IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

IMTIAZ AHMED
03/10/2023

Outline

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

Executive Summary

- Summary of methodologies

  This project was to analyse the success rate of landings of space craft, the data was collected from spacex website and using python and sql the data was cleaned, processed, visualized and subjected to predictive analysis

- Summary of all results

  The findings in the data were pretty clear, the payload weight had great impact on landing and after 2013 there were a lot of successful landings. The predictive analysis had better outcomes on decision tree model.

Introduction

- Project background and context

  The customer wanted to find answers on successful landings of rocket. So we collected data from spacex website. We analysed the data to find deep insights on rocket landings and factors that influenced successful rocket landing.

- Problems you want to find answers

  1. Is there any direct factors that influenced landing?

Section 1

Methodology

Methodology

Executive Summary

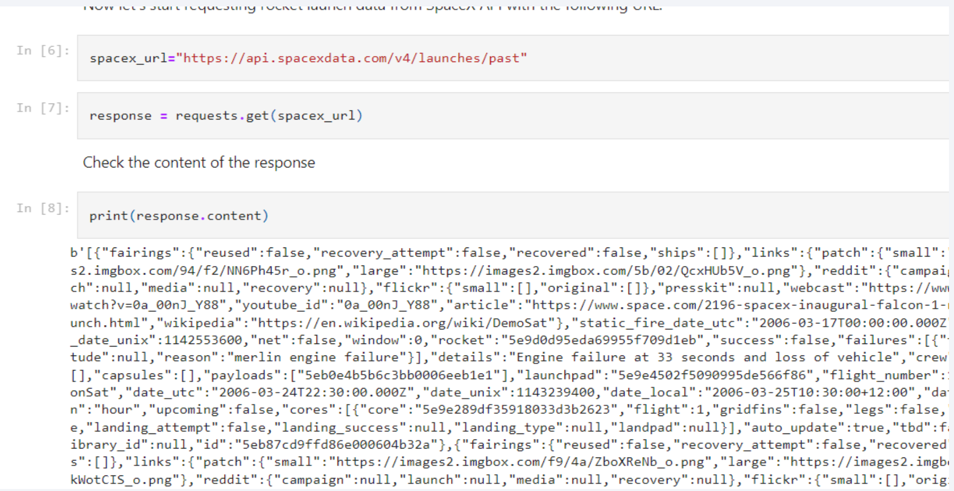
- Data collection methodology:
  - The data was collected from spacex website through api and webscraping of html pages
- Perform data wrangling
  - Cleaned the data to make it usable for further process.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Used scikit learn and various ml models for predictive analysis

Data Collection

- Describe how data sets were collected.
    - The datasets were collected by two methods
        - API – Used the spacex website api to extract data directly from the server.
        - Webscraping – Used beautiful soup library to scrape html pages of spacex website to get data.
- The collected data was in a table format with relevant information

Data Collection – SpaceX API

- Github: https://github.com/imti03/Data_Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Now let's start requesting rocket launch data from SpaceX API with the following URL:

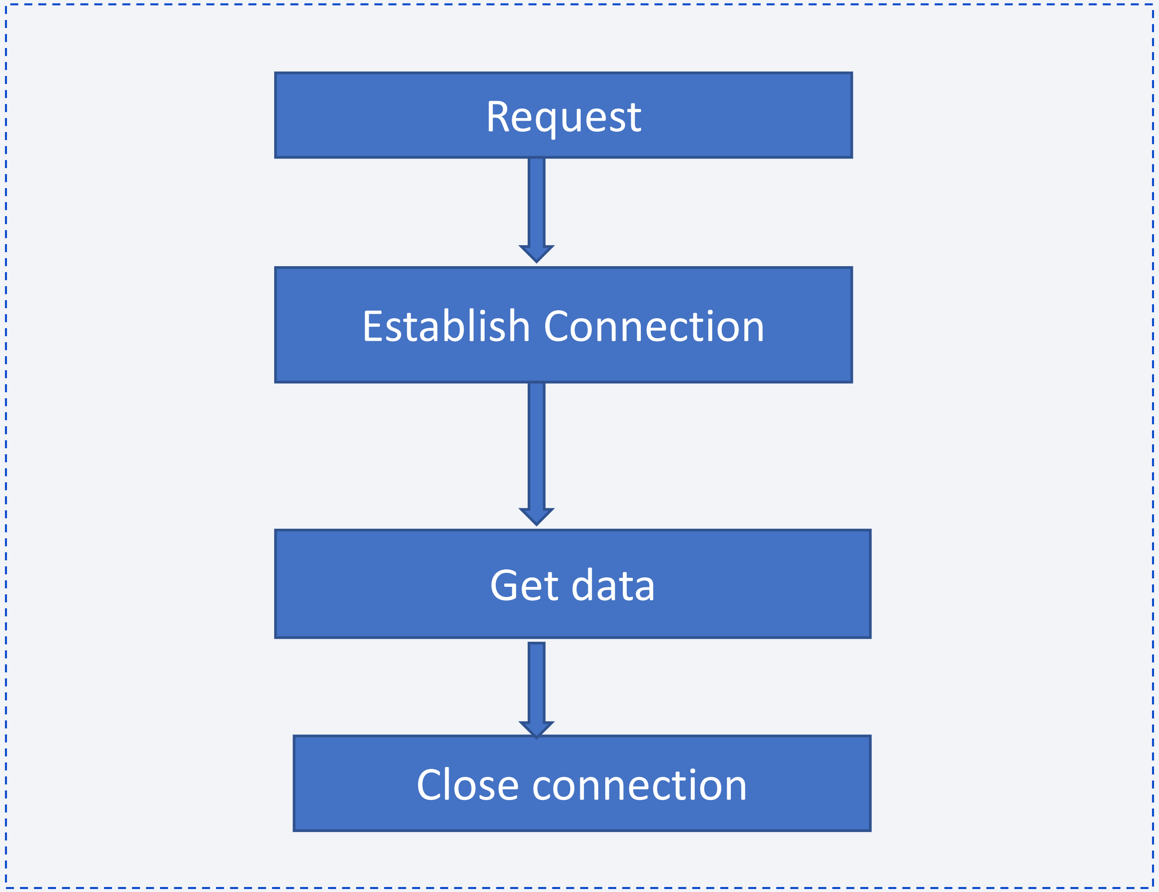
```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
In [7]: response = requests.get(spacex_url)
```

Check the content of the response

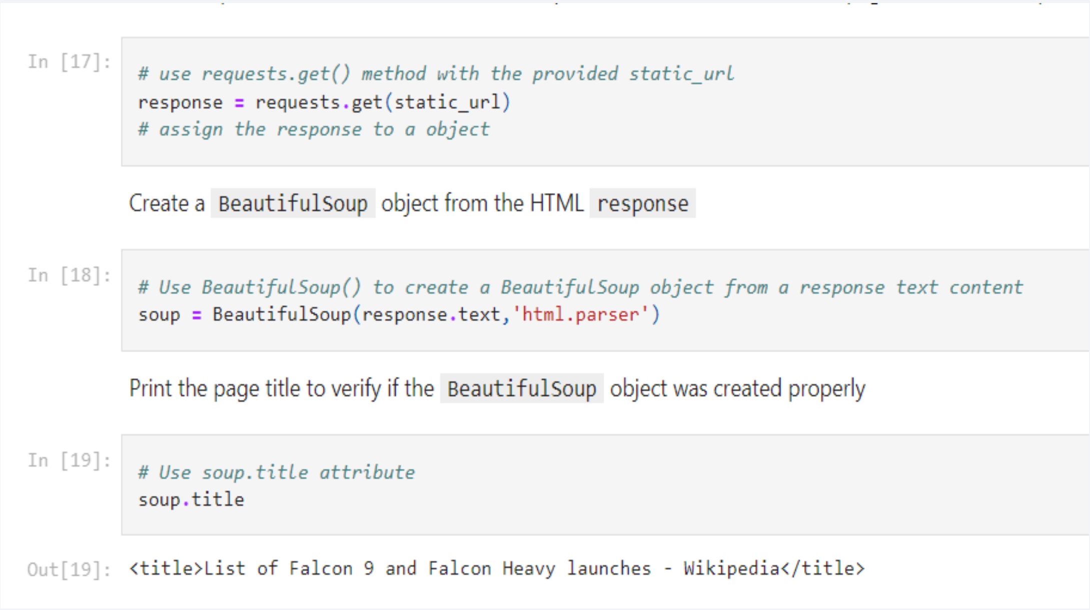
```
In [8]: print(response.content)
```

```
b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":
s2.imgbox.com/94/f2/NN6Ph45r_o.png","large":"https://images2.imgbox.com/5b/02/QcxHUb5V_o.png"},"reddit":{"campai
ch":null,"media":null,"recovery":null},"flickr":{"small":[],"original":[]},"presskit":null,"webcast":"https://www
watch?v=0a_00nJ_Y88","youtube_id":"0a_00nJ_Y88","article":"https://www.space.com/2196-spacex-inaugural-falcon-1-
unch.html","wikipedia":"https://en.wikipedia.org/wiki/DemoSat"},"static_fire_date_utc":"2006-03-17T00:00:00.000Z
_date_unix":1142553600,"net":false,"window":0,"rocket":"5e9d0d95eda69955f709d1eb","success":false,"failures":[{"
tude":null,"reason":"merlin engine failure"}],"details":"Engine failure at 33 seconds and loss of vehicle","crew
[],"capsules":[],"payloads":["5eb0e4b5b6c3bb0006eeb1e1"],"launchpad":"5e9e4502f5090995de566f86","flight_number":
onSat","date_utc":"2006-03-24T22:30:00.000Z","date_unix":1143239400,"date_local":"2006-03-25T10:30:00+12:00","da
n":"hour","upcoming":false,"cores":[{"core":"5e9e289df35918033d3b2623","flight":1,"gridfins":false,"legs":false,
e,"landing_attempt":false,"landing_success":null,"landing_type":null,"landpad":null}],"auto_update":true,"tbd":f
ibrary_id":null,"id":"5eb87cd9ffd86e000604b32a"},{"fairings":{"reused":false,"recovery_attempt":false,"recovered
s":[]},"links":{"patch":{"small":"https://images2.imgbox.com/f9/4a/ZboXReNb_o.png","large":"https://images2.imgb
kWotCIS_o.png"},"reddit":{"campaign":null,"launch":null,"media":null,"recovery":null},"flickr":{"small":[],"orig
```



