Winning Space Race with Data Science

Chaninthon Theerakulvanich
May 15, 2023

Outline

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

Executive Summary

- Our presentation focuses on the analysis of SpaceX data using various methodologies. We collected data through the SpaceX API and web scraping techniques, ensuring a comprehensive dataset for analysis. By applying data wrangling techniques, we prepared the data for further analysis.
- Using SQL queries and Python libraries like Pandas and Matplotlib, we conducted Exploratory Data Analysis (EDA) to uncover patterns and insights. The EDA revealed interesting findings related to launch sites, success rates, and payload masses.
- To enhance our analysis, we utilized interactive visual analytics tools such as Folium and Plotly Dash. These tools allowed us to create dynamic maps, interactive charts, and insightful dashboards for a comprehensive understanding of the data.
- Furthermore, we employed machine learning techniques to predict landing outcomes. By training various classification models, we were able to accurately predict the success or failure of landings based on available data.
- In summary, our analysis provides valuable insights into the success rates of different launch sites, the impact of payload masses, and the predictive capabilities of our models. Our interactive visualizations and dashboards offer an engaging way to explore and understand the data. These findings can assist in making informed decisions related to SpaceX missions and contribute to the advancement of space exploration.

Introduction

- ## Project background and context

SpaceX, a leading aerospace company, offers Falcon 9 rocket launches at a significantly lower cost compared to other providers. This cost advantage is primarily due to SpaceX's ability to reuse the first stage of the rocket. To assess the competitiveness of other companies bidding for rocket launches against SpaceX, it is crucial to determine the likelihood of successful first stage landings. This project aims to analyze data from Falcon 9 rocket launches advertised on SpaceX's website to predict the success of first stage landings.

- ## Problems you want to find answers

The main objective of this capstone project is to predict the success of the first stage landing in Falcon 9 rocket launches. By leveraging the available data from SpaceX's website, we aim to develop a predictive model that can accurately determine if the first stage will land successfully. This prediction will be valuable for assessing the cost of a launch and can provide insights for alternate companies interested in competing with SpaceX for rocket launch contracts.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

To gather the necessary data for the SpaceX Falcon 9 analysis, two primary methods were employed: utilizing the SpaceX API and performing web scraping on a Wikipedia page.

1. Data Collection using SpaceX API: The data collection process began by utilizing the SpaceX API, which is a RESTful API. By making a GET request to the SpaceX API, data was extracted using identification numbers associated with each launch. To facilitate this process, a set of helper functions were defined to effectively use the API and retrieve information from the launch data. The rocket launch data was requested from the SpaceX API URL. The JSON results obtained from the GET request were then parsed and decoded as a JSON object. The decoded response content was converted into a Pandas data frame, ensuring consistency and ease of analysis.

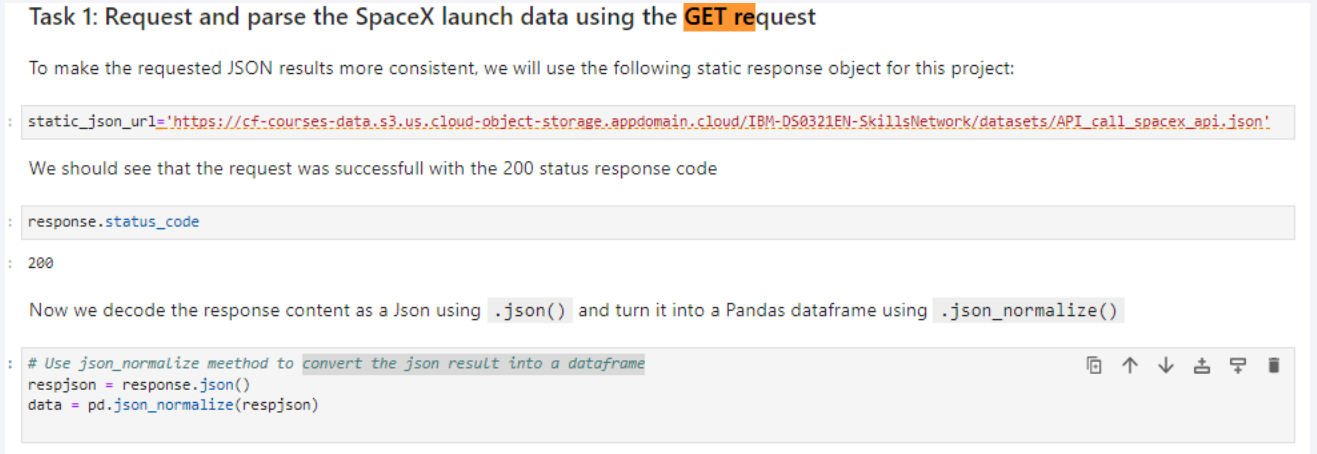
2. Web Scraping for Historical Launch Records: To supplement the API data, web scraping was performed to collect Falcon 9 historical launch records from a Wikipedia page titled "List of Falcon 9 and Falcon Heavy launches." The launch records were stored in an HTML format on the Wikipedia page. To extract the relevant information, the BeautifulSoup and request libraries were utilized. The Falcon 9 launch HTML table records were extracted from the Wikipedia page, parsed, and transformed into a Pandas data frame for further analysis.

By combining the data obtained from the SpaceX API and the web scraping process, a comprehensive dataset was created, enabling a thorough analysis of Falcon 9 rocket launches.

Data Collection – SpaceX API

- The data collection process involved utilizing the SpaceX API, a RESTful API. By making a GET request to the SpaceX API, data was obtained. The SpaceX launch data was requested and retrieved using the GET request. The response content was then parsed and decoded as a JSON result. This JSON result was further processed and converted into a Pandas data frame, providing a structured and organized format for analysis.
- This link to GitHub URL of the completed SpaceX API calls notebook

https://github.com/mchaninthon/Applied_Data_Science_Capstone/blob/fe0e7aa5764b3a3d15f2a956ad6f9e75039bb97e/1.%20Space-X%20Data%20Collection%20API.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:

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

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

```
response.status_code
```

```
200
```

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

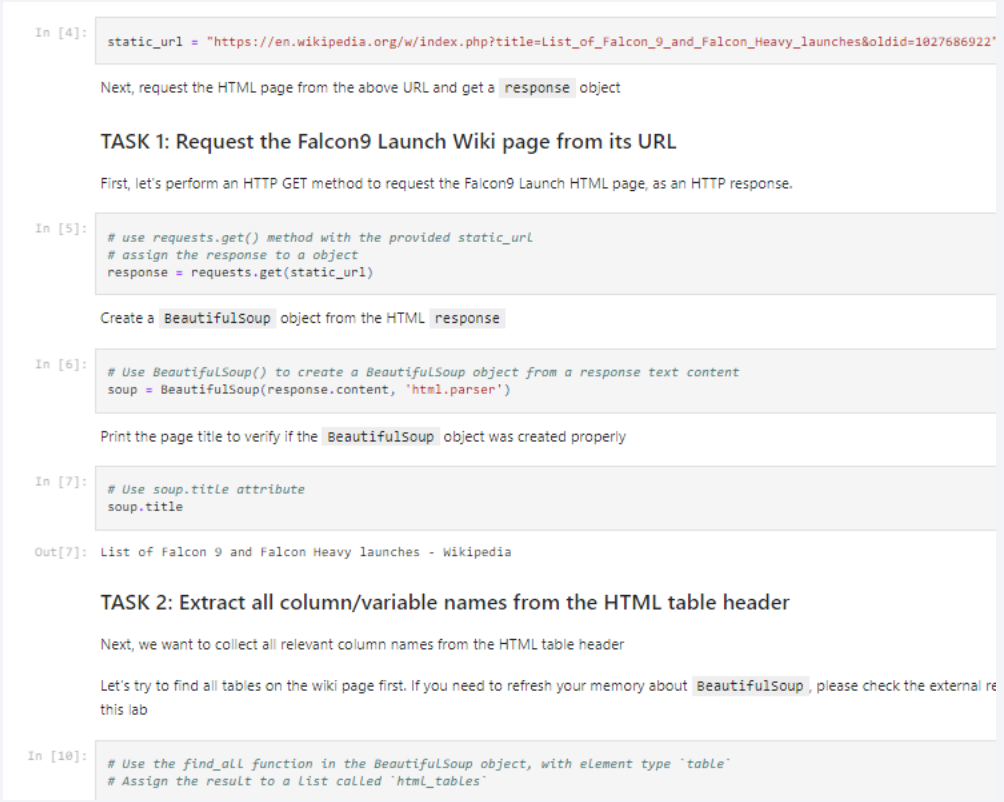
```
# Use json_normalize meethod to convert the json result into a dataframe
respjson = response.json()
data = pd.json_normalize(respjson)
```

Data Collection - Scraping

- Web scraping was used to collect Falcon 9 launch records from a Wikipedia page. Using BeautifulSoup and Requests libraries, the Falcon 9 launch records were extracted from an HTML table and converted into a data frame for further analysis.

