IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Yuvan Rajadeva
October 2024

Outline

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

Executive Summary

In this project, Space Y, a rival company to Space X, determines the rocket first stage landing successes using Space X Falcon 9 data.

- Summary of methodologies
  - API Data Collection
  - Web Scraping Data Collection
  - Data Wrangling
  - Exploratory Data Analysis using SQL and MatplotLib
  - Interactive Visual Analysis with Folium
  - Predictive Analysis using Machine Learning

- Summary of all results
  - Exploratory Data Analysis Results
  - Interactive Visual Analysis Results
  - Predictive Analysis Results

Introduction

- Project Background:

In this capstone, we will 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 can be used if an alternate company wants to bid against SpaceX for a rocket launch like SpaceY.

- Problems you want to find answers

We need to determine if the first stage will land and determine the cost of a launch.

The factors that determine the success rate for a successful landing.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The data was collected through the SPACEX API and web scraping from wiki pages.
- Perform data wrangling
  - Data was converted to panda dataframes for visualization and analysis. JSON objects, and HTML methods were used.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Machine Learning was used in the predictive analysis

Data Collection

- Data collection was obtained through various methods
  - Get Requests
  - JSON functions
  - Panda Dataframes
  - BeautifulSoup
  - HTML Tables

Data Collection – SpaceX API

- Here are examples of the key methods being implemented to collect the data needed:

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df[df['BoosterVersion']!='Falcon 1']
```

Now that we have removed some values we should reset the FlgihtNumber column

```
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9
```

```
# Calculate the mean value of PayloadMass column
payLoadMassAvg = data_falcon9['PayloadMass'].mean()

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

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

- Github Link: https://github.com/YuvanR/spaceYproject/blob/d451aac68f9ca9d2b8d99a578ef3cc13e32d2ccc/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- Web Scrapping was used to retrieve the Falcon 9 launch records with Beautiful Soup
- The tables were also converted to pandas dataframe.
- Here are images of the methids being implemented:
- Github: https://github.com/YuvanR/spaceYproject/blob/d451aac68f9ca9d2b8d99a578ef3cc13e32d2ccc/jupyter-labs-webscraping%20(1).ipynb

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

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')
```

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

```
# Use soup.title attribute
print(soup.title)
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

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

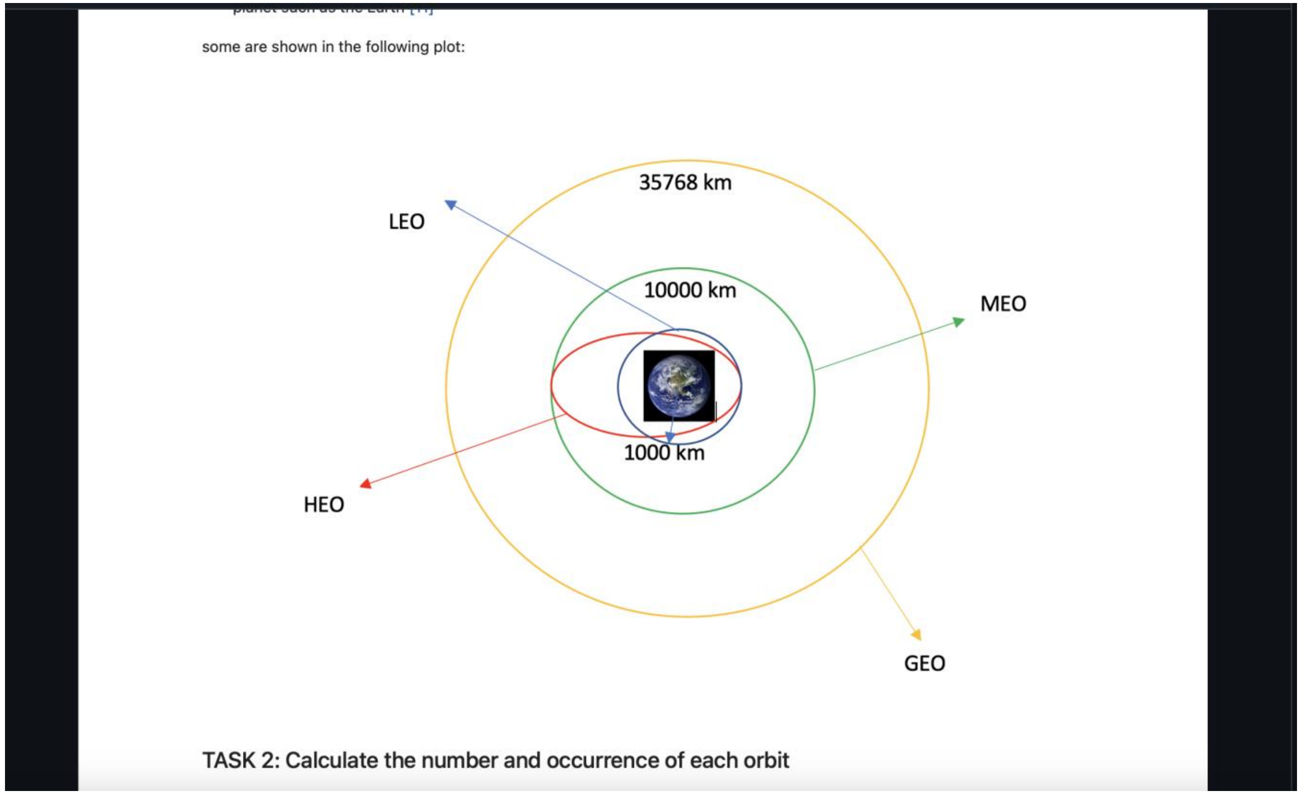
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all("table")
print(html_tables)
```

