IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Emilia Brunert
28.09.2023

Outline

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

Executive Summary

Summary of methodologies

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

Summary of all results

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

Introduction

Project background and context

SpaceX, a leader in the space industry, has disrupt the space industry by offering a rocket launches specifically Falcon 9 as low as 62 million dollars; other providers cost upwards of 165 million dollars each. SpaceX makes this possible due to its novel reuse of the first stage of its Falcon 9 rocket.

By determining if the first stage will land, we can determine the price of it. In order to achieve this, we can use data and implement machine learning models to predict whether SpaceX or a competing company can reuse the first stage.

Problems you want to find answers

Identifying factors that influence the landing outcome.

Rate of successful landings overtime.

Best predictive model for successful landing.

Section 1

Methodology

Methodology

Executive Summary

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

Data Collection

API

- Request data using SpaceX REST API.
- Decode response using .json() and turn it into a pandas dataframe usingusing .json_normalize().
- Request information about the launches from SpaceX API using custom functions.
- Create dictionary from the data and then create data frame from the dictionary.
- Clean Data: check for missing values and fill them with whatever needed.
- Export data to csv file.

Web Scraping

- Request data from Wikipedia.
- Create Beautiful Soup object from HTML response.
- Collect data from HTML tables.
- Create dictionary from the data a create data frame from the dictionary.
- Export data to csv file.

Data Collection – SpaceX API

Request Data to the SpaceX API

Use json_normalize method to convert json result to dataframe

Performed data cleaning and filling the missing value

https://github.com/poncemilia/Applied-Data-Science-Capstone/blob/main/1.Data%20Collection%20API%20Lab.ipynb

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

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

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

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

# We will remove rows with multiple cores because those are falcon rocke
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then e
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

