IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Parth Kharadi
19/04/2023

Outline

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

Executive Summary

1. This analysis is based on SpaceX launch data

2. Here we present some key findings from our analysis and prediction
    - Objective here is to predict if landing will be successful or not for the given launch.

    - This prediction will help figure out cost of launch in advance.

Introduction

- The objective of project is to analyze SpaceX launch data and build model to predict whether landing will be successful or not.
- This prediction can help estimate launch cost and bid for space program

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - We’ve collected SpaceX launch data using REST API and Web Scrapping
- Perform data wrangling
  - Data was cleaned
  - Removed unnecessary data. We removed Falcon 1 data since we’re concerned with Falcon 9
  - Converted some data to categorical data for machine learning. i.e. Converted landing outcome to Class of 0 and 1.
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Using various visualizations and SQL queries performed EDA to identify relationship between various features. i.e How Payload_Mass or Booster_Version can impact landing outcome
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models. i.e Logistic Regression, SVM, KNN, Decision Tree

Data Collection

- SpaceX launch data was collected by using REST API provided by SpaceX
  - This launch data was used to fetch other information about rockets, payloads, Launchpad, cores using individual REST API for each of feature
  - Also, we collect historical data for Falcon 9 launches from Wikipedia using web scrapping We've used Python's BeautifulSoup library for this

Data Collection – SpaceX API

- SpaceX launch data collection using REST API
- GitHub URL to REST API data collection notebook:
  - https://github.com/PHK28/DataScienceCapstoneProject/blob/ffbb8aa4c03730258dcaa3b5fbc697553a2dea84/jupyter-labs-spacex-data-collection-api.ipynb

Place your flowchart of SpaceX API calls here

Data Collection - Scraping

- SpaceX launch data collection using web scrapping
- GitHub URL to web scarpping data collection notebook:
    - https://github.com/PHK28/DataScienceCapstoneProject/blob/ffbb8aa4c03730258dcaa3b5fbc697553a2dea84/jupyter-labs-webscraping.ipynb

Place your flowchart of web scraping here

Data Wrangling

- Cleaned data. Removed null and empty values
- Removed unnecessary data. We removed Falcon 1 data since we’re concerned with Falcon 9
- Converted some data to categorical data for machine learning. i.e. Converted landing outcome to Class of 0 and 1
- GitHub URL: https://github.com/PHK28/DataScienceCapstoneProject/blob/ffbb8aa4c03730258dcaa3b5fbc697553a2dea84/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

EDA with Data Visualization

- Used scatter plot to understand relationship between FlightNumber-Orbit, FlightNumber-Payload, Payload-LaunchSite, Orbit-Payload
- Used Bar chart to find out which orbit has higher chance of successful landing
- Used line chart to find out launch success yearly trend
- GitHub URL: https://github.com/PHK28/DataScienceCapstoneProject/blob/ffbb8aa4c03730258dcaa3b5fbc697553a2dea84/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- Performed various SQL queries to summarize data and to get insight from data
  - Dates when first successful landing on ground pad was achieved
  - Name of the boosters which have success in drone ship and payload mass between 4000-6000
  - Total number of successful and failure outcomes
  - Boosters which carried maximum payload mass
  - Successful landing between given date range
- GitHub URL: https://github.com/PHK28/DataScienceCapstoneProject/blob/ffbb8aa4c03730258dcaa3b5fbc697553a2dea84/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Used folium map to plot launch site locations on map. Used circles, markers and marker clusters
- GitHub URL: https://github.com/PHK28/DataScienceCapstoneProject/blob/ffbb8aa4c03730258dcaa3b5fbc697553a2dea84/lab_jupyter_launch_site_location.jupyterlite.ipynb

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
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification)

- Built various classification models like Logistic Regression, SVM, KNN, Decision Tree
- Used confusion matrix evaluation matrix to figure out performance of model
- Considering which model has highest accuracy we can use that model to predict success rate of landing
- GitHub URL: https://github.com/PHK28/DataScienceCapstoneProject/blob/ffbb8aa4c03730258dcaa3b5fbc697553a2dea84/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite%20(1).ipynb

Results

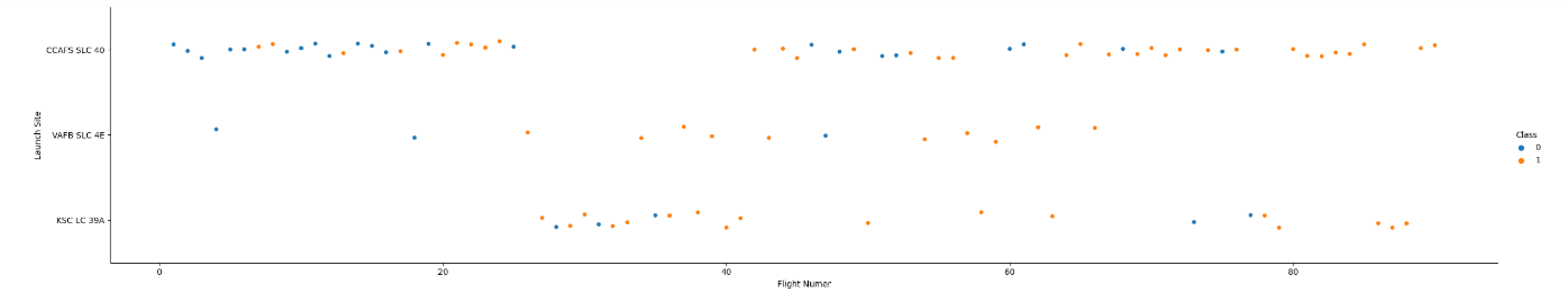
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

