IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Zach Chen
March 7th 2023

Outline

Executive Summary

- Summary of methodologies
  - Data Collection: Data Collection API, Web-scraping
  - Exploratory Data Analysis (EDA): SQL, Data Visualization
  - Interactive Map: Follium
  - Dash-borad: Plotly Dash
  - Predictive Analysis: Python Libraires
- Summary of all results
  - Exploratory Data Analysis Results
  - Interactive Display Results
  - Predictive Analysis Results

Introduction

- Project background and context
  - The commercial space age is here, companies are making space travel affordable for everyone. Among all the pioneers, perhaps the most successful is SpaceX. SpaceX's accomplishments include: Sending spacecraft to the International Space Station. Starlink, a satellite internet constellation providing satellite Internet access. Sending manned missions to Space. One reason SpaceX can do this is the rocket launches are relatively inexpensive. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards 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.
- Problems to be investigated
  - What are the key variables in affecting the results of a launch?
  - Given the data collected, build a model to predict the results of future launches and review the reliability of the model.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - SpaceX REST API & Web-Scraping from Wikipedia
- Perform data wrangling
    - Clean null values and covert several key features with One-Hot Encoding
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - Apply standardized data with predictive models such as, logistic regression, support vector machine, decision tree classifier, and k nearest neighbors.
    - Investigate the accuracy of method mentioned above.

Data Collection

- The flow charts below shows the processes of data collection via API and Web Scraping
- Data Collection API (SpaceX REST API)

Using SpaceX Rest API → Return Data in JSON format → Review the raw data and convert into a data frame → Clean, filter modify the data frame → Export to CVS

- Web scraping from Wikipeida

Connecting to Wikipedia HTML → Using Python Library Beautifulsou p to scrape data → Review the raw data and convert into a data frame → Clean, filter modify the data frame → Export to CVS

Data Collection – SpaceX API

- Data collection with SpaceX REST calls phrases and flowcharts

Now let's start requesting rocket launch data from SpaceX API with the following URL:

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

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

1. Call SpaceX REST API

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

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

2. Retrieve data as JSON file and Normalize the data

Finally we will remove the Falcon 1 launches keeping only the Falcon 9 launches. Filter the data dataframe using the `BoosterVersion` column to only keep the Falcon 9 launches. Save the filtered data to a new dataframe called `data_falcon9`.

```
# Hint data['BoosterVersion']!='Falcon 1'
df.dropna(inplace=True)
data_falcon9 = df[df['BoosterVersion']!='Falcon 1']
data_falcon9.head(5)
```

Python

3. Filter and Clean the data

Calculate below the mean for the `PayloadMass` using the `.mean()`. Then use the mean and the `.replace()` function to replace `np.nan` values in the data with the mean you calculated.

```
# Calculate the mean value of PayloadMass column

# Replace the np.nan values with its mean value
data_falcon9.replace((data_falcon9['PayloadMass'].mean()), np.nan, inplace = True)
# data_falcon9.head(30)
```

Python

4. Dealing the missing value and review the data frame

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

Python

5. Export the data frame into CSV

Code Link https://github.com/ZachJHChen/IBM_Applied_Data_Science_Capstone/blob/a31758b03dce20e6aaac42dd21a9fef6680174dc/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- Web scraping process phrases and flowcharts

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

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

1. Request response from Wikipedia

Create a BeautifulSoup object from the HTML response

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

2. Using html parser from BeautifulSoup to collect data

Let's try to find all tables on the wiki page first. If you need to refresh your memory about BeautifulSoup, please check the external reference link towards the end of this lab

+ Code + Markdown

```
# Use the find_all function in the BeautifulSoup object, with element ty
# Assign the result to a list called `html_tables`
html_tables = soup.find_all("table")
# print(html_tables)
```

3. Search "Tables" in the website to find the specific information needed

We will create an empty dictionary with keys from the extracted column names in the previous task. Later, this dictionary will be converted into a Pandas dataframe

```
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']=[]
```

Python

4. Create a data frame by parsing the launch HTML tables

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

5. Export the data frame into CSV

Code Link

https://github.com/ZachJHChen/IBM_Applied_Data_Science_Capstone/blob/a31758b03dce20e6aaac42dd21a9fef6680174dc/jupyter-labs-webscraping.ipynb

Data Wrangling

- Data Wrangling phrases and flowcharts

Data Analysis

Load Space X dataset, from last section.

```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv")
df.head(10)
```

1.Load data from previous data collection and review

Identify which columns are numerical and categorical:

```
df.dtypes
```

```
df.isnull().sum()/df.count()*100
```

2. Identify the missing data and formats

```
# Apply value_counts() on column LaunchSite
df.value_counts("LaunchSite")
```

```
# Apply value_counts on Orbit column
df.value_counts("Orbit")
```

```
# landing_outcomes = values on Outcome column
landing_outcomes=df.value_counts("Outcome")
landing_outcomes
```

