IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Sebastián Andrade>
<September 8th, 2024>

Outline

EXECUTIVE SUMMARY

INTRODUCTION

METHODOLOGY

RESULTS

CONCLUSION

APPENDIX

Executive Summary

General Summary

- In this project we will predict if the SpaceX Falcon 9 rocket would land successfully, using machine learning algorithms.

Summary of methodologies

- **Data Collection through API** and with Web Scraping
- **Data Wrangling**
- **Exploratory Data Analysis with SQL** and Visualization
- Interactive **Visual Analytics with Folium** and Dash
- **Machine Learning Predictions**

Summary of all results

- **Exploratory Data Results**: The results show that some of the features of the rocket launches, have a correlation with the outcome of the launching.
- **Machine Learning predictions**: The results also show that the best algorithm to predict the results is the **DecisionTree.**

Introduction

Project background and context	SpaceX advertises the Flacon 9 rocket launches on its platform and its costs. One of the things that allows them to save the costs, is the ability of reusing the first stage of the launching. Therefore, if we can predict the outcome of the first stage, we can use that information to help a SpaceY to go against SpaceX on the rocket race.
Problems you want to find answers	Identifying the factors that influence the landing outcome.
	The relationship between each variable studied and how they affect the outcome of the launching.
	The best conditions needed to increase the probability of a successful landing.

Section 1

Methodology

Methodology

Data collection methodology:

The data was collected using SpaceX API and web scraping from Wikipedia.

Perform data wrangling

The data was filtered by dropping irrelevant columns, and One Hot Encoding was applied to categorical values.

Perform exploratory data analysis (EDA) using visualization and SQL

Perform interactive visual analytics using Folium and Plotly Dash

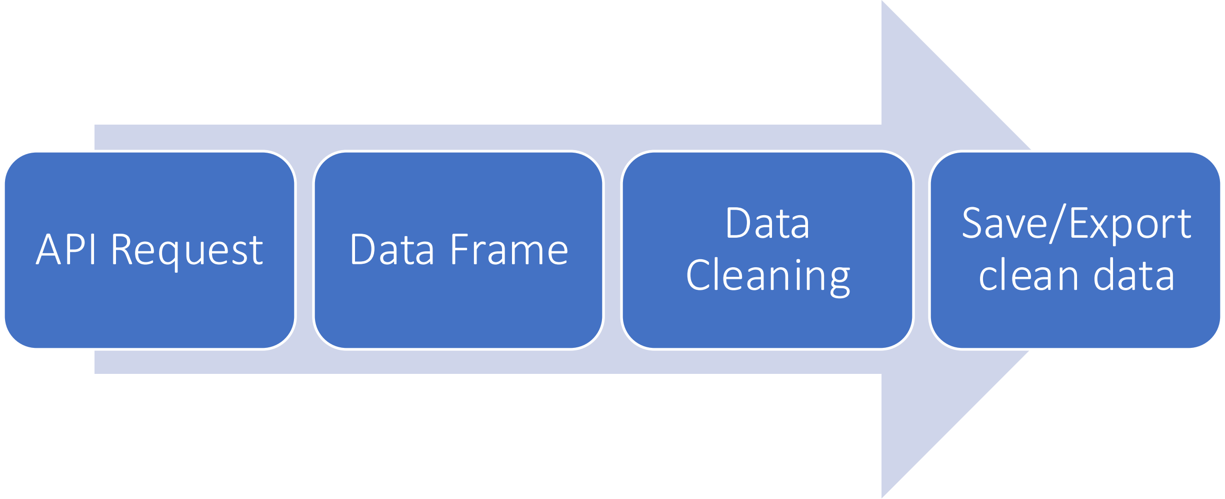
Perform predictive analysis using classification models

Using classification models, tuning them and finding the best parameters.

Data Collection

The process of collecting the SpaceX data starts by sending a request to de API using the request library.
After that a JSON file is received that can be converted to a DataFrame using Pandas

The data is then cleaned by dropping irrelevant columns to keep only the essential and valid data for analysis.



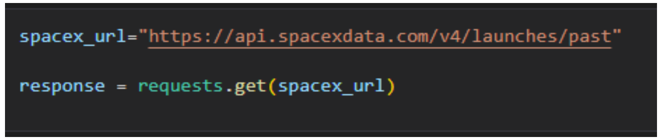
API Request

Data Frame

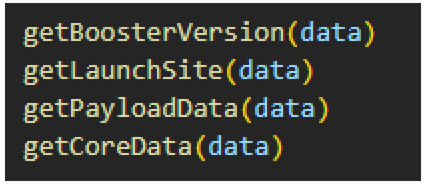
Data Cleaning

Save/Export clean data

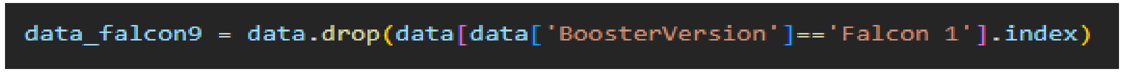
Data Collection – SpaceX API



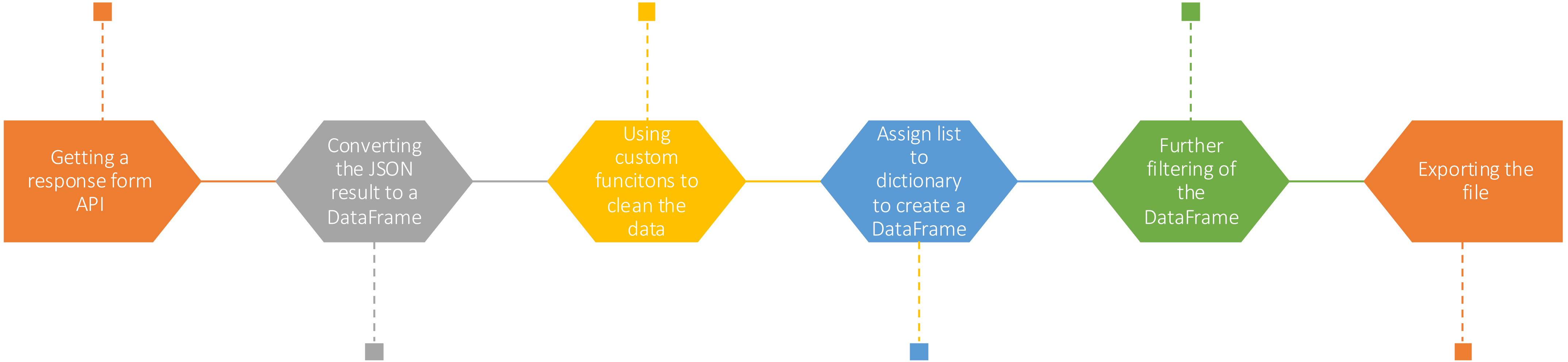
```
spacex_url="https://api.spacexdata.com/v4/launches/past"

response = requests.get(spacex_url)
```



```
getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```



```
data_falcon9 = data.drop(data[data['BoosterVersion']=='Falcon 1'].index)
```



Getting a response form API

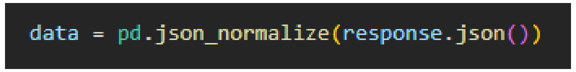
Converting the JSON result to a DataFrame

Using custom funcitons to clean the data

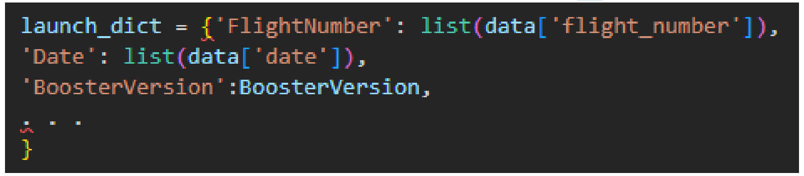
Assign list to dictionary to create a DataFrame

Further filtering of the DataFrame

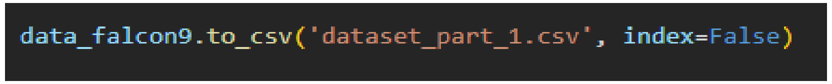
Exporting the file



```
data = pd.json_normalize(response.json())
```



```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
. . .
}
```

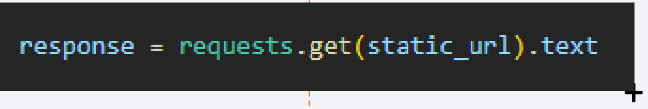


```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

GitHub link

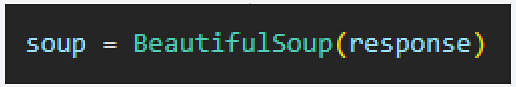
Data Collection – Web Scraping

Request a response from HTML



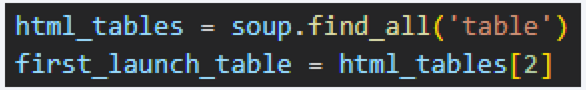
```
response = requests.get(static_url).text
```

Create a BeautifulSoup object



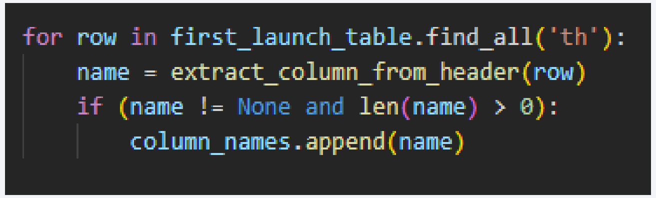
```
soup = BeautifulSoup(response)
```

Finding the relevant tables



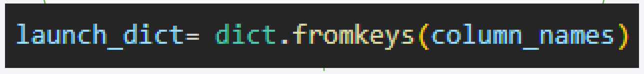
```
html_tables = soup.find_all('table')
first_launch_table = html_tables[2]
```

Getting the names of the columns



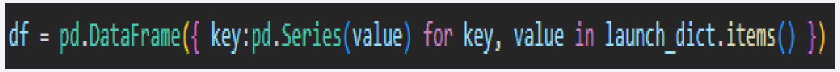
```
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if (name != None and len(name) > 0):
        column_names.append(name)
```

Create a dictionary and appending the data to keys



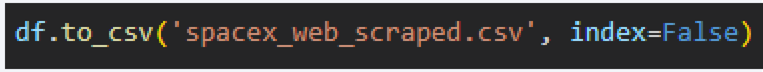
```
launch_dict= dict.fromkeys(column_names)
```

Creating a DataFrame from the dictionary



```
df = pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

Exporting as a .CVS file



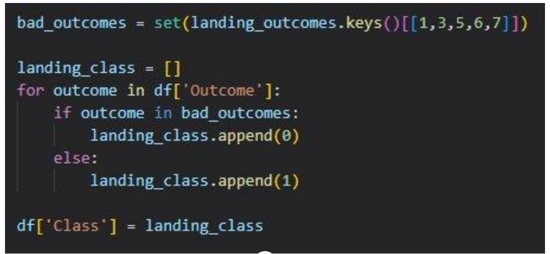
```
df.to_csv('spacex_web_scraped.csv', index=False)
```

GitHub Link

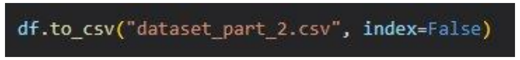
Data Wrangling

This is a the process of cleaning and sorting the data for easy access and analysis. Here the data of the landing outcome of the rockets is studied to find relevant information.

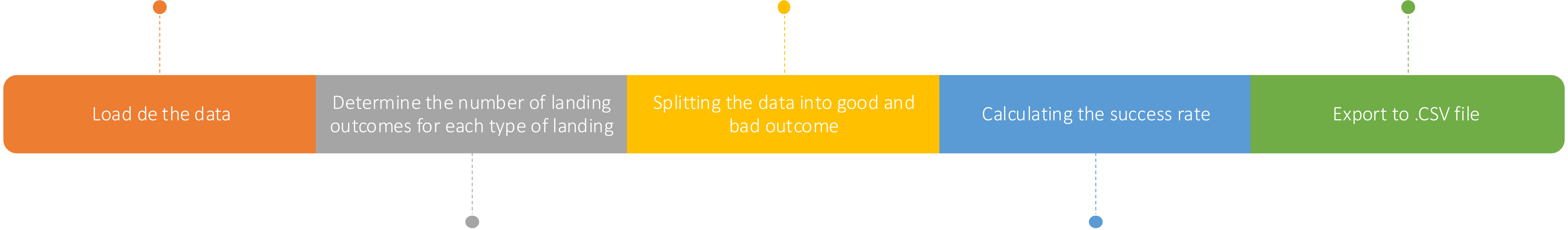
Creating a DataFrame out of the data



```
bad_outcomes = set(landing_outcomes.keys()[[1,3,5,6,7]])

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

df['Class'] = landing_class
```



```
df.to_csv("dataset_part_2.csv", index=False)
```



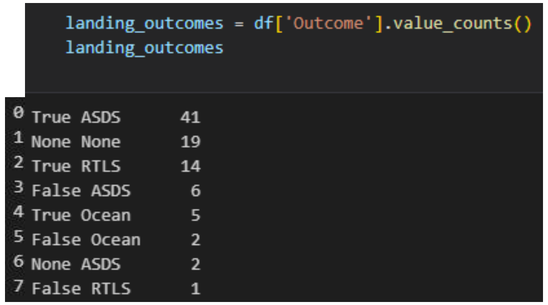
Load de the data

Determine the number of landing outcomes for each type of landing

Splitting the data into good and bad outcome

Calculating the success rate

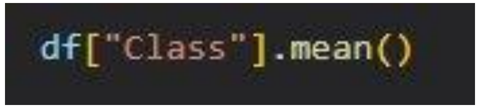
Export to .CSV file



```
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

```
0 True ASDS      41
1 None None      19
2 True RTLS      14
3 False ASDS      6
4 True Ocean      5
5 False Ocean     2
6 None ASDS       2
7 False RTLS      1
```

60%



```
df["Class"].mean()
```

GitHub URL

EDA with Data Visualization

GitHub URL

Different kind of charts were created to visualize a variety of relationships between features of the Rocket Launch.

