IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Muhammad Tamam Setia
22 June 2023

Outline

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

Executive Summary

Step of my data science project is:

1. I collect the data from SpaceX API,
2. Do data wrangling,
3. Exploratory data analysis with SQL and Python,
4. Make interactive data visualization with Python, and
5. Make predictive analysis.

The target of this project is to make a predictive analysis of how to choose reusable the first stage of rocket launch

Introduction

In this presentation, I will make a new dummy company with the name SpaceY. The company will follow the ways of SpaceX to choose the first stage of a rocket launch to reuse by data SpaceX. Because SpaceX's success makes reduced the cost by only 62 million dollars while other companies make costs by 165 million dollars. So I will process the data of SpaceX to make the decision with data science.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data collection with SpaceX REST API (https://api.spacexdata.com/v4/)
  - Also collect the data with web scraping
- Perform data wrangling
  - Process data only will use, handle the missing value, and make new column named ‘class’
- 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

I collect the data in two ways, REST API and web scraping.

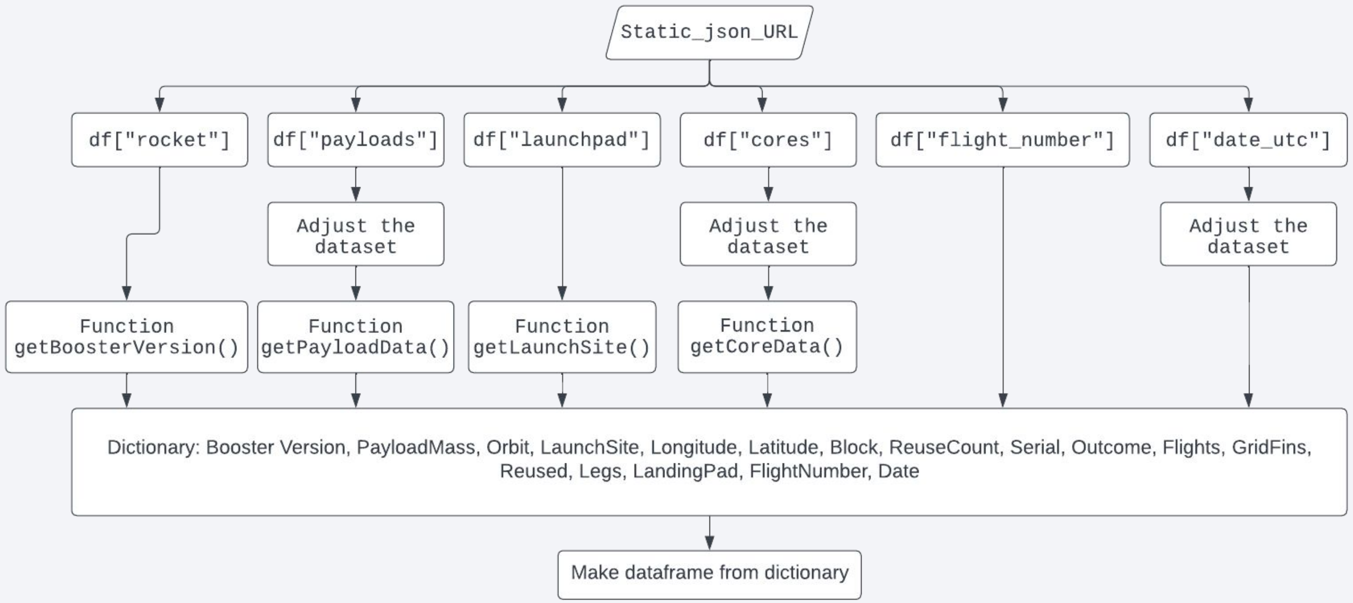
In REST API, I use static URL from IBM Data Science because in web API SpaceX has more complicated data. But the data will process with API SpaceX only for my need data by function Python

In web scraping, I use wikipedia.org to pull the request, and then I take only the table on that HTML file. After that, I extract column names from html table and make a dictionary from that.

Data Collection – SpaceX API

From static_json_URL, I take subset only rocket, payload, launchpad, cores, flight_number, and date_utc. After that, I adjust only the column needs that refers to the function. The function contains SpaceX API to process and pull the data from SpaceX. After that, I make a dictionary to accommodate the data from the function. Last, I convert the dictionary into a dataframe.

GitHub Data Collection Link



Static_json_URL

df["rocket"]

df["payloads"]

df["launchpad"]

df["cores"]

df["flight_number"]

df["date_utc"]

Adjust the dataset

Adjust the dataset

Adjust the dataset

Function getBoosterVersion()

Function getPayloadData()

Function getLaunchSite()

Function getCoreData()

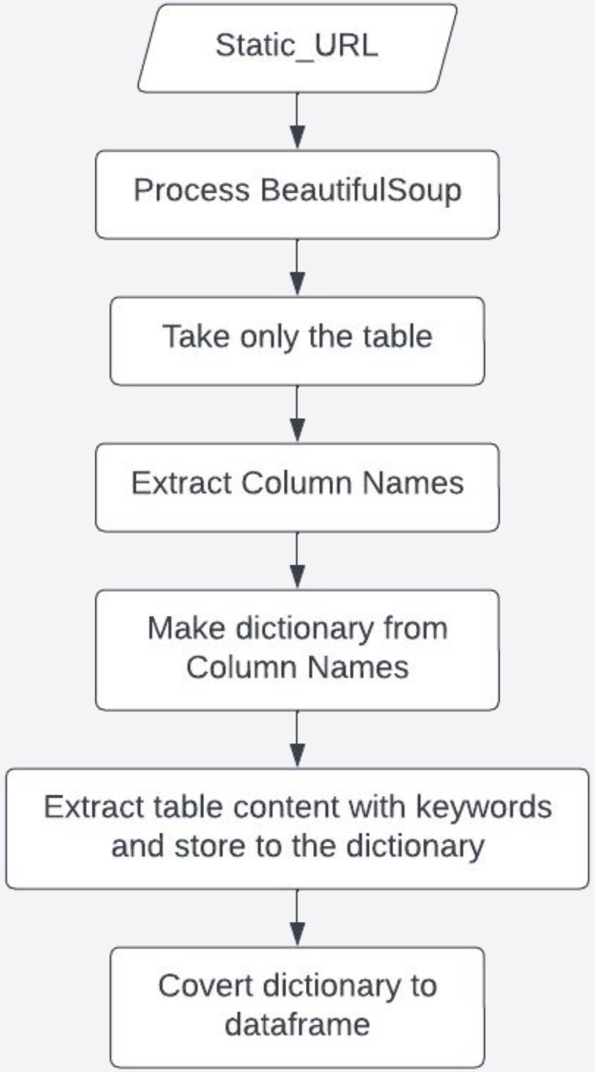
Dictionary: Booster Version, PayloadMass, Orbit, LaunchSite, Longitude, Latitude, Block, ReuseCount, Serial, Outcome, Flights, GridFins, Reused, Legs, LandingPad, FlightNumber, Date

Make dataframe from dictionary

Data Collection - Scraping

I take the static_URL from wikipedia.org. With beautifulSoup can change the URL into an HTML file, and then I take only the table HTML. From the table, I extract the column names and make a dictionary from that. After that, I extract table content with keywords and store it in the dictionary. Last, I convert the dictionary into a dataframe.

