IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Tadiwos Zerihun Desta
03/04/2024

Outline

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

Executive Summary

This presentation uses publicly available data to evaluate the success rate of the Space X program using the Falcon 9 launcher. Data science methods are used to collect and wrangle the data. Afterwards the data is analysed and visualised. Further more predictive models were proposed and evaluated.

Introduction

- Project background and context
  - Space x is the cheapest space travel provider because it re-uses the first stage. Therefore if we determine that the first stage would land, then we can determine the cost of the launch. The aim of this project is to start a new company Space Y that can compite with Space X.
- Problems you want to find answers
  - Determine the price of each launch of Space X
  - Whether or not the first stage is re-used or not
  - Use machine learning and publicly available data to perform the tasks

Section 1

Methodology

Methodology

Executive Summary

The data are collected from Space X REST API and from Wiki pages using web scrabing methods. The collected data was wrangles and cleaned up. The cleaned data analysed and visualised using exploratory data analysis (EDA) and SQL approaxches. In addition Folium and Plotly Dash methods are used to interactively analyse the data. Finally predictive analysis was performed using classification models.

Data Collection

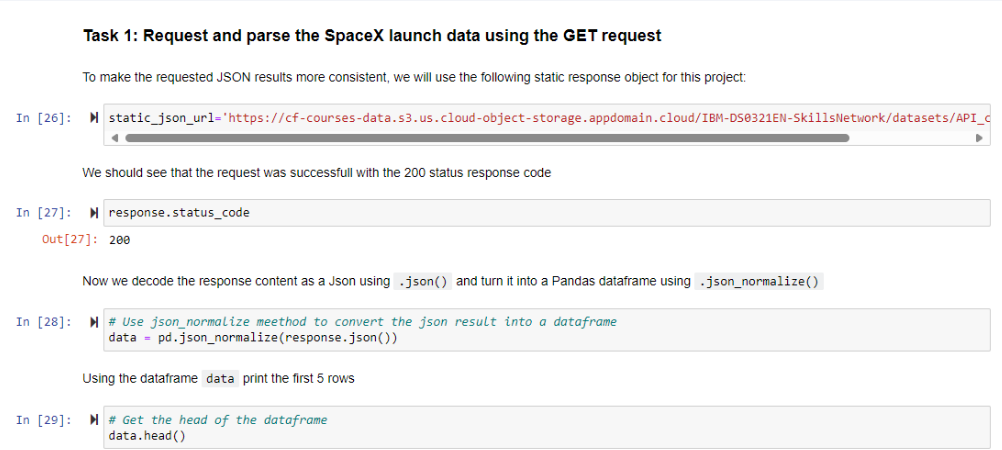
Space x data is gathered from Space X REST API. Information about the rocket used, payload delivered, launcher specification, landing specification and landing outcome are gathered. The goal is to predict whether space x will attempt to land a rocket or not. The data were collected in three steps:

- **Step 1: Request and parse the SpaceX launch data using the GET request**
- **Step 2: Filter the dataframe to only include Falcon 9 launches**
- **Step 3: Data Wrangling - Dealing with Missing Values**

Data Collection – SpaceX API

- Request and parse the Space X launch data using the GET request from Space X API and clean the data.
- Below the GitHub URL of the complete Space X API calls and data cleaning

https://github.com/Tadiwosz/Applied-Data-Science-Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
In [26]:  static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_c
```

We should see that the request was successfull with the 200 status response code

```
In [27]:  response.status_code

Out[27]:  200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
In [28]:  # Use json_normalize meethod to convert the json result into a dataframe
          data = pd.json_normalize(response.json())
```

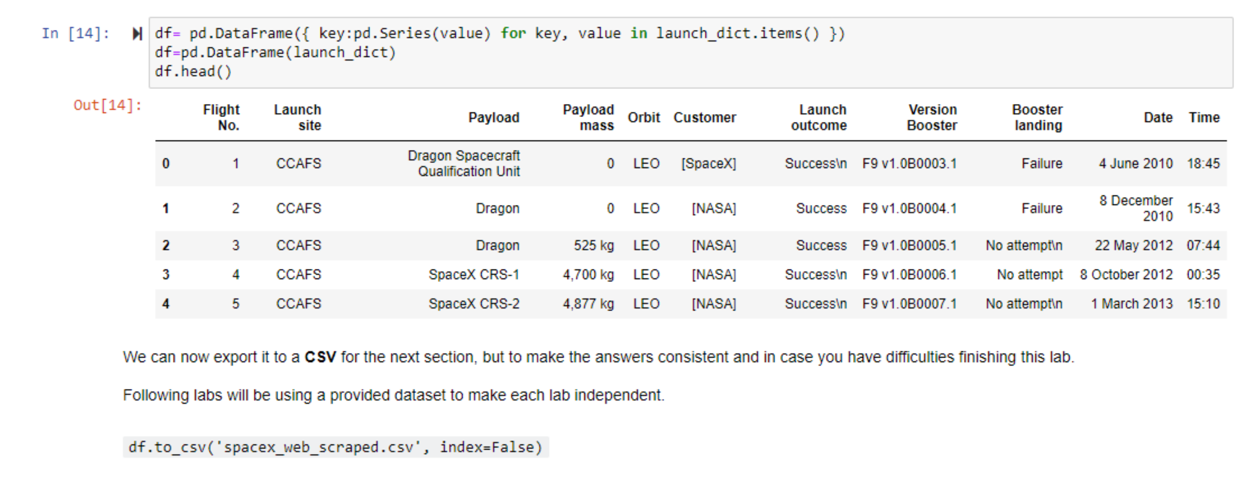
Using the dataframe `data` print the first 5 rows

