IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Luis A. Gil Lares
December 2023

Outline

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

Executive Summary

Summary of methodologies

- SpaceX Data Collection using SpaceX API
- SpaceX Data Collection with Web Scraping
- SpaceX Data Wrangling
- SpaceX Exploratory Data Analysis using SQL
- Space-X EDA DataViz Using Python Pandas and Matplotlib
- Space-X Launch Sites Analysis with Folium-Interactive Visual Analytics and Ploty Dash
- SpaceX Machine Learning Landing Prediction

Summary of all results

- EDA results
- Interactive Visual Analytics and Dashboards
- Predictive Analysis(Classification)

Introduction

- **Project background and context**

SpaceX promotes Falcon 9 rocket launches on its website at a price of $62 million, while other providers charge over $165 million per launch. The significant cost difference arises from SpaceX's ability to reuse the initial stage of the rocket. Consequently, by assessing whether the first stage can be successfully recovered, one can ascertain the overall cost of a launch. This knowledge becomes valuable in scenarios where another company seeks to compete with SpaceX in bidding for a rocket launch contract.

- **Problems you want to find answers**

In this capstone project, our objective is to forecast the successful landing of the Falcon 9 first stage by analyzing data extracted from Falcon 9 rocket launches as presented on its official website.

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

Initially, data retrieval commenced through the utilization of the SpaceX API, a RESTful API, achieved by initiating a GET request to the SpaceX API. This involved the creation of a set of auxiliary functions designed to facilitate the extraction of information from the API using unique identification numbers within the launch data. Subsequently, rocket launch data was solicited from the SpaceX API URL.

To enhance the consistency of the obtained JSON results, the SpaceX launch data was subjected to a GET request, followed by parsing and decoding the response content into a JSON format. The resultant JSON data was then transformed into a Pandas data frame for further analysis.

Simultaneously, web scraping techniques were employed to amass historical Falcon 9 launch records from a Wikipedia page titled "List of Falcon 9 and Falcon Heavy launches." The launch records were embedded in an HTML structure. Leveraging BeautifulSoup and request libraries, I extracted the Falcon 9 launch HTML table records from the Wikipedia page. Subsequently, the extracted table was parsed and converted into a Pandas data frame for comprehensive utilization.

Data Collection – SpaceX API

- Data acquisition involved utilizing the SpaceX API, a RESTful API, through a GET request to retrieve information. Subsequently, the SpaceX launch data was requested and parsed using another GET request. The response content was decoded into a JSON result, which was then transformed into a Pandas data frame for further analysis.

- https://github.com/luisgil1989/IBM-Capstone-Data-science/blob/main/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
```

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

I employed web scraping techniques to gather historical Falcon 9 launch records from a Wikipedia page. Using BeautifulSoup and the requests library, I extracted the Falcon 9 launch records from the HTML table on the Wikipedia page. Subsequently, I created a data frame by parsing the launch HTML, facilitating further analysis and exploration of the collected information.

https://github.com/luisgil1989/IBM-Capstone-Data-science/blob/main/2.%20Space-X%20Web%20scraping%20Falcon%209%20and%20Falcon%20Heavy%20Launches%20Records%20from%20Wikipedia.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.

```
# 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`

```
# 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