3. Calculate the Launch site, mission outcome per orbit type

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []
for i in df['Outcome']:
    if i in set(bad_outcomes):
        landing_class.append(0)
    else:
        landing_class.append(1)
```

4. Create a landing outcome label from Outcome column

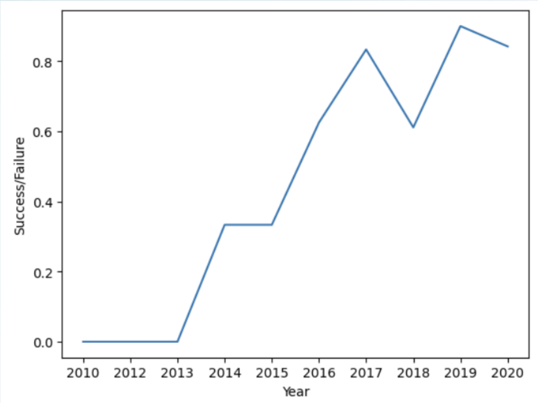
```
df.to_csv("dataset_part_2.csv", index=False)
```

5. Export the data frame into CSV

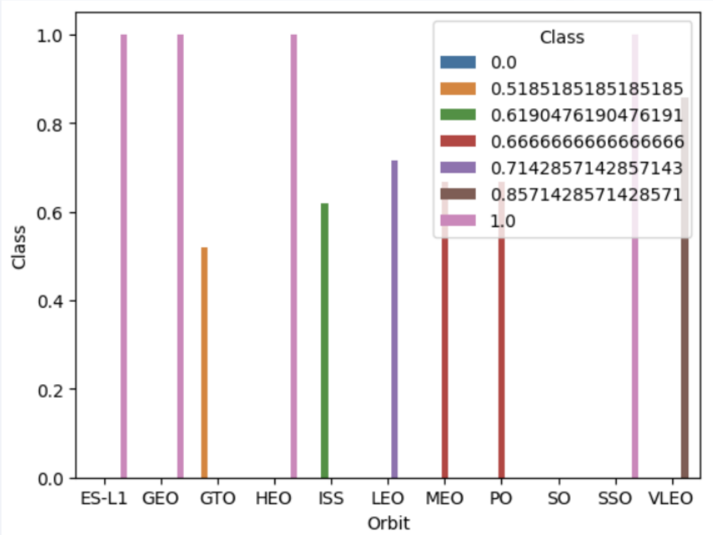
Code Link:
https://github.com/ZachJHChen/IBM_Applied_Data_Science_Capstone/blob/a31758b03dce20e6aaac42dd21a9fef6680174dc/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Scatter point chart (Refer to the link)
  - Flight Number VS Pay Load Mass (Kg)
  - Flight Number VS Launch Site
  - Pay Load Mass (Kg) VS Launch Site
  - Flight Number VS Orbit
  - Pay Load Mass (Kg) VS Orbit
- launch success yearly trend



- Bar chart



relationship between success rate of each orbit type

Code Link:
https://github.com/ZachJHChen/IBM_Applied_Data_Science_Capstone/blob/a31758b03dce20e6aaac42dd21a9fef6680174dc/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

The report conducted SQL from the Data with the following requests

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA’
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was acheived.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the   names of the booster_versions which have carried the maximum payload mass. Use a subquery
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Rank 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

Please refer to the link for detail results

https://github.com/ZachJHChen/IBM_Applied_Data_Science_Capstone/blob/a31758b03dce20e6aaac42dd21a9fef6680174dc/jupyter-labs-eda-sql-coursera.ipynb

Build an Interactive Map with Folium