Flight Number vs Launch Site

- A scatter plot for the number of the flight out of 90 flights, and the launch sites. Also showing which ones succeeded and failed.
- Launch sites:
  - CCAFS SLC 40
  - VAFB SLC 4E
  - KSC LC 39A

Payload vs Launch Site

- A scatter plot to observe if there's any relationship between the launch sites and the payload mass of the rocket. Here it's shown which ones succeeded and failed.

Orbit vs Success Rate

- A bar chart to visualize the relationship between the success rate of each orbit type.

Flight Number vs Orbit

- A scatter plot to visualize the relationship between the Flight Number and the type of orbit, showing which ones have succeeded and failed.

Payload vs Orbit

- A scatter plot to visualize the relationship between the Mass of the Payload and the type of orbit, showing which ones have succeded

Date vs Success Rate

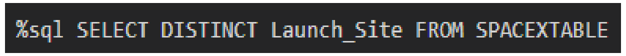
- A line chart to visualize the launch success yearly trend.

EDA with SQL

The data is connected to the SQL extension to execute queries to further understand the data.

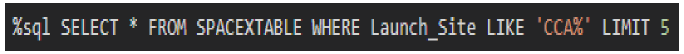
GitHub URL

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



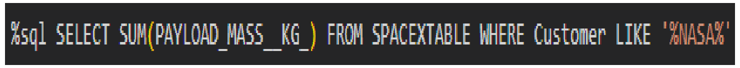
```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE
```

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



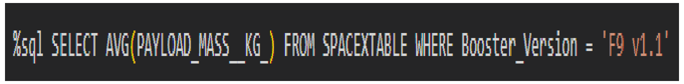
```
%sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5
```

Displaying the total payload mass carried by boosters launched by NASA



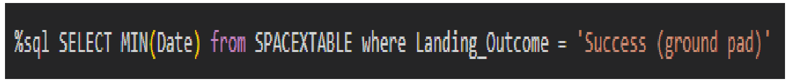
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer LIKE '%NASA%'
```

Displaying the average payload mass carried by booster version F9 V1.1



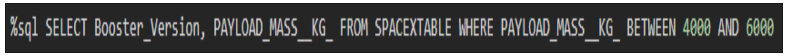
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1'
```

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



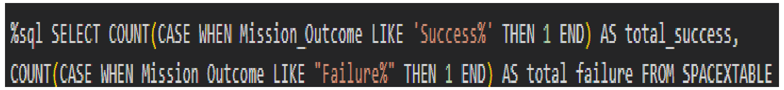
```
%sql SELECT MIN(Date) from SPACEXTABLE where Landing_Outcome = 'Success (ground pad)'
```

Listing names of the boosters which have success in drone ship and have payload mass between 4000 and 6000



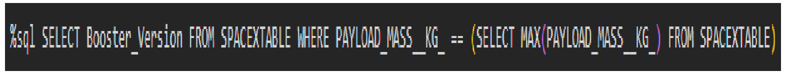
```
%sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

Listing the total number of successful and failed mission outcomes



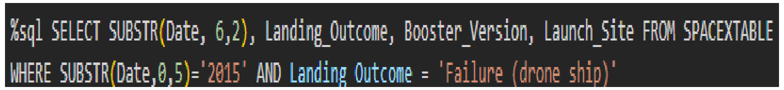
```
%sql SELECT COUNT(CASE WHEN Mission_Outcome LIKE 'Success%' THEN 1 END) AS total_success,
COUNT(CASE WHEN Mission_Outcome LIKE "Failure%" THEN 1 END) AS total_failure FROM SPACEXTABLE
```

Listing the names of booster versions which have carried the maximum payload mass



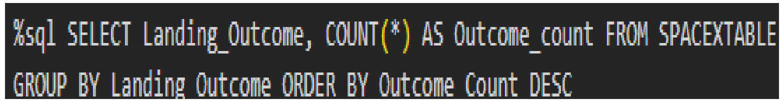
```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ == (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

Listing the records, failed landing outcomes in drone ships, booster versions, and launch site for the months in 2015



```
%sql SELECT SUBSTR(Date, 6,2), Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE
WHERE SUBSTR(Date,0,5)='2015' AND Landing_Outcome = 'Failure (drone ship)'
```

Ranking the count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order



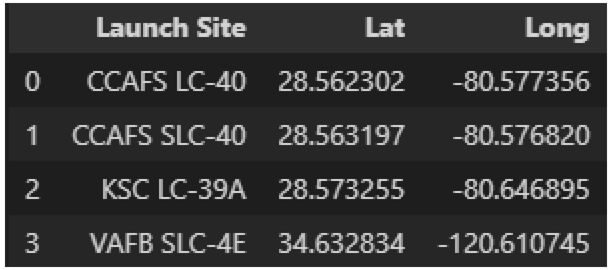
```
%sql SELECT Landing_Outcome, COUNT(*) AS Outcome_count FROM SPACEXTABLE
GROUP BY Landing_Outcome ORDER BY Outcome_Count DESC
```

Interactive Map with Folium

GitHub URL

- The launch success rate may also depend on the location and proximities of a launch site, i.e., the initial position of rocket trajectories. Finding an optimal location for building a launch site certainly involves many factors and it's possible to discover some of the factors by analyzing the existing launch site locations.

- Using a folium Map object centered on NASA Johnson Space Center at Houston, Texas, it's possible to add **circle markers** to mark the launch sites:



	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610745

- **MarkerClusters** are used to identify the successes and failures for each launch locating them on the map and also **PolyLines** to create lines between the launch sites and certain locations to later measure the distances.

Build a Dashboard with Plotly Dash

[GitHub URL]

This dashboard application contains input components such as a dropdown of all the launch sites, and **a range slider** for the payload mass to interact **with a pie chart and a scatter point chart.** The steps to build **this dashboard application** are:

Adding **a Launch Site Drop-down Input Component**

- This will allow to choose a launch site to analyze it's success rate, or all of them.

Adding **a callback function to** render success-pie-chart based **on selected site dropdown**

- This will show the pie chart associated to the launch site selected previously.

Adding **a Range Slider to Select Payload**

- This will display a slider to select the range of the Payload Mass on Kg that will be shown after.

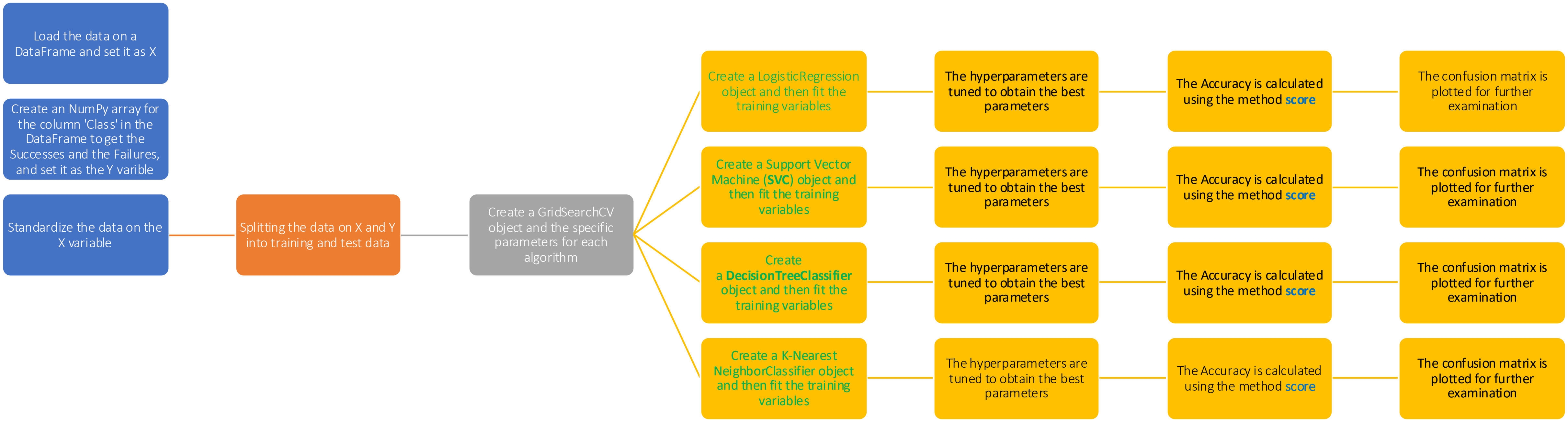
Adding **a callback function to render** the success-payload-scatter-chart scatter **plot**

- This will show on a scatter plot for the launch sites according to the range previously chosen. This also will show the Booster Versions of the rocket.

Predictive Analysis (Classification)

GitHub URL

A machine learning pipeline is created to predict if the first stage will land given the data from the preceding labs. It's performed an exploratory Data Analysis to determine Training Labels, and then we have to find the best Hyperparameter for SVM, Classification Trees and Logistic Regression. Then we find the method performs best using test data.



Load the data on a DataFrame and set it as X

Create an NumPy array for the column 'Class' in the DataFrame to get the Successes and the Failures, and set it as the Y varible

Standardize the data on the X variable

Splitting the data on X and Y into training and test data

Create a GridSearchCV object and the specific parameters for each algorithm

Create a LogisticRegression object and then fit the training variables

The hyperparameters are tuned to obtain the best parameters

The Accuracy is calculated using the method **score**

The confusion matrix is plotted for further examination

Create a Support Vector Machine (**SVC**) object and then fit the training variables

The hyperparameters are tuned to obtain the best parameters

The Accuracy is calculated using the method **score**

The confusion matrix is plotted for further examination

Create a **DecisionTreeClassifier** object and then fit the training variables

The hyperparameters are tuned to obtain the best parameters

The Accuracy is calculated using the method **score**

The confusion matrix is plotted for further examination

Create a K-Nearest NeighborClassifier object and then fit the training variables

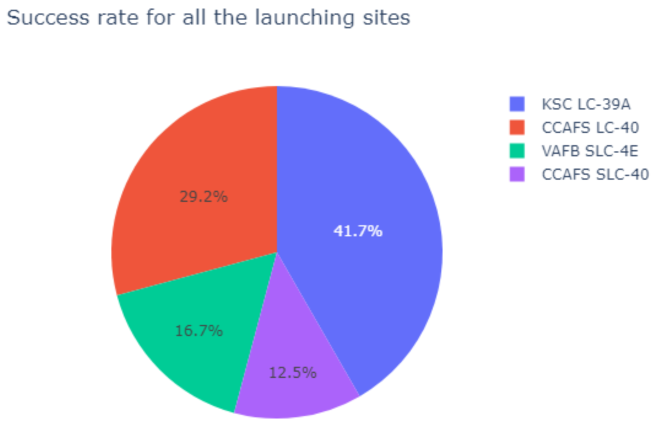
The hyperparameters are tuned to obtain the best parameters