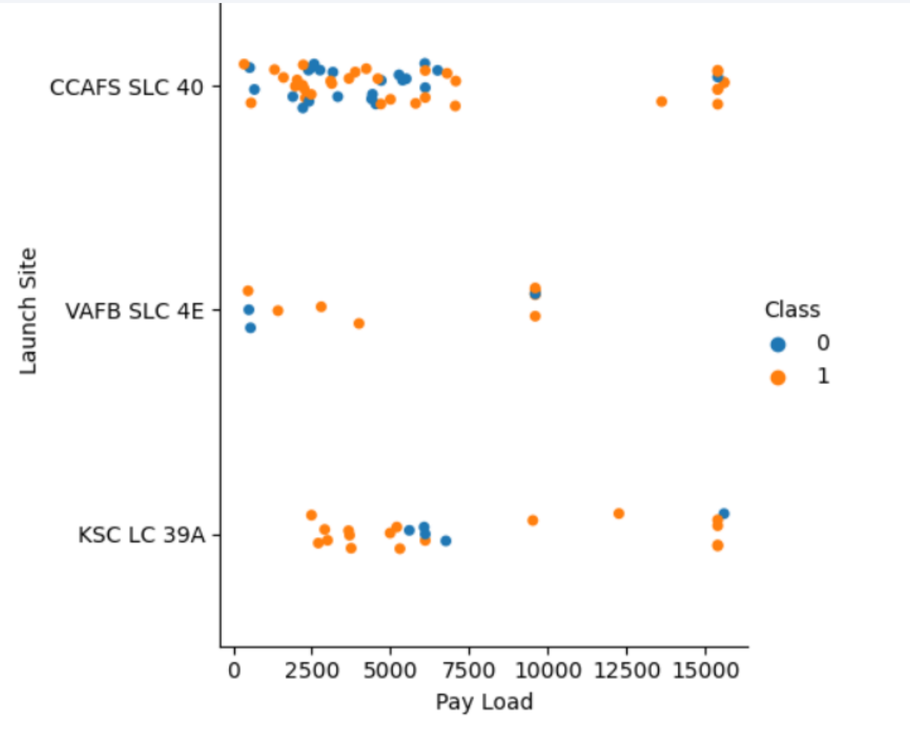
Flight Number vs. Launch Site

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



Payload vs. Launch Site

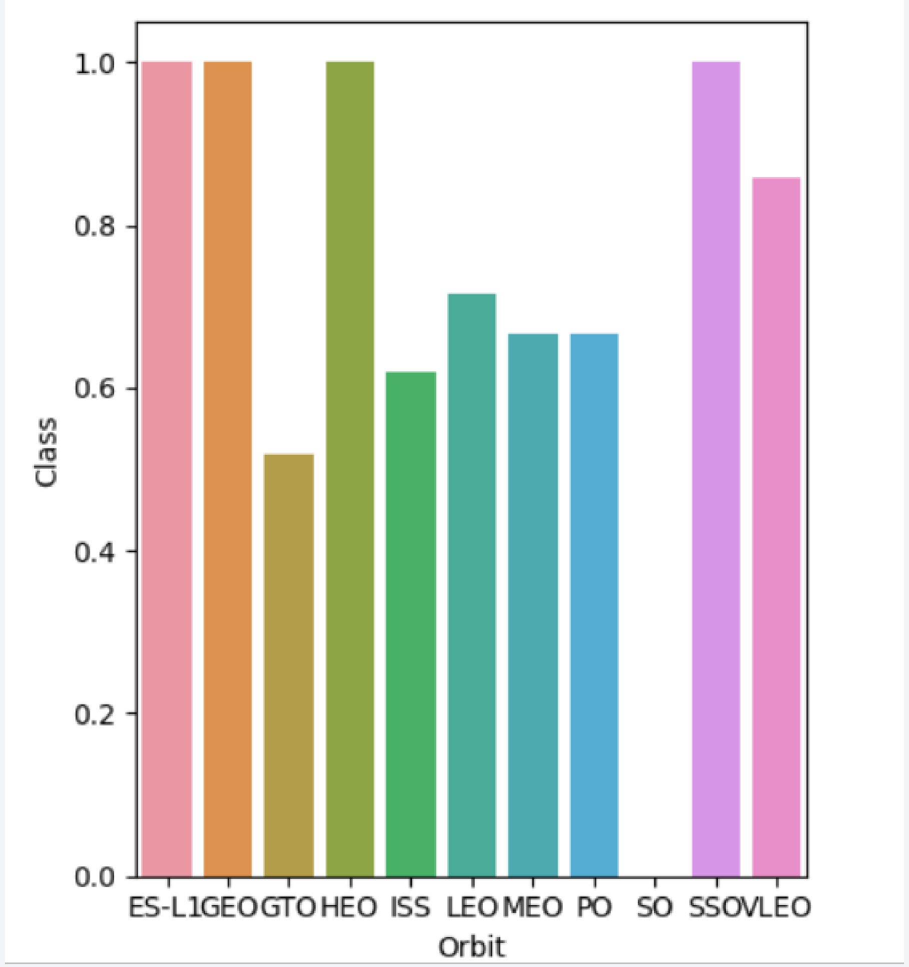
- Show a scatter plot of Payload vs. Launch Site



Success Rate vs. Orbit Type

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

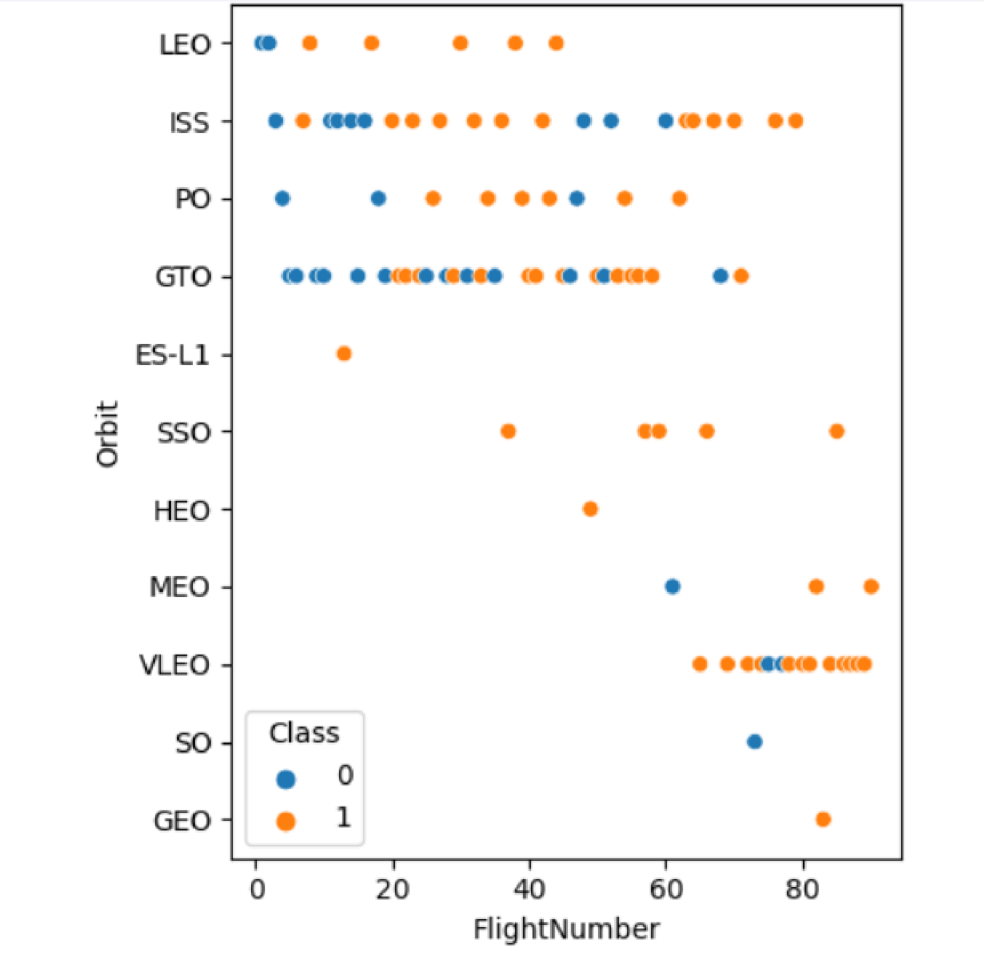
- Show the screenshot of the scatter plot with explanations