- This link to the GitHub URL of the completed web scraping notebook,

https://github.com/mchaninthon/Applied_Data_Science_Capstone/blob/fe0e7aa5764b3a3d15f2a956ad6f9e75039bb97e/2.%20Space-X%20Web%20scraping%20Falcon%209%20and%20Falcon%20Heavy%20Launches%20Records%20from%20Wikipedia.ipynb



In [4]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

Next, request the HTML page from the above URL and get a `response` object

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 [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.content, 'html.parser')
```

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

In [7]:
```
# Use soup.title attribute
soup.title
```

Out[7]:
```
List of Falcon 9 and Falcon Heavy launches - Wikipedia
```

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

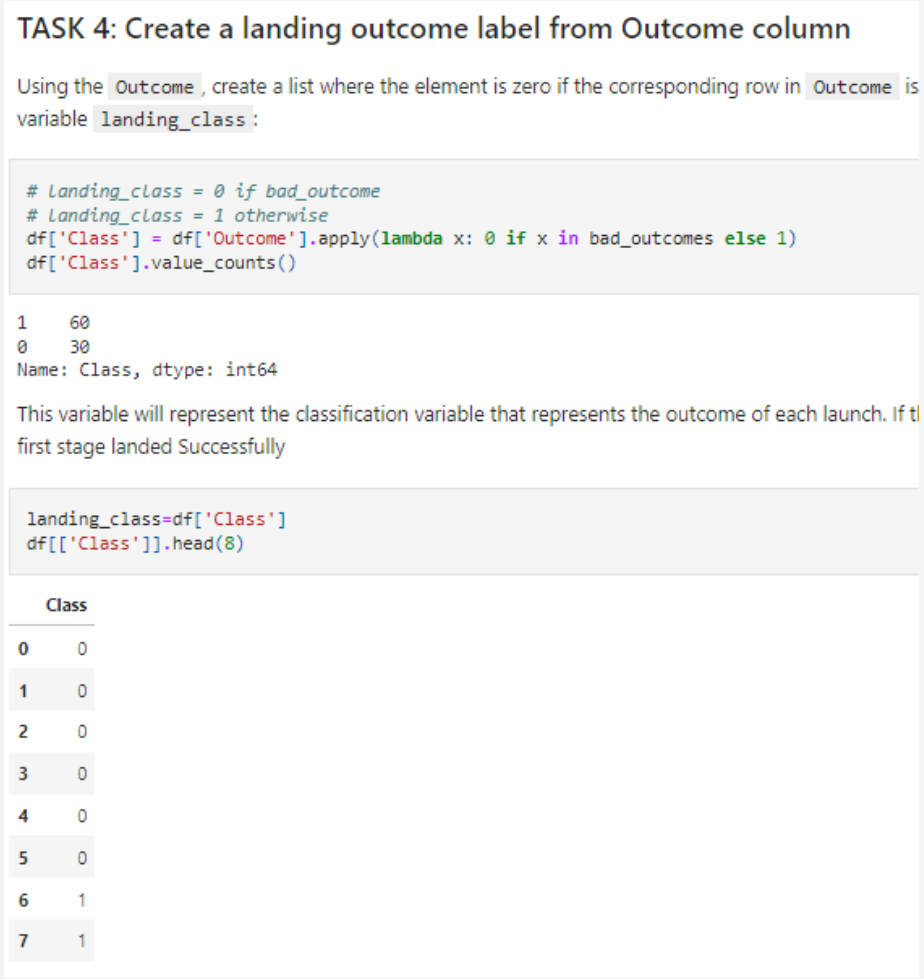
Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external re
this lab

In [10]:
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
```

Data Wrangling

- The data processing involved filtering the data to focus on Falcon 9 launches based on the BoosterVersion column.
- Missing values in the LandingPad and PayloadMass columns were addressed. For the PayloadMass, the missing values were replaced with the mean value of the column.
- Exploratory Data Analysis (EDA) was conducted to identify patterns in the data and determine the label for training supervised models.
- This link to GitHub URL of completed data wrangling related notebooks.

https://github.com/mchaninthon/Applied_Data_Science_Capstone/blob/fe0e7aa5764b3a3d15f2a956ad6f9e75039bb97e/3.%20Space-X%20Data%20Wrangling%20spacex.ipynb



TASK 4: Create a landing outcome label from Outcome column

Using the `Outcome`, create a list where the element is zero if the corresponding row in `Outcome` is variable `landing_class`:

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
df['Class'] = df['Outcome'].apply(lambda x: 0 if x in bad_outcomes else 1)
df['Class'].value_counts()
```

```
1    60
0    30
Name: Class, dtype: int64
```

This variable will represent the classification variable that represents the outcome of each launch. If t
first stage landed Successfully

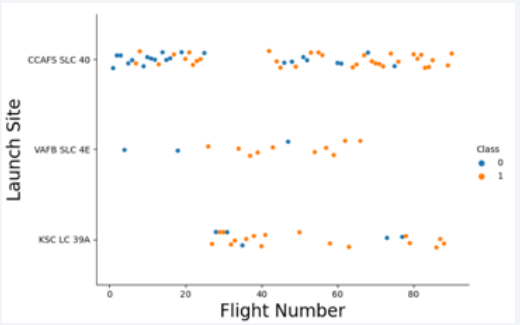
```
landing_class=df['Class']
df[['Class']].head(8)
```

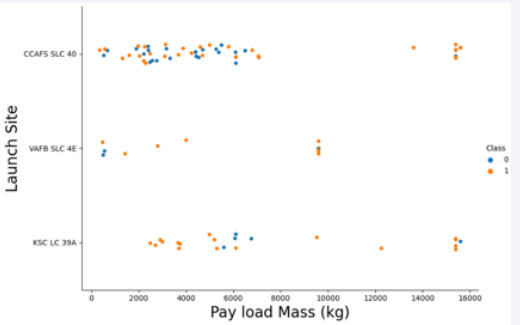
	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

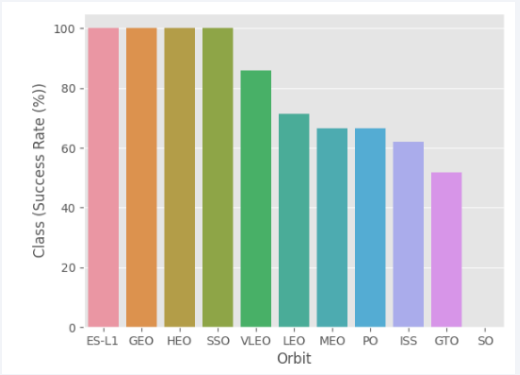
EDA with Data Visualization

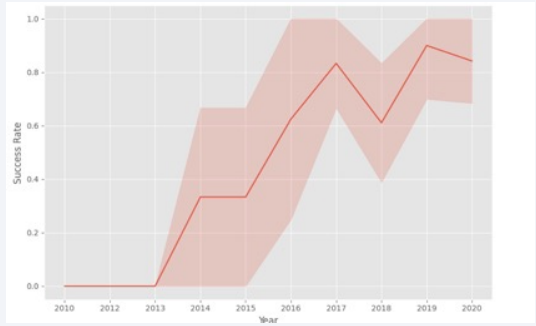
During the data analysis and feature engineering stage, the following tasks were performed using Pandas and Matplotlib:

- Exploratory Data Analysis
- Preparing Data Feature Engineering
- Scatter plots were used to visualize the relationships between Flight Number and Launch Site, Payload and Launch Site, Flight Number and Orbit type, and Payload and Orbit type.
- Bar charts were used to visualize the success rate of each orbit type.
- A line plot was used to visualize the yearly trend of launch success.









- [This link] to the GitHub URL of completed EDA with data visualization notebook.

https://github.com/mchaninthon/Applied_Data_Science_Capstone/blob/fe0e7aa5764b3a3d15f2a956ad6f9e75039bb97e/5.%20Space-X%20EDA%20DataViz%20Using%20Pandas%20and%20Matplotlib%20-%20SpaceX.ipynb

EDA with SQL

- The following SQL queries were performed for EDA
  - Display the names of the unique launch sites in the space mission

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

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

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA (CRS)';
```

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Version LIKE 'F9 v1.1%';
```

  - List the date when the first successful landing outcome in ground pad was achieved

```
%sql SELECT MIN(DATE) FROM 'SPACEXTBL' WHERE "Landing _Outcome" = "Success (ground pad)";
```

EDA with SQL (Cont.)

- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000 (%sql SELECT DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE "Landing _Outcome" = "Success (drone ship)" AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;)
- List the total number of successful and failure mission outcomes