The Accuracy is calculated using the method score

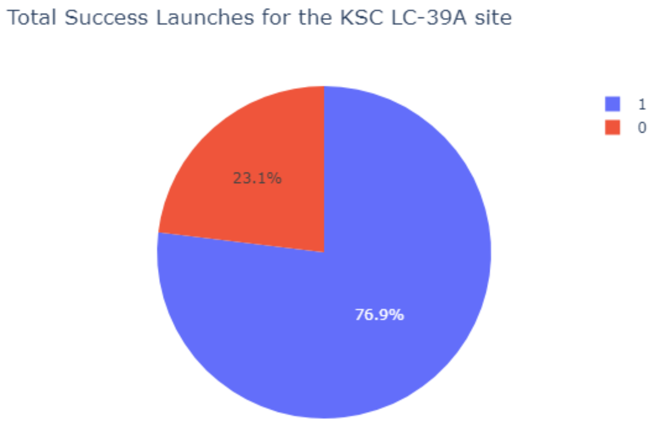
The confusion matrix is plotted for further examination

Results

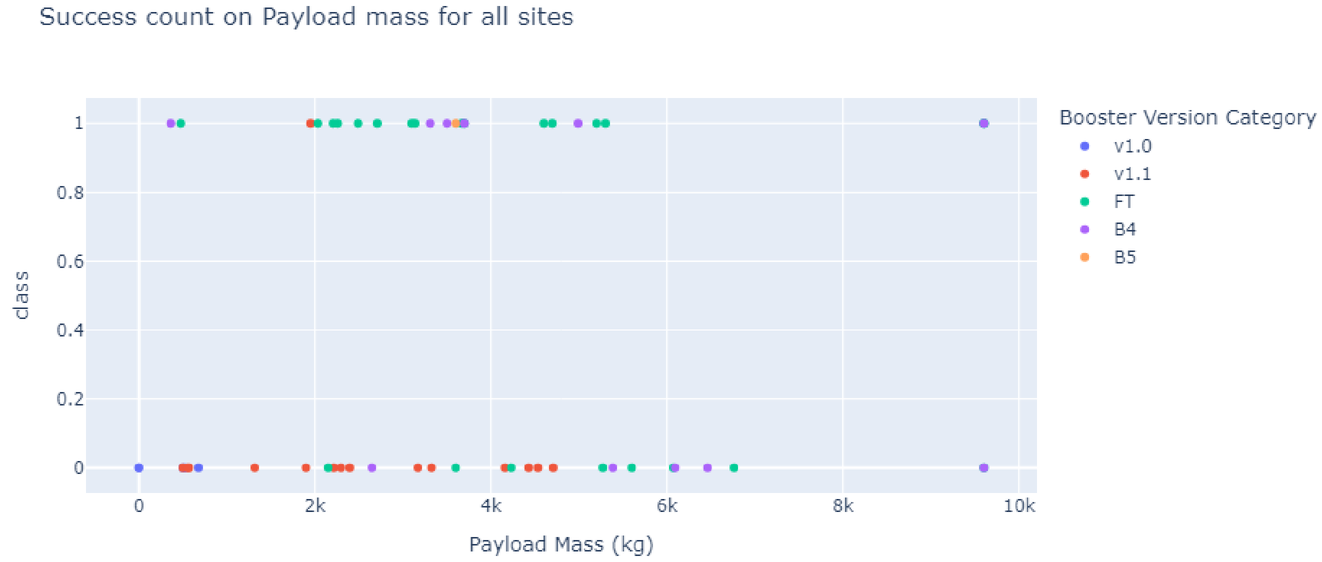
- Out of the exploratory data analysis we can gather several results:
    - From the relationship between Flight Number and Launch Site, we can see that as the number of flights increases, the flight success rate improves. Additionally, **better results are observed at the 'CCAFS SLC 40' Launch Site.**
    - From the relationship between Payload Mass and Launch Site, we can see that at the 'VAFB-SLC' Launch Site, there have been no launches with a payload mass greater than 10,000kg. Moreover, we can see that **the highest success is at the 'KSC LC 39A' Launch Site**.
    - From the relationship between Success Rate and Orbit Type, we find that **the highest success rates are in orbits of type 'ES-L1', 'GEO', 'HEO', and 'SSO'**.
    - From the relationship between Flight Number and Orbit Type, it appears that success is correlated with the Flight Number, and that **the 'VLEO' orbit has the highest number of successful flights by flight number.**
    - From the relationship between Payload Mass and Orbit, **the success rate seems to be concentrated in the area where the payload mass is less than 6,000kg.**
    - From the yearly trend of the success rate, we can see that **since 2013, the success rate has been increasing.**
- We can see from the Interactive analytics that the **KSC-LC-39A** launch site has a bigger success rate compared to the other sites. Also, we can see that the success rate is higher for a Payload Mass between 2000kg and 6000kg, using the Booster Version **FT.**
- According to the predictive analysis results, all methods performed practically the same, except for the **DecisionTree**, which fit train data slightly better but test data slightly worse. With that in mind, considering the amount of true positives and true negatives, the **DecisionTree** gave better results.



Success rate for all the launching sites



Total Success Launches for the KSC LC-39A site



Success count on Payload mass for all sites

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

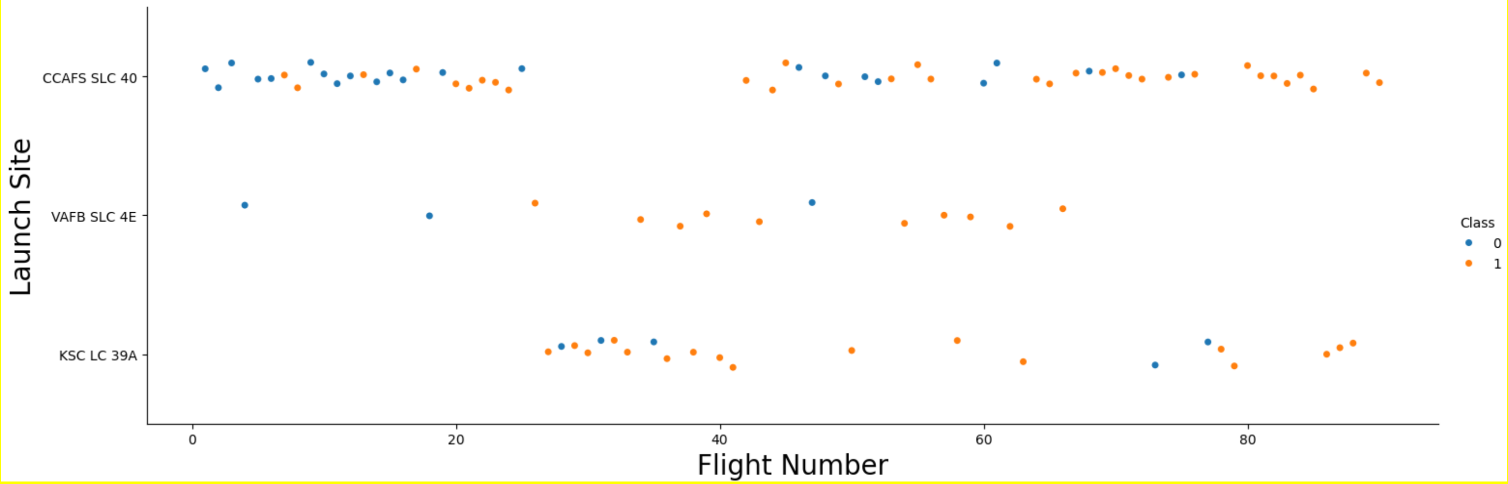
FLIGHT NUMBER

- The number of the flight out of 90 flights

Launch Site

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

Class:
0: Failed
1: Successful



- We can see that as the number of flights increases, the flight success rate improves. Additionally, **better results are observed at the 'CCAFS SLC 40' Launch Site.**

Payload vs. Launch Site

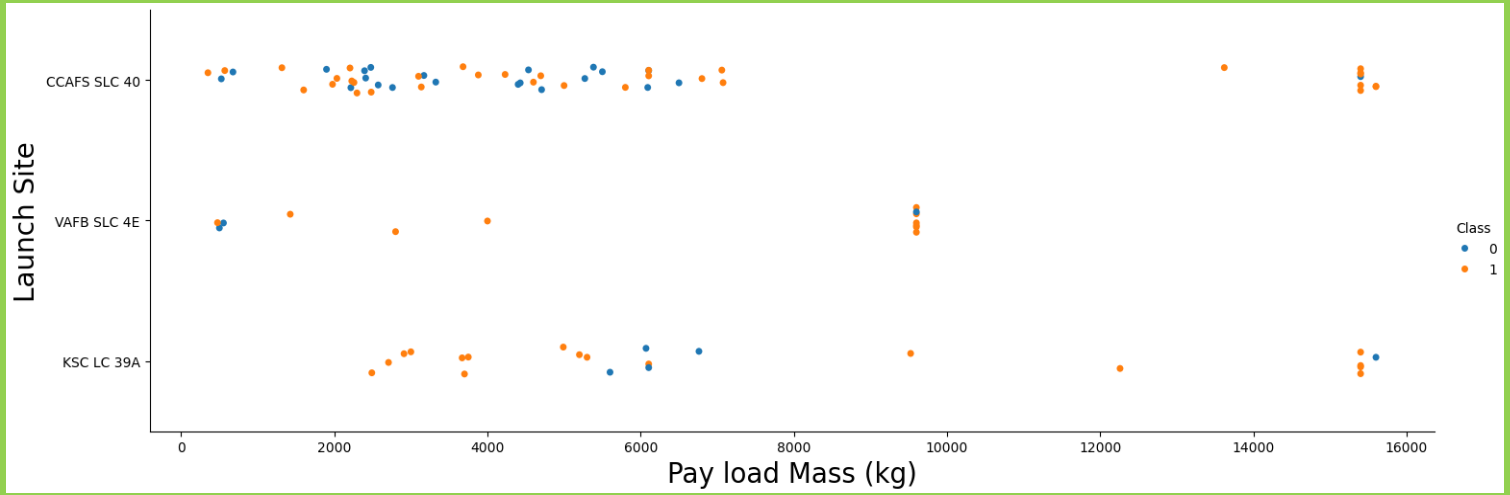
Payload (kg)

- The amount of mass of the payload on the rocket

Launch Site

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

Class:
0: Failed
1: Successful

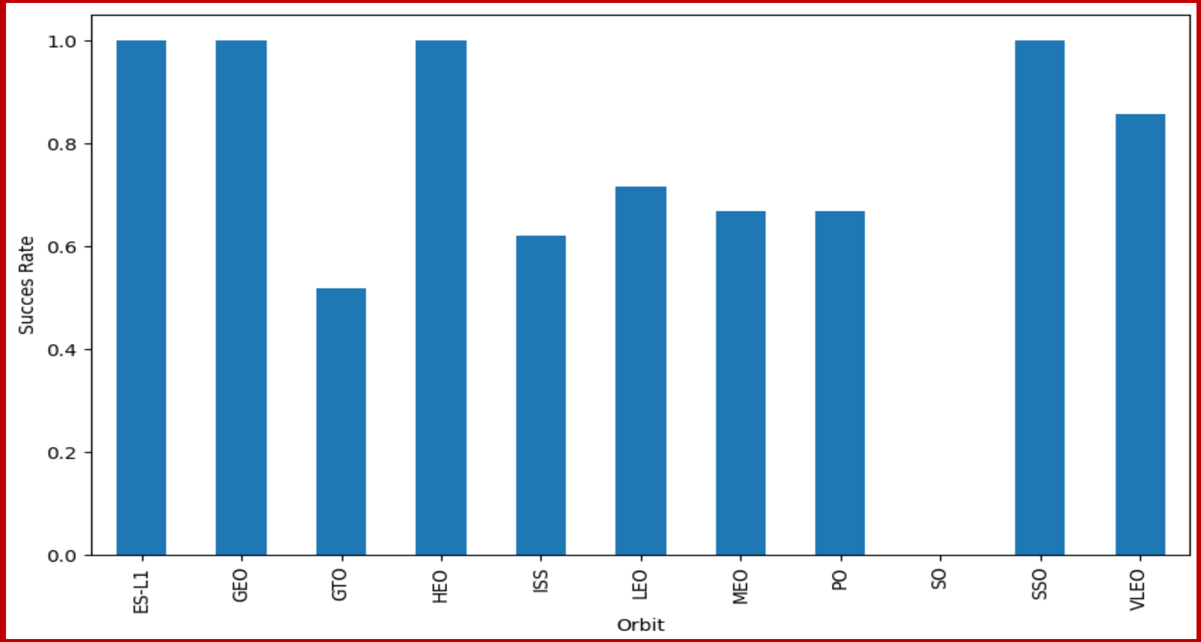


- We can see that at the 'VAFB-SLC' Launch Site, there have been no launches with a payload mass greater than 10,000kg. Moreover, we can see that **the highest success is at the 'KSC LC 39A' Launch Site**.