Flight Number vs. Orbit Type

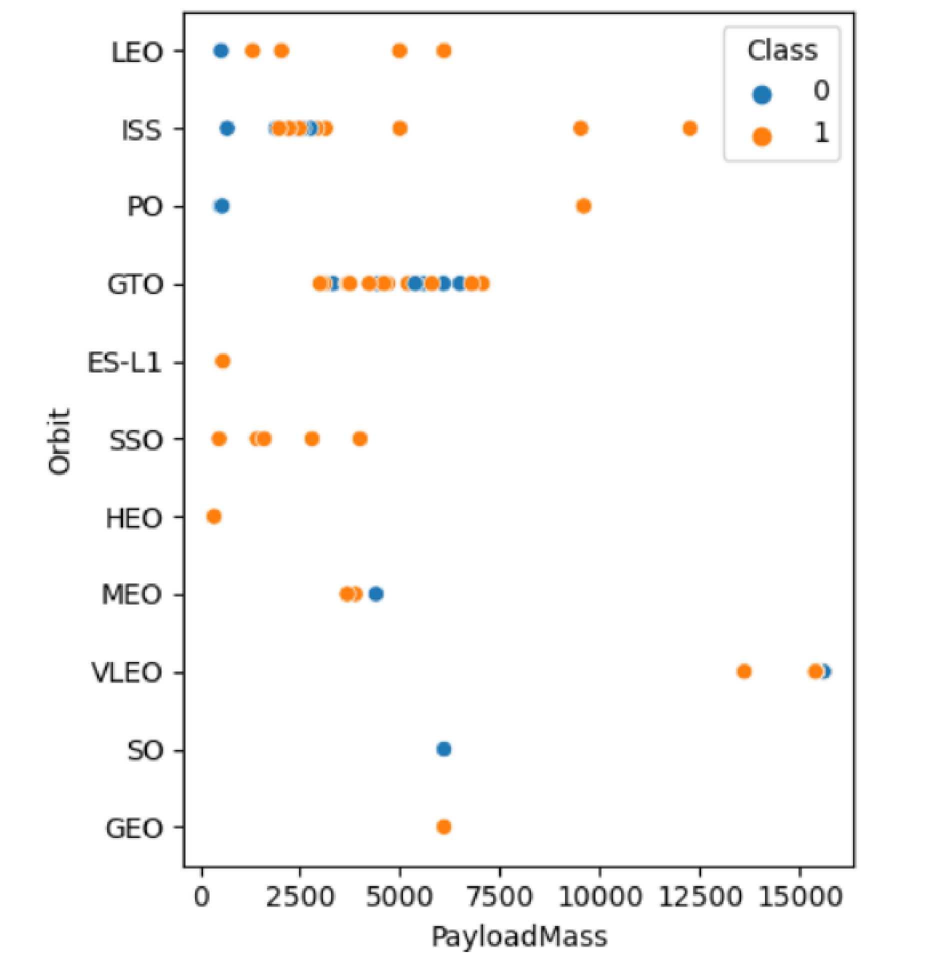
- Show a scatter point of Flight number vs. Orbit type

- Show the screenshot of the scatter plot with explanations



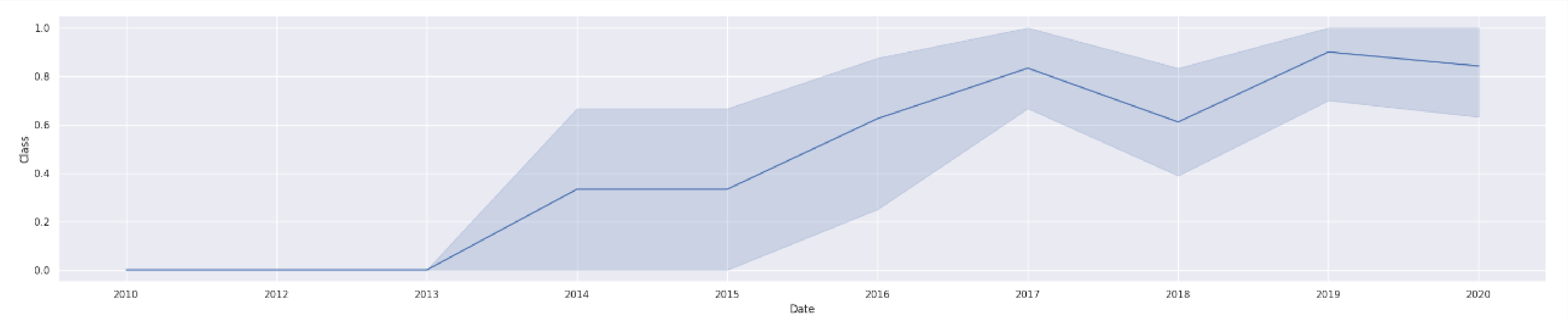
Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type
- Show the screenshot of the scatter plot with explanations



