IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

HIMAANI SRIVATSAVA
13 Jul 2023

# Index

# Executive Summary

## Summary of methodologies

Data Collection API:

We have collected data from public SpaceX API and by scrapping SpaceX Wikipedia page.

Created labels column ‘Class’ which denotes all the successful landings by:

- Data Collection with Web Scraping
- Data Wrangling

## Summary of all results

We performed EDA (Exploratory Data Analysis) using SQL, visualization, folium maps, and dashboards.

- Exploratory Data Analysis using SQL
- Exploratory Data Analysis using Pandas and Matplotlib
- Interactive Visual Analytics with Folium
- Interactive Dashboard with Plotly

Machine Learning Prediction

- We standardized data and used GridSearchCVto find the best parameters for our machine learning models.We visualized the accuracy score of allmodels.
- We used four machine learning models-Logistic Regression, Support Vector Machine (SVM), Decision Tree Classifier, and K Nearest Neighbors. All produced similar results with the accuracy rate of almost 83.33%. All models over predicted successful landings. It is clear that more data is required for a better model determination andaccuracy

# Introduction

## Project background and context

The company SpaceY is a new rocket company. Their current main competitor is the company SpaceX, founded by Allon Musk. SpaceX is currently leading the affordable space travel, with accomplishments including sending a spacecraft to the International Space Station; sending Starlink, a satellite internet constellation providing satellite Internet access; and sending manned missions to Space. One reason SpaceX can do this is because the rocket launches are relatively inexpensive. SpaceX advertises Falcon 9 rocket launches with a cost of 62 million dollars; other providers cost upwards of165 million dollars each. Much of the savings is because SpaceX can reuse the first stage. Sometimes the first stage doesn’t land or crashes, and other times SpaceX sacrifices the first stage stage due to the mission parameters like payload, orbit, and customer, However, unlike other rocket providers, SpaceX's Falcon 9 can recover the first stage.

## Problems you want to find answers

The company SpaceY wants to be more competitive with the SpaceX. In order to work towards this, information about SpaceX was gathered to determine the price of each launch, and a machine learning model was trained to predict whether SpaceX will reuse the first stage.

Section 1

# Methodology

# Methodology

- Data collection methodology:
  Data was collected using the SpaceX REST API, alongside Web Scraping from Wikipedia

- Perform data wrangling
  Data was processed by cleaning and filtering the data, as well as addressing missing values, to prepare it for data analysis
- Perform exploratory data analysis (EDA) using visualization and SQL

- Perform interactive visual analytics using Folium and Plotly Dash

- Perform predictive analysis using classification models
  Logistic regression, support vector machine (SVM), decision tree, and k nearest neighbour (KNN) models were created, and accuracy of each model was assessed

# Data Collection

- Data collection process involved a combination of API requests from Space X public API and web scraping data from a table in Space X's Wikipedia entry.
- For consistency, the data was decoded as a Json using .json() and turned it into a Pandas dataframe using .json_normalize().
- The data was filtered to attain relevant information about the launch, and this extracted information was turned into a dataframe
- Data was cleaned and checked for missing values, which were filled in using the mean where necessary
- Web scraping from Wikipedia was done using BeautifulSoup to attain data for Falcon 9 launches
- A pandas data frame was created by parsing the launch information as HTML tables
- **Space X API Data Columns:**
- FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude
- **Wikipedia WebscrapeData Columns:**
- Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time

# Data Collection – SpaceX API

SpaceX APIThe API used is https://api.spacexdata.com/v4/rockets/

The API provides data about many types of rocket launches done by SpaceX, the data is therefore filtered to include only Falcon9 launches.

Every missing value in the data is replaced by the mean of the column that the missing value belongs to .

We end up with 90 rows or instances and 17 columns or features. The picture below shows

Here's the link to the notebook :

https://github.com/himaanisrivatsava/IBM-DataScience-professional-certification/blob/main/10.applied%20datascience%20capstone/10.1.jupyter-labs-spacex-data-collection-api.ipynb

### Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

In [37]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_
```

We should see that the request was successfull with the 200 status response code

In [38]:
```
response.status_code
```

Out[38]: 200

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

In [39]:
```
# Use json_normalize meethod to convert the json result into a dataframe
df=pd.json_normalize(response.json())
data=df
```

Using the dataframe `data` print the first 5 rows

In [40]:
```
# Get the head of the dataframe
df.head()
```

Out[40]:

| | static_fire_date_utc | static_fire_date_unix | net | window | rocket | success | failures | details | crew | ships | cap |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2006-03-17T00:00:00.000Z | 1.142554e+09 | False | 0.0 | 5e9d0d95eda69955f709d1eb | False | [{'time': 33, 'altitude': None, 'reason': 'merlin engine failure'}] | Engine failure at 33 seconds and loss of vehicle | [] | [] | |

# Data Collection - Scraping

- The data is scraped fromhttps://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Heavy_launches&oldid=1027686922
- The website contains only the data about Falcon9 launches.
- We end up with 121 rows or instances and 11 columns or features.
- Here's the link to the full code:

https://github.com/himaanisrivatsava/IBM-DataScience-professional-certification/blob/main/10.applied%20datascience%20capstone/10.1.jupyter-labs-webscraping.ipynb

### TASK 1: Request the Falcon9 Launch Wiki page from its URL

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

In [6]:
```
# 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`

In [7]:
```
# 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

In [8]:
```
# 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

In [9]:
```
# 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')
```

Starting from the third table is our target table contains the actual launch records.

# Data Wrangling

We Created a training label with landing outcomes where successful = 1 & failure =0.

Outcomecolumnhastwocomponents:‘MissionOutcome’‘LandingLocation’

New training label column ‘class’ with a value of 1 if ‘Mission Outcome’ is True and 0 otherwise.

**Value Mapping:**

True ASDS, True RTLS, & True Ocean –set to ->1

None None, False ASDS, None ASDS, False Ocean, False RTLS –set to ->0

Full code at:

In [13]:
```
df['Class']=landing_class
df[['Class']].head(8)
```

Out[13]:

| | Class |
|---|---|
| 0 | 0 |
| 1 | 0 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |
| 5 | 0 |
| 6 | 1 |
| 7 | 1 |

In [14]:
```
df.head(5)
```

Out[14]:

| | FlightNumber | Date | BoosterVersion | PayloadMass | Orbit | LaunchSite | Outcome | Flights | GridFins | Reused | Legs | LandingPad | Block |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2010-06-04 | Falcon 9 | 6104.959412 | LEO | CCAFS SLC 40 | None None | 1 | False | False | False | NaN | 1.0 |
| 1 | 2 | 2012-05-22 | Falcon 9 | 525.000000 | LEO | CCAFS SLC 40 | None None | 1 | False | False | False | NaN | 1.0 |
| 2 | 3 | 2013-03-01 | Falcon 9 | 677.000000 | ISS | CCAFS SLC 40 | None None | 1 | False | False | False | NaN | 1.0 |
| 3 | 4 | 2013-09-29 | Falcon 9 | 500.000000 | PO | VAFB SLC 4E | False Ocean | 1 | False | False | False | NaN | 1.0 |
| 4 | 5 | 2013-12-03 | Falcon 9 | 3170.000000 | GTO | CCAFS SLC 40 | None None | 1 | False | False | False | NaN | 1.0 |

# EDA with Data Visualization

• Scatterplot: Flight Number vs Payload Mass (kg):To see how the Flight Number (indicating the continuous launch attempts) and Payload variablesaffect the launch outcome

• Scatterplot: Flight Number vs Launch Site:To see how the Flight and Launch Site affect the launch outcome

• Scatterplot: Payload vs Launch Site:To observe if there is any relationship between launch sites and their payload mass (kg)

• Bar chart: Orbit Type vs Success Rate

• Visually check if there are any relationship between success rate and orbit type

• Scatterplot: Flight Number vs Orbit type

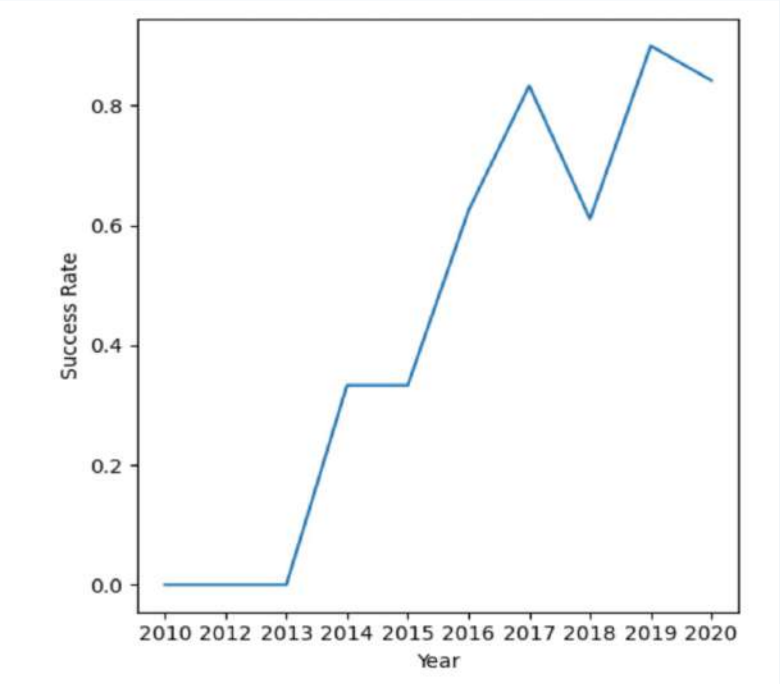
• To see for each orbit if there’s any relationship between Flight Number and Orbit type

• Scatterplot: Payload vs Orbit Type

• To see the relationship between Payload and Orbit type

• Line graph: Year vs Success Rate

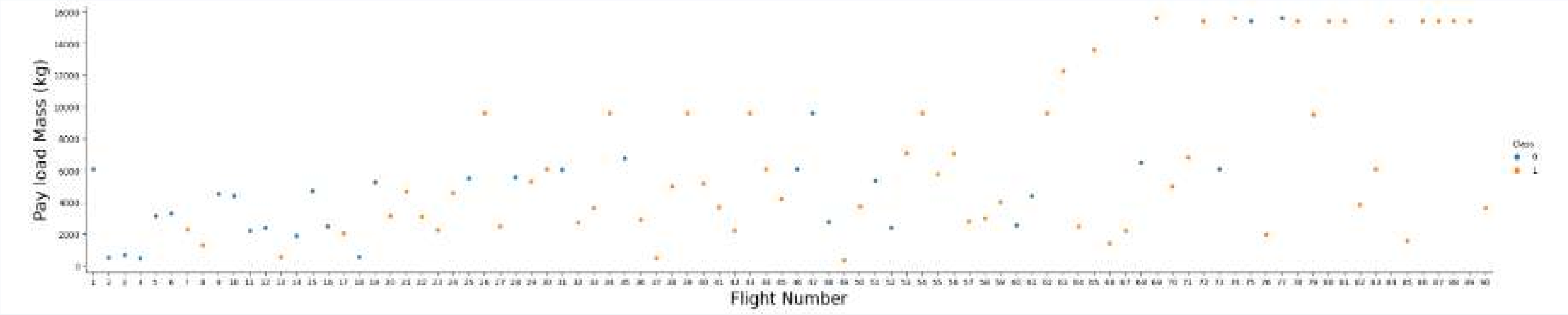
• To see the success rate change over time