Success Rate vs. Orbit Type

Orbit
- The type of orbit on which the rocket is launched

Success Rate
- How succesful out of the total amount of launches



- We find that **the highest success rates are in orbits of type 'ES-L1', 'GEO', 'HEO', and 'SSO'.**

Flight Number vs. Orbit Type

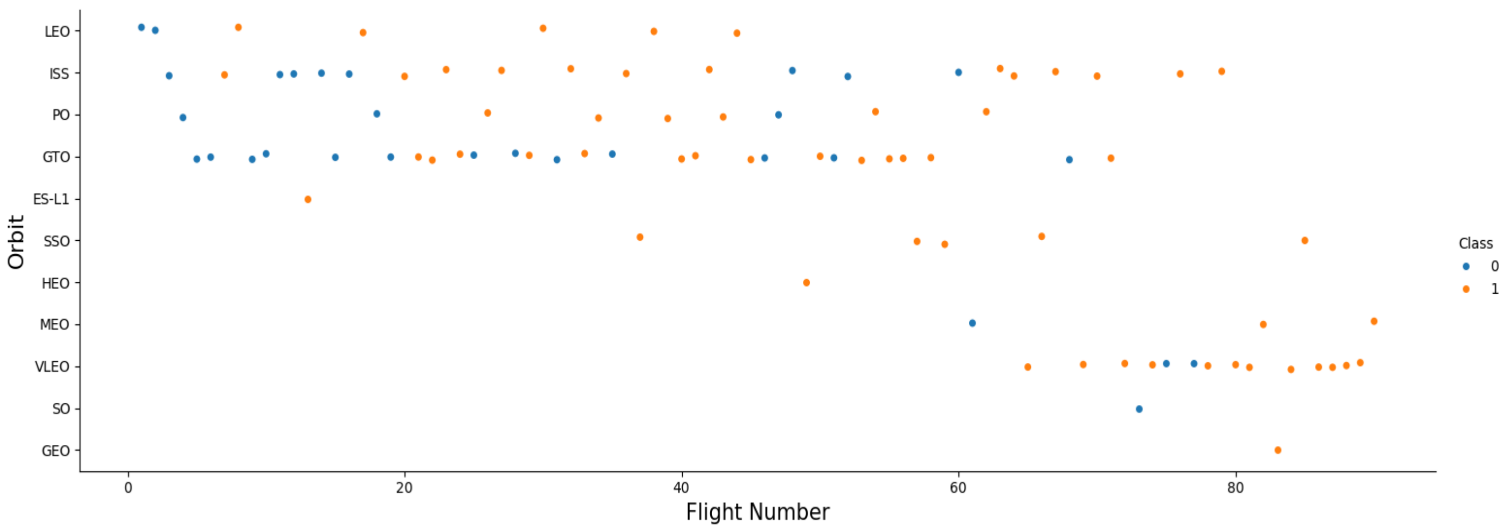
FLIGHT NUMBER

- The number of the flight out of 90 flights

Orbit

- The type of orbit on which the rocket is launched

Class:
0: Failed
1: Successful



- It appears that success is correlated with the Flight Number, and that **the 'VLEO' orbit has the highest number of successful flights by flight number.**

Payload vs. Orbit Type

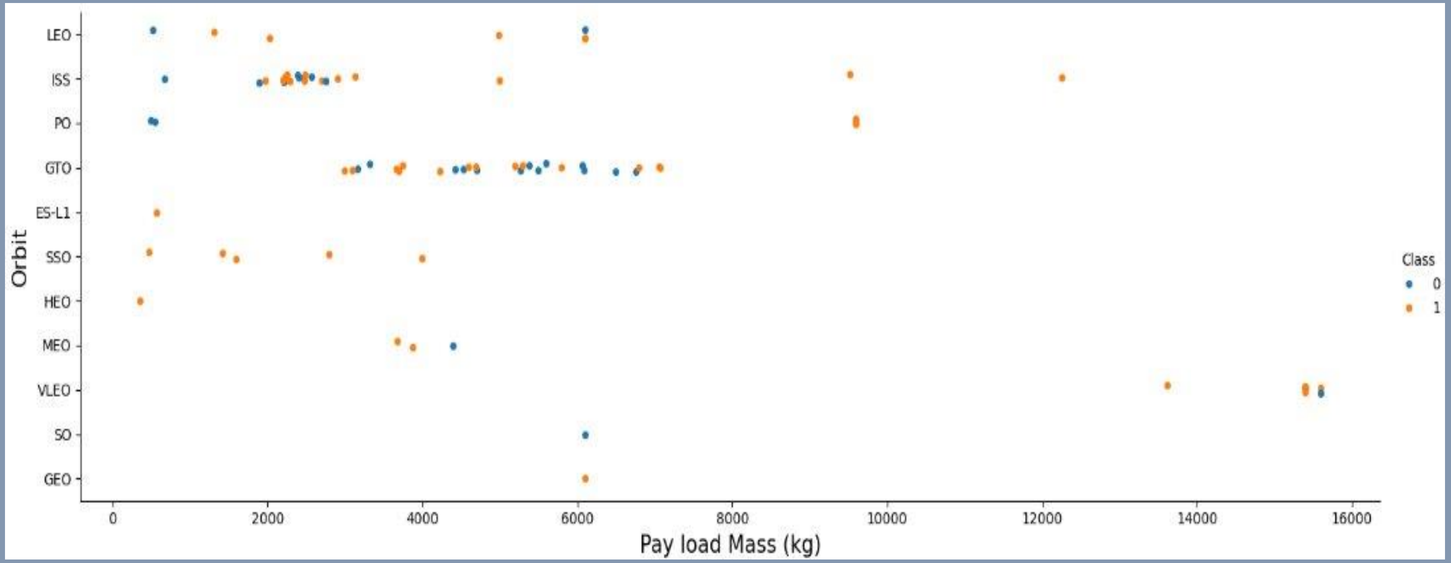
Payload (kg)

- The amount of mass of the payload on the rocket

Orbit

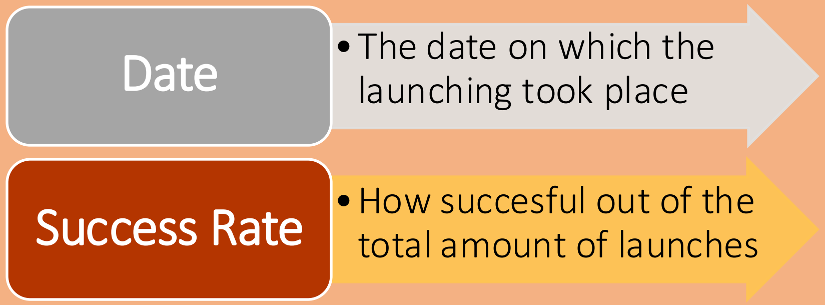
- The type of orbit on which the rocket is launched

Class:
0: Failed
1: Successful



- The success rate seems to be concentrated in the area where the payload mass is less than 6,000kg.

Launch Success Yearly Trend

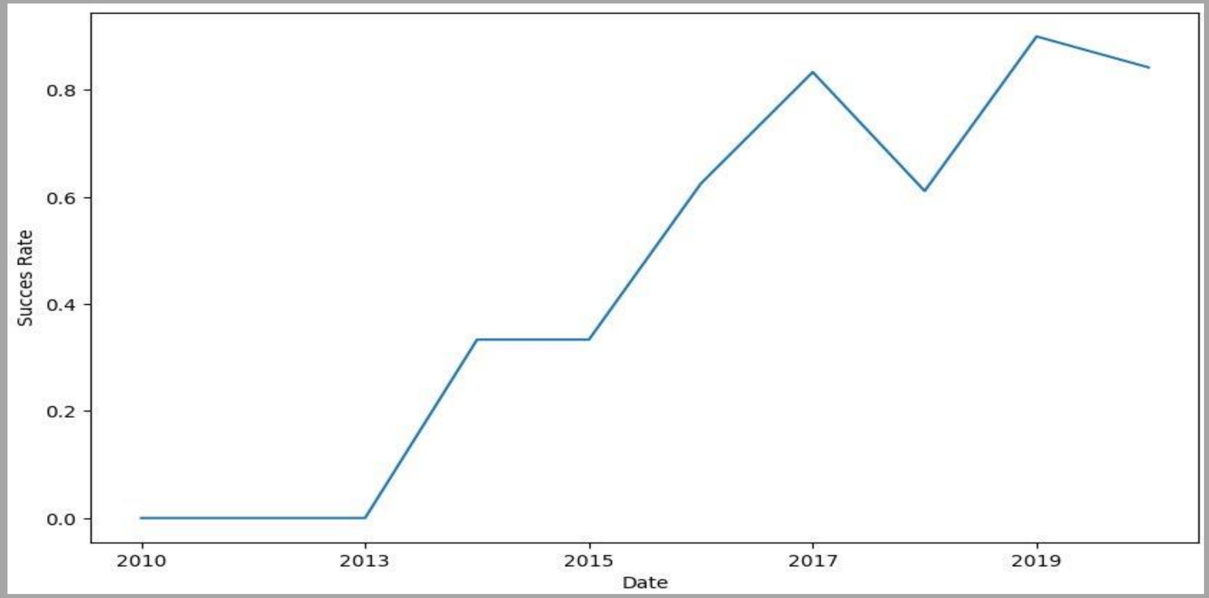


Date

- The date on which the launching took place

Success Rate

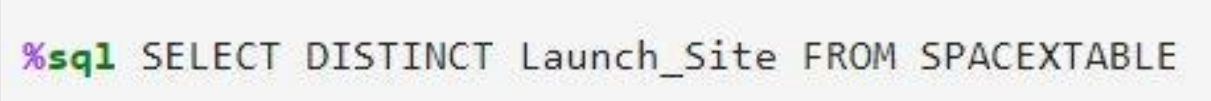
- How succesful out of the total amount of launches



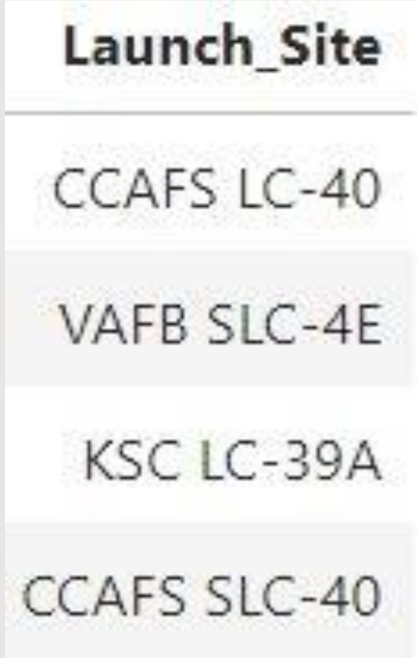
- We can see that **since 2013, the success rate has been increasing.**

All Launch Site Names

- We write a SQL query that displays the unique launch sites in the space mission, from the SPACEXTABLE table.



