IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Audry Surendra
Jun 21, 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis using SQL
  - Exploratory Data Analysis using Pandas and Matplotlib
  - Interactive Visual Analytics with Folium
  - Interactive Dashboard with Plotly
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis results
  - Interactive Visual Component results
  - Predictive Analysis results

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

Executive Summary

- 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 was collected using a get request to the SpaceX API.
- 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

Data Collection – SpaceX API

- Collecting, cleaning, and wrangling the data from the SpaceX API.
- Full code: https://github.com/AS2308/IBM-datasci-capstone/blob/main/(1)%20Collecting%20the%20Data/Collecting_the_data.ipynb

1. Request data from SpaceX API

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

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

2. Decode as JSON

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

3. Clean the data

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

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

4. Data wrangling/cleaning

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

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

Data Collection - Scraping

- Web scraped the SpaceX Wikipedia page using BeautifulSoup to attain data for Falcon 9 launches
- Full code: https://github.com/AS2308/IBM-datasci-capstone/blob/main/(1)%20Collecting%20the%20Data/Webscraping.ipynb

1. Request the Falcon9 Launch HTML page as an HTTP response

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

2. Create a BeautifulSoup object

```
soup = BeautifulSoup(response, 'html.parser')
```

3. Extract all column/variable names from the HTML table header

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

```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

th_elements = first_launch_table.find_all('th')

for i in range(len(th_elements)):
    col_name = extract_column_from_header(th_elements[i])
    if (col_name is not None and len(col_name) > 0):
        column_names.append(col_name)
```

4. Create a data frame by parsing the launch 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']=[]
```

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

Data Wrangling

- Performed exploratory data analysis, calculated the launches per site and the mission outcome occurrences, and created a landing outcome column
- Full code: https://github.com/AS2308/IBM-datasci-capstone/blob/main/(2)%20Data%20Wrangling/DataWrangling.ipynb

1. Calculate the number of launches on each site

```
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
```

2. Calculate the number and occurrence of each orbit
and the number and occurrence of mission outcome per orbit type

```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

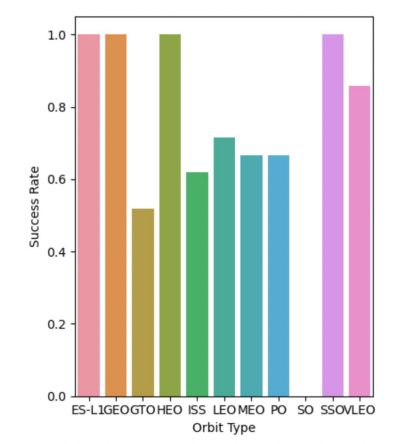
```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

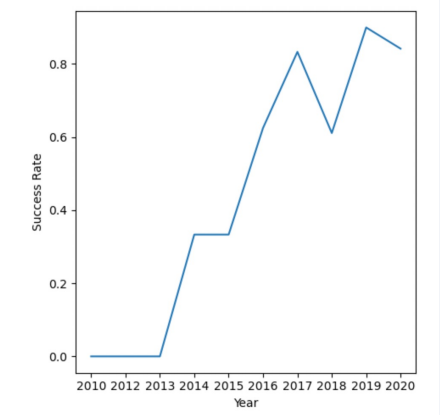
3. Create a landing outcome label

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise

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

landing_class
```

EDA with Data Visualization

- Scatterplot: Flight Number vs Payload Mass (kg)
  - To see how the Flight Number (indicating the continuous launch attempts) and Payload variables affect 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/AS2308/IBM-datasci-capstone/blob/main/(3)%20Exploratory%20Data%20Analysis%20(EDA)/EDA_Pandas_and_Matplotlib.ipynb

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/AS2308/IBM-datasci-capstone/blob/main/(3)%20Exploratory%20Data%20Analysis%20(EDA)/EDA_SQL.ipynb

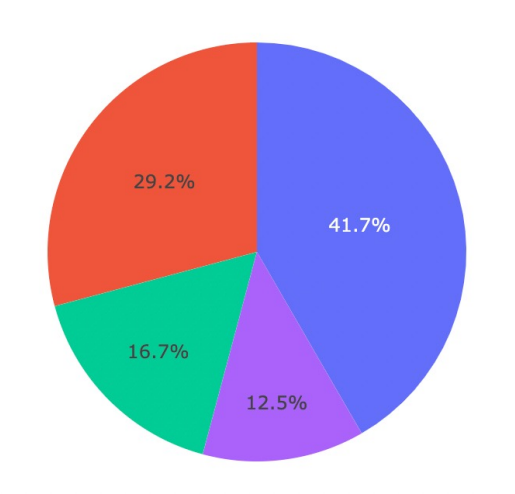
Build an Interactive Map with Folium

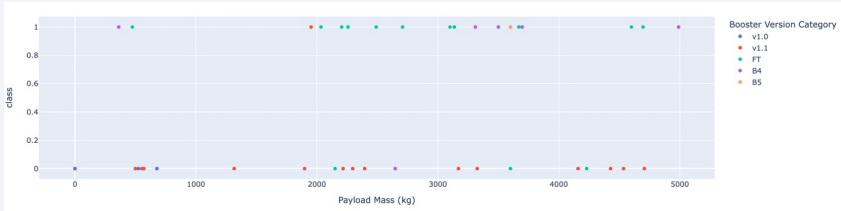
- Launch sites
  - Marked launch site locations using circles and test labels
  - Marked success/failures for each site using markers
- Distances
  - Marked distances between a launch site and the closest coastline, railway, highway, and city
  - Used circles, lines, markers, and text labels

Full Code: https://github.com/AS2308/IBM-datasci-capstone/blob/main/(4)%20Interactive%20Visual%20Analytics%20and%20Dashboard/visual_analytics_with_folium.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/AS2308/IBM-datasci-capstone/blob/main/(4)%20Interactive%20Visual%20Analytics%20and%20Dashboard/spacex_dash_app.py

Predictive Analysis (Classification)