```
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']=[]
```

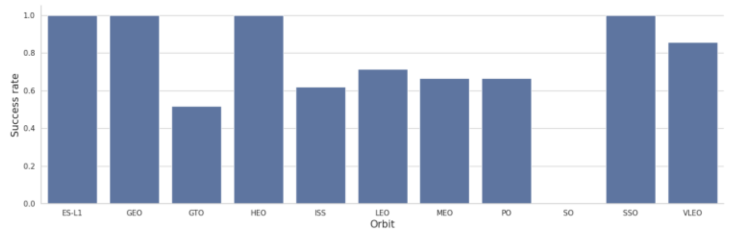
Data Wrangling

- The data was processed through exploratory data analysis:
    - First the number of launches were calculated at each site
    - Next the number and occurrence of each orbit was calculated
    - The number and occurrence of mission outcome of the orbits was also calculated.
    - A landing outcome label from Outcome column was created.
    - Finally the results were exported to csv.

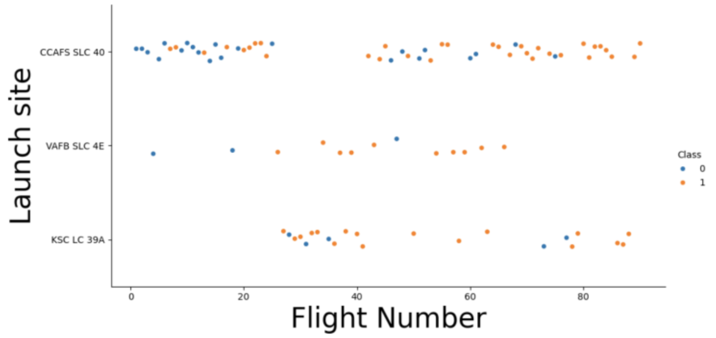
- Github: https://github.com/YuvanR/spaceYproject/blob/d451aac68f9ca9d2b8d99a578ef3cc13e32d2ccc/labs-jupyter-spacex-Data%20wrangling.ipynb



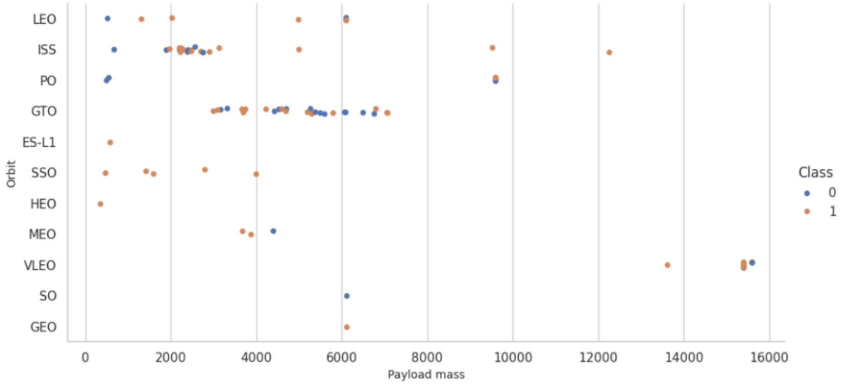
EDA with Data Visualization

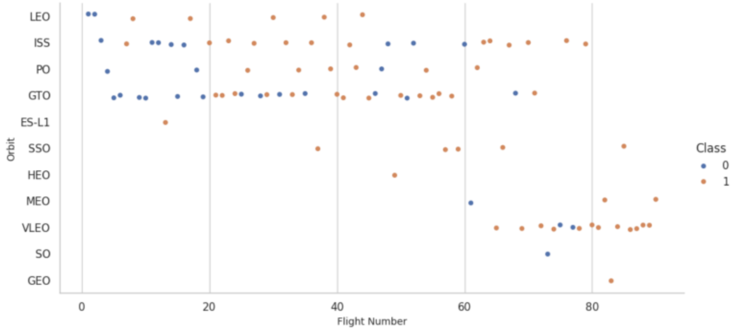


- The data was further analyzed through visualization and the exploration of the relationship between flight number and launch site, flight number and orbit type, payload and launch site, and the success rate of each orbit type.



- Some examples of the charts created:





- Github: https://github.com/YuvanR/spaceYproject/blob/1f7f0d733b54028e23b175a4441c8d138de16d8d/EDA_with_Visualization_Lab.ipynb

EDA with SQL

- EDA with SQL was used to gain more information on the data. Queries were used to extract specific data such as:

```
%sql SELECT Distinct(Launch_Site) FROM SPACEXTBL;

 * sqlite:///my_data1.db
Done.
```

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

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

 * sqlite:///my_data1.db
Done.
```

SUM(PAYLOAD_MASS__KG_)
45596

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

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

 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (p
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 (p
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	N
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	N
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	N

```
%sql SELECT PAYLOAD FROM SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AN

 * sqlite:///my_data1.db
Done.
```

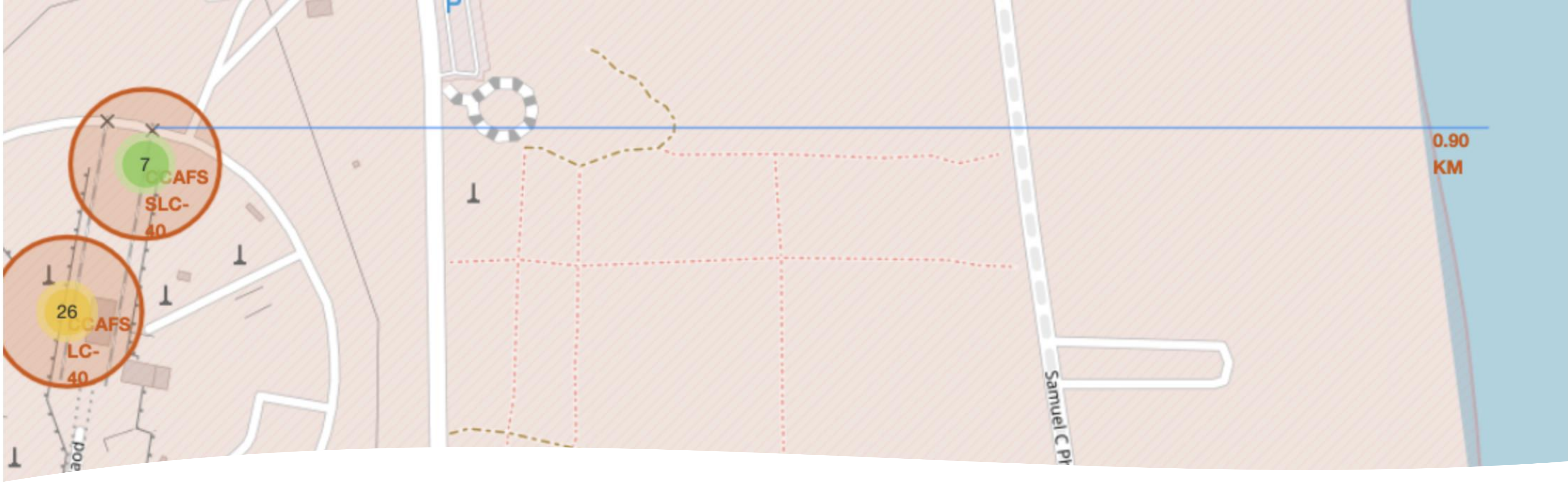
Payload
JCSAT-14
JCSAT-16
SES-10
SES-11 / EchoStar 105

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1';

