IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Hong>
<Nov28 2023>

Outline

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

Executive Summary

- Summary of methodologies
    - Data collection
    - Date wrangling
    - EDA with data visualization
    - EDA with SQL
    - Building interactive map
    - Building dashboard with Plotly
    - Building predictive model

- Summary of all results
    - EDA result
    - Interactive analysis
    - Predictive analysis

Introduction

- Project background and context
  - A new rocket company Space Y would like to compete with SpaceX.

- Problems you want to find answers
  - To determine the price of each launch
    - Gathering information about Space X and creating dashboards
  - Determine if SpaceX will reuse the first stage
    - Training a machine learning model and use public information to predict if SpaceX will reuse the first stage

Section 1

Methodology

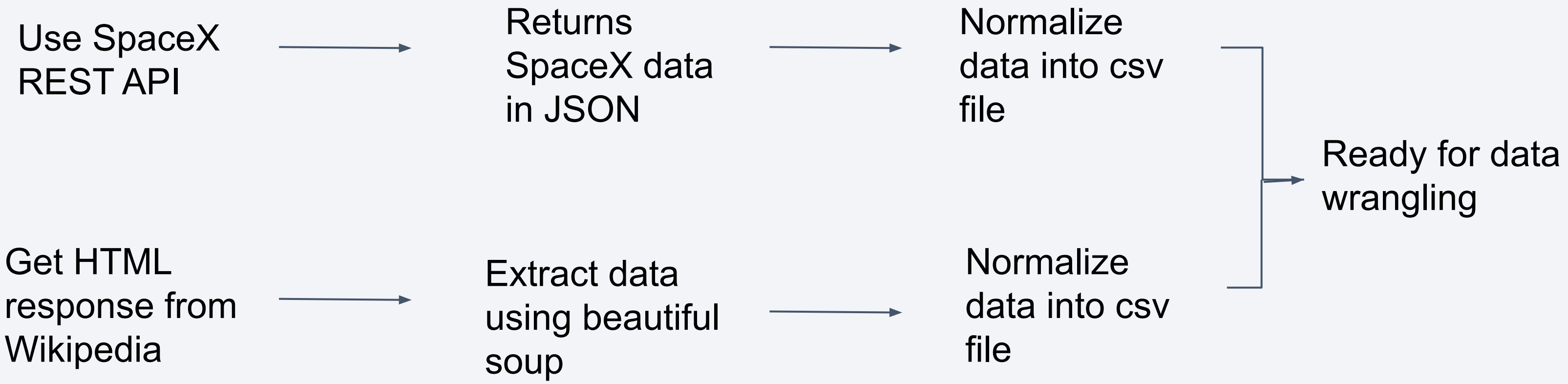
Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Web scraping from Wikipedia
- Perform data wrangling
  - One hot encoding data field for machine learning and data cleaning non values and irrelevant columns
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Linear Regression, Decision Tree, KNN and SVM were built to evaluate the best classifier

Data Collection

- The following datasets was collected
  - SpaceX launch data is gathered from SpaceX REST API
  - Data including launches, the rocket used, payload delivered, launch specifications, landing specification and landing outcomes
  - Falcon 9 launch data was collected through web scraping from Wikipedia website