The report built an interactive map with Folium

- The map uses circles to show key locations such as launch sites and NASA
- The map use colored markers showing success launches in Green and not success launches in Red at each launch sites.
- The map draw lines to show the distance (in KM) from launch site to coast line, nearest cities, closet railways.
- Adding those markers allow viewers easily see the geographic relationships of the results and intuitively understand key information for further investigations.

Please refer to the code link for more detail.

Code Link:
https://github.com/ZachJHChen/IBM_Applied_Data_Science_Capstone/blob/a31758b03dce20e6aaac42dd21a9fef6680174dc/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

The report built a Dashbaord with Plotly Dash

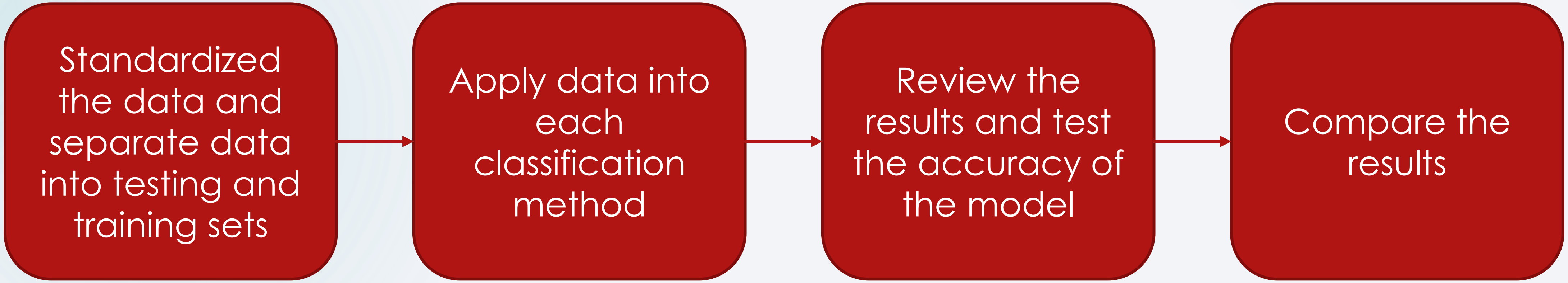
- From this Dashboard, users can choose the launch site from a dropdown list, change the Payload range (Kg) from a range slider whereas the dashboard will change accordingly.
- The dash board shows the success rate of each site with a pie chart and each launch record with payload mass, booster type and outcome with a scatter plot.

Code Link:
https://github.com/ZachJHChen/IBM_Applied_Data_Science_Capstone/blob/a31758b03dce20e6aaac42dd21a9fef6680174dc/spacex_dash_app.ipynb

Predictive Analysis (Classification)

The report built a classification models

- 4 type of classification models were built, Logistic Regression Model, Support Vector Machine, Decision Tree Classifier, K Nearest Neighbors.
- The accuracy of the 4 type were tested and compared.
- Flow chart and key processes



Code Link:
https://github.com/ZachJHChen/IBM_Applied_Data_Science_Capstone/blob/a31758b03dce20e6aaac42dd21a9fef6680174dc/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