```
# Use soup.title attribute
soup.title
```

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

Data Wrangling

- Once the Pandas DataFrame was successfully generated from the gathered data, a filtration process was implemented using the BoosterVersion column to retain only the Falcon 9 launches. Following this, attention was directed towards addressing missing data values in the LandingPad and PayloadMass columns.
- For the PayloadMass column, the missing data points were substituted with the mean value of the column, ensuring a more complete dataset.
- Additionally, an Exploratory Data Analysis (EDA) was conducted to discern patterns within the data. This analysis aimed to identify potential labels for training supervised models, contributing to a more informed understanding of the dataset.
- https://github.com/luisgil1989/IBM-Capstone-Data-science/blob/main/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 in the set `bad_outcome`; otherwise, it's one. Then assign it to the 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 the value is zero, the first stage did not land successfully; one means the 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

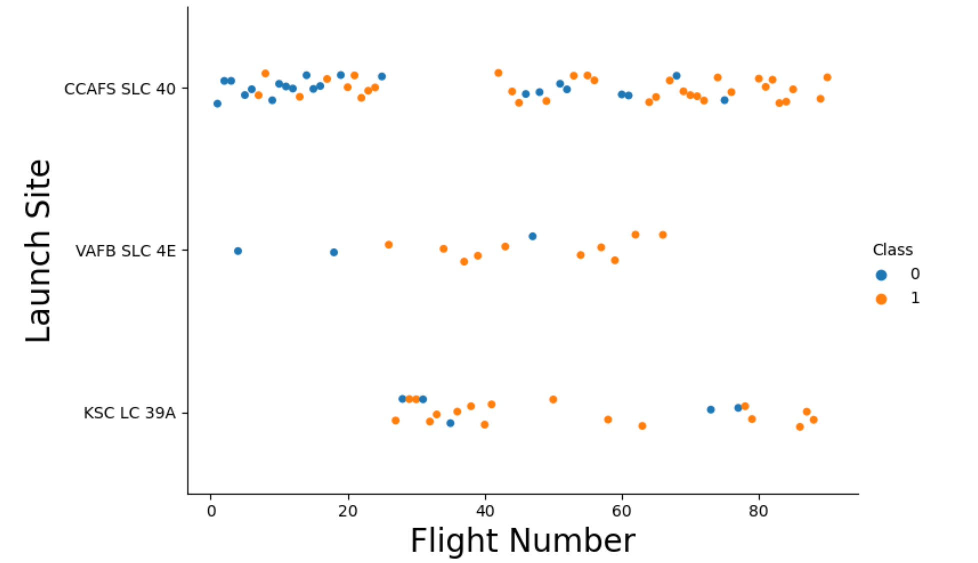
EDA with Data Visualization

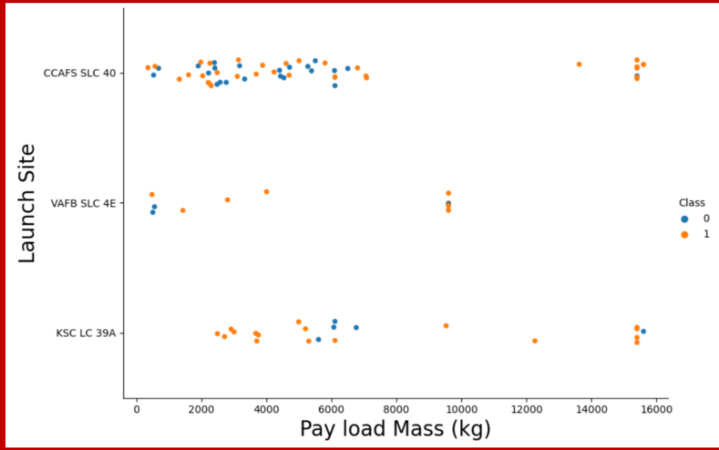
Conducted a comprehensive data analysis and feature engineering process using Pandas and Matplotlib, encompassing:

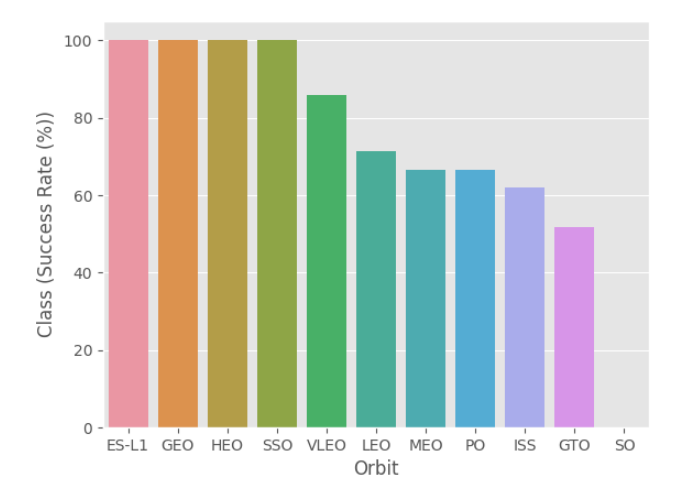
- **Exploratory Data Analysis (EDA):** Explored and examined the dataset to gain insights into its structure and characteristics.
- **Preparing Data and Feature Engineering:** Engaged in the manipulation and enhancement of dataset features to optimize for subsequent analyses.
- **Utilized Scatter Plots:** Employed scatter plots to visually represent relationships between variables such as Flight Number and Launch Site, Payload and Launch Site, Flight Number and Orbit type, as well as Payload and Orbit type.
- **Implemented Bar Charts:** Utilized bar charts to illustrate the success rate of each orbit type, providing a clear visual representation of the data.
- **Leveraged Line Plots:** Employed line plots to depict the annual trends in launch success, offering a visual narrative of the historical performance over time.

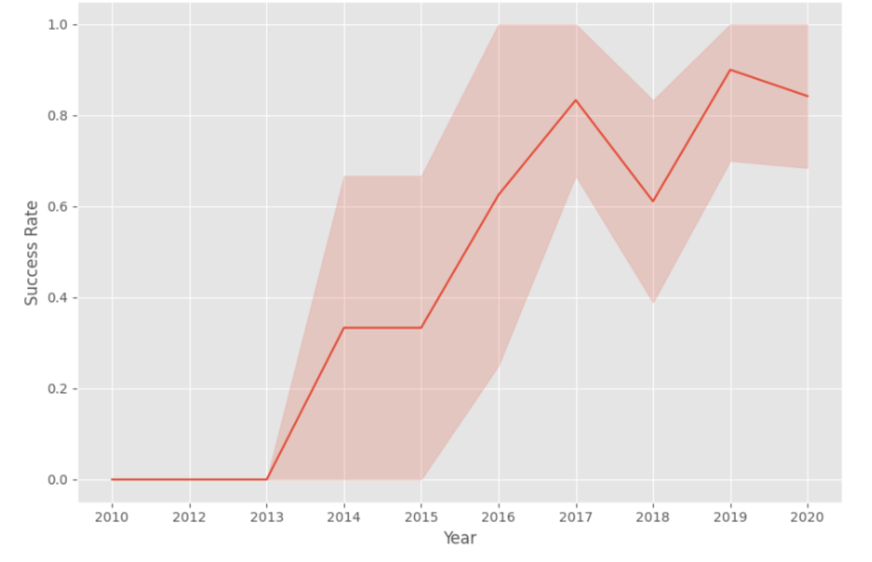
https://github.com/luisgil1989/IBM-Capstone-Data-science/blob/main/5.%20Space-X%20EDA%20DataViz%20Using%20Pandas%20and%20Matplotlib%20-%20SpaceX.ipynb

EDA with Data Visualization









EDA with SQL

https://github.com/cgatama/SpaceX-Falcon-9-1st-stage-Success-Landing-Prediction/blob/main/4.%20Space-X%20EDA%20Using%20SQL.ipynb

- 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%';
```