```
In [29]:  # Get the head of the dataframe
          data.head()
```

Data Collection - Scraping

- Webscrap Falcon 9 launch records using BeutifulSoup
  - Extract a Falcon 9 launch records HTML table from Wikipedia
  - Parse the table and convert it into a Pandas data frame
- Below is the GitHub URL of the completed web scraping notebook:

https://github.com/Tadiwosz/Applied-Data-Science-Capstone/blob/main/jupyter-labs-webscraping.ipynb



In [14]:

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

Out[14]:

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	[SpaceX]	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	[NASA]	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	[NASA]	Success	F9 v1.0B0005.1	No attempt\n	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	[NASA]	Success\n	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	[NASA]	Success\n	F9 v1.0B0007.1	No attempt\n	1 March 2013	15:10

We can now export it to a **CSV** for the next section, but to make the answers consistent and in case you have difficulties finishing this lab.

Following labs will be using a provided dataset to make each lab independent.

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

Data Wrangling

- Analyse the collected data,
  - Proportion of missing values in each attributes
  - Understand the data (number of launches per site, occurrence in each orbit, mission outcome)
  - Create landing outcome label (0 or 1)
- perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.

- Below is the GitHub URL of my completed data wrangling related notebooks:

https://github.com/Tadiwosz/Applied-Data-Science-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Summary of chart types and the reason of the plots
  - Visualize the effect of flight number and pay load mass on launch outcome (scatter plot)
  - Visualise the relationship between flight no. and launch site (scatter plot)
  - Visualize the relationship between Payload and Launch Site (scatter plot)
  - Visualize the relationship between success rate of each orbit type (bar chart)
  - Visualize the relationship between FlightNumber and Orbit type (scatter plot)
  - Visualize the launch success yearly trend (line chart)
  - Visualize the relationship between Payload and Orbit type (scatter plot)
  - Visualize the launch success yearly trend (line chart)
- Berlow is the GitHub URL of my completed EDA with data visualization notebook, as an external reference and peer-review purpose
- https://github.com/Tadiwosz/Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- List of the SQL querie:
  - Display the names of the unique launch sites in the space mission
  - Display 5 records where launch sites begin with the string 'CCA’
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1¶
  - List the date when the first succesful landing outcome in ground pad was acheived.