```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";
```

- This link to GitHub URL of completed EDA with SQL notebook.

https://github.com/mchaninthon/Applied_Data_Science_Capstone/blob/fe0e7aa5764b3a3d15f2a956ad6f9e75039bb97e/4.%20Space-X%20EDA%20Using%20SQL.ipynb

Build an Interactive Map with Folium

The following steps were taken to analyze the launch sites and their outcomes:

- Created a Folium map to mark all the launch sites.
- Utilized map objects such as markers, circles, and lines to indicate the success or failure of launches at each launch site.
- Created a launch set outcomes variable, assigning a value of 0 for failure and 1 for success.
- This is link to the GitHub URL of completed interactive map with Folium map

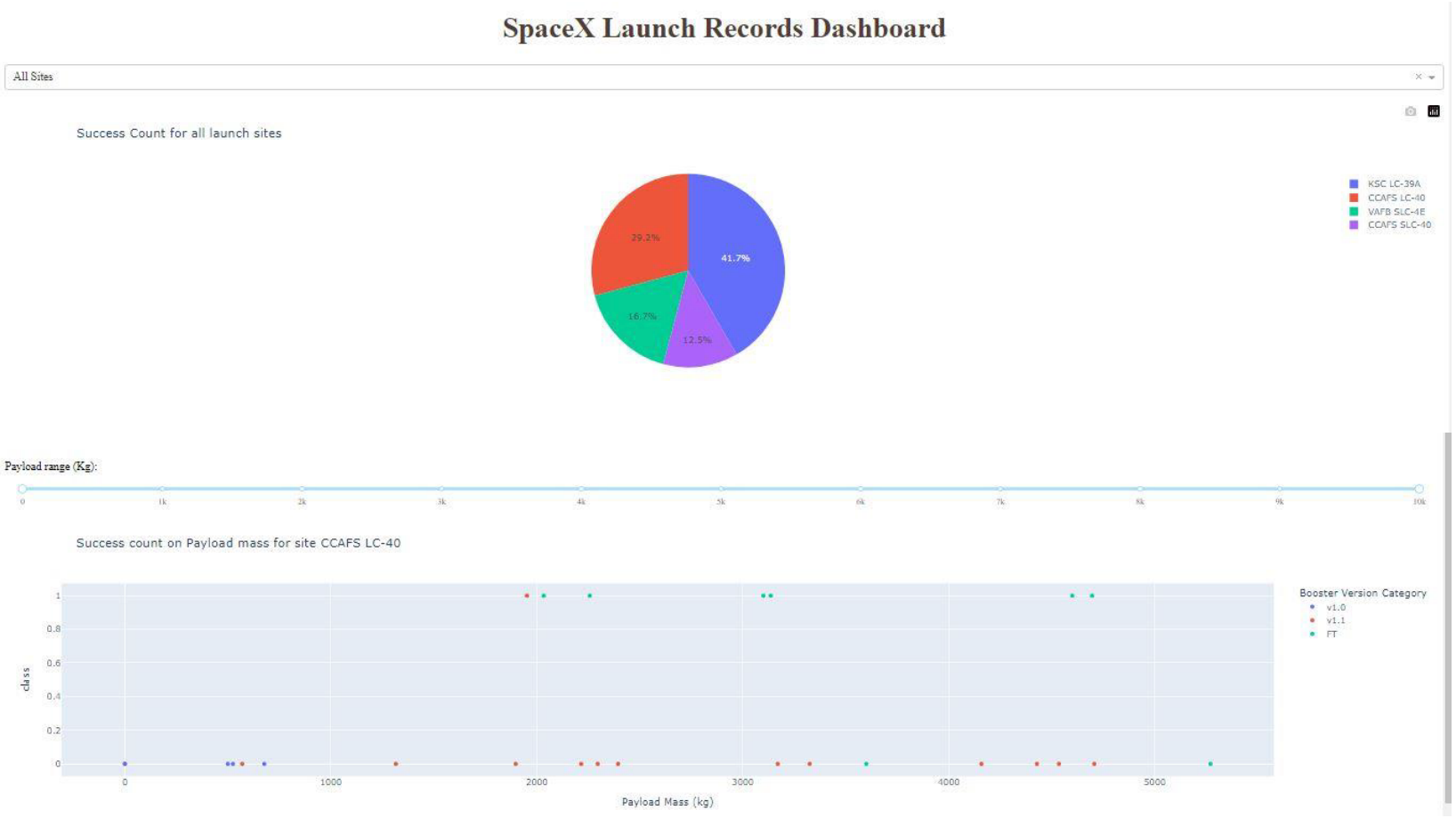
https://github.com/mchaninthon/Applied_Data_Science_Capstone/blob/fe0e7aa5764b3a3d15f2a956ad6f9e75039bb97e/6.Space-X%20Launch%20Sites%20Locations%20Analysis%20with%20Folium-Interactive%20Visual%20Analytics.ipynb

Build a Dashboard with Plotly Dash

- To create an interactive dashboard with Plotly Dash, the following steps were followed:
  - Added a Launch Site Drop-down Input Component to allow site selection.
  - Implemented a callback function to dynamically render a success-pie-chart based on the selected launch site from the dropdown.
  - Included a Range Slider to enable payload selection.
  - Integrated a callback function to generate a success-payload-scatter-chart scatter plot based on the chosen payload range.
- This link is the GitHub URL of completed Plotly Dash lab.

https://github.com/mchaninthon/Applied_Data_Science_Capstone/blob/fe0e7aa5764b3a3d15f2a956ad6f9e75039bb97e/7.%20Build%20an%20Interactive%20Dashboard%20with%20Ploty%20Dash%20-%20spacex_dash_app.py

SpaceX Dash App



Predictive Analysis (Classification)

During the data analysis process, the following steps were taken:

1. Creation of Training Labels:
    1. The column "Class" in the data was converted into a NumPy array using the to_numpy() method.
    2. The resulting array was assigned to the variable Y, representing the outcome variable.
2. Standardization of the Feature Dataset:
    1. The feature dataset (x) was standardized by applying the preprocessing.StandardScaler() function from the Sklearn library.
3. Splitting the Data into Training and Testing Sets:
    1. The data was divided into training and testing sets using the train_test_split function from the sklearn.model_selection module.
    2. The test_size parameter was set to 0.2, indicating that 20% of the data would be used for testing.
    3. The random_state parameter was set to 2 for reproducibility.

Predictive Analysis (Classification)

To determine the best machine learning model, we followed these steps:

1. Model Creation:
    - Created objects for SVM, Classification Trees, k-nearest neighbors, and Logistic Regression.
2. Hyperparameter Tuning:
    - Used GridSearchCV to find the best hyperparameters for each model.
3. Training and Evaluation:
    - Fitted the models with the training data using cross-validation.
    - Identified the best hyperparameters based on validation accuracy.
4. Test Data Evaluation:
    - Calculated the accuracy on the test data for each model.
    - Plotted confusion matrices to compare test outcomes with predictions.

These steps allowed us to evaluate the models, optimize their hyperparameters, and assess their performance on the test data using confusion matrices.

Predictive Analysis (Classification)

- The table below presents the accuracy scores of different methods on the test data, allowing us to compare their performance
- By examining the accuracy scores, we can assess the performance of each method and determine which one performed the best on the test data.
- This Link to GitHub URL of the completed predictive analysis lab

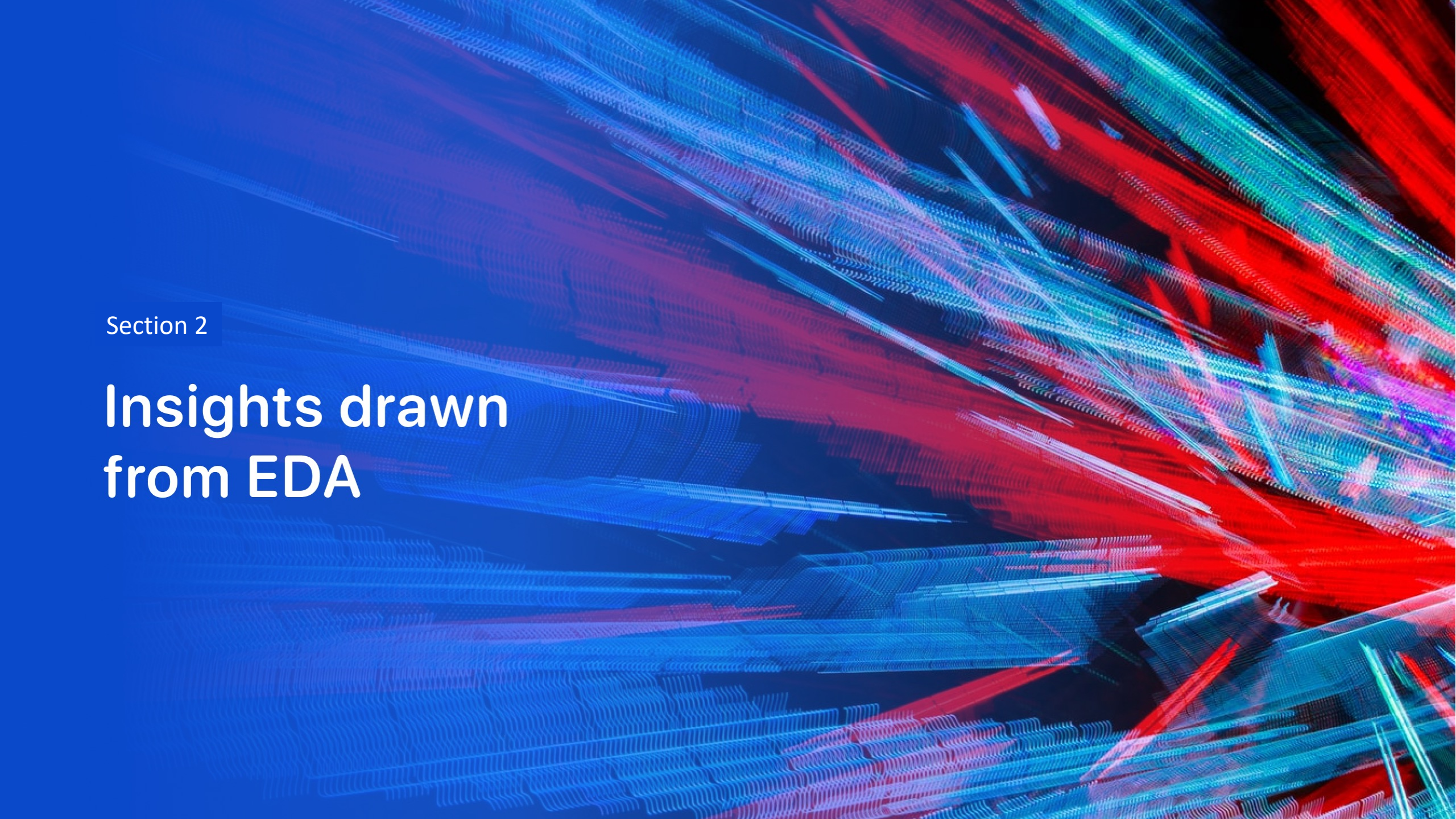