Build an Interactive Map with Folium

Generated a Folium map to pinpoint all the launch sites and utilized map objects such as markers, circles, and lines to visually represent the success or failure of launches at each launch site. Additionally, established a launch set outcomes variable where failure is denoted as 0 and success as 1

https://github.com/luisgil1989/IBM-Capstone-Data-science/blob/main/5.%20Space-X%20EDA%20DataViz%20Using%20Pandas%20and%20Matplotlib%20-%20SpaceX.ipynb

Build a Dashboard with Plotly Dash

Developed an interactive dashboard application using Plotly Dash by:

- Incorporating a Launch Site Drop-down Input Component
- Establishing a callback function to dynamically generate a success-pie-chart based on the selected launch site from the dropdown
- Integrating a Range Slider to facilitate the selection of payload ranges
- Implementing a callback function to generate a success-payload-scatter-chart scatter plot based on the chosen payload range.

https://github.com/luisgil1989/IBM-Capstone-Data-science/blob/main/5.%20Space-X%20EDA%20DataViz%20Using%20Pandas%20and%20Matplotlib%20-%20SpaceX.ipynb

Predictive Analysis (Classification)

- Providing an overview of the process involved in constructing, assessing, refining, and identifying the most effective classification model.

- Upon loading the data into a Pandas Dataframe, my initial steps involved conducting Exploratory Data Analysis (EDA) and determining the Training Labels. This was achieved by creating a NumPy array from the 'Class' column in the dataset, utilizing the to_numpy() method, and then assigning the resulting array to the variable 'Y' as the outcome variable.

- Subsequently, I standardized the feature dataset ('X') by applying the preprocessing.StandardScaler() function from Scikit-learn. Following this normalization step, the data was partitioned into training and testing sets using the train_test_split function from sklearn.model_selection. The test_size parameter was set to 0.2, and the random_state was configured to 2 for reproducibility.

Predictive Analysis (Classification)

To identify the most effective machine learning model/method for optimal performance with the test data among SVM, Classification Trees, k-nearest neighbors, and Logistic Regression, the following steps were undertaken:

Initialized an object for each algorithm and created a GridSearchCV object, assigning a specific set of parameters for each model.

For each model under consideration, a GridSearchCV object was instantiated with cross-validation (cv) set to 10. Subsequently, the training data was fitted into the GridSearch object for each model to determine the best hyperparameters.

After the training set was successfully fitted, the GridSearchCV object was examined for each model, and the best parameters were displayed using the best_params_ attribute. Additionally, the accuracy on the validation data was revealed using the best_score_ attribute.

Finally, the accuracy on the test data for each model was calculated using the score method, and a confusion matrix was plotted for each, utilizing both the actual and predicted outcomes.

Predictive Analysis (Classification)

The table provided below presents the accuracy scores of the test data for each method, facilitating a comparison to determine the superior performance among SVM, Classification Trees, k-nearest neighbors, and Logistic Regression.

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

https://github.com/luisgil1989/IBM-Capstone-Data-science/blob/main/8.%20SpaceX%20Machine%20Learning%20Prediction.ipynb

Results

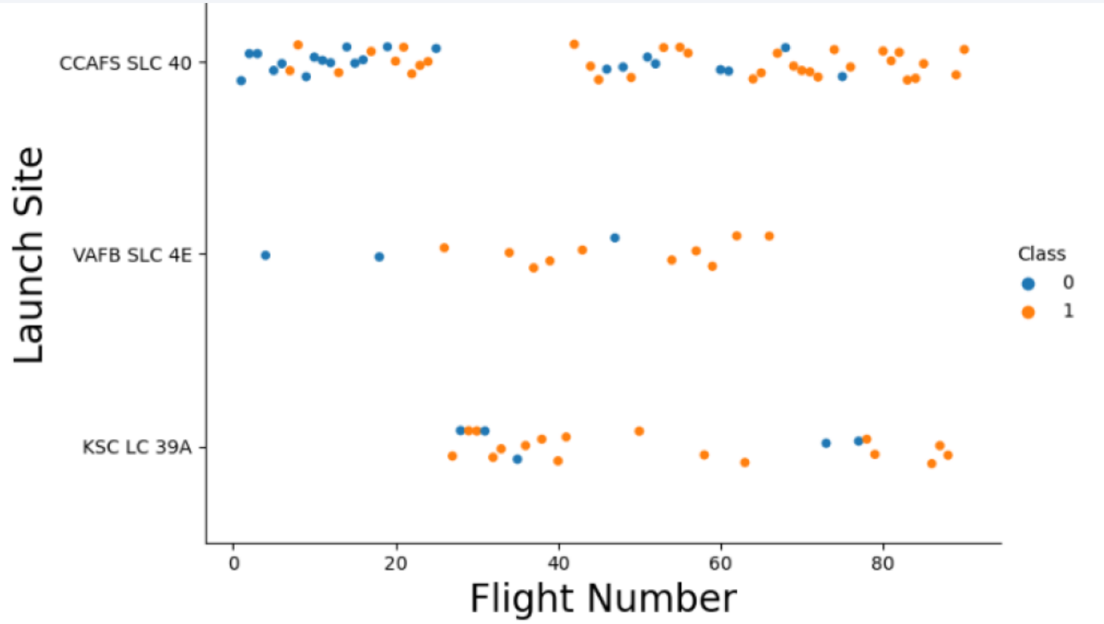
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site