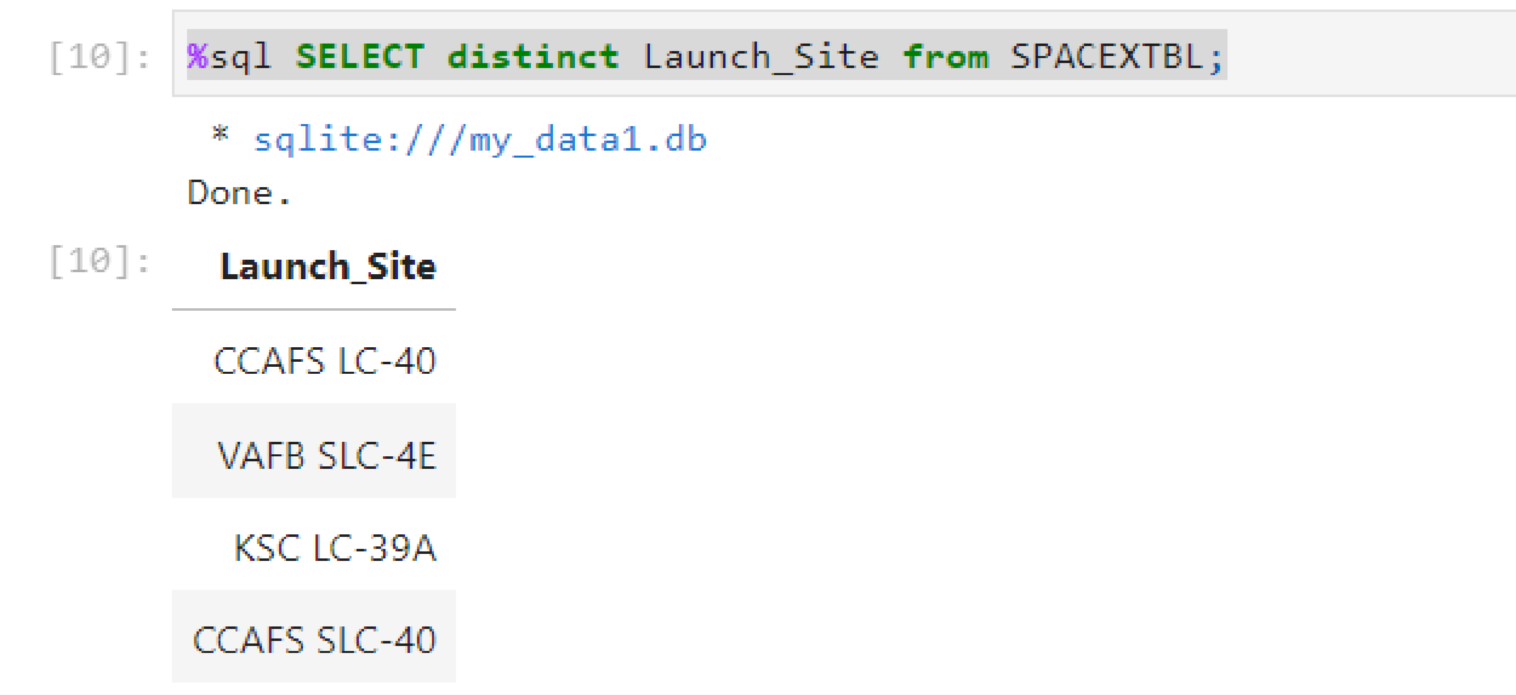
Launch Success Yearly Trend

- Show a line chart of yearly average success rate



All Launch Site Names

- Names of the unique launch sites



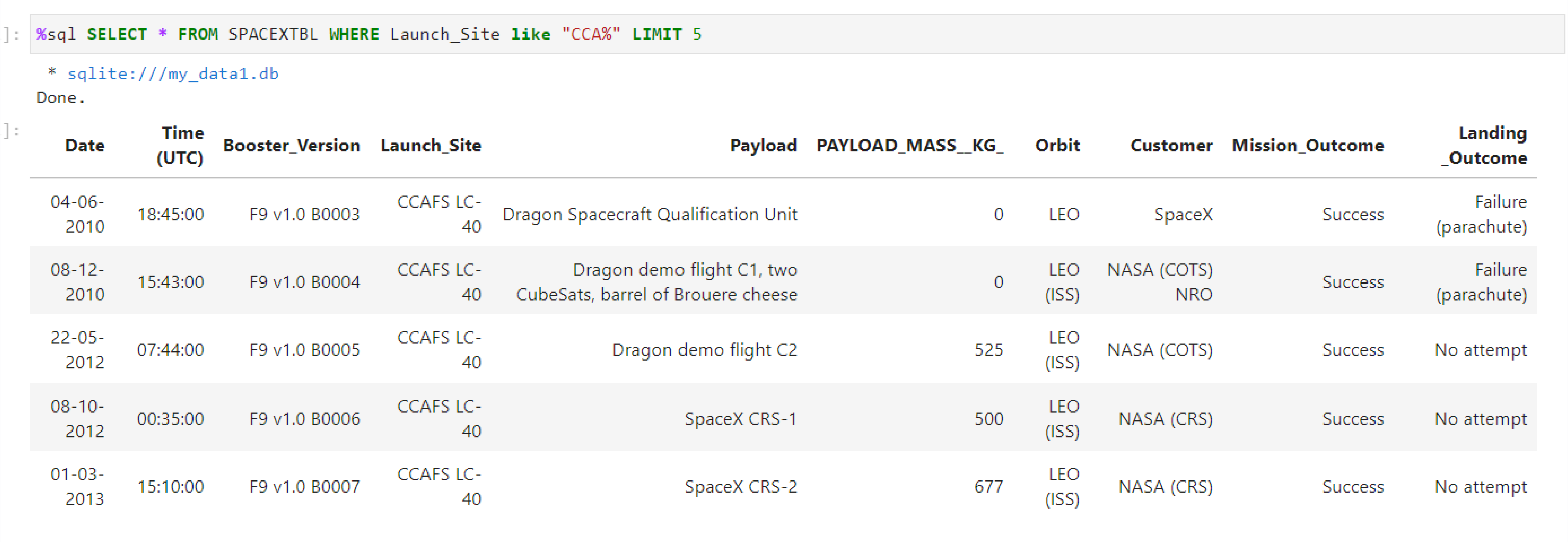
```
[10]: %sql SELECT distinct Launch_Site from SPACEXTBL;

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

[10]:

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

Launch Site Names Begin with 'CCA'

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



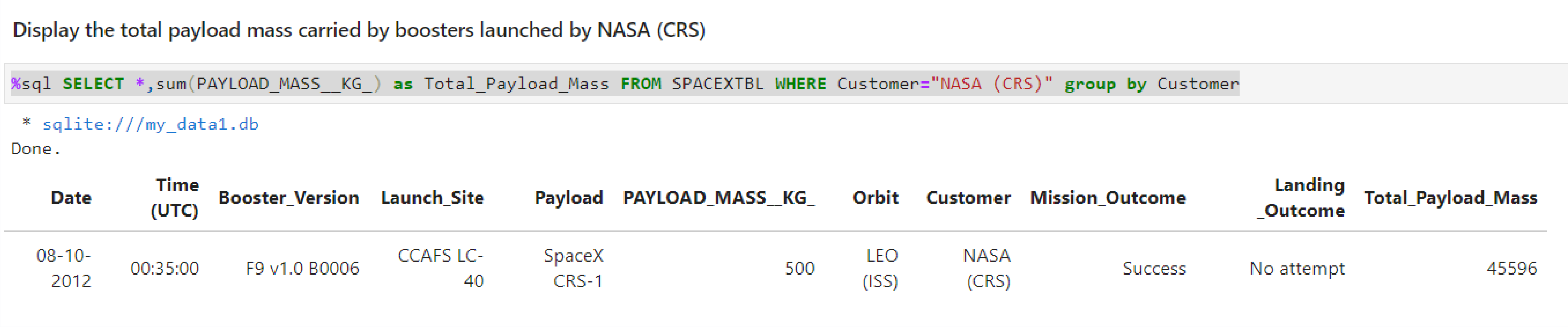
```
]: %sql SELECT * FROM SPACEXTBL WHERE Launch_Site like "CCA%" LIMIT 5
```