GitHub Webscraping Link


Static_URL
Process BeautifulSoup
Take only the table
Extract Column Names
Make dictionary from Column Names
Extract table content with keywords and store to the dictionary
Covert dictionary to dataframe

Data Wrangling

If I start from data collection REST API SpaceX, I need to adjust only Falcon 9 in BoosterVersion and fill the missing value of PayloadMass with the mean.

After that, I make a new column named Class for bad and good outcomes.

GitHub Data Wrangling Link

EDA with Data Visualization

This is the list of chart I made:

- Flight Number vs. Launch Site
- Payload vs. Launch Site
- Success Rate vs. Orbit Type
- Flight Number vs. Orbit Type
- Payload vs. Orbit Type
- Launch Success Yearly Trend

GitHub EDA with DaViz Link

EDA with SQL

This is the list of query I made:

- All Launch Site Names
- Launch Site Names Begin with 'CCA'
- Total Payload Mass
- Average Payload Mass by F9 v1.1
- First Successful Ground Landing Date
- Successful Drone Ship Landing with Payload between 4000 and 6000
- Total Number of Successful and Failure Mission Outcomes
- Boosters Carried Maximum Payload
- 2015 Launch Records
- Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

GitHub EDA with SQL Link

Build an Interactive Map with Folium

First I make markers for all launch sites and give them launch site names. And then, I make a marker again, but in this section, I make a marker for the launch outcome on every launch site. So the marker will be stacked. After that, I make a line and give the distance for the line.

GitHub Folium Link

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions

GitHub Plotly Dash Link

Predictive Analysis (Classification)

I start with preprocessing data and make X and Y. X data from IBM Data Science, it is the data with one hot encoder. Y data is a list of ‘class’ (the target of predictive supervised learning). After that I process the data with four models, it is linear regression, support vector machine, decision tree, and k-nearest neighbors. Every model will have an accuracy and confusion matrix, and I can compare the accuracy to find best model.

GitHub Predictive Analysis Link

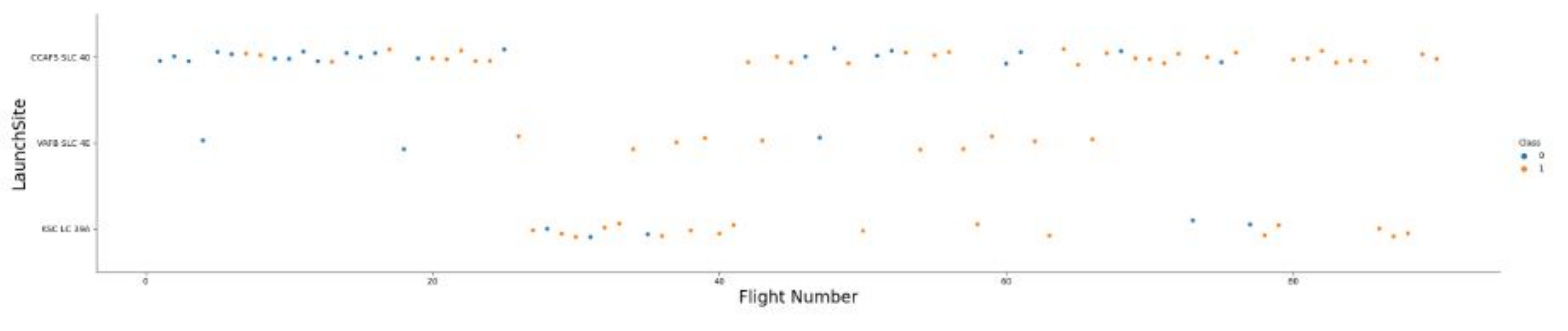
Results

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

Section 2

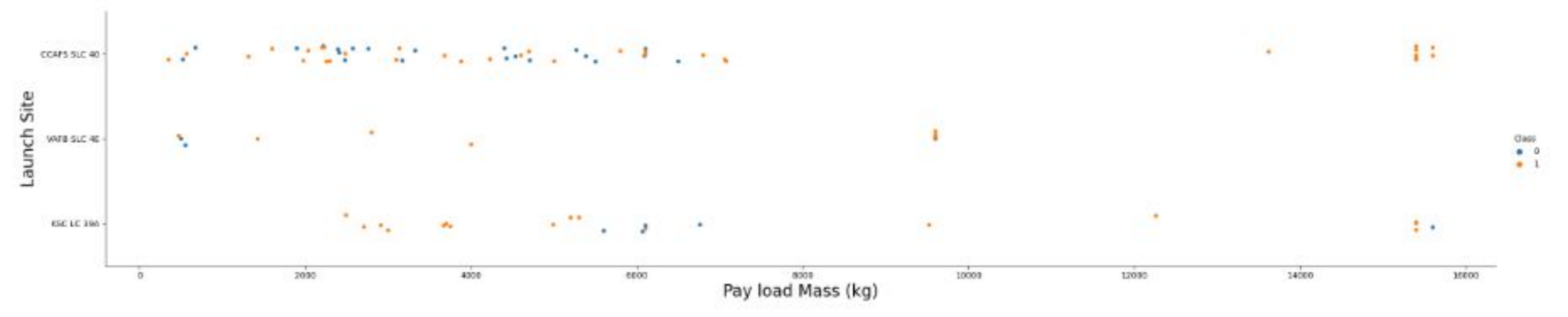
Insights drawn from EDA

Flight Number vs. Launch Site



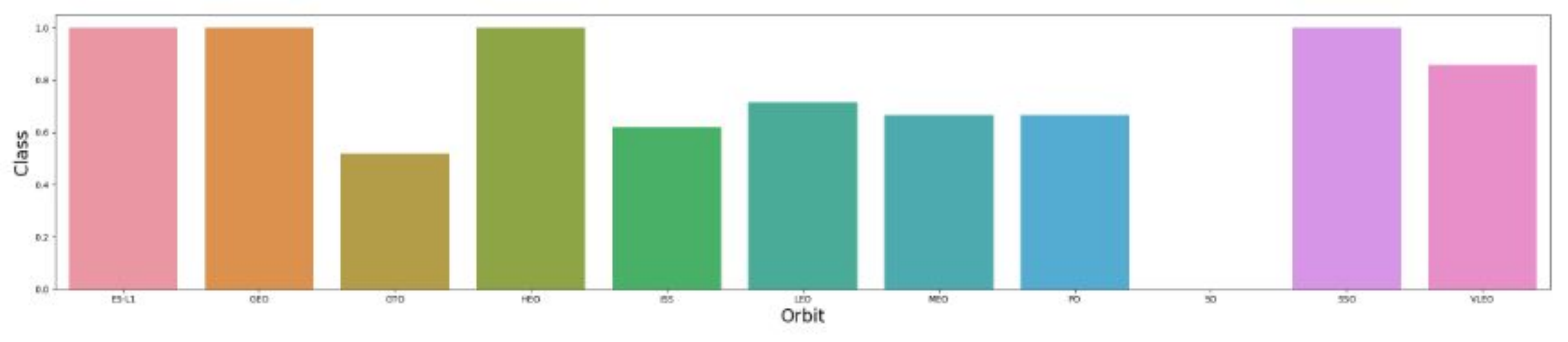
In the beginning, CCAFS SLC 40 was more often used but made more failures. After 25, the launch site moves to KSC LC 39A and makes the success rate better than before. After 40, CCAFS SLC 40 is used again and make a better success rate. VAFB SLC 4E is often used.