Use SpaceX REST API → Returns SpaceX data in JSON → Normalize data into csv file

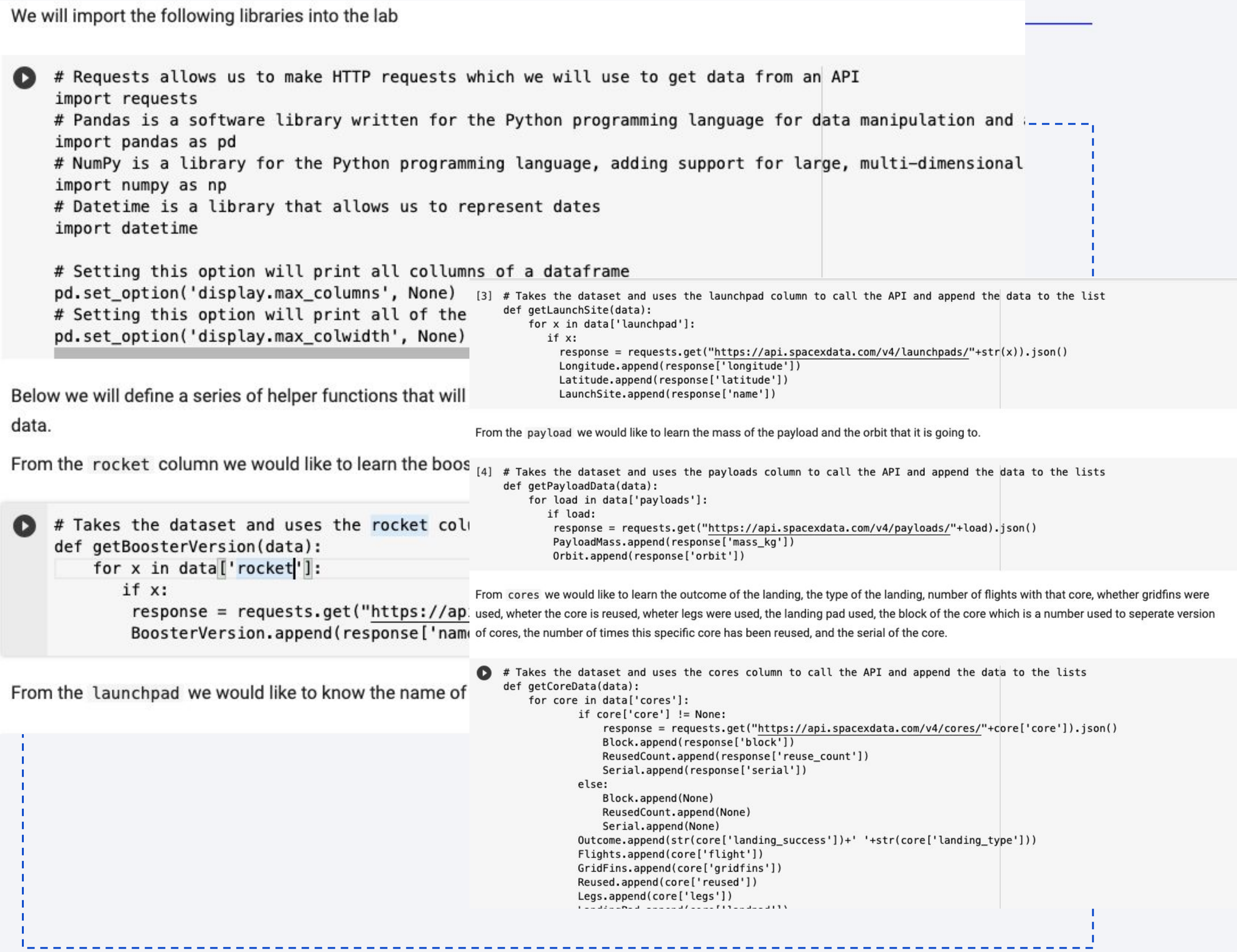
Get HTML response from Wikipedia → Extract data using beautiful soup → Normalize data into csv file

→ Ready for data wrangling

Data Collection – SpaceX API

- Data collection with SpaceX REST SpaceX api

- https://github.com/cahzheng/capstone_spaceX_hz/blob/main/week1_lab1_jupyter_labs_spacex_data_collection_api.ipynb



We will import the following libraries into the lab

```
# Requests allows us to make HTTP requests which we will use to get data from an API
import requests
# Pandas is a software library written for the Python programming language for data manipulation and
import pandas as pd
# NumPy is a library for the Python programming language, adding support for large, multi-dimensional
import numpy as np
# Datetime is a library that allows us to represent dates
import datetime

# Setting this option will print all collumns of a dataframe
pd.set_option('display.max_columns', None)
# Setting this option will print all of the
pd.set_option('display.max_colwidth', None)
```

Below we will define a series of helper functions that will

data.

From the `rocket` column we would like to learn the boos

```
# Takes the dataset and uses the rocket colu
def getBoosterVersion(data):
    for x in data['rocket']:
        if x:
         response = requests.get("https://ap
         BoosterVersion.append(response['nam
```

From the `launchpad` we would like to know the name of

```
[3] # Takes the dataset and uses the launchpad column to call the API and append the data to the list
    def getLaunchSite(data):
        for x in data['launchpad']:
           if x:
             response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).json()
             Longitude.append(response['longitude'])
             Latitude.append(response['latitude'])
             LaunchSite.append(response['name'])
```