 * sqlite:///my_data1.db
Done.
```

AVG(PAYLOAD_MASS__KG_)
2928.4

- Github: https://github.com/YuvanR/spaceYproject/blob/4442ade7c9294db05ba0bf077d5c3b6ac84d8cf2/jupyter-labs-eda-sql-coursera_sqllite.ipynb



Build an Interactive Map with Folium

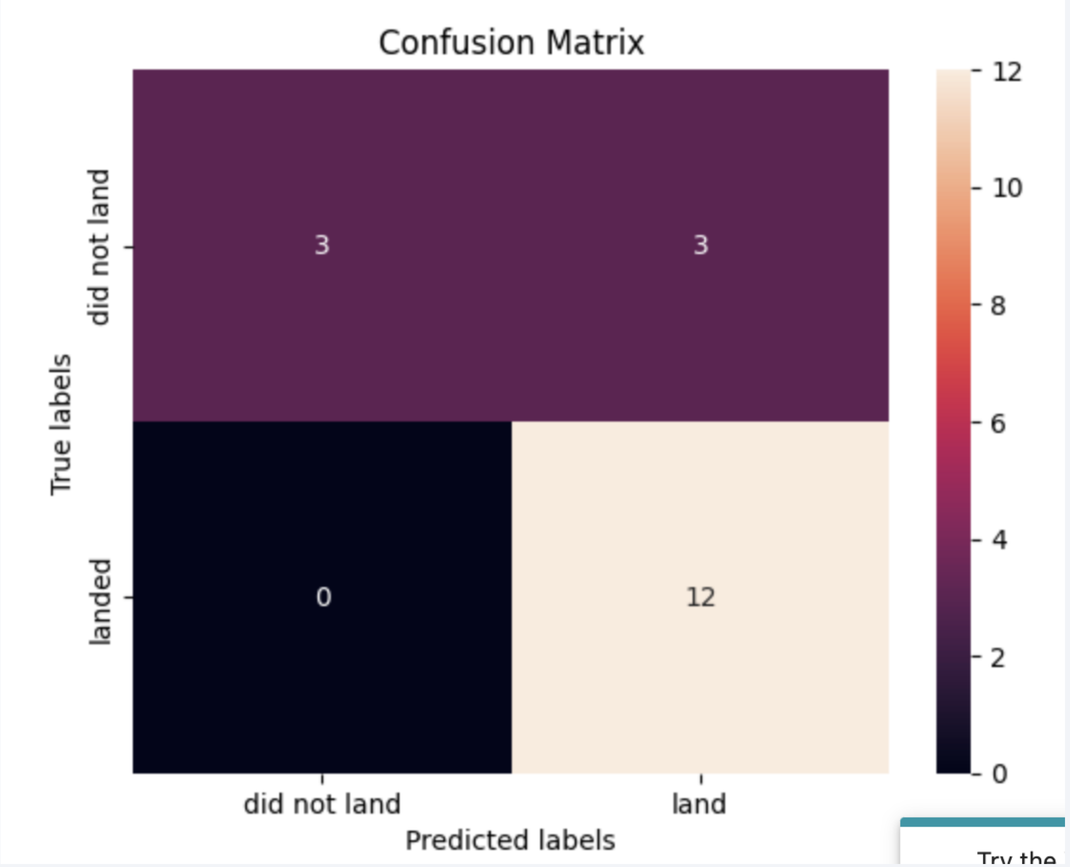
- All launch sites were marked
- Mab objects that were added were markers, circles, and lines
  - These indicated the success or failure for a launch sites
- Marker Clusters that were colored showed the specific launch sites that have a high success rate compared to others.
- Distance between launch sites and its proximities were also calculated as shown.
- Github: https://github.com/YuvanR/spaceYproject/blob/1f7f0d733b54028e23b175a4441c8d138de16d8d/Interactive_Visual_Analytics_with_Folium_lab.ipynb

Build a Dashboard with Plotly Dash

- Plotted pie charts that showed the total launches by a certain site
- Scatter graphs were plotted to show the relationship between Outcome and Payload Mass for the different booster version