  - List the total number of successful and failure mission outcomes
  - List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - 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.¶
  - 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
- Here the GitHub URL of mycompleted EDA with SQL notebook,

https://github.com/Tadiwosz/Applied-Data-Science-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Circles, markesr and polyline were created and added to the map and used for:

- To mark all launch sites
- Mark the success/failed launches for each site on the map
- Calculate the distances between a launch site to its proximities

- Here the GitHub URL of your completed interactive map with Folium map

https://github.com/Tadiwosz/Applied-Data-Science-Capstone/blob/main/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

The plotly lab did not work for me !

Predictive Analysis (Classification)

- The following steps were followed to build, evaluate, improve, and find the best performing classification model
- create a column for the class
- Standardize the data
- Split into training data and test data
- Find best Hyperparameter for SVM, Classification Trees and Logistic Regression
- Find the method performs best using test d

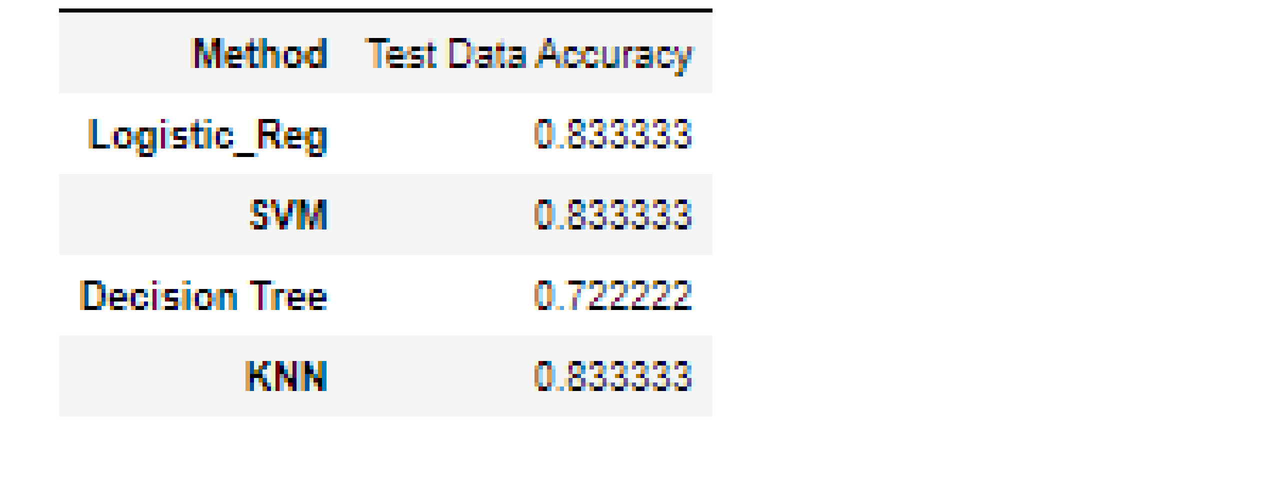
Detailed steps:

Split the data to training and test data → standardize the training data →choodse the model →Set model parameters→ grid search→fit the model using the training data →predict using the model → Evaluate the accuracy using confusion matrix and relevant scores

- Here is the GitHub URL of my completed predictive analysis lab,

https://github.com/Tadiwosz/Applied-Data-Science-Capstone/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results



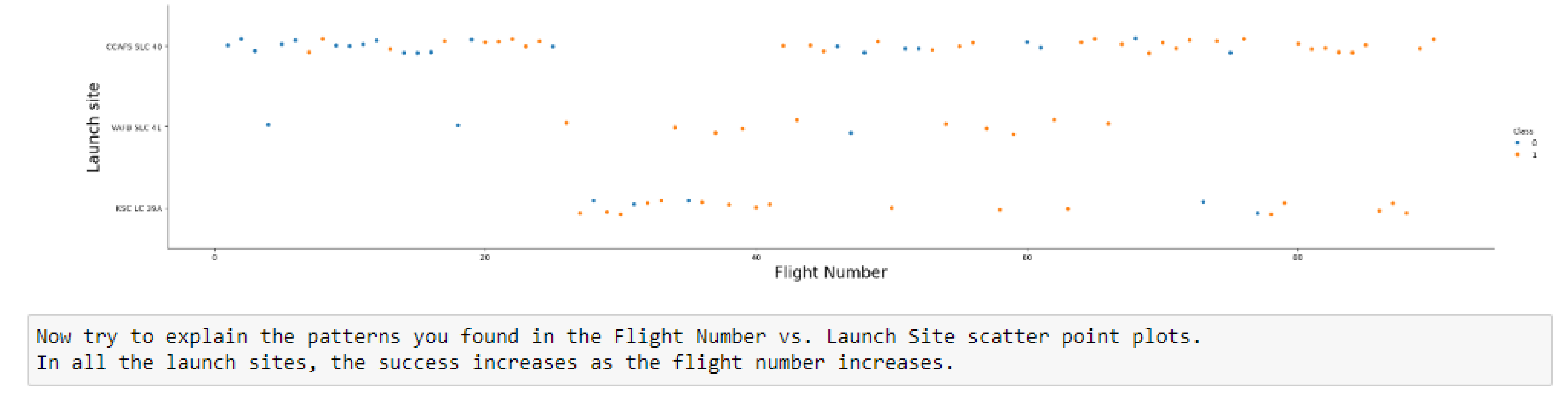
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.722222
KNN	0.833333

Best performers: Logistic reg, SVM and KNN

Section 2

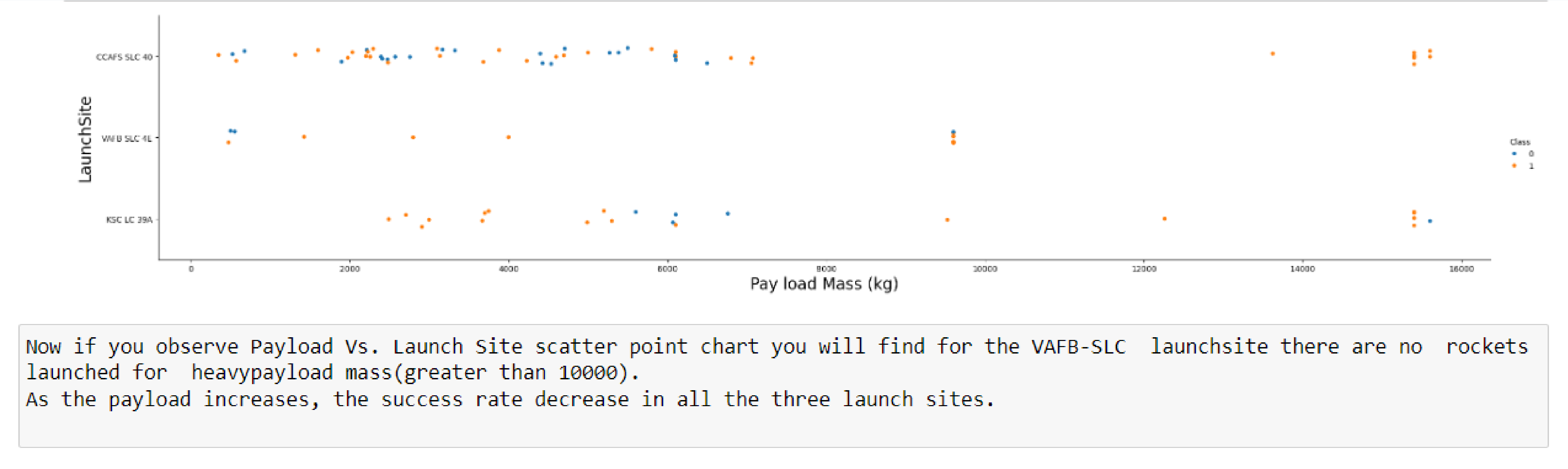
Insights drawn from EDA

Flight Number vs. Launch Site



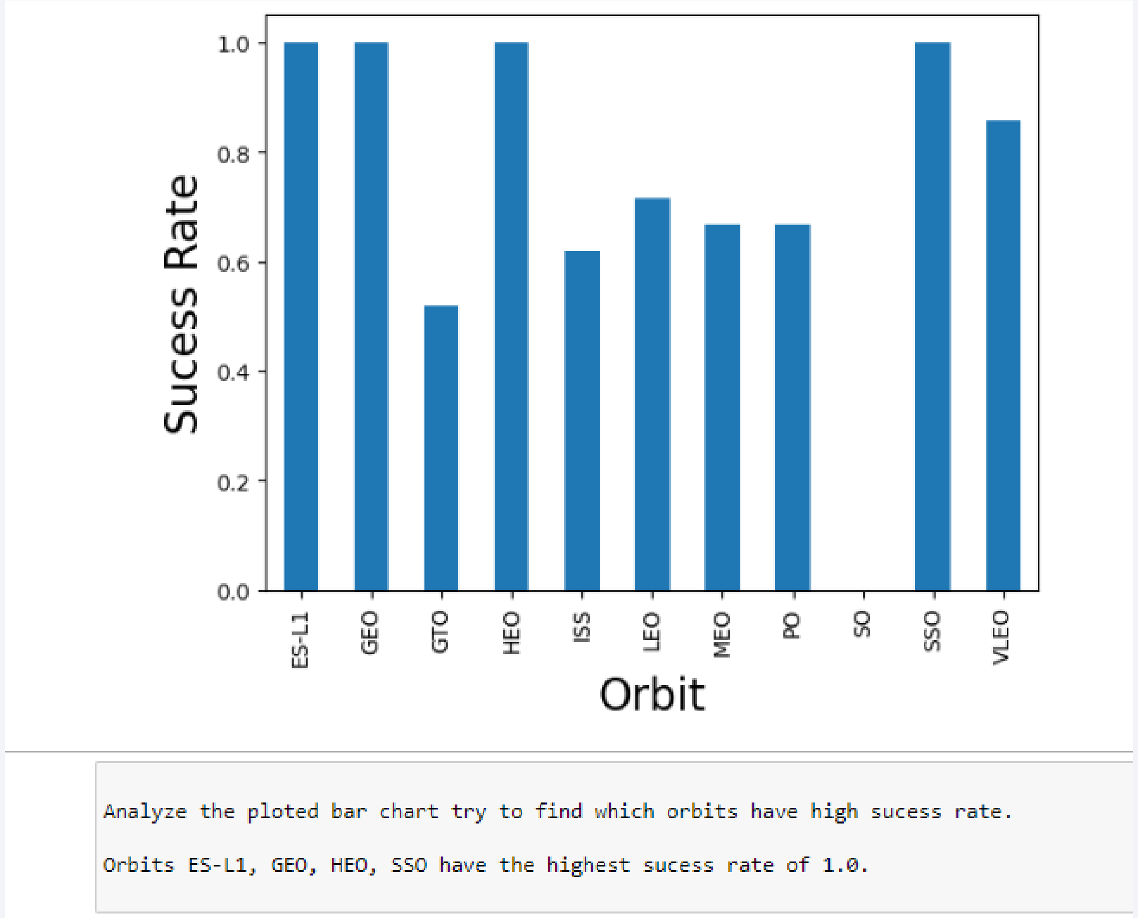
```
Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.
In all the launch sites, the success increases as the flight number increases.
```

Payload vs. Launch Site



Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).
As the payload increases, the success rate decrease in all the three launch sites.

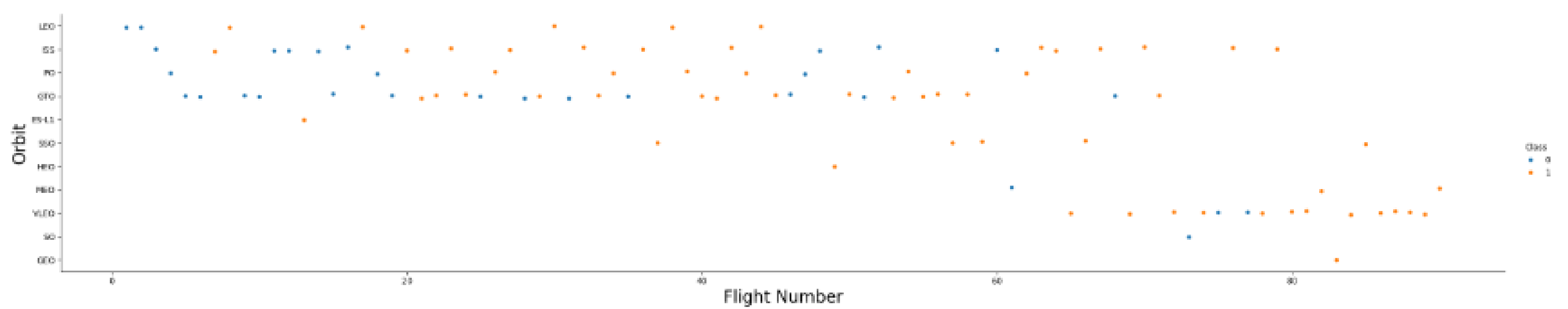
Success Rate vs. Orbit Type



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

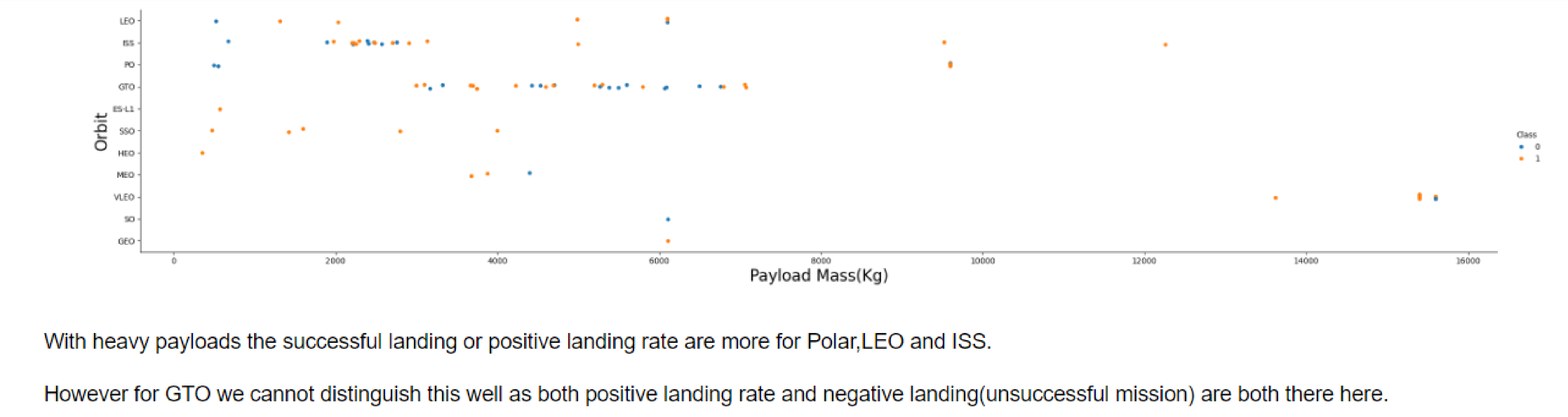
Orbits ES-L1, GEO, HEO, SSO have the highest sucess rate of 1.0.

Flight Number vs. Orbit Type



You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

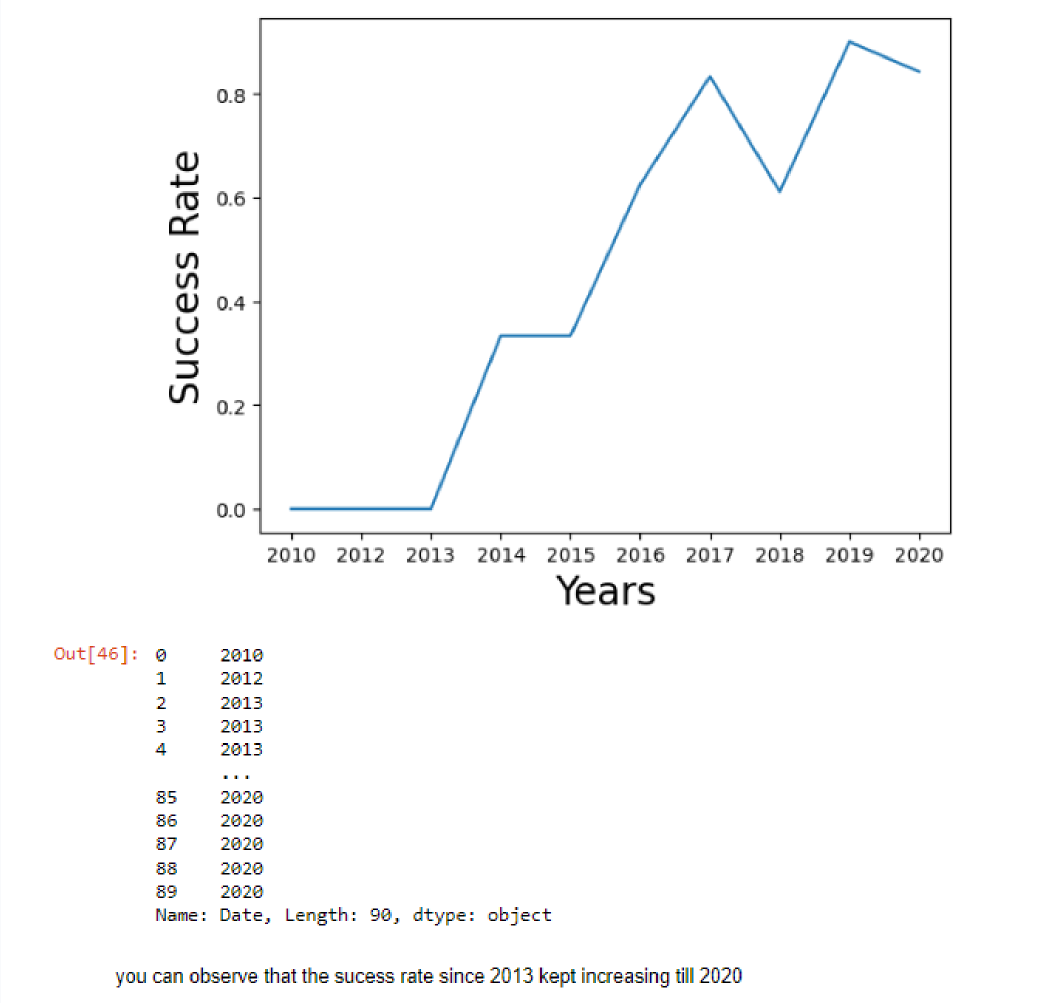
Payload vs. Orbit Type



With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

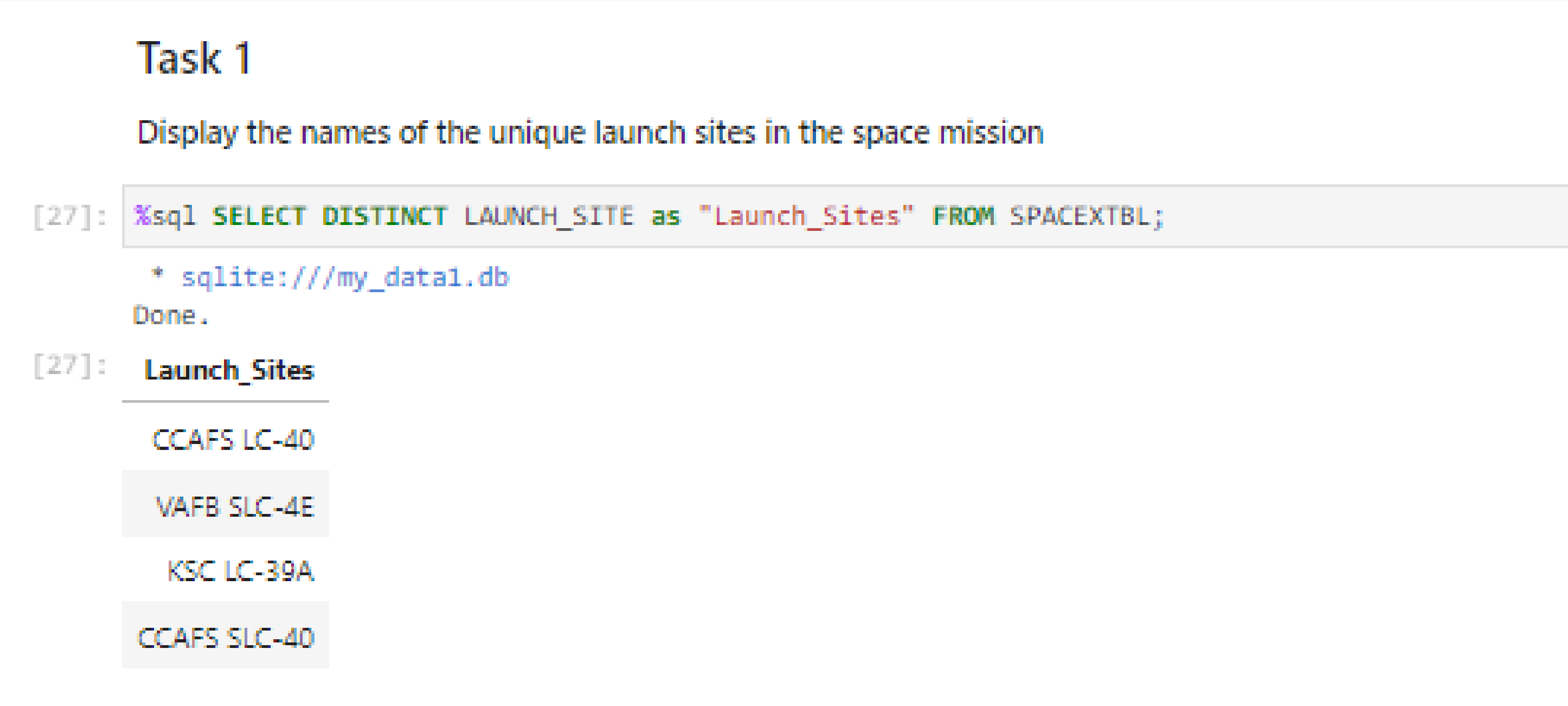
Launch Success Yearly Trend