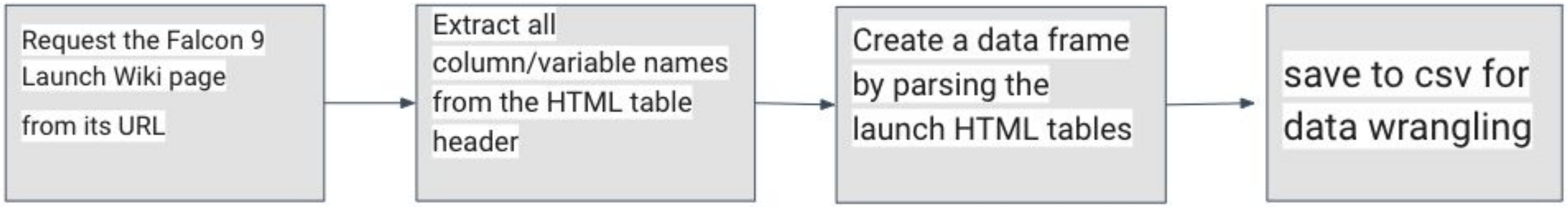
From the `payload` we would like to learn the mass of the payload and the orbit that it is going to.

```
[4] # Takes the dataset and uses the payloads column to call the API and append the data to the lists
    def getPayloadData(data):
        for load in data['payloads']:
           if load:
            response = requests.get("https://api.spacexdata.com/v4/payloads/"+load).json()
            PayloadMass.append(response['mass_kg'])
            Orbit.append(response['orbit'])
```

From `cores` we would like to learn the outcome of the landing, the type of the landing, number of flights with that core, whether gridfins were used, wheter the core is reused, wheter legs were used, the landing pad used, the block of the core which is a number used to seperate version of cores, the number of times this specific core has been reused, and the serial of the core.

```
# Takes the dataset and uses the cores column to call the API and append the data to the lists
def getCoreData(data):
    for core in data['cores']:
            if core['core'] != None:
                response = requests.get("https://api.spacexdata.com/v4/cores/"+core['core']).json()
                Block.append(response['block'])
                ReusedCount.append(response['reuse_count'])
                Serial.append(response['serial'])
            else:
                Block.append(None)
                ReusedCount.append(None)
                Serial.append(None)
            Outcome.append(str(core['landing_success'])+' '+str(core['landing_type']))
            Flights.append(core['flight'])
            GridFins.append(core['gridfins'])
            Reused.append(core['reused'])
            Legs.append(core['legs'])
```

Data Collection - Scraping



Request the Falcon 9 Launch Wiki page from its URL

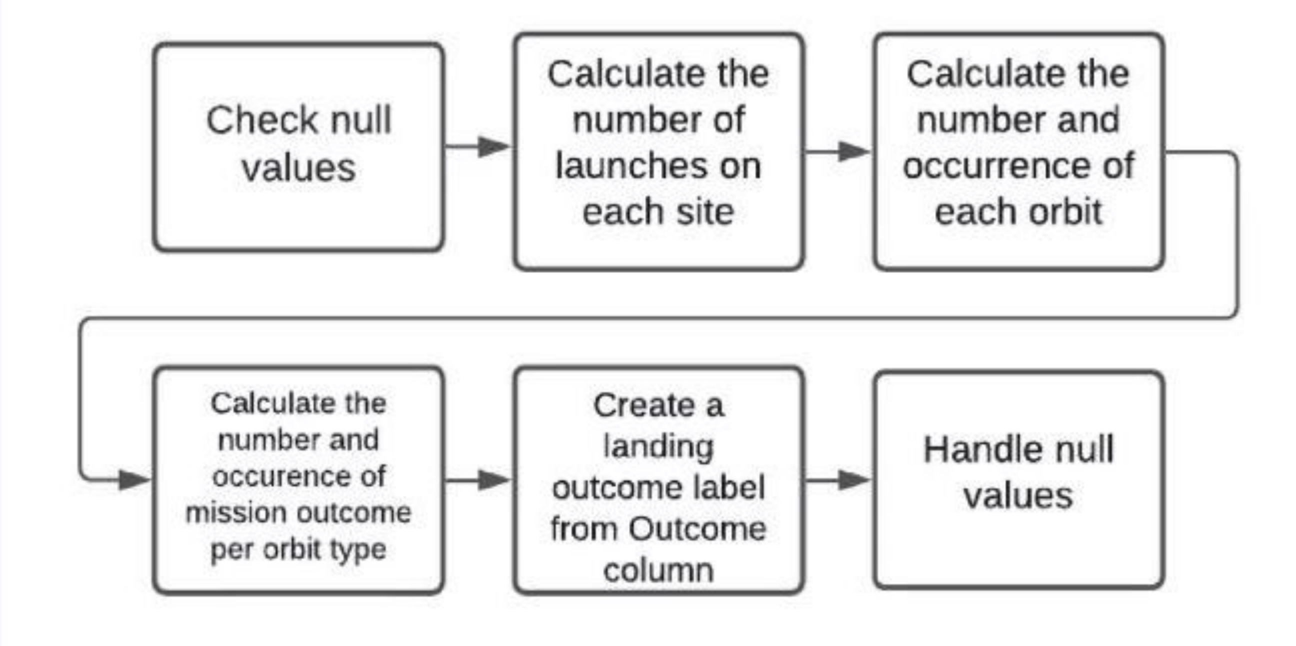
Extract all column/variable names from the HTML table header