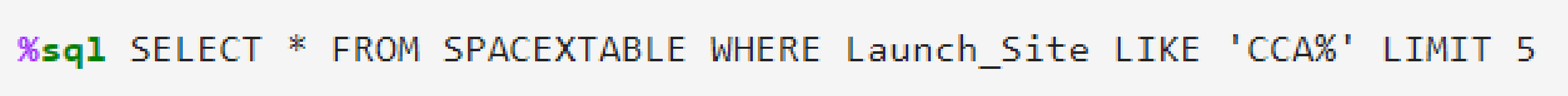
```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE
```



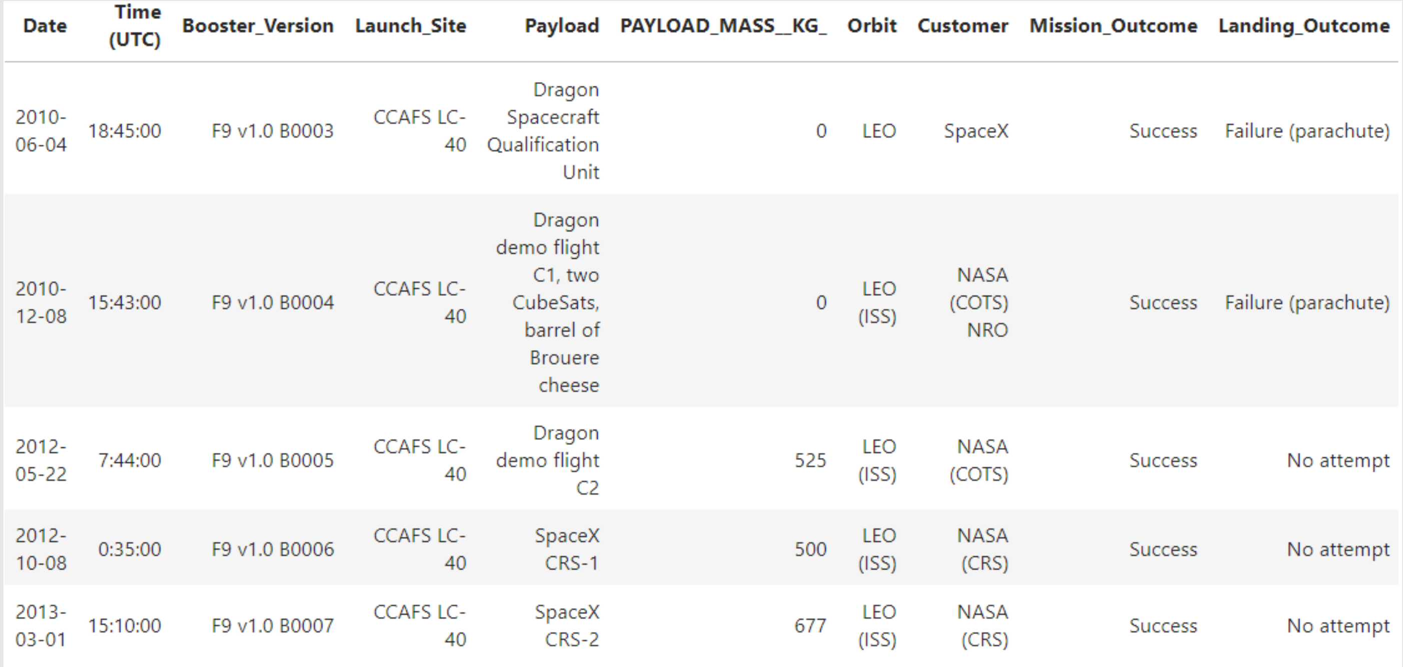
Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Launch Site Names Begin with 'CCA'

- A query is written to find 5 records where launch sites begin with 'CCA', from the SPACEXTABLE table.



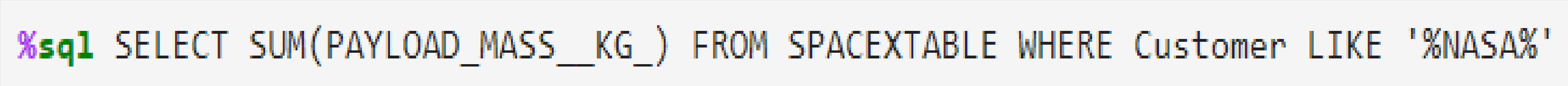
```
%sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5
```



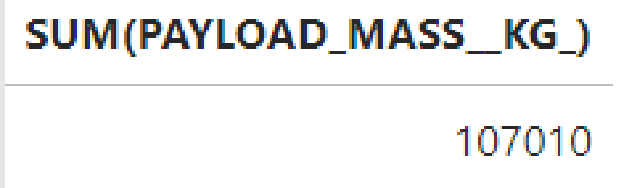
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- A query is written to calculate the total payload carried by boosters launched by NASA.



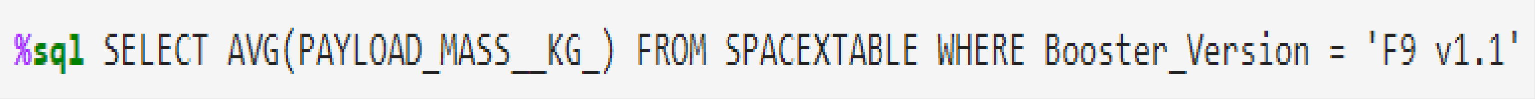
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer LIKE '%NASA%'
```



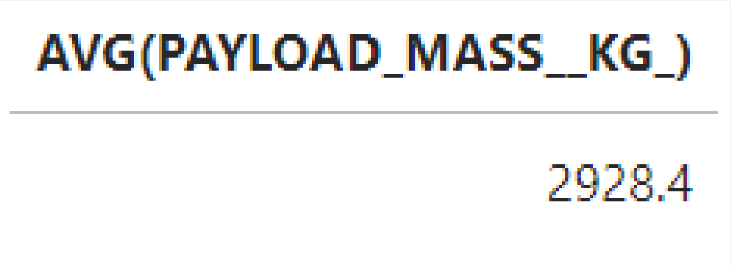
SUM(PAYLOAD_MASS__KG_)
107010

Average Payload Mass by F9 v1.1

- A query is written to calculate the average payload mass carried by booster version F9 v1.1.



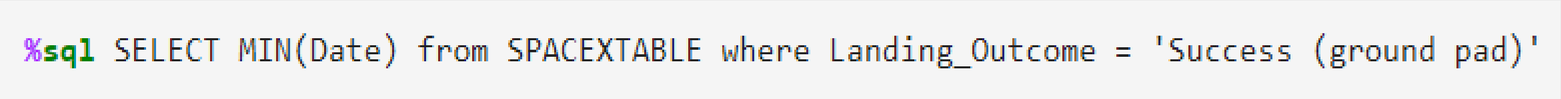
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version = 'F9 v1.1'
```



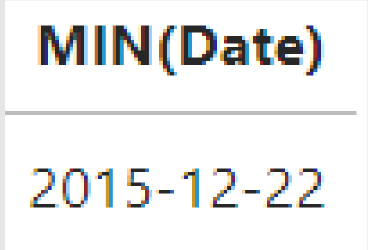
AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- A query is written to find the date when the first successful landing outcome on ground pad was achieved.



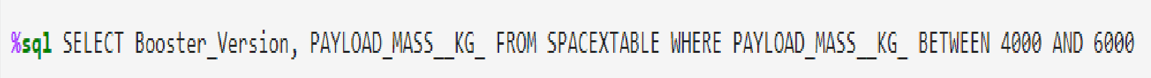
```
%sql SELECT MIN(Date) from SPACEXTABLE where Landing_Outcome = 'Success (ground pad)'
```



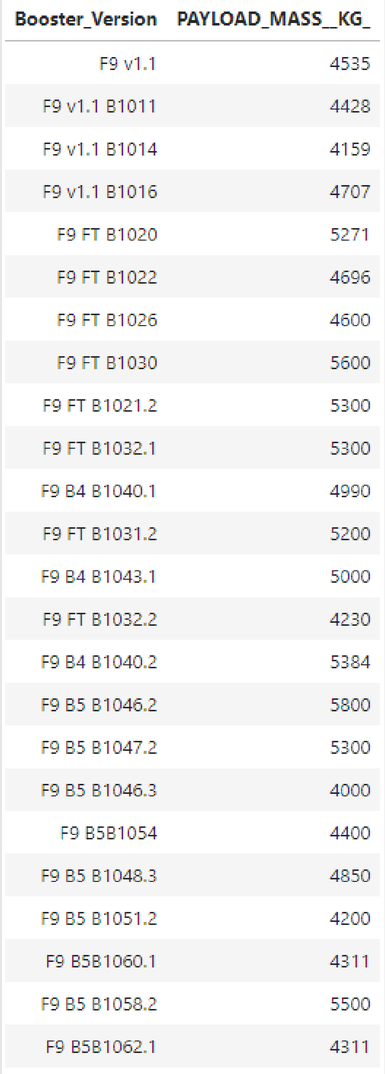
MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- A query is written to list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000.



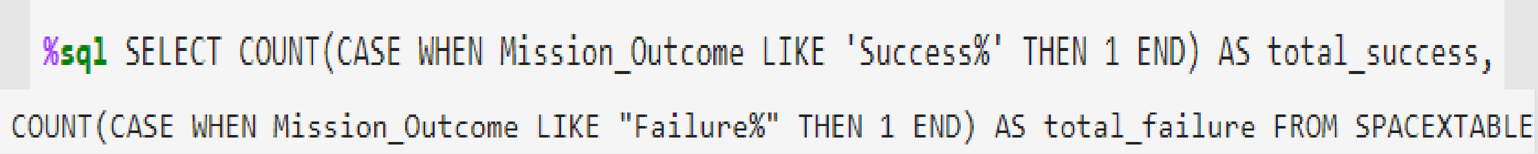
```
%sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```



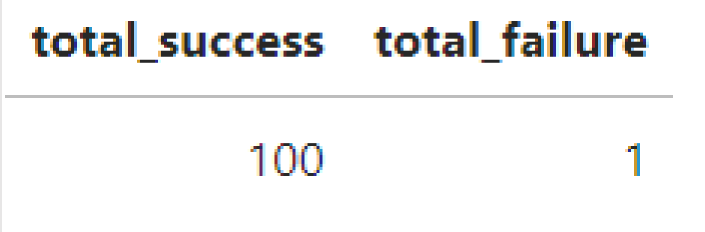
Booster_Version	PAYLOAD_MASS__KG_
F9 v1.1	4535
F9 v1.1 B1011	4428
F9 v1.1 B1014	4159
F9 v1.1 B1016	4707
F9 FT B1020	5271
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1030	5600
F9 FT B1021.2	5300
F9 FT B1032.1	5300
F9 B4 B1040.1	4990
F9 FT B1031.2	5200
F9 B4 B1043.1	5000
F9 FT B1032.2	4230
F9 B4 B1040.2	5384
F9 B5 B1046.2	5800
F9 B5 B1047.2	5300
F9 B5 B1046.3	4000
F9 B5B1054	4400
F9 B5 B1048.3	4850
F9 B5 B1051.2	4200
F9 B5B1060.1	4311
F9 B5 B1058.2	5500
F9 B5B1062.1	4311

Total Number of Successful and Failure Mission Outcomes

- A query is written to calculate the total number of successful and failure mission outcomes.
- For this query, the 'CASE' command is used to get the total amount of successes and failures



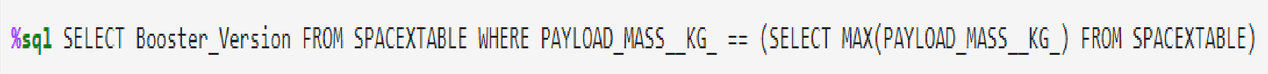
```
%sql SELECT COUNT(CASE WHEN Mission_Outcome LIKE 'Success%' THEN 1 END) AS total_success,
COUNT(CASE WHEN Mission_Outcome LIKE "Failure%" THEN 1 END) AS total_failure FROM SPACEXTABLE
```



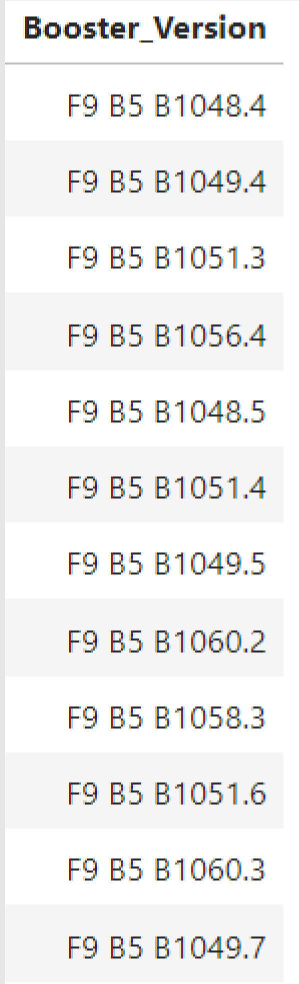
total_success	total_failure
100	1

Boosters Carried Maximum Payload

- A query is written to list the names of the booster which have carried the maximum payload mass.
- For this query, we use a subquery to get the maximum payload mass.



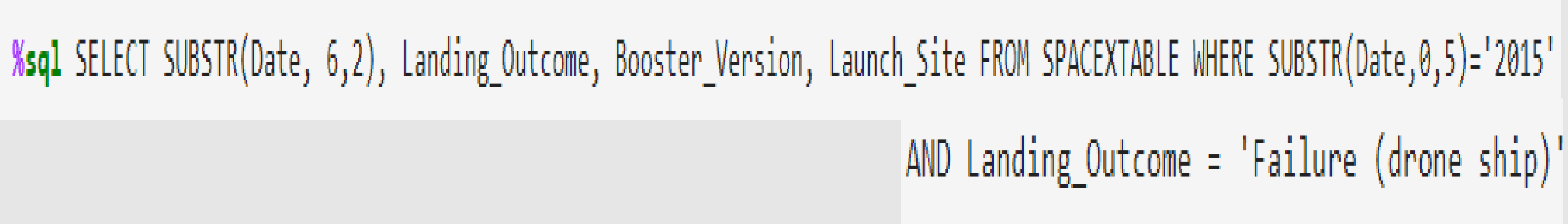
```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ == (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```



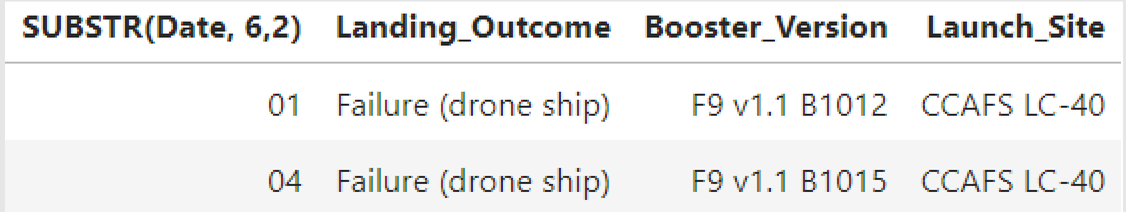
Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

- A query is written to list the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015.
- The command "SUBSTR(Date, 6,2)" to get the months and "SUBSTR(Date,0,5)='2015'" for the year.