1. Create a NumPy array.

```
Y = data['Class'].to_numpy()
```

2. Standardize the data.

```
transform = preprocessing.StandardScaler()

X = transform.fit_transform(X)
```

3. Split the data into training and test data.

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)
```

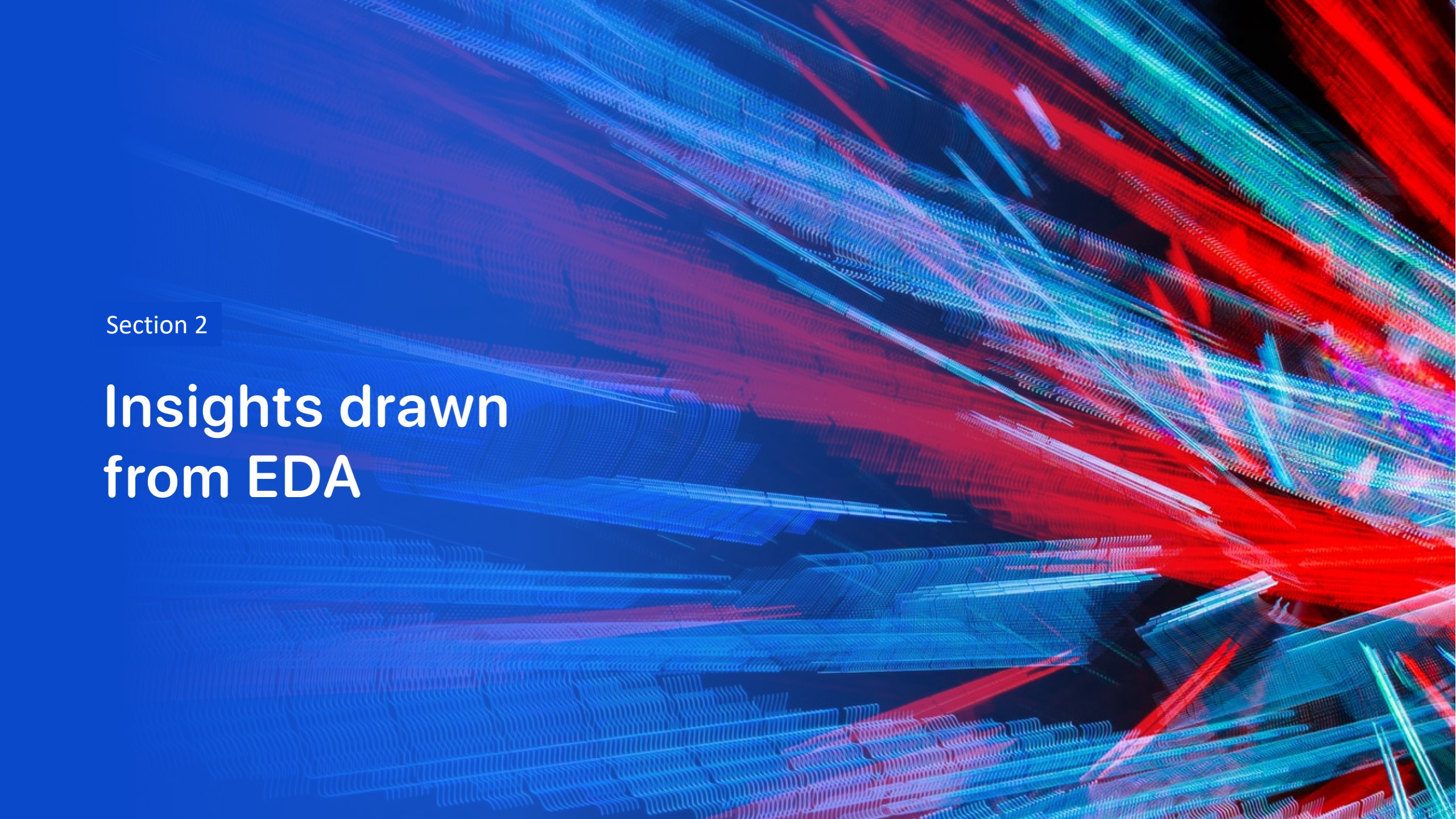
4. Create a GridSearchCV object with cv = 10. Fit the object on four different methods: logistic regression, SVM, decision tree, and KNN.
5. Calculate accuracy of the models using the score() method.
6. Create a confusion matrix for each model.
7. Determine the best model using the scores.

```
methods = {'logreg method':logreg_cv.score(X_test, Y_test),
           'svm_method':svm_cv.score(X_test, Y_test),
           'tree_method':tree_cv.score(X_test, Y_test),
           'knn_method':knn_cv.score(X_test, Y_test)}