Create a data frame by parsing the launch HTML tables

save to csv for data wrangling

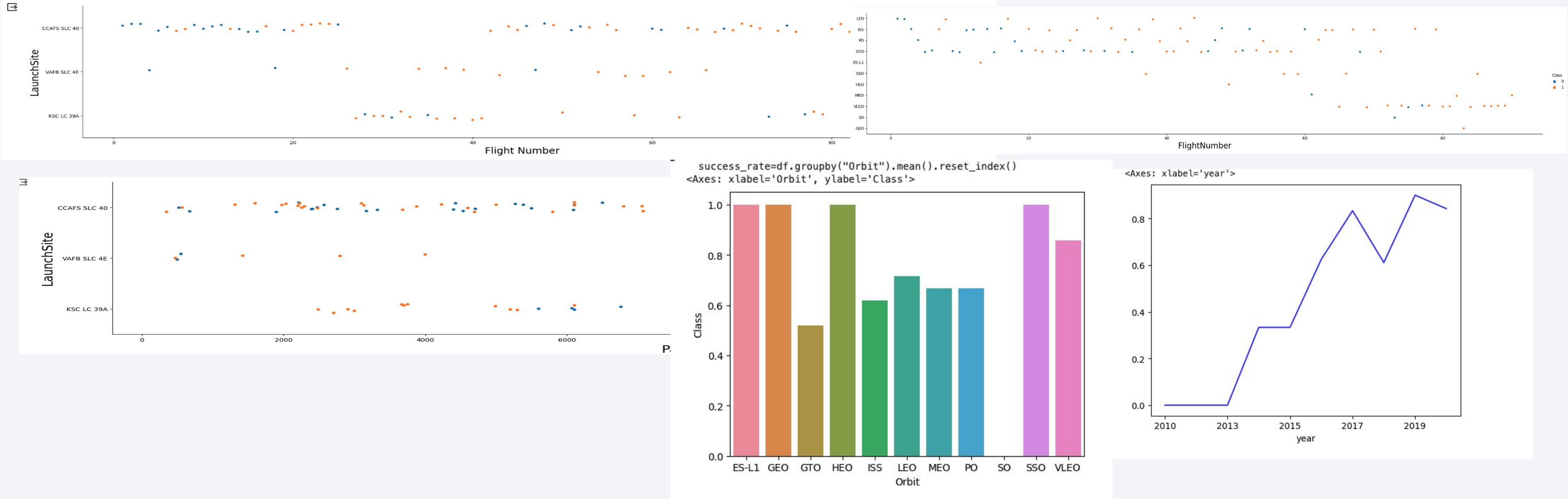
- https://github.com/cahzheng/capstone_spaceX_hz/blob/main/week1_lab1_2_jupyter_spacex_webscraping.ipynb

Data Wrangling


Check null values
Calculate the number of launches on each site
Calculate the number and occurrence of each orbit
Calculate the number and occurence of mission outcome per orbit type
Create a landing outcome label from Outcome column
Handle null values

https://github.com/cahzheng/capstone_spaceX_hz/blob/main/week1_lab2_jupyter_spacex_Data_wrangling.ipynb

EDA with Data Visualization



```
success_rate=df.groupby("Orbit").mean().reset_index()
<Axes: xlabel='Orbit', ylabel='Class'>
```

```
<Axes: xlabel='year'>
```

`Flight Number` vs. `PayloadMass`, `then`, study relation between, Payload and Launch Site,then success rate of each orbit type