https://github.com/mchaninthon/Applied_Data_Science_Capstone/blob/fe0e7aa5764b3a3d15f2a956ad6f9e75039bb97e/8.%20SpaceX%20Machine%20Learning%20Prediction.ipynb

Out[68]:

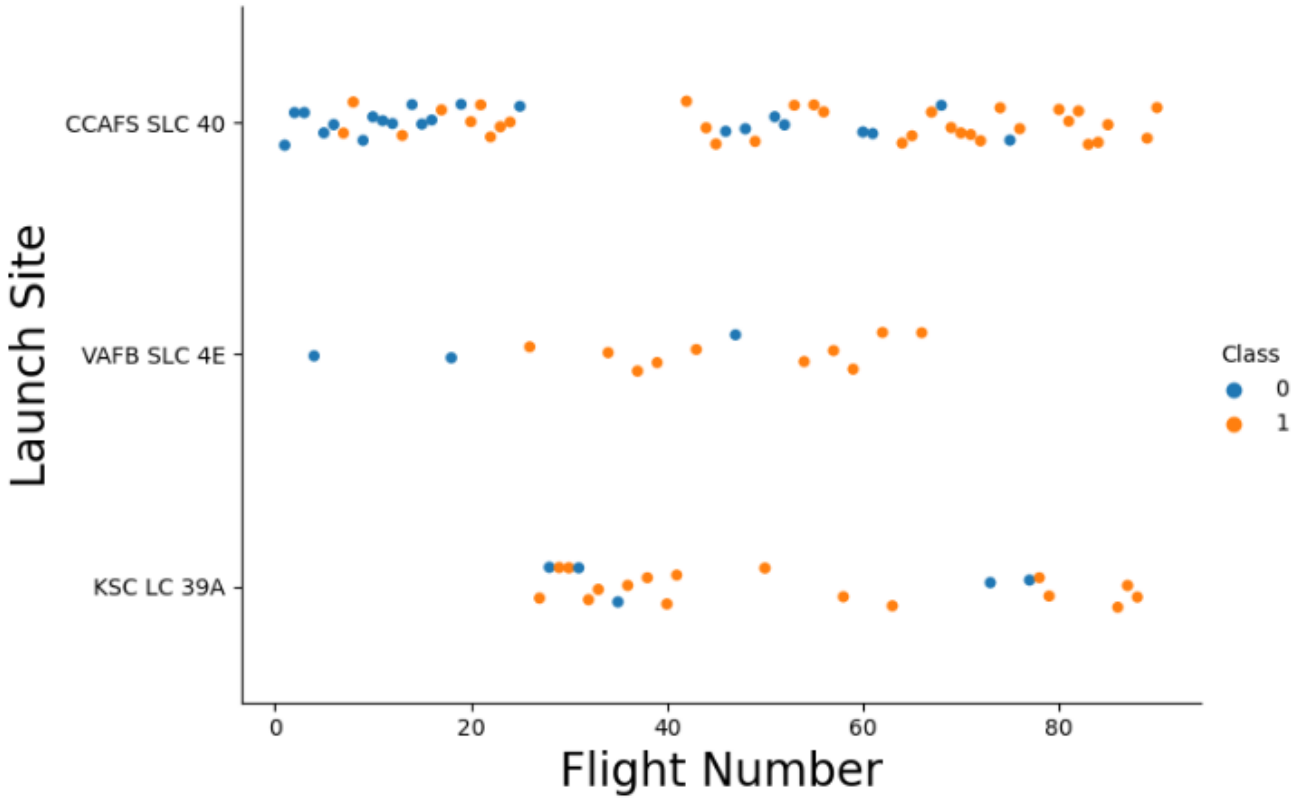
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

Results

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


Section 2
Insights drawn from EDA

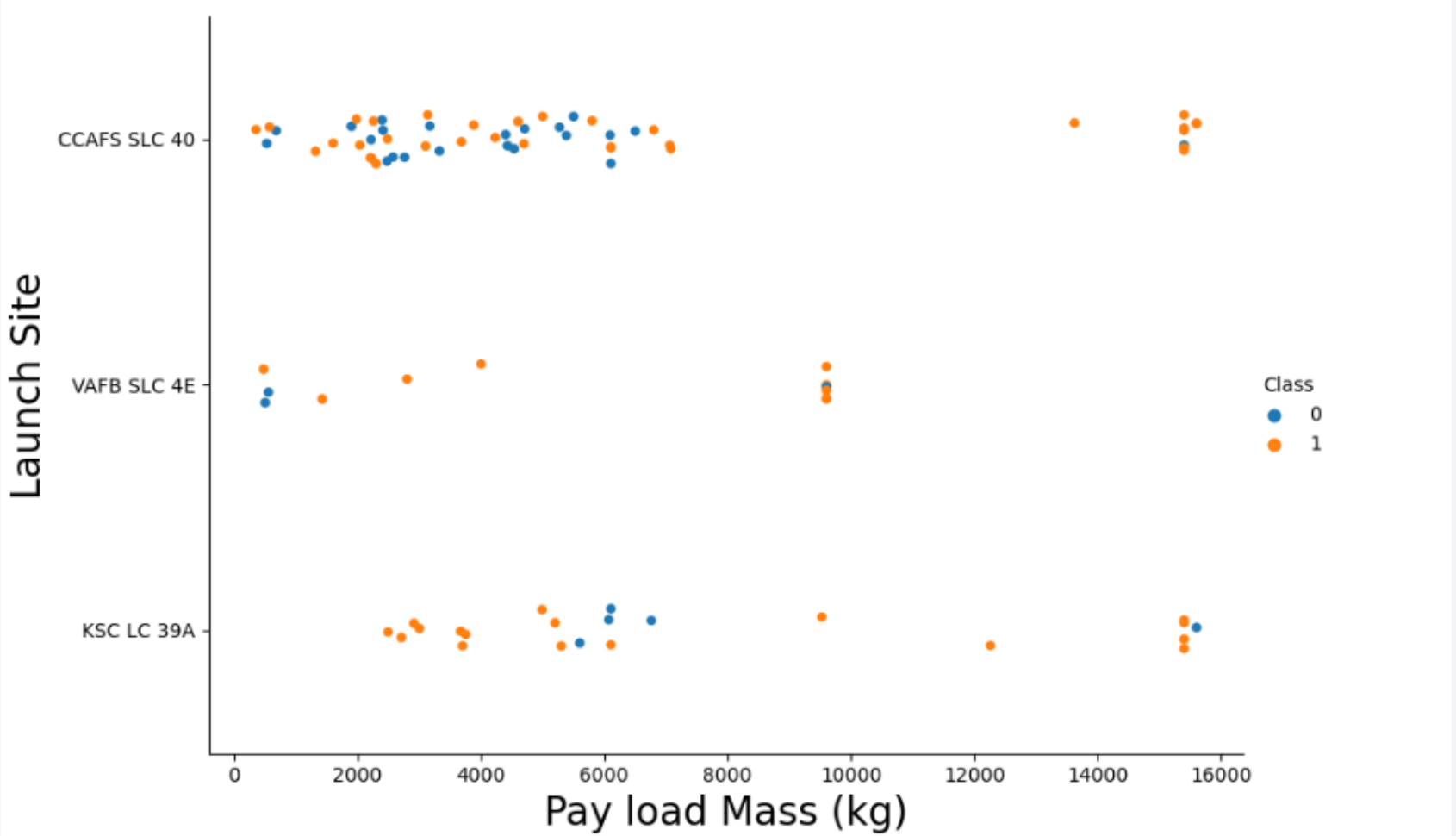
Flight Number vs. Launch Site



Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

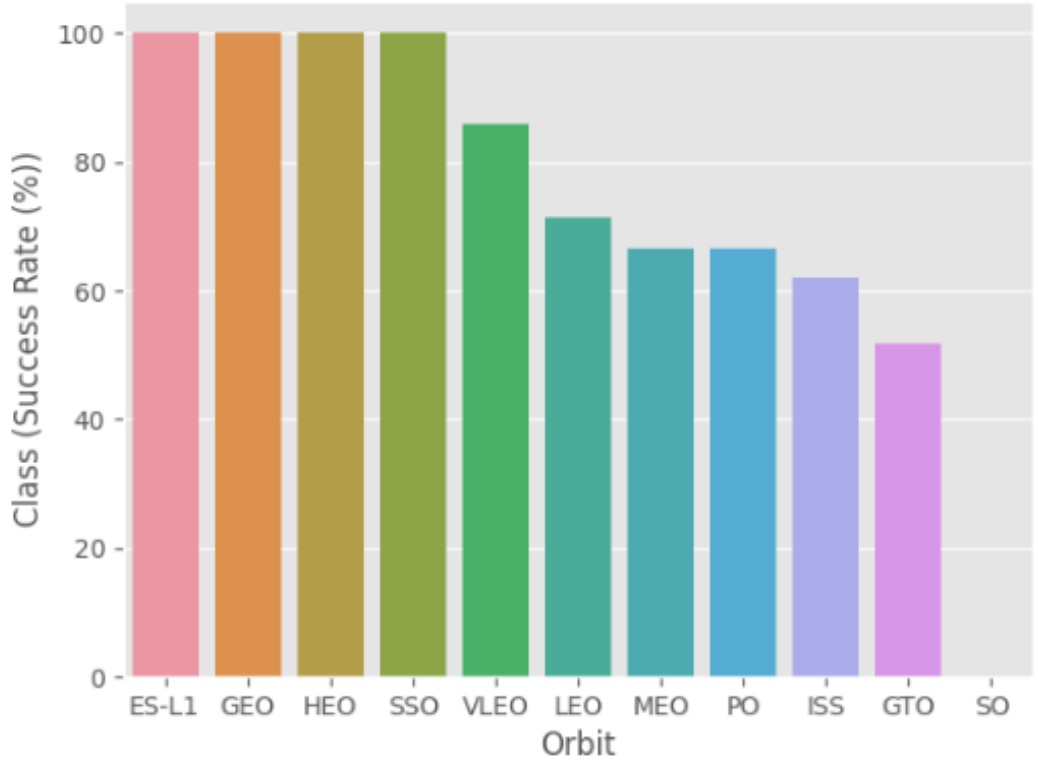
We can deduce that, as the flight number increases in each of the 3 launcg sites, so does the success rate. For instance, the success rate for the VAFB SLC 4E launch site is 100% after the Flight number 50. Both KSC LC 39A and CCAFS SLC 40 have a 100% success rates after 80th flight.

Payload vs. Launch Site



Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

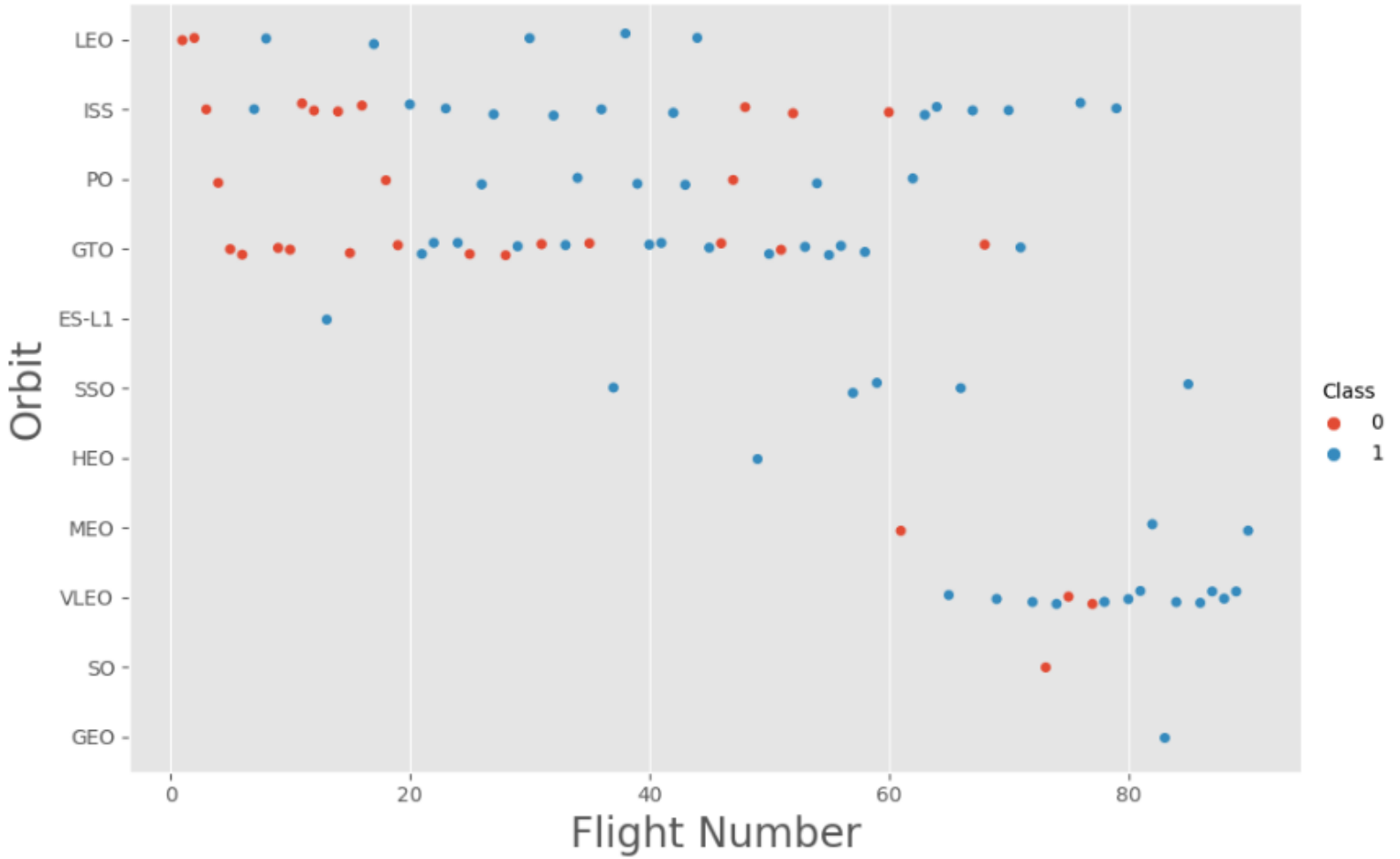
Success Rate vs. Orbit Type



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

Orbits ES-L1, GEO, HEO & SSO have the highest success rates at 100%, with SO orbit having the lowest success rate at ~50%. Orbit SO has 0% success rate.

Flight Number vs. Orbit Type



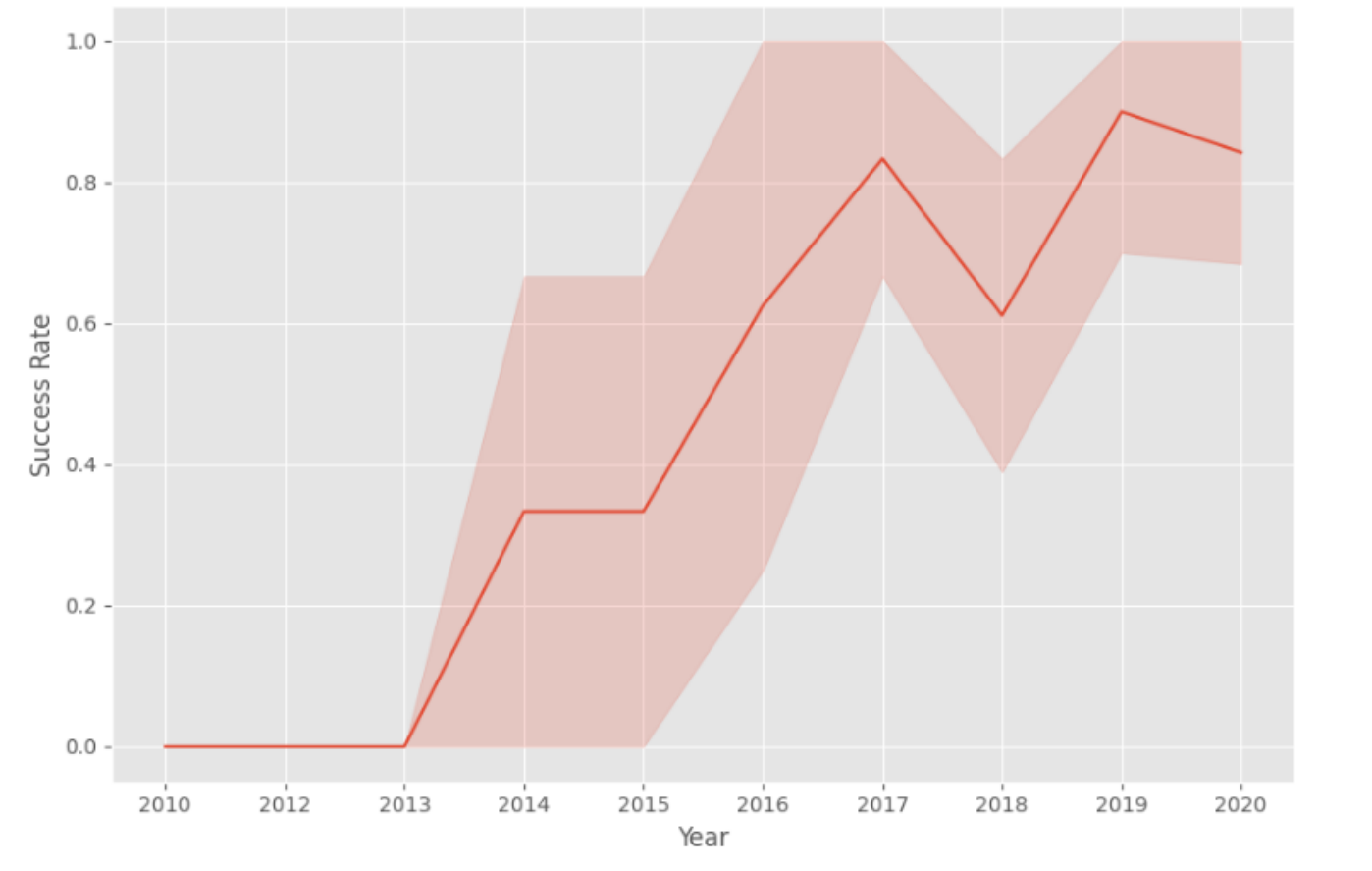
You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type

Heavy payloads have a higher success rate for positive landings in Polar, LEO, and ISS orbits. However, distinguishing the success rate for GTO (Geostationary Transfer Orbit) is challenging as both positive and negative landing rates have similar chances.

Launch Success Yearly Trend

Yearly average success rate



The success rate has consistently increased from 2013 to 2020.

All Launch Site Names

- The 'SELECT DISTINCT' statement was used to retrieve only the unique launch sites from the 'LAUNCH_SITE' column of the SPACEXTBL table.

Task 1

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

In [31]:
```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