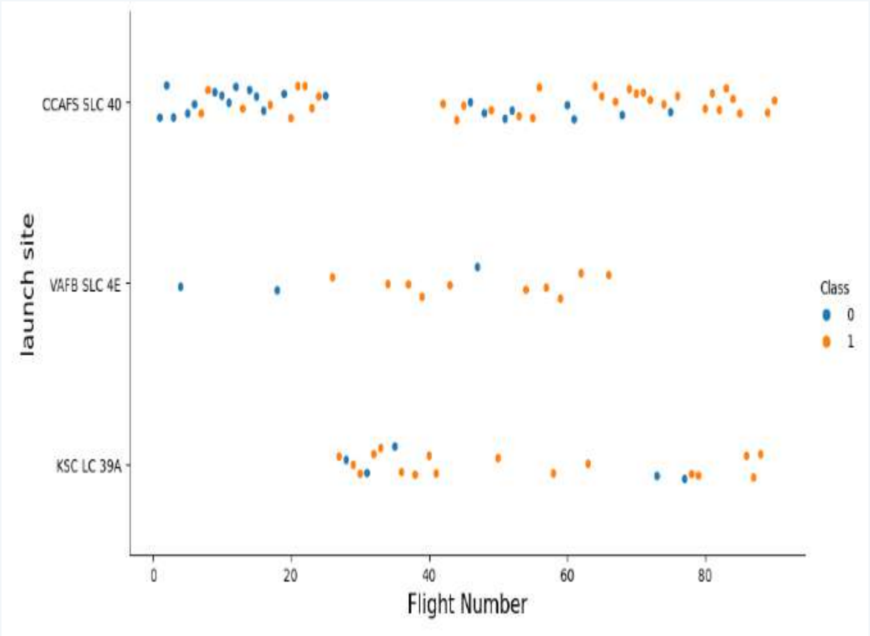


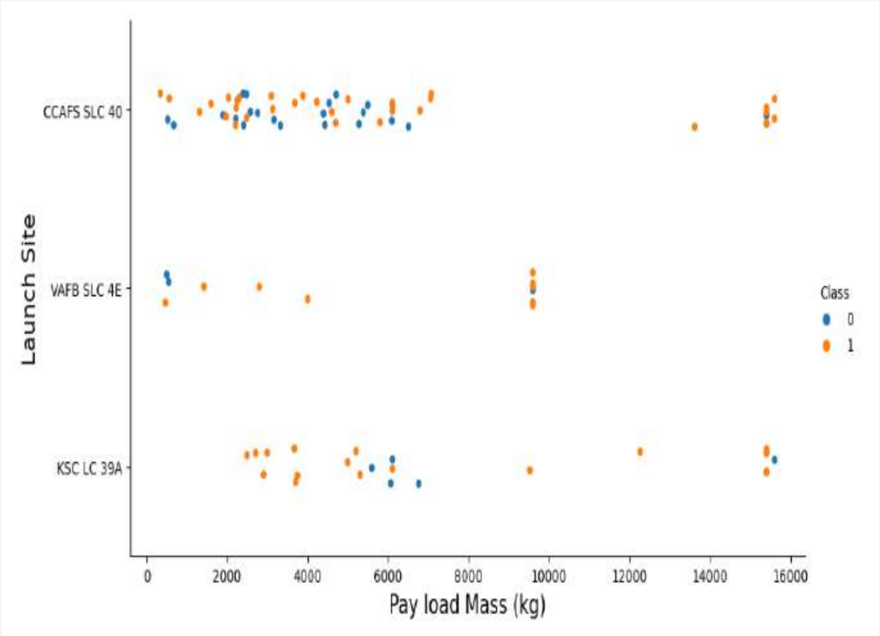

Full code:

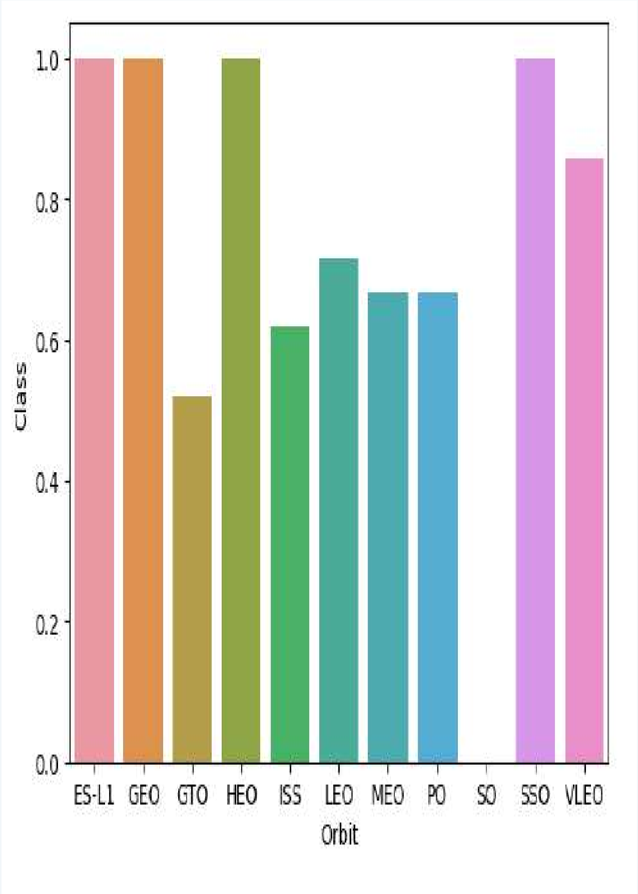
https://github.com/himaanisrivatsava/IBM-DataScience-professional-certification/blob/main/10.applied%20datascience%20capstone/10.2.1jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

# EDA with Data Visualization




Analyze the ploted bar chart try to find which orbits have high sucess rate.

# EDA with SQL

- 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 succesful landing outcome in ground pad was achieved
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000kg but less than 6000kg
- List the total number of successful and failure mission outcomes
- List the names of the booster versions which have carried the maximum payload mass
- List the records which will display the month names, failure landing outcomes in drone ship, booster versions, and launch site for the months in year 2015
- Rank the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order

Full Code: https://github.com/himaanisrivatsava/IBM-DataScience-professional-certification/blob/main/10.applied%20datascience%20capstone/10.2.2jupyter-labs-eda-sql-coursera_sqllite.ipynb

## Task 1

Display the names of the unique launch sites in the space mission

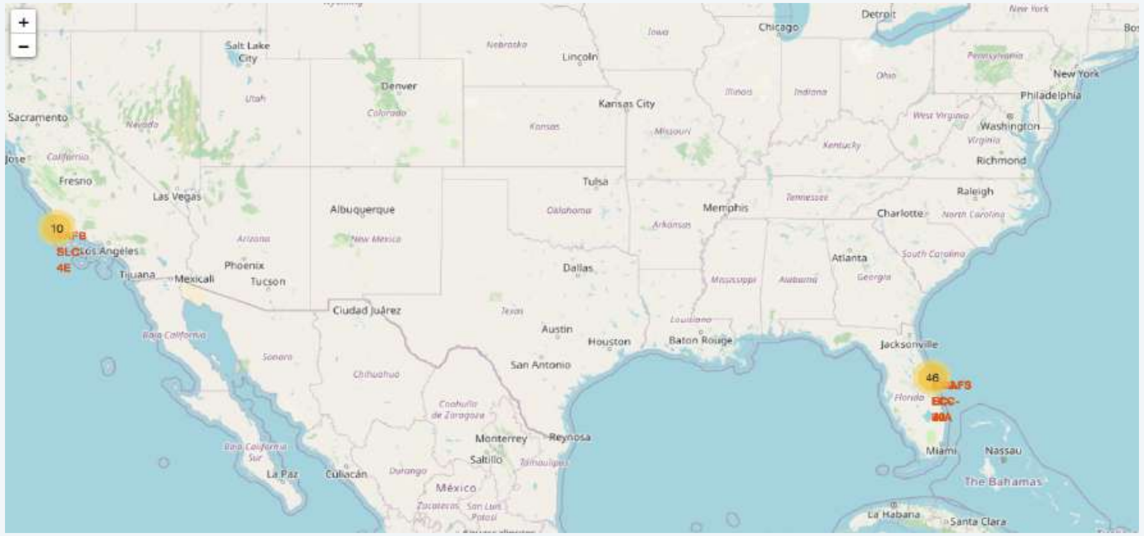
In [7]:
```
%sql Select distinct Launch_Site from SPACEXTBL;
```

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

Out[7]:

| Launch_Site |
| --- |
| CCAFS LC-40 |
| VAFB SLC-4E |
| KSC LC-39A |
| CCAFS SLC-40 |
| None |