The results can be drawn into 3 main categories

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

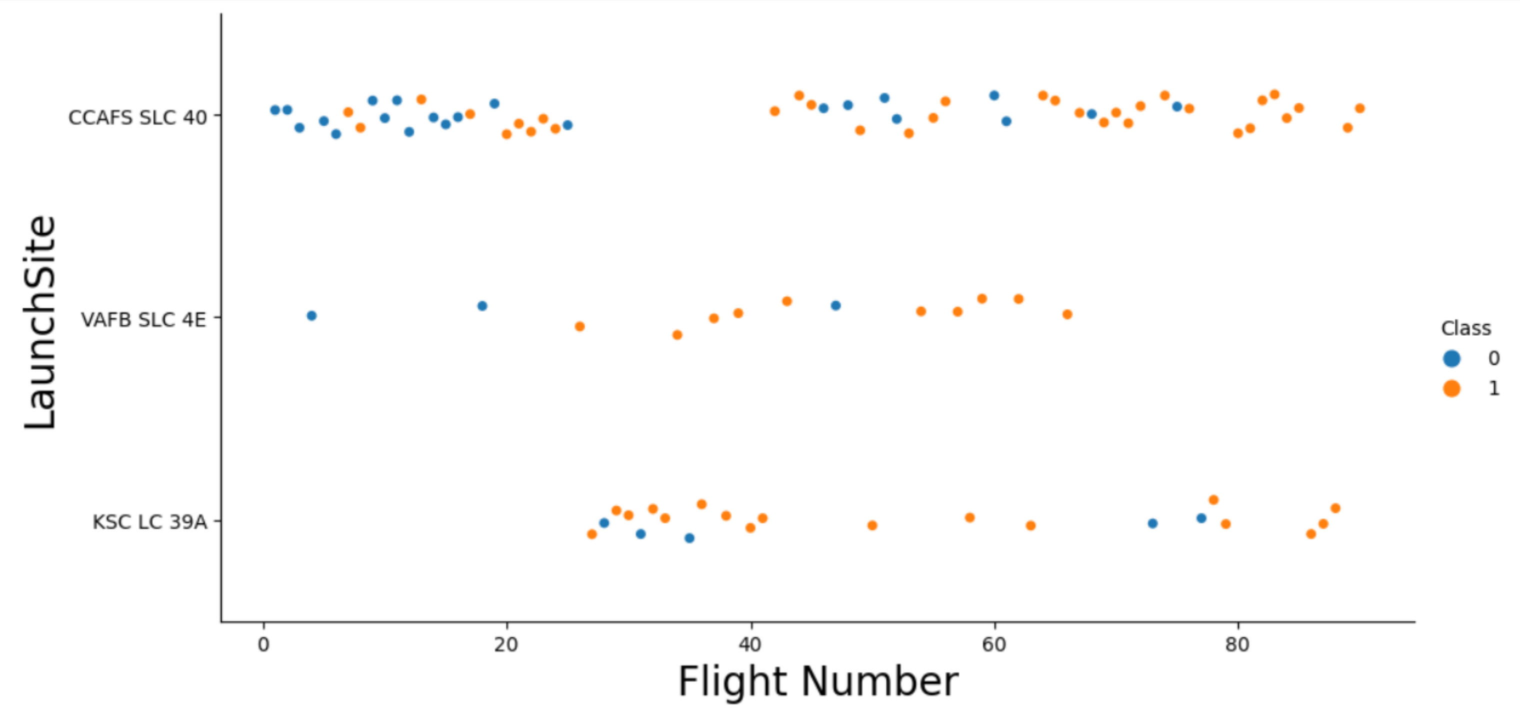
Next part of the report will deliver key finding from those results.

For details, please refer to code link provided in the previous slides.

Section 2

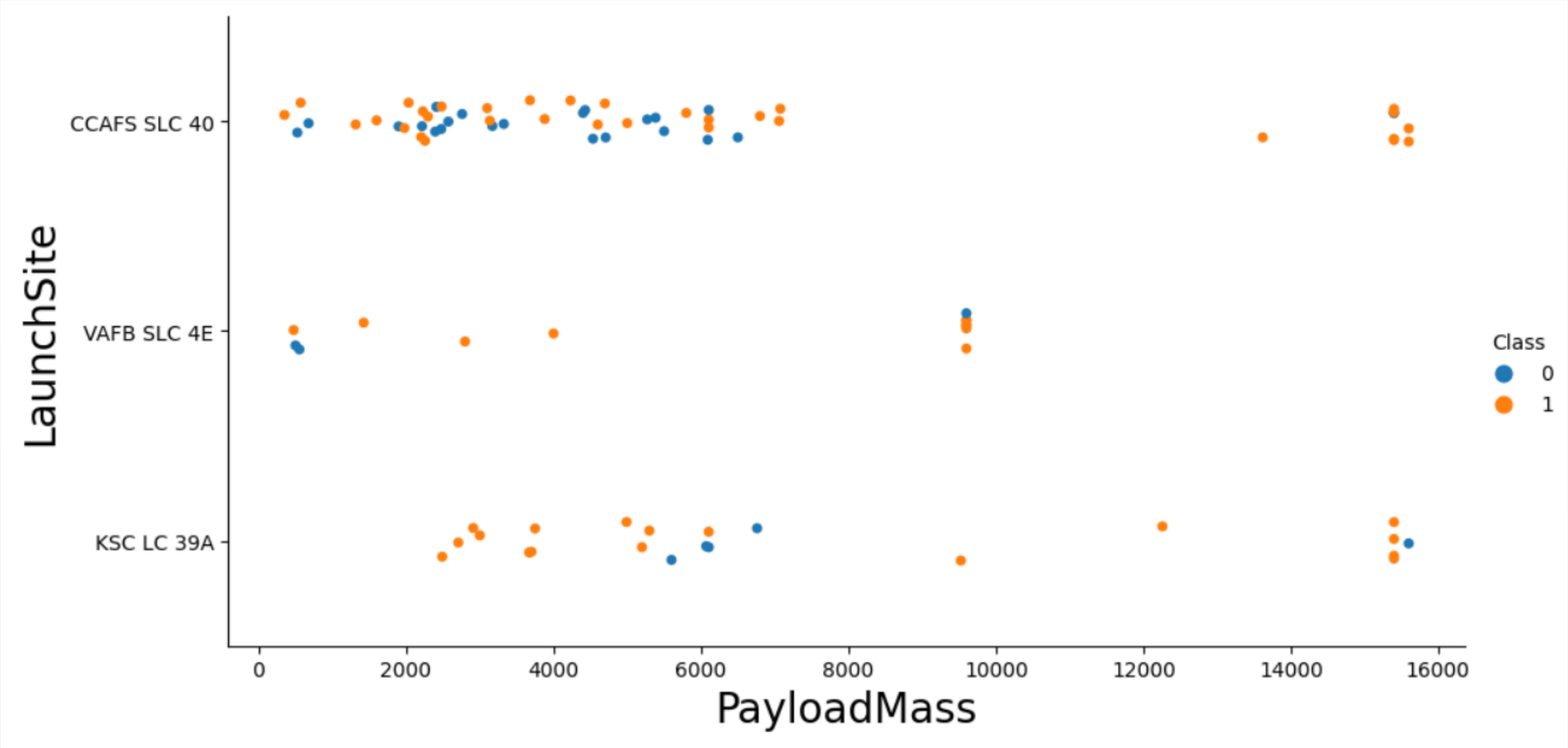
Insights drawn from EDA

Flight Number vs. Launch Site



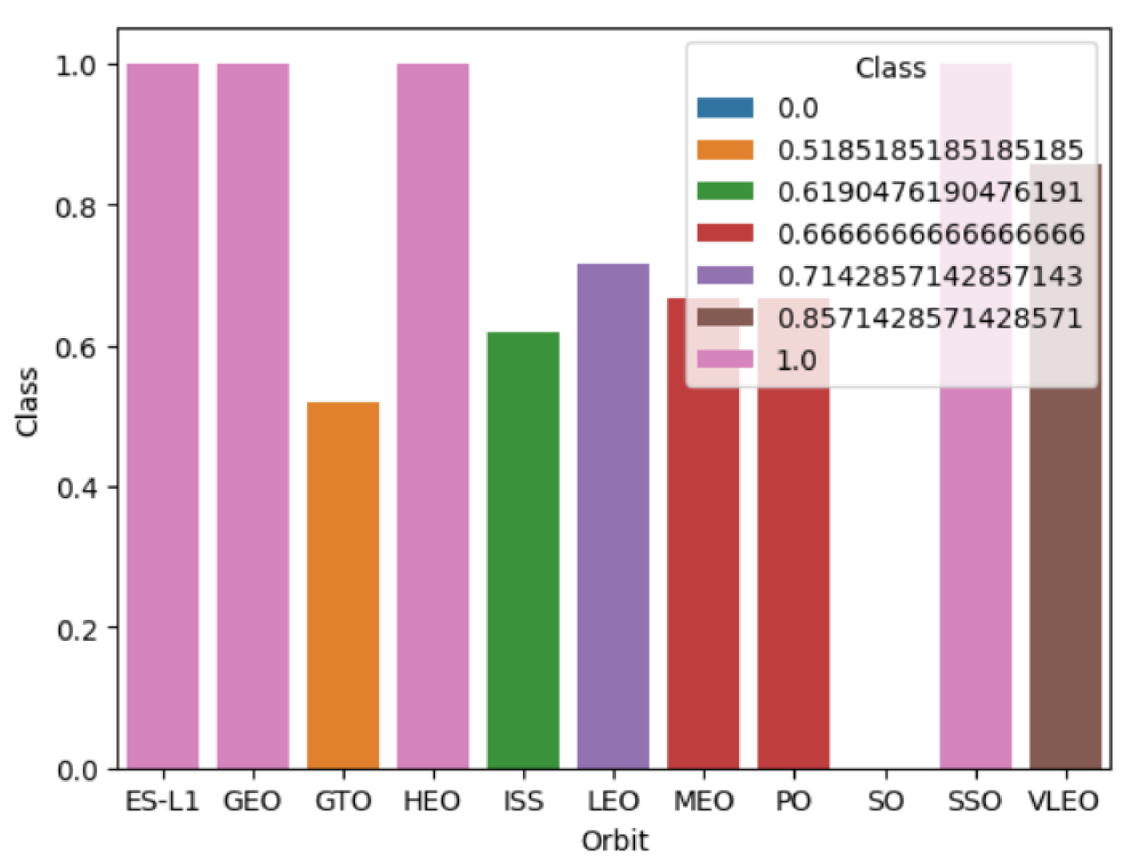
- As shown in the scatter plot, in general as flight number increases, success rate also increases, indicating that the technology was maturing.

Payload vs. Launch Site



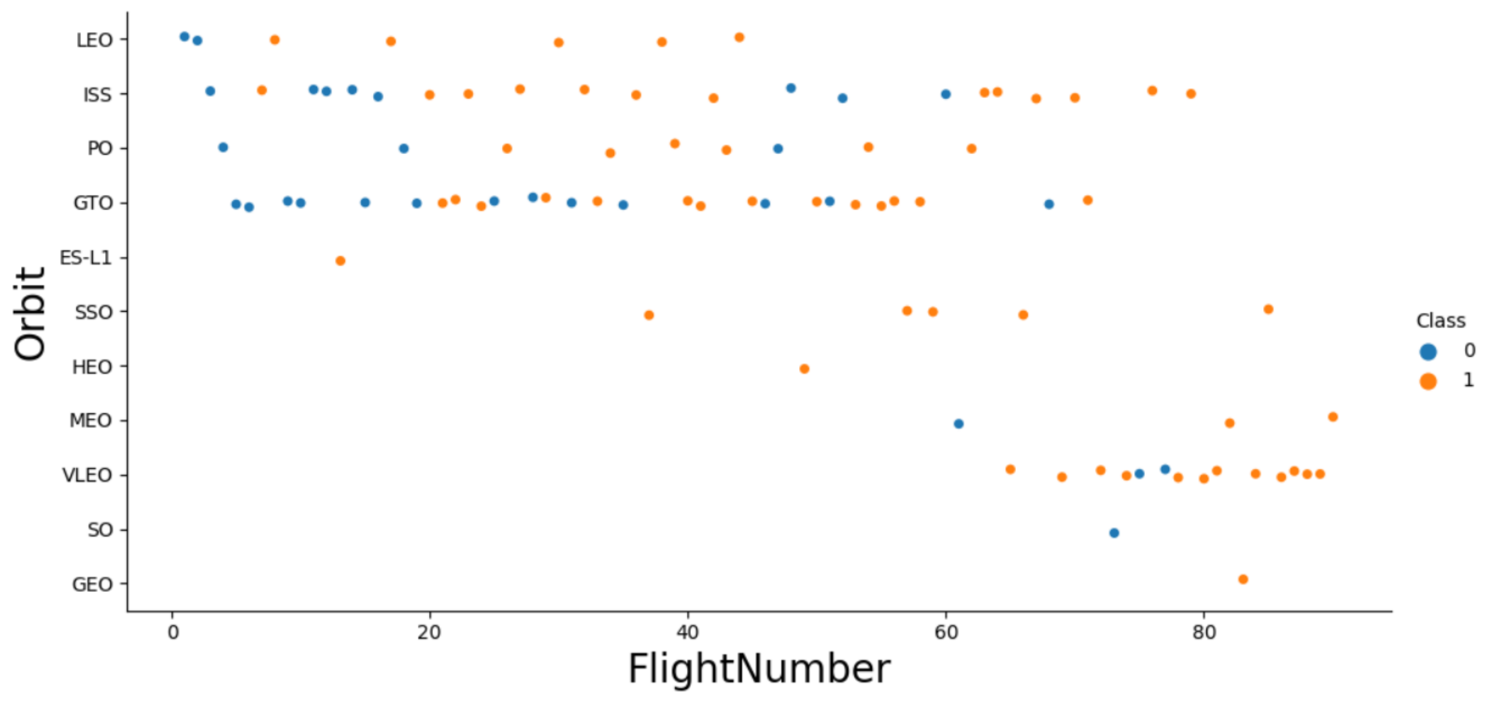
- As shown in the scatter plot, counterintuitively, a launch with heavier payload mass resulted higher success rate.
- More parameters needed to be investigated to gather insights.

Success Rate vs. Orbit Type



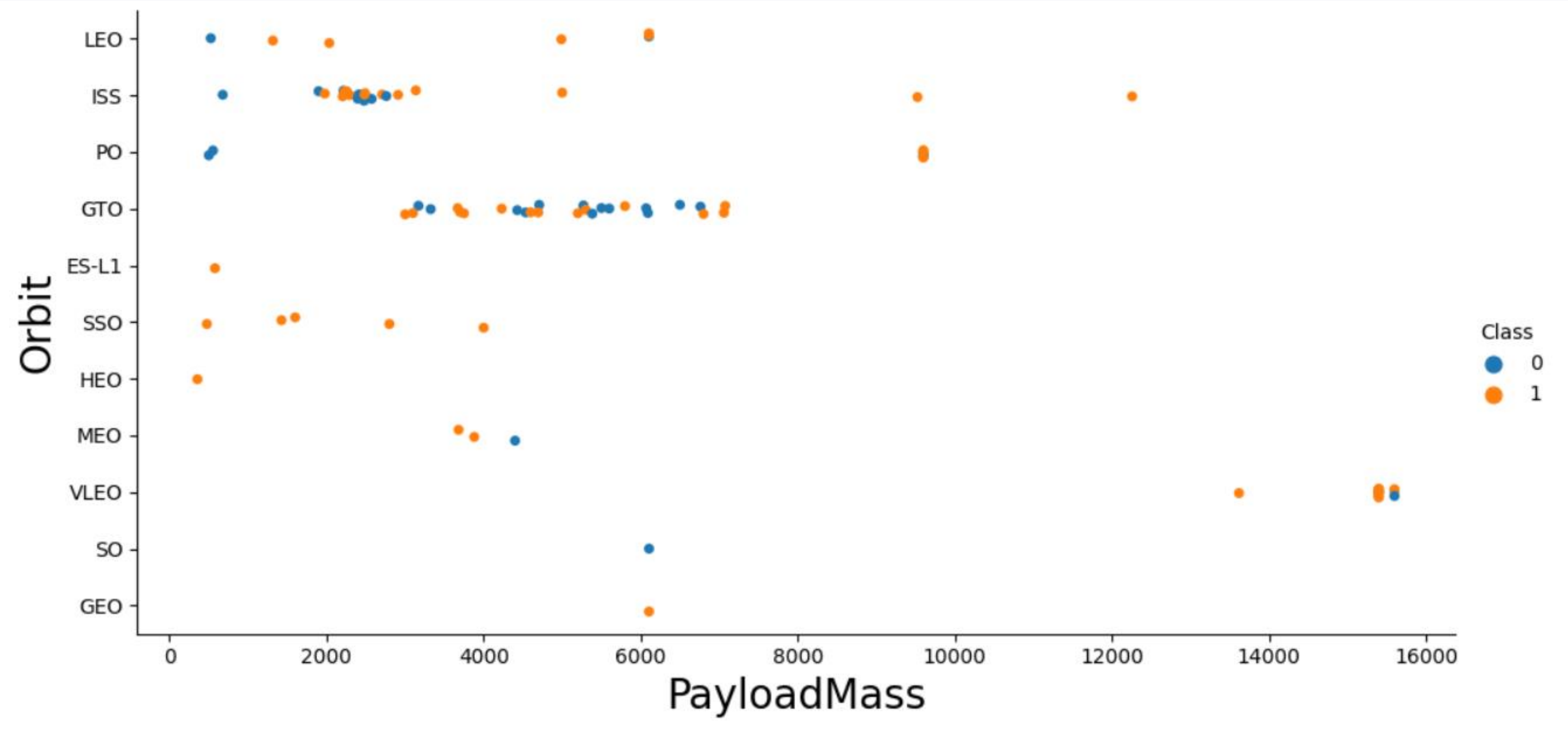
- ES-L1, GEO, HEP and SSO has the highest success rate. However, some of the orbit types only has few data points. More information would be needed to draw a clear pattern.

Flight Number vs. Orbit Type



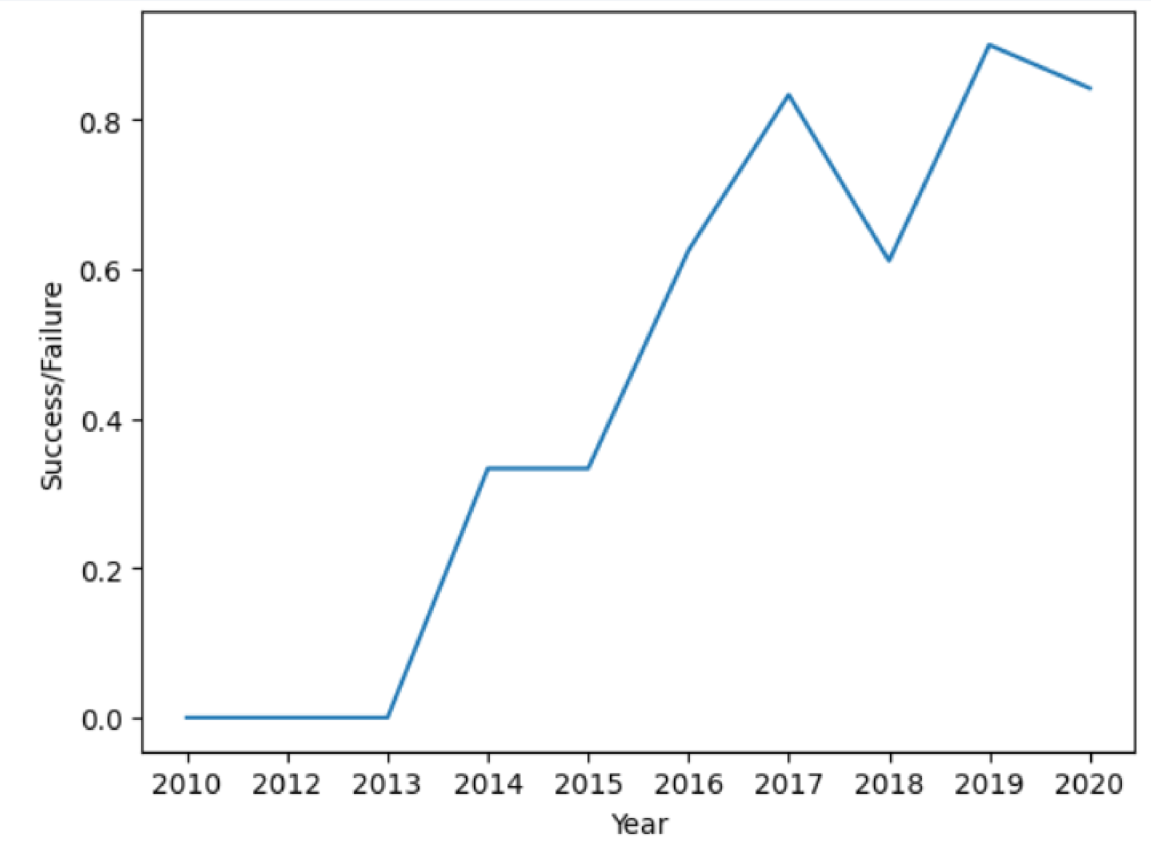
- The scatter plot shows the orbit pattern of launch, starting from at “low earth orbit” moving toward at “very low earth orbit” and at “further outer earth orbit”
- The trend shows the succuss rate improved over launch occurrence increases.

Payload vs. Orbit Type



- The scatter plot shows the designed payload for VLEO “very low earth orbit” is higher than the payload of other orbit type
- The design payloads for outer orbits generally lower than low earth orbit.

Launch Success Yearly Trend



- The success rate improved starting from 2013, even with a dip in 2018, and 2020, the trend of improvement is expected to carry on.

All Launch Site Names