Out[31]:

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

Task 2

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

In [72]:
```
%sql SELECT * FROM 'SPACEXTBL' WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

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

Out[72]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- The table was filtered using the 'LIKE' command and the '%' wildcard in the 'WHERE' clause to retrieve and present a subset of records where the launch sites start with the characters 'CCA'.

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

Task 3

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

In [17]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA (CRS)';
```

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

Out[17]:

Total Payload Mass(Kgs)	Customer
45596	NASA (CRS)

- The 'SUM()' function was utilized to calculate and present the cumulative sum of the 'PAYLOAD_MASS_KG' column for the customer 'NASA(CRS)'.

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

Task 4

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Version LIKE 'F9 v1.1%';
```

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

Payload Mass Kgs	Customer	Booster_Version
2534.6666666666665	MDA	F9 v1.1 B1003

- The 'AVG()' function was employed to compute and showcase the average payload mass transported by the booster version F9 v1.1.

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

Task 5

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

Hint:Use min function

```
%sql SELECT MIN(DATE) FROM 'SPACEXTBL' WHERE "Landing _Outcome" = "Success (ground pad)";
```

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

MIN(DATE)
01-05-2017

- The 'MIN()' function was utilized to determine and present the earliest date when the first successful landing on the ground pad ('Success (ground pad)') occurred.

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Task 6

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

```
# %sql SELECT * FROM 'SPACEXTBL'
```

```
%sql SELECT DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE "Landing _Outcome" = "Success (drone ship)" AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOA
```

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

Booster_Version	Payload
F9 FT B1022	JCSAT-14
F9 FT B1026	JCSAT-16
F9 FT B1021.2	SES-10
F9 FT B1031.2	SES-11 / EchoStar 105

- The 'SELECT DISTINCT' statement was employed to retrieve and list the unique names of boosters that have operators greater than 4000 and less than 6000, while also meeting the criteria of having payloads between 4000 and 6000 and a landing outcome of 'Success (drone ship)'.

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

Task 7

List the total number of successful and failure 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
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

- The 'COUNT()' function was utilized in conjunction with the 'GROUP BY' statement to determine the overall count of mission outcomes.

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

```
%sql SELECT "Booster_Version",Payload, "PAYLOAD_MASS__KG_" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

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

Booster_Version	Payload	PAYLOAD_MASS__KG_
F9 B5 B1048.4	Starlink 1 v1.0, SpaceX CRS-19	15600
F9 B5 B1049.4	Starlink 2 v1.0, Crew Dragon in-flight abort test	15600
F9 B5 B1051.3	Starlink 3 v1.0, Starlink 4 v1.0	15600
F9 B5 B1056.4	Starlink 4 v1.0, SpaceX CRS-20	15600
F9 B5 B1048.5	Starlink 5 v1.0, Starlink 6 v1.0	15600
F9 B5 B1051.4	Starlink 6 v1.0, Crew Dragon Demo-2	15600
F9 B5 B1049.5	Starlink 7 v1.0, Starlink 8 v1.0	15600
F9 B5 B1060.2	Starlink 11 v1.0, Starlink 12 v1.0	15600
F9 B5 B1058.3	Starlink 12 v1.0, Starlink 13 v1.0	15600
F9 B5 B1051.6	Starlink 13 v1.0, Starlink 14 v1.0	15600
F9 B5 B1060.3	Starlink 14 v1.0, GPS III-04	15600
F9 B5 B1049.7	Starlink 15 v1.0, SpaceX CRS-21	15600

- By employing a subquery, the maximum payload value was obtained and utilized to identify all boosters that have transported this maximum payload of 15600 kilograms.

2015 Launch Records

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

Task 9

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.

```
%sql SELECT substr(Date,7,4), substr(Date, 4, 2),"Booster_Version", "Launch_Site", Payload, "PAYLOAD_MASS__KG_", "Mission_Outcome", "Landing _Outcome"
```

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

substr(Date,7,4)	substr(Date, 4, 2)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Mission_Outcome	Landing _Outcome
2015	01	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	Success	Failure (drone ship)
2015	04	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	Success	Failure (drone ship)

- The 'substr()' function was applied in the select statement to extract the month and year from the date column. The filter criteria were set to retrieve records where the year was '2015' and the landing outcome was 'Failure (drone ship)'. The query returned the matching records based on these filters.