Data Collection - Scraping

- Github: https://github.com/imti03/Data_Capstone/blob/main/jupyter-labs-webscraping.ipynb



```
In [17]: # use requests.get() method with the provided static_url
         response = requests.get(static_url)
         # assign the response to a object
```

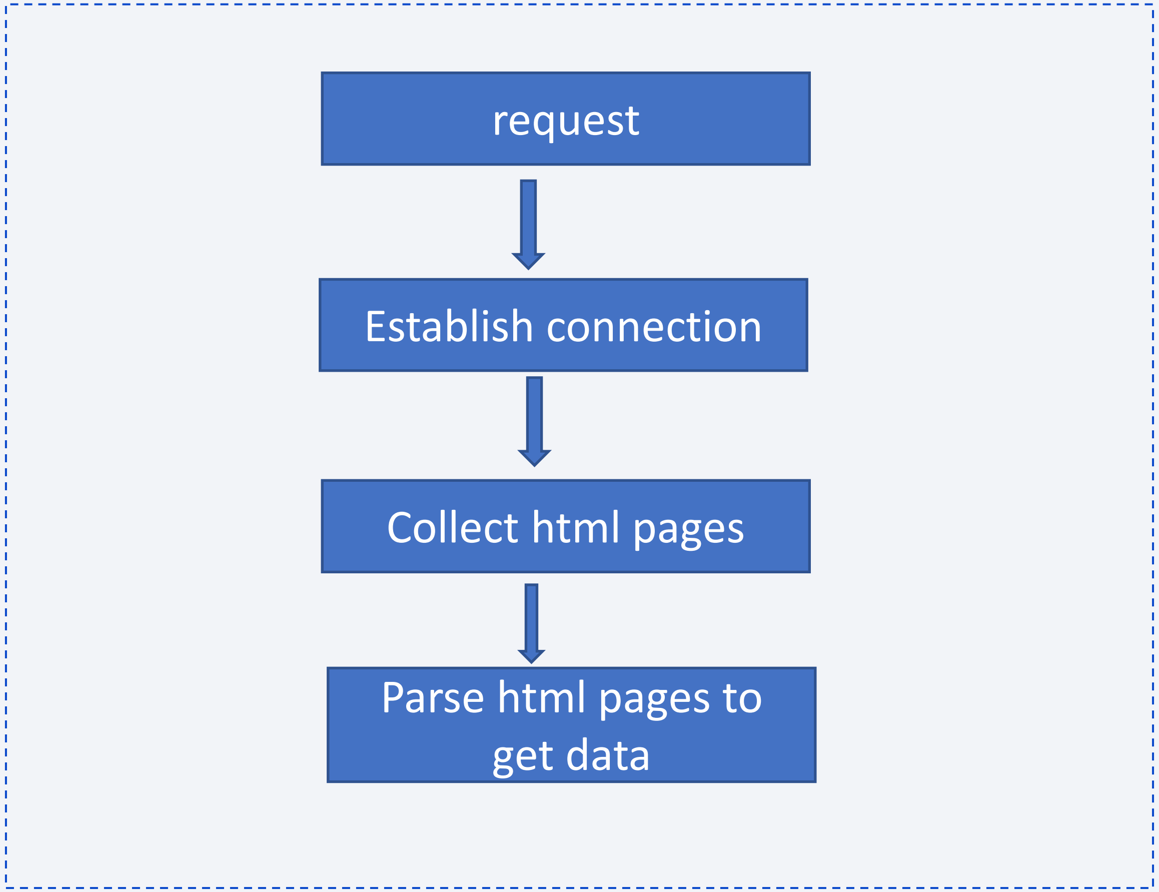
Create a `BeautifulSoup` object from the HTML `response`

```
In [18]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
         soup = BeautifulSoup(response.text,'html.parser')
```