Payload vs. Launch Site



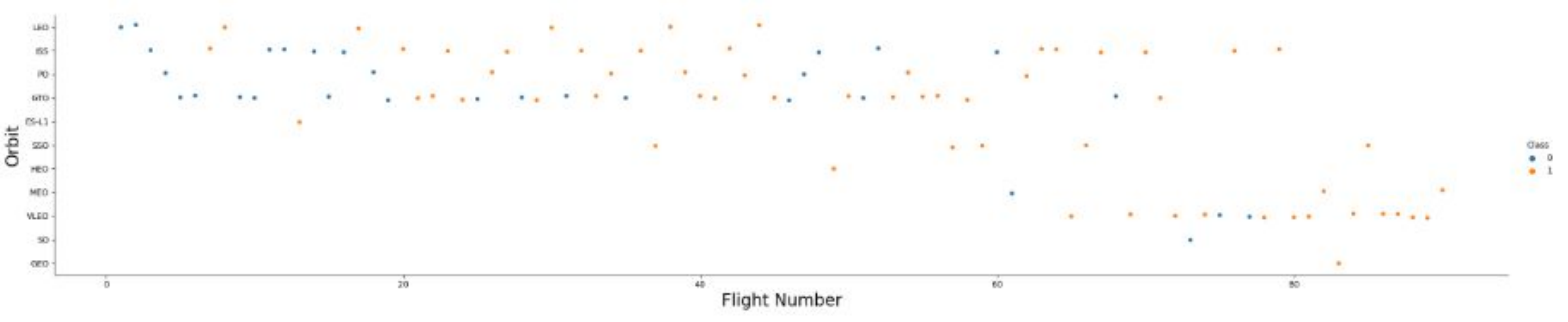
Launch at payload mass below 8000 kg, more than payload mass above 8000 kg. But I think payload and launch site haven’t relationship.

Success Rate vs. Orbit Type



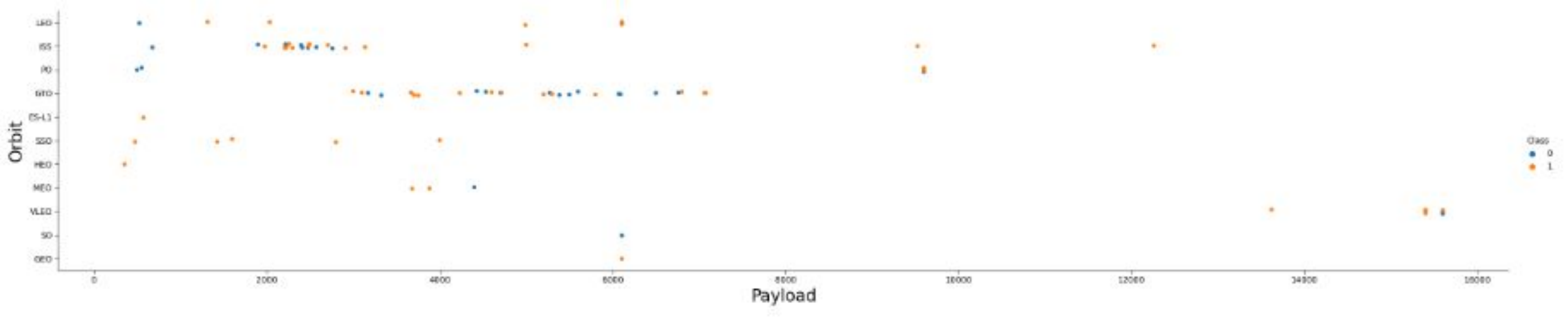
ES-L1, GEO, HEO, and SSO have a 100 percent success rate, and SO has a 0 percent success rate. Other orbit types have above 50 percent success rate.

Flight Number vs. Orbit Type



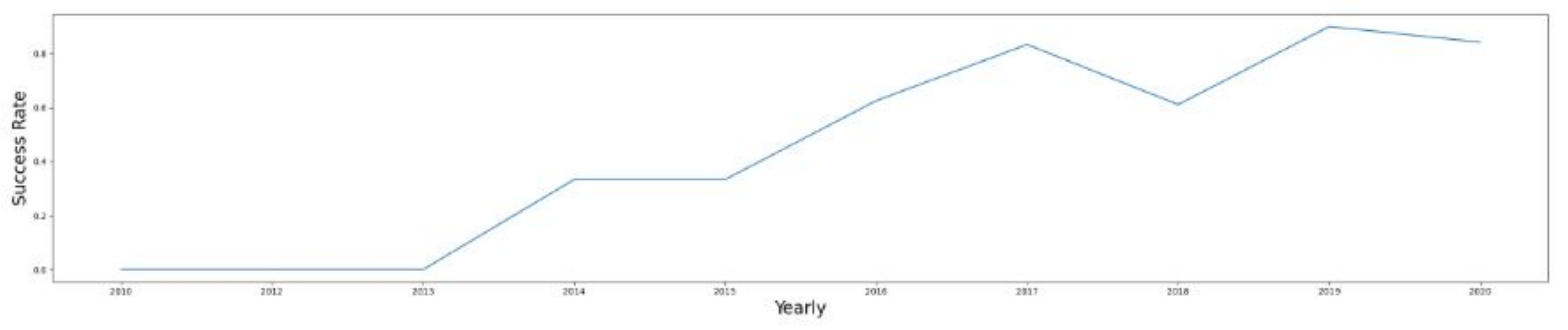
I think flight number and orbit type haven’t relationship.

Payload vs. Orbit Type



I think payload and orbit type haven’t relationship.

Launch Success Yearly Trend



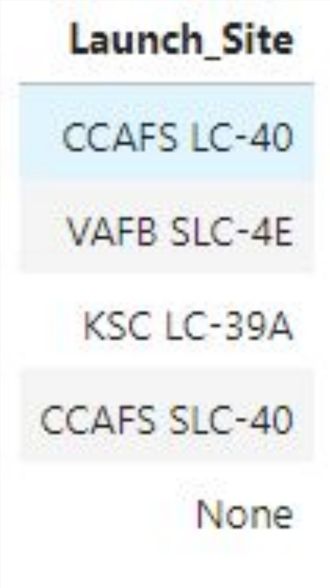
The trend of Success Rate is good. The success rate increase year to year. Only in the years 2017 and 2019, the success rate has decreased.