Data Collection - Scraping

Request the Falcon9 Launch Wiki page from url

Create a BeautifulSoup from the HTML response

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

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text co
soup = BeautifulSoup(data,'html5lib')
```

https://github.com/poncemilia/Applied-Data-Science-Capstone/blob/main/2.Data%20Collection%20with%20Web%20Scraping%20lab.ipynb

Data Wrangling

- Calculate the **number of launches on each site**
- Calculate the **number and occurrence of mission outcome per orbit type**.
- Create a **landing outcome label** from the outcome column.
- Export the result to a **CSV**.

https://github.com/poncemilia/Applied-Data-Science-Capstone/blob/main/3.Data%20Wrangling%20Lab.ipynb

EDA with Data Visualization

Charts

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

- Why scatter plots? They easily show if there are relationships between them.
- Why bar graphs? They show comparisons among discrete categories
- Why line plots? Visualize data easily to rank.

https://github.com/poncemilia/Applied-Data-Science-Capstone/blob/main/5.EDA%20with%20Visualization%20Lab.ipynb

EDA with SQL

Queries – Display:

- 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.

Queries – Lists:

- List the date when the first succesful 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. Using a subquery.
- 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.
- 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.

https://github.com/poncemilia/Applied-Data-Science-Capstone/blob/main/4.EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

Markers indicating Launch Records:

- If a launch was successful, green marker (class=1).
- If a launch was failed, red marker (class=0)

Markers indicating Launch Sites:

- A blue circle at NASA Johnson Space Center's coordinate

https://github.com/poncemilia/Applied-Data-Science-Capstone/blob/main/6.Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Build a Dashboard with Plotly Dash

Dropdown List

- Allow user to select Launch sites.

Pie Chart

- Show user the total launches by a certain sites.

Scatter Chart

- Show user the correlation between payload and launch success.

https://github.com/poncemilia/Applied-Data-Science-Capstone/blob/main/7.Interactive%20Dashboard%20with%20Ploty%20Dash.py

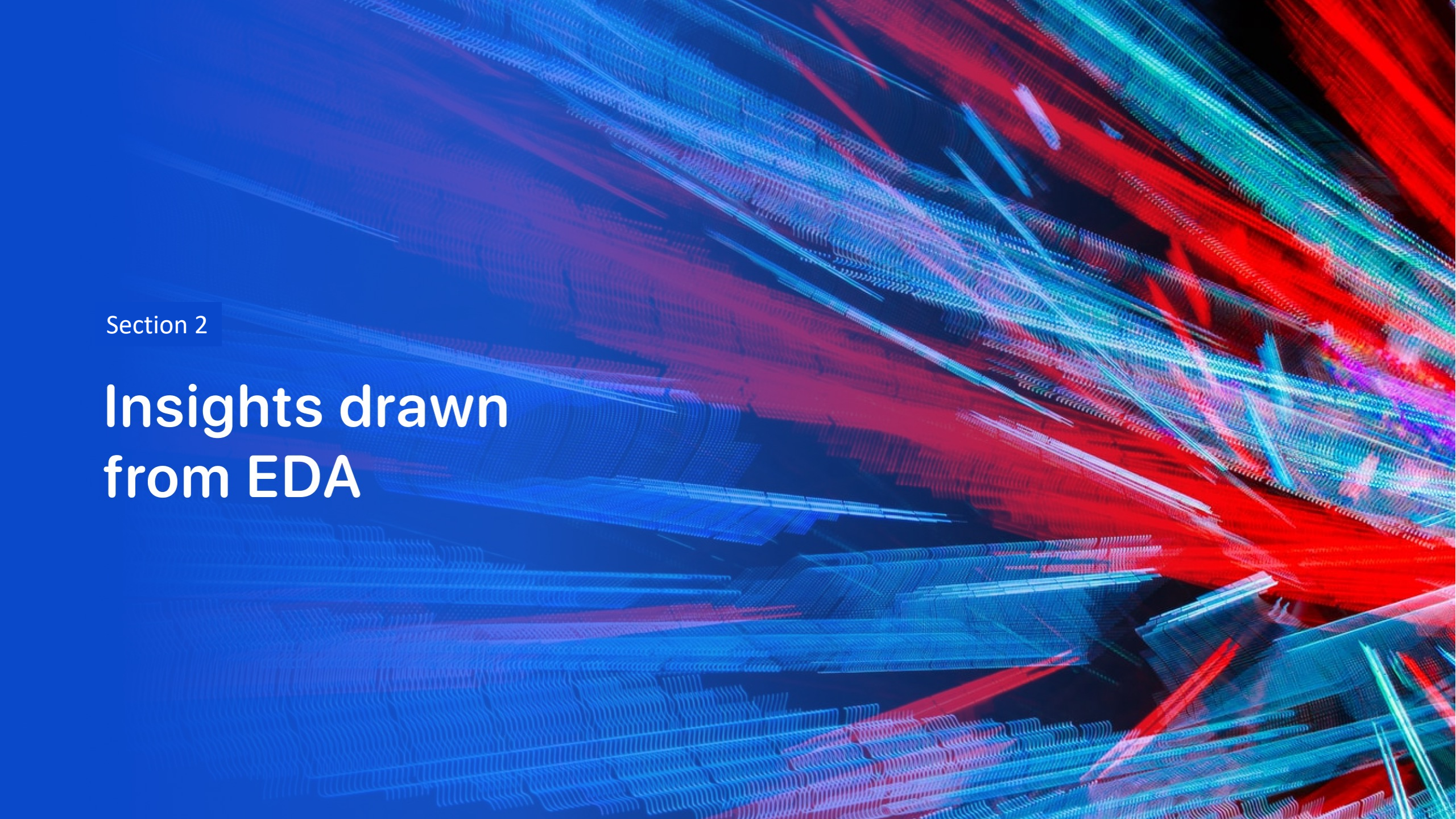
Predictive Analysis (Classification)

- Load the dataset into NumPy and Pandas
- Standardize and transform data
- Split Data into training and test datasets
- Set parameters and algorithms to GridSearchCV
- Calculate accuracy on the test data
- Plot confusion matrix
- Identify best model using Accuracy