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

```
In [19]: # Use soup.title attribute
         soup.title

Out[19]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```



request

Establish connection

Collect html pages

Parse html pages to get data

Data Wrangling

- The first step was to clean the data, null values from a column was replaced with its mean.
- The class feature was created for classification, success – 1 ,failure – 0.
- https://github.com/imti03/Data_Capstone/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_1_L3_labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

EDA with Data Visualization

- Plots used for key viz:
  - Scatter plot – Payload weight VS Launchpad
  - Bar plot – Launchpad VS Outcome
  - Line graph – Year VS Outcome
- https://github.com/imti03/Data_Capstone/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- Key Queries Used:
  - %%sql SELECT MIN(Date) FROM SPACEXTABLE WHERE Landing_Outcome LIKE '%Ground%' – To find first successful landing.
  - %sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ IN (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE) – Boosters that carried max payload
- https://github.com/imti03/Data_Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Added circles for launchpads in map and markers to mark it
- This was to identify launchpad locations
- https://github.com/imti03/Data_Capstone/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

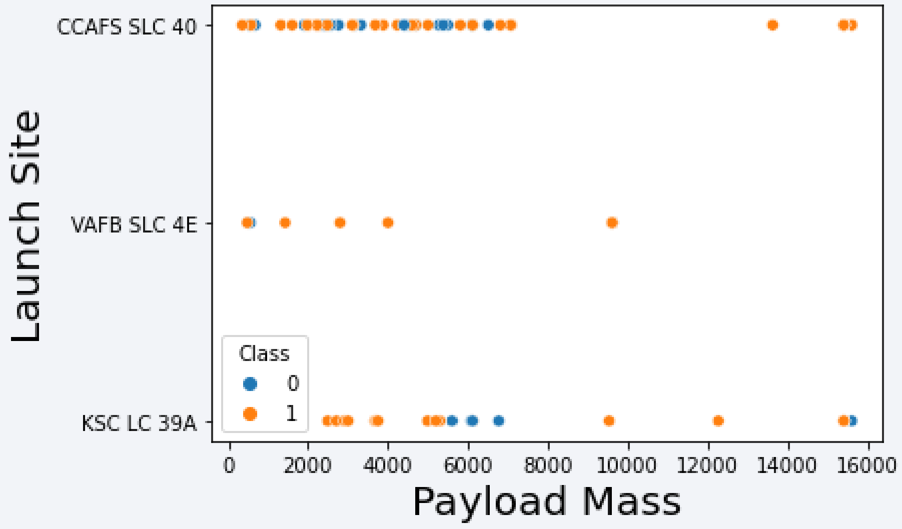
Build a Dashboard with Plotly Dash

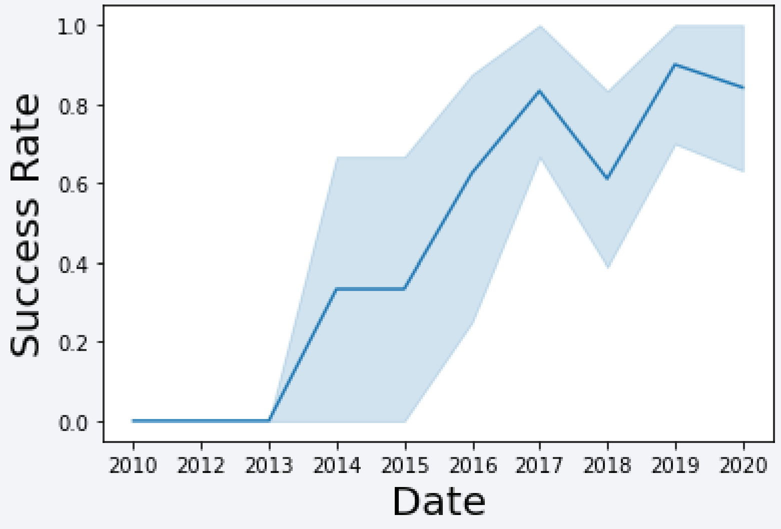
- Added pie chart and scatter plot to dashboard
- To show the success rate and payload correlation to success rate
- https://github.com/imti03/Data_Capstone/blob/main/spacex_dash_app.pys

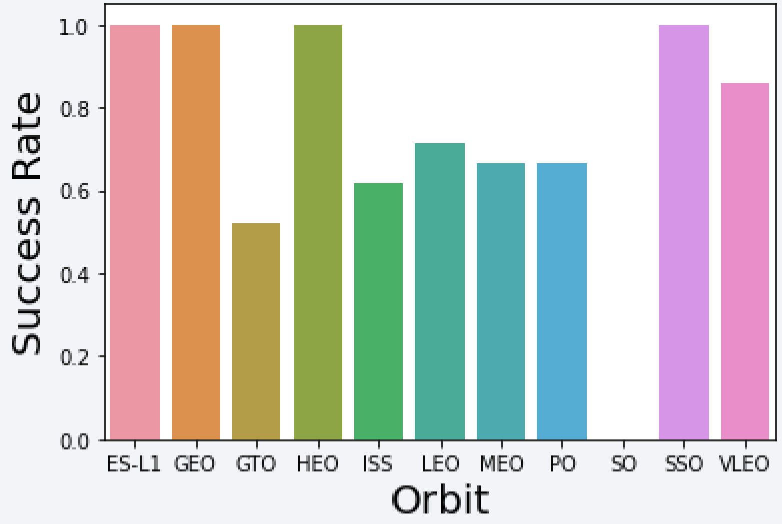
Predictive Analysis (Classification)

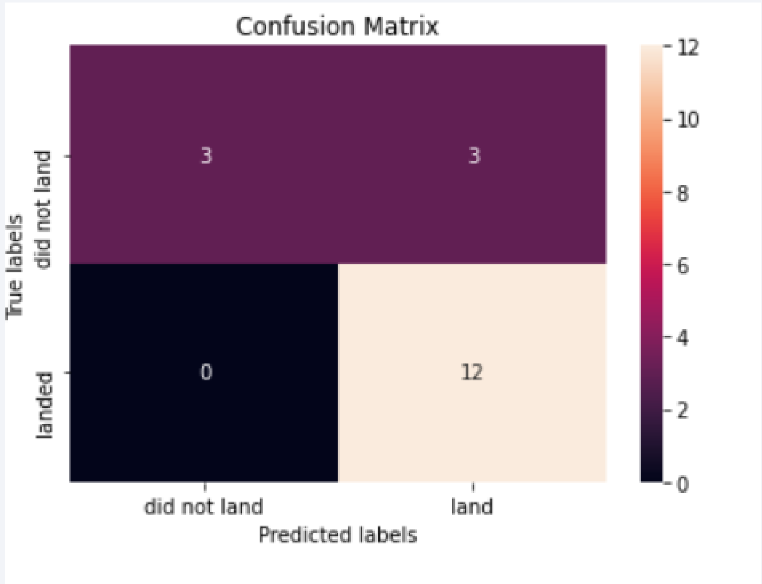
- We built 4 predictive models – Logistic regression, KNN, SVM, decision tree – among which decision tree had higher accuracy.
- The data was split into train and test sets and was transformed by standard scaler, then used grid search cv to use different params and models.
- https://github.com/imti03/Data_Capstone/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results


Launch Site
CCAFS SLC 40
VAFB SLC 4E
KSC LC 39A
Class
● 0
● 1
0
2000
4000
6000
8000
10000
12000
14000
16000
Payload Mass


Success Rate
1.0
0.8
0.6
0.4
0.2
0.0
2010
2012
2013
2014
2015
2016
2017
2018
2019
2020
Date


Success Rate
1.0
0.8
0.6
0.4
0.2
0.0
ES-L1
GEO
GTO
HEO
ISS
LEO
MEO
PO
SO
SSO
VLEO
Orbit

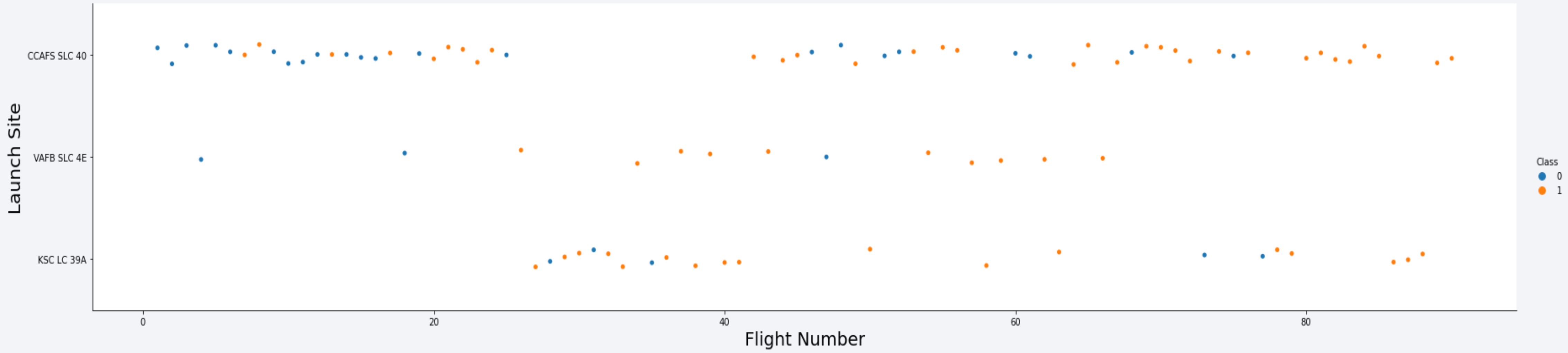

Confusion Matrix
True labels
did not land
landed
3
3
0
12
did not land
land
Predicted labels
12
10
8
6
4
2
0

Section 2

Insights drawn from EDA