best_method = max(methods.values())
print("Max accuracy:",best_method)
```

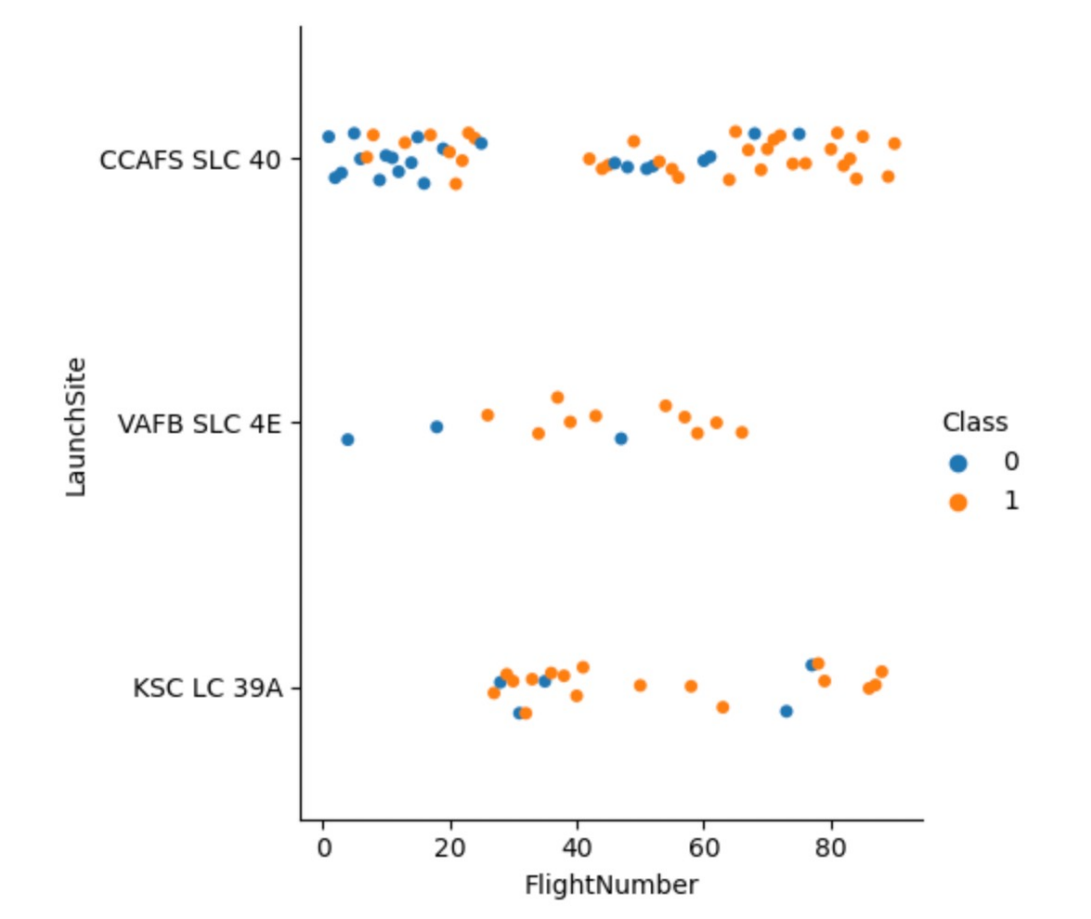
Full Code:
https://github.com/AS2308/IBM-datasci-capstone/blob/main/(5)%20Predictive%20Analysis%20-%20Classification/machine_learning_prediction.ipynb

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

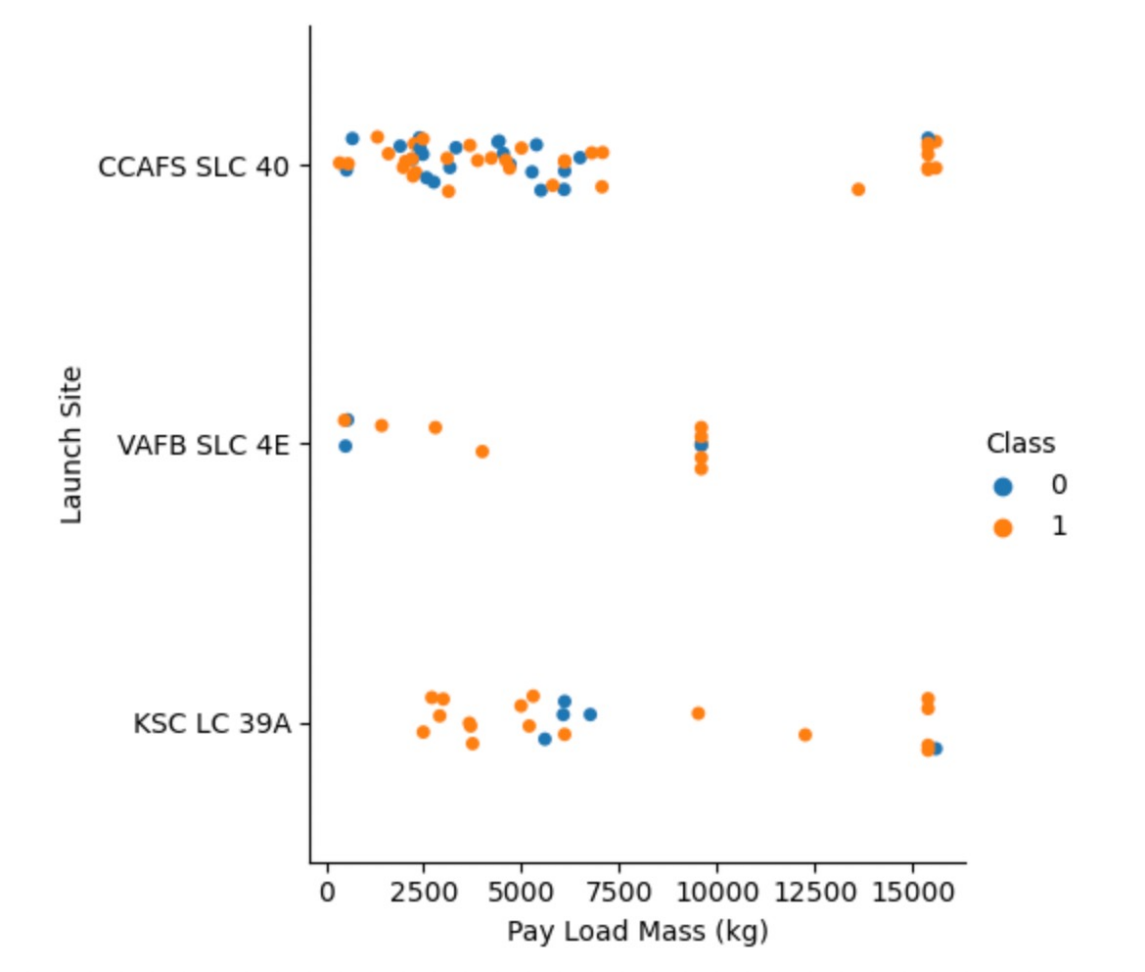
Flight Number vs. Launch Site



Given Class 0 (blue) represents failed launches and Class 1 (orange) represents successful launches, the following conclusions can be drawn:

- Earlier flights have a greater number of failures compared to later flights
- The greatest number of launches were from CCAFS SLC 40
- VAFB SLC 4E and KSC LC 39A have greater success rates than CCAFS SLC 40

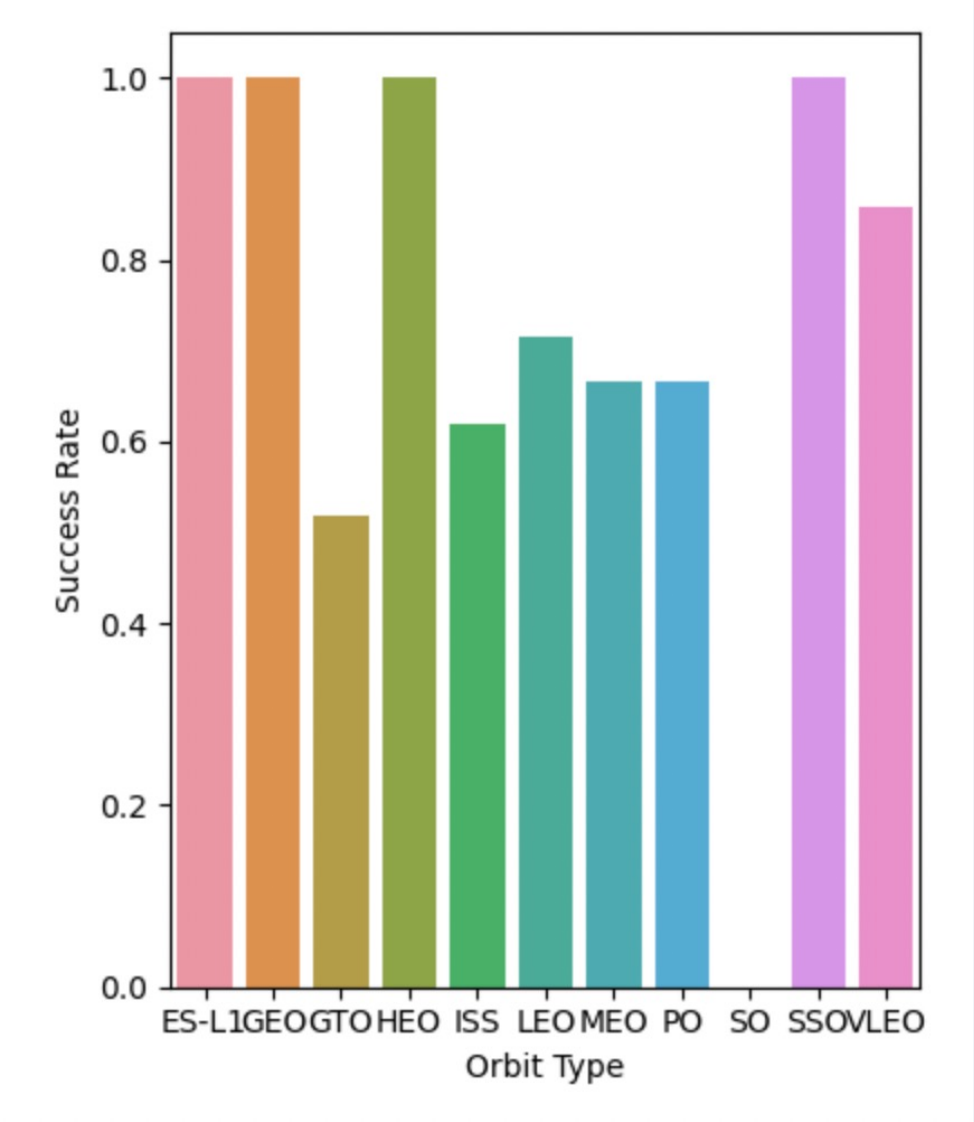
Payload vs. Launch Site