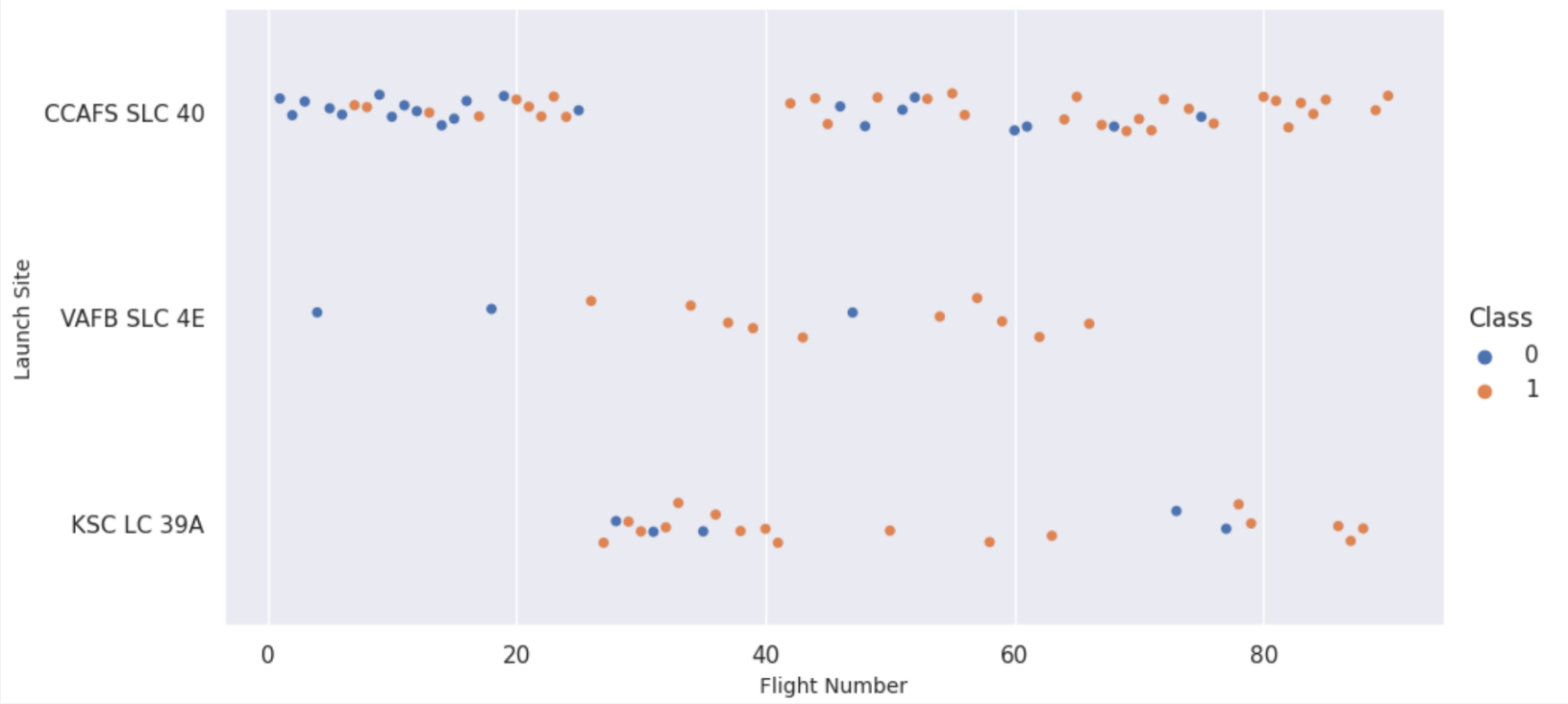
https://github.com/poncemilia/Applied-Data-Science-Capstone/blob/main/8.Machine%20Learning%20Prediction%20lab.ipynb

Results

- Exploratory data analysis (EDA) results
- Interactive analytics demo in screenshots
- Predictive analysis results

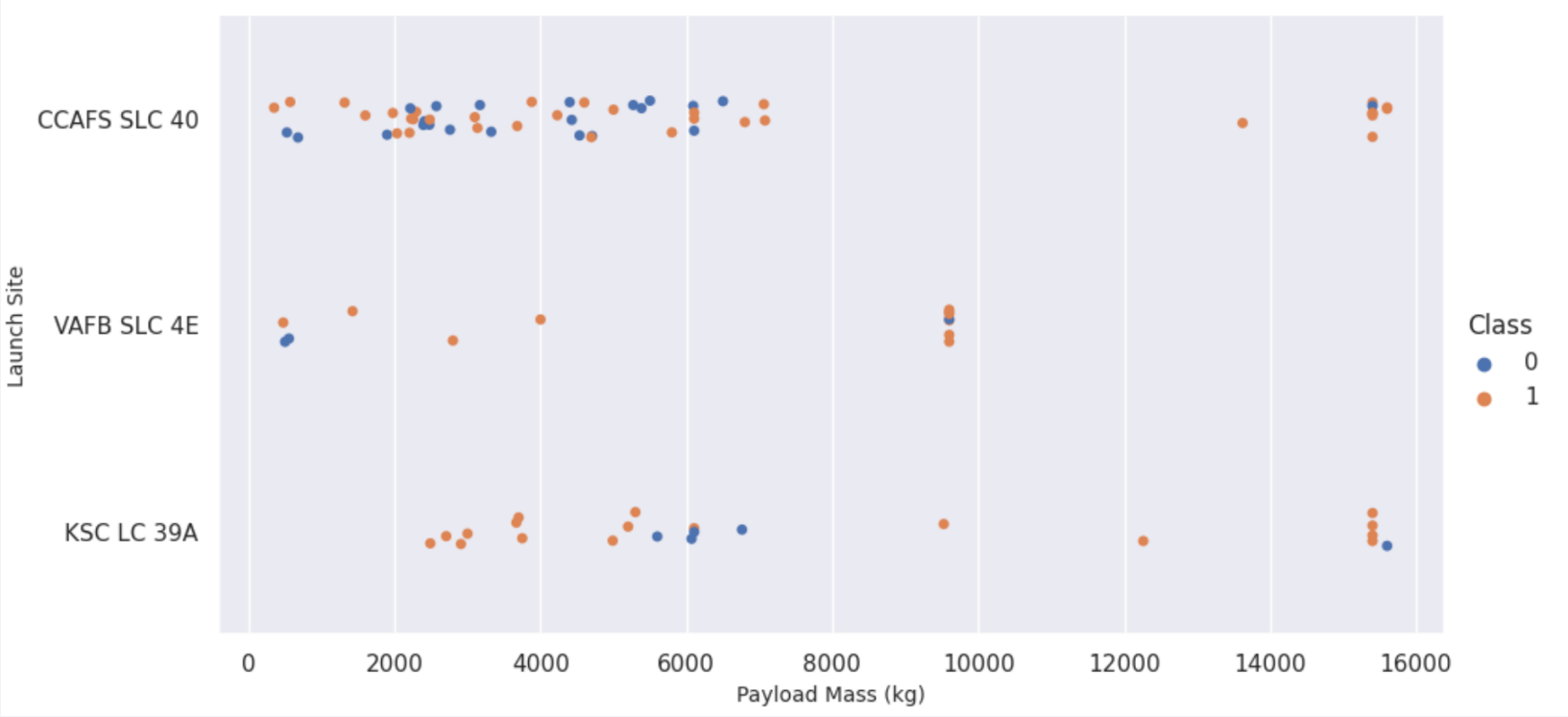

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



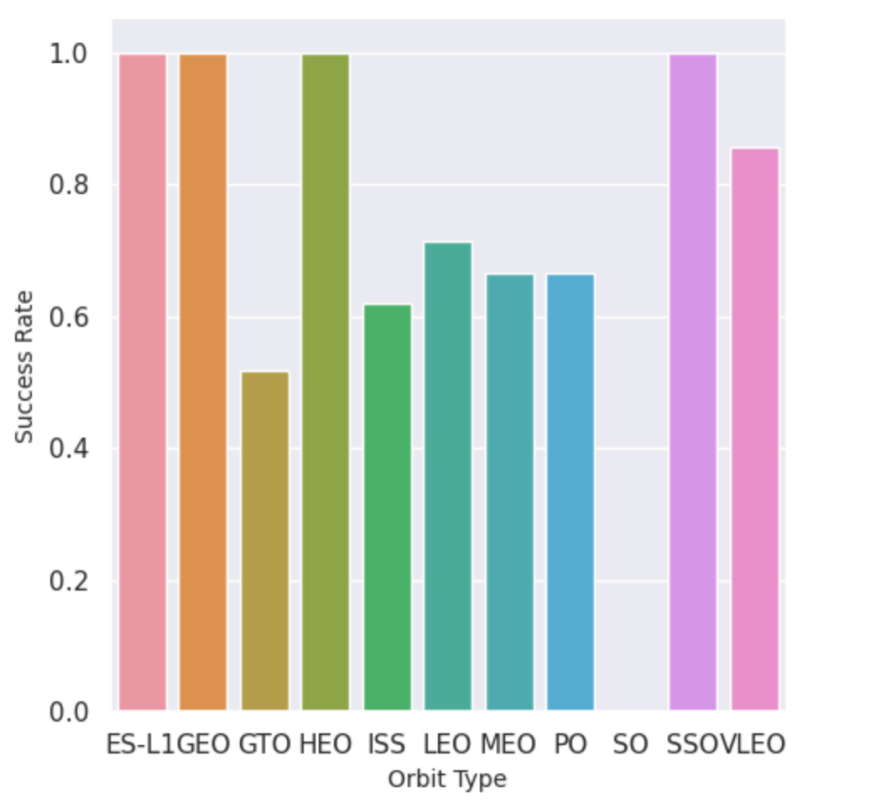
We can infer that earlier flights had a lower success rate while later flights had a higher success rate.

Payload vs. Launch Site



We can observe that the higher the payload mass is,the higher the success rate.

Success Rate vs. Orbit Type

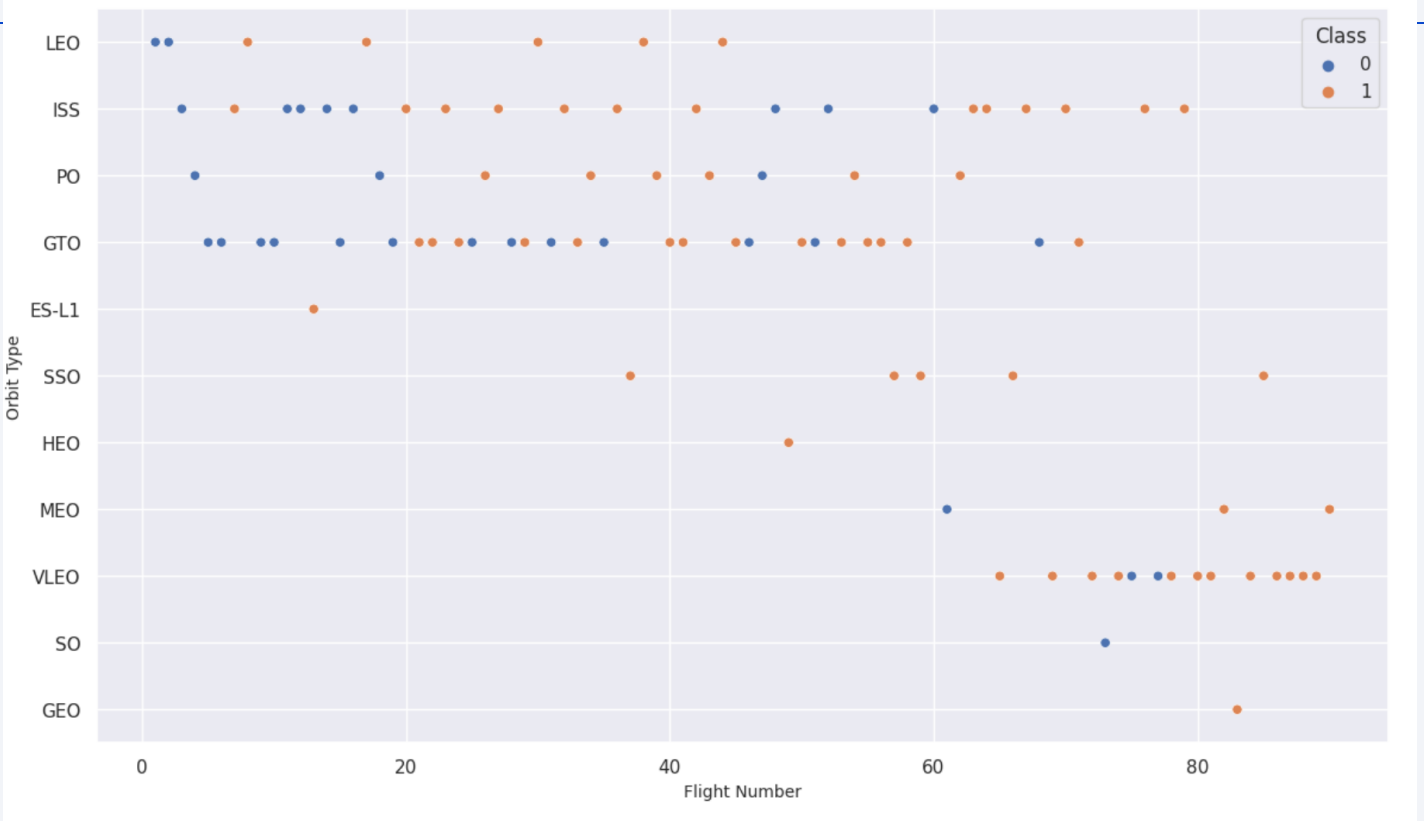


100% Success Rate: ES-L1, GEO, HEO and SSO.

50% to 80% Success Rate: GTO, ISS, LEO, MEO, PO, VLEO.

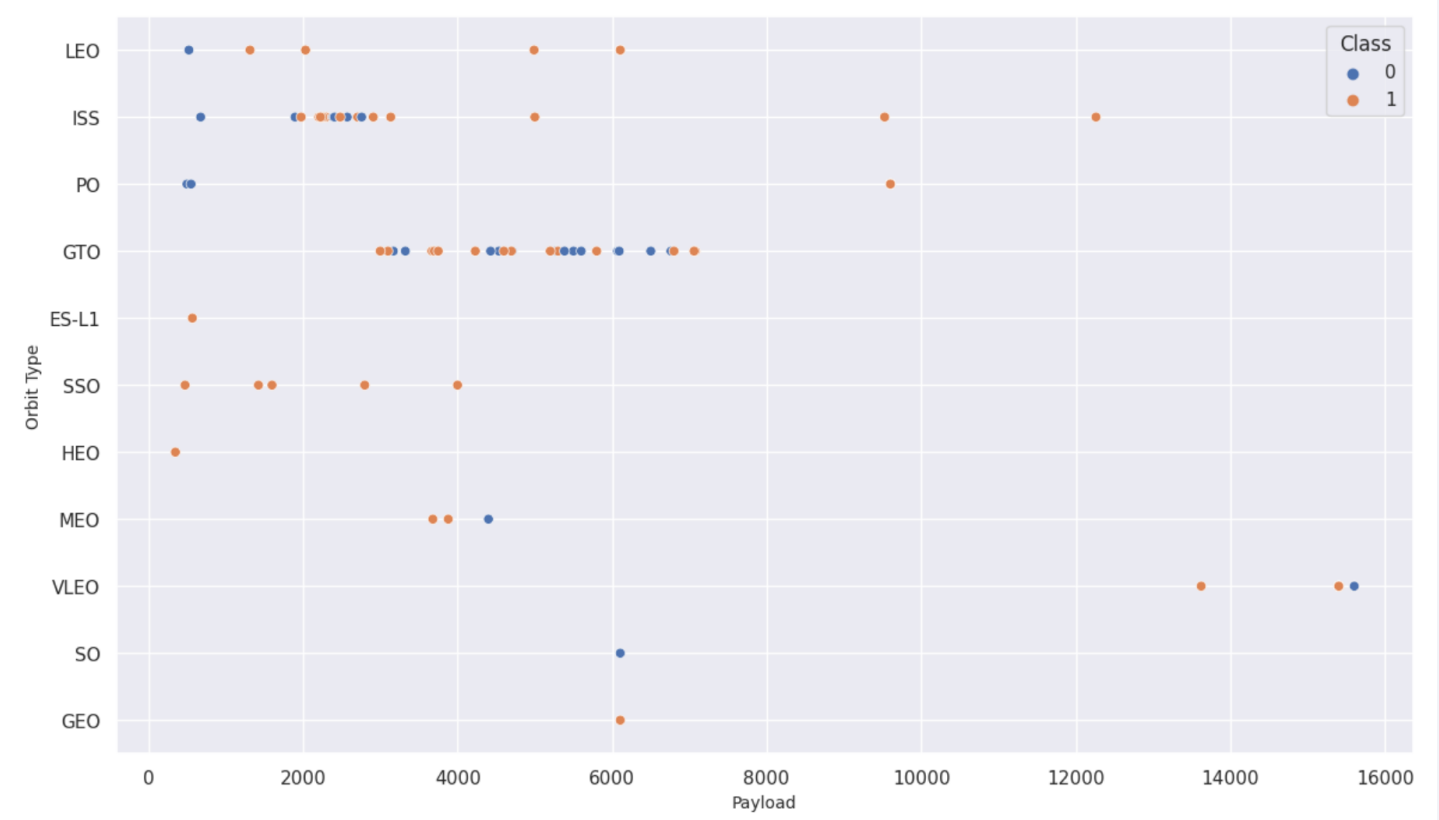
0% Success Rate: SO.

Flight Number vs. Orbit Type



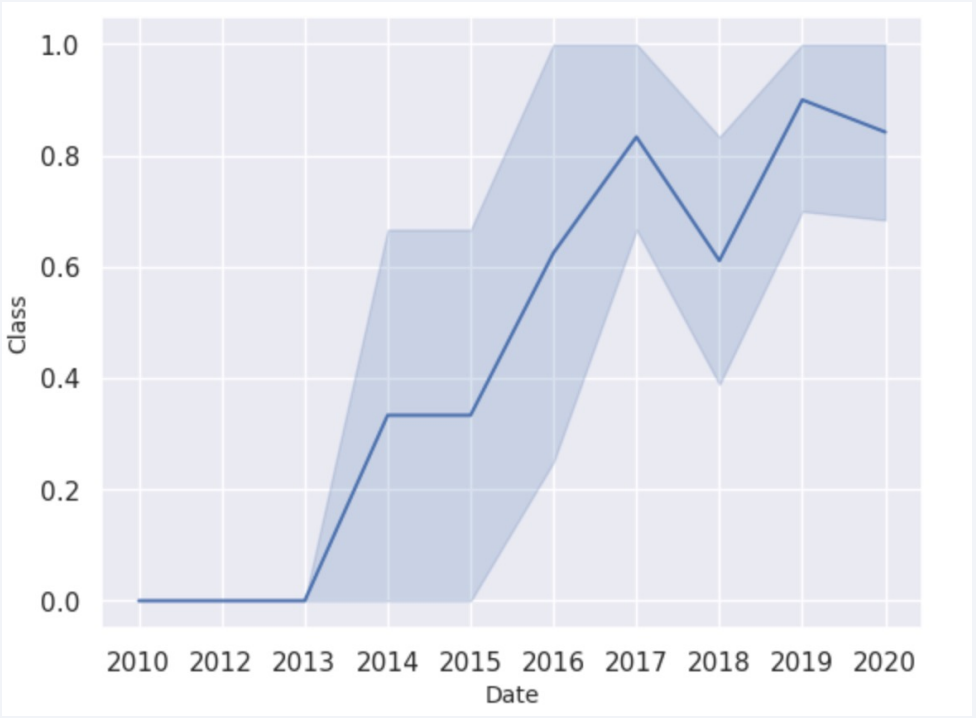
The success rate tends to increases with the number of flights for each orbit (LEO orbit).

Payload vs. Orbit Type



The GTO and ISS orbits show a combination of success with heavier payloads.

Launch Success Yearly Trend



There is a improvement of the success rate from 2013 to 2017 and from 2018 to 2019.

All Launch Site Names

In [11]:
```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