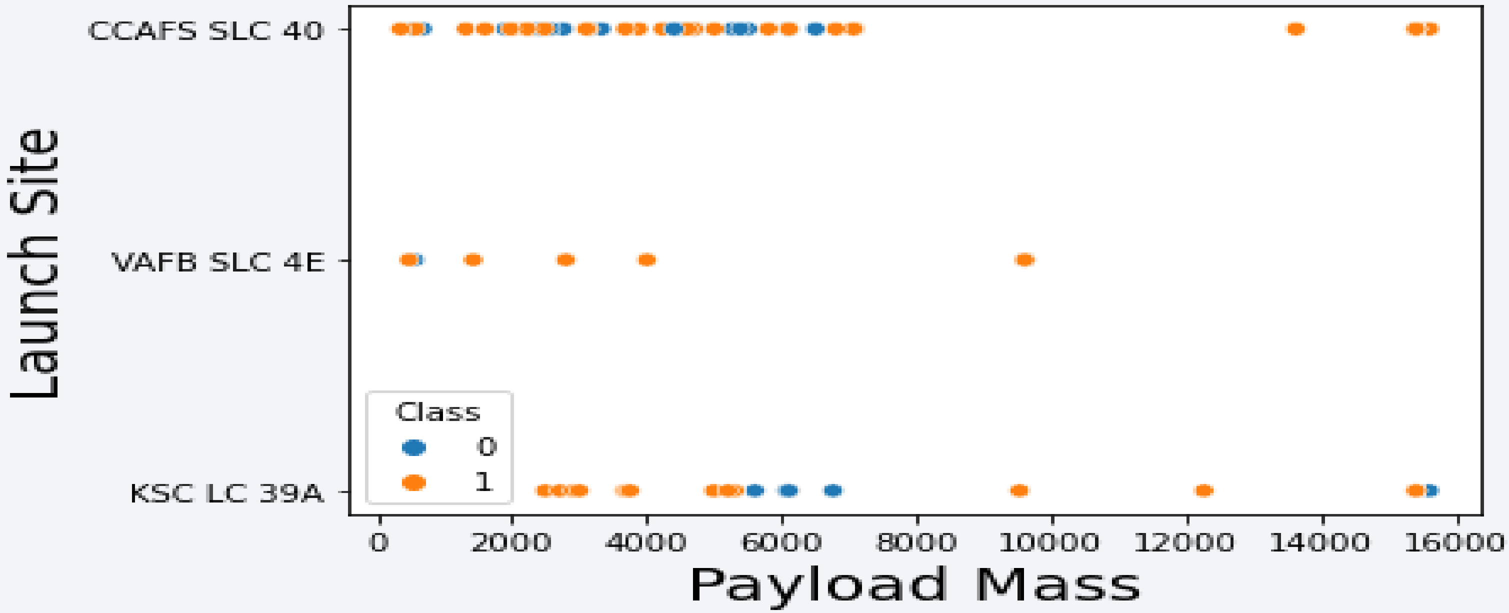
Flight Number vs. Launch Site

- scatter plot of Flight Number vs. Launch Site



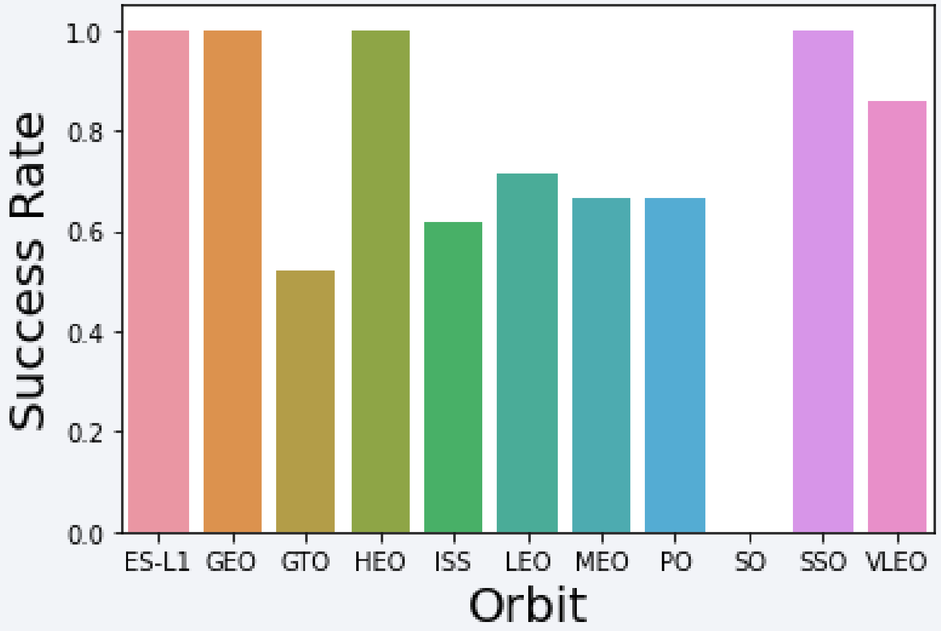
Payload vs. Launch Site

- scatter plot of Payload vs. Launch Site



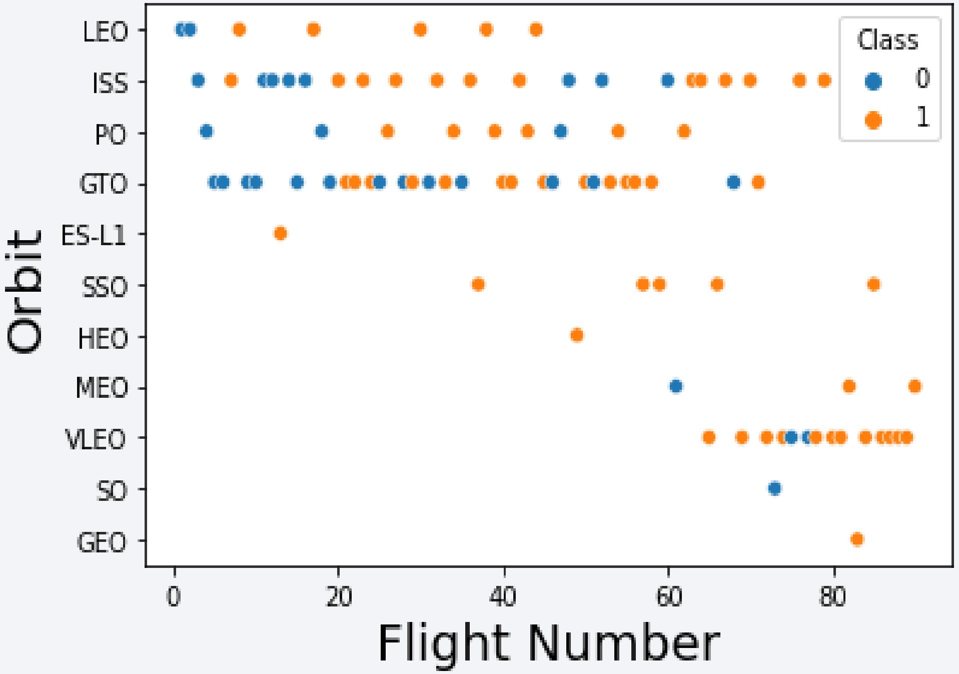
Success Rate vs. Orbit Type

- bar chart for the success rate of each orbit type



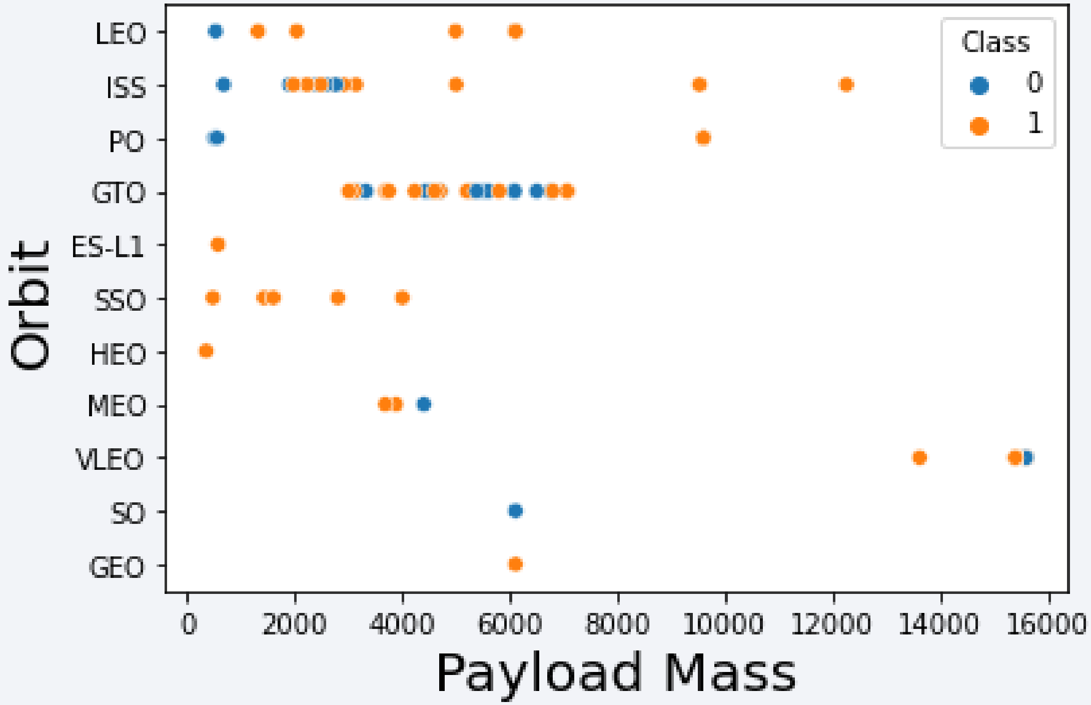
Flight Number vs. Orbit Type

- scatter point of Flight number vs. Orbit type



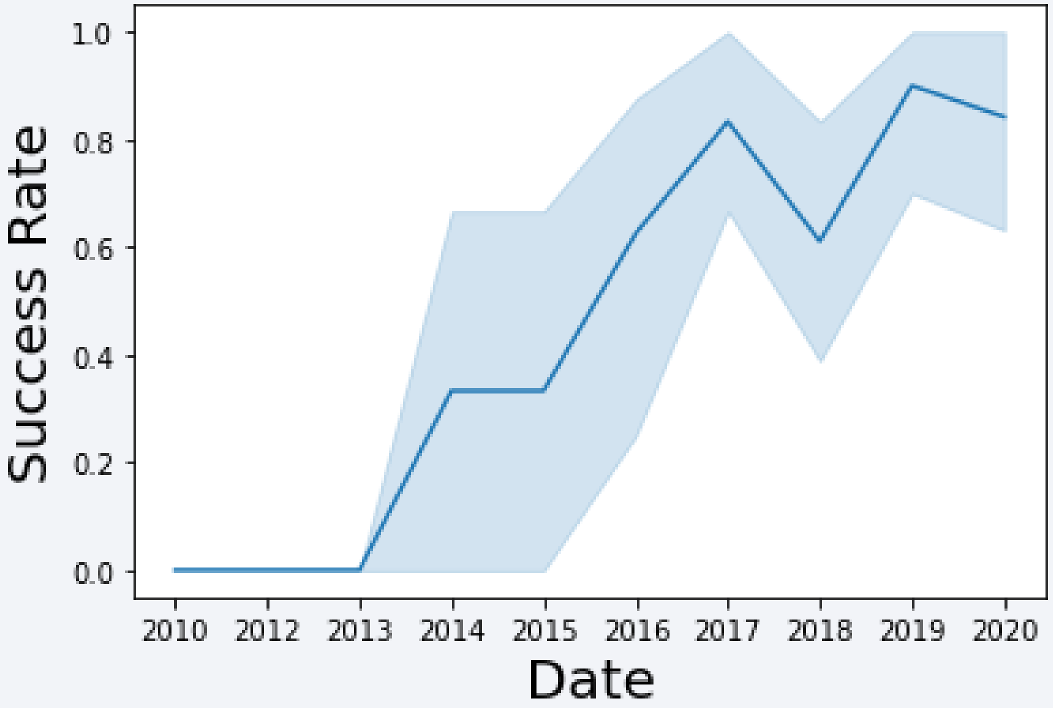
Payload vs. Orbit Type

- scatter point of payload vs. orbit type



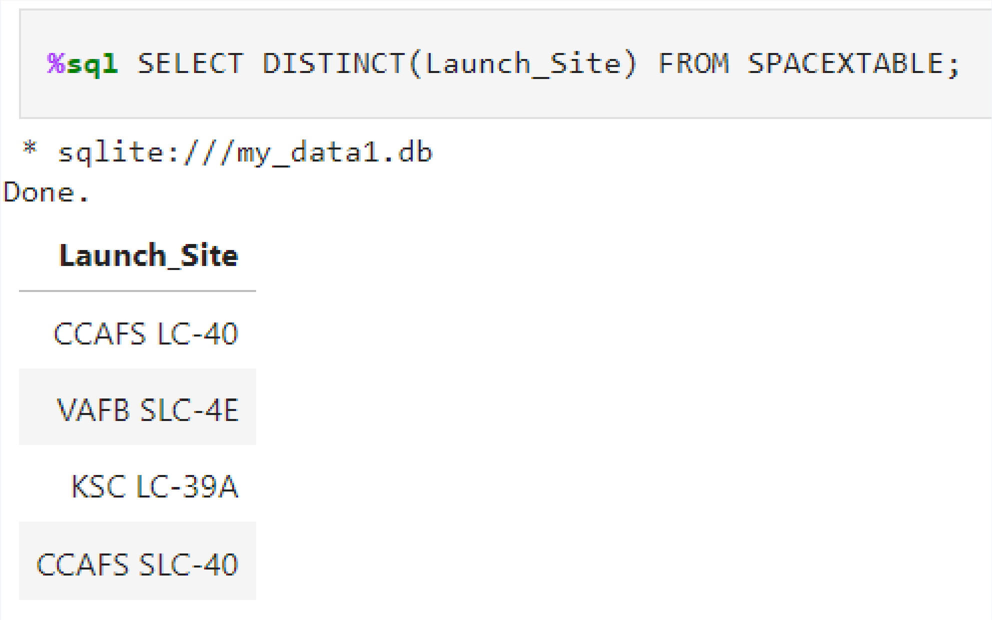
Launch Success Yearly Trend

- line chart of yearly average success rate



All Launch Site Names

- names of the unique launch sites



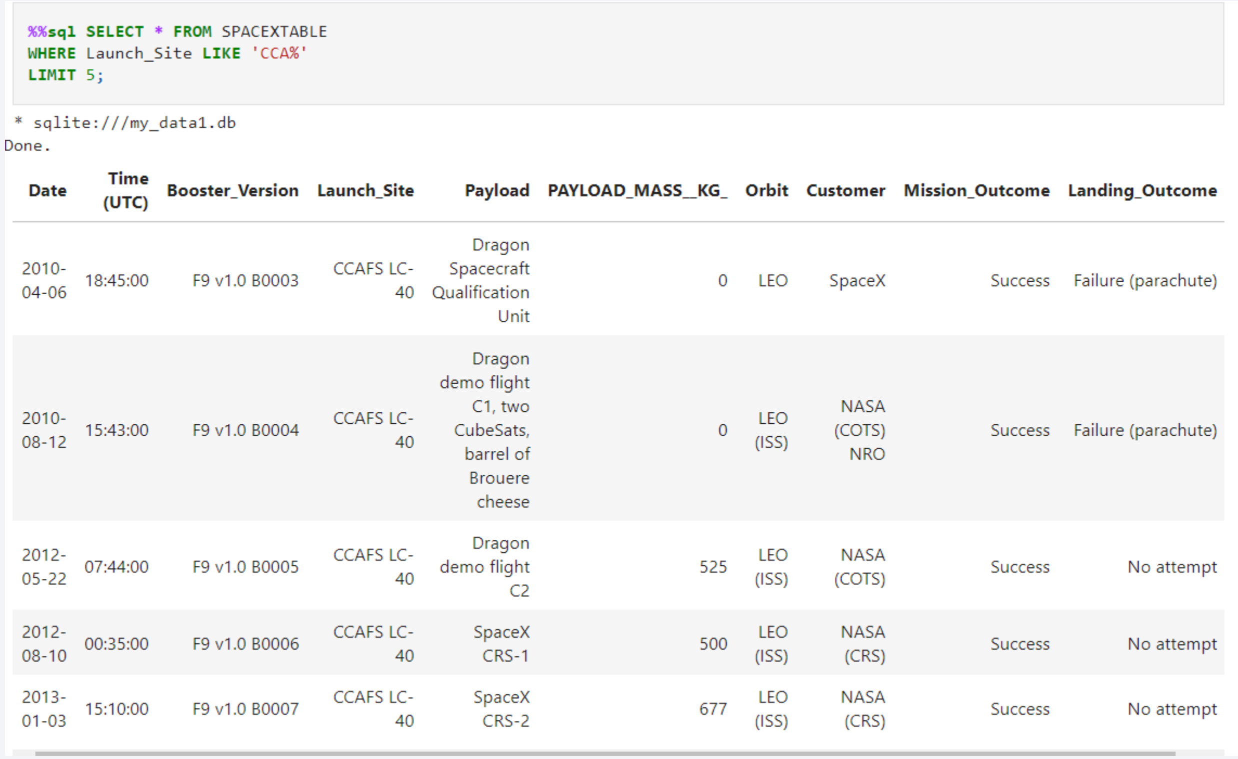
```
%sql SELECT DISTINCT(Launch_Site) FROM SPACEXTABLE;
```

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

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

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`



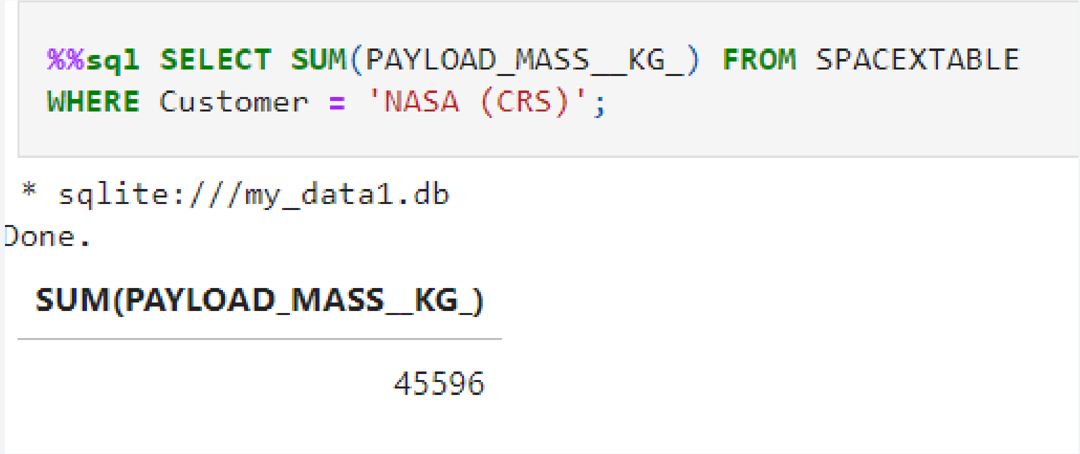
```
%%sql SELECT * FROM SPACEXTABLE
WHERE Launch_Site LIKE 'CCA%'
LIMIT 5;
```

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

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

Total Payload Mass

- total payload carried by boosters from NASA



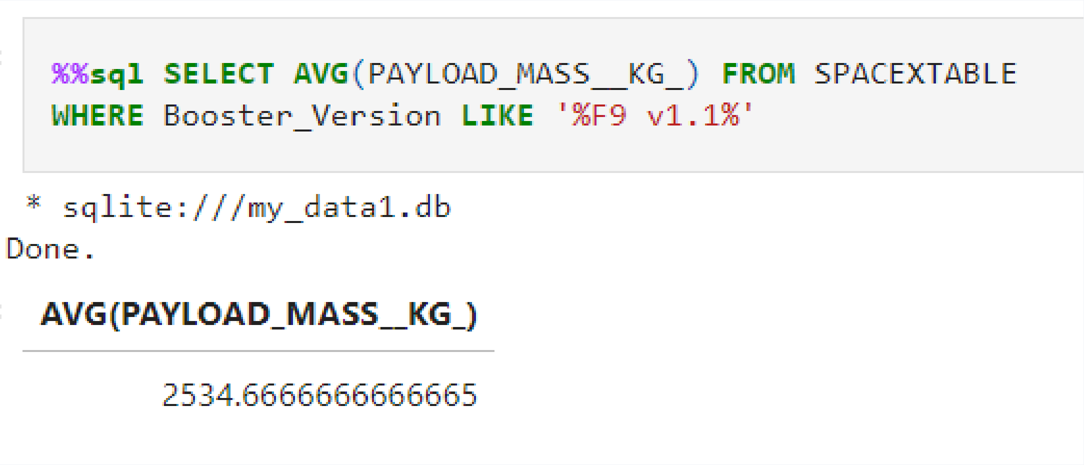