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

Task 10

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

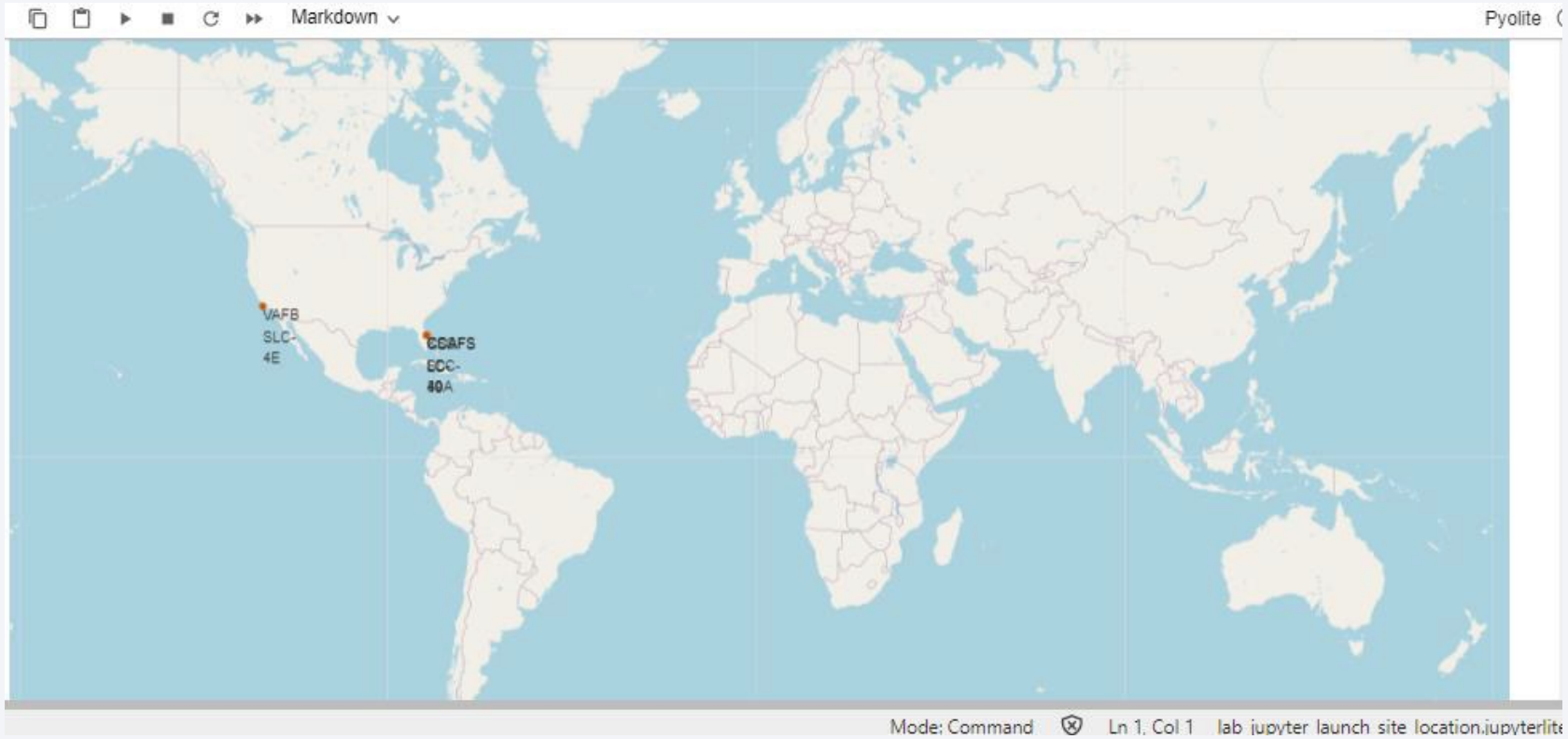
```
%sql SELECT * FROM SPACEXTBL WHERE "Landing _Outcome" LIKE 'Success%' AND (Date BETWEEN '04-06-2010' AND '20-03-2017') ORDER BY Date DESC;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
19-02-2017	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
18-10-2020	12:25:57	F9 B5 B1051.6	KSC LC-39A	Starlink 13 v1.0, Starlink 14 v1.0	15600	LEO	SpaceX	Success	Success
18-08-2020	14:31:00	F9 B5 B1049.6	CCAFS SLC-40	Starlink 10 v1.0, SkySat-19, -20, -21, SAOCOM 1B	15440	LEO	SpaceX, Planet Labs, PlanetIQ	Success	Success
18-07-2016	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
18-04-2018	22:51:00	F9 B4 B1045.1	CCAFS SLC-40	Transiting Exoplanet Survey Satellite (TESS)	362	HEO	NASA (LSP)	Success	Success (drone ship)


Section 3
Launch Sites
Proximities Analysis

Mapping Global Launch Sites



Proximity of Launch Sites to the Equator and Coastline

During the analysis, it was observed that all launch sites are situated in close proximity to the Equator. These sites are primarily located southwards of the US map. Additionally, it was noted that all the launch sites are situated near coastal areas, indicating their strategic placement for efficient launch operations.

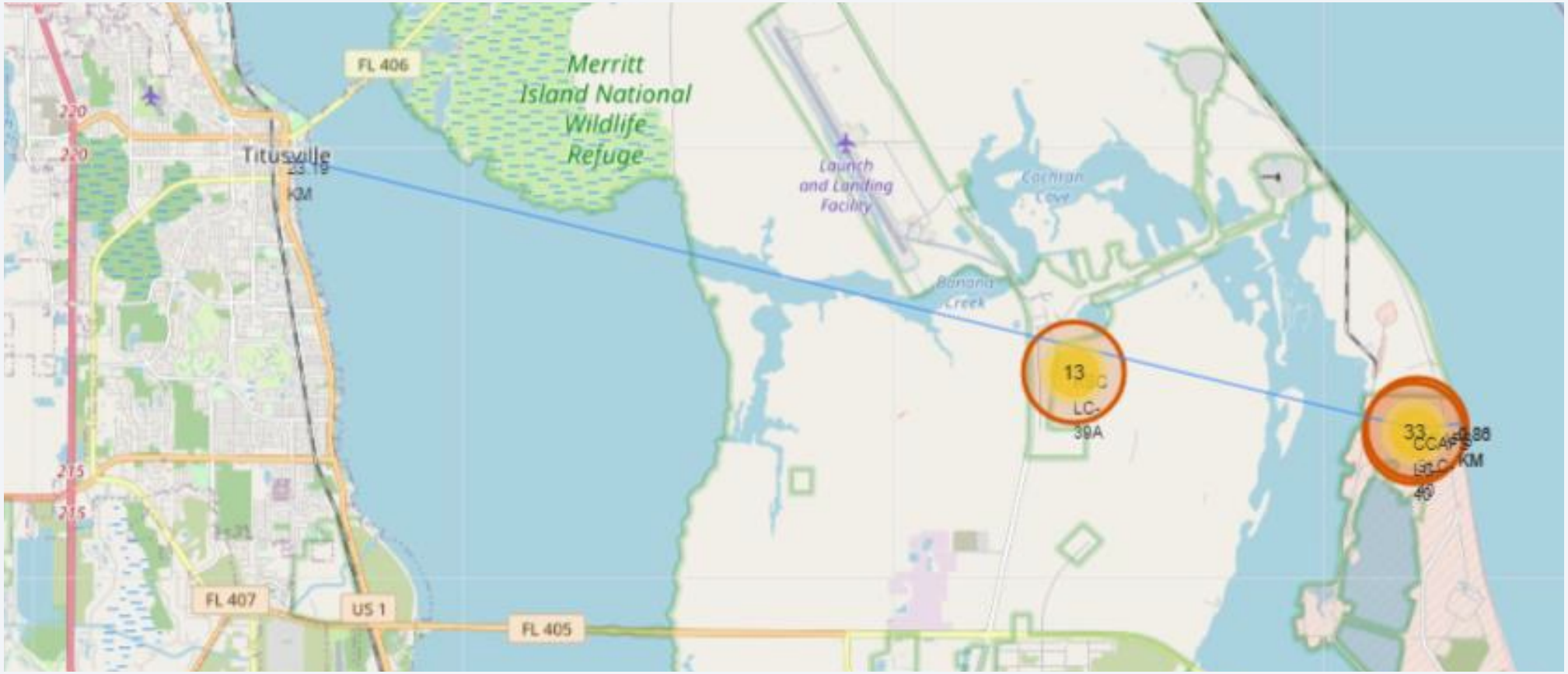
Launch Site Outcomes on Map with Color Markers

Comparing launch sites on the eastern coast, KSC LC-39A at Florida exhibits higher success rates in contrast to CCAFS SLC-40 and CCAFS LC-40.

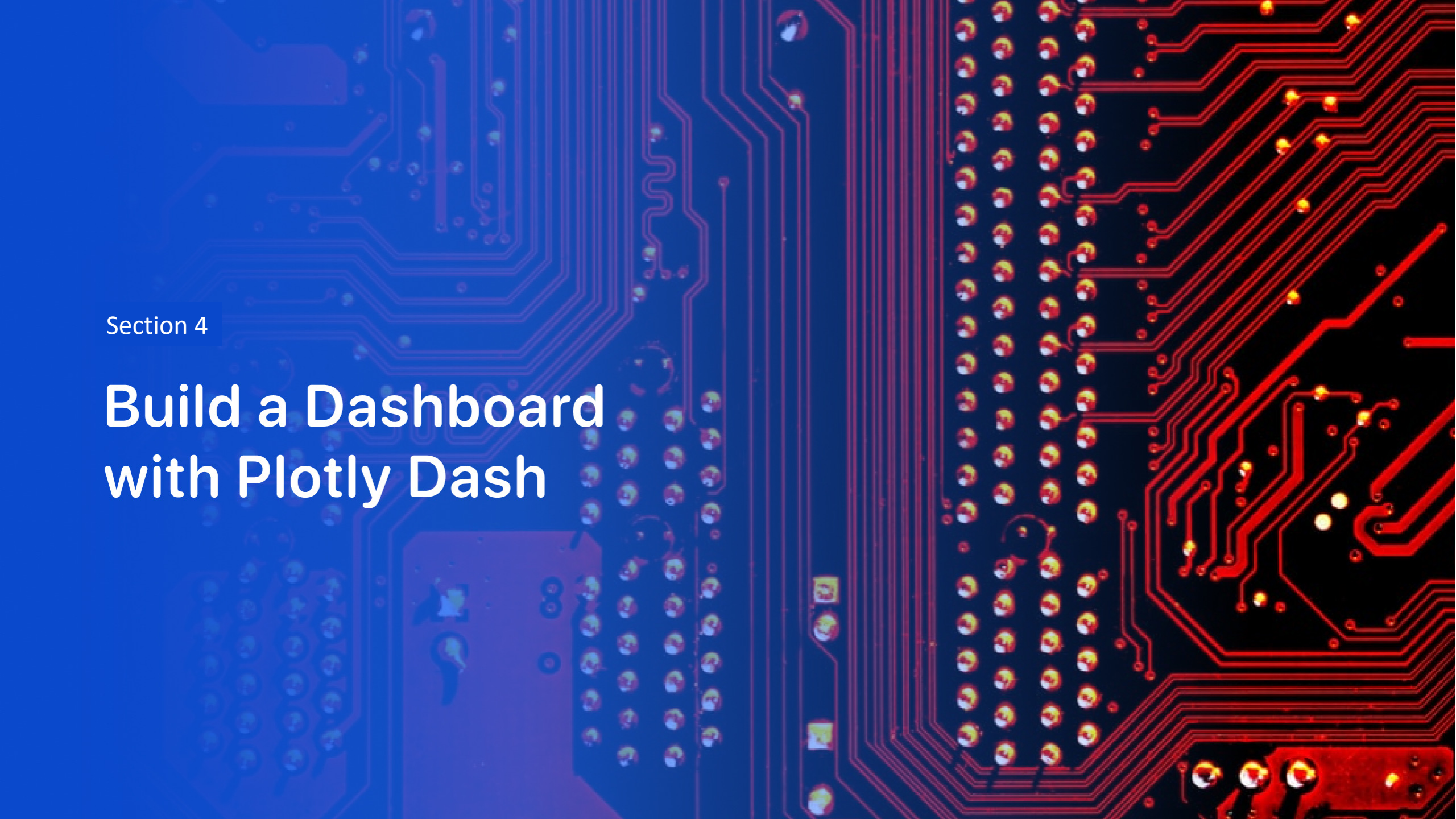
Proximities of Launch Sites: Measuring the Distances

Proximity of Launch Site CCAFS SLC-40 to Coastline: 0.86km

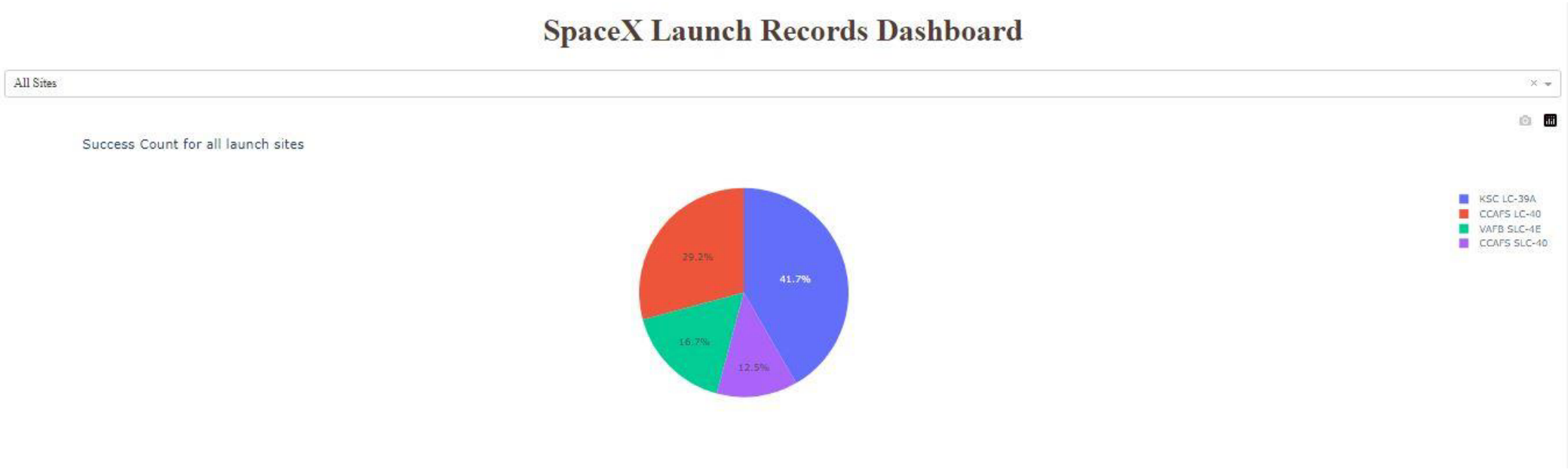
Proximity Distances of Launch Sites



Closest Highway Distance to Launch Site CCAFS SLC-40: 23.19km

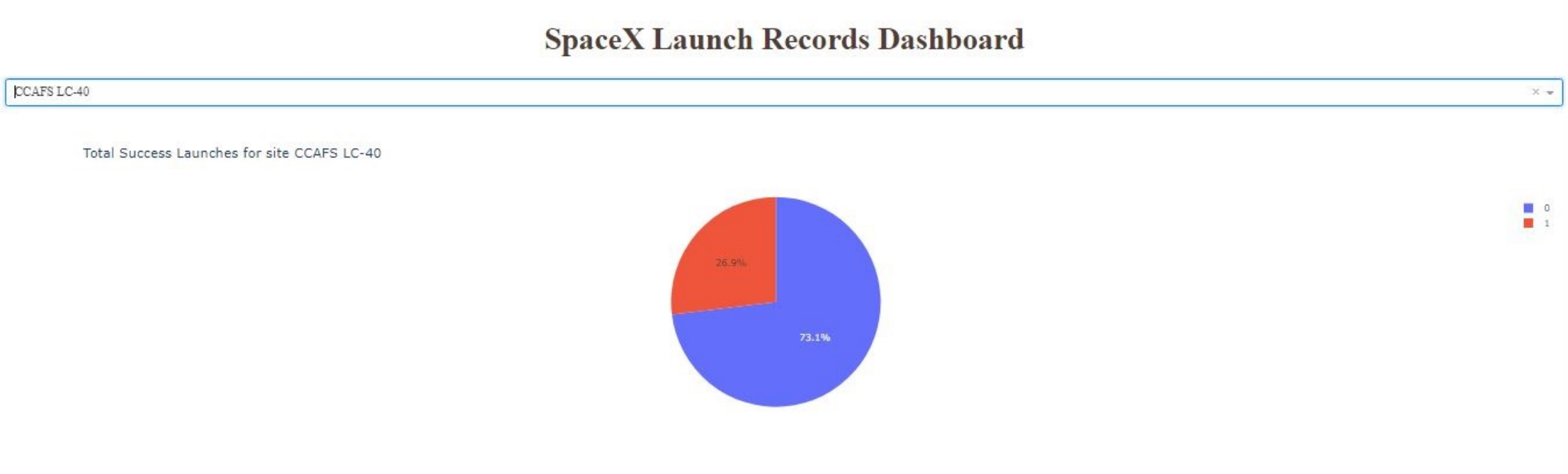

Section 4
Build a Dashboard with Plotly Dash

Launch Success Count for All Sites: Pie Chart