Out[11]:

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

Launch Site Names Begin with 'CCA'

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

In [13]:
```
%sql SELECT LAUNCH_SITE FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

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

Out[13]:

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

```
In [14]: %sql SELECT SUM (PAYLOAD_MASS__kg_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA(CRS)' ;
 * sqlite:///my_data1.db
Done.
Out[14]: SUM (PAYLOAD_MASS__kg_)
                                None
```

Average Payload Mass by F9 v1.1

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

```
 * sqlite:///my_data1.db
(sqlite3.OperationalError) no such function: AVERAGE
[SQL: SELECT AVERAGE (PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';]
(Background on this error at: http://sqlalche.me/e/e3q8)
```

First Successful Ground Landing Date

```
In [19]: %sql SELECT MIN (DATE) AS "First Successful Landing" FROM SPACEXTBL WHERE LANDING_OUTCOME =
 * sqlite:///my_data1.db
Done.
```

Out[19]:

First Successful Landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

In [20]:
```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (drone ship)' AN
```

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

Out[20]:

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

Total Number of Successful and Failure Mission Outcomes

In [26]:
```
%sql SELECT sum(case when MISSION_OUTCOME LIKE '%Success%' then 1 else 0 end) AS "Successful
    sum(case when MISSION_OUTCOME LIKE '%Failure%' then 1 else 0 end) AS "Failure Mission" \
FROM SPACEXTBL;
```

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

Out[26]:

Successful Mission	Failure Mission
100	1

Boosters Carried Maximum Payload

In [29]:

```
%sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload
WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

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

Out[29]:

Booster Versions which carried the Maximum Payload Mass
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

In [63]:
```
%sql select "Landing_Outcome", substr(Date,1,4), substr(Date,6,2),  "Booster_Version", "Launch_Site" from SPACEXT
```

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

Out[63]:

Landing_Outcome	substr(Date,1,4)	substr(Date,6,2)	Booster_Version	Launch_Site
Failure (drone ship)	2015	10	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	2015	04	F9 v1.1 B1015	CCAFS LC-40