Given Class 0 (blue) represents failed launches and Class 1 (orange) represents successful launches, the following conclusions can be drawn:

- The greater the pay load mass, the greater the chance of success, with pay load masses greater than 7500kg having the best success
- Launches with mass less than 7500kg have approximately an equal chance of failing or succeeded for site CCAFS SLC 40
- Site VAFB SLC 4E did not launch anything greater than 10000kg

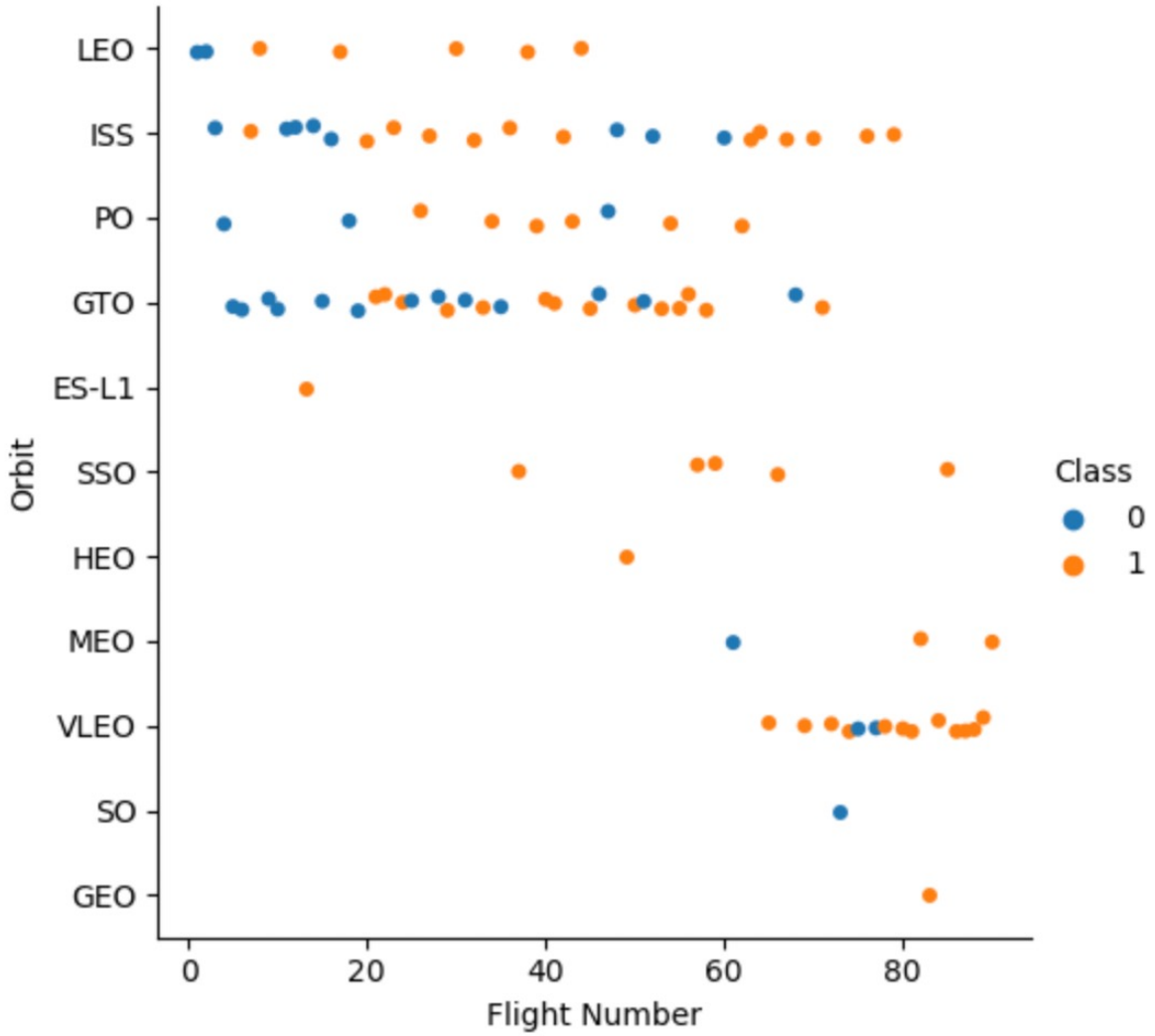
Success Rate vs. Orbit Type



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%)

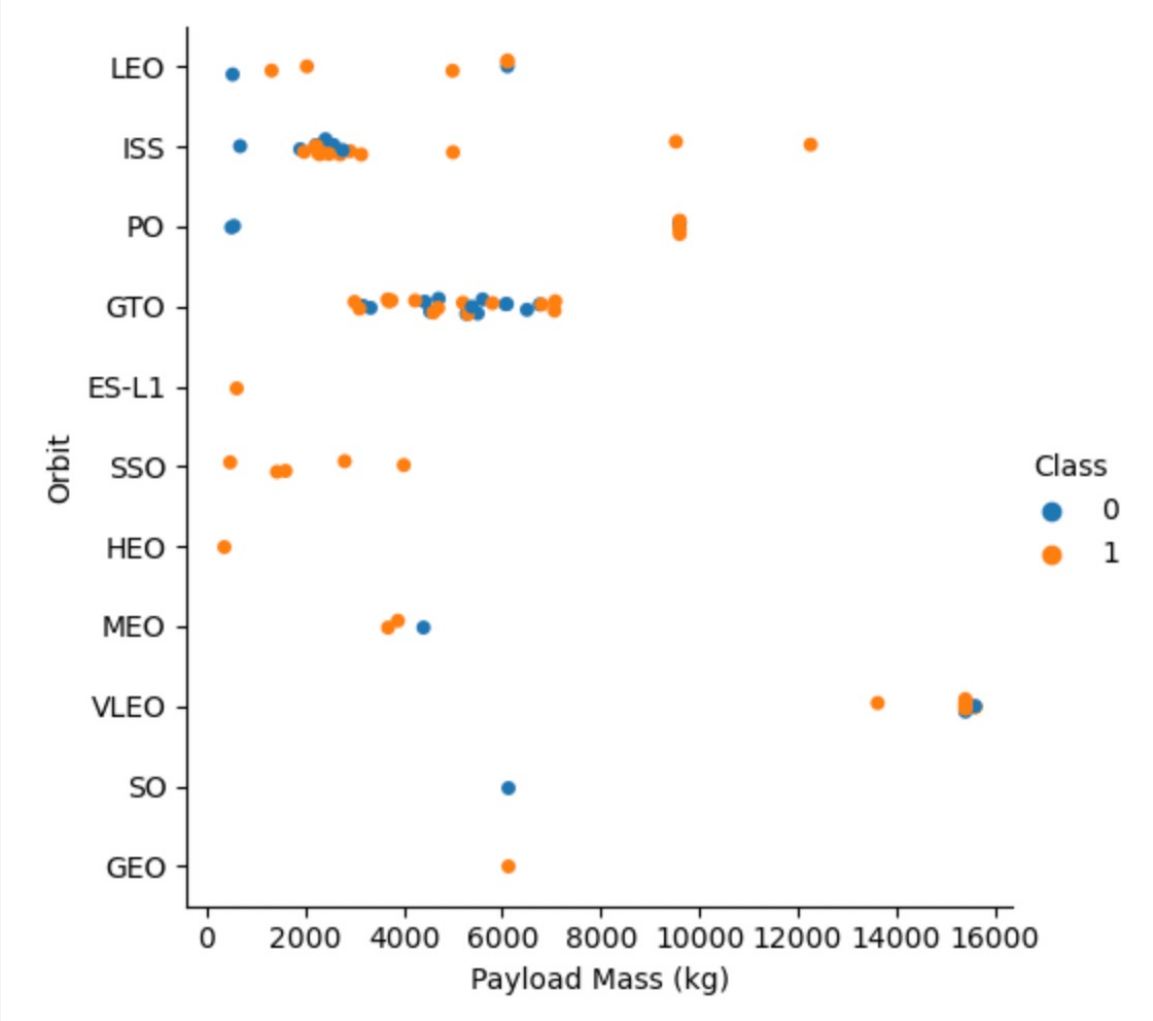
Flight Number vs. Orbit Type



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

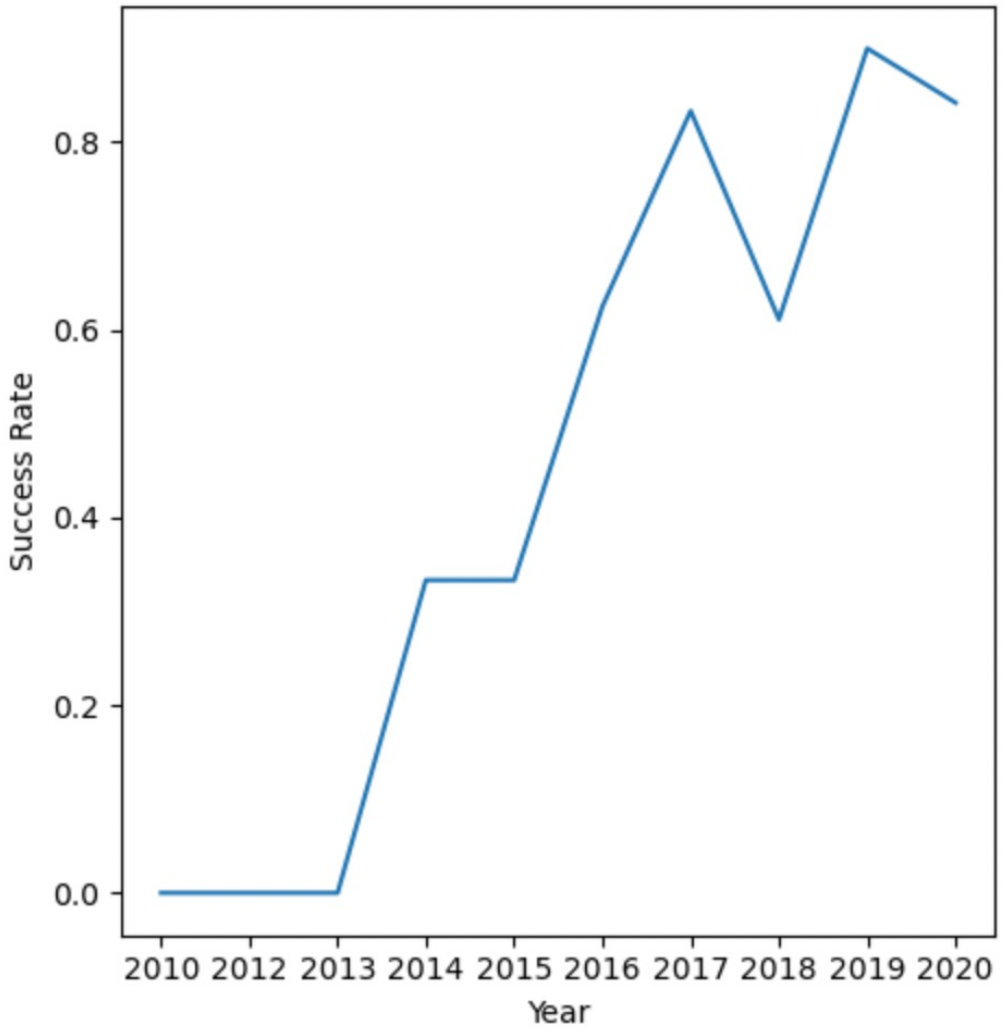
Payload vs. Orbit Type



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, and 2019-2020
- 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, and CCAFS SLC-40.