then, Flight Number and Orbit type, and then Payload and Orbit type

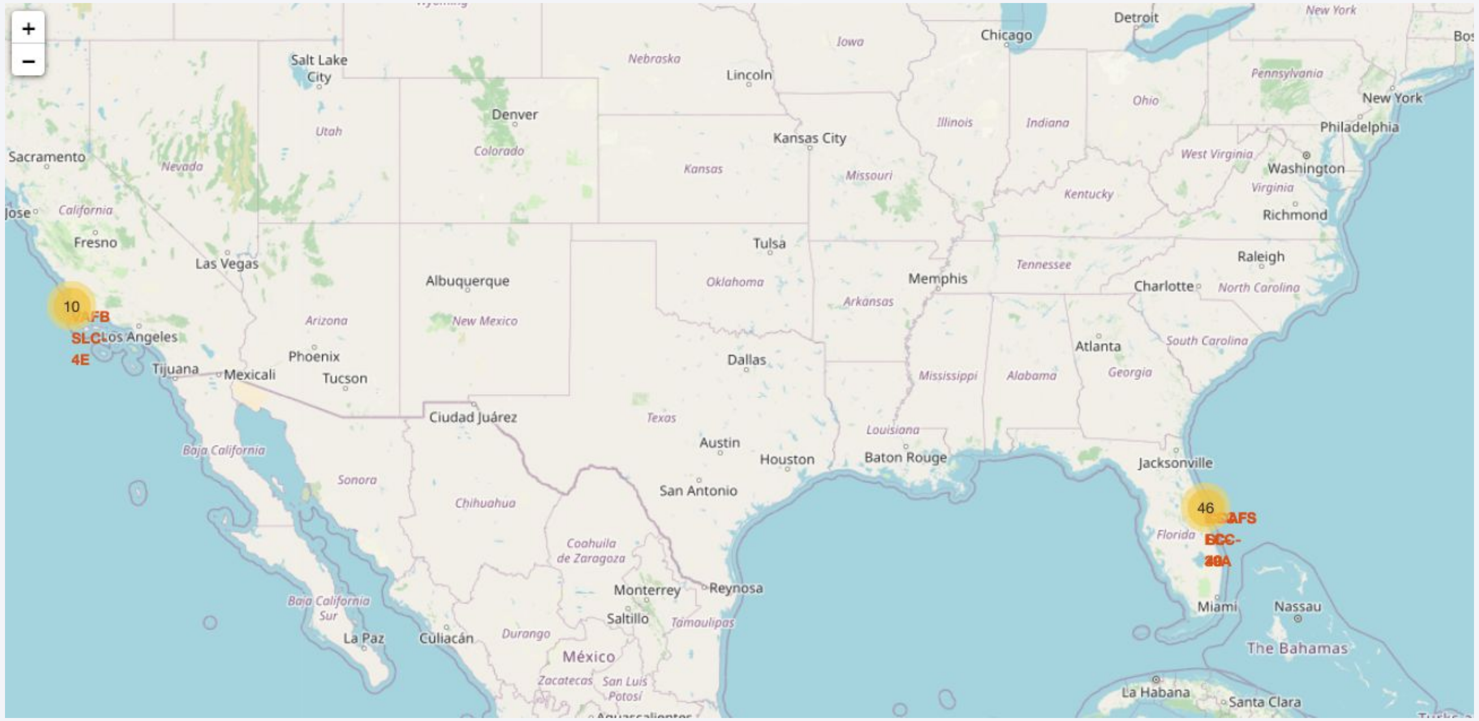
https://github.com/cahzheng/capstone_spaceX_hz/blob/main/week2_lab4_1_jupyter_labs_eda_dataviz.ipynb

EDA with SQL

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

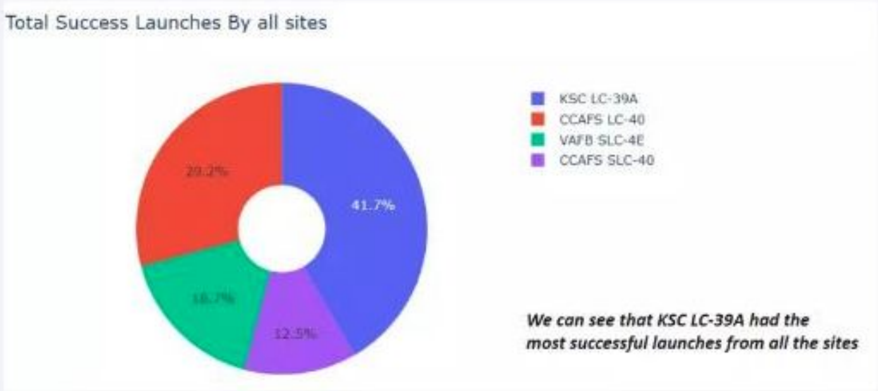
https://github.com/cahzheng/capstone_spaceX_hz/blob/main/week2_jupyter_labs_eda_sql_coursera_sqllite_solution.ipynb