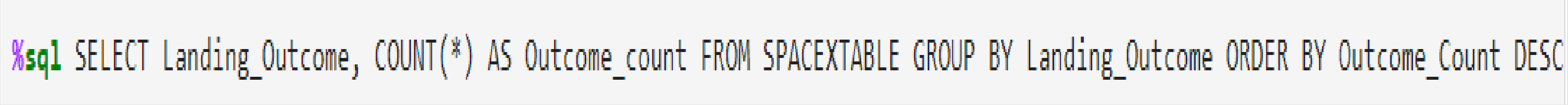
```
%sql SELECT SUBSTR(Date, 6,2), Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE SUBSTR(Date,0,5)='2015'
AND Landing_Outcome = 'Failure (drone ship)'
```



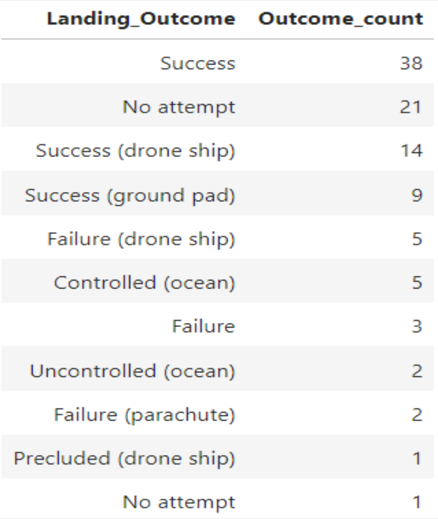
SUBSTR(Date, 6,2)	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- A query is written to 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.



```
%sql SELECT Landing_Outcome, COUNT(*) AS Outcome_count FROM SPACEXTABLE GROUP BY Landing_Outcome ORDER BY Outcome_Count DESC
```



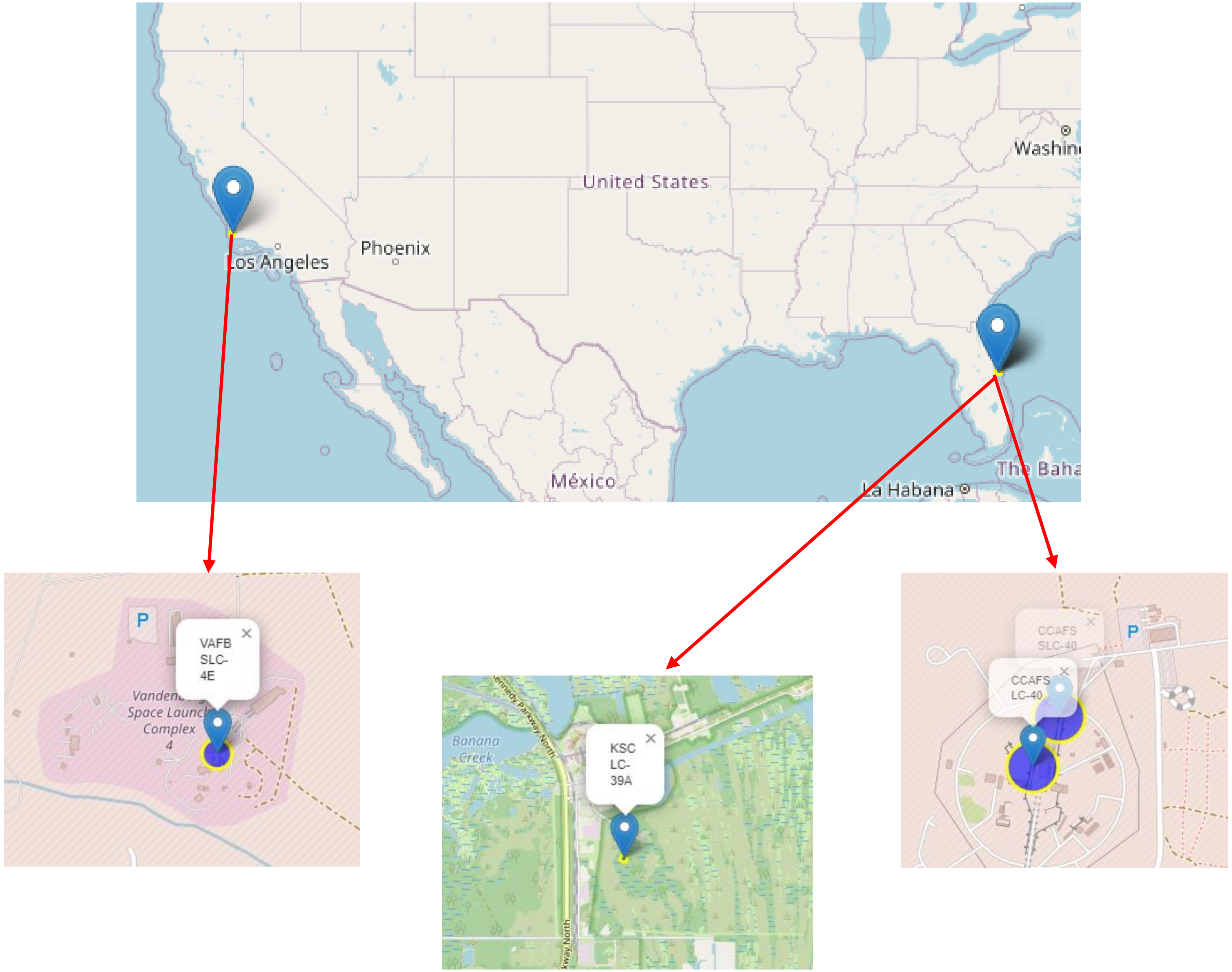
Landing_Outcome	Outcome_count
Success	38
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Failure	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1
No attempt	1

Section 3

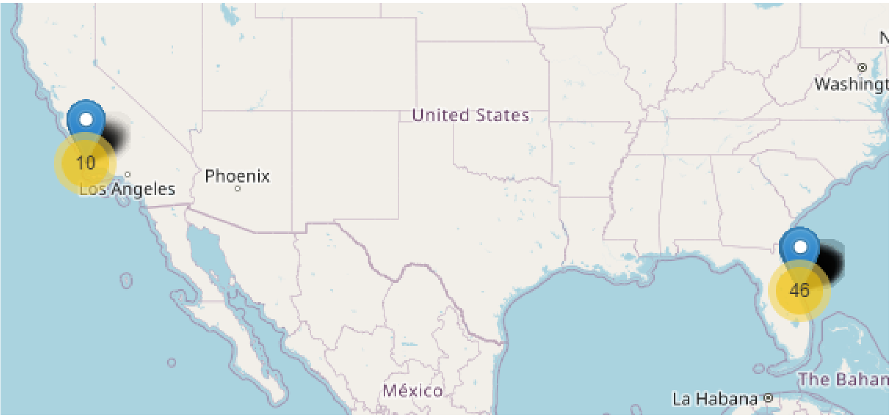
Launch Sites Proximities Analysis

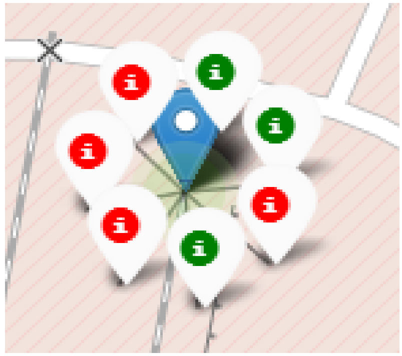
Launch Sites on the map with Folium

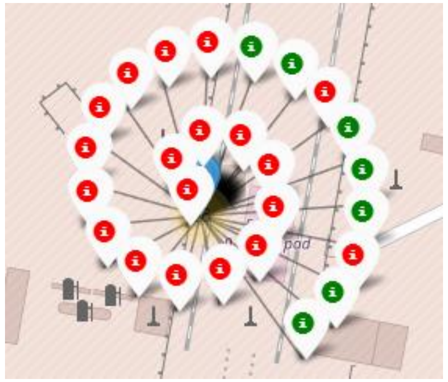
- We use **folium.Circle** to show each launch site, and **folium.Marker** to display the name of each site.