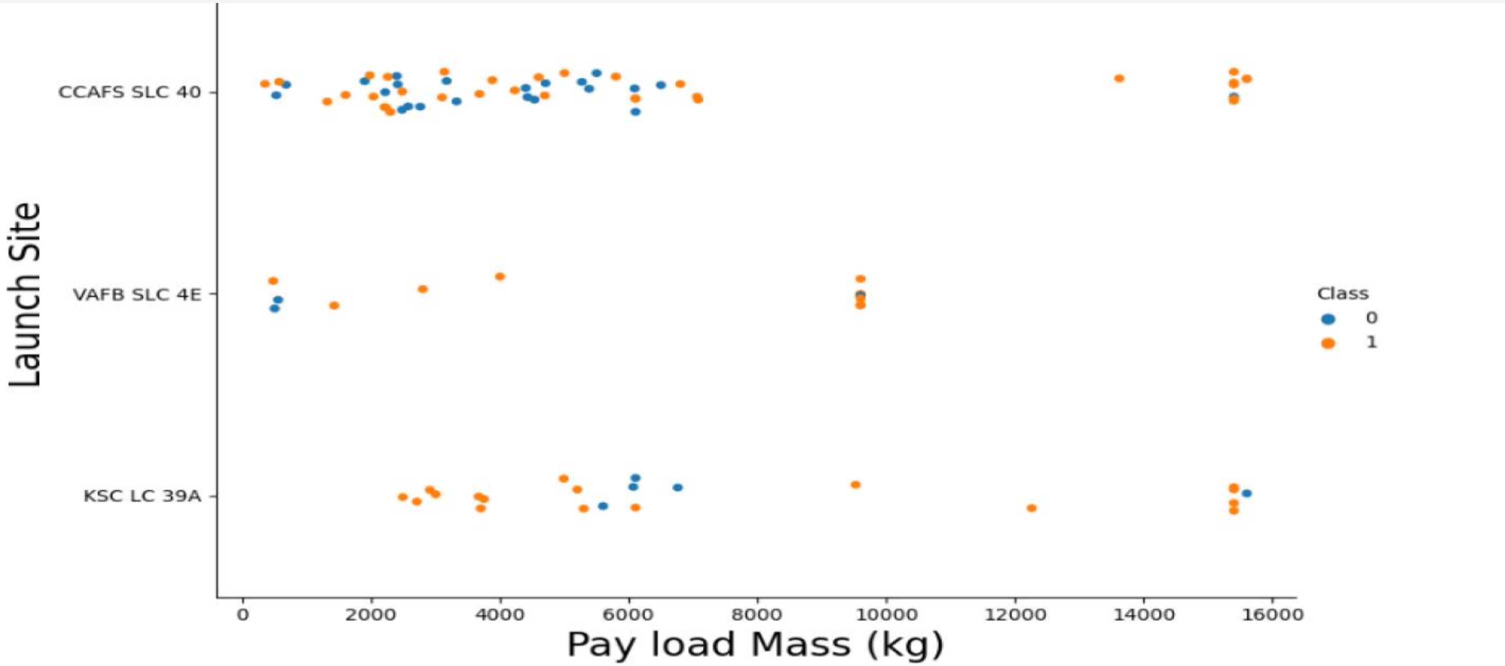
Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

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.