```
%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

Launch Site Names Begin with 'CCA'

The query used LIMIT 5 to only list 5 launch sites with names beginning with ‘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_Outcome
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
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 (parachute)
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 attempt
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attempt
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

The SUM function was used to add all the payload masses, to get a total payload mass of 619967kg.

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS total_payload_mass FROM SPACEXTBL;
```

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

total_payload_mass
619967.0

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 2928.3 kg.

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS F9_payload_mass FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1';
```

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

F9_payload_mass
2928.4

First Successful Ground Landing Date

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.

```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (ground pad)';
```

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

MIN(Date)
01/08/2018

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
SELECT Booster_Version
FROM SPACEXTBL
WHERE LANDING_OUTCOME = 'Success (drone ship)'
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

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

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

4 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

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

```
%sql SELECT Mission_Outcome, COUNT(Mission_Outcome) AS total FROM SPACEXTBL GROUP BY Mission_Outcome;
```

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

Mission_Outcome	total
None	0
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

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.

```
%%sql
SELECT Booster_Version
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

2015 Launch Records

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

```
%%sql
SELECT substr(Date, 4, 2) AS month, Landing_Outcome, Booster_Version, Launch_Site
FROM SPACEXTBL
WHERE substr(Date,7,4)='2015' AND Landing_Outcome = 'Failure (drone ship)';
```

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

month	Landing_Outcome	Booster_Version	Launch_Site
10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