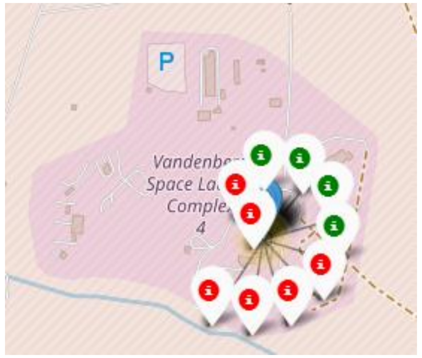


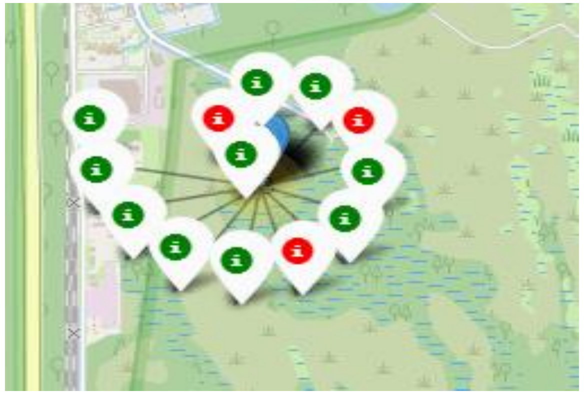
Launch outcome on the map with Folium







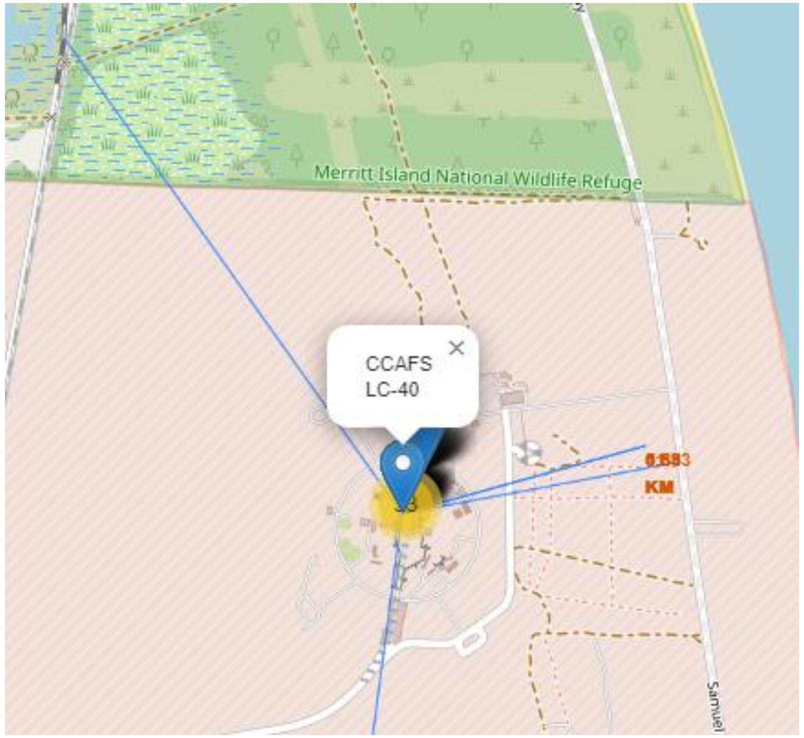


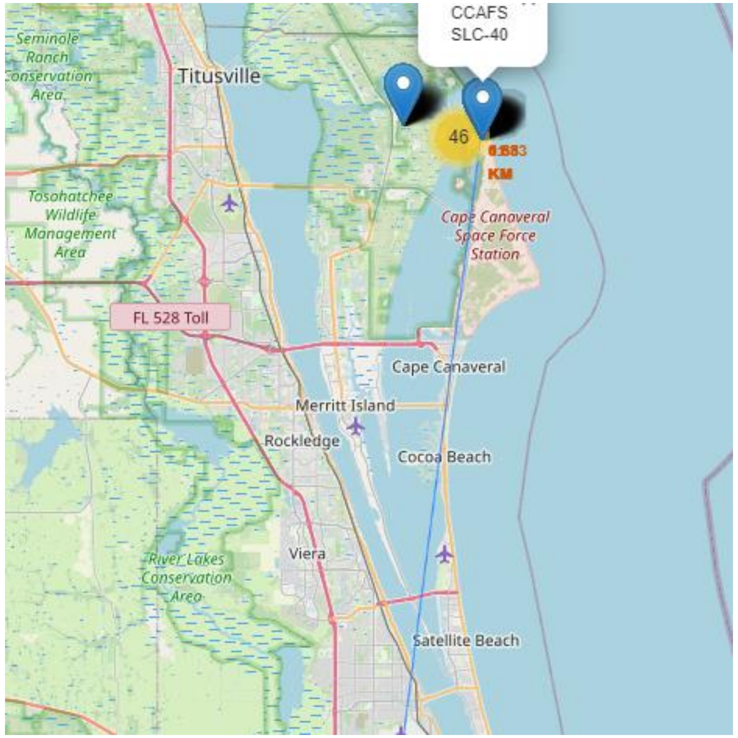


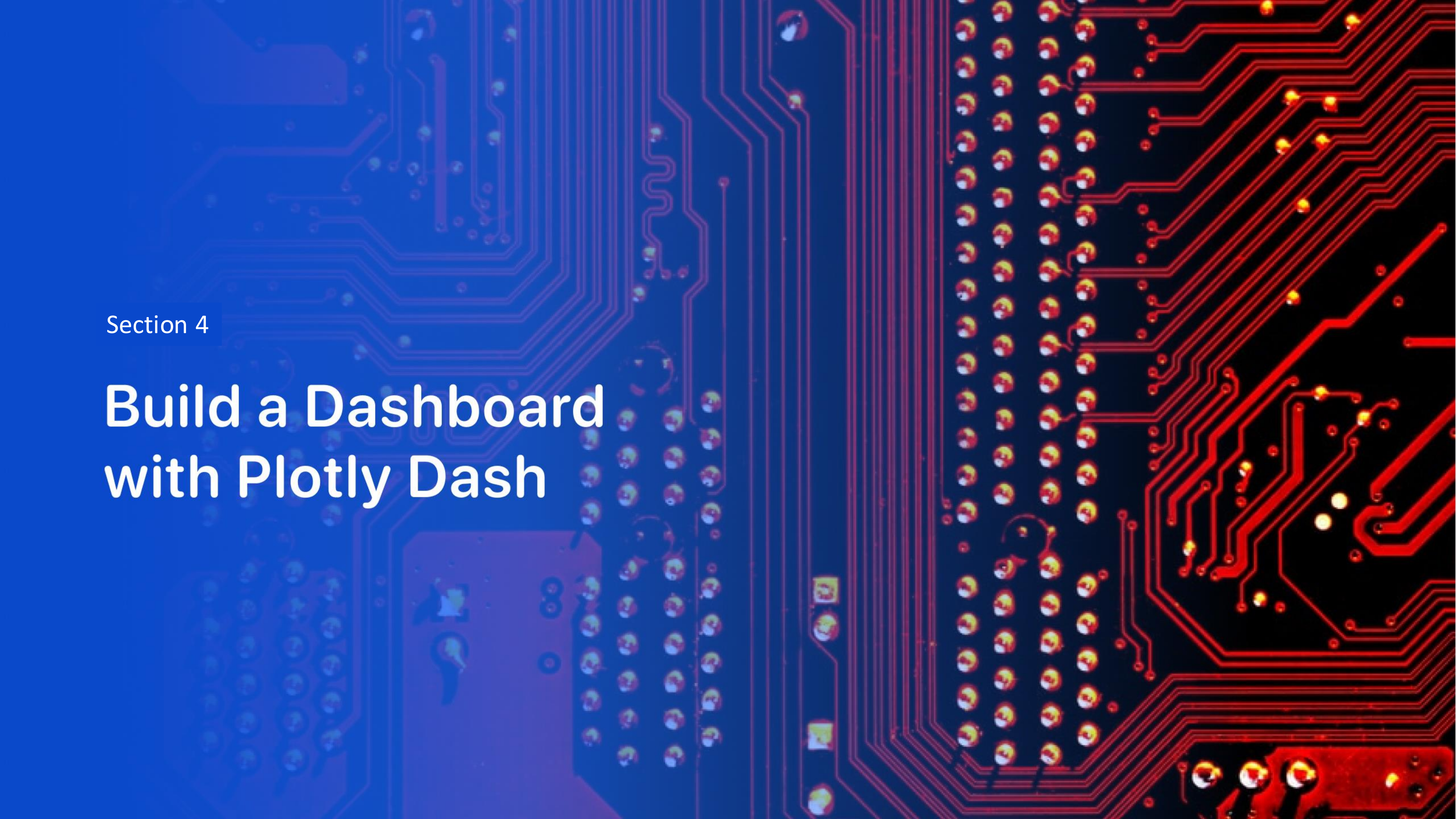
- Using a MarkerCluster object we can identify each launch site with the **successful outcomes** (green), and the **failed ones (red)**.

Distance between launch sites and its proximities

- Using **MousePosition** and a function to calculate distance, we can add lines with **folium.Polyline** between the launch site **CCAFS LC-40** and its proximities such as railway, highway, coastline, with distance calculated and displayed on the map.




CCAFS
SLC-40
Seminole Ranch Conservation Area
Titusville
46
KM
Tosohatchee Wildlife Management Area
Cape Canaveral Space Force Station
FL 528 Toll
Cape Canaveral
Merritt Island
Rockledge
Cocoa Beach
Viera
River Lakes Conservation Area
Satellite Beach

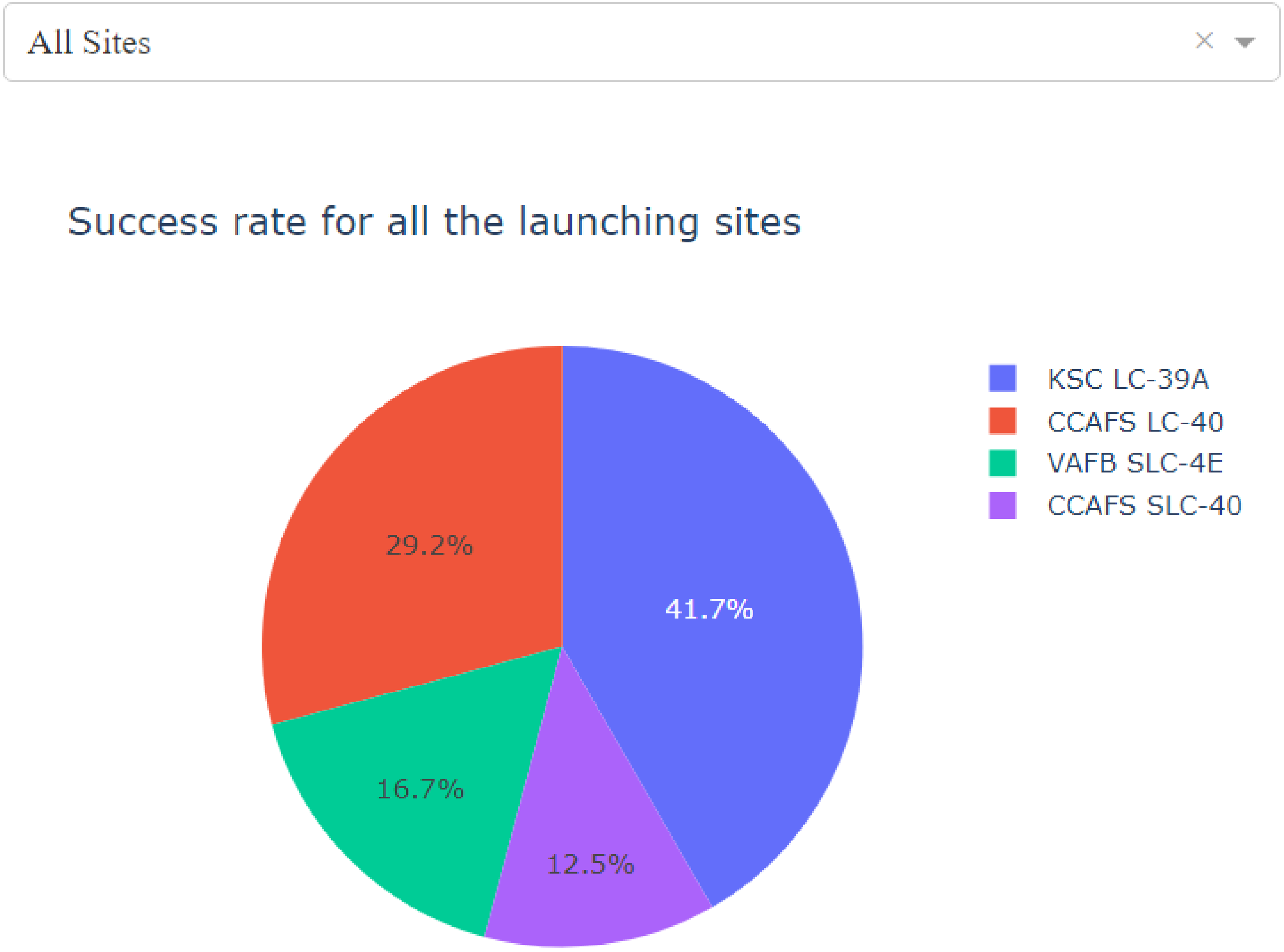


Section 4

Build a Dashboard with Plotly Dash

Launch Sites Success Pie chart

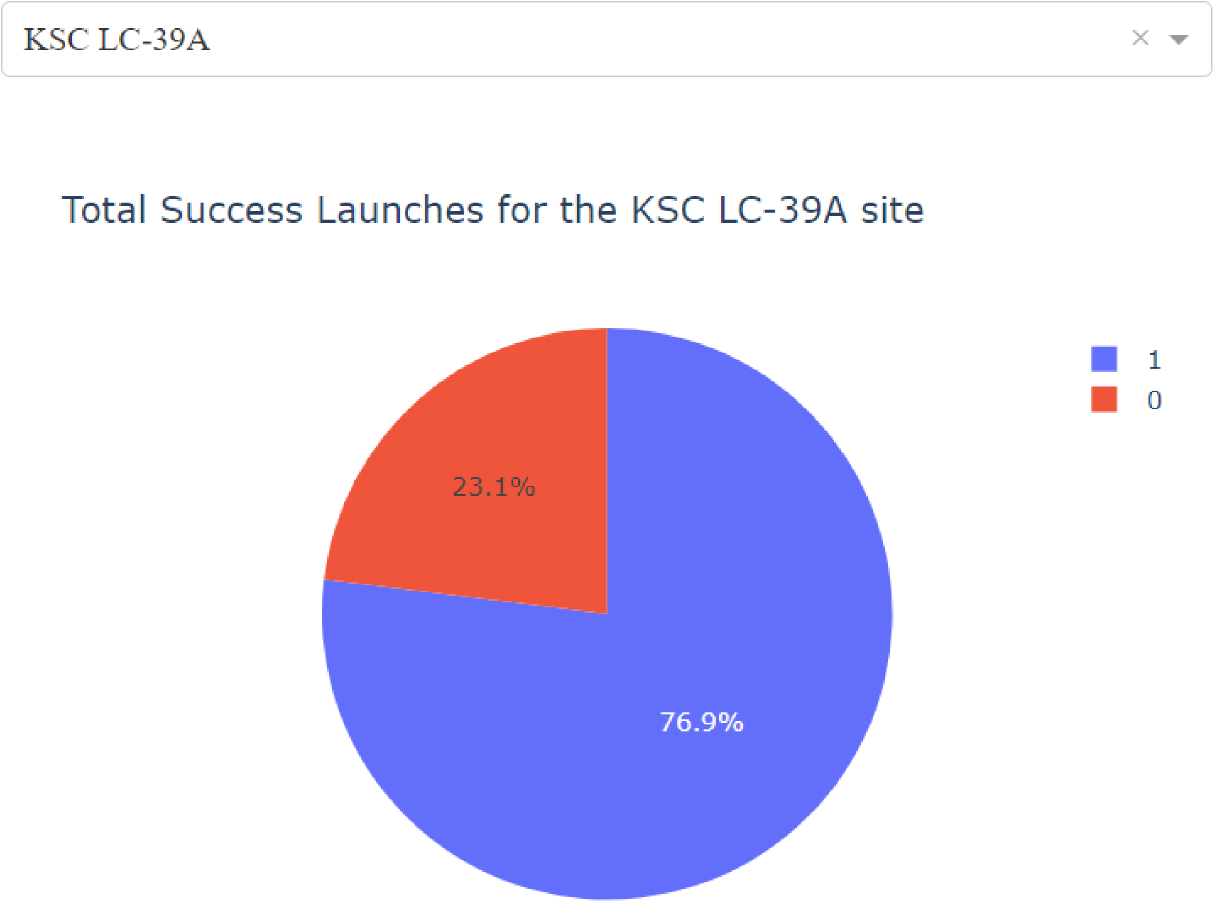
- We can see that que KSC LC-39A has the biggest success rate out of all the launch sites, with a **41.7%** of the total of launches.