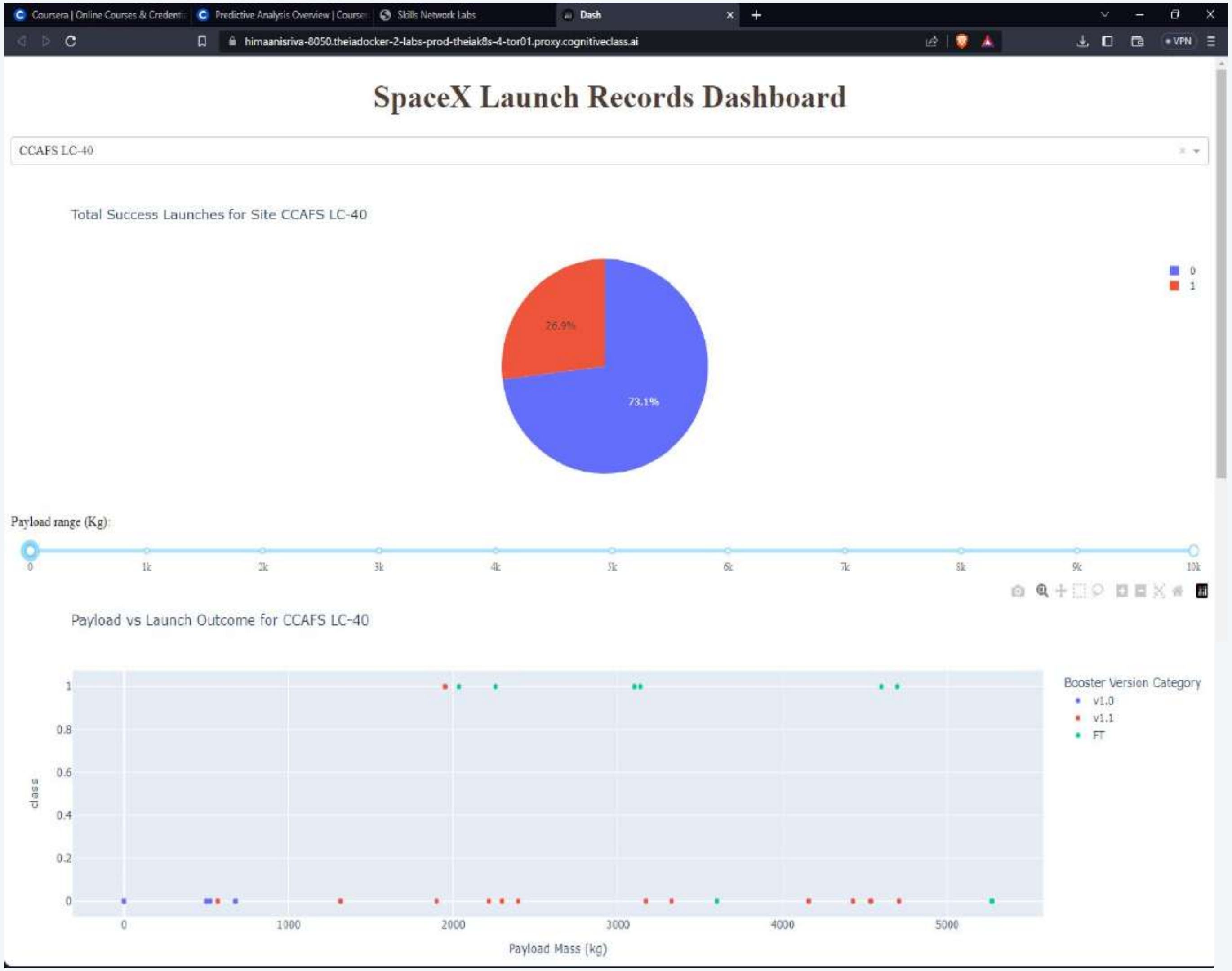
# Build an Interactive Map with Folium





- Folium maps mark Launch Sites, successful and unsuccessful landings, and a proximity example  to key locations: Railway, Highway, Coast, andCity.
- This allows us to understand why launch sites may be located where they are. Also visualizes successful landings relative to location.
- Full notebook: https://github.com/himaanisrivatsava/IBM-DataScience-professional-certification/blob/main/10.applied%20datascience%20capstone/10.3.1%20IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

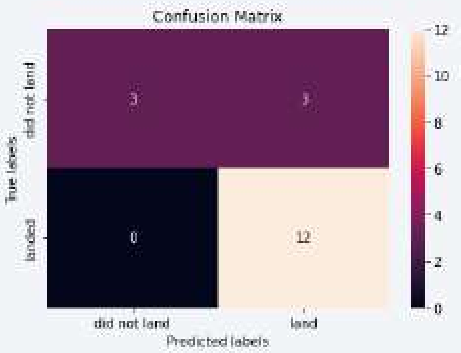
# Build a Dashboard with Plotly Dash




Built a dropdown menu with options to select all launch sites or one at a time
- Included a pie chart graphing the total success launches
- To compare which sites had the best success rates and what those rates were
- Included a slider to allow selection of the payload mass range (0kg to 10000kg)
- Included a scatter plot graphing the payload mass vs the success rate by
booster version
- To see the relationship between payload mass and launch success

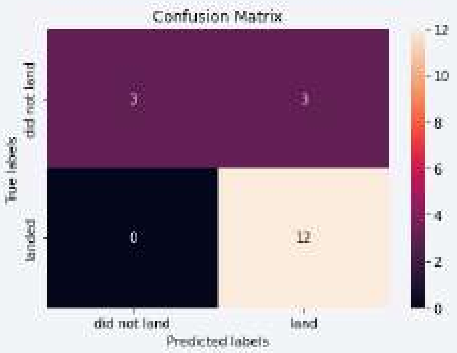
Full Code:
https://github.com/himaanisrivatsava/IBM-DataScience-professional-certification/blob/main/10.applied%20datascience%20capstone/10.3.2_spacex_dash_app.py

# Predictive Analysis (Classification)



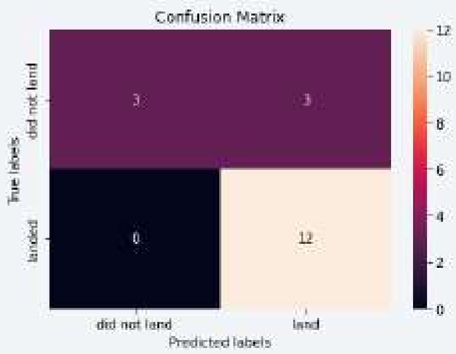

LogReg

Accuracy: 83.33%



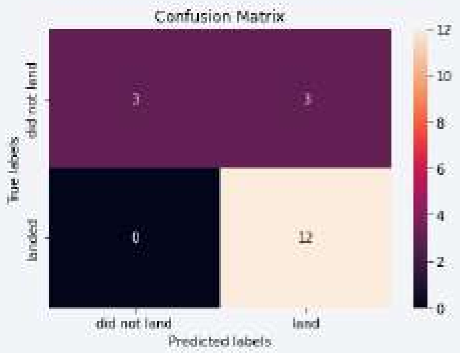

SVM

Accuracy: 83.33%




Decision Tree

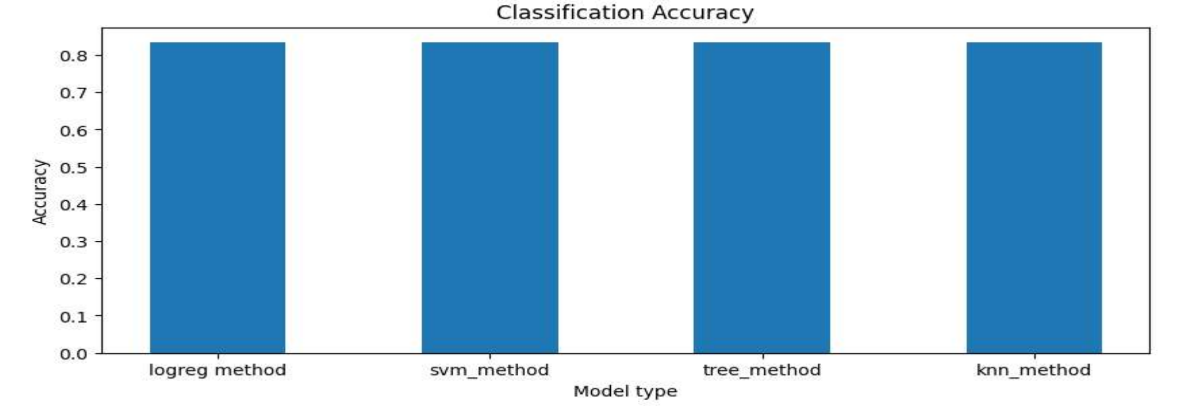
Accuracy: 83.33%




KNN

Accuracy: 83.33%

Results notebook link:
https://github.com/himaanisrivatsava/IBM-DataScience-professional-certification/blob/main/10.applied%20datascience%20capstone/10.4.1IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb



# Results

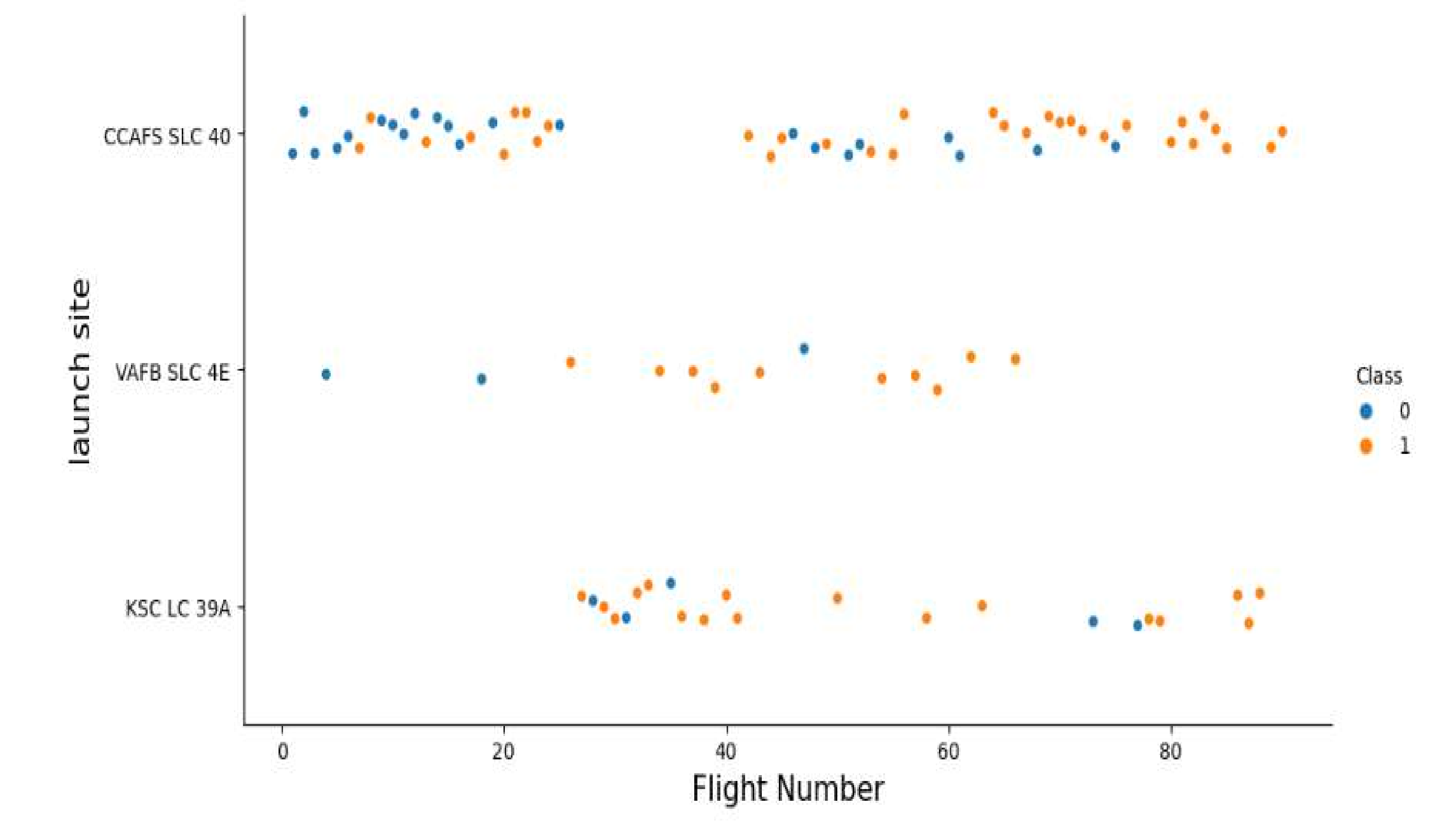
- Exploratory data analysis results
    - The chances of the launch being successful increases with flight number and with a greater payload mass
    - Orbit types ES-L1, GEO, HEO, SSO have the greatest success rates (100%)
    - The chances of the launch being successful increased overtime from 2013 to 2020
    - Site KSC LC-39A had the greatest success rate, and site CCAFS SLC-40 had the lowest
- Interactive analytics demo in screenshots
    - Launch sites are close enough to railways, highways, and cities that workers and materials can be transported, but not close enough to cause any public damage
- Predictive analysis results
- All four models used (logistic regression, SVM, decision tree, and KNN) were all equally good in terms of prediction accuracy