```
 * 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

Total Payload Mass

- Calculate the total payload carried by boosters from NASA



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

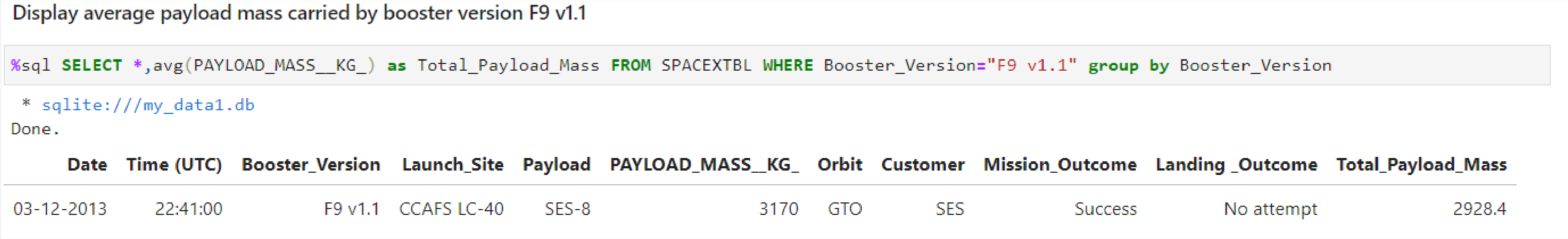
```
%sql SELECT *,sum(PAYLOAD_MASS__KG_) as Total_Payload_Mass FROM SPACEXTBL WHERE Customer="NASA (CRS)" group by Customer
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome	Total_Payload_Mass
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt	45596

Average Payload Mass by F9 v1.1

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



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

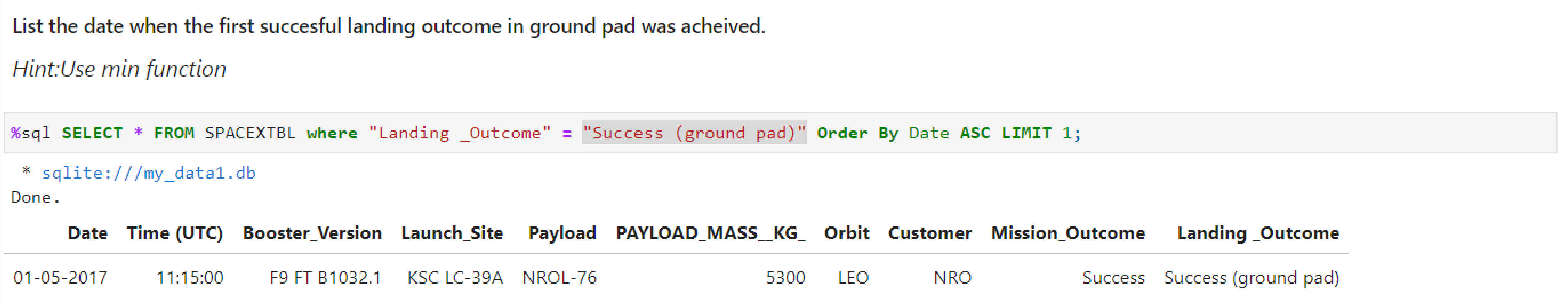
```
%sql SELECT *,avg(PAYLOAD_MASS__KG_) as Total_Payload_Mass FROM SPACEXTBL WHERE Booster_Version="F9 v1.1" group by Booster_Version
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome	Total_Payload_Mass
03-12-2013	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES	Success	No attempt	2928.4