Build an Interactive Map with Folium

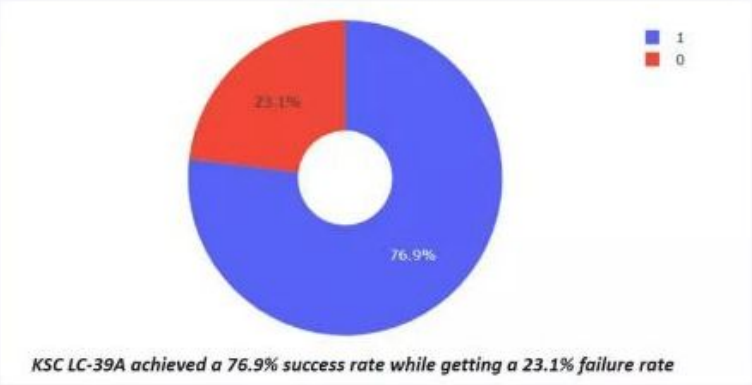


https://github.com/cahzheng/capstone_spaceX_hz/blob/main/week3_lab3_jupyter_Launch_Sites_Locations_Analysis_with_Folium.ipynb

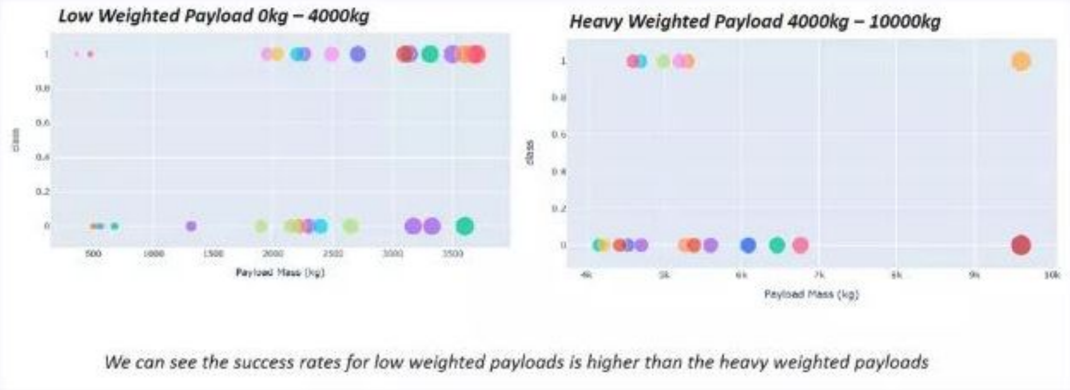
Build a Dashboard with Plotly Dash


Total Success Launches By all sites

We can see that KSC LC-39A had the most successful launches from all the sites