The launch site with the highest launch success rate is KSC LC-39A, with a rate of 42%. CCAFS LC-40 follows closely at 29%, while VAFB SLC-4E has a success rate of 17%. The launch site with the lowest success rate is CCAFS SLC-40, with only 13% of launches resulting in success.

2nd Highest Launch Success Ratio : Pie Chart



The launch site CCAFS LC-40 achieved a commendable success ratio of 73%, with 73% of the launches being successful and only 27% resulting in failure. This places CCAFS LC-40 as the second highest in terms of success ratio among all the launch sites.

All sites Payload vs Launch Outcome : Scatter plot

The launch site CCAFS LC-40 has achieved the highest success rate for the booster version FT when carrying payload masses greater than 2000kg.

Section 5

Predictive Analysis (Classification)

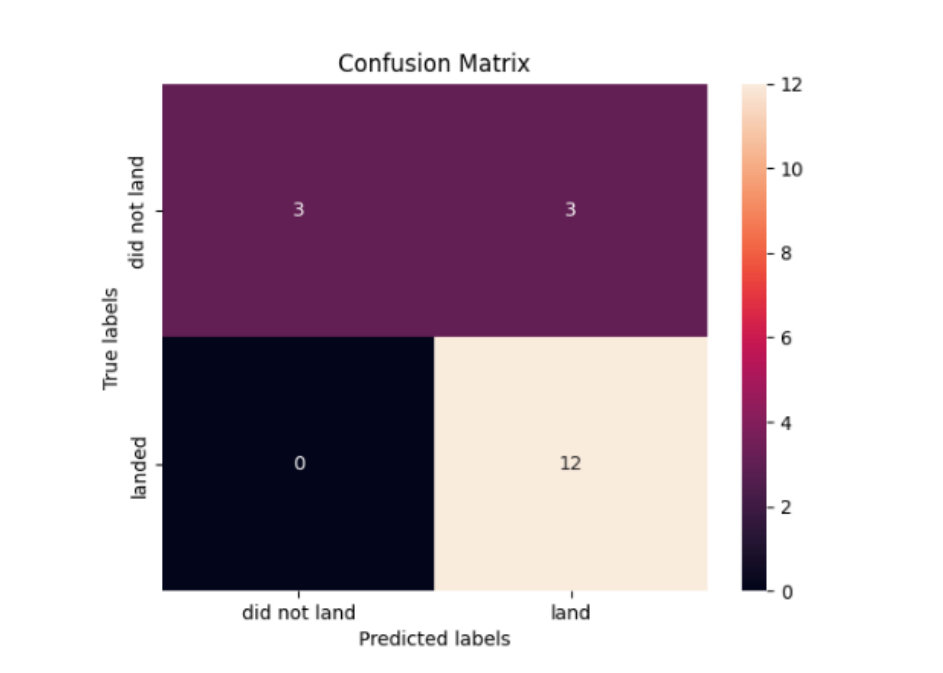
Classification Accuracy

Out[68]:

	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

All the methods perform equally on the test data: i.e. They all have the same accuracy of 0.833333 on the test Data

Confusion Matrix



In the evaluation of all four classification models, it was observed that they exhibited similar confusion matrixes, indicating their ability to equally distinguish between different classes. However, a common challenge identified across all models was the occurrence of false positives, which emerged as a significant issue to address.

Conclusions

- Different launch sites have different success rates:
  - CCAFS LC-40: 60% success rate
  - KSC LC-39A and VAFB SLC 4E: 77% success rate
- Success rate increases with flight number for each launch site.
- No rockets launched for heavy payload masses (>10000) at VAFB-SLC launch site.
- Orbits ES-L1, GEO, HEO, and SSO have 100% success rates, while the SO orbit has a 50% success rate, and the GTO orbit has no clear distinction between success and failure.
- Success in LEO orbit is related to the number of flights, while GTO orbit shows no correlation.
- Positive landing rates are higher for Polar, LEO, and ISS orbits with heavy payloads, but not distinguishable for GTO.
- Success rate has been increasing from 2013 to 2020.

Thank you!