- Show the screenshot of the scatter plot with explanations

Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site

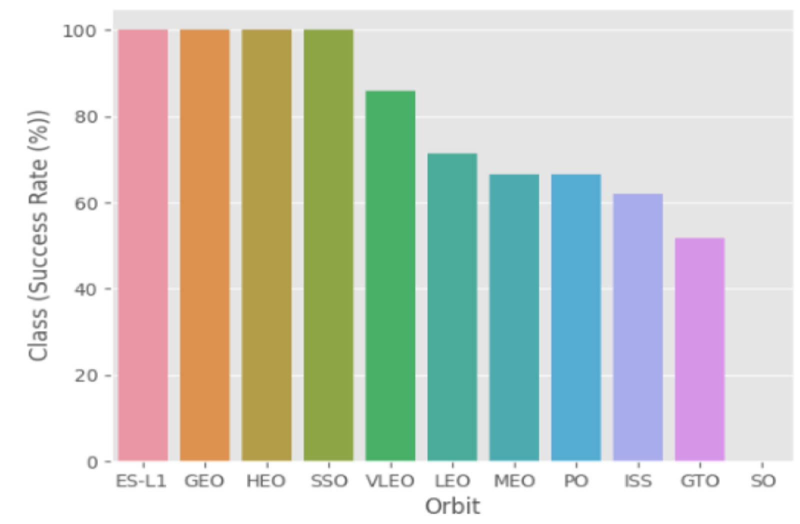


Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

- Show the screenshot of the scatter plot with explanations

Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type



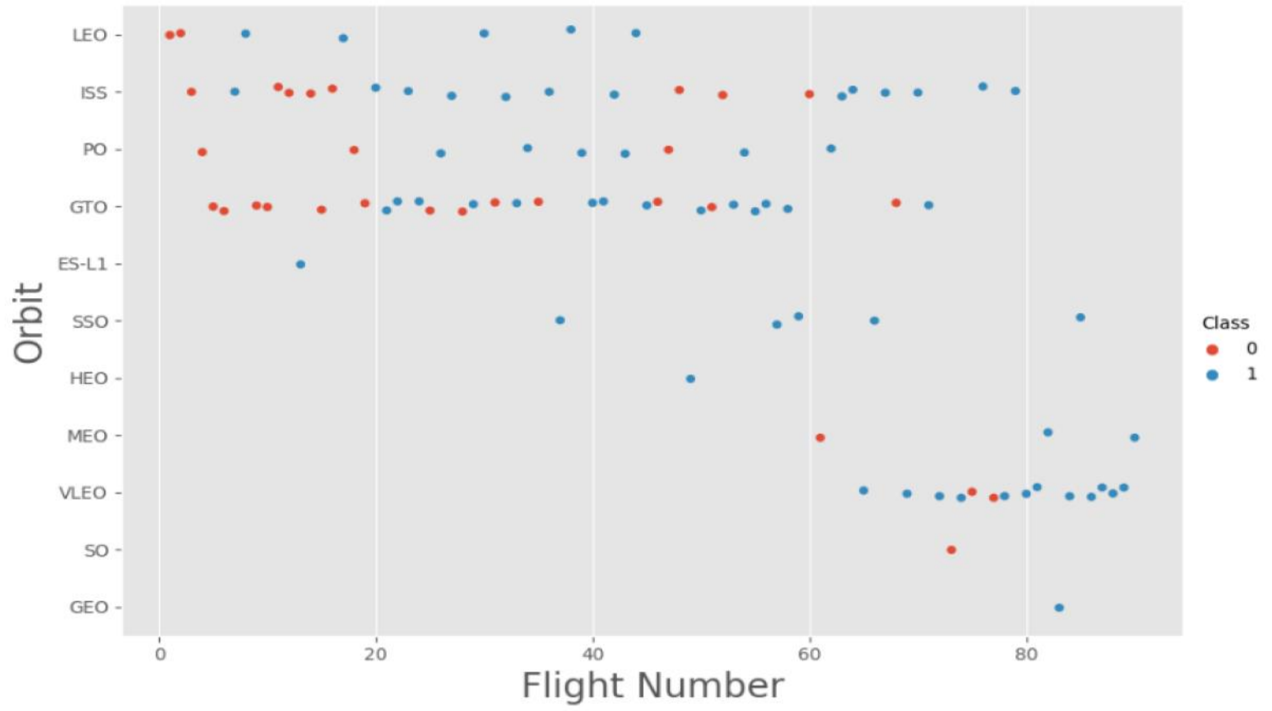
Analyze the ploted bar chart try to find which orbits have high sucess rate.

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.

- Show the screenshot of the scatter plot with explanations

Flight Number vs. Orbit Type

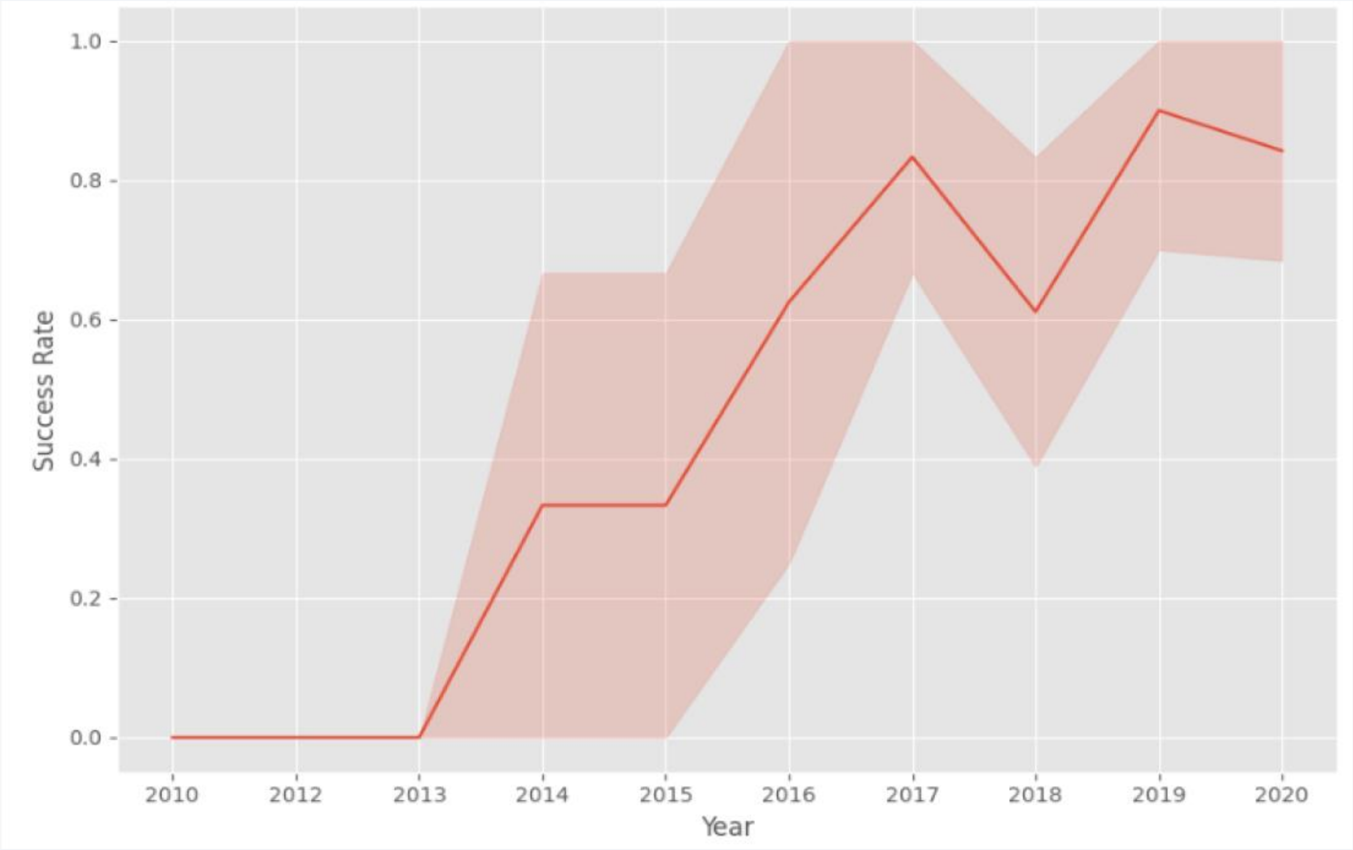
- Show a scatter point of Flight number vs. Orbit type
- Show the screenshot of the scatter plot with explanations