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate


Low Weighted Payload 0kg – 4000kg

Heavy Weighted Payload 4000kg – 10000kg

We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

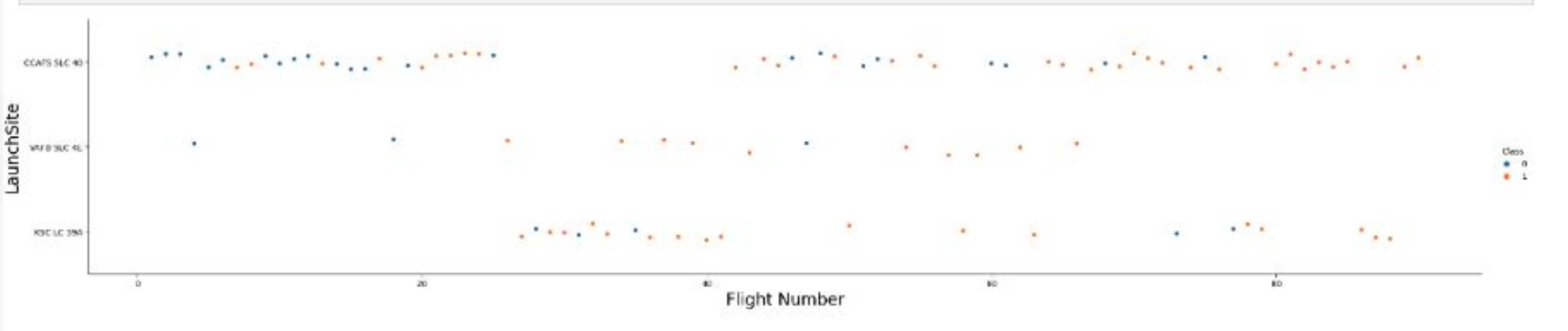
Results

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

Section 2

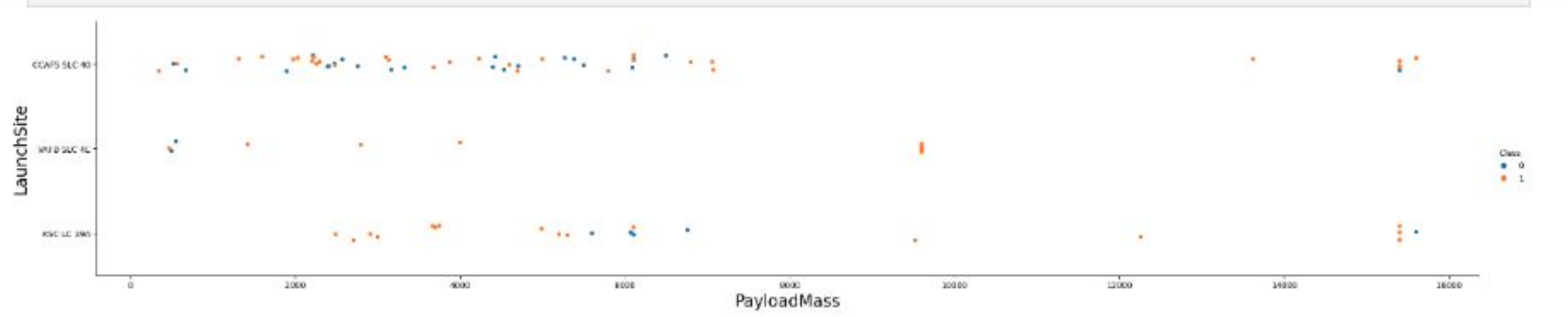
Insights drawn from EDA

Flight Number vs. Launch Site



Launches from CCAFS SLC 40 has higher number than other sites