```
%%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE
WHERE Customer = 'NASA (CRS)';
```

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

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



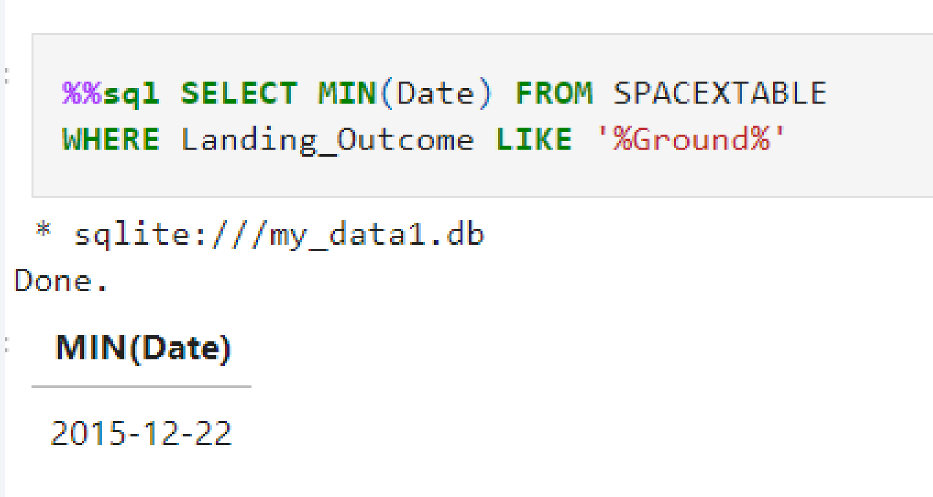
```
%%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE
WHERE Booster_Version LIKE '%F9 v1.1%'
```

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

The date of the first successful landing outcome on ground pad



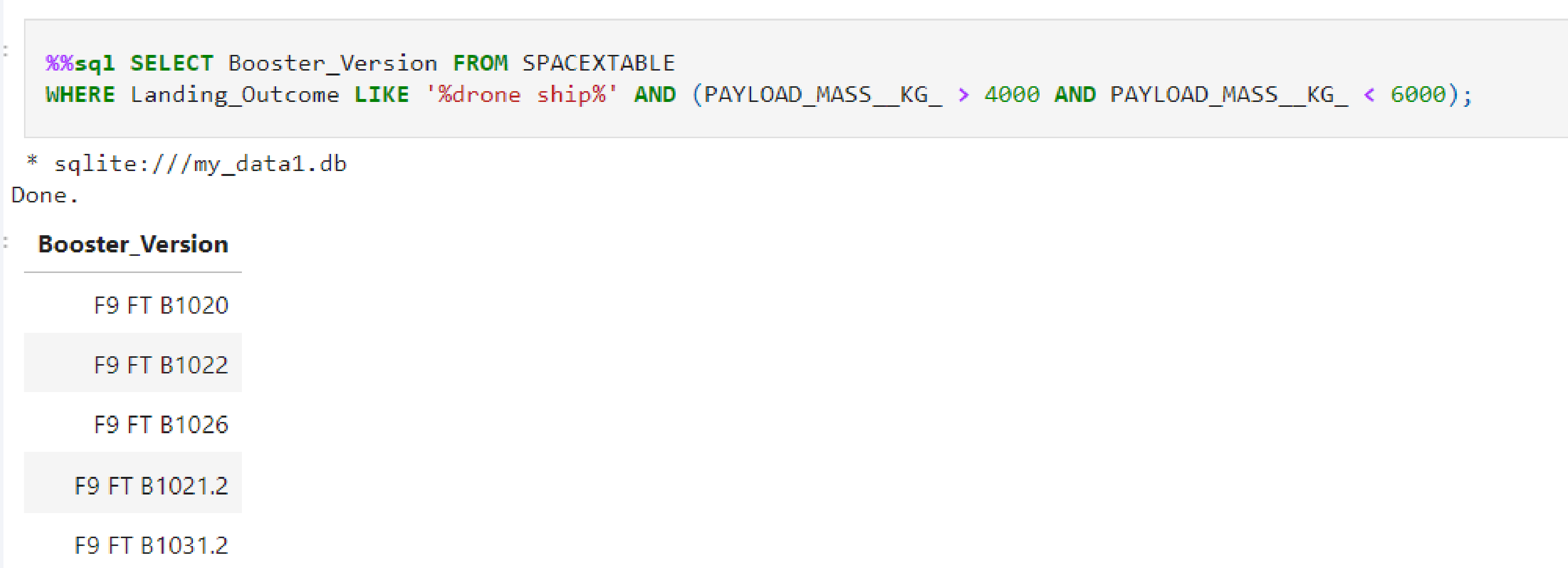
```
%%sql SELECT MIN(Date) FROM SPACEXTABLE
WHERE Landing_Outcome LIKE '%Ground%'
```

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

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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



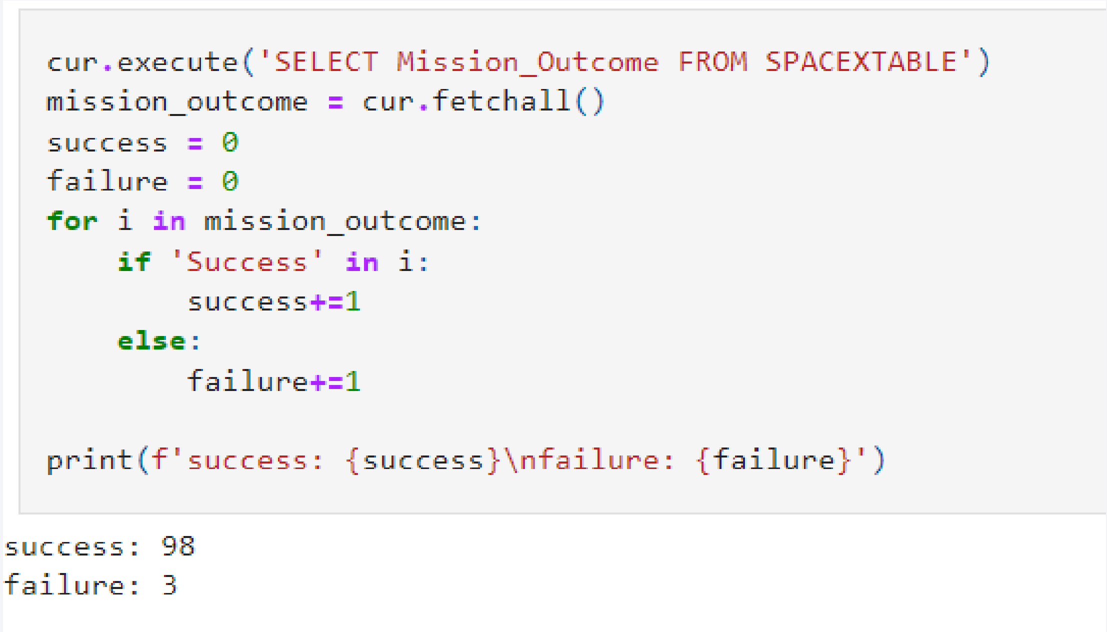
```
%%sql SELECT Booster_Version FROM SPACEXTABLE
WHERE Landing_Outcome LIKE '%drone ship%' AND (PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000);
```

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

Booster_Version
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failure mission outcomes



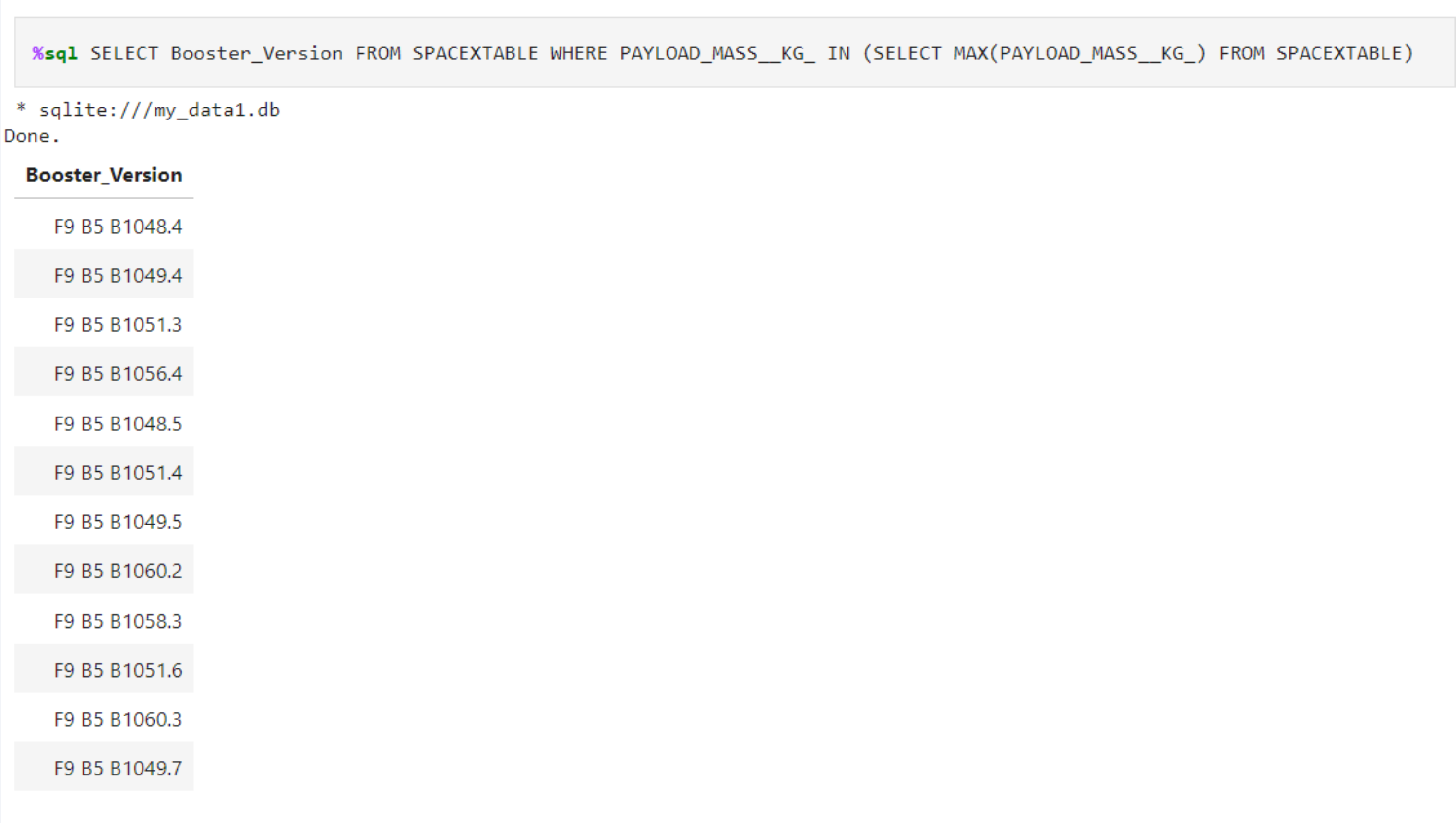
```
cur.execute('SELECT Mission_Outcome FROM SPACEXTABLE')
mission_outcome = cur.fetchall()
success = 0
failure = 0
for i in mission_outcome:
    if 'Success' in i:
        success+=1
    else:
        failure+=1