    - This was important information that helped determine the success rate for a first stage launch.
- Github: https://github.com/YuvanR/spaceYproject/blob/4442ade7c9294db05ba0bf077d5c3b6ac84d8cf2/dash_interactivity.py

Predictive Analysis (Classification)

- Created a NumPy array from the column class in data.
- Function train_test_split was used to split the data into a training and test data.
- Built different machine learning models to find the best hyperparameter using GridSearchCV
- GitHub: https://github.com/YuvanR/spaceYproject/blob/d5e254a5585a9f7e3c92dad4ec50330b87c2b972/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb



Results

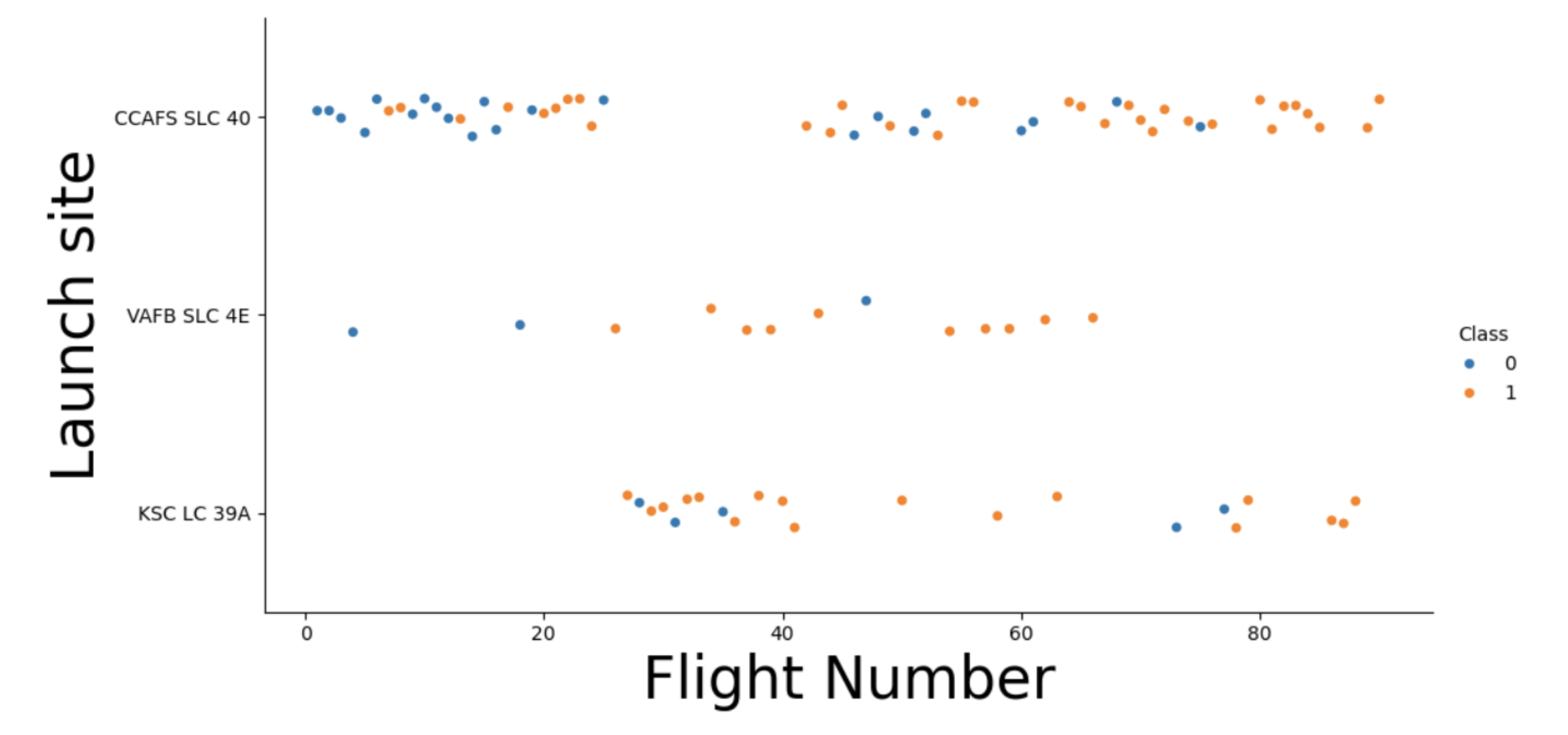
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

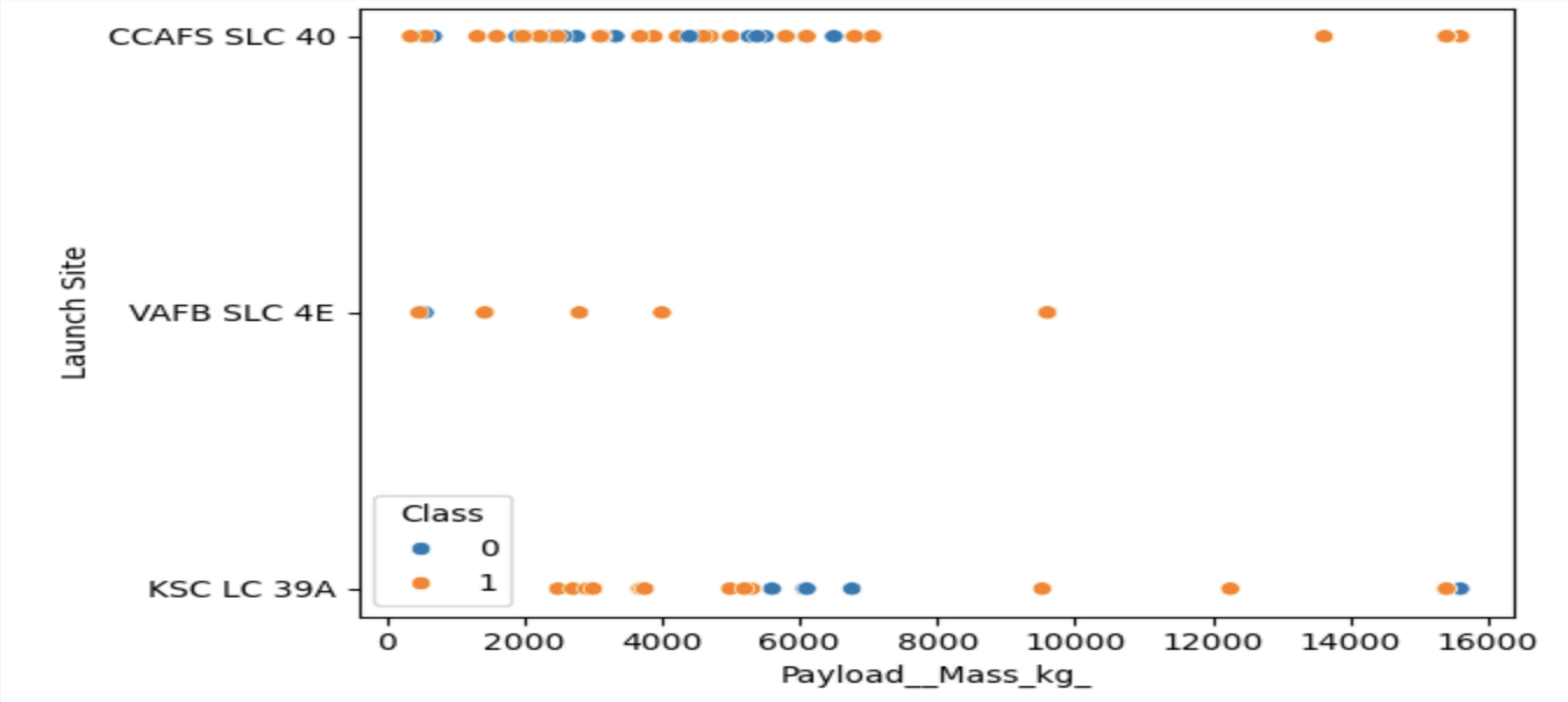
Insights drawn from EDA

Flight Number vs. Launch Site

- The success rate for every Launch Site has increased with time.

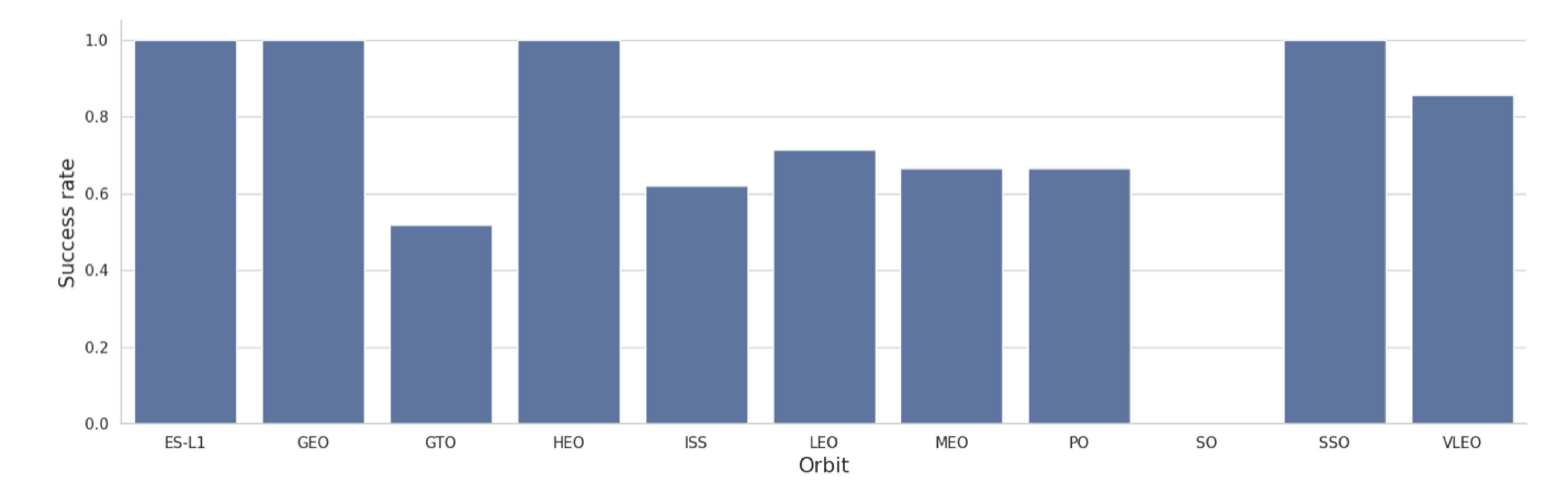


Payload vs. Launch Site



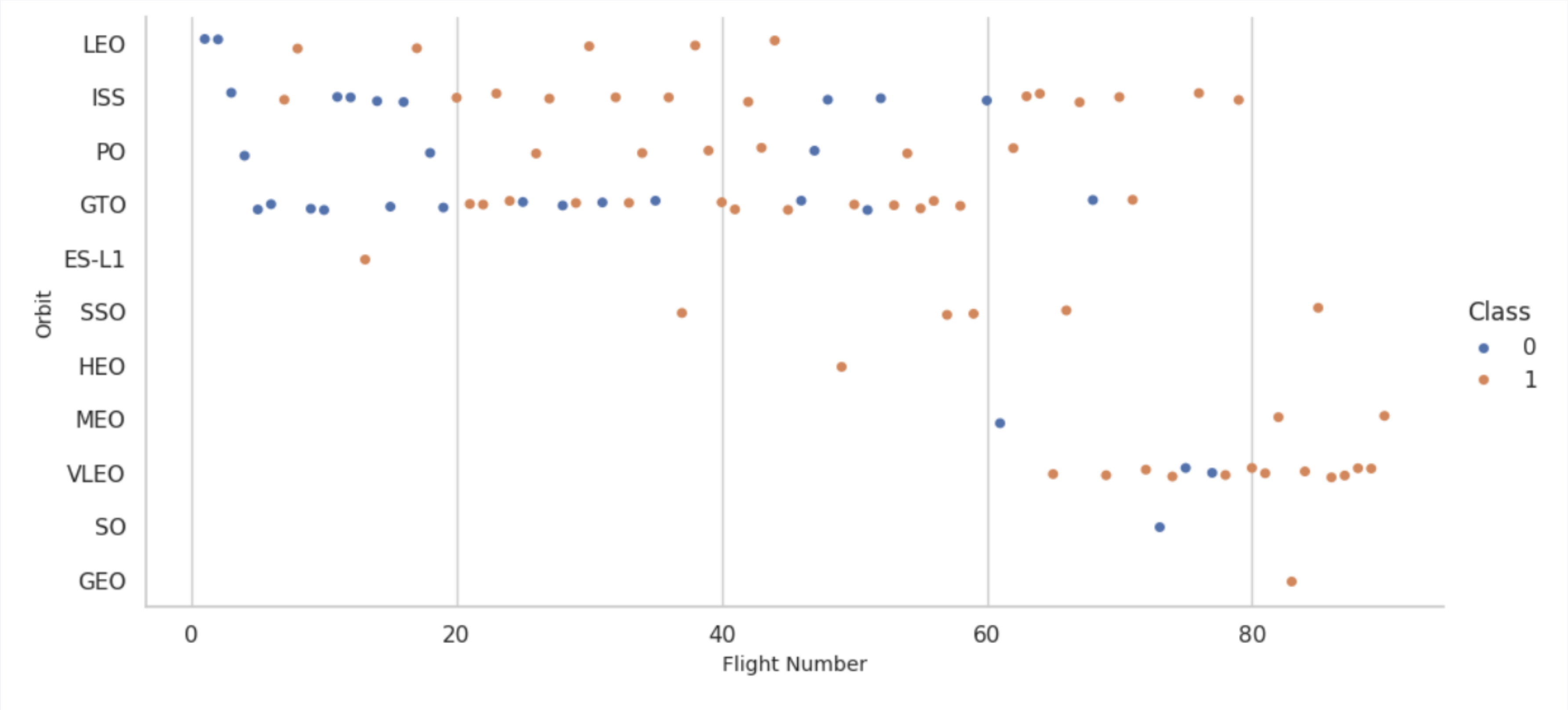