```
%sql select distinct(LAUNCH_SITE) from SPACEX;
```

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

- There are 4 sites in the database CCAFS LC-40, CCAFS SLC-40, KSC LC-39A, VAFB SLC-4E

Launch Site Names Begin with 'CCA'


```
%sql select * from SPACEX where LAUNCH_SITE like 'CCA%' limit 5;
```

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	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- With the SQL code, the report retrieve 5 records of launch site with “CCA”

Total Payload Mass

```
%sql select sum(PAYLOAD_MASS_KG_) from SPACEX where CUSTOMER='NASA (CRS)';
```

1
45596

- With the SQL code, the results of total payload carried by boosters from NASA were calculated, the number is 45596.

Average Payload Mass by F9 v1.1


```
%sql select avg(PAYLOAD_MASS_KG_) from SPACEX where Booster_Version ='F9 v1.1';
```

1
2928

- With the SQL code, average payload mass carried by booster version F9 v1.1 was calculated, the result is 2928.

First Successful Ground Landing Date


```
%sql select DATE, Landing_Outcome from SPACEX where Landing_Outcome like 'Success (ground pad)';
```

DATE	landing_outcome
2015-12-22	Success (ground pad)

- With the SQL code, the dates of the first successful landing outcome on ground pad was 2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000


```
%sql select BOOSTER_VERSION from SPACEX where PAYLOAD_MASS_KG_ between 4000 and 6000 and Landing_Outcome = 'Success (drone ship)';
```

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

- With the SQL code, a list of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 was called

Total Number of Successful and Failure Mission Outcomes


```
%sql select MISSION_OUTCOME, count(MISSION_OUTCOME) as count FROM SPACEX group by MISSION_OUTCOME;
```

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

- With the SQL code, the total number of successful and failure mission outcomes was calculated

Boosters Carried Maximum Payload


```
%sql SELECT BOOSTER_VERSION, PAYLOAD_MASS_KG_ from SPACEX where PAYLOAD_MASS_KG_ = (select MAX(PAYLOAD_MASS_KG_) from SPACEX);
```

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

- With the SQL code, a list of names of the booster which have carried the maximum payload mass was called

2015 Launch Records


```
%sql select DATE, BOOSTER_VERSION, LAUNCH_SITE from SPACEX where LANDING_OUTCOME ='Failure (drone ship)' and DATE like '%2015%';
```

DATE	booster_version	launch_site
2015-01-10	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	F9 v1.1 B1015	CCAFS LC-40

- With the SQL code , a list of failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015 was generate

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