Section 2

# Insights drawn from EDA

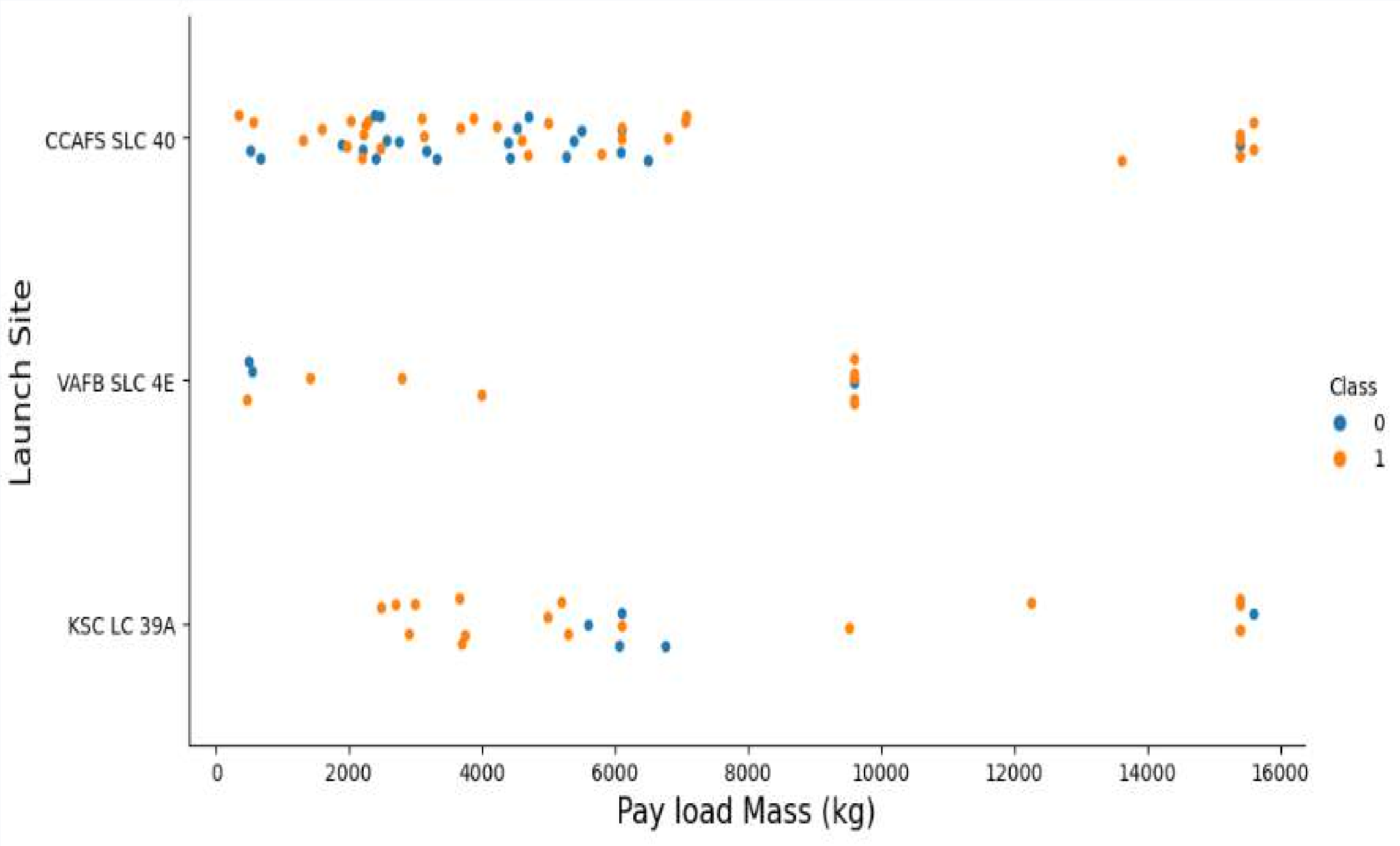
# Flight Number vs. Launch Site

- Blue indicates successful launch and orange indicates unsuccessfullaunch. Graphic suggests an increase in success rate over time (indicated in Flight Number). Likely a big breakthrough around flight 20 which significantly increased success rate.
- CCAFS appears to be the main launch site as it has the mostvolume.



# Payload vs. Launch Site

- Blue indicates successful launch and orange indicates unsuccessfullaunch.
- Payload mass appears to fall mostly between 0-6000 kg.
- Different launch sites also seem to use different payloadmass.



# Success Rate vs. Orbit Type

The ranked success rate by orbit type is as

follows:

1. ES-L1, GEO, HEO, SSO (100%)

2. VLEO

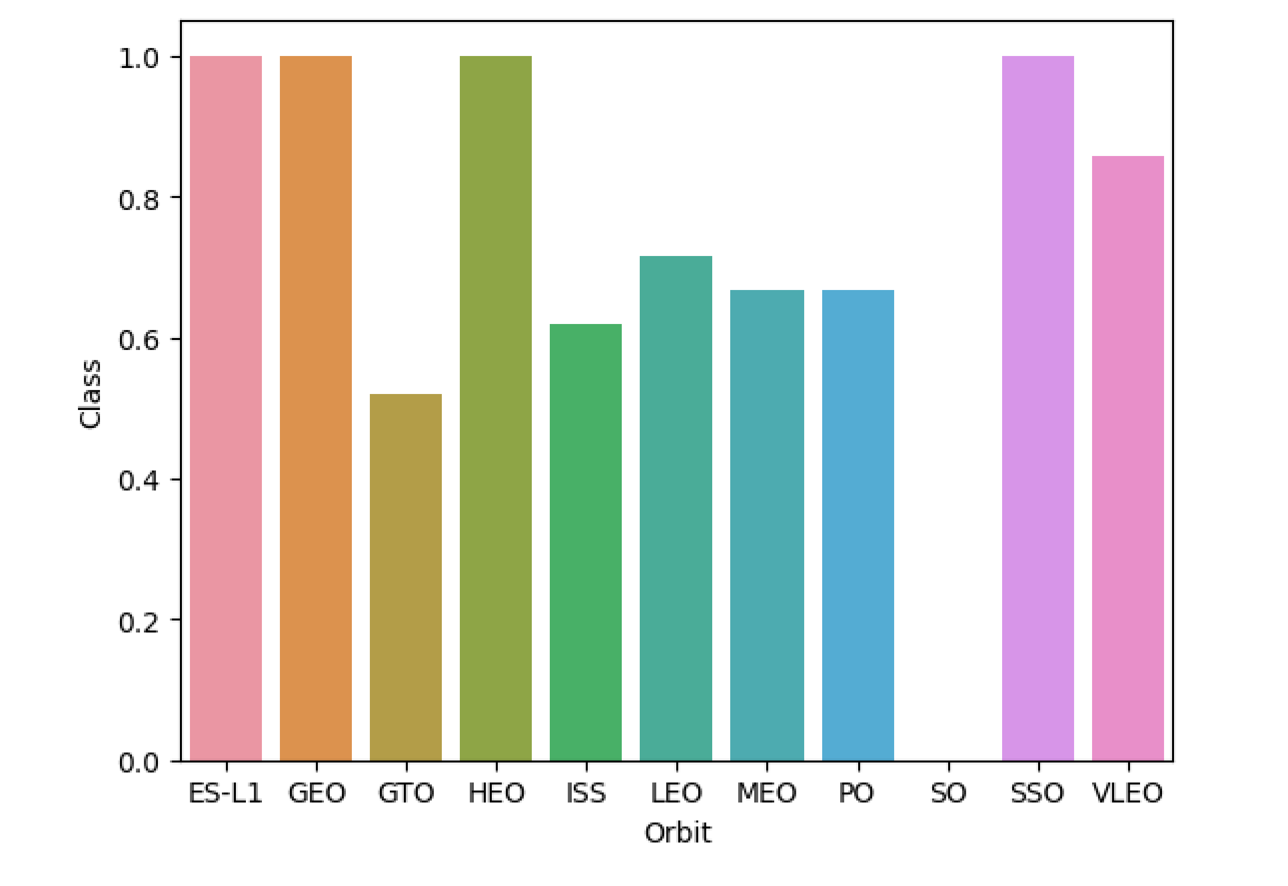
3. LEO

4. MEO, PO

5. ISS

6. GTO

7. SO (0%)



# Flight Number vs. Orbit Type

Given Class 0 (blue) represents failed

launches and Class 1 (orange) represents

successful launches, the following conclusions

can be drawn:

• Typically, the greater the flight number, the

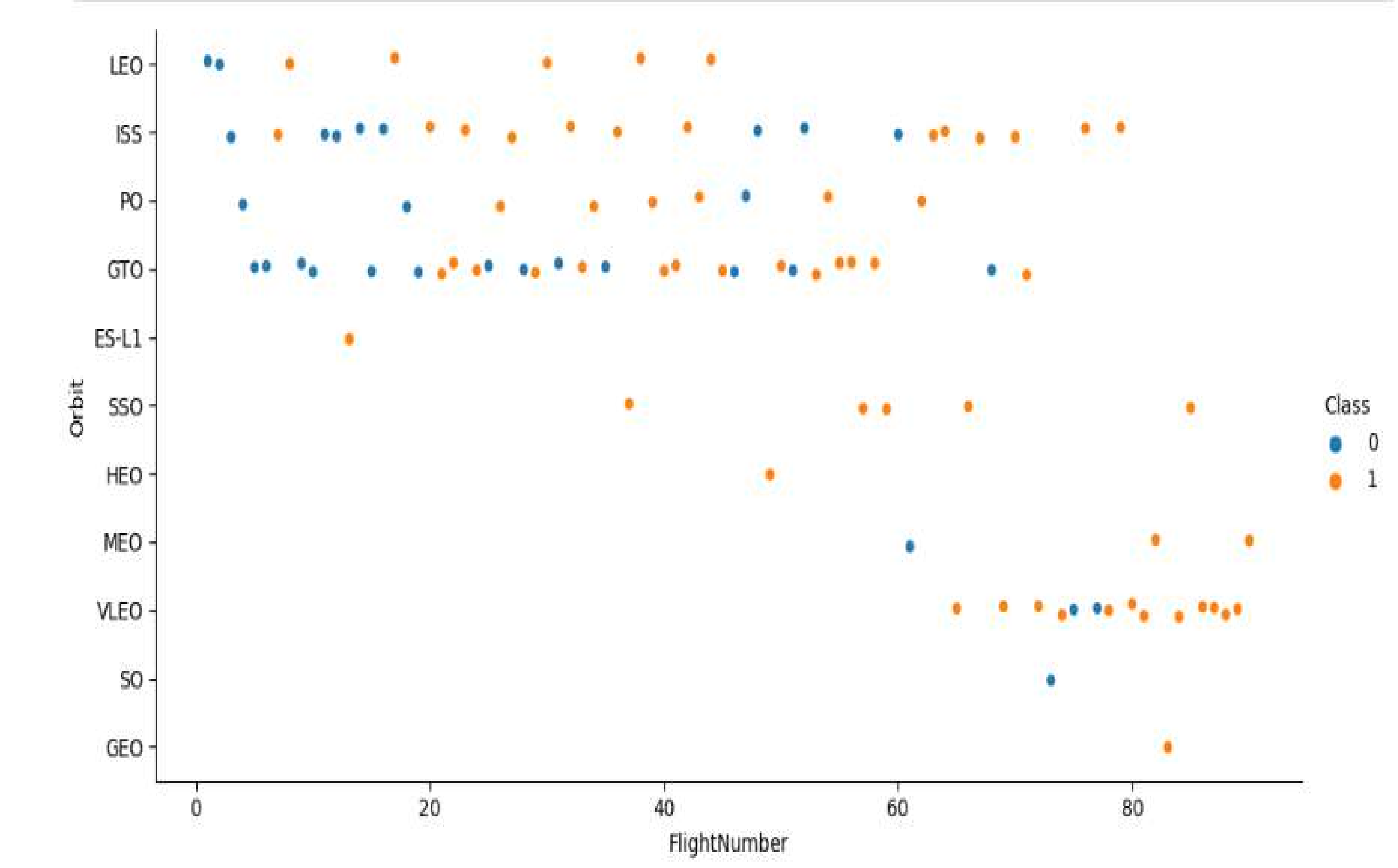
higher the success rate, with the exception

of the GTO orbit

• ES-L1, HEO, SO, and GEO only contain one

flight number, so no accurate conclusions

can be drawn for these orbits



# Payload vs. Orbit Type

Given Class 0 (blue) represents failed

launches and Class 1 (orange) represents

successful launches, the following conclusions

can be drawn:

• Greater payload masses have higher

success for LEO, ISS, PO, and VLEO

• GTO has approximately an even split of

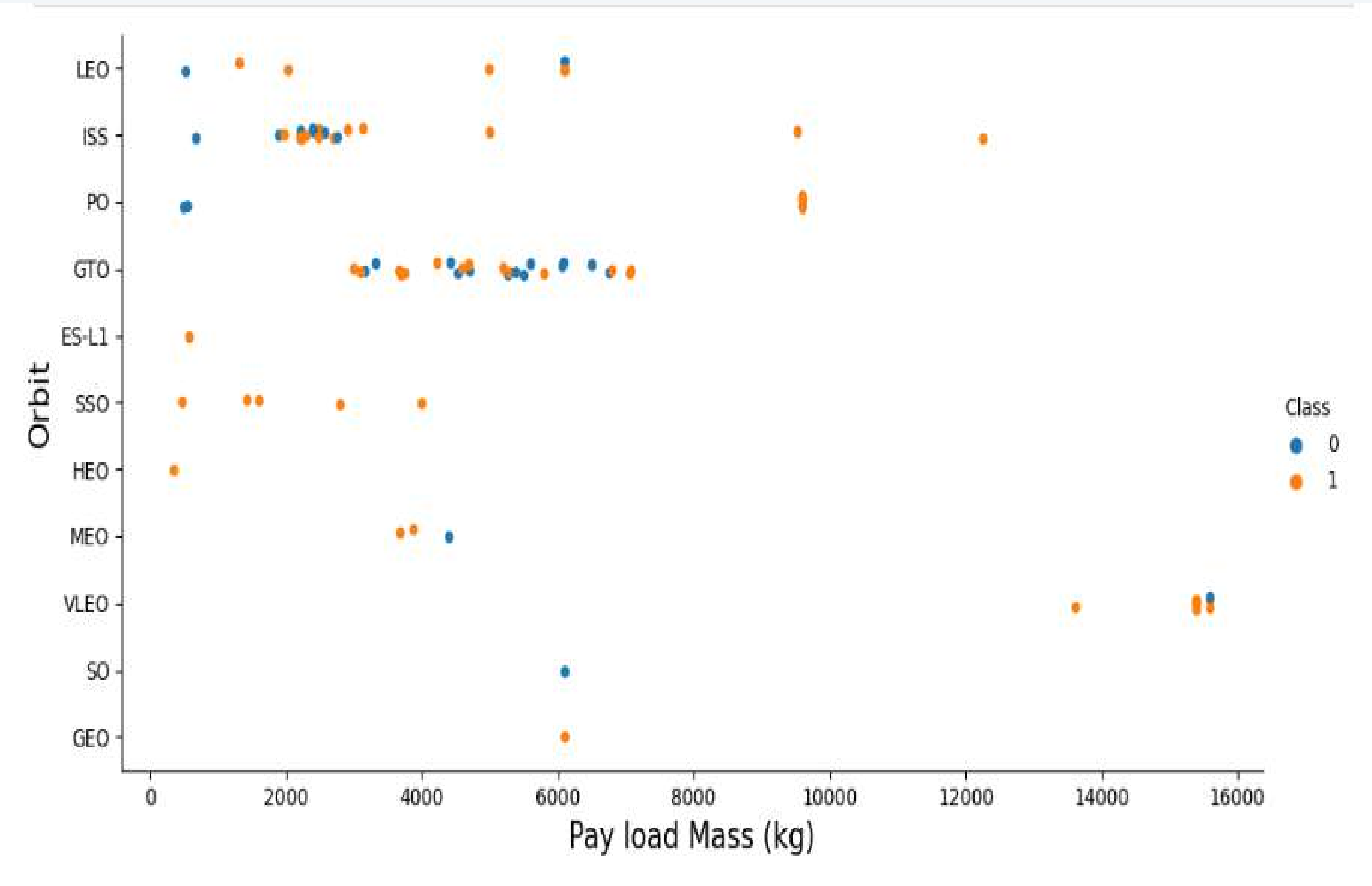
successes and failures regardless of

payload mass

• ES-L1, HEO, SO, and GEO only contain one

data point, so no accurate conclusions can

be drawn for these orbits



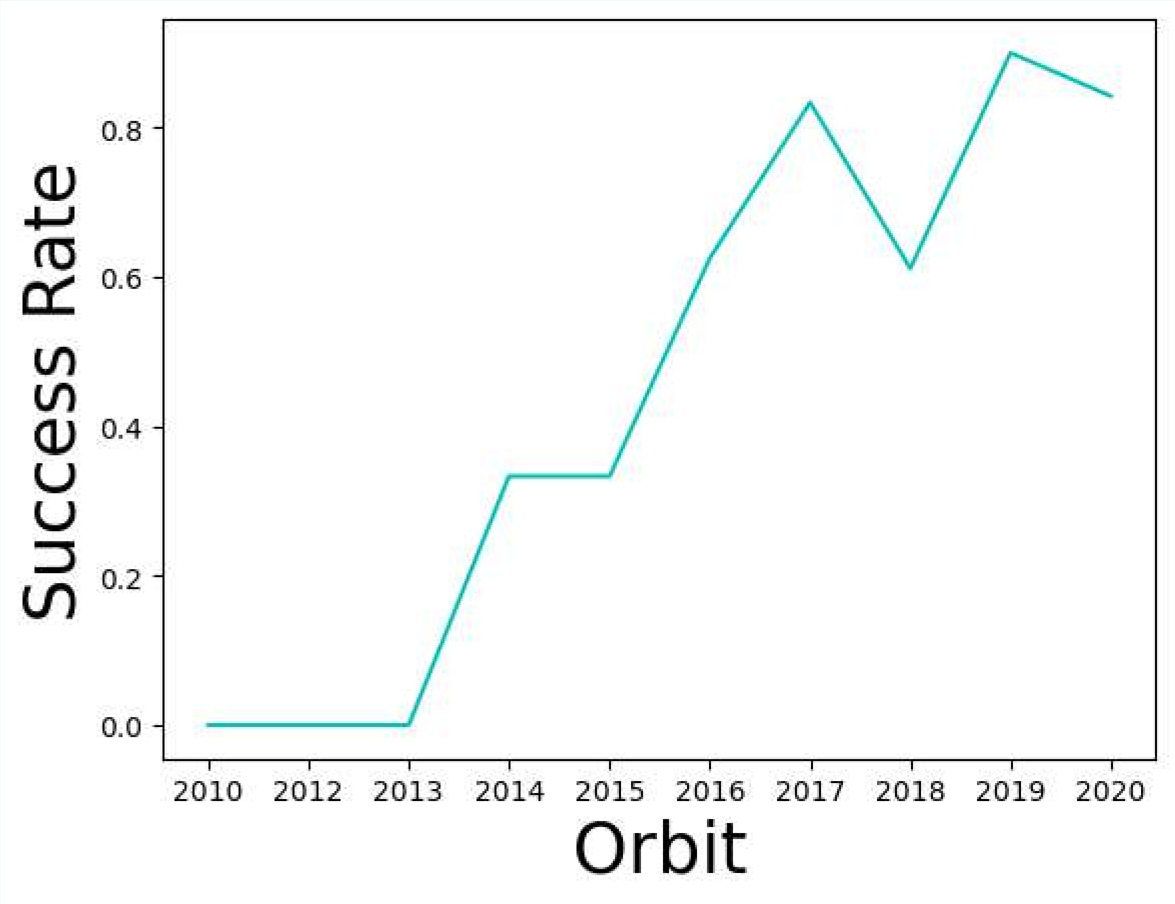
# Launch Success Yearly Trend

The following conclusions can be drawn:

• There is a general upwards trend from 2013-2020

• There is a decrease from 2017-2018, and2019-2020

• There is no change from 2014-2015



# All Launch Site Names

The query used the keyword DISTINCT

in order to only list the unique launch

site names. The result was 4 sites:

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

## Task 1

Display the names of the unique launch sites in the space mission

```
In [7]: %sql Select distinct Launch_Site from SPACEXTBL;
```

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

Out[7]:

| Launch_Site |
|---|
| CCAFS LC-40 |
| VAFB SLC-4E |
| KSC LC-39A |
| CCAFS SLC-40 |
| None |

# Launch Site Names Begin with 'CCA'

### Task 2

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

In [8]:
```
%sql Select * from SPACEXTBL where Launch_Site LIKE 'CCA%' limit 5
```

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

Out[8]:

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer | Mission_Outcome | Landing_Outc |
|---|---|---|---|---|---|---|---|---|---|
| 06/04/2010 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0.0 | LEO | SpaceX | Success | Failure (paracl |
| 12/08/2010 | 15:43:00 | F9 v1.0 B0004 | CCAFS LC-40 | Dragon demo flight C1, two CubeSats, barrel of Brouere cheese | 0.0 | LEO (ISS) | NASA (COTS) NRO | Success | Failure (paracl |
| 22/05/2012 | 7:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525.0 | LEO (ISS) | NASA (COTS) | Success | No att |
| 10/08/2012 | 0:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500.0 | LEO (ISS) | NASA (CRS) | Success | No att |
| 03/01/2013 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677.0 | LEO (ISS) | NASA (CRS) | Success | No att |

- The query used LIMIT 5 to only list 5 launch sites with names beginning with ‘CCA’.

# Total Payload Mass

Display the total payload mass carried by boosters launched by NASA (CRS)

In [9]:
```
%sql Select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer='NASA (CRS)';
```

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

Out[9]:

| sum(PAYLOAD_MASS__KG_) |
|---|
| 45596.0 |

The SUM function was used to add all the payload masses, to get a total payload mass of 45596.0 kg.