- CCAFS SLC – Launch rockets that is not between 7500 kg and 130000 kg.
- VAFB SLC 4E – No rockets launched after 100000 kg
- KSC LC – No rockets launched less than 2400 kg

Success Rate vs. Orbit Type

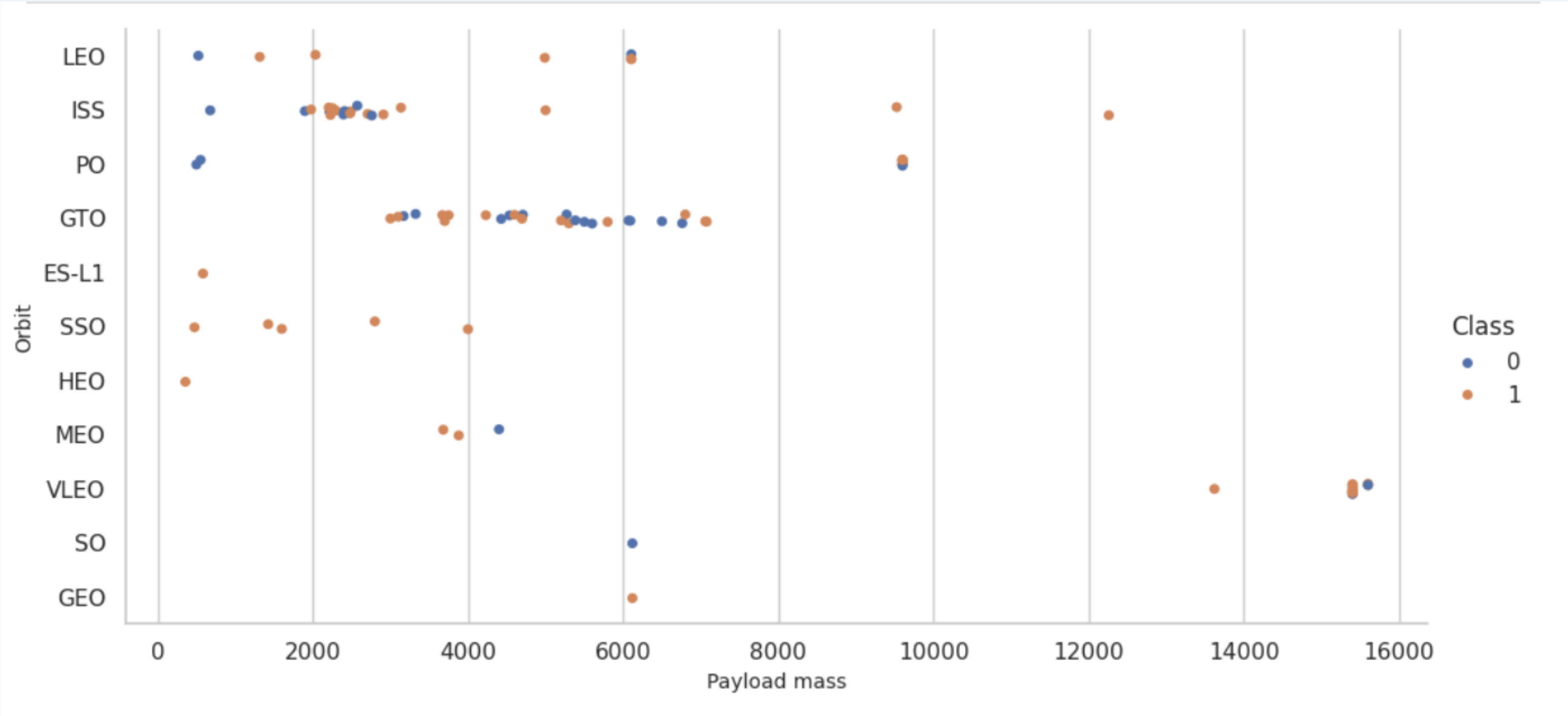


Flight Number vs. Orbit Type

- There are more failures at the beginning of the launches but it improves after the first 40 launces

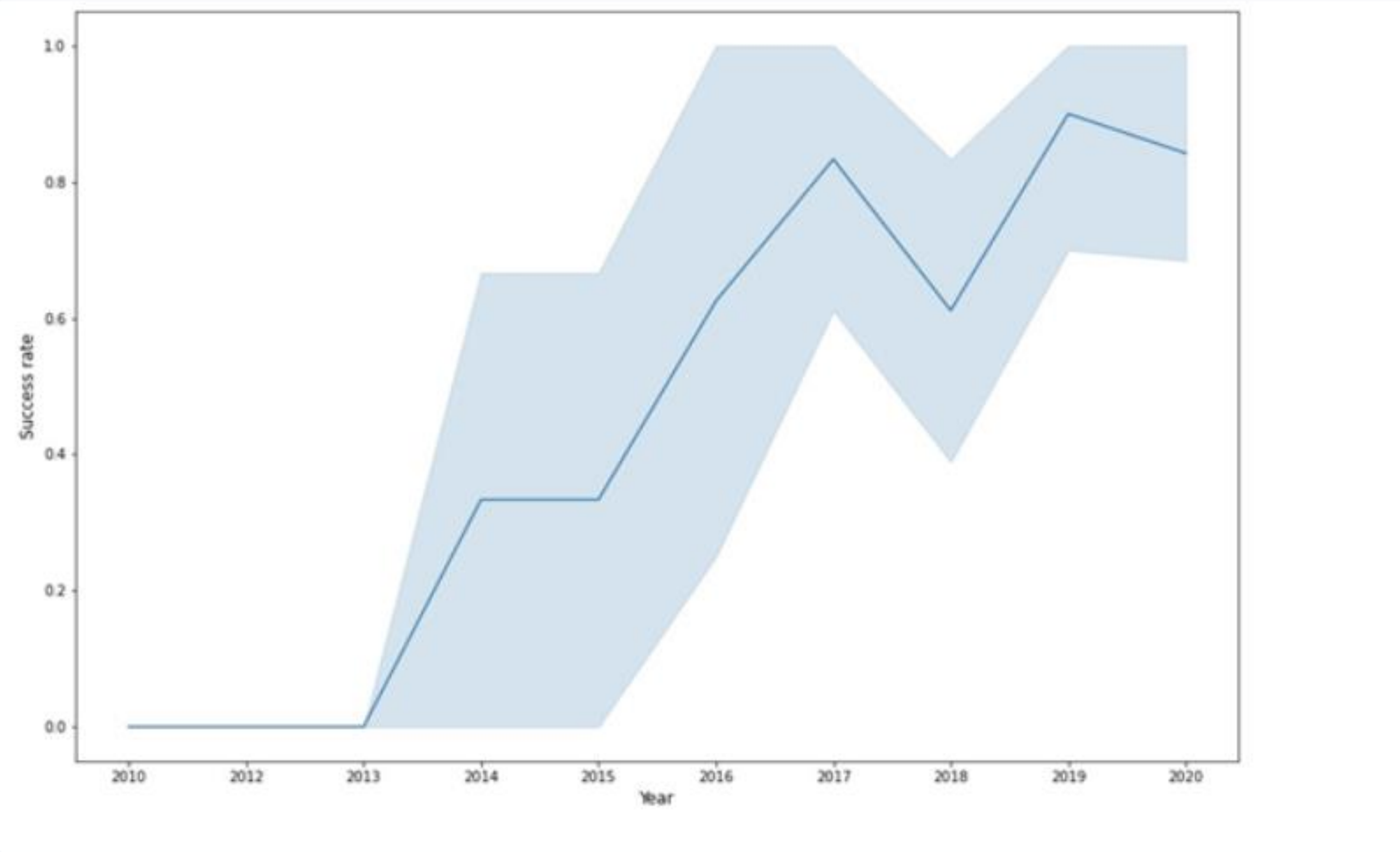


Payload vs. Orbit Type



- When the payloads are heavier, the successful landing rate is higher for Po, LEO, and ISS.
- More launces below 7600 kg then after 7600 kg.

Launch Success Yearly Trend



- Success rate increases from 2013 to 2020

All Launch Site Names

- DISTINCT was used to find the unique launch site names

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

Launch Site Names Begin with 'CCA'

- WHERE, LIKE, LIMIT, were used to find the 5 records beginning with 'CCA'

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (p
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 (p
