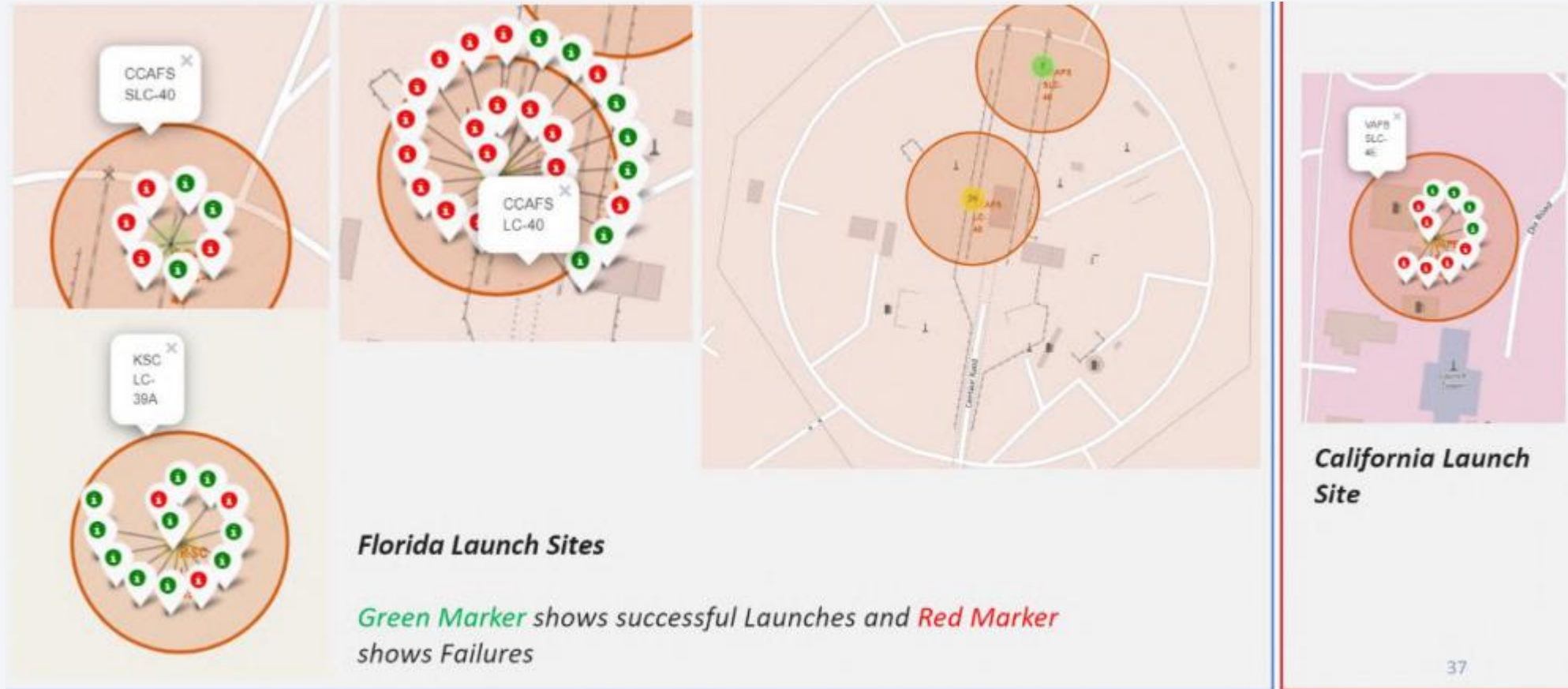
<Folium Map Screenshot 1>

- Replace <Folium map screenshot 1> title with an appropriate title
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
- Explain the important elements and findings on the screenshot