print(f'success: {success}\nfailure: {failure}')
```

```
success: 98
failure: 3
```

Boosters Carried Maximum Payload

- Names of the booster which have carried the maximum payload mass



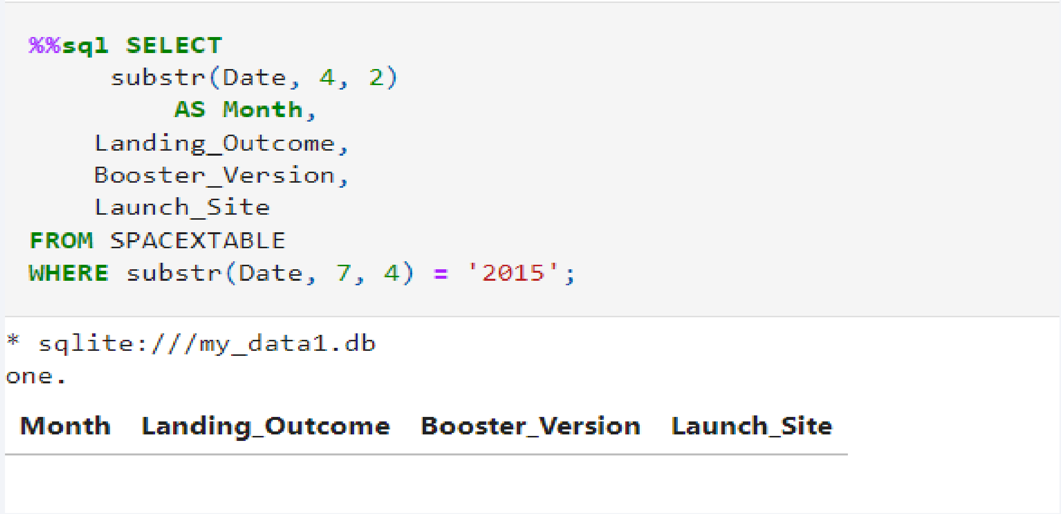
```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ IN (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

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

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

- Failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015



```
%%sql SELECT
     substr(Date, 4, 2)
         AS Month,
    Landing_Outcome,
    Booster_Version,
    Launch_Site
FROM SPACEXTABLE
WHERE substr(Date, 7, 4) = '2015';
```

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

Month	Landing_Outcome	Booster_Version	Launch_Site

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

Section 3

Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

- Replace <Folium map screenshot 1> title with an appropriate title

- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map

- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 2>

- Replace <Folium map screenshot 2> title with an appropriate title
- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map
- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 3>

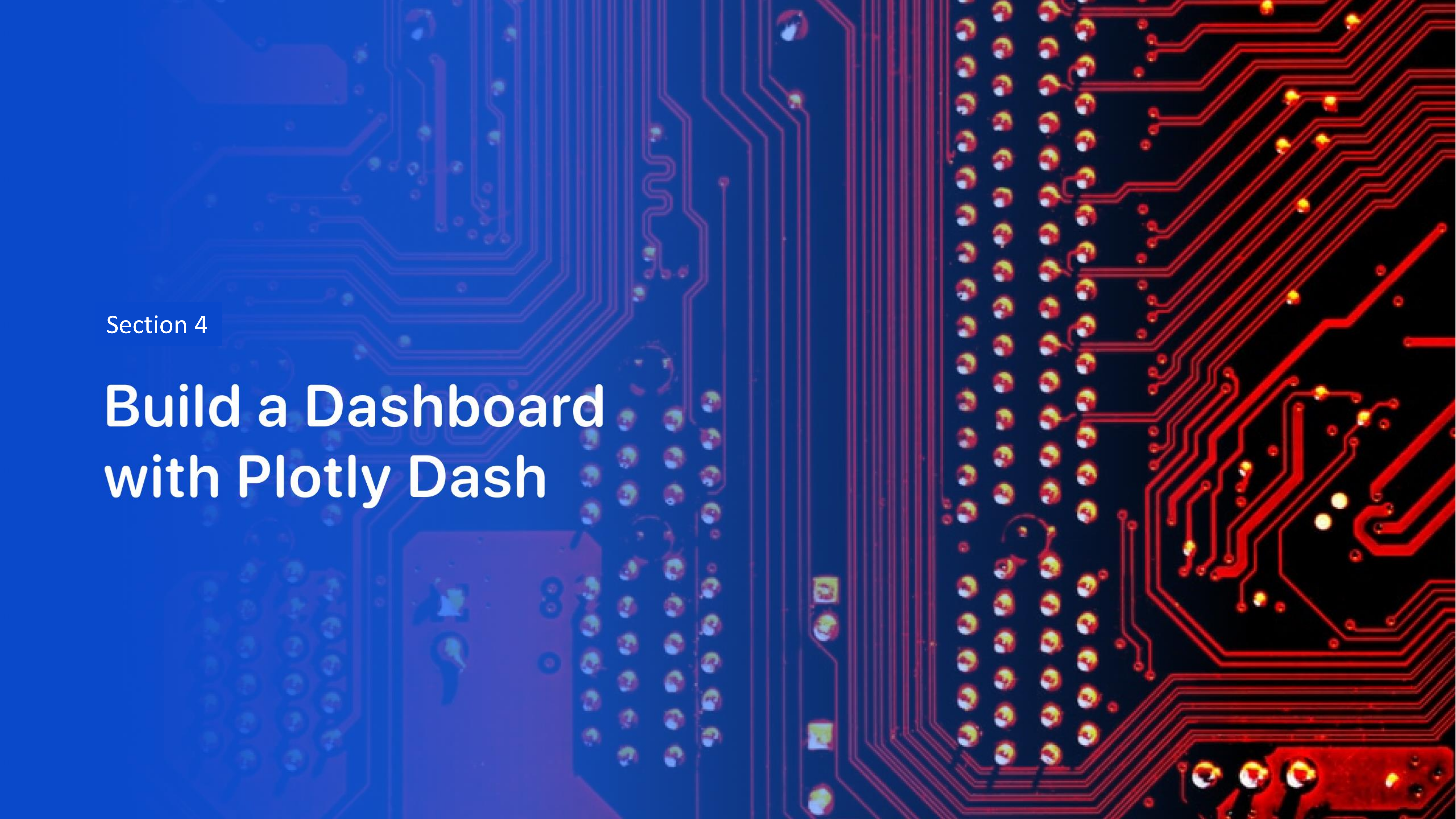
- Replace <Folium map screenshot 3> title with an appropriate title

- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

- Explain the important elements and findings on the screenshot

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

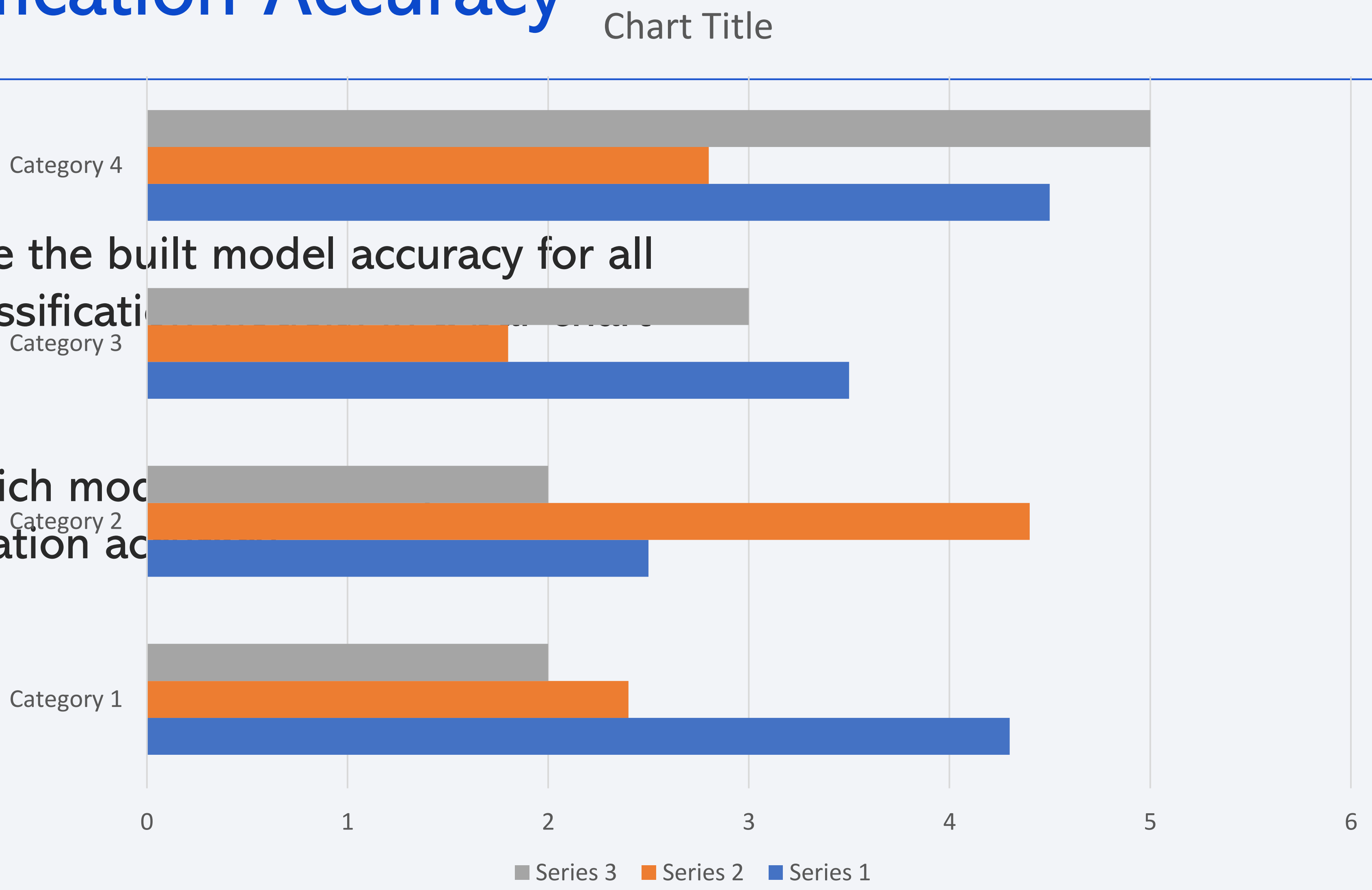
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section 5

Predictive Analysis (Classification)

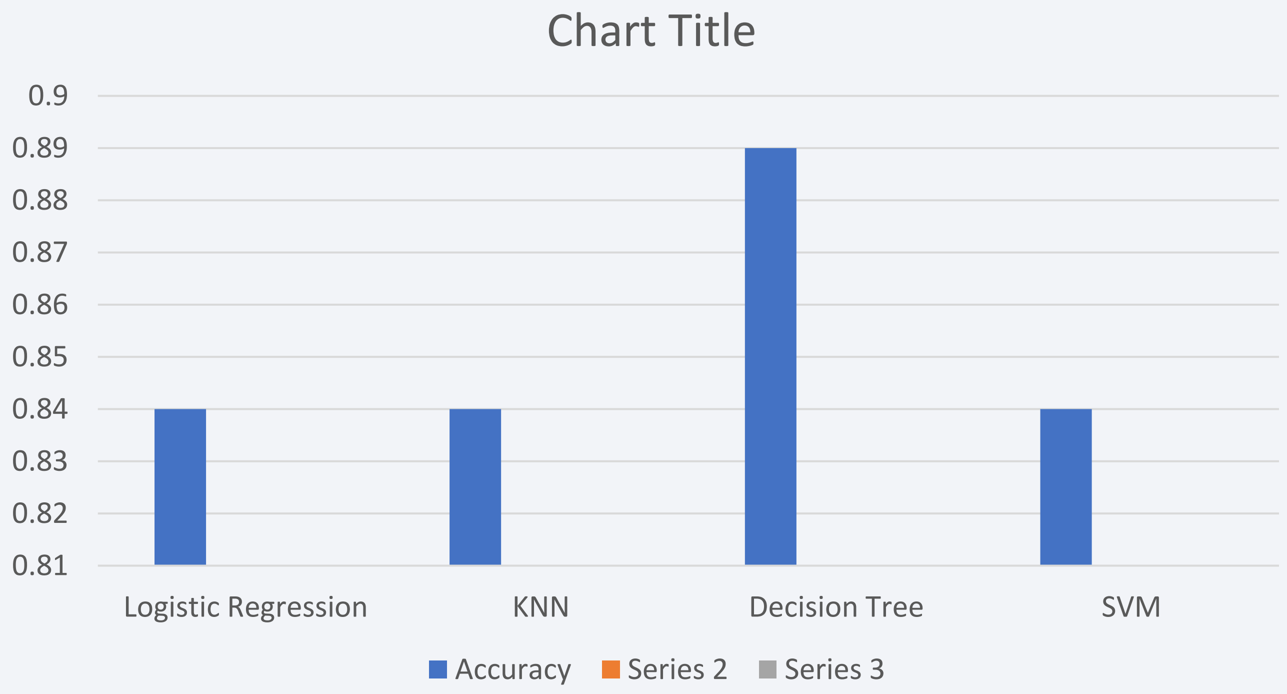
Classification Accuracy

- Visualize the built model accuracy for all built classificati
- Find which moc classification ac



Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation



Conclusions

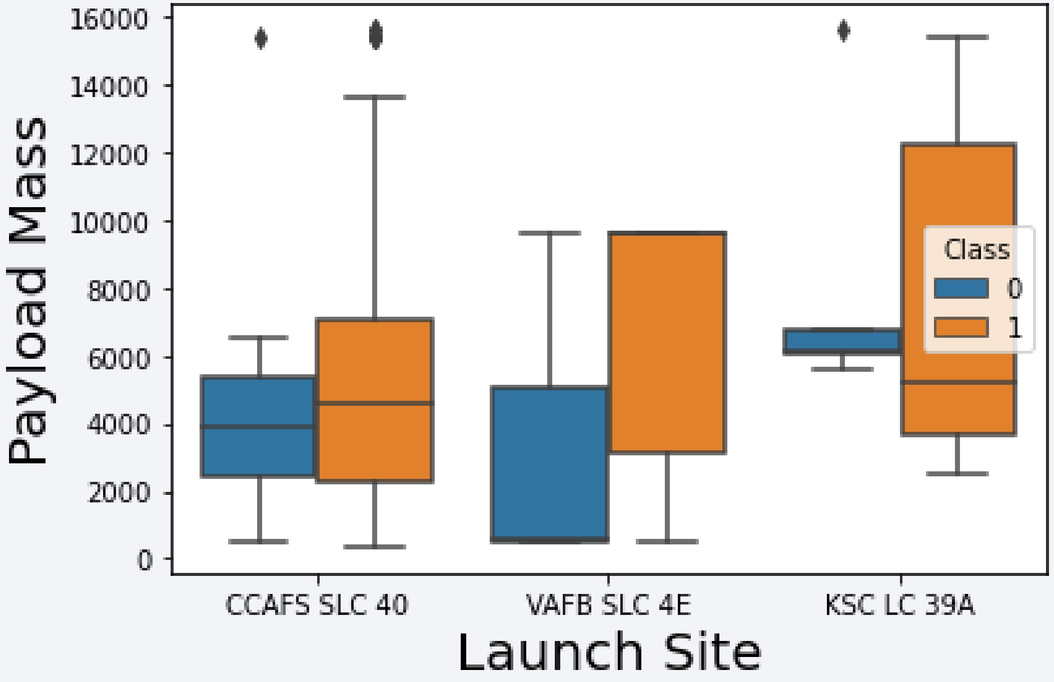
- The four models had the same testing accuracy
- The four models had identical confusion matrix values
- Only decision tree had higher training accuracy of 89%
- All models performed well on the given data.

Appendix

KSC launch pad clearly has high success outcomes.

In VAFB launch pad it can be clearly distinguished that higher payload has greater success outcome.

In CCAFS launch pad there are a lot of outliers and the data is pretty much same for both class, so it is hard to distinguish between success and failure.



Thank you!