First Successful Ground Landing Date

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



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

Hint:Use min function

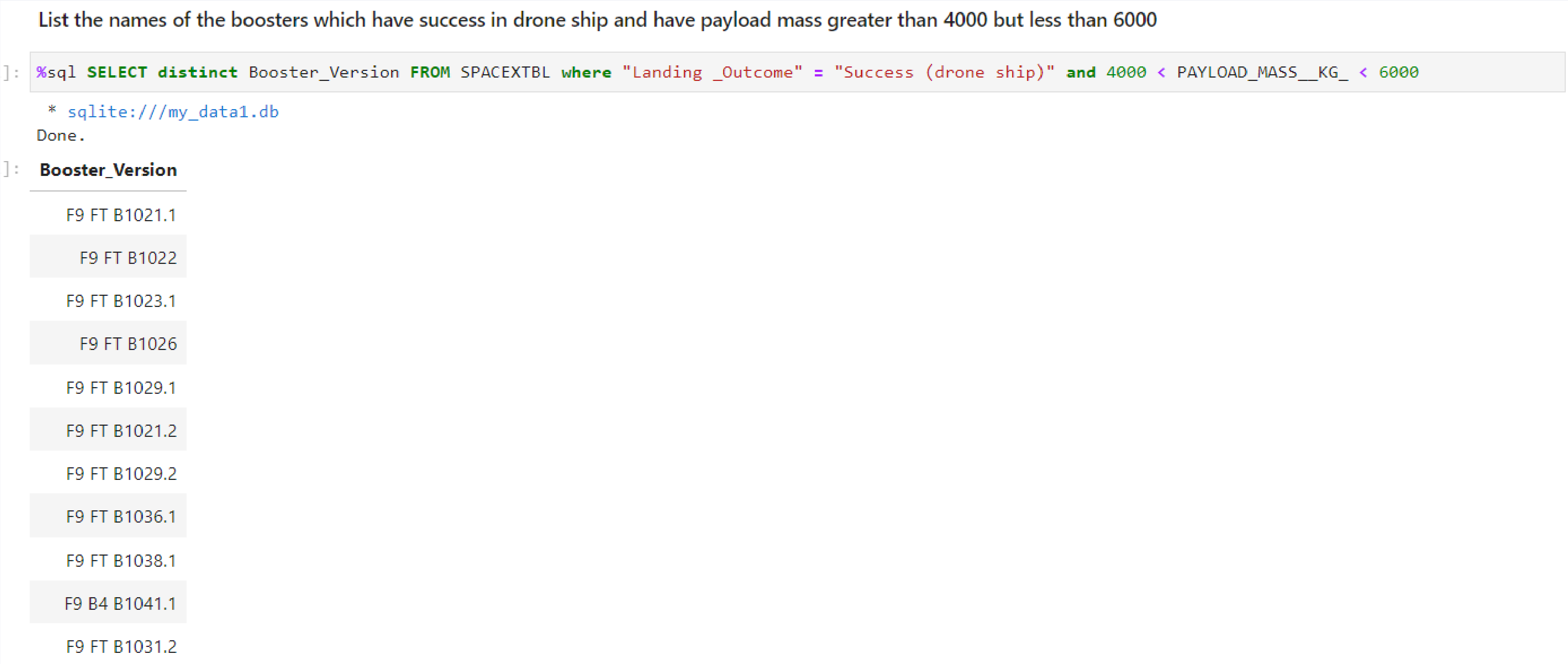
```
%sql SELECT * FROM SPACEXTBL where "Landing _Outcome" = "Success (ground pad)" Order By Date ASC LIMIT 1;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
01-05-2017	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	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



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

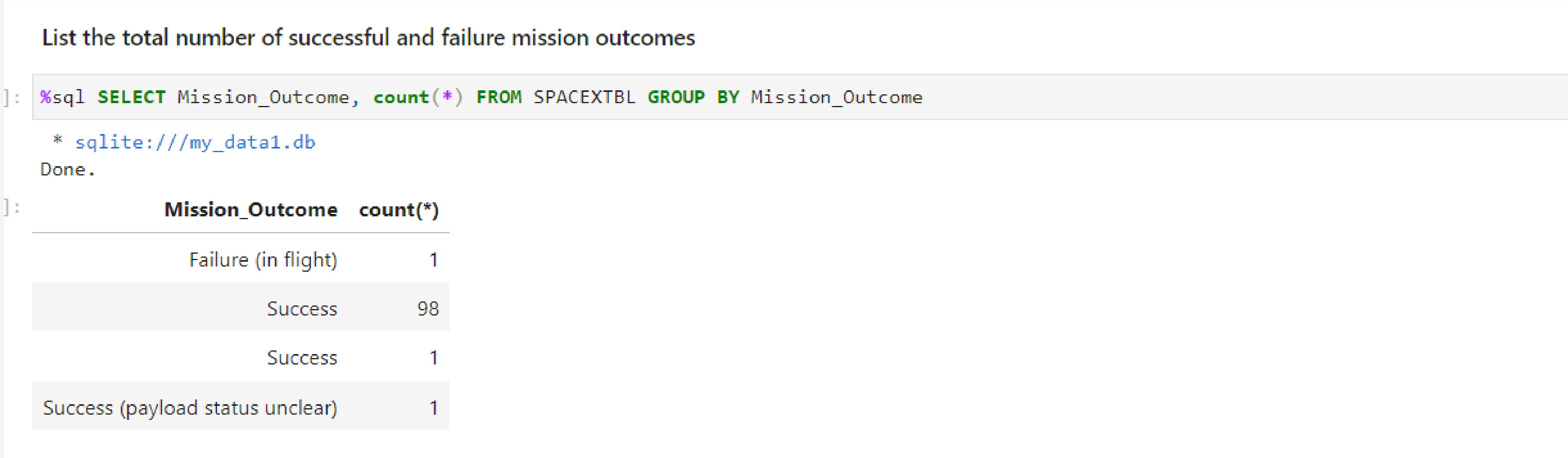
```
]: %sql SELECT distinct Booster_Version FROM SPACEXTBL where "Landing _Outcome" = "Success (drone ship)" and 4000 < PAYLOAD_MASS__KG_ < 6000
```

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

Booster_Version
F9 FT B1021.1
F9 FT B1022
F9 FT B1023.1
F9 FT B1026
F9 FT B1029.1
F9 FT B1021.2
F9 FT B1029.2
F9 FT B1036.1
F9 FT B1038.1
F9 B4 B1041.1
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes



List the total number of successful and failure mission outcomes