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	N
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	N
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	N

Total Payload Mass

- SUM and WHERE were used to find the total payload mass.

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- AVG was used to find the average payload mass

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- MIN was used to find the date of the first successful landing outcome

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- WHERE and AND were used to find the PAYLOADS

Payload
JCSAT-14
JCSAT-16
SES-10
SES-11 / EchoStar 105

Total Number of Successful and Failure Mission Outcomes

- COUNT and GROUP BY were used to find the total number fo successful and failure mission outcomes.

Mission_Outcome	total_number
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- SUBQUERY was used to find the names of the boosters.

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

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

landing_outcome	booster_version	launch_site	DATE
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015-01-10
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015-04-14

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

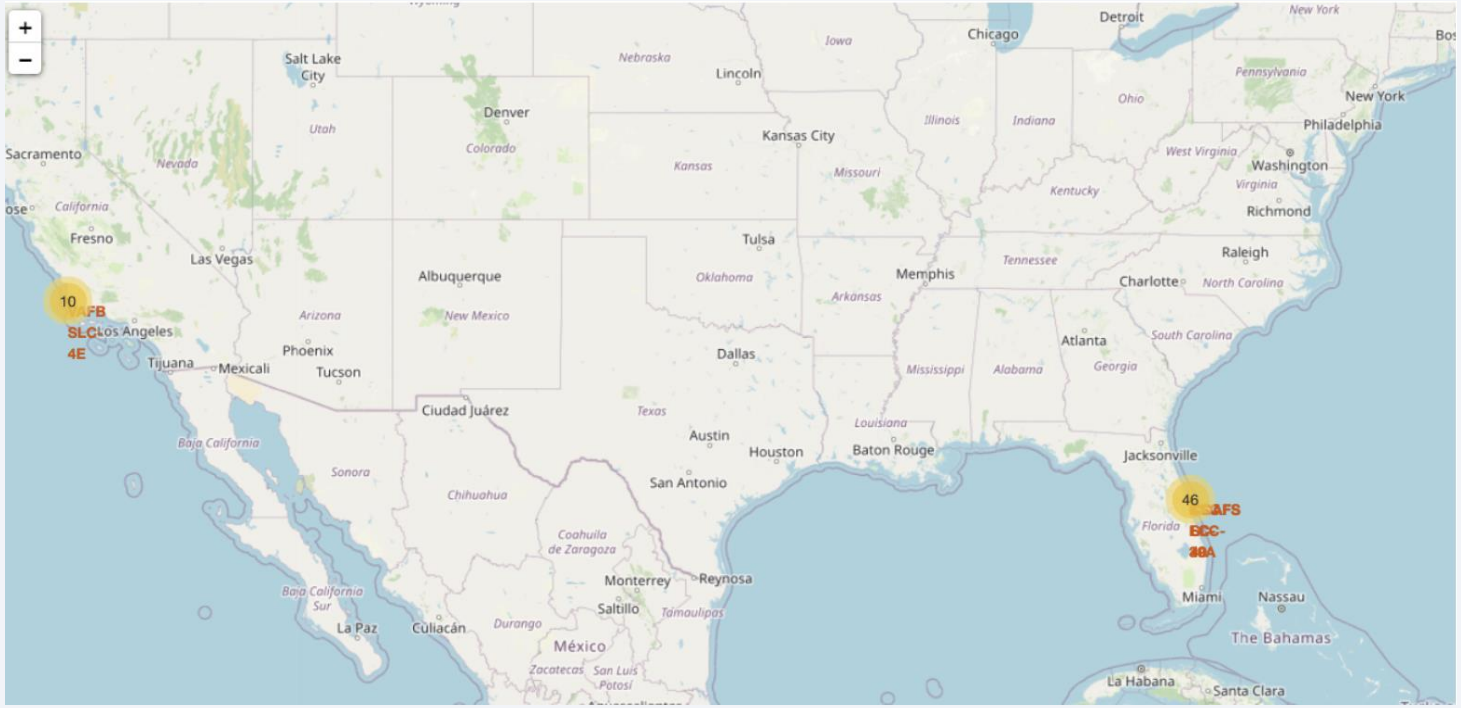
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

landing_outcome	total
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

Launch Sites on a Map



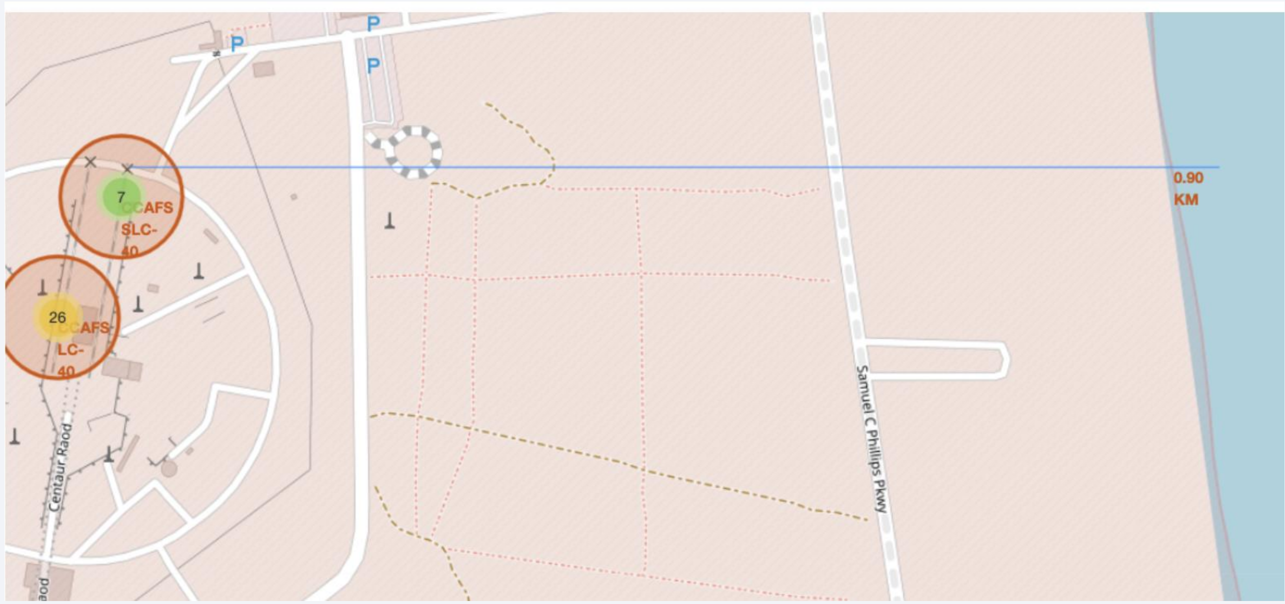
- The launch sites are located on the coasts of Florida and California

Launch Sites with Color Labels

- The Green marker shows the successful launches and the red markers shows the failed launches.

Launch Sites with color markers and distance

- The distance marker shows the distance from the successful launch to the coast.

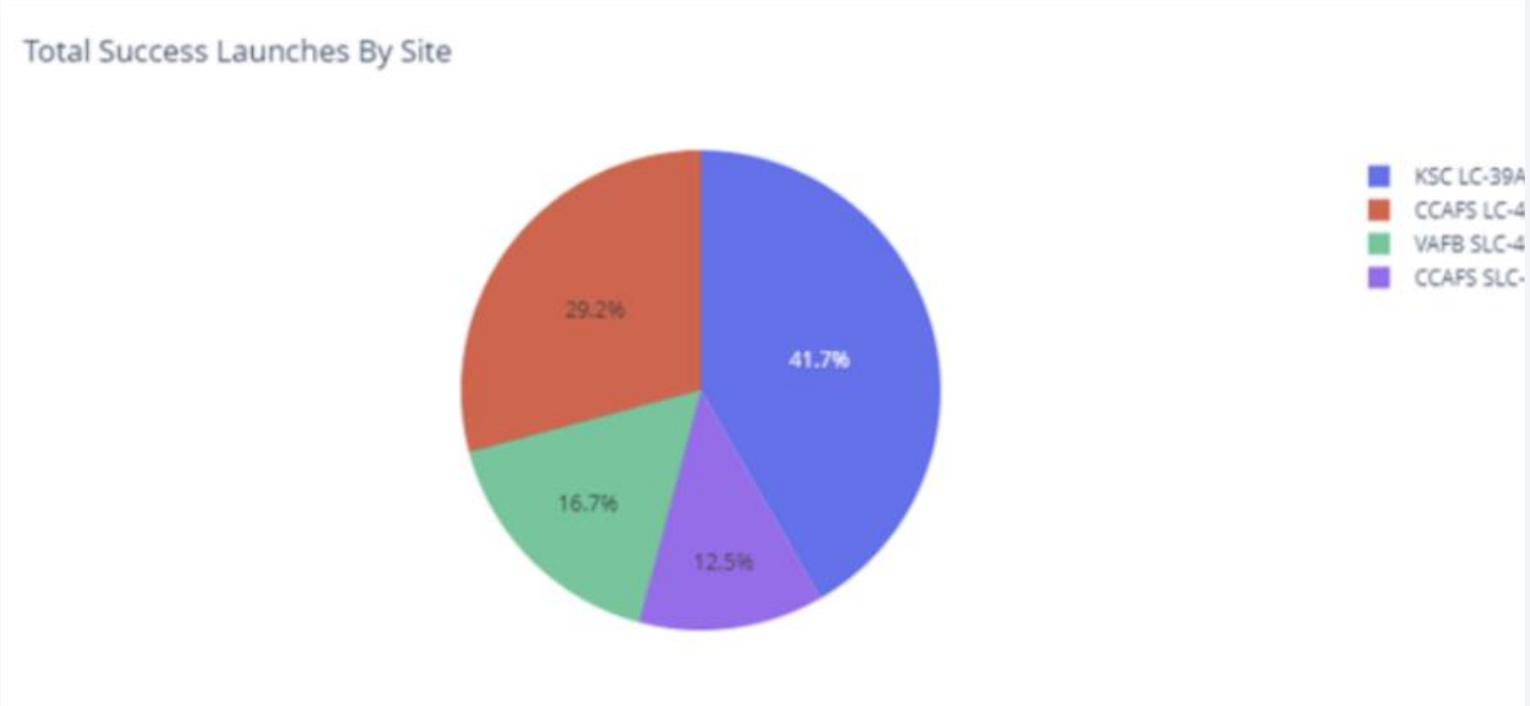


Section 4

Build a Dashboard with Plotly Dash

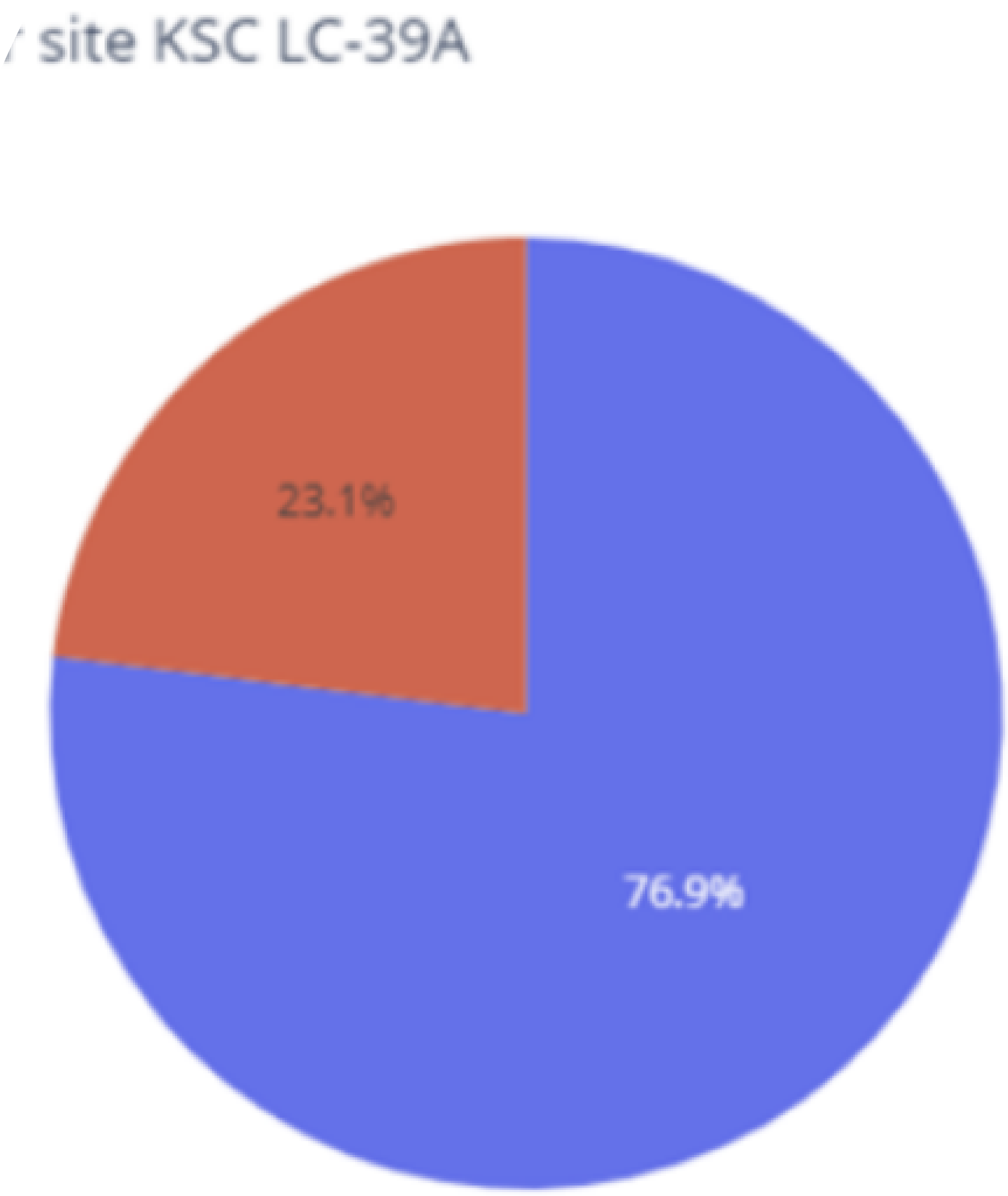
Success Launches by Site

- KSC LC-39A had the most successful launches between all the launches.