```
%sql select LANDING_OUTCOME, count(*) as COUNT from SPACEX where DATE between '2010-06-04' and '2017-03-20' group by LANDING_OUTCOME order by COUNT DESC;
```

landing_outcome	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

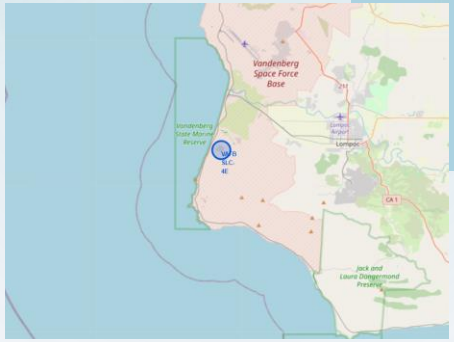
- With the SQL Code, a ranking the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order was generated

Section 3

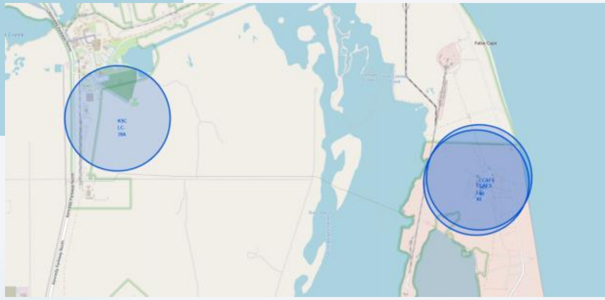
Launch Sites Proximities Analysis

Launch Sites in the MAP

There are 4 launch Sites in the Folium Map



VAFB SLC-4E located in West Coast

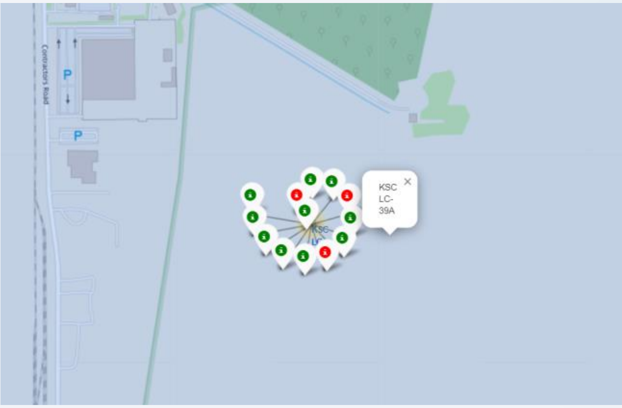


KSC LC-39A, CCAFS SLC-40 CCAFS LC-40 located in East Coast

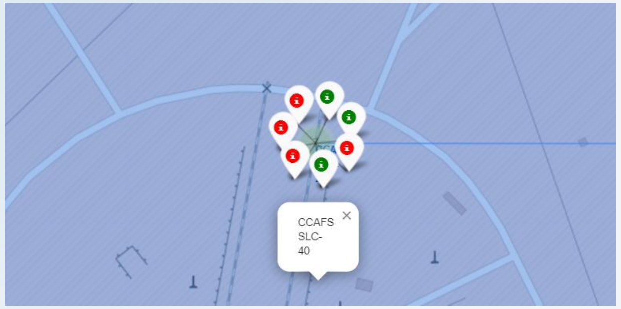
Launch outcomes of each site

Red indicated Fail launch, and Green indicated success launch
Launch number staring from the center of each launch site

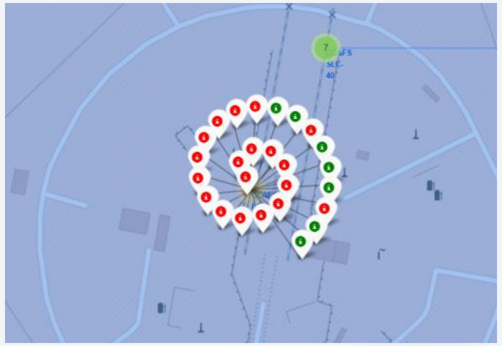
VAFB SLC-4E



KSC LC-39A

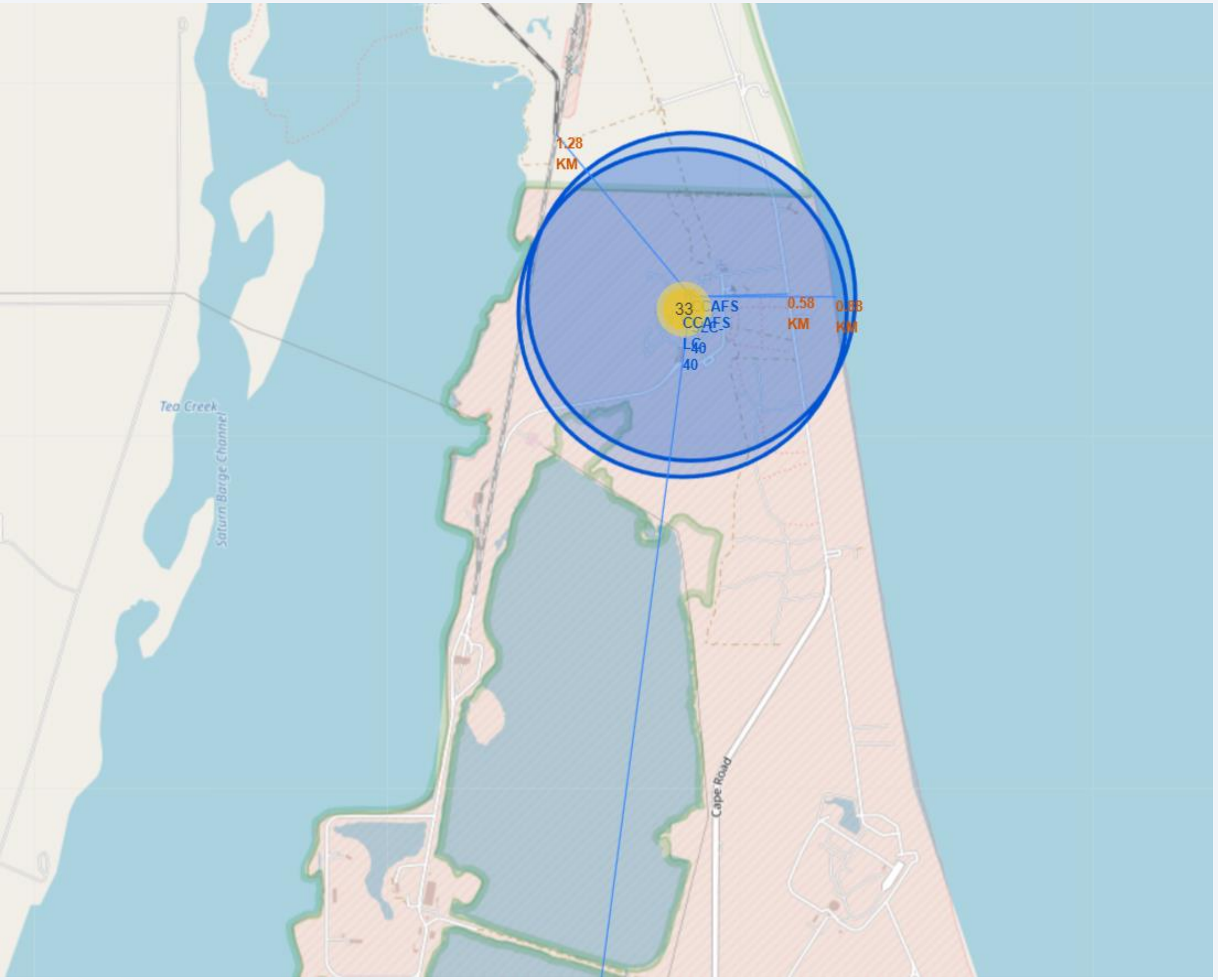


CCAFS SLC-40



CCAFS LC-40

Key landmarks distance from launch site



As shown in the screen shot, the line indicated distance to key facilities to the launch site:

Distance to highway : 0.58 km

Distance to railroad: 1.28 km

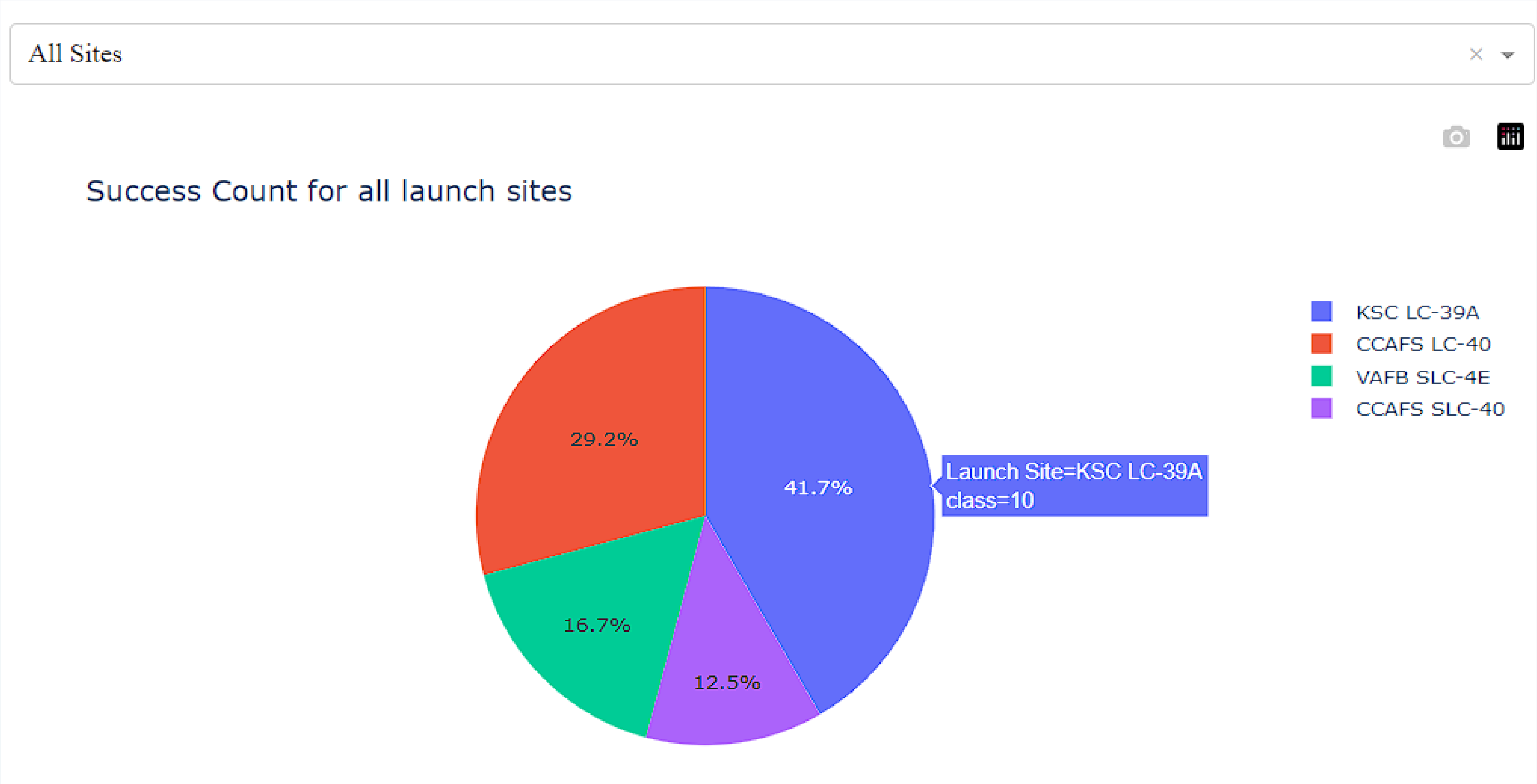
Distance to city:51.43 km

Distance to coast line: 0.88 km

Section 4

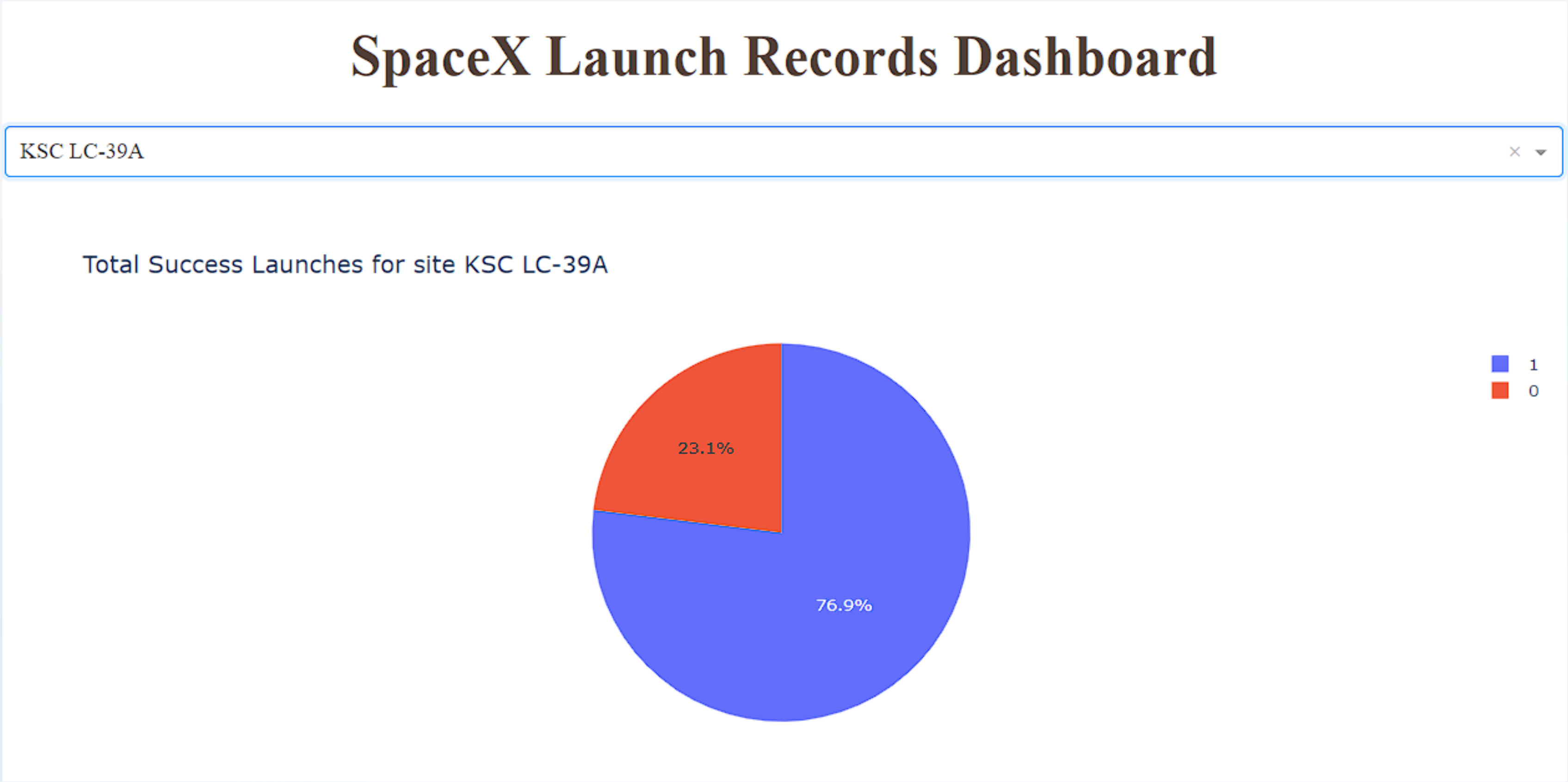