```
Out[46]: 0      2010
         1      2012
         2      2013
         3      2013
         4      2013
                ...
         85     2020
         86     2020
         87     2020
         88     2020
         89     2020
         Name: Date, Length: 90, dtype: object
```

you can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names



Task 1

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

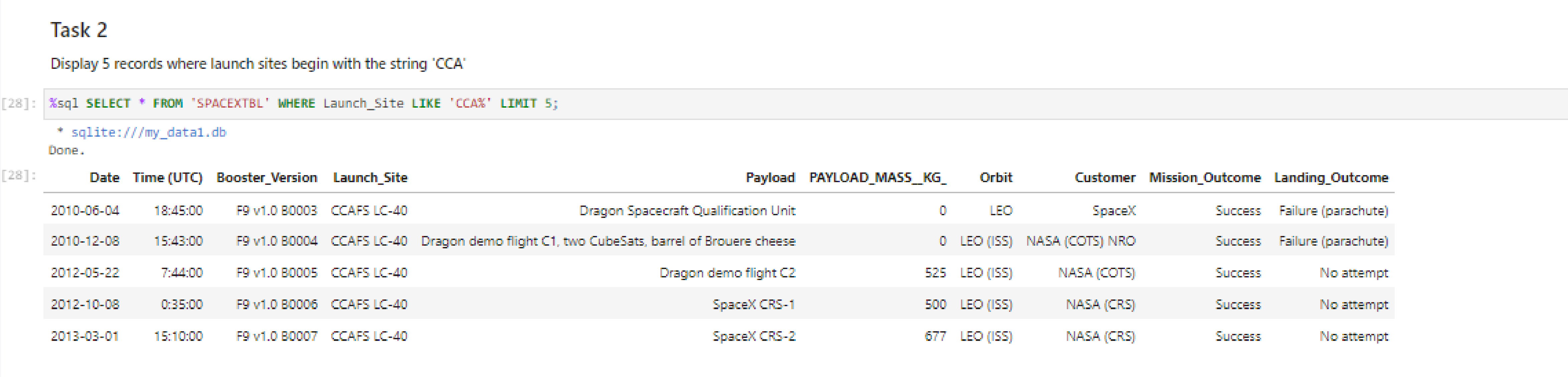
```
[27]: %sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