```
]: %sql SELECT Mission_Outcome, count(*) FROM SPACEXTBL GROUP BY Mission_Outcome
```

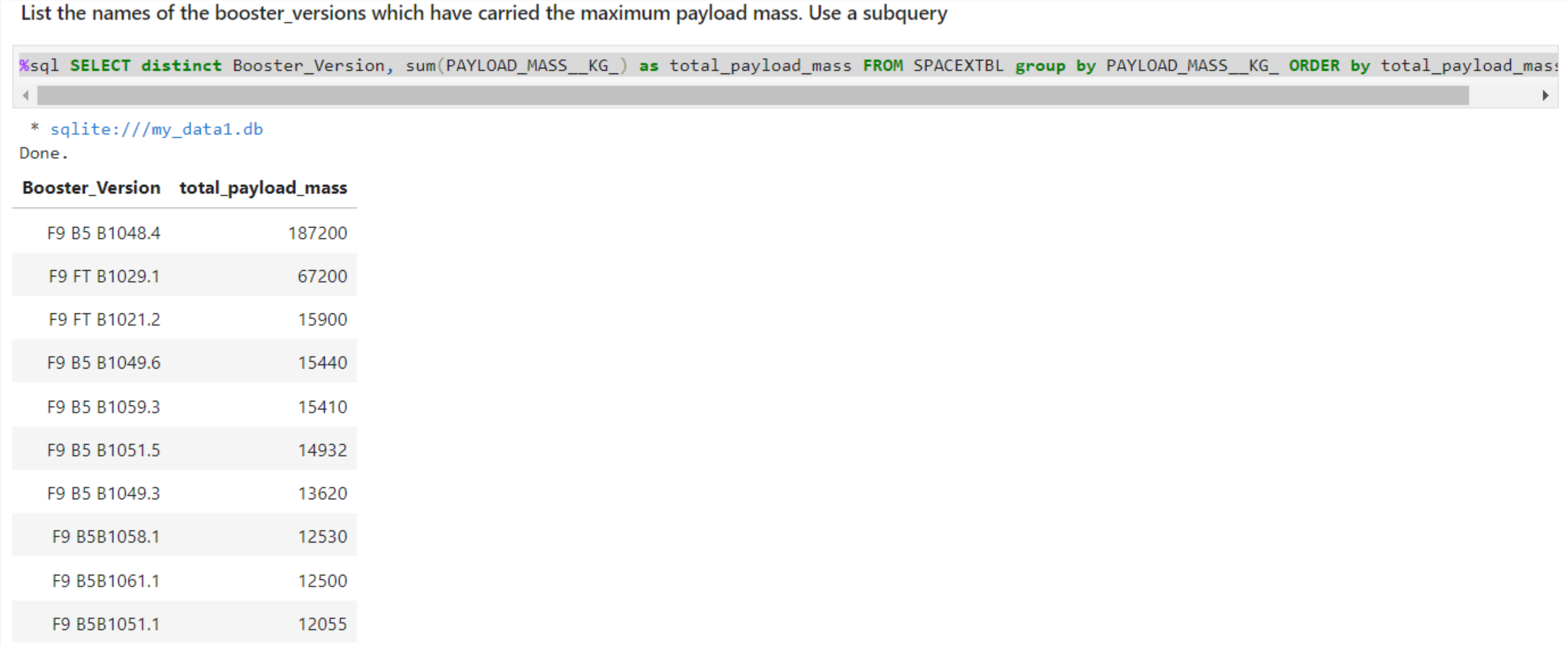
```
 * sqlite:///my_data1.db
Done.
```

]:

Mission_Outcome	count(*)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
  - %sql SELECT distinct Booster_Version, sum(PAYLOAD_MASS__KG_) as total_payload_mass FROM SPACEXTBL group by PAYLOAD_MASS__KG_ ORDER by total_payload_mass desc;



List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

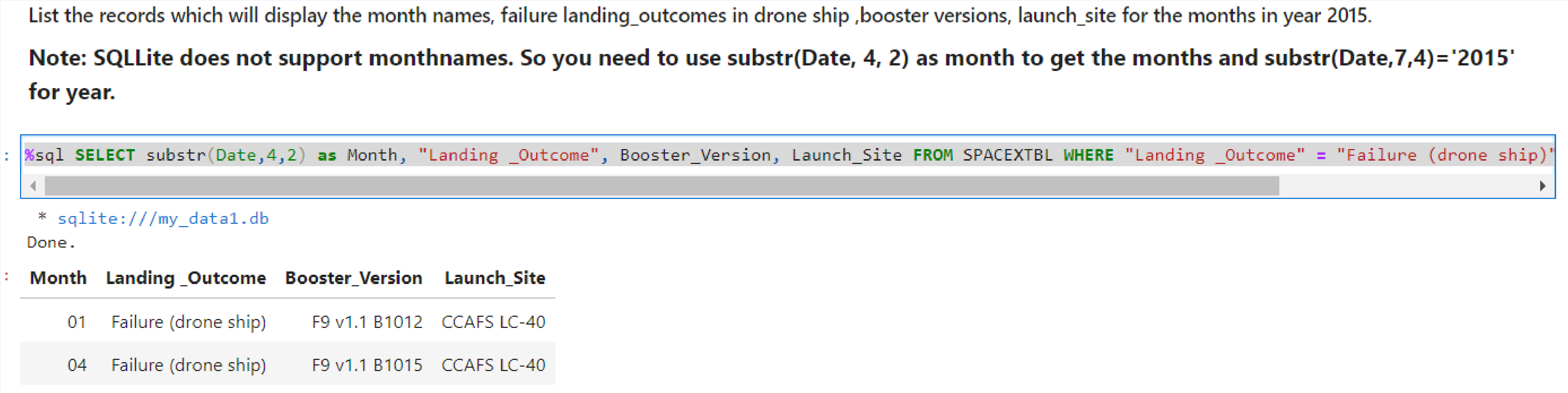
```
%sql SELECT distinct Booster_Version, sum(PAYLOAD_MASS__KG_) as total_payload_mass FROM SPACEXTBL group by PAYLOAD_MASS__KG_ ORDER by total_payload_mas
```

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

Booster_Version	total_payload_mass
F9 B5 B1048.4	187200
F9 FT B1029.1	67200
F9 FT B1021.2	15900
F9 B5 B1049.6	15440
F9 B5 B1059.3	15410
F9 B5 B1051.5	14932
F9 B5 B1049.3	13620
F9 B5B1058.1	12530
F9 B5B1061.1	12500
F9 B5B1051.1	12055

2015 Launch Records

- List the 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 SPACEXTBL WHERE "Landing _Outcome" = "Failure (drone ship)" AND substr(Date,7,4) = "2015“



List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

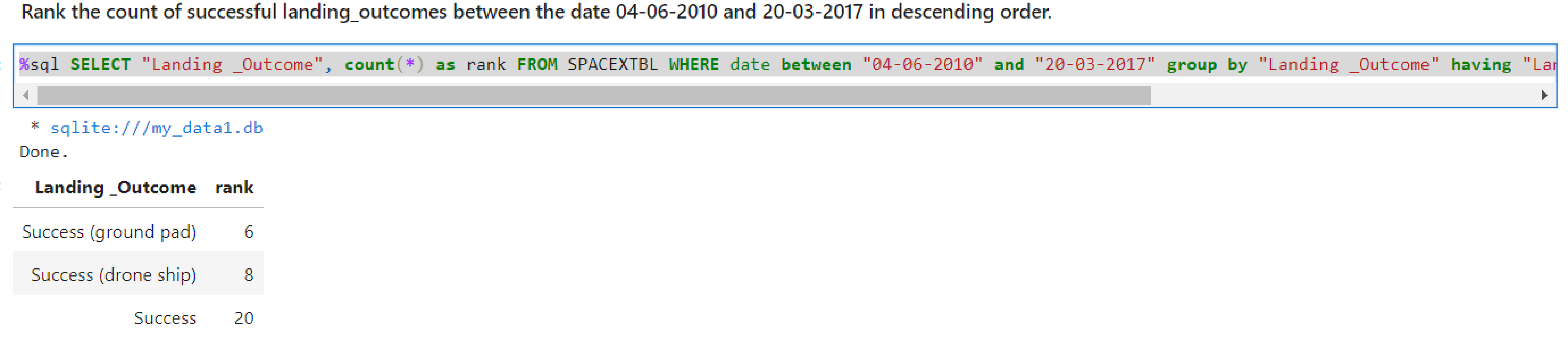
```
%sql SELECT substr(Date,4,2) as Month, "Landing _Outcome", Booster_Version, Launch_Site FROM SPACEXTBL WHERE "Landing _Outcome" = "Failure (drone ship)"
```

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

Month	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

- 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 rank FROM SPACEXTBL WHERE date between "04-06-2010" and "20-03-2017" group by "Landing _Outcome" having "Landing _Outcome" like "%Success%" order by date desc



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