All Sites

KSC LC-39A Success Rate Piechart

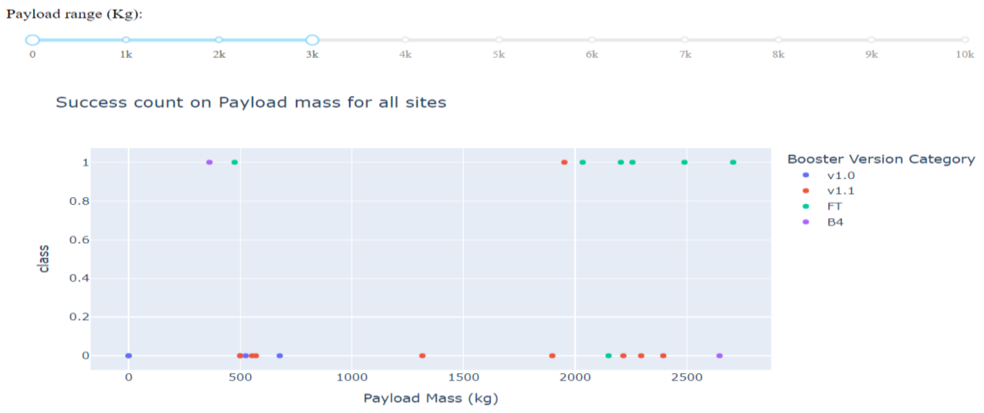
- We can see that the success rate for this specific launch site is **76.9%.** We can infer from this chart alone that this would be the best place to launch the rockets of SpaceY.

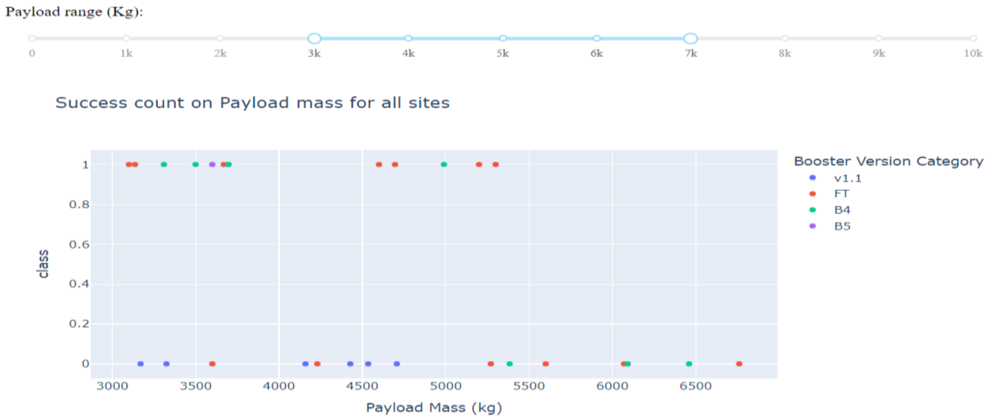


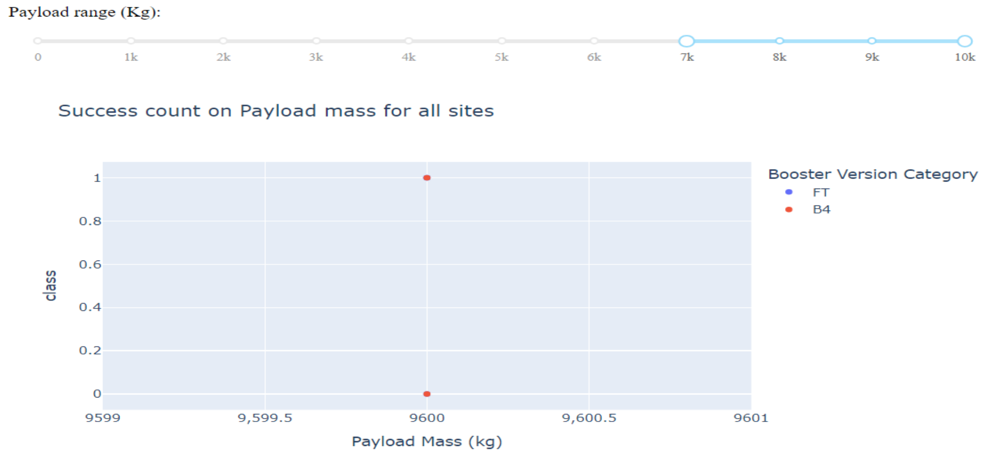
Success Rate on Payload mass for all sites per Booster Version

- Showing screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload ranges. [0kg - 3000kg], [3000kg – 7000kg] and [7000kg – 10000kg].

- We can see that the success rate is higher for a Payload Mass between 2000kg and 6000kg, using the Booster Version **FT.**





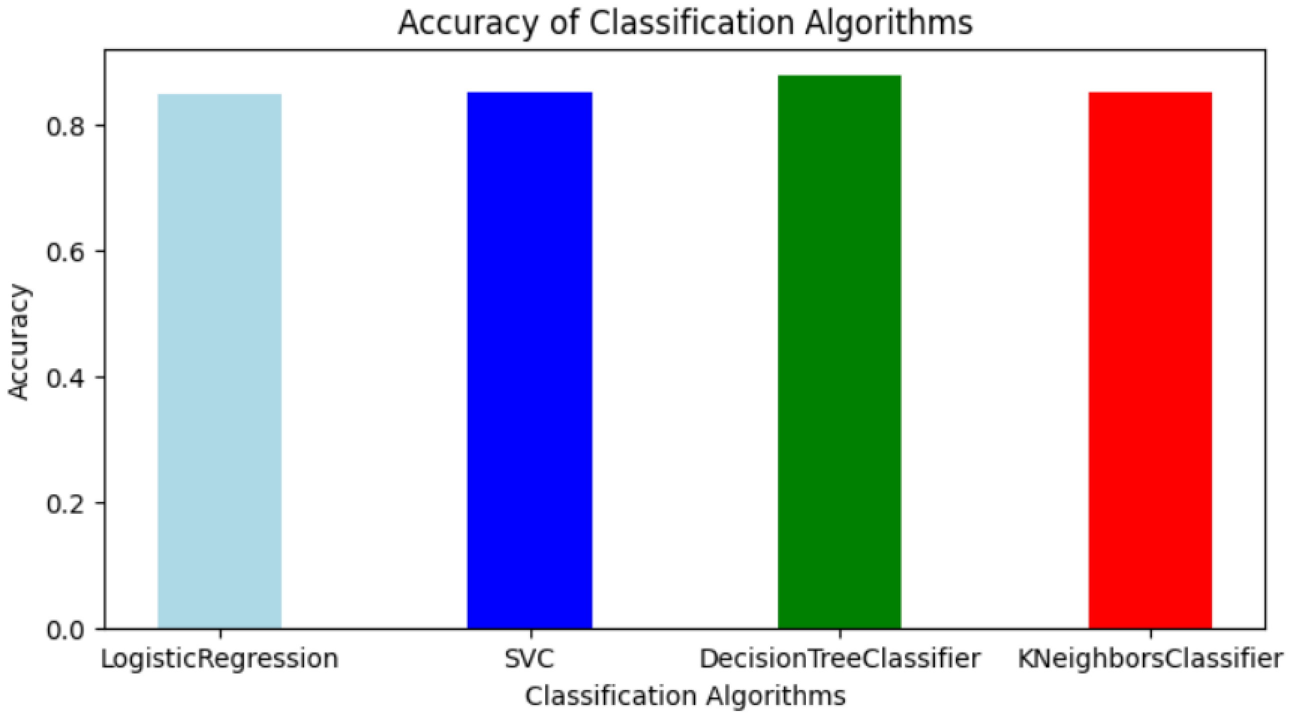


Section 5

Predictive Analysis (Classification)

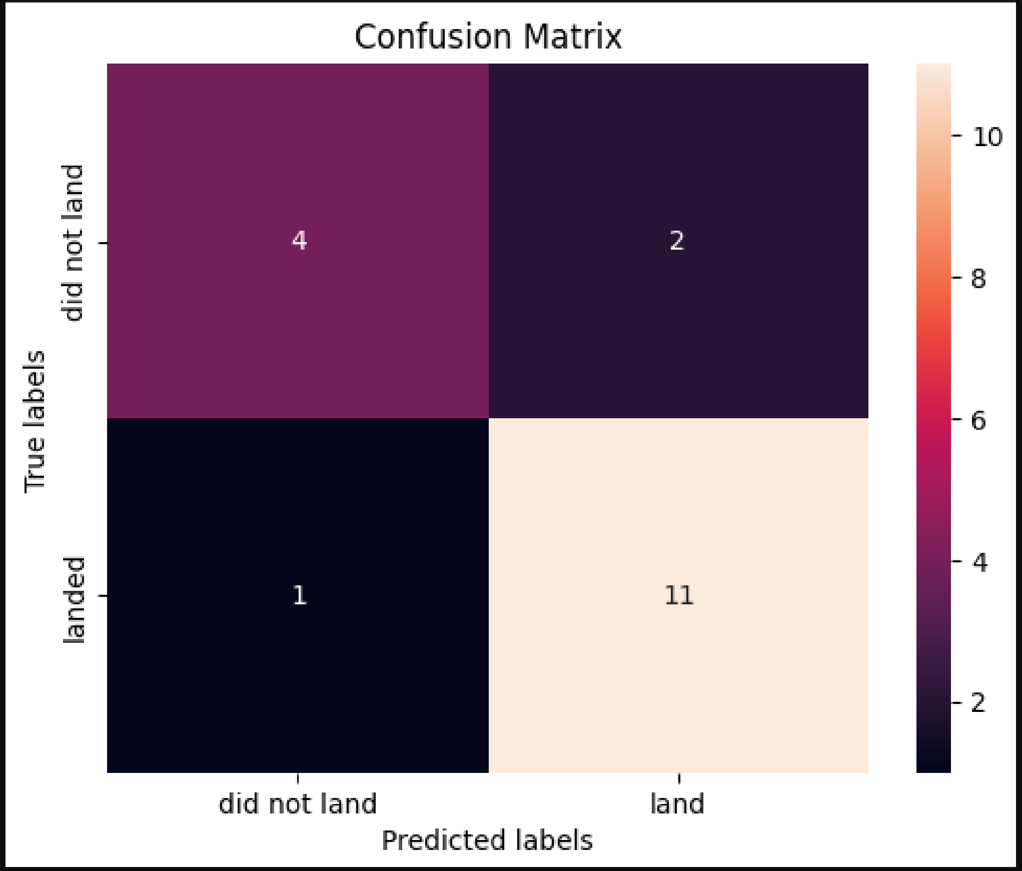
Classification Accuracy

- According to the bar chart, we can see that the **DecisionTreeClassifier** algorithm has the highest accuracy out of the four algorithms studied. We can also see that the other algorithms have a similar if not equal accuracy.



Confusion Matrix

- Looking at the confusion Matrix from the DecisionTreeClassifier method, considering the amount of true positives and true negatives that the results gave, we can say that it is an accurate method to get **true results.**



Conclusions

We can conclude that:

- The succes rate of the launches will increase the more flights are performed.
- The success rate increased in 2013 till 2020.
- The highest success rates are in orbits of type 'ES-L1', 'GEO', 'HEO', 'SSO' and 'VLEO'.
- The highest success rate and in the launch site '**KSC LC-39A'.**
- The Decision Tree Classifier algorithm, is the best machine learning algorithm for the task in question.

Thank you!