[27]:

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

Launch Site Names Begin with 'CCA'



Task 2

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

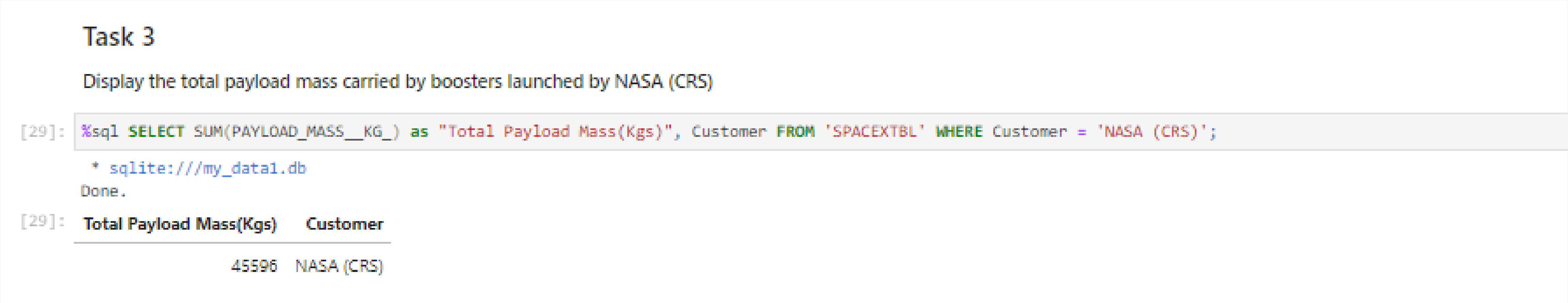
```
[28]: %sql SELECT * FROM 'SPACEXTBL' WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

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

[28]:

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



Task 3

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

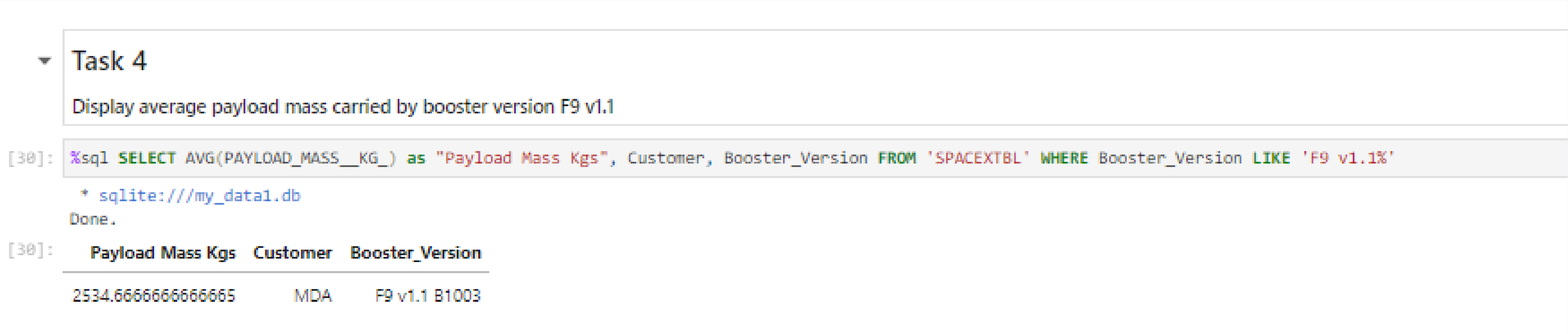
```
[29]: %sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA (CRS)';
```

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

[29]:

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

Average Payload Mass by F9 v1.1



Task 4

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

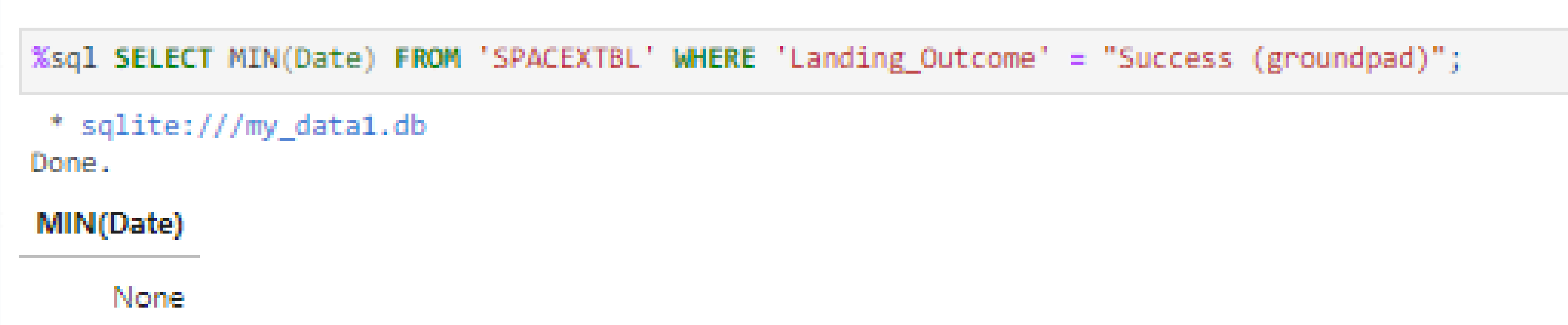
```
[30]: %sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Version LIKE 'F9 v1.1%'
```

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

[30]:

Payload Mass Kgs	Customer	Booster_Version
2534.6666666666665	MDA	F9 v1.1 B1003

First Successful Ground Landing Date



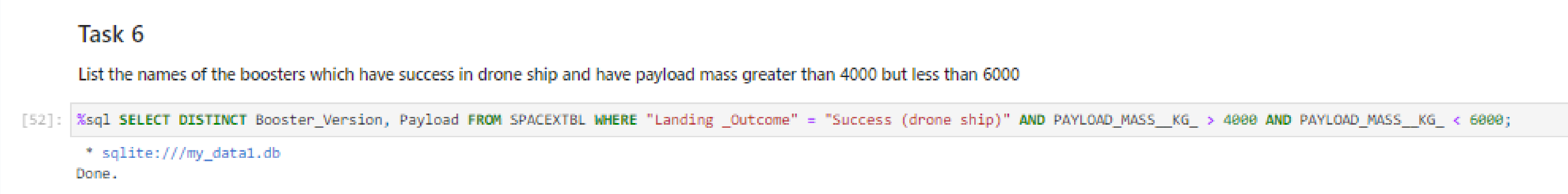
```
%sql SELECT MIN(Date) FROM 'SPACEXTBL' WHERE 'Landing_Outcome' = "Success (groundpad)";
```

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

MIN(Date)
None

The sql command seems ok, but the results are not displayed

Successful Drone Ship Landing with Payload between 4000 and 6000



Task 6

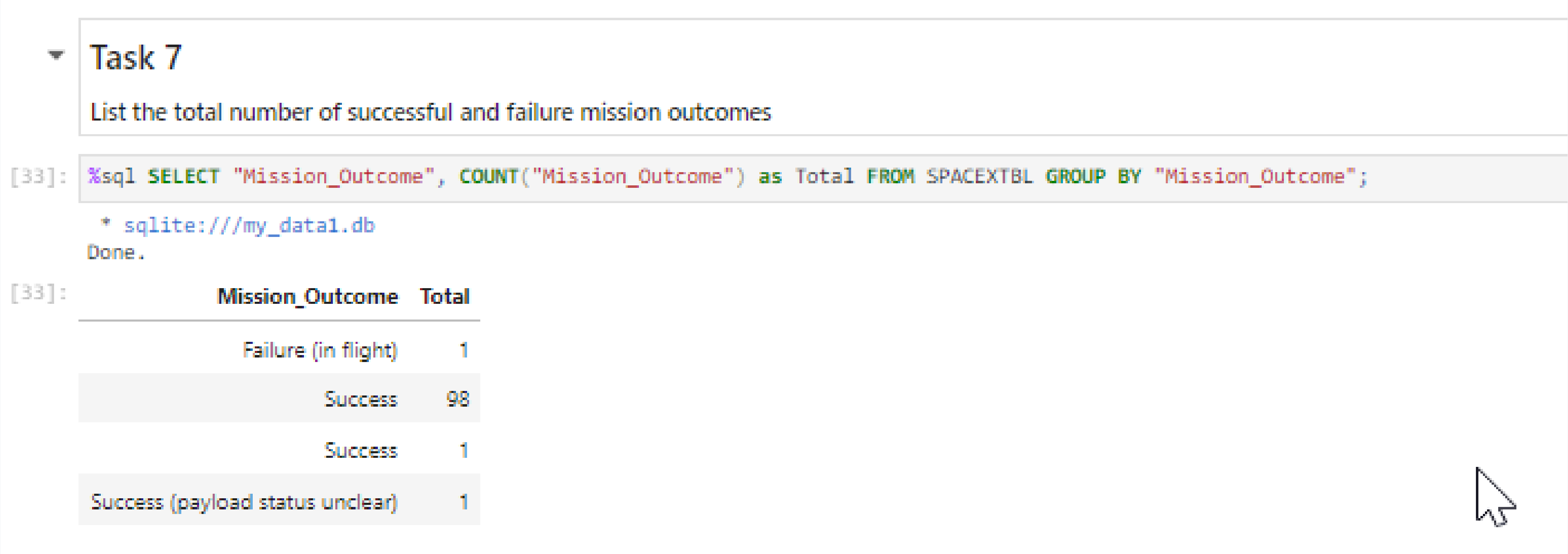
List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
[52]: %sql SELECT DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE "Landing _Outcome" = "Success (drone ship)" AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

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

The sql command seems ok, but the results are not displayed

Total Number of Successful and Failure Mission Outcomes



Task 7

List the total number of successful and failure mission outcomes

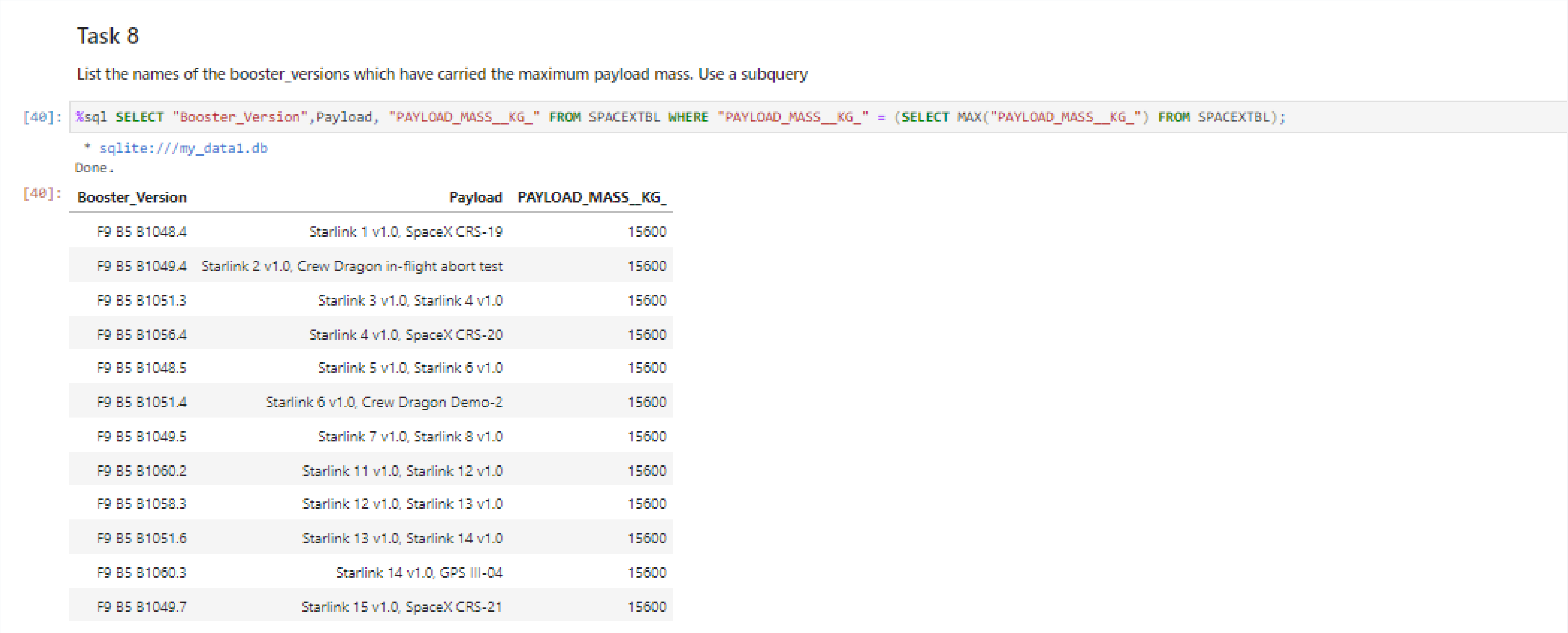
```
[33]: %sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";
```

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

[33]:

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

Boosters Carried Maximum Payload



Task 8

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