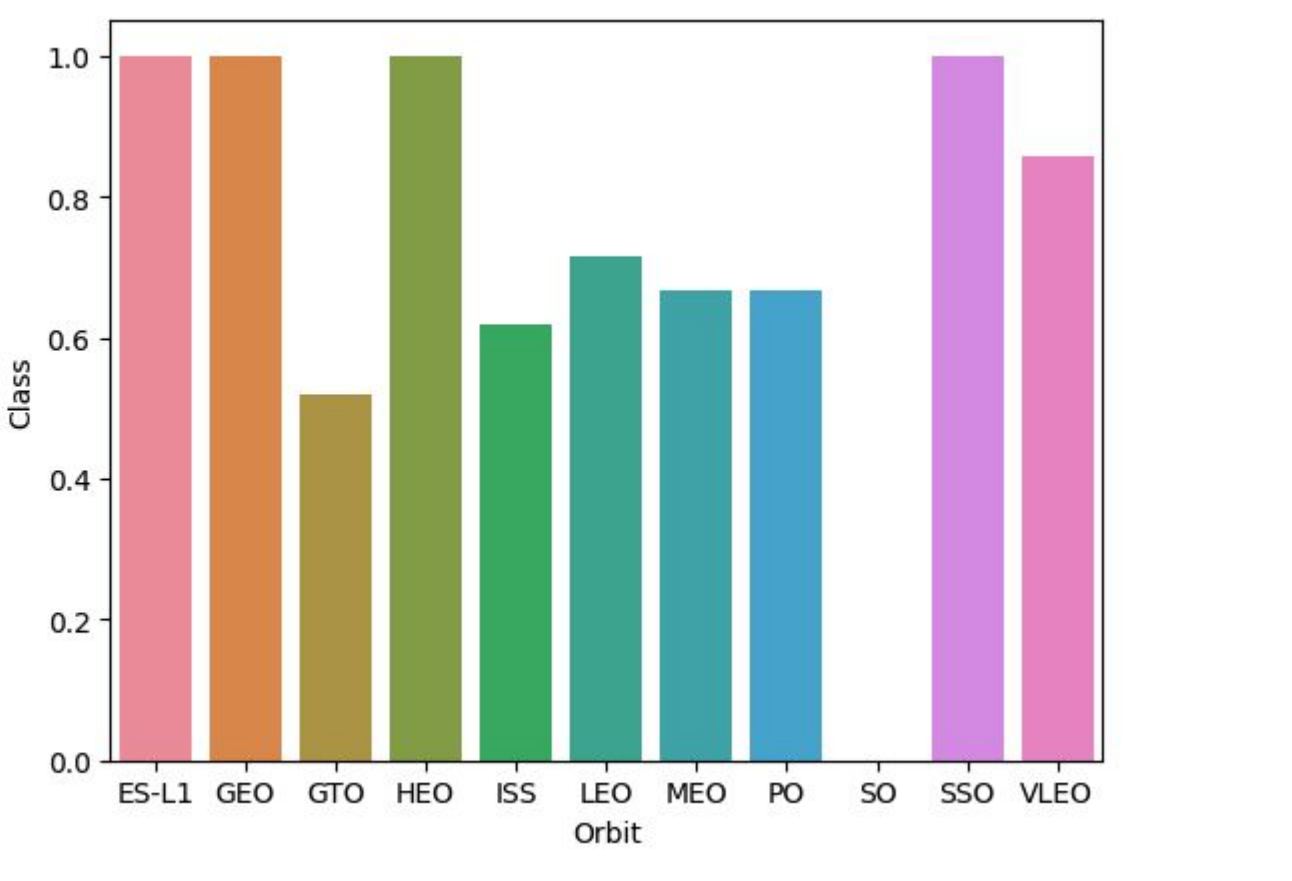
Payload vs. Launch Site



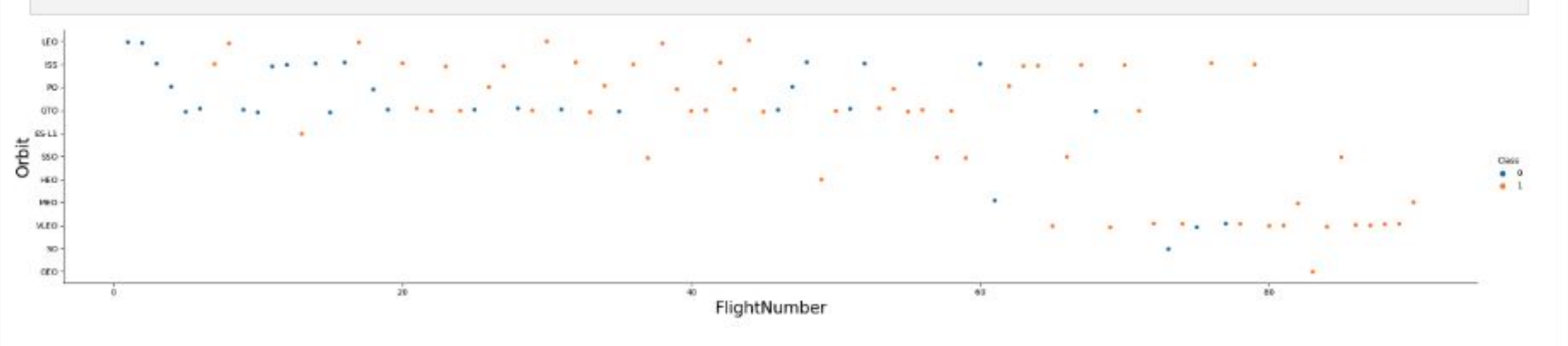
- majority of payloads with lower mass for launches from CCAFS SLC 40

Success Rate vs. Orbit Type

- EL-L1, GEO, HEO and SSO have high success rate

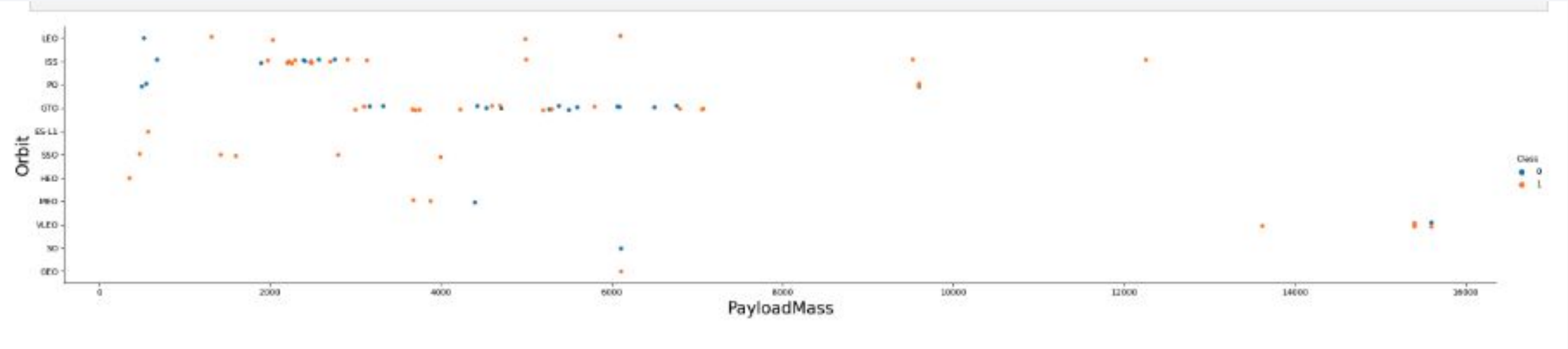


Flight Number vs. Orbit Type



- In recent years more launches shifted to VLEO orbit type