```
%%sql
SELECT Landing_Outcome, COUNT(*) AS num_launches
FROM SPACEXTBL
WHERE DATE BETWEEN '04/06/2010' AND '20/03/2017'
GROUP BY Landing_Outcome
ORDER BY num_launches DESC;
```

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

Landing_Outcome	num_launches
Success	20
No attempt	9
Success (drone ship)	8
Success (ground pad)	7
Failure (drone ship)	3
Failure	3
Failure (parachute)	2
Controlled (ocean)	2
No attempt	1

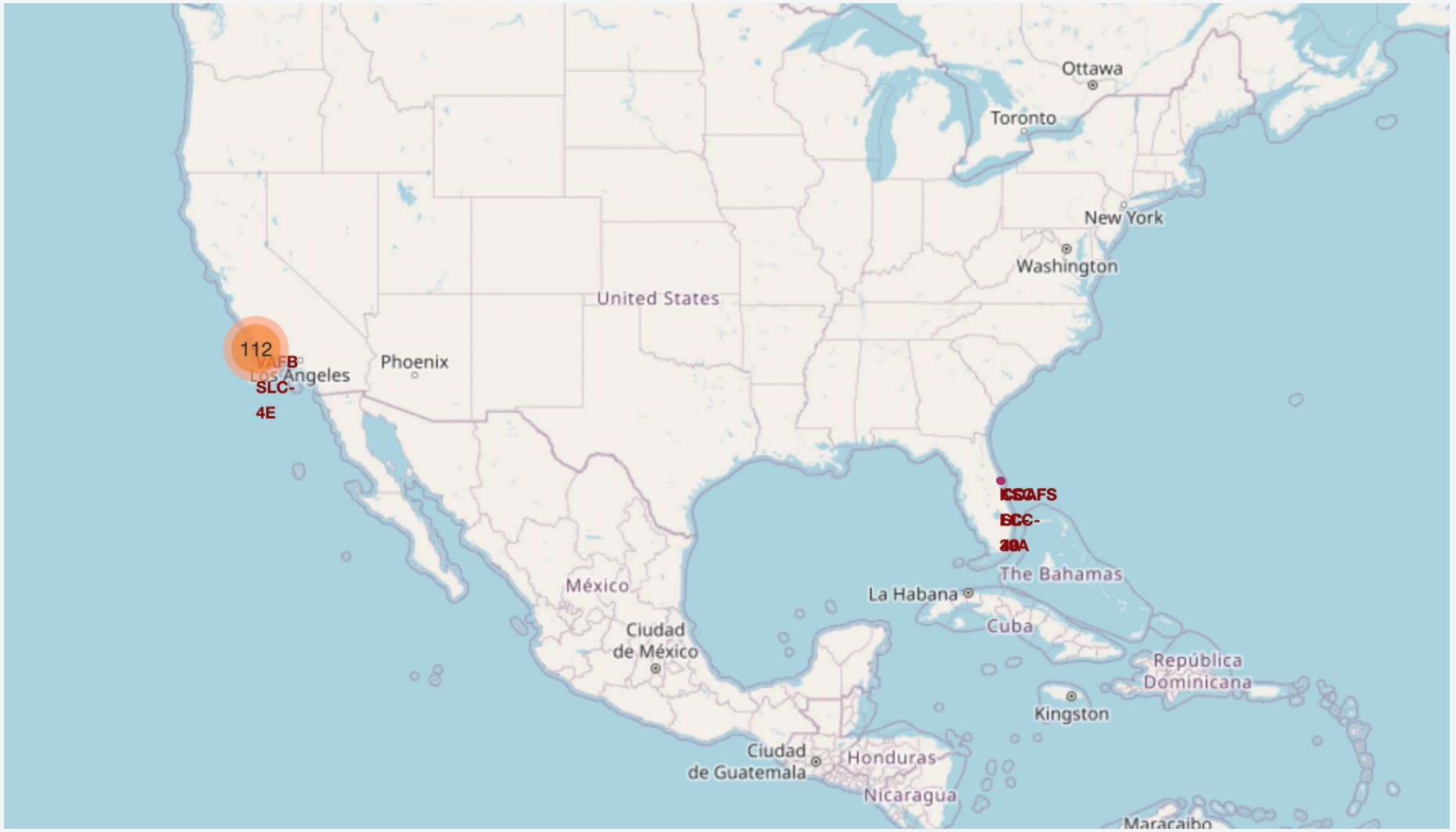
Landing outcomes between 2010-06-04 and 2017-03-20 were ranked, by grouping by landing outcome and ordering by the number of launches for each outcome, ordered in descending order. It can be seen that success is the most common outcome, and no attempt is the least common.


Section 3
Launch Sites
Proximities Analysis

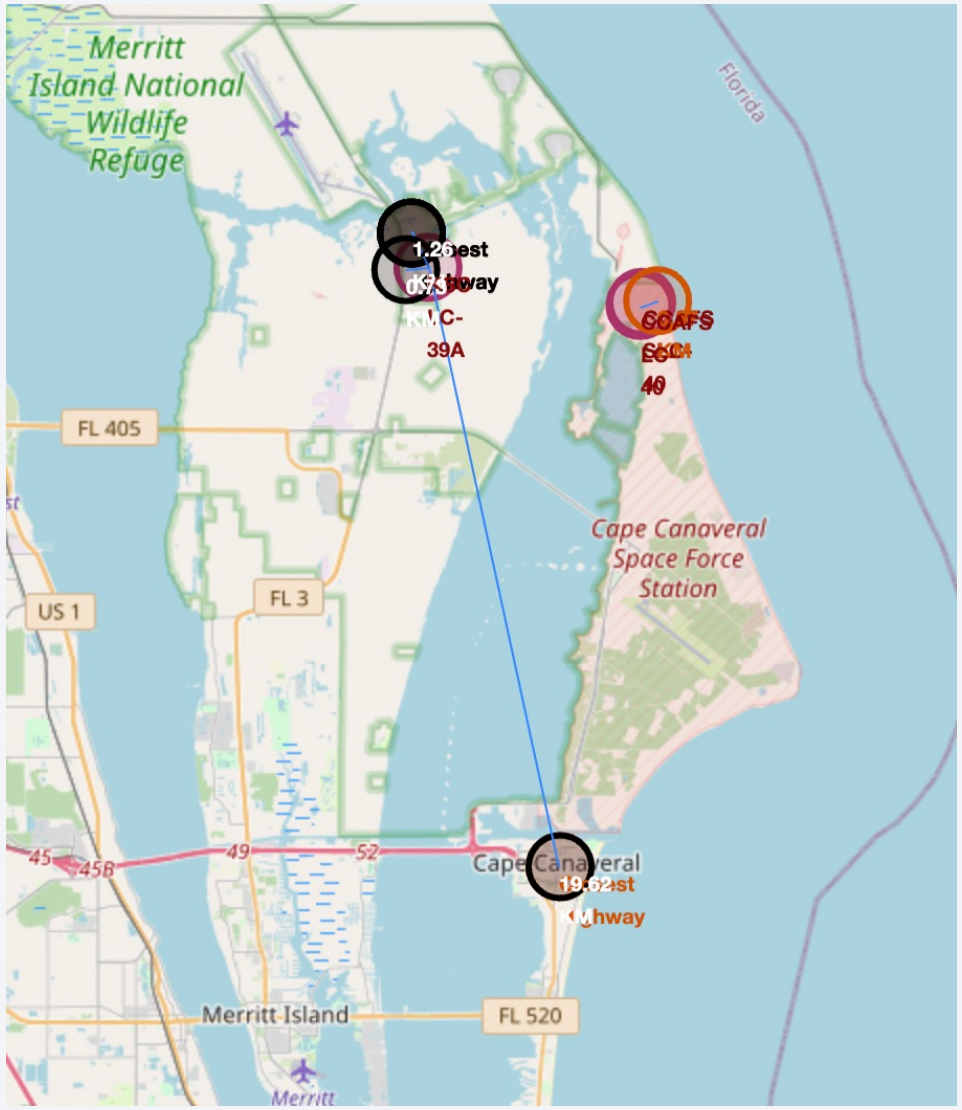
Launch Site Locations

It can be seen that the launch sites are all located in the United States

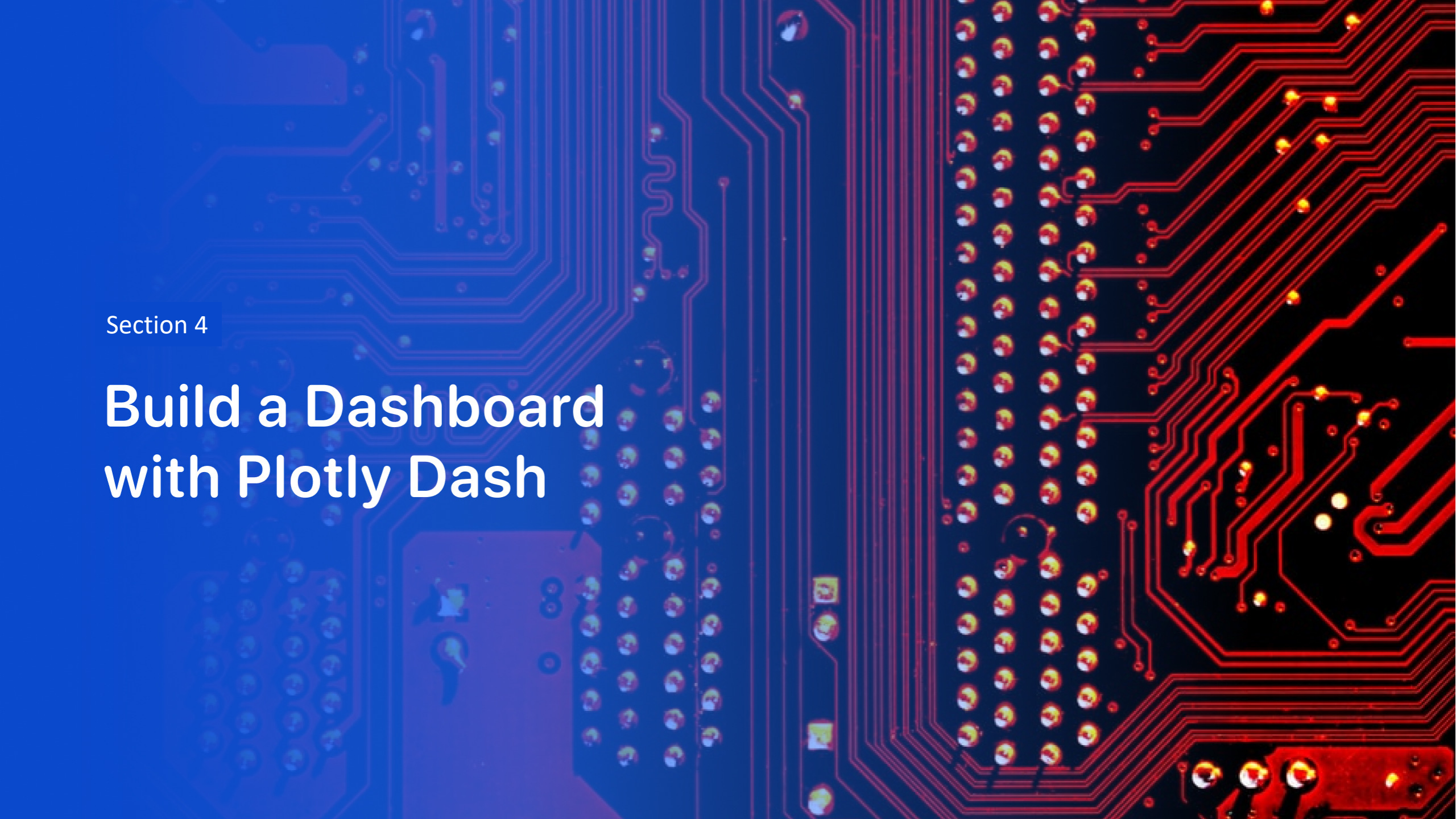
Success/Failed Launches