```
[40]: %sql SELECT "Booster_Version",Payload, "PAYLOAD_MASS__KG_" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL);
```

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

[40]:

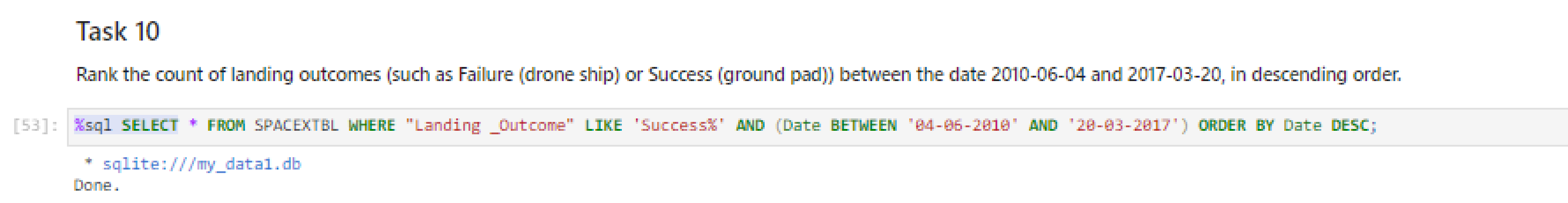
Booster_Version	Payload	PAYLOAD_MASS__KG_
F9 B5 B1048.4	Starlink 1 v1.0, SpaceX CRS-19	15600
F9 B5 B1049.4	Starlink 2 v1.0, Crew Dragon in-flight abort test	15600
F9 B5 B1051.3	Starlink 3 v1.0, Starlink 4 v1.0	15600
F9 B5 B1056.4	Starlink 4 v1.0, SpaceX CRS-20	15600
F9 B5 B1048.5	Starlink 5 v1.0, Starlink 6 v1.0	15600
F9 B5 B1051.4	Starlink 6 v1.0, Crew Dragon Demo-2	15600
F9 B5 B1049.5	Starlink 7 v1.0, Starlink 8 v1.0	15600
F9 B5 B1060.2	Starlink 11 v1.0, Starlink 12 v1.0	15600
F9 B5 B1058.3	Starlink 12 v1.0, Starlink 13 v1.0	15600
F9 B5 B1051.6	Starlink 13 v1.0, Starlink 14 v1.0	15600
F9 B5 B1060.3	Starlink 14 v1.0, GPS III-04	15600
F9 B5 B1049.7	Starlink 15 v1.0, SpaceX CRS-21	15600

2015 Launch Records

```
%sql SELECT substr(Date,6,2), substr(Date, 6, 2),"Booster_Version", "Launch_Site", Payload,
"PAYLOAD_MASS__KG_", "Mission_Outcome", "Landing _Outcome" FROM SPACEXTBL WHERE
substr(Date,0,5)='2015' AND "Landing _Outcome" = 'Failure (drone ship)';
```

The sql command seems ok, but the results are not displayed

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



Task 10

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.