# Average Payload Mass by F9 v1.1

The AVG function was used to calculate the average payload mass, with the specific qualification of having the booster version F9 v1.1. The average payload mass was calculated to be 2534.666 kg.

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

In [10]:
```
%sql Select avg(PAYLOAD_MASS__KG_) as payavg from SPACEXTBL where Booster_Version like 'F9 v1.1%';
```

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

Out[10]:

| payavg |
| --- |
| 2534.6666666666665 |

# First Successful Ground Landing Date

List the date when the first succesful landing outcome in ground pad was acheived.

*Hint:Use min function*

In [13]:

```
%sql Select min(date) as first from SPACEXTBL where Landing_outcome='Success (ground pad)' ;
```

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

Out[13]:

| first |
|---|
| 01/08/2018 |

The MIN function was used on the Date column, specifying a landing outcome of a successful ground landing, in order to find the date of the first successful ground landing date. This was found to be 01/08/2018.

# Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [15]:
```
%sql select booster_version from spacextbl where Landing_outcome='Success (drone ship)' and payload_mass__kg_>4000 and paylo
```

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

Out[15]:

| Booster_Version |
|---|
| F9 FT B1022 |
| F9 FT B1026 |
| F9 FT B1029.1 |
| F9 FT B1021.2 |
| F9 FT B1036.1 |
| F9 B4 B1041.1 |
| F9 FT B1031.2 |

7 boosters were found which had successful drone ship landings, as well as a payload mass between 4000kg and 6000kg

# Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

In [16]:
```
%sql select count(mission_outcome),mission_outcome from spacextbl group by mission_outcome;
```

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

Out[16]:

| count(mission_outcome) | Mission_Outcome |
|---:|---:|
| 0 | None |
| 1 | Failure (in flight) |
| 98 | Success |
| 1 | Success |
| 1 | Success (payload status unclear) |

The total number of successful and failed mission outcomes was found by counting and then grouping by mission outcomes.

# Boosters Carried Maximum Payload

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [17]:
```
%sql select booster_version from spacextbl where payload_mass__kg_=(select max(payload_mass__kg_) from spacextbl)
```

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

Out[17]:

| 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 |

12 different booster versions carrying the maximum payload mass were found using a subquery.

# 2015 Launch Records

2 records were found in 2015 with a failed landing outcome in drone ship.

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

**Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.**

In [20]:

```
%sql select substr(Date,4,2) as months, booster_version, launch_site from SPACEXTBL where landing_outcome = 'Failure (drone
```

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

Out[20]:

| months | Booster_Version | Launch_Site |
|---|---|---|
| 10 | F9 v1.1 B1012 | CCAFS LC-40 |
| 04 | F9 v1.1 B1015 | CCAFS LC-40 |

# 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.

In [21]:
```
%sql Select count(landing_outcome),landing_outcome   from SPACEXTBL group by landing_outcome order by count(landing_outcome)
```

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

Out[21]:

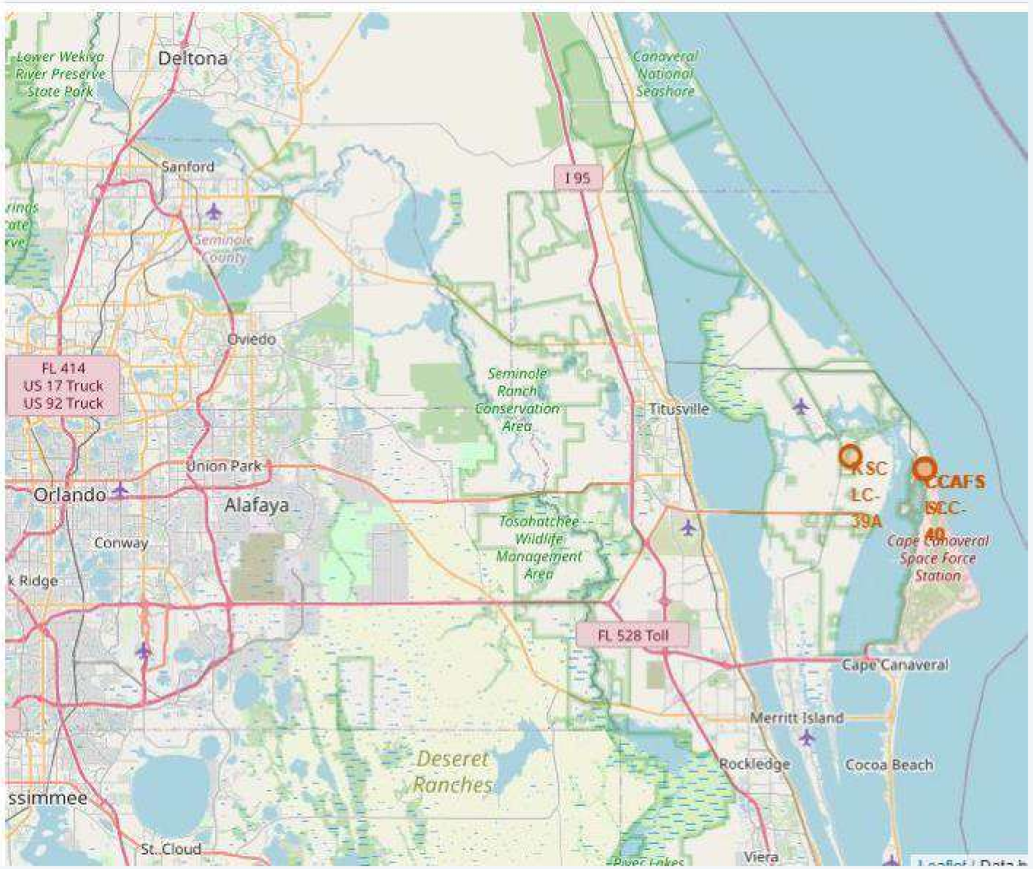
| count(landing_outcome) | Landing_Outcome |
|---|---|
| 38 | Success |
| 21 | No attempt |
| 14 | Success (drone ship) |
| 9 | Success (ground pad) |
| 5 | Failure (drone ship) |
| 5 | Controlled (ocean) |
| 3 | Failure |
| 2 | Uncontrolled (ocean) |
| 2 | Failure (parachute) |
| 1 | Precluded (drone ship) |
| 1 | No attempt |
| 0 | None |

Landing outcomes between 2010-06-04 and 2017-03-20 were ranked, by grouping by landing outcome and ordering by the number of launches for each outcome, ordered in descending order. It can be seen that success is the most common outcome, and no attempt is the least common.