Success launches for KSC LC-39A

Achieved a 76.9 percent success rate!



Payload Mass vs Success

- The highest success rate for payloads is between 5000 and 7000 kg.

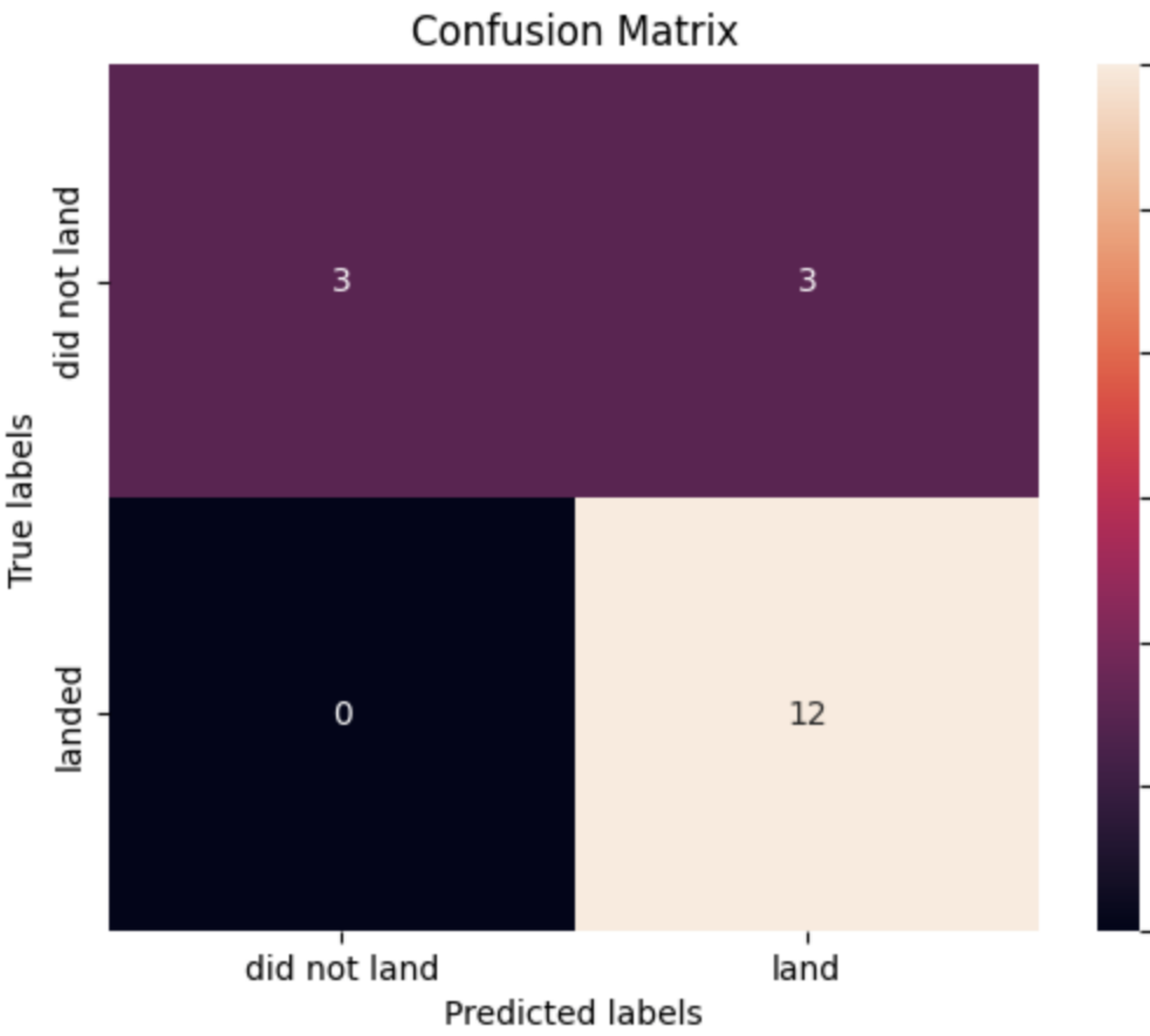
Section 5

Predictive Analysis (Classification)

Confusion Matrix

- The confusion matrix predicts 12 true positives, 3 false positives, true negatives, and 0 false negatives.

```
yhat=svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- When the flight amount at a launch site is larger, the success rate at the launch site is higher.
- Launch success rate started to increase after 2013
- KSC LC-39A was the launch site that had the highest success rate
- The best machine learning model was the Decesion Tree classifier as it provided the most accurate results and could analyze the spaceX data properly.

Appendix

- Github link to entire project: https://github.com/YuvanR/spaceYproject.git
- https://www.coursera.org/learn/applied-data-science-capstone/home/module/5

Thank you!