```
[53]: %sql SELECT * FROM SPACEXTBL WHERE "Landing _Outcome" LIKE 'Success%' AND (Date BETWEEN '04-06-2010' AND '20-03-2017') ORDER BY Date DESC;
```

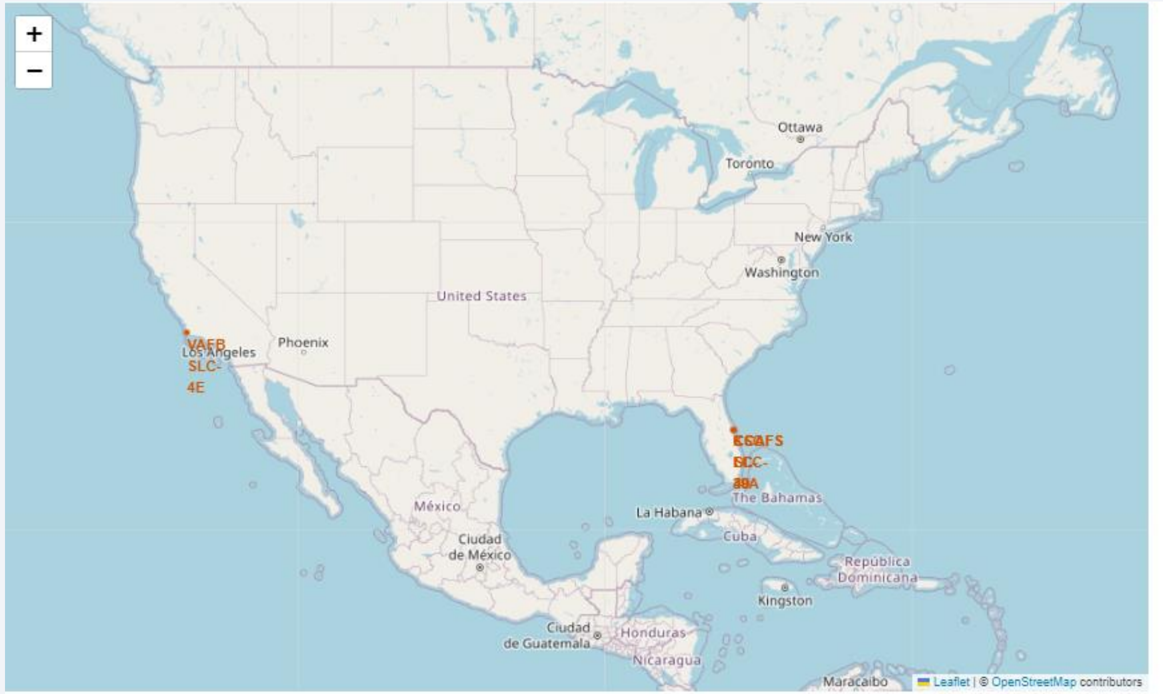
```
 * sqlite:///my_data1.db
Done.
```

The sql command seems ok, but the results are not displayed

Section 3

Launch Sites Proximities Analysis

Launch site location



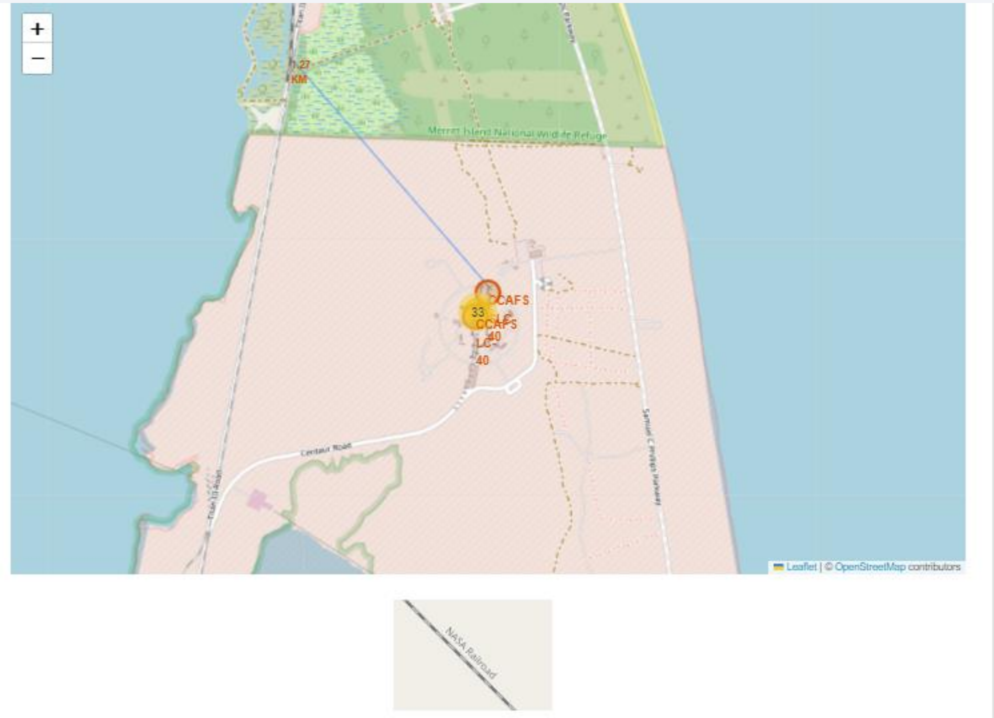
The four launch sites

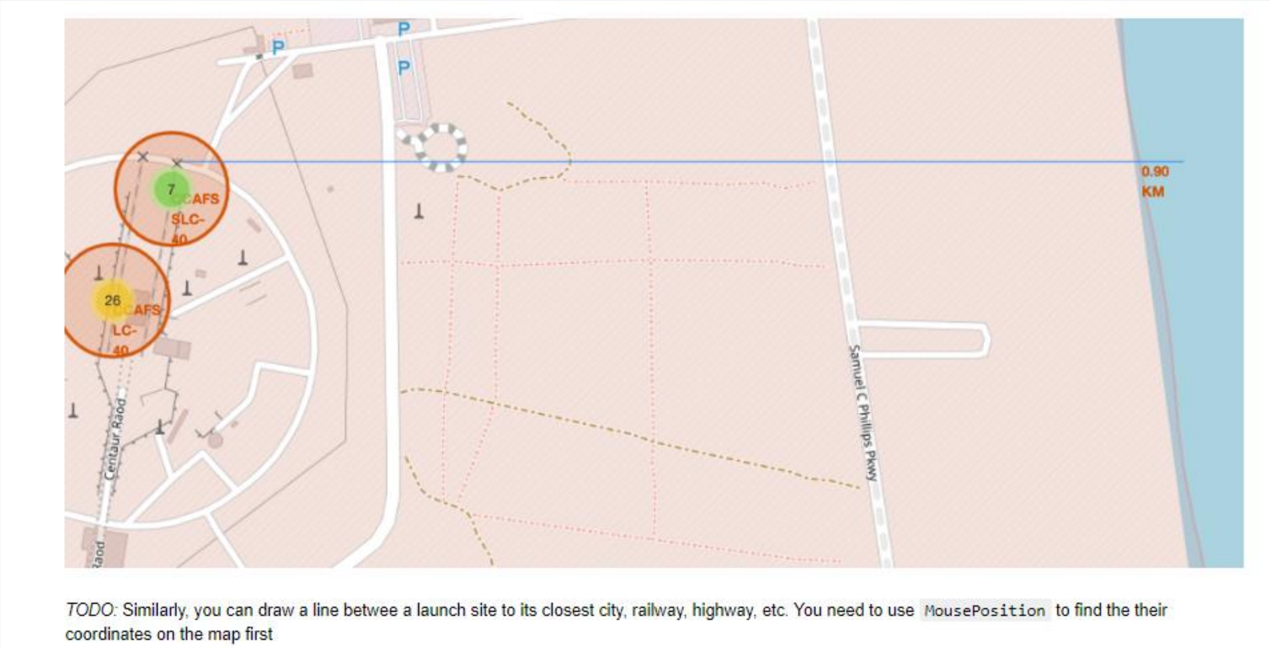
Vandenberg
Space Force
Base
Lompoc
VAFB
SLC-
4E

KSC
LC-
39A
Kennedy Parkway North
Saturn Barge Channel
CCAFS
SLC-
40
CCAFS
LC-
40

CCAFS
SLC-
40
CCAFS
LC-
40
Falcon
our Road

Launch site proximity to important locations

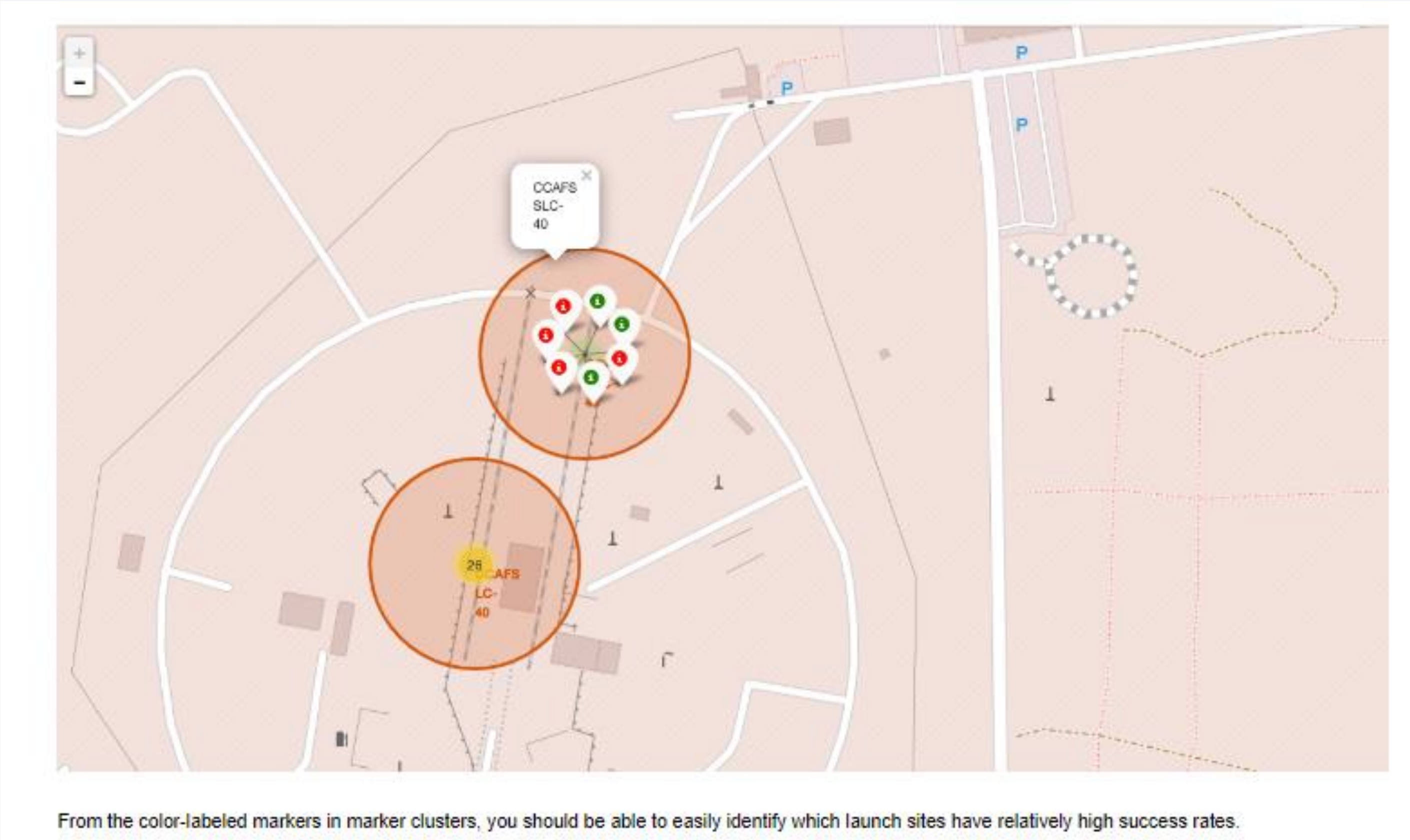

+
−
1.27
KM
Merritt Island National Wildlife Refuge
CCAFS
33
LC-
CCAFS
40
LC-
40
Centaur Road
Samuel C Phillips Parkway
Leaflet | © OpenStreetMap contributors
NASA Railroad



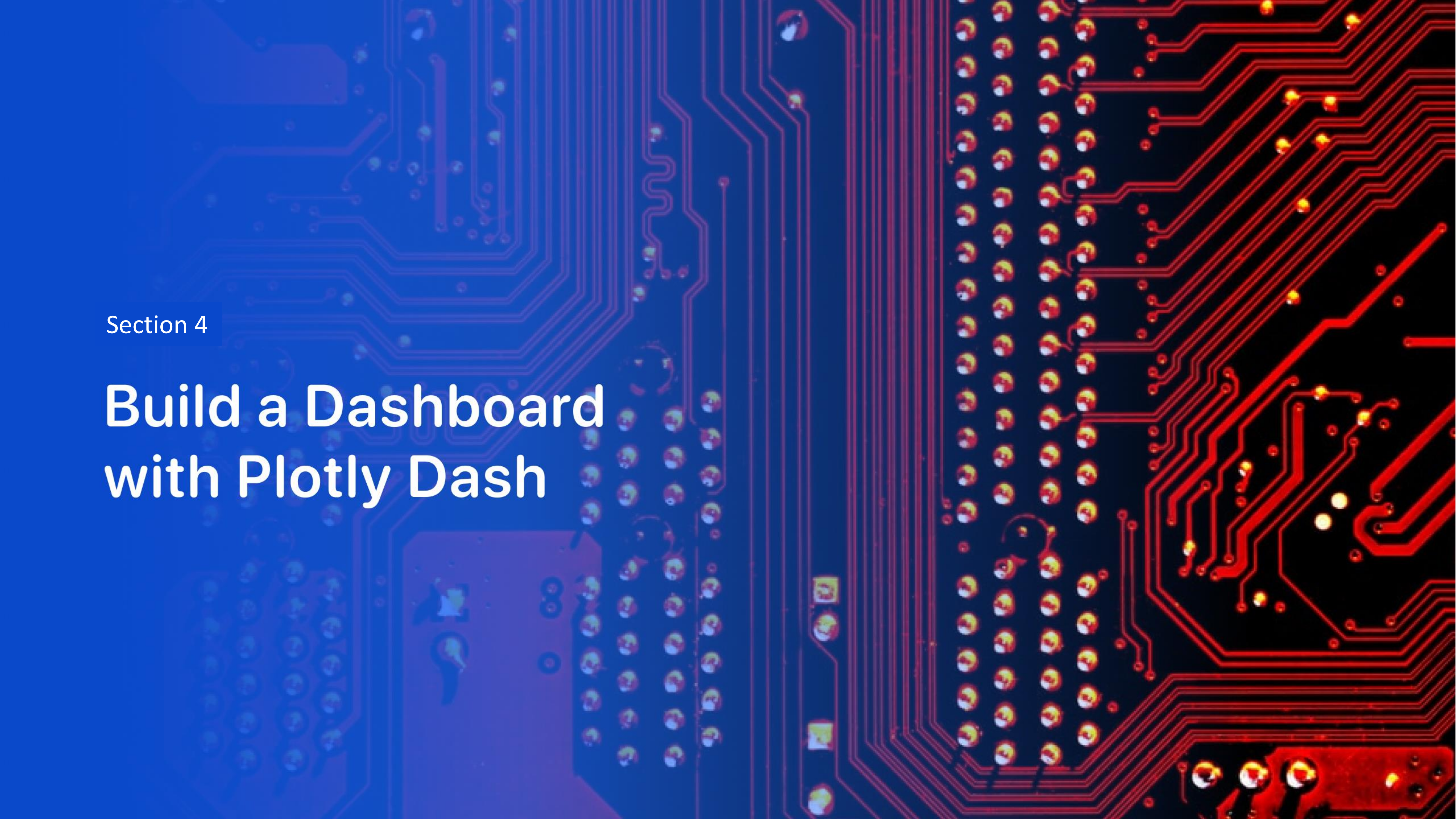
TODO: Similarly, you can draw a line betwee a launch site to its closest city, railway, highway, etc. You need to use `MousePosition` to find the their coordinates on the map first

The proximity of the lanch site to important locations such as railways and city centers can be determined using folium

Use of colour markers to identify launch properties



From the color-labeled markers in marker clusters, you should be able to easily identify which launch sites have relatively high success rates.