Section 3

# Launch Sites Proximities Analysis

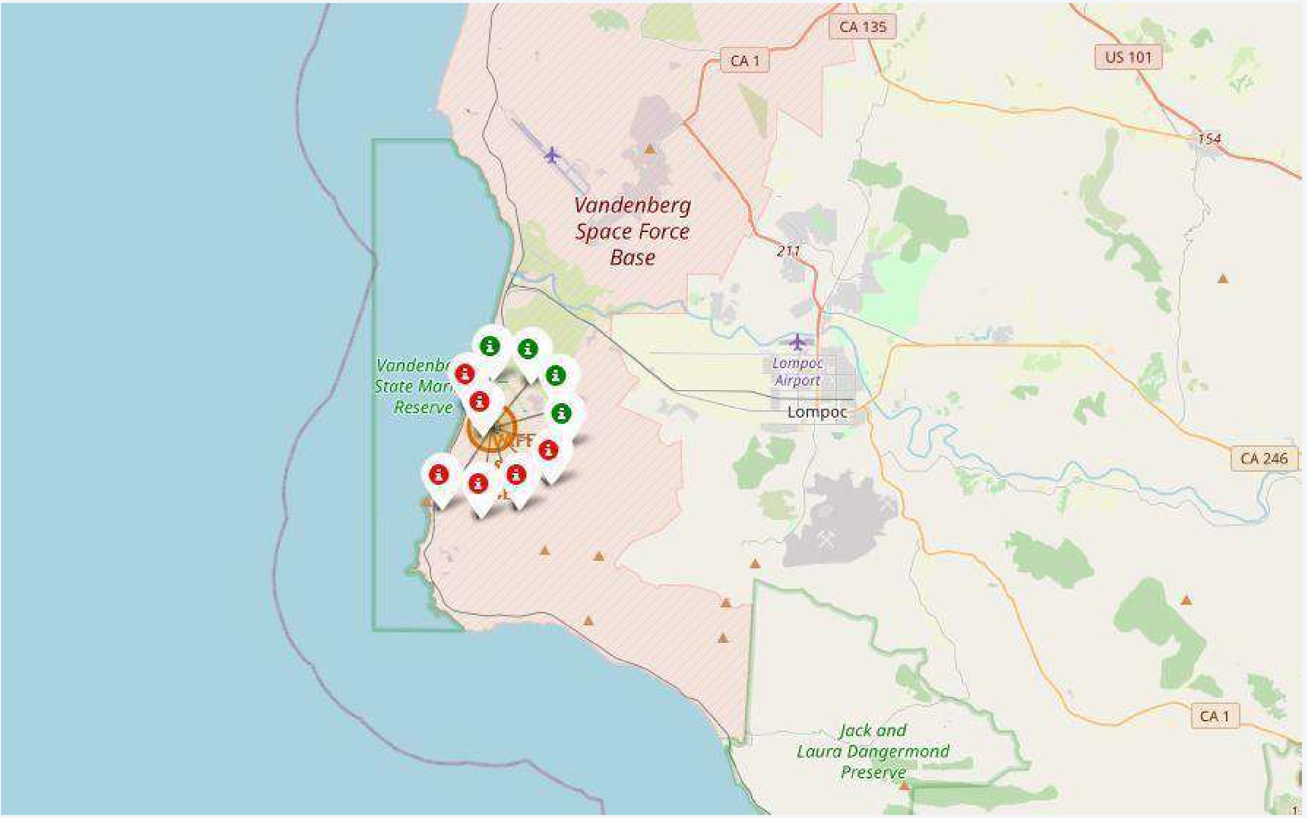
# Launch Site Locations

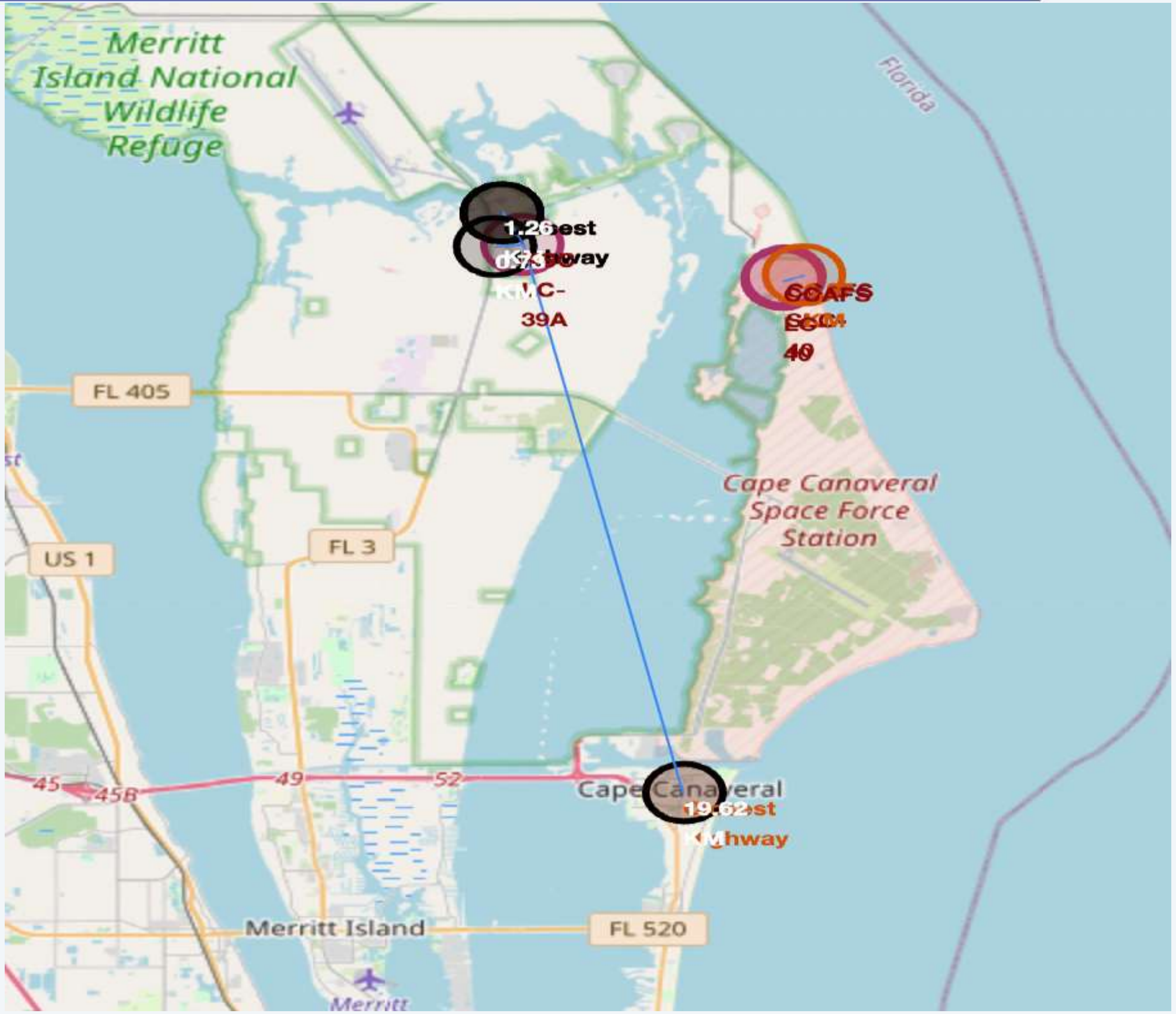

The left map shows all launch sites relative US map. The right map shows the two Florida launch  sites since they are very close to each other. All launch sites are near theocean.

# Color Coded Launch Markers

Clusters on Folium map can be clicked on to display each successful landing (green icon) andfailedlanding (red icon). In this example VAFB SLC-4E shows 4 successful landings and 6 failedlandings.



# Launch Site Proximities




- It can be seen that launch sites are close enough to railways, highways, and cities that workers and materials can be transported, but not close enough to cause any public damage.

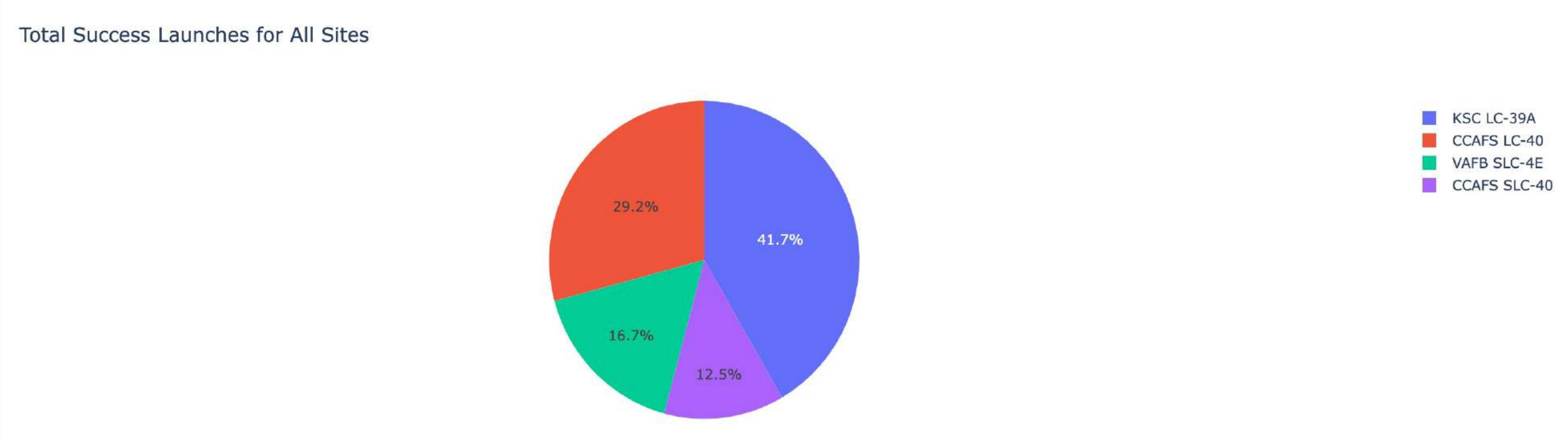
- They're also close to coasts so that if damage does occur, it'll be close to water and won't harm any people or properties.

Section 4

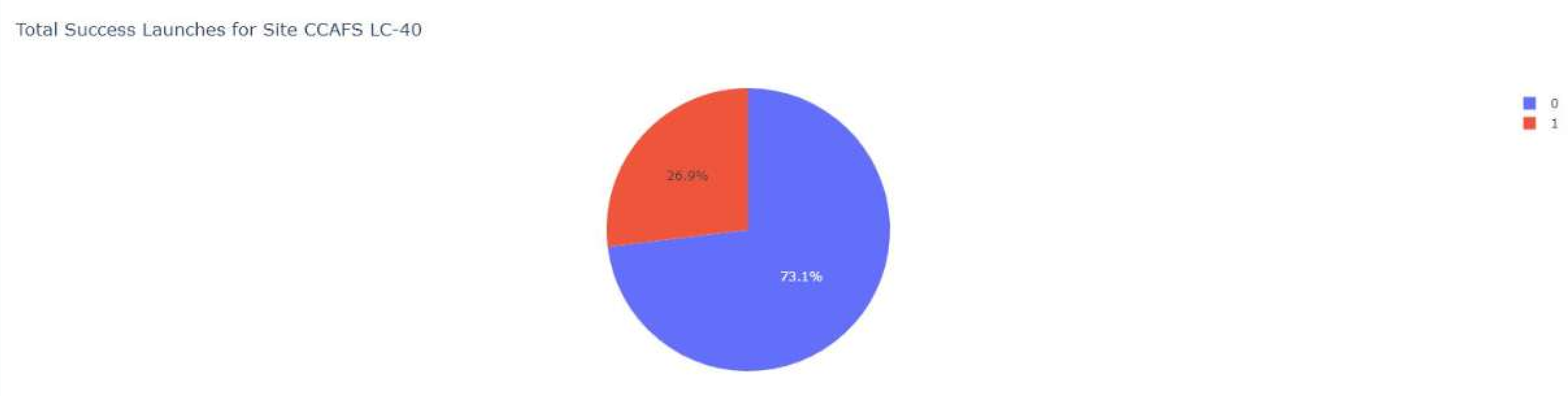
# Build a Dashboard with Plotly Dash

# Total Success Launches for All Sites

KSC LC-39A has to highest total success launches, making up for 41.7% of the total successes, and CCAFS SLC-40 has the lowest success, making up 12.5%.