Launch Sites



<Launch Sites with colors labels>

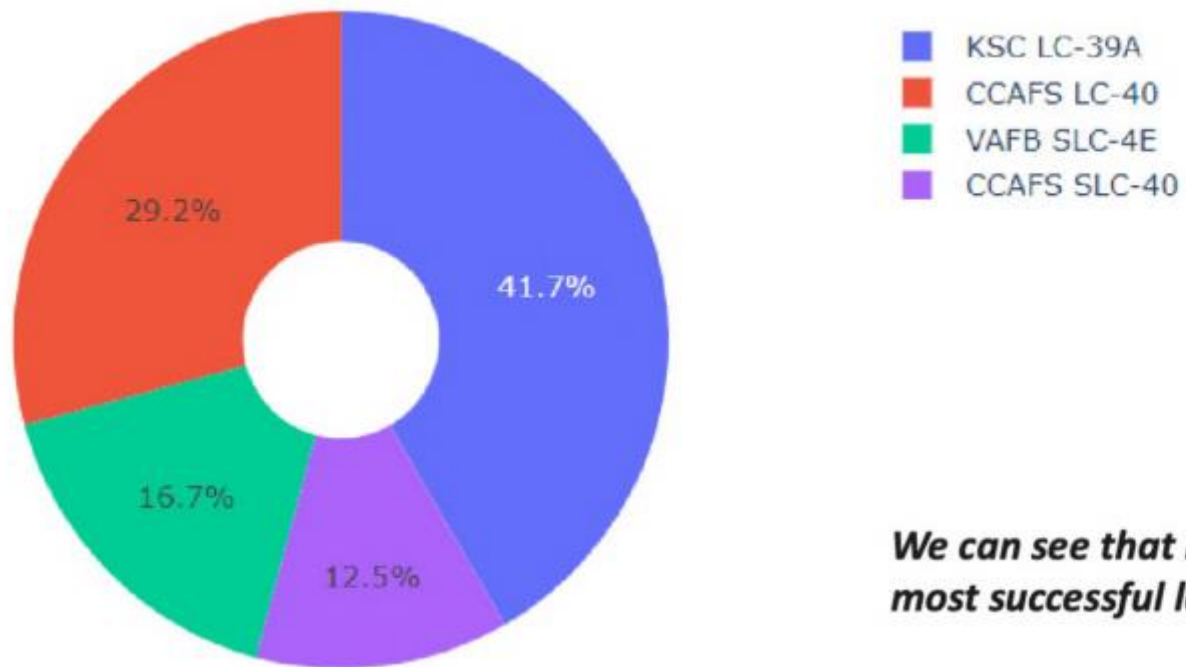




Section 4

Build a Dashboard with Plotly Dash

Launch success by site

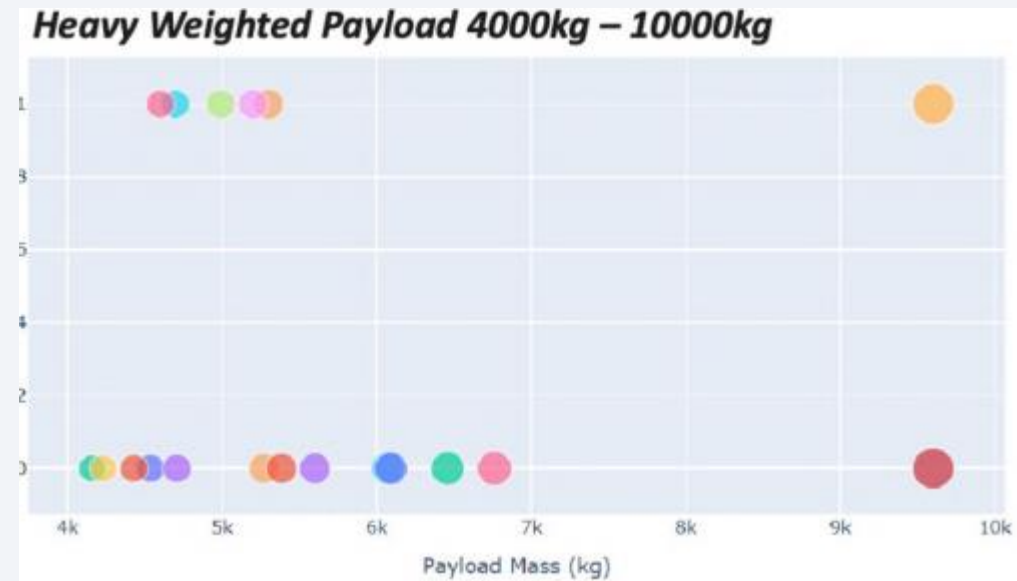
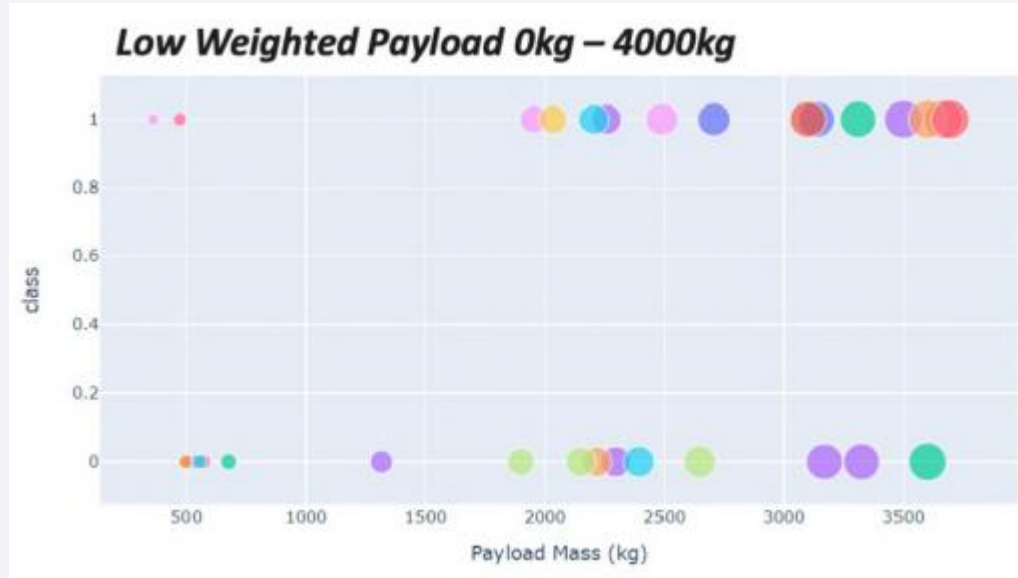