Section 4

Build a Dashboard with Plotly Dash

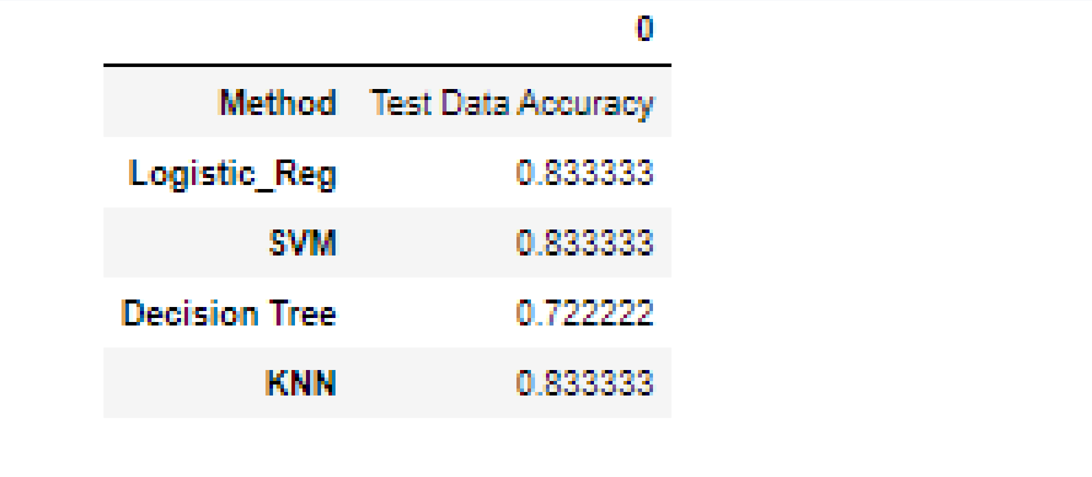
<Dashboard Screenshot 1>

The plotly lab did not work for me !

Section 5

Predictive Analysis (Classification)

Classification Accuracy

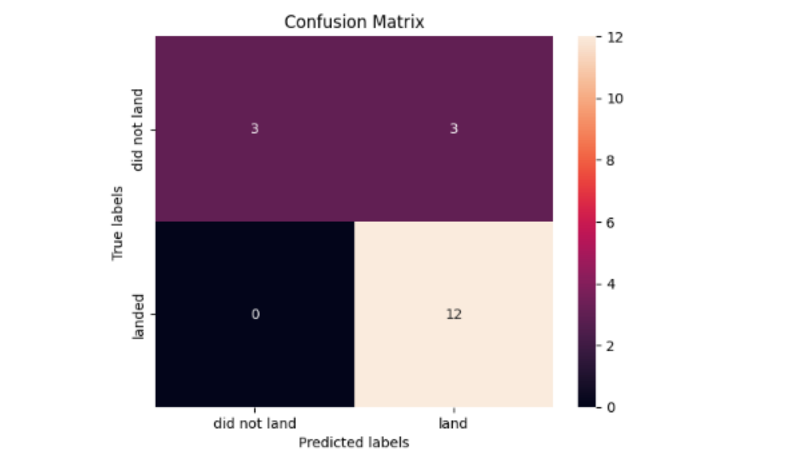


	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.722222
KNN	0.833333

Logistic regression, SVM and KNN have equal accuracy of 0.833 while Decision tree has accuracy of 0.72

Confusion Matrix

- All the 3 best performing models have the same confusion matrix, The main problem is of the models is the false landing prediction.



Conclusions

- Real business case can be studied using publicly available data sets
- The success rate of Falcon 9 laches in space x program is proportional to the number of flights, the more flights the more the success.
- Increased pay load is linked to launch failures
- Some launch sites have better success rate than other sites. The reason for the difference need to be studied.
- Similarly there are differences in the success rate of launches to different orbits. Here also the reason need to be investigated.
- The success rate since 2012 is increasing till 2020. Is there a reason for it? This is something to be looked at
- Classification models could predict the mission outcome with high accuracy and these models can be used for similar tasks in the fuiture

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

Thank you!