All Launch Site Names

The Query is:

```
SELECT
    DISTINCT(Launch_Site)
FROM
    SPACEXTBL
```

The result of query is:



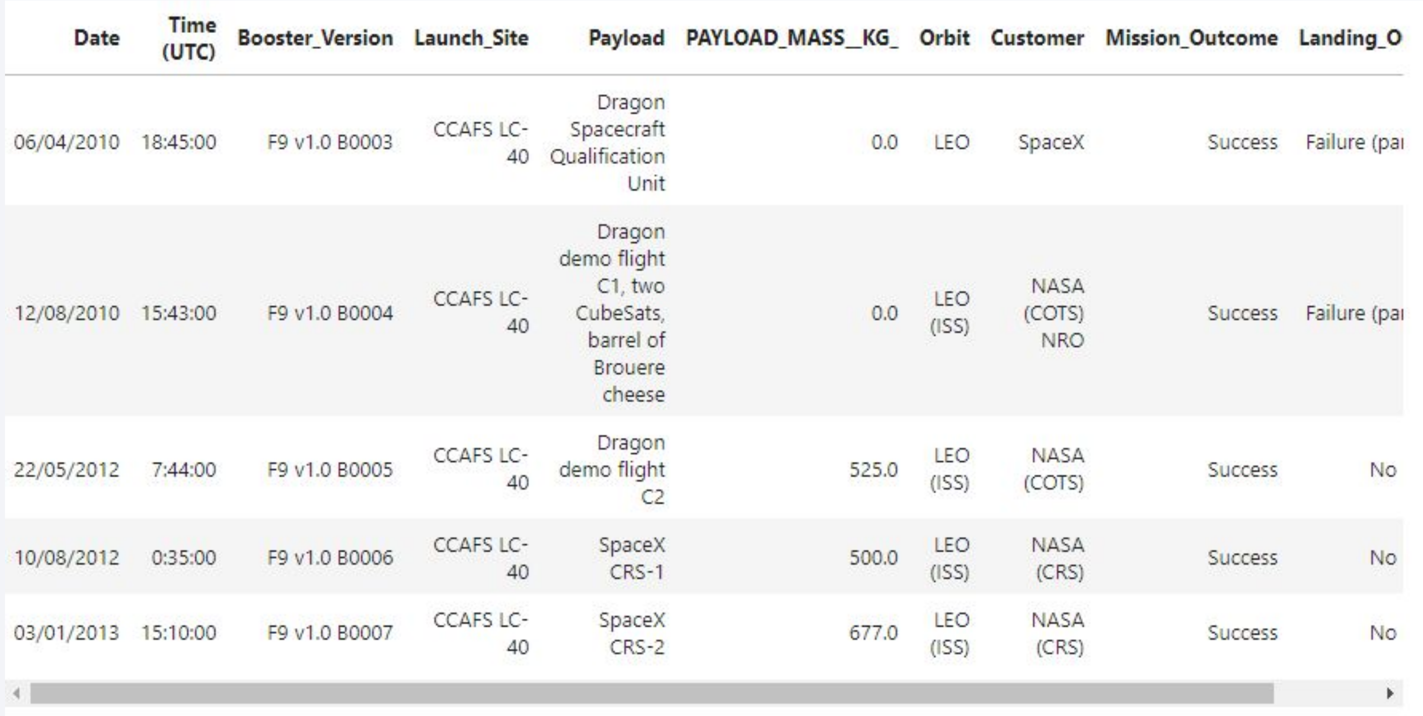
Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Launch Site Names Begin with 'CCA'

The query is:

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

The result of the query is:



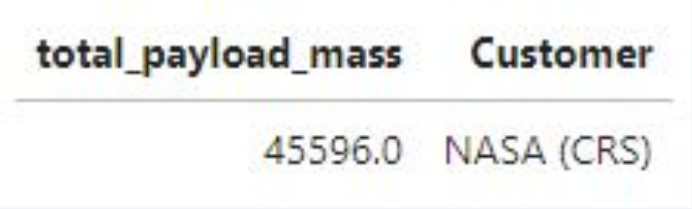
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_O
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (pa
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (pa
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No

Total Payload Mass

The query is:

```
SELECT
    SUM(PAYLOAD_MASS__KG_) AS total_payload_mass,
    Customer
FROM
    SPACEXTBL
WHERE
    Customer = 'NASA (CRS)' GROUP BY Customer
```

The result of the query is:



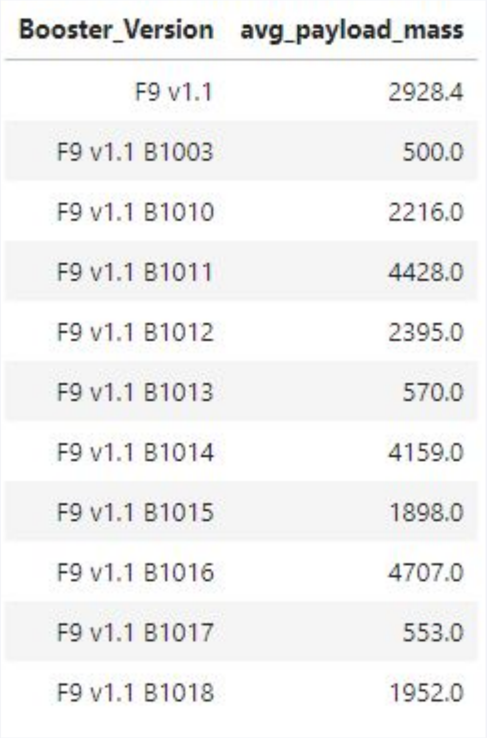
total_payload_mass	Customer
45596.0	NASA (CRS)

Average Payload Mass by F9 v1.1

The query is:

```
SELECT
    Booster_Version,
    AVG(PAYLOAD_MASS__KG_) AS avg_payload_mass
FROM
    SPACEXTBL
WHERE
    Booster_Version LIKE 'F9 v1.1%'
GROUP BY
    Booster_Version
```

The result of the query is:



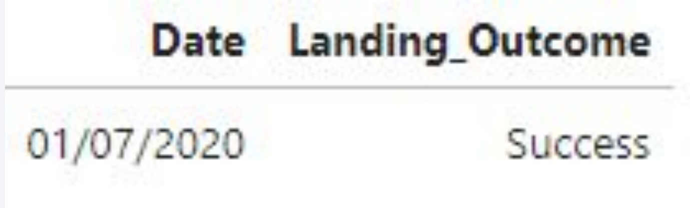
Booster_Version	avg_payload_mass
F9 v1.1	2928.4
F9 v1.1 B1003	500.0
F9 v1.1 B1010	2216.0
F9 v1.1 B1011	4428.0
F9 v1.1 B1012	2395.0
F9 v1.1 B1013	570.0
F9 v1.1 B1014	4159.0
F9 v1.1 B1015	1898.0
F9 v1.1 B1016	4707.0
F9 v1.1 B1017	553.0
F9 v1.1 B1018	1952.0

First Successful Ground Landing Date

The query is:

```
SELECT
      MIN(DATE) AS Date,
      Landing_Outcome
FROM
      SPACEXTBL
WHERE
      Landing_Outcome = 'Success'
```

The result of the query is:



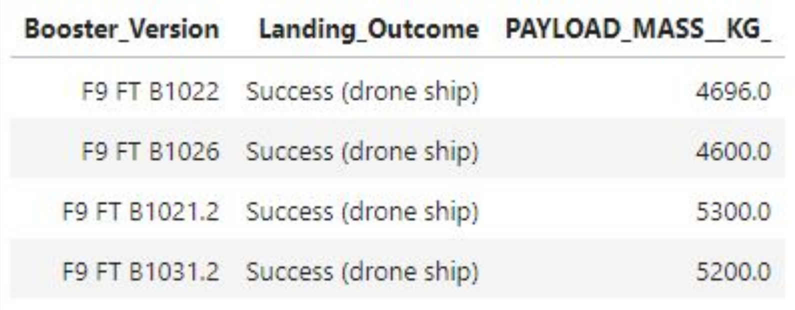
Date	Landing_Outcome
01/07/2020	Success

Successful Drone Ship Landing with Payload between 4000 and 6000

The query is

```
SELECT
    Booster_Version,
    Landing_Outcome,
    PAYLOAD_MASS__KG_
FROM
    SPACEXTBL
WHERE
    Landing_Outcome = 'Success (drone ship)' AND
    PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

The result of the query is:



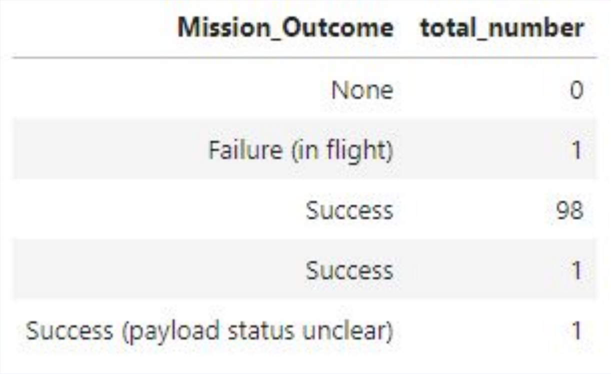
Booster_Version	Landing_Outcome	PAYLOAD_MASS__KG_
F9 FT B1022	Success (drone ship)	4696.0
F9 FT B1026	Success (drone ship)	4600.0
F9 FT B1021.2	Success (drone ship)	5300.0
F9 FT B1031.2	Success (drone ship)	5200.0

Total Number of Successful and Failure Mission Outcomes

The query is:

```
SELECT
      Mission_Outcome,
      COUNT(Mission_Outcome) AS total_number
FROM
      SPACEXTBL
GROUP BY
      Mission_Outcome
```

The result of the query is:



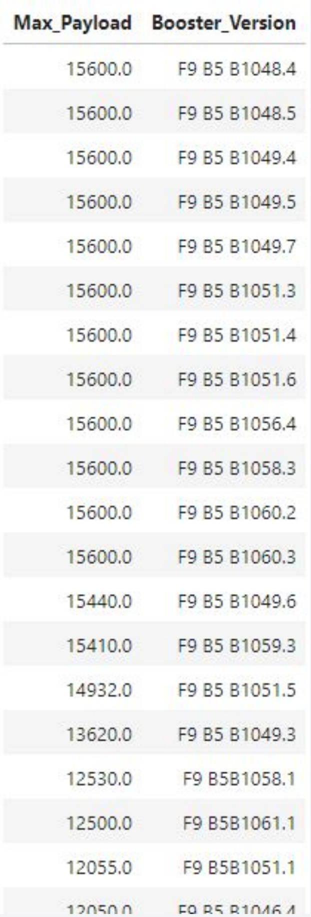
Mission_Outcome	total_number
None	0
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

The query is:

```
SELECT
    *
FROM
    (SELECT
        MAX(PAYLOAD_MASS__KG_) AS Max_Payload,
        Booster_Version
    FROM
        SPACEXTBL
    GROUP BY
        Booster_Version)
ORDER BY
    Max_Payload
```

The result of the query is:



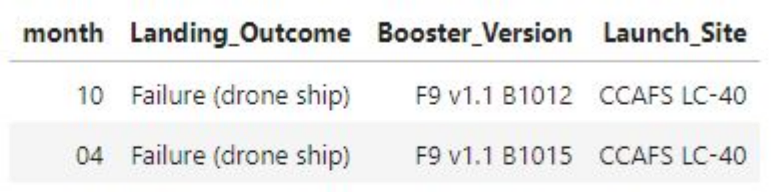
Max_Payload	Booster_Version
15600.0	F9 B5 B1048.4
15600.0	F9 B5 B1048.5
15600.0	F9 B5 B1049.4
15600.0	F9 B5 B1049.5
15600.0	F9 B5 B1049.7
15600.0	F9 B5 B1051.3
15600.0	F9 B5 B1051.4
15600.0	F9 B5 B1051.6
15600.0	F9 B5 B1056.4
15600.0	F9 B5 B1058.3
15600.0	F9 B5 B1060.2
15600.0	F9 B5 B1060.3
15440.0	F9 B5 B1049.6
15410.0	F9 B5 B1059.3
14932.0	F9 B5 B1051.5
13620.0	F9 B5 B1049.3
12530.0	F9 B5B1058.1
12500.0	F9 B5B1061.1
12055.0	F9 B5B1051.1
12050.0	F9 B5 B1046.4

2015 Launch Records

The query is:

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

The result of the query is:



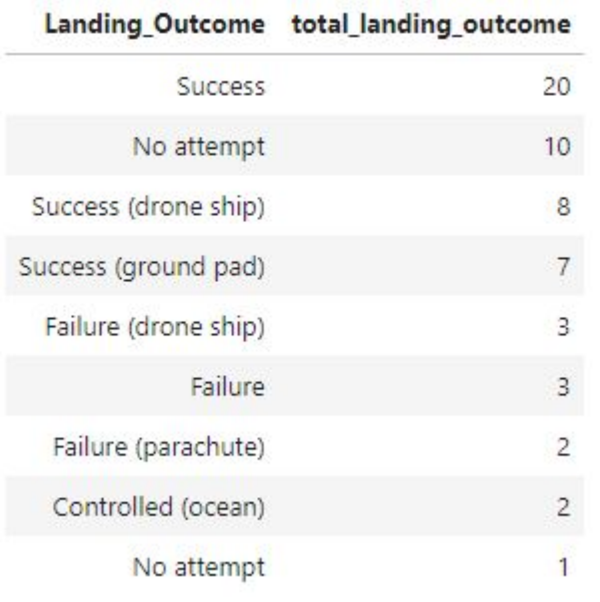
month	Landing_Outcome	Booster_Version	Launch_Site
10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

The query is:

```
SELECT
    Landing_Outcome,
    COUNT(Landing_Outcome) AS total_landing_outcome
FROM
    (SELECT
        Date,
        Landing_Outcome
    FROM
        SPACEXTBL
    WHERE
        Date BETWEEN '04-06-2010' AND '20-03-2017')
GROUP BY 1
ORDER BY 2 DESC
```

The result of the query is:

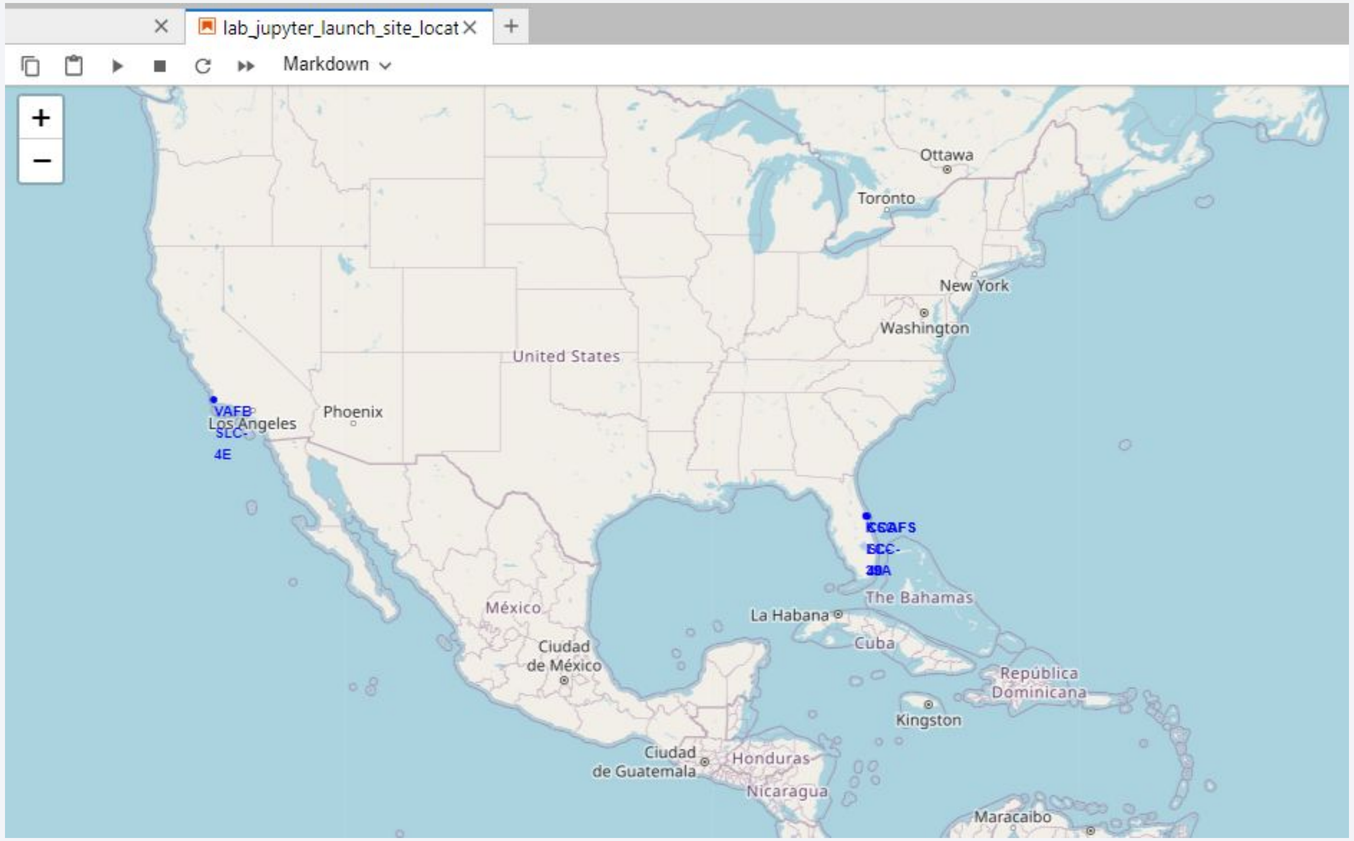


Landing_Outcome	total_landing_outcome
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	7
Failure (drone ship)	3
Failure	3
Failure (parachute)	2
Controlled (ocean)	2
No attempt	1

Section 3

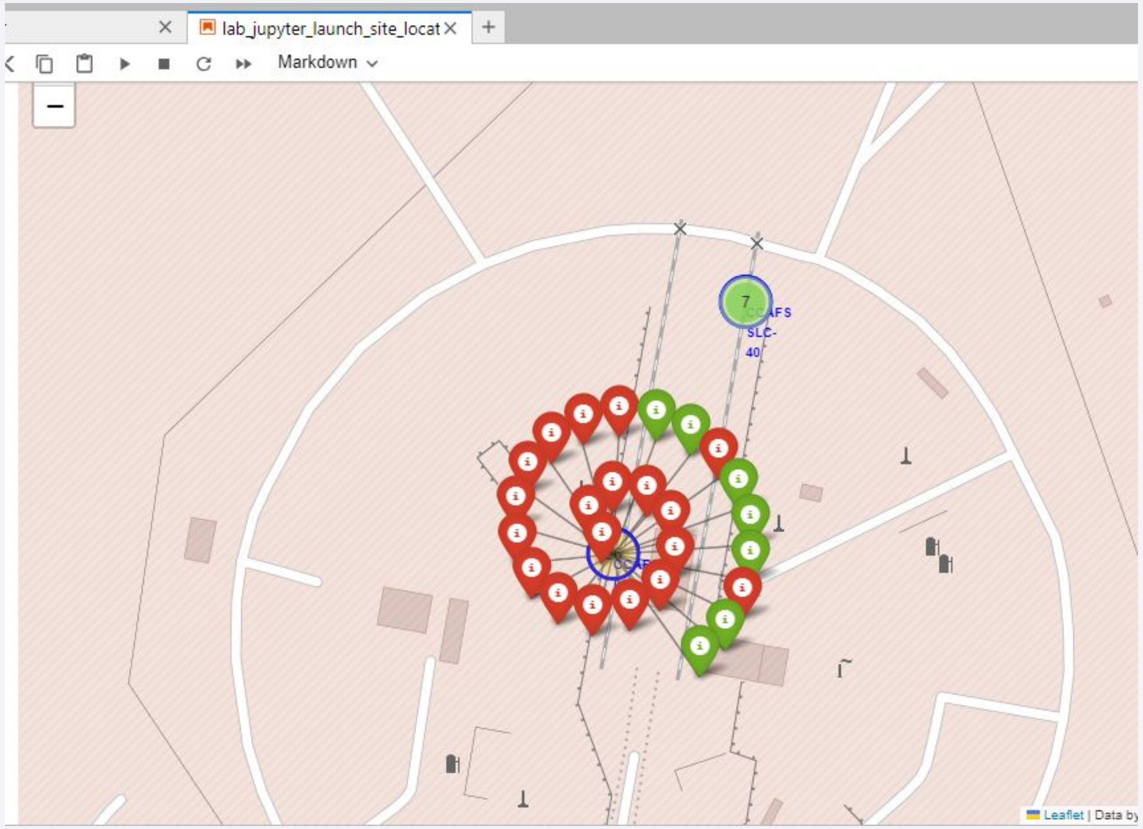
Launch Sites Proximities Analysis

Launch Site in Folium



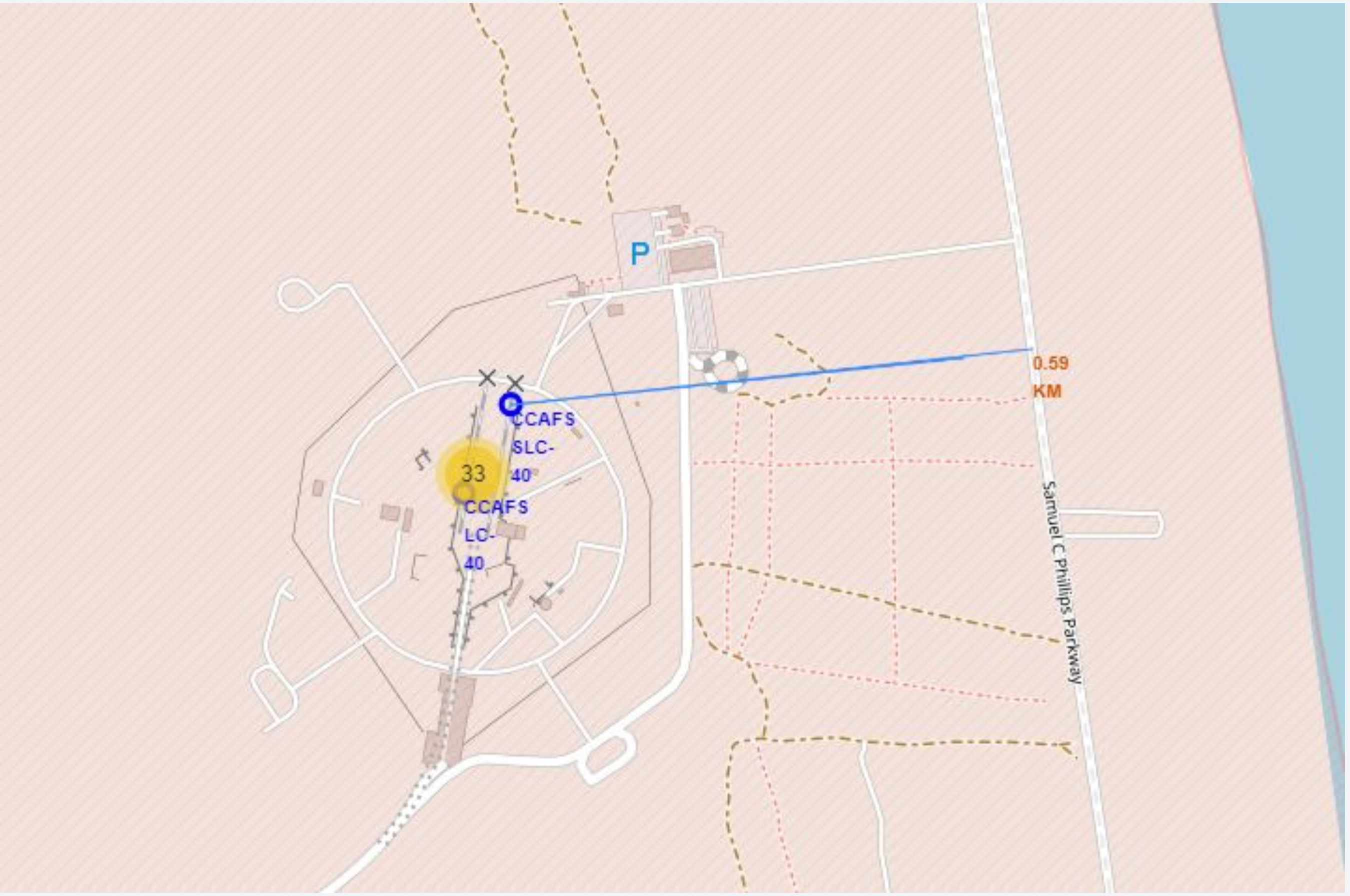
We can see 3 launch sites on this map. But CCAFS SLC 40 and KSC LC 39A are nearby. So in this map, CCAFS SLC 40 and KSC LC 39A will be pilled up.

CCSAF SLC 40 Landing Outcome