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

Launch Success Yearly Trend



From 2013 onward, the success rate consistently increased until the year 2020.

All Launch Site Names

Task 1

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

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

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

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

Utilized the 'SELECT DISTINCT' statement to retrieve exclusively the distinct launch sites from the 'LAUNCH_SITE' column within the SPACEXTBL table.

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'

```
%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
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

Employed the 'LIKE' command with the '%' wildcard in the 'WHERE' clause to specifically choose and present a table comprising all records where launch sites commence with the string '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)

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

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

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

- Employed the 'SUM()' function to retrieve and showcase the total 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

- Utilized the 'AVG()' function to retrieve and present the average payload mass carried 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

- Employed the 'MIN()' function to obtain and showcase the earliest date, representing the first successful landing outcome on the ground pad labeled 'Success (ground pad)'.

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

- Utilized the 'SELECT DISTINCT' statement to retrieve and list the unique names of boosters with operators greater than 4000 and less than 6000, narrowing down the list to include only those boosters with 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

Employed the 'COUNT()' function in conjunction with the 'GROUP BY' statement to determine and display the total number 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

Used a subquery to retrieve and convey the maximum payload, then utilized it to list all the boosters that have transported the 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 displav 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)

Utilized the 'SUBSTR()' function in the SELECT statement to extract the month and year from the date column, considering records where 'substr(Date, 7, 4) = '2015' for the year, and the landing outcome was 'Failure (drone ship)'. Returned the records matching the specified filter.

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

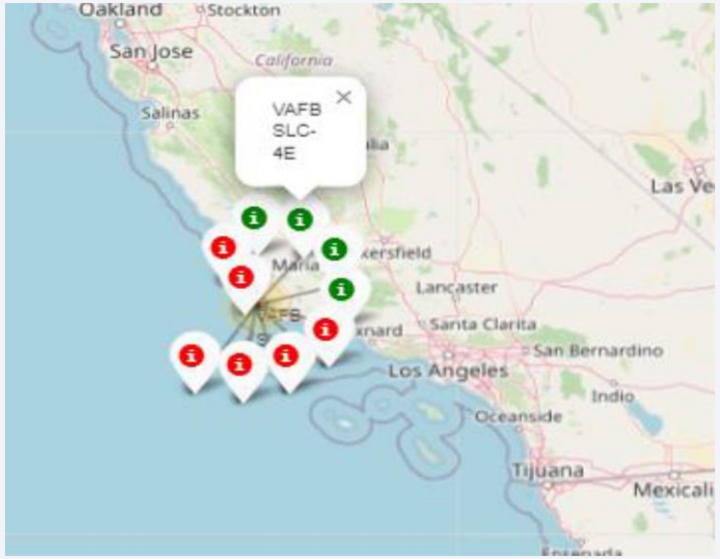
Launch Sites

- All launch sites are situated near the Equator, located to the south on the US map. Additionally, all the launch sites are in very close proximity to the coast.

Launch Outcomes in Florida

On the eastern coast, specifically in Florida, Launch site KSC LC-39A exhibits relatively higher success rates compared to CCAFS SLC-40 and CCAFS LC-40.

Launch Outcomes in California

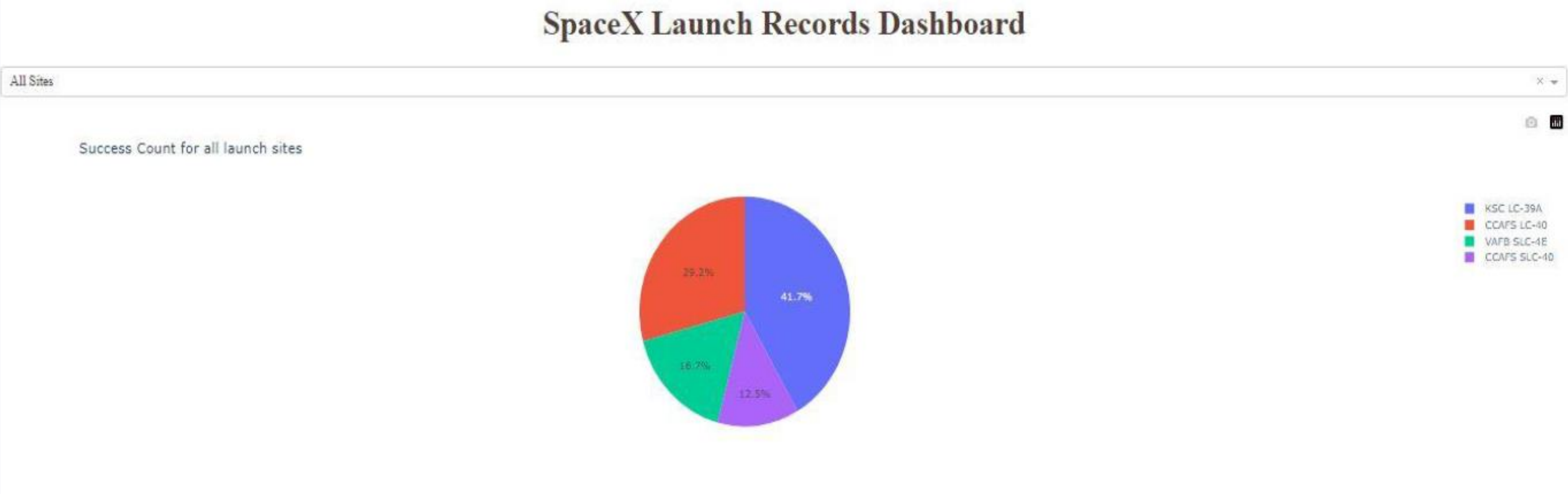


On the West Coast (California), Launch site VAFB SLC-4E has a relatively lower success rate, achieving 4 successes out of 10 launches, compared to the launch site KSC LC-39A on the Eastern Coast of Florida.