Launch Site Proximities



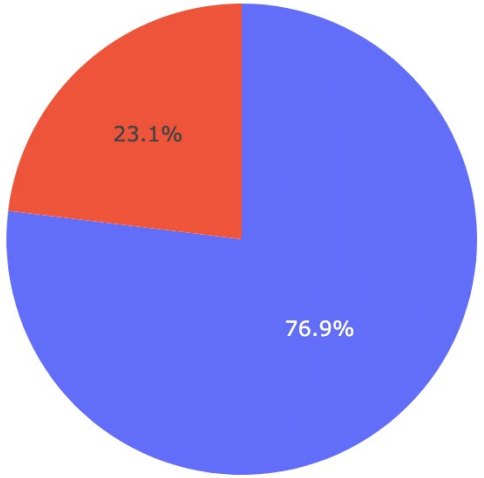
It can be seen that launch sites are close enough to railways, highways, and cities that workers and materials can be transported, but not close enough to cause any public damage. They're also close to coasts so that if damage does occur, it'll be close to water and won't harm any people or properties.


Section 4
Build a Dashboard with Plotly Dash

Total Success Launches for All Sites

It can be seen that 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%.

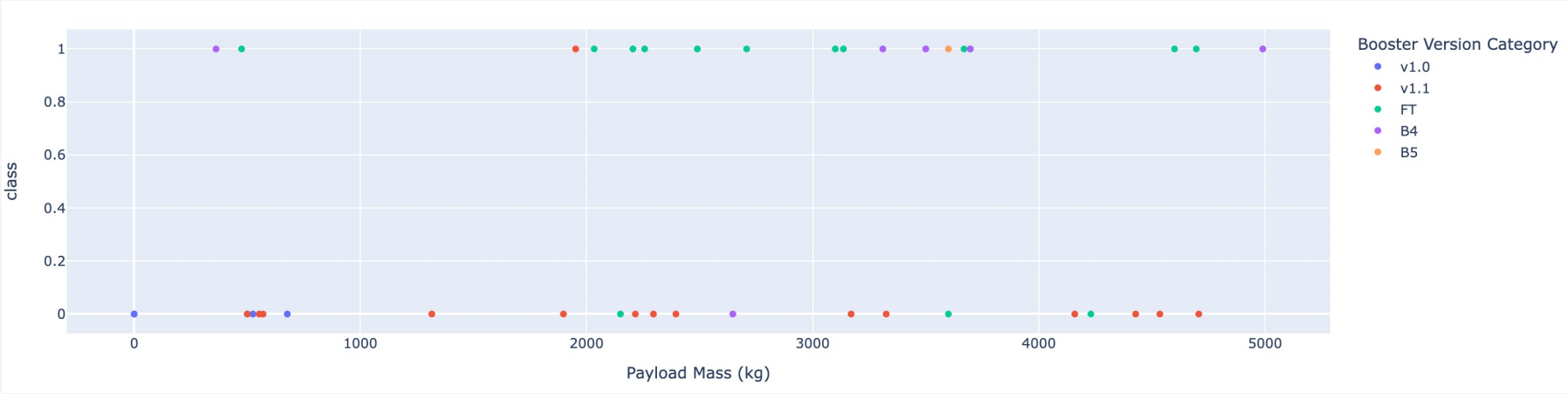
Total Success Launches for Site KSC LC-39A



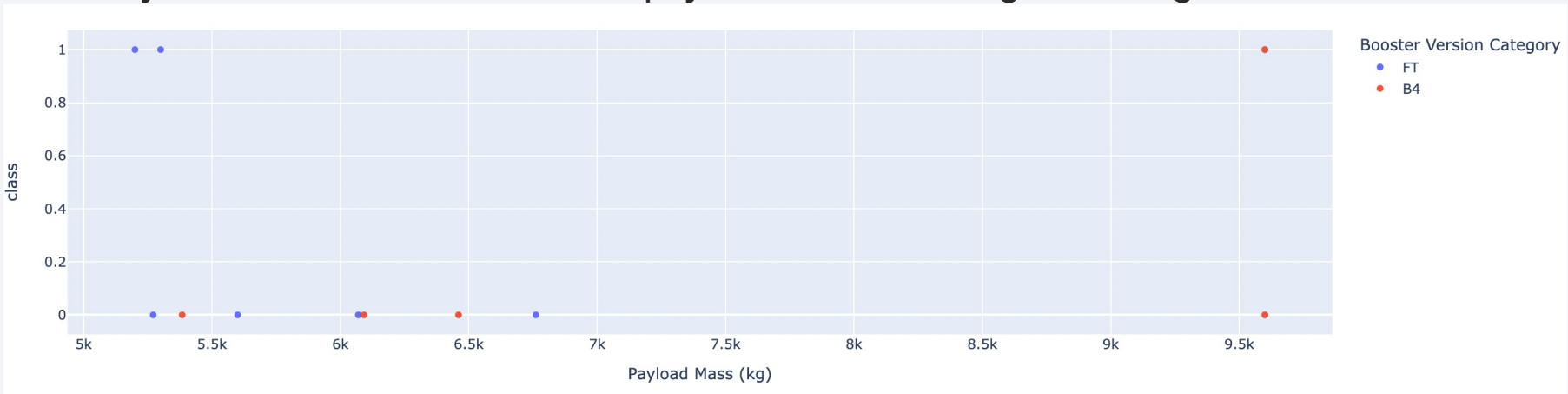
It can be seen that site KSC LC-39A, the site making up the greatest number of successes, has a success rate of 76.9% and failure of 23.1%.

Payload vs Launch Outcome for All Sites

Payload vs Launch Outcome for payload mass 0kg-5000kg



Payload vs Launch Outcome for payload mass 5000kg-10000kg



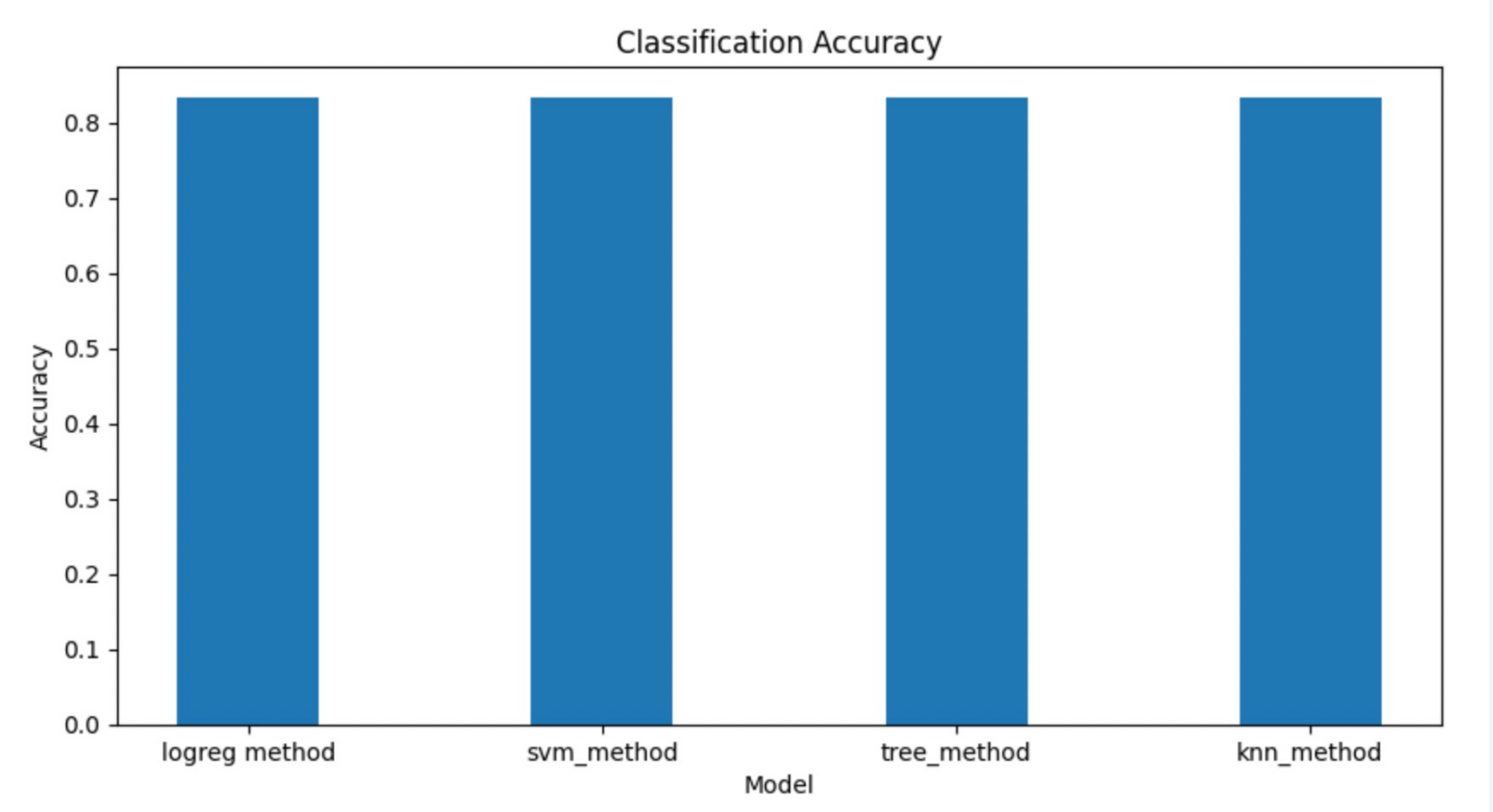