```
%sql SELECT "Landing _Outcome", count(*) as rank FROM SPACEXTBL WHERE date between "04-06-2010" and "20-03-2017" group by "Landing _Outcome" having "Lar
```

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

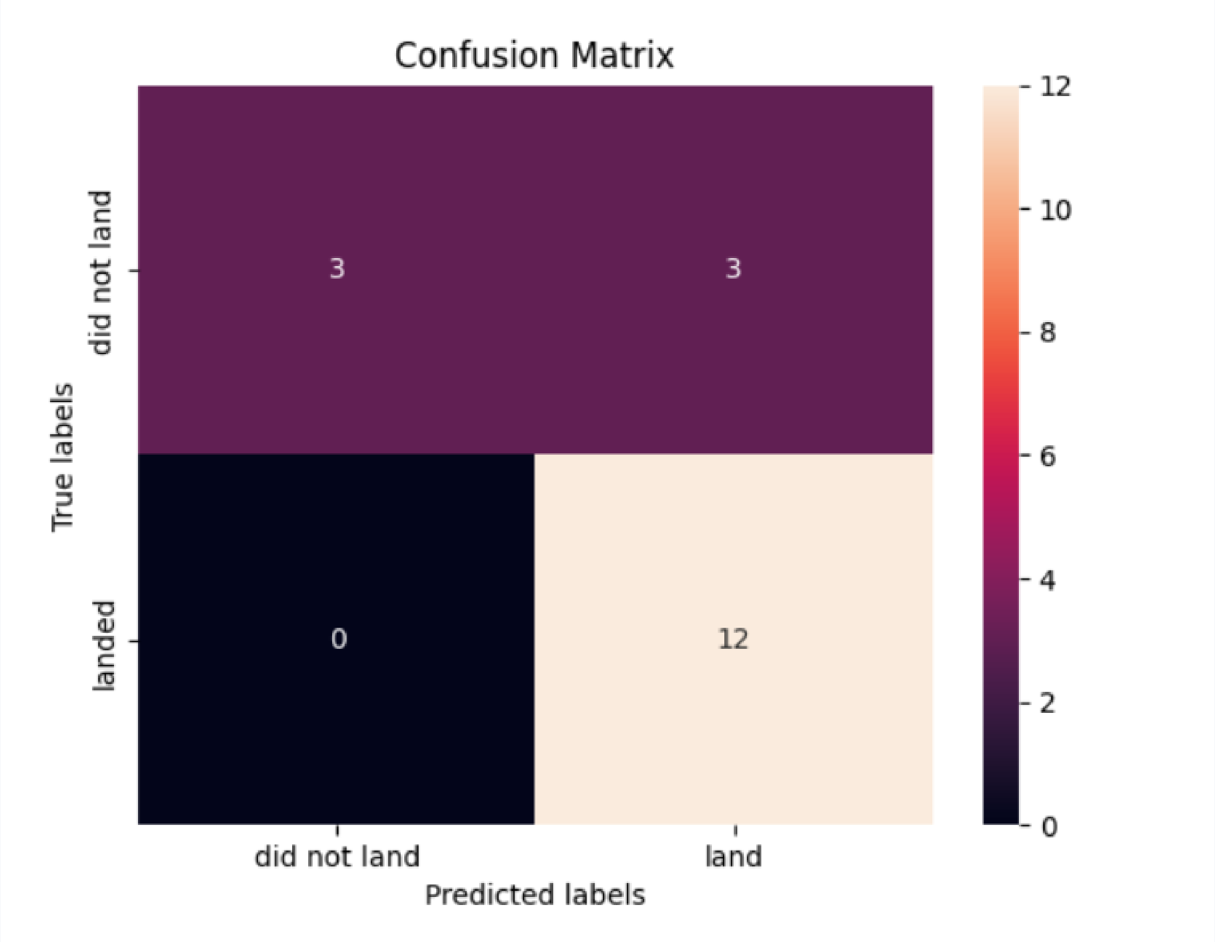
Landing _Outcome	rank
Success (ground pad)	6
Success (drone ship)	8
Success	20

Section 5

Predictive Analysis (Classification)

Confusion Matrix

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

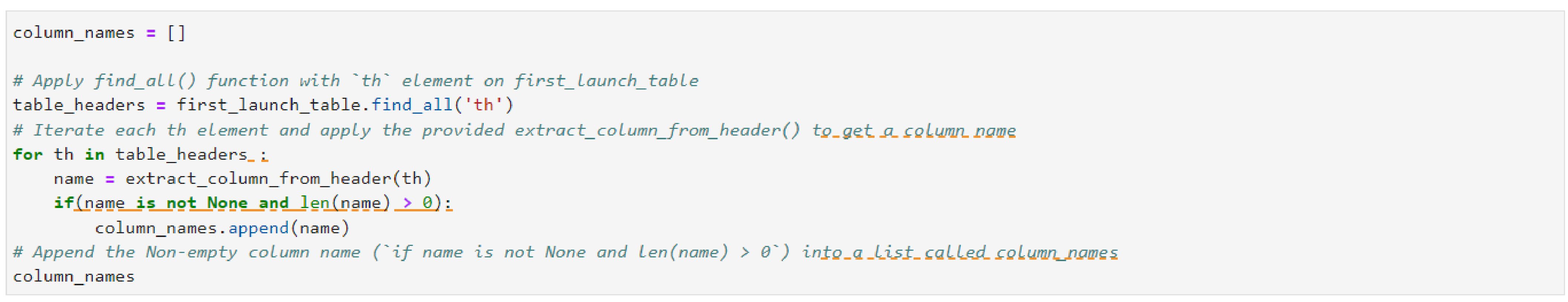


Conclusions

- We used SpaceX launch data and analyzed it to use it so that we can predict if landing will be successful or not using classification models.
- This prediction can help in estimating space launch cost.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project



```python
column_names = []

# Apply find_all() function with `th` element on first_launch_table
table_headers = first_launch_table.find_all('th')
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
for th in table_headers :
    name = extract_column_from_header(th)
    if(name is not None and len(name) > 0):
        column_names.append(name)
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names
column_names
```

Thank you!