Section 4

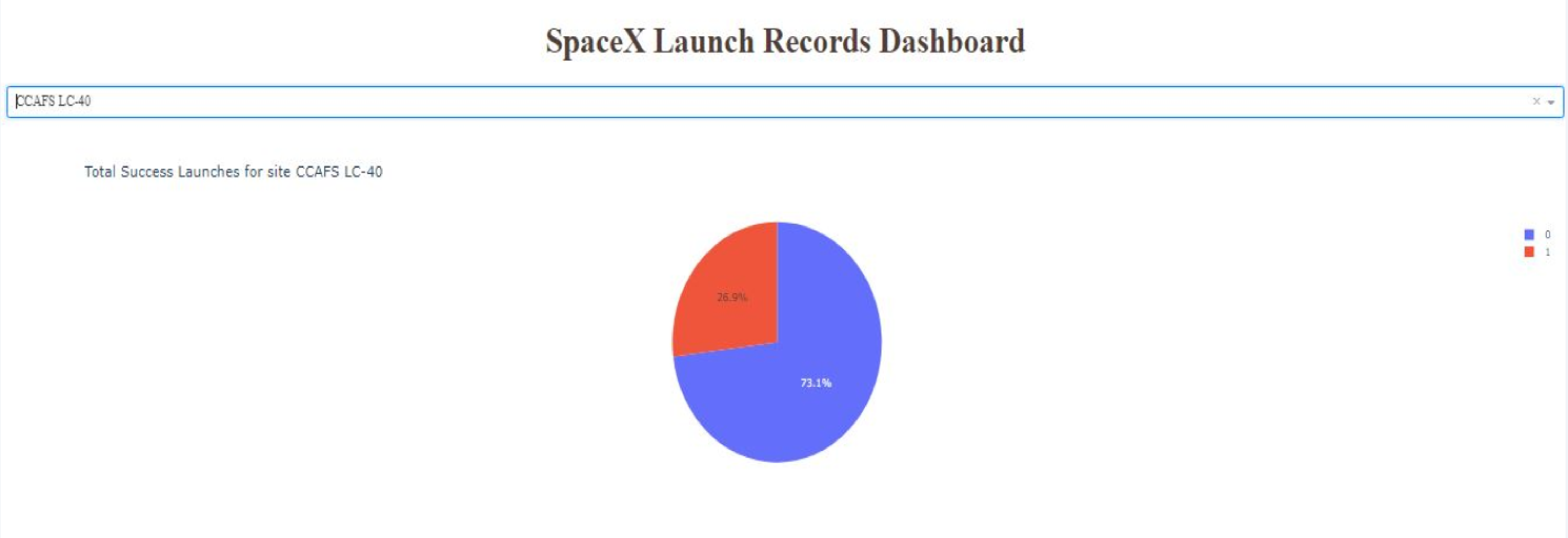
Build a Dashboard with Plotly Dash

Pie chart 1



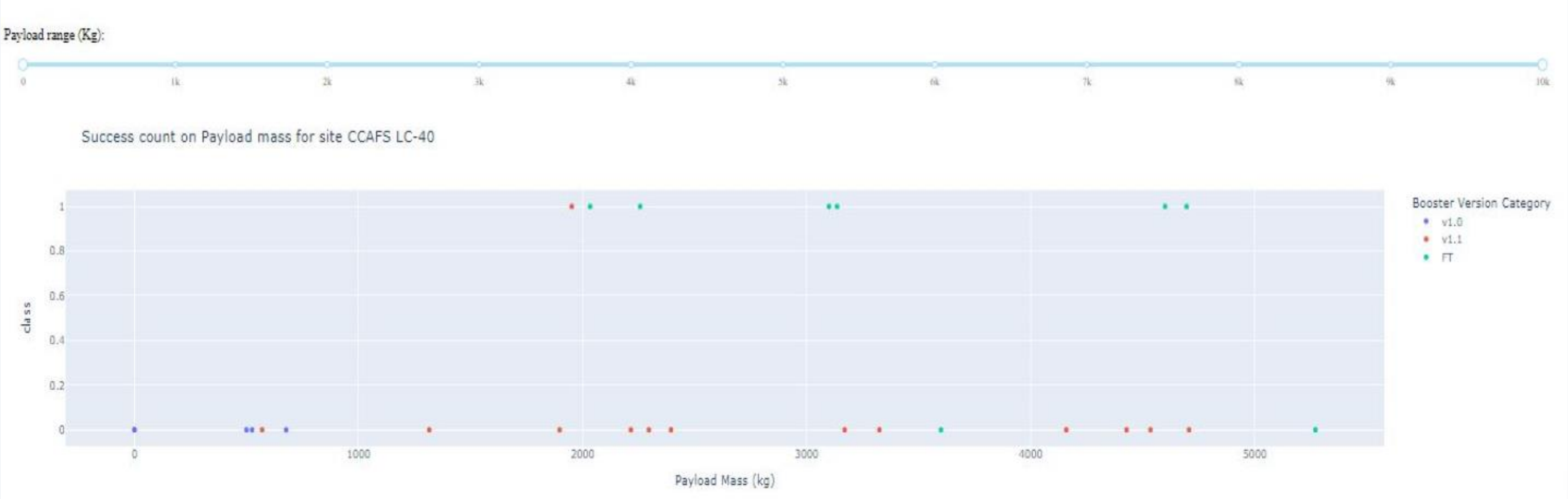
Launch site KSC LC-39A boasts the highest launch success rate at 42%, followed by CCAFS LC-40 with a rate of 29%. VAFB SLC-4E comes next with a success rate of 17%, and lastly, launch site CCAFS SLC-40 has a success rate of 13%.

Pie Chart 2



Launch site CCAFS LC-40 achieved the second-highest success ratio with a 73% success rate compared to a 27% rate of failed launches.

Scatter Plot



For Launch site CCAFS LC-40, the booster version FT exhibits the highest success rate for payloads exceeding 2000 kg.

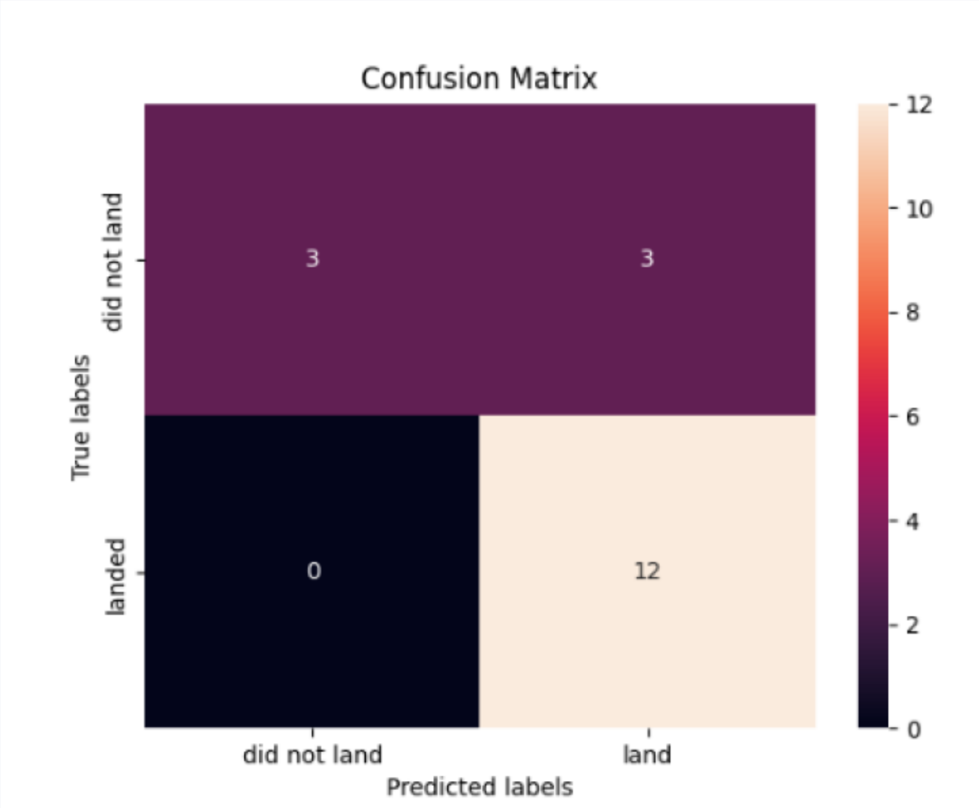
Section 5

Predictive Analysis (Classification)

Classification Accuracy

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

Confusion Matrix



All four classification models displayed identical confusion matrices and demonstrated equal proficiency in distinguishing between various classes. The primary challenge encountered across all models is the occurrence of false positives.

Conclusions

- The overall trend showed an increase in success rates from 2013 to 2020.
- The dashboard provided an interactive interface for exploring space mission data.
- Machine learning models were evaluated with a focus on accuracy, and the models performed similarly, with false positives being a notable challenge.
- Launch site success rates varied, with some sites showing higher success rates.
- Booster and payload analyses provided insights into performance metrics and specific characteristics associated with success.

Thank you!