It can be seen that the success rate is higher when the payload mass is lower, i.e. below 5500 kg, where 1 indicates success and 0 indicates failure.

It can also be seen that boosters FT and B4 are the only boosters with payload mass above 5000kg.

Section 5

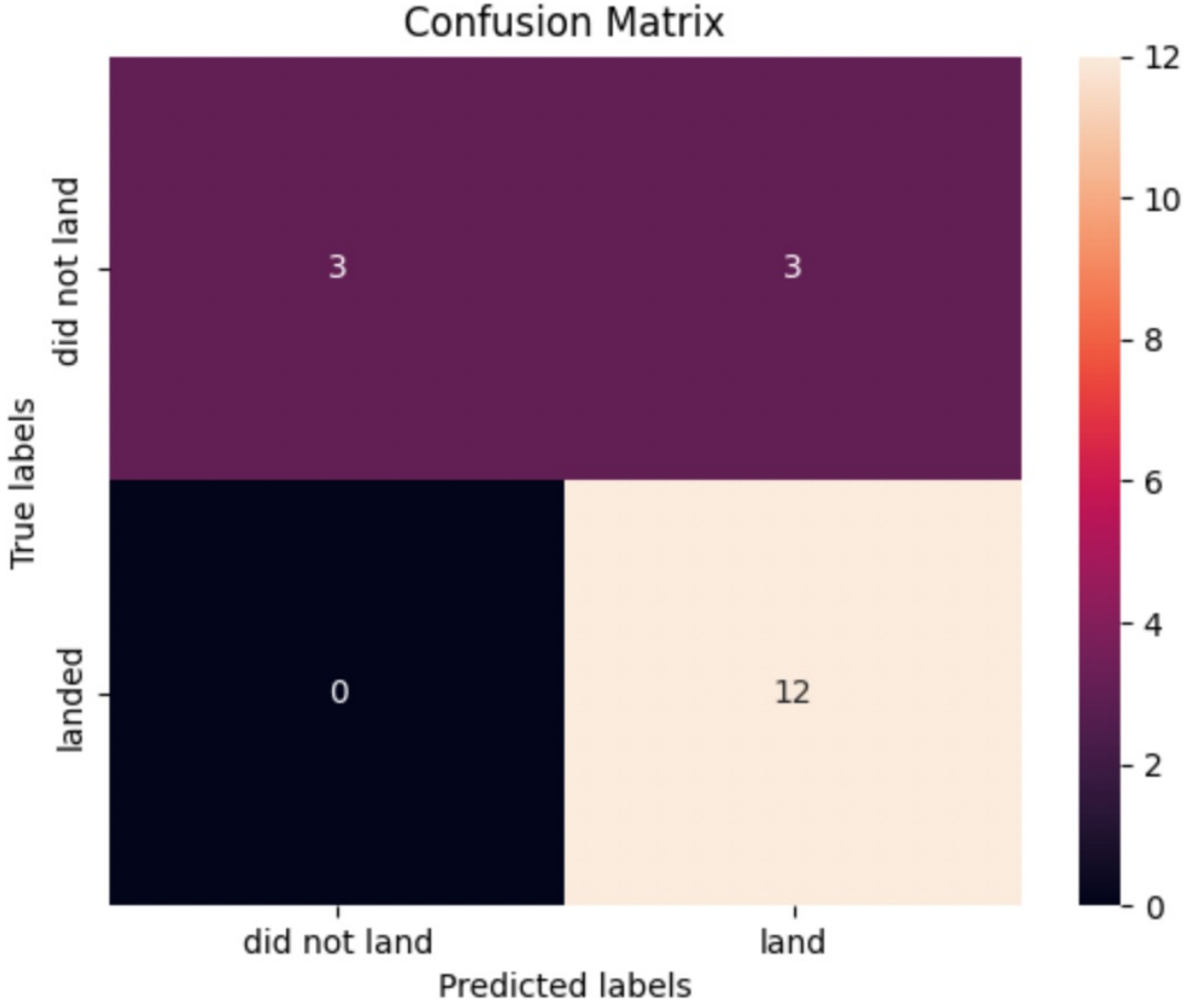
Predictive Analysis (Classification)

Classification Accuracy



It can be seen that the four models (logistic regression, SVM, decision tree, and KNN) all have the same accuracy, of about 0.833.

Confusion Matrix



The classifications all had the same confusion, as shown. We can conclude:

- There were 3 true negatives (predicted to not land and didn't land)
- There were 3 false positives (predicted to land but didn't land)
- There were 0 false negatives (predicted to not land but landed)
- There were 12 true positives (predicted to land and landed)

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 to 2020
- 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

Thank you!