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

In [84]:

```
%sql SELECT "Landing_Outcome", COUNT("Landing_Outcome") as 'Count'  \
FROM SPACEXTBL \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY "Landing_Outcome" \
ORDER BY 'Count' desc
```

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

Out[84]:

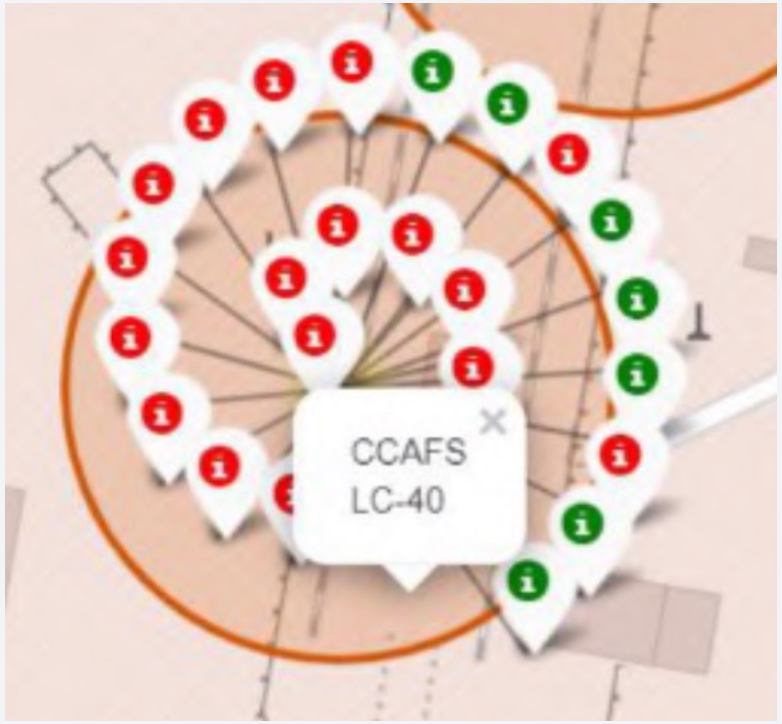
Landing_Outcome	Count
Uncontrolled (ocean)	2
Success (ground pad)	5
Success (drone ship)	5
Precluded (drone ship)	1
No attempt	10
Failure (parachute)	1
Failure (drone ship)	5
Controlled (ocean)	3


Section 3
Launch Sites
Proximities Analysis

Location of all the Launch Sites

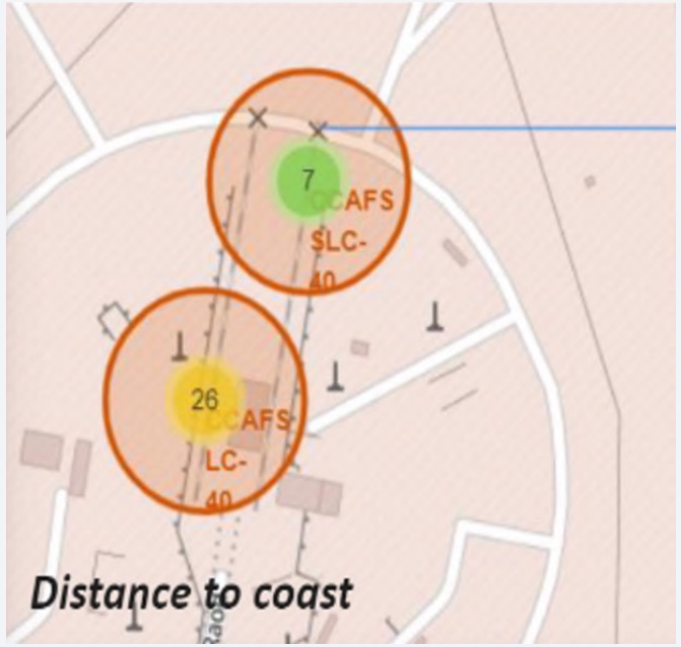
All the SpaceX launch sites are located at the United States