# Total Success Launches for Site KSC LC-40

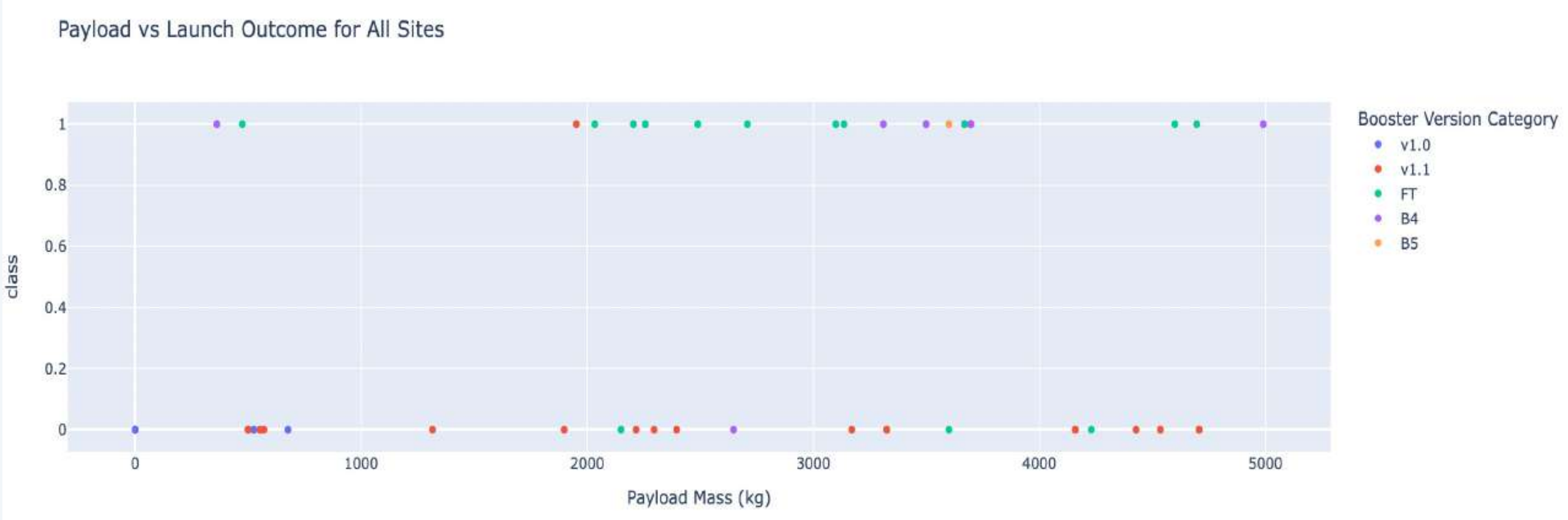


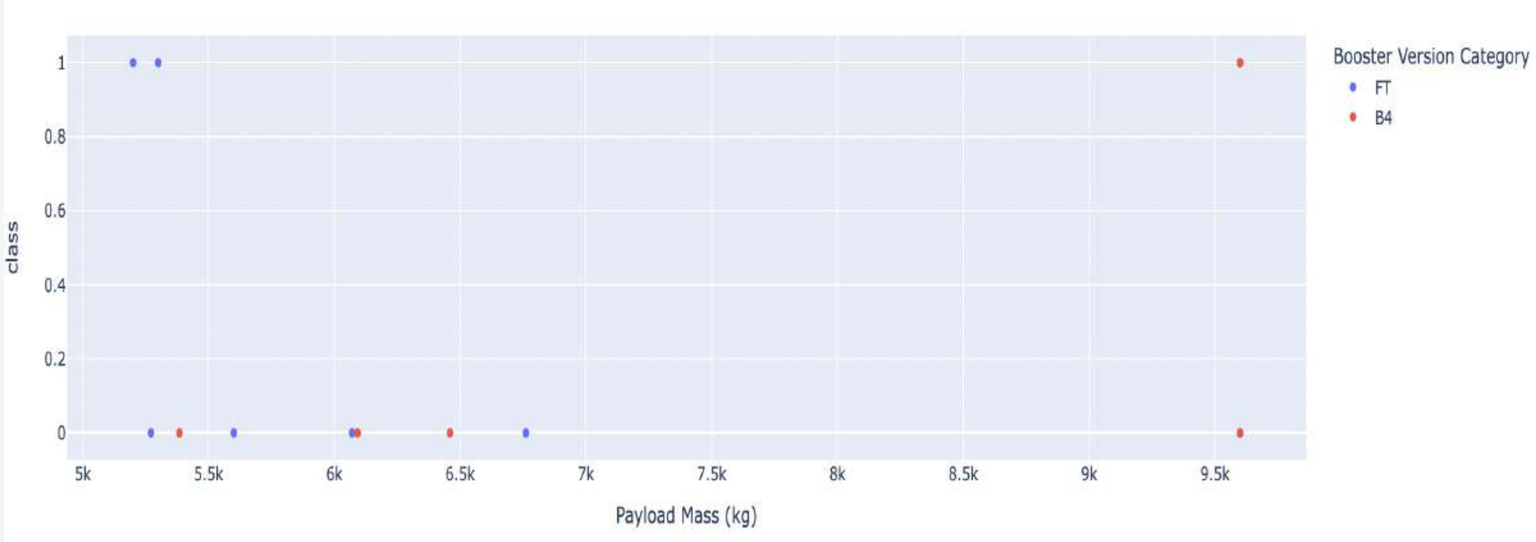

Site KSC LC-39A, the site making up the greatest number of successes, has a success rate of 73.1% and failure of 26.9%.

# Payload vs Launch Outcome for All Sites

<u>Payload vs Launch Outcome for payload mass 0kg-5000kg</u>

- It can be seen that the success rateis higher when the payload mass islower, i.e. below 5500 kg, where 1indicates success and 0 indicates failure.



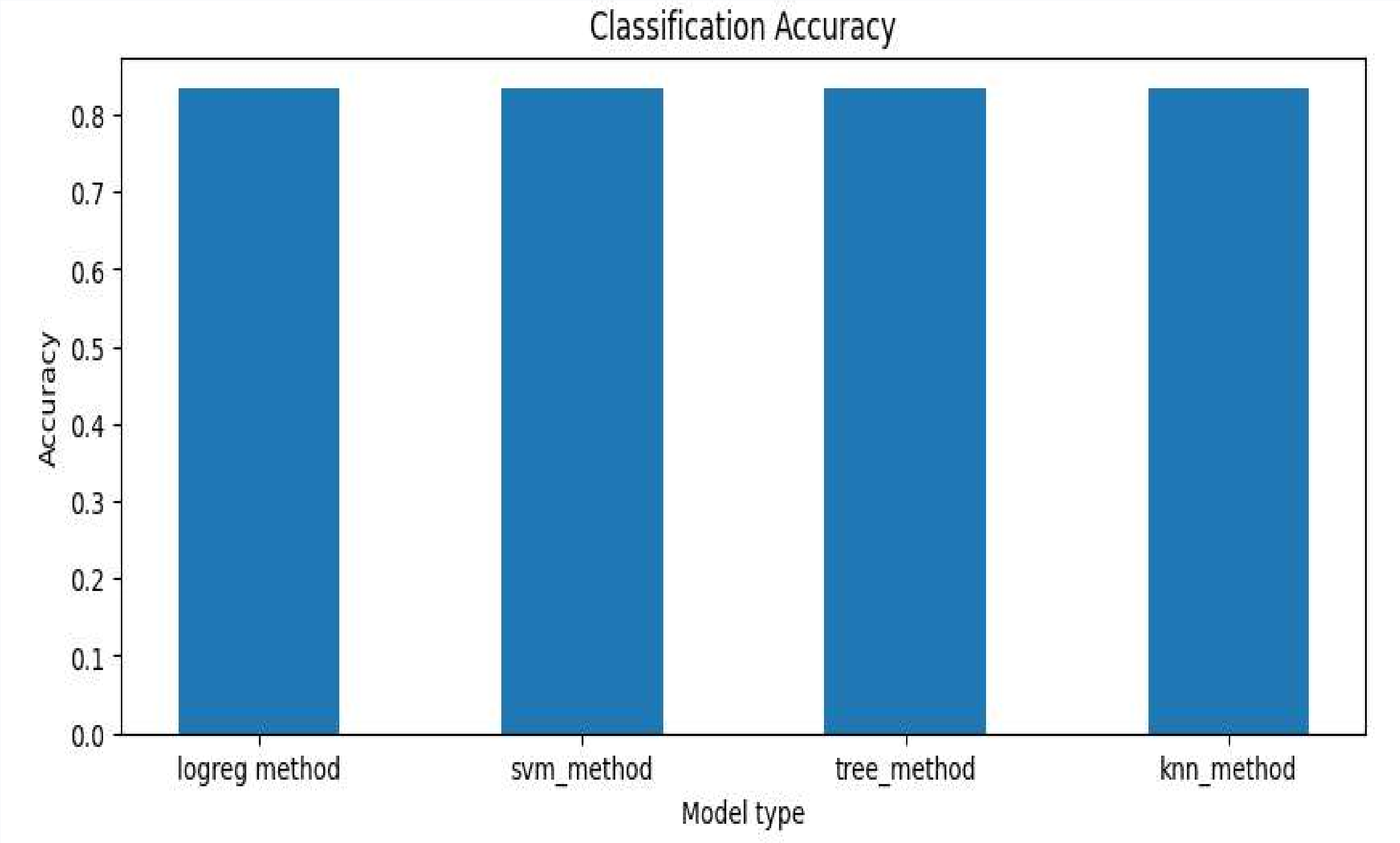

<u>Payload vs Launch Outcome for payload mass 5000kg-10000kg</u>

- It can also be seen that boosters FT and B4 are the only boosters withpay load mass above 5000kg

Section 5

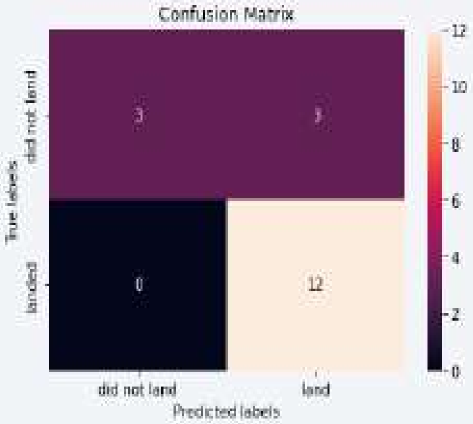
# Predictive Analysis (Classification)

# Classification Accuracy



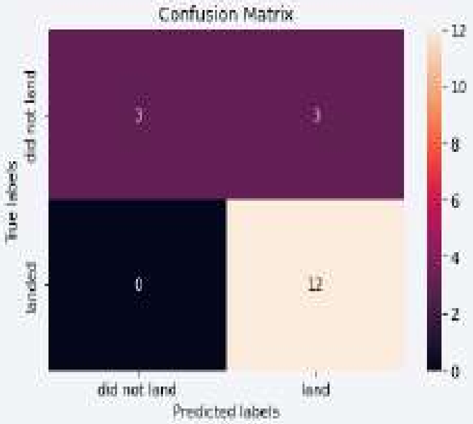

- All models had virtually the same accuracy on the test set at 83.33% accuracy. It should be noted that test size is small at only sample size of 18.
- This can cause large variance in accuracy results, such as those in Decision Tree Classifier model in repeatedruns.
- We likely need more data to determine the bestmodel.

# Confusion Matrix



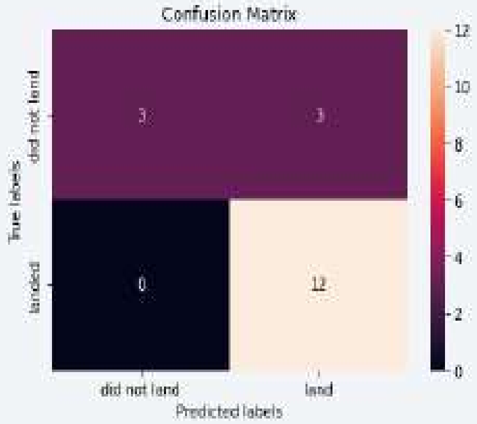

LogReg

Accuracy: 83.33%



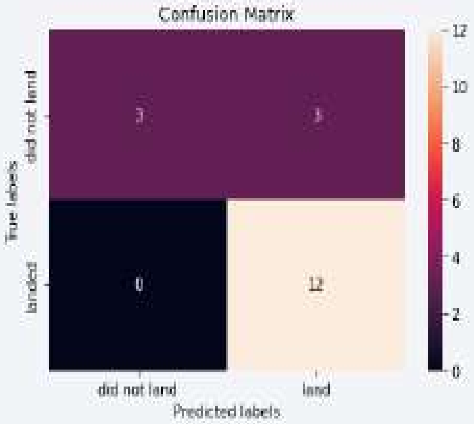

SVM

Accuracy: 83.33%




Decision Tree

Accuracy: 83.33%




KNN

Accuracy: 83.33%

- Since all models performed the same for the test set, the confusion matrix is the same across all models. The models predicted 12 successful landings when the true labelwas successful landing.
- The models predicted 3 unsuccessful landings when the true label was unsuccessfullanding.
- The models predicted 3 successful landings when the true label was unsuccessful landings (false positives). Our models over predict successfullandings.

# Conclusions

- The chances of the launch being successful increases with flight number and with a greater payload mass
- Orbit types ES-L1, GEO, HEO, SSO have the greatest success rates(100%)
- The chances of the launch being successful increased overtime from 2013 to2020
- Site KSC LC-39A had the greatest success rate, and site CCAFS SLC-40 had the lowest
- All four models used (logistic regression, SVM, decision tree, and KNN) were all equally good in terms of prediction accuracy

# Appendix

**Instructors:**
RavAhuja, Alex Aklson, AijeEgwaikhide, Svetlana Levitan, Romeo Kienzler, PolongLin, Joseph Santarcangelo, Azim Hirjani, HimaVasudevan, SaishruthiSwaminathan, Saeed Aghabozorgi, Yan Luo

**Special Thanks to All Instructors:**
**https://www.coursera.org/professional-certificates/ibm-data-science?#instructors**

**GitHub Repository link :**
**https://github.com/himaanisrivatsava/IBM-DataScience-professional-certification/tree/main/10.applied%20datascience%20capstone**

Thank you!