This CCAFS SLC 40 detail for landing outcome. We can see 19 in 26 are failures, and only 7 are successes.

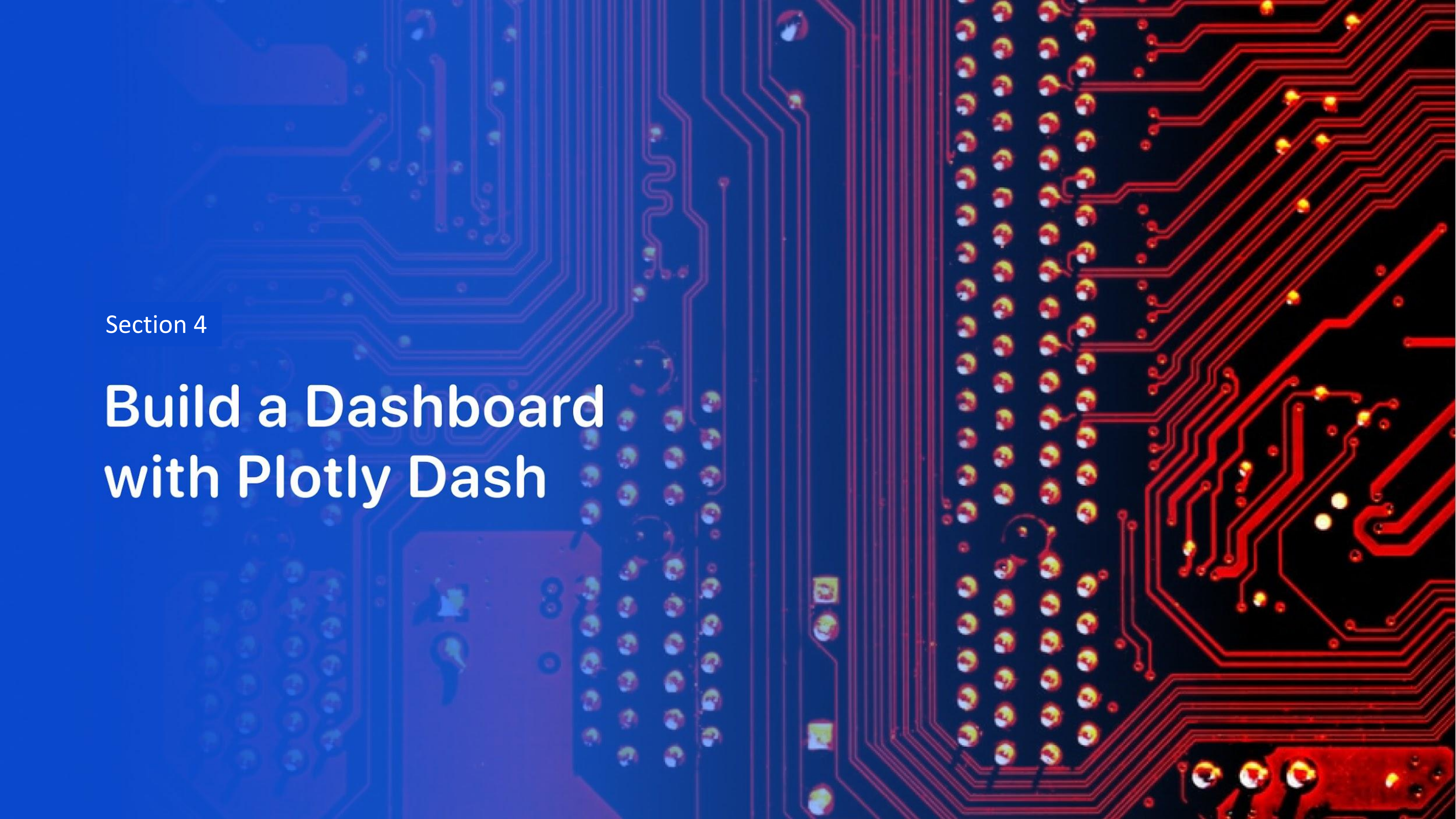
Folium Distance



This figure show line from CCAFS SLC 40 to Samuel C Phillips Parkway. Also display the distance is 0,59 KM.

Section 4

Build a Dashboard with Plotly Dash



<Dashboard Screenshot 1>

- Replace <Dashboard screenshot 1> title with an appropriate title
- Show the screenshot of launch success count for all sites, in a piechart
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 2>

- Replace <Dashboard screenshot 2> title with an appropriate title

- Show the screenshot of the piechart for the launch site with highest launch success ratio

- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 3>

- Replace <Dashboard screenshot 3> title with an appropriate title

- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