We can see that KSC LC-39A had the most successful launches from all the sites

<Dashboard Screenshot 2>

- Replace <Dashboard screenshot 2> title with an appropriate title
- Show the screenshot of the piechart for the launch site with highest launch success ratio
- Explain the important elements and findings on the screenshot

Payload vs Launch Outcome





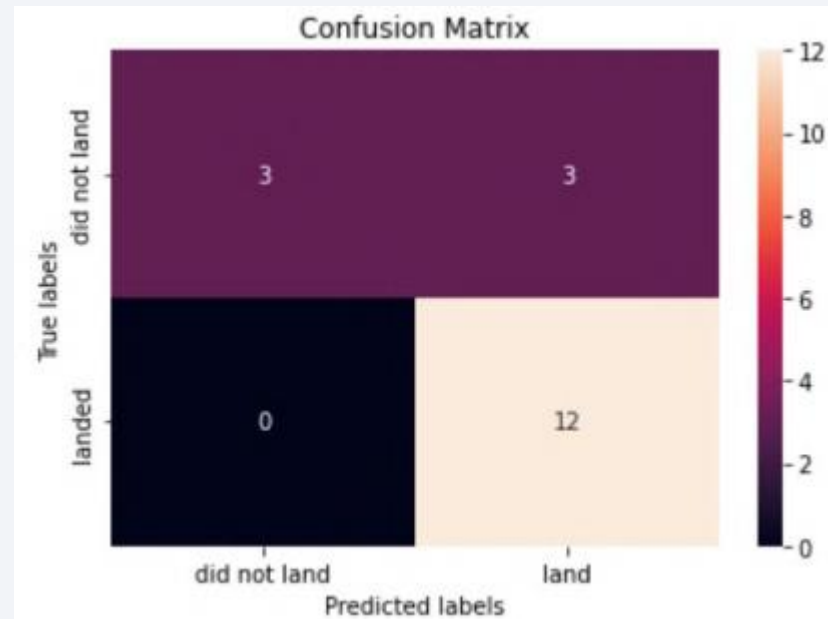
Section 5

Predictive Analysis (Classification)

Classification Accuracy

```
Best Algorithm is Tree with a score of 0.9017857142857142  
Best Params is : {'criterion': 'entropy', 'max_depth': 10, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 10, 'splitter': 'random'}
```

Confusion Matrix



Conclusions

- The Tree Classifier Algorithm is the best Machine Learning approach for this dataset.
- The low weighted payloads (which define as 4000kg and below) performed better than the heavy weighted payloads.
- Starting from the year 2013, the success rate for SpaceX launches is increased, directly proportional time in years to 2020, which it will eventually perfect the launches in the future.
- KSC LC-39A have the most successful launches of any sites; 76.9%
- SSO orbit have the most success rate; 100% and more than 1 occurrence.

Thank you!