Build a Dashboard with Plotly Dash

Success Count from Dash Board



The pie-chart indicates that KSC LC-39A launch site has the highest success count whereas CCAFS SLC-40 has the fewest success count.

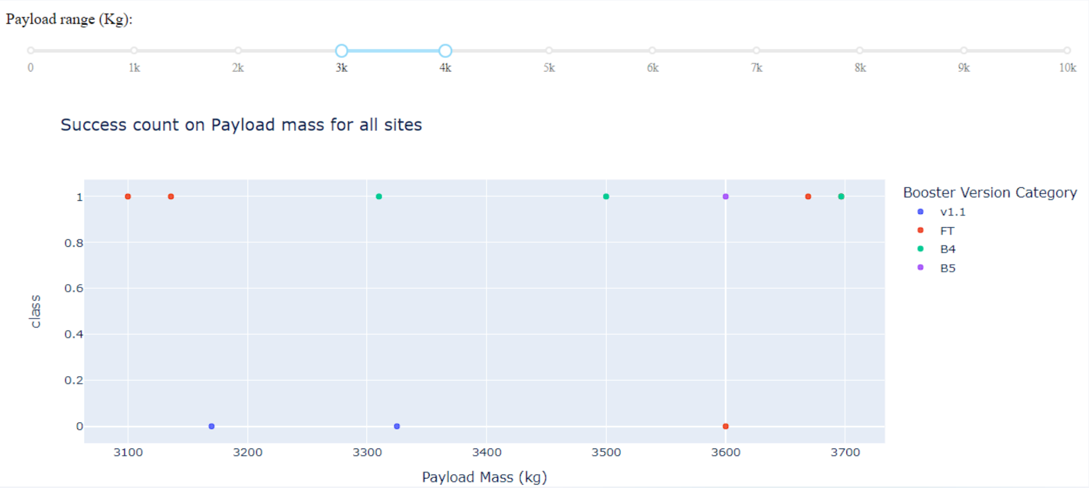
Success Rate of KSC LC-39A



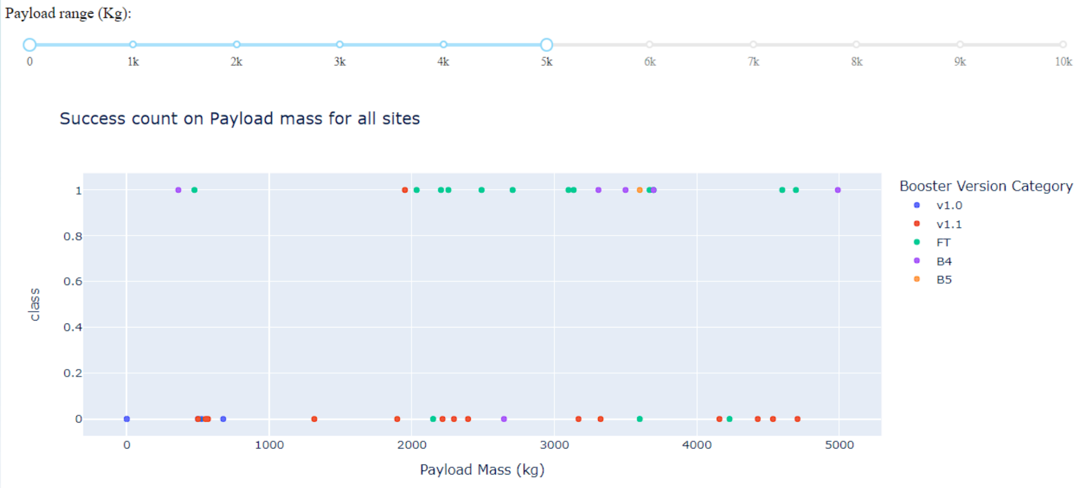
The success rate of KSC LC-39A is 76.9%, which is also the highest among other launch sites.
In summary, KSC LC-39A has the highest success rate as well as the highest launch occurrence.

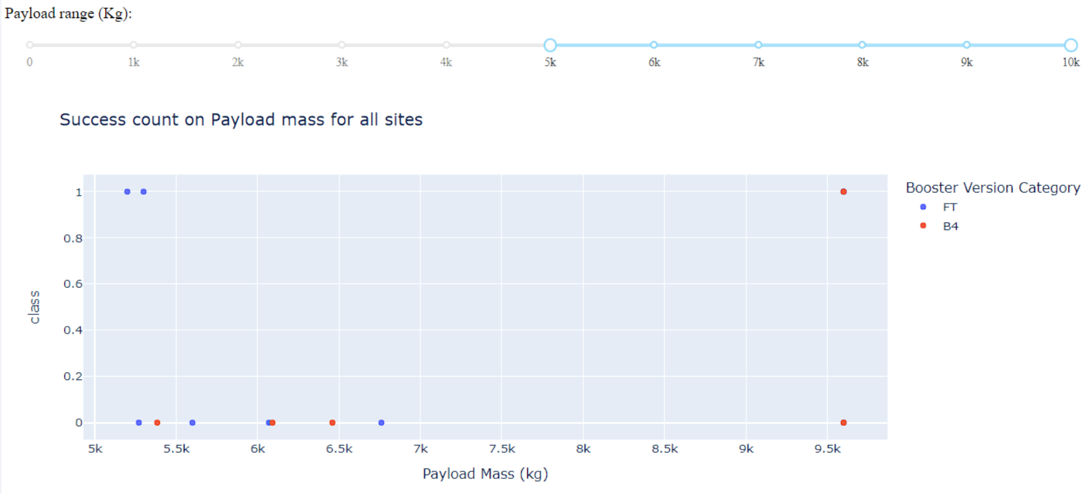
Interactive Dash Board Scatter Plots

The scatter plot shows the relationship between success count VS Playload mass with different booster version



In the range of Payload mass between 3K and 4K, the success rate is the highest.



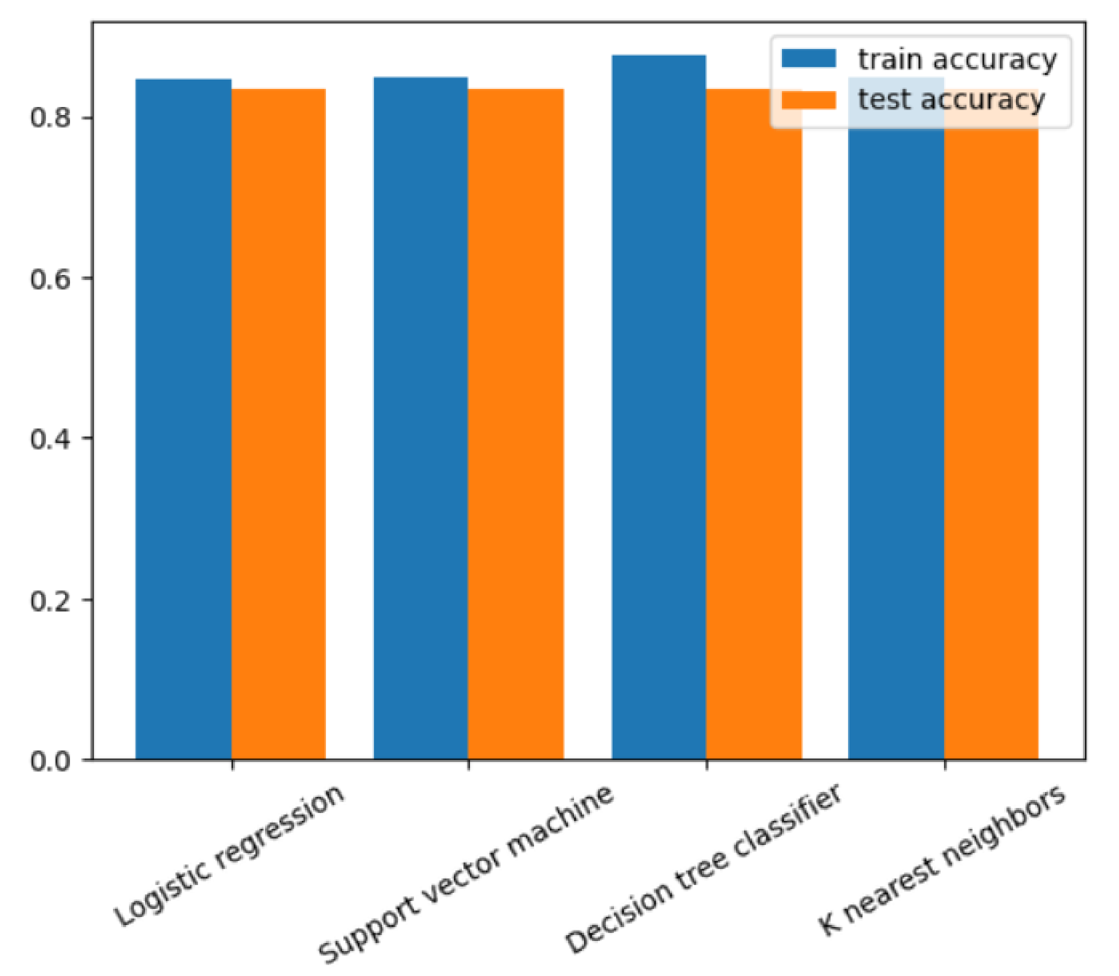


Compared to heavy payload (>5K), lighter payload mass has more launch occurrence and higher success rate.

Section 5

Predictive Analysis (Classification)

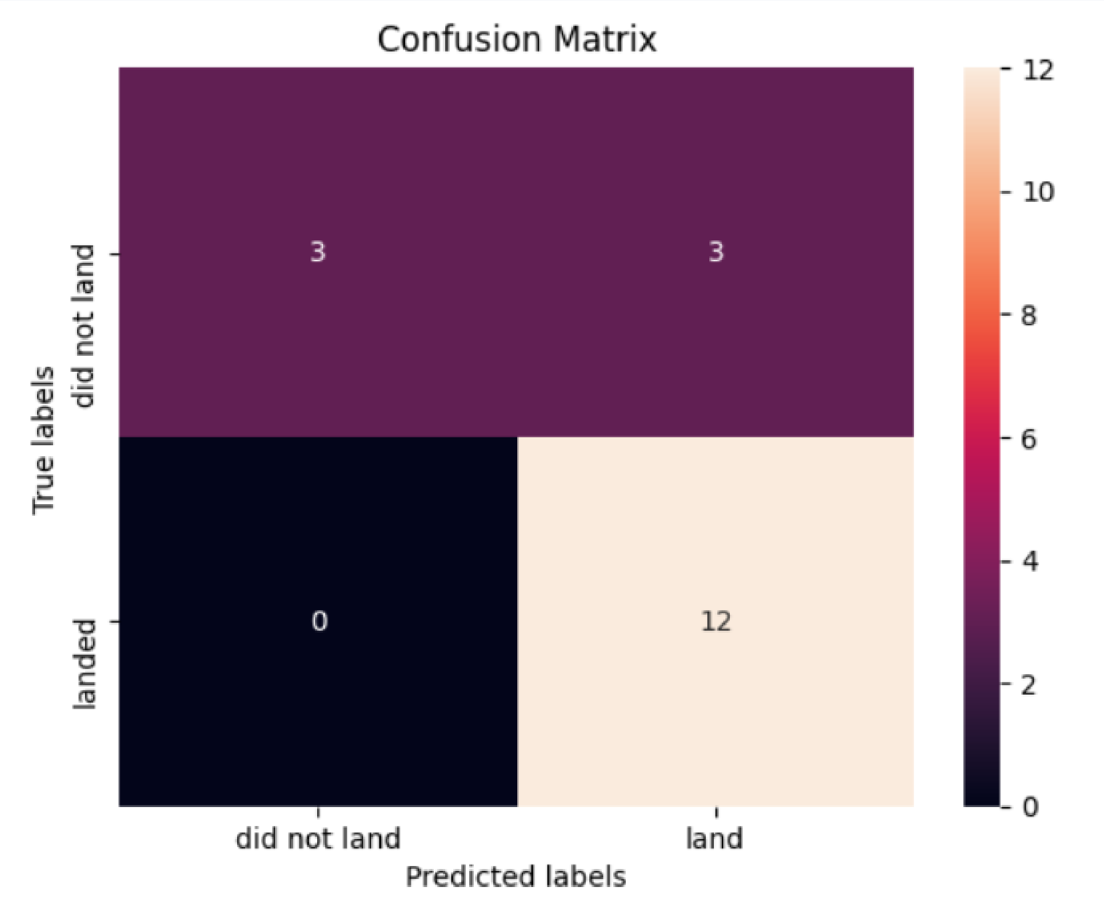
Classification Accuracy



Decision Tree Classifier has the highest train accuracy among the 4 classification method.

method	train accuracy	test accuracy
Logistic regression	0.846429	0.833333
Support vector machine	0.848214	0.833333
Decision tree classifier	0.875000	0.833333
K nearest neighbors	0.848214	0.833333

Confusion Matrix



The confusion matrix shows that the model is accurate as true positive value is high.

Conclusions

- The success rate of SpaceX improved starting from 2013 from 0% to 80%.
- KSC LC-39A launch site is considered to be most sophisticated site for launch with most occurrence and highest success rate.
- ES-L1, GEO, HEP orbit has 100% successful rate, however, the occurrence is only once. More data would needed to numerically conclude the practice is mature.
- Low earth orbit is considered to be more mature practice as the success rate improved over time. However, very low earth orbit and outer orbit are the field to be discovered.
- Decision Tree Classifier would be a suitable model for predicting a launch result.

Appendix and Final Remarks

- Restricted by the limited launch occurrence (around 90), this report does not remove outliers. With more data available, research in to future could use boxplot to detect and remove outliers.

- The time span of the data is around 5 years, many of the technology advancement might be elevated, resulting inaccuracy of the model. User of the model should verify the existing model with date in the future.

- The mission of each launch may be different which may cause the difficulties of each launch differs. When making predictions, user of the model should consider nuances in each launch and discover hidden parameter to provide sound forecast.

Thank you!