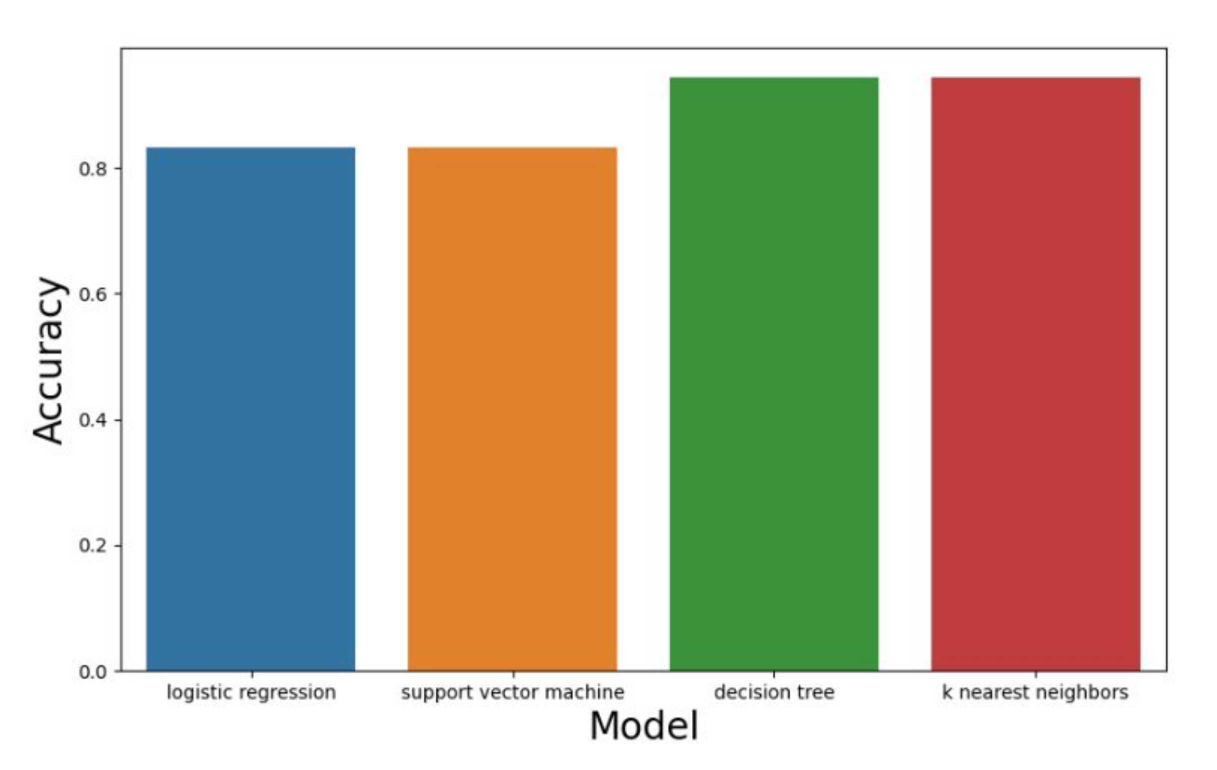
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section 5

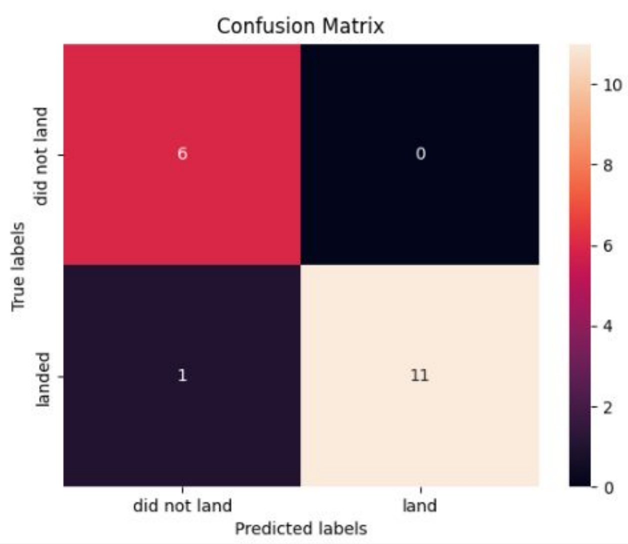
Predictive Analysis (Classification)

Classification Accuracy

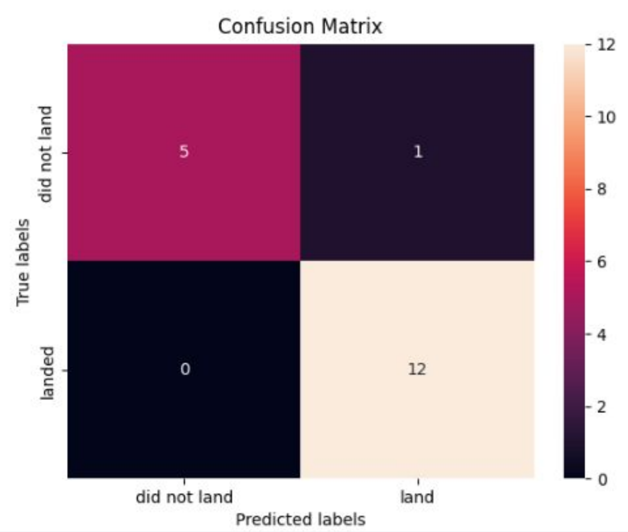
The best model in this prediction analysis is the decision tree and k nearest neighbors.



Confusion Matrix



This is the confusion matrix decision tree. We can see, only one prediction wrong, it is predict ‘did not land’ but actually is ‘landed’



This is the confusion matrix k-nearest neighbors. We can see, only one prediction wrong, it is predict ‘land’ but actually is ‘did not land’

Conclusions

- We can use decision tree or k-nearest neighbors to this predictive analysis.
- The accuracy of predictive analysis is 94%

Appendix

https://github.com/tamamsetia/finalproject_ds_coursera/tree/main

Thank you!