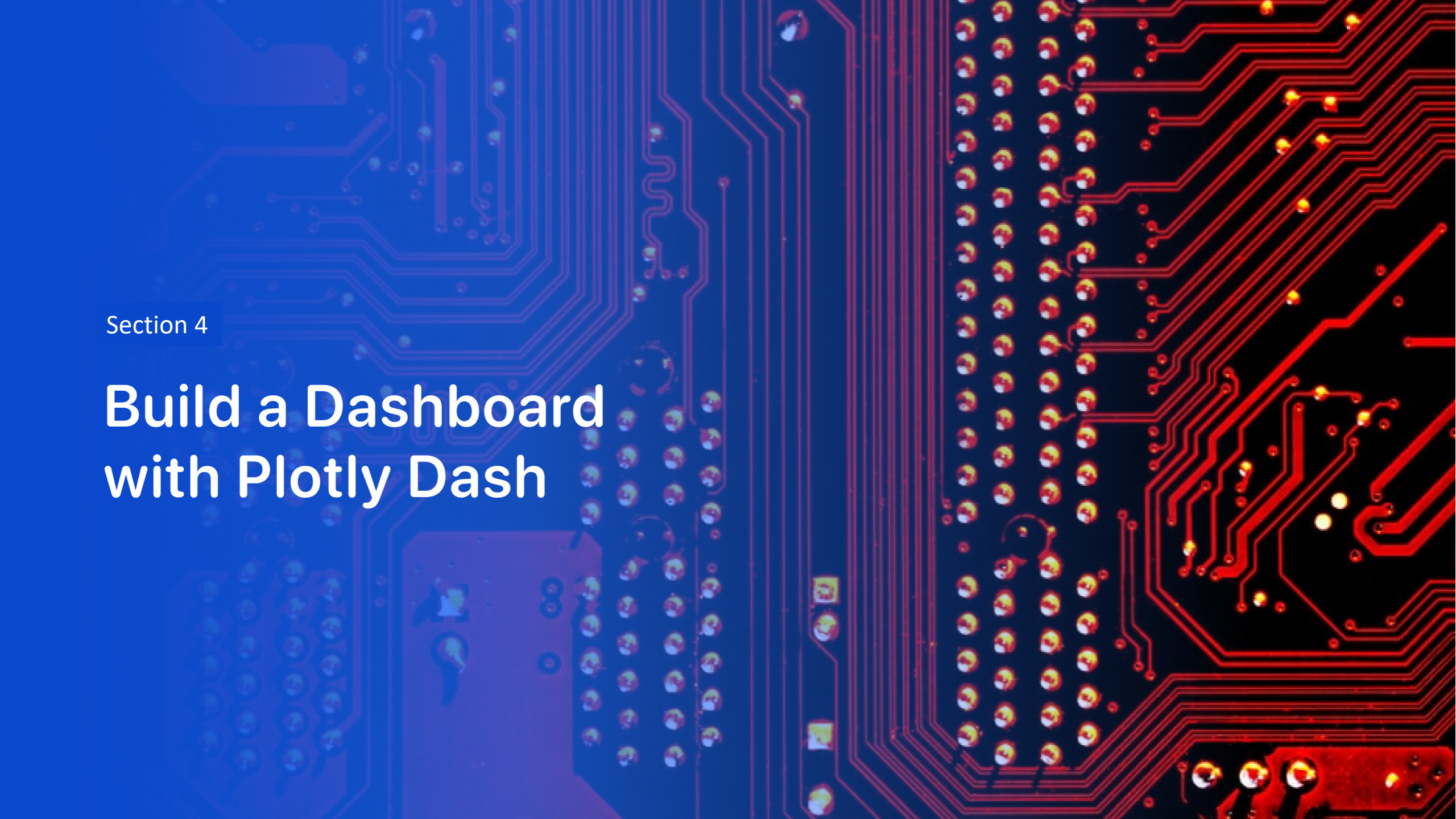
Launch Outcomes



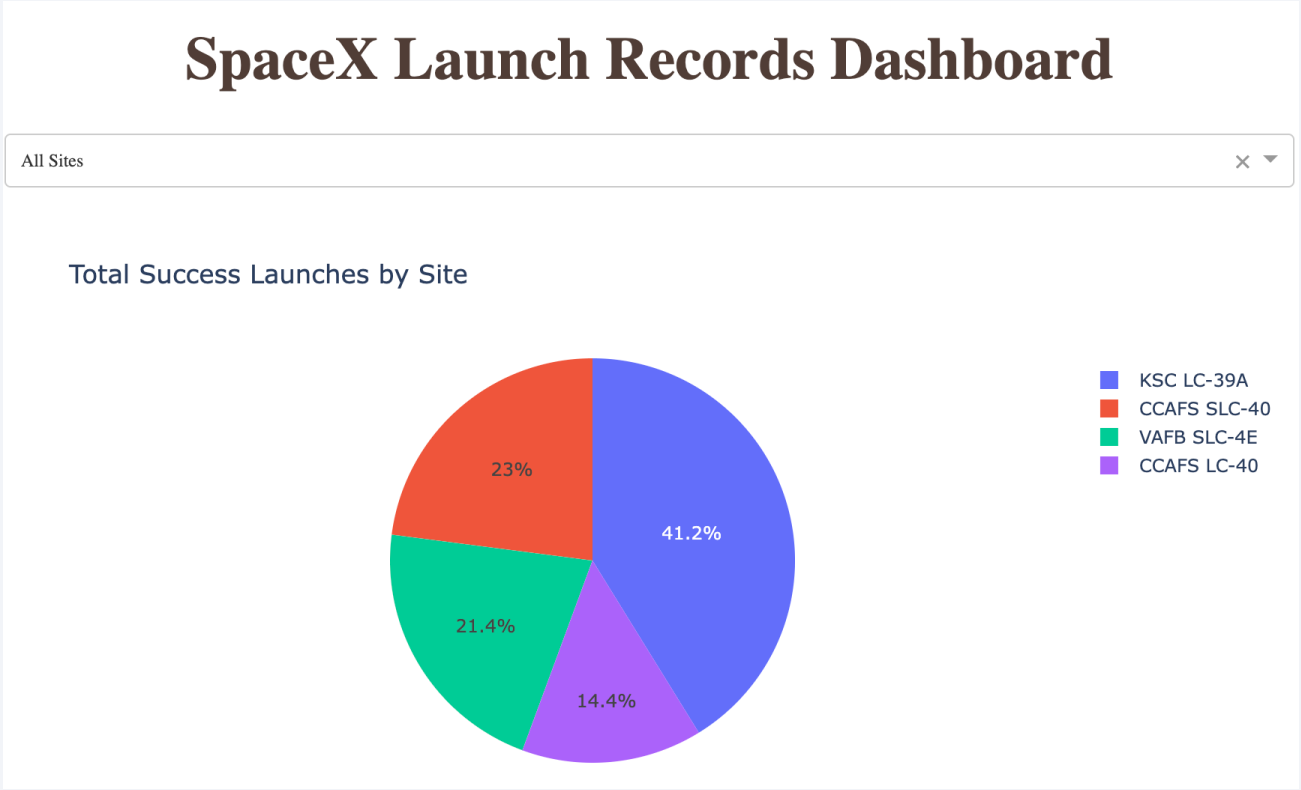
Green markers for successful launches.
Red markers for unsuccessful launches.

Launch Sites Distance to Landmarks



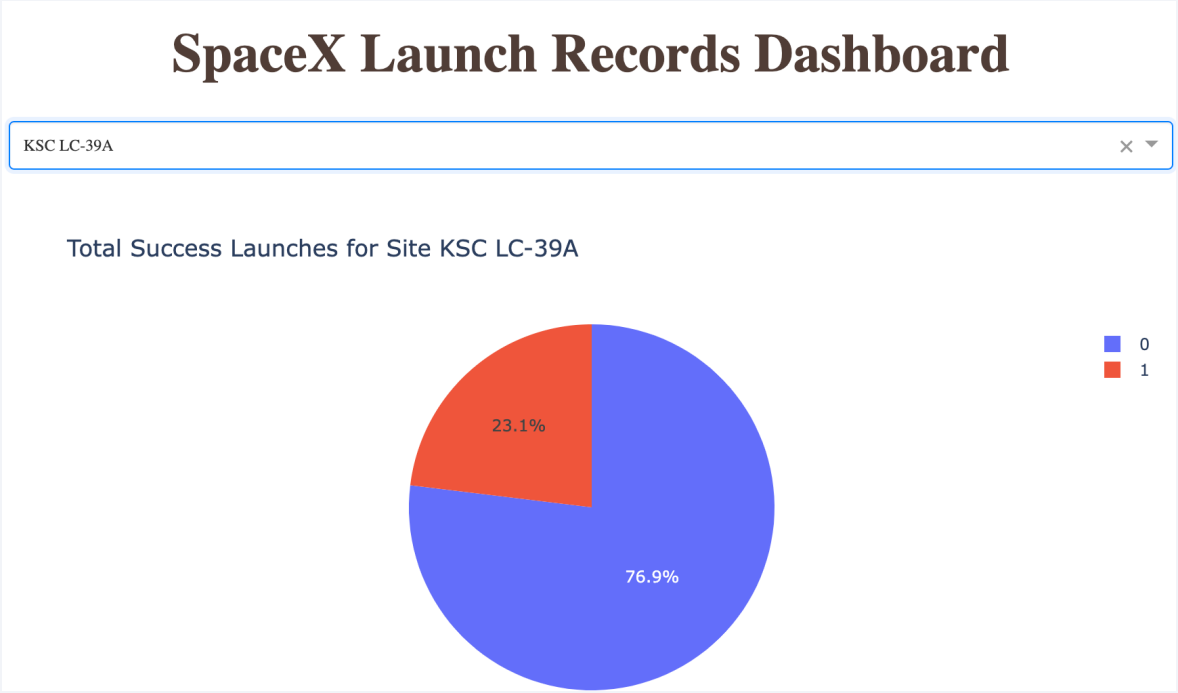

Section 4
Build a Dashboard with Plotly Dash

Launch Success by Site



KSC LC-39A has the most successful launches among launch sites.

Launch site with highest launch success ratio



KSC LC-39A has the highest success rate amongst launch sites with 76.9% of success.

Payload vs. Launch Outcome scatter plot for all sites

Payloads between 2,000 kg and 5,000 kg have the highest success rate

Section 5

Predictive Analysis (Classification)

Classification Accuracy

TASK 12

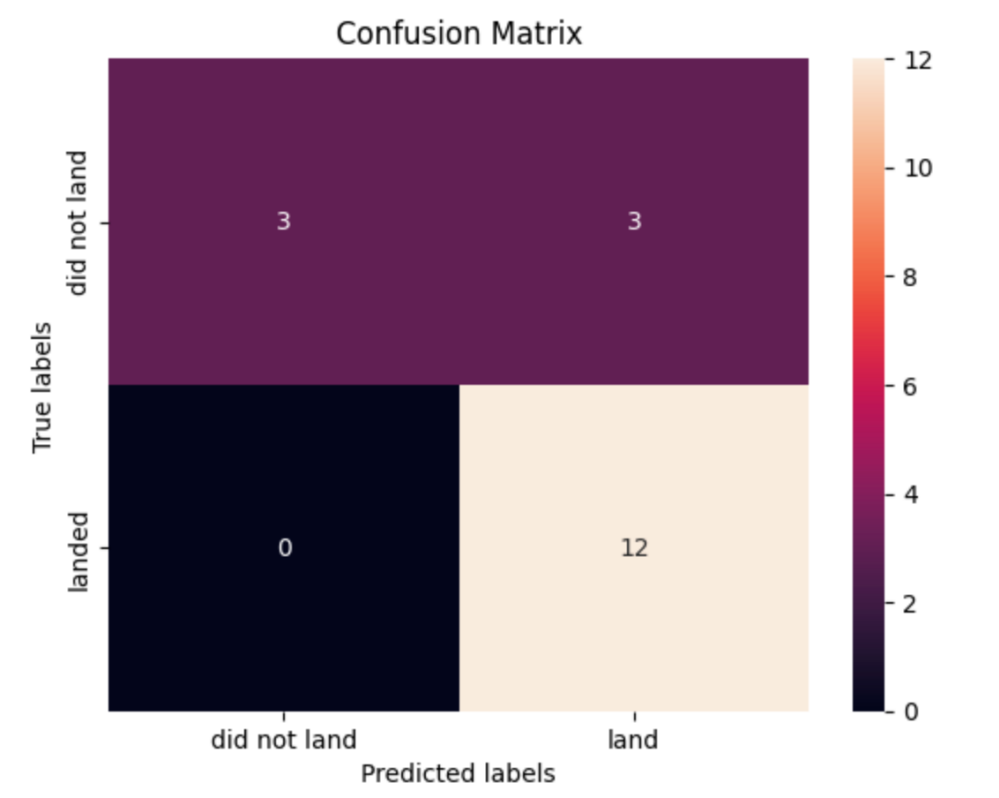
Find the method performs best:

In [57]:
```
print('Accuracy for Logistics Regression method:', logreg_cv.score(X_test, Y_test))
print('Accuracy for Support Vector Machine method:', svm_cv.score(X_test, Y_test))
print('Accuracy for Decision tree method:', tree_cv.score(X_test, Y_test))
print('Accuracy for K nearsdt neighbors method:', knn_cv.score(X_test, Y_test))
```

```
Accuracy for Logistics Regression method: 0.8333333333333334
Accuracy for Support Vector Machine method: 0.8333333333333334
Accuracy for Decision tree method: 0.8333333333333334
Accuracy for K nearsdt neighbors method: 0.8333333333333334
```

According to the accuracy analysis, all models are 83 equally likely to be accurate in predicting landing success.

Confusion Matrix



The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes.

Conclusions

- It is always important to remember that a larger dataset will help build on the predictive analytics results to understand if the findings can be generalizable to a larger data set
- It is possible that the study requires additional feature analysis in order to check the accuracy rate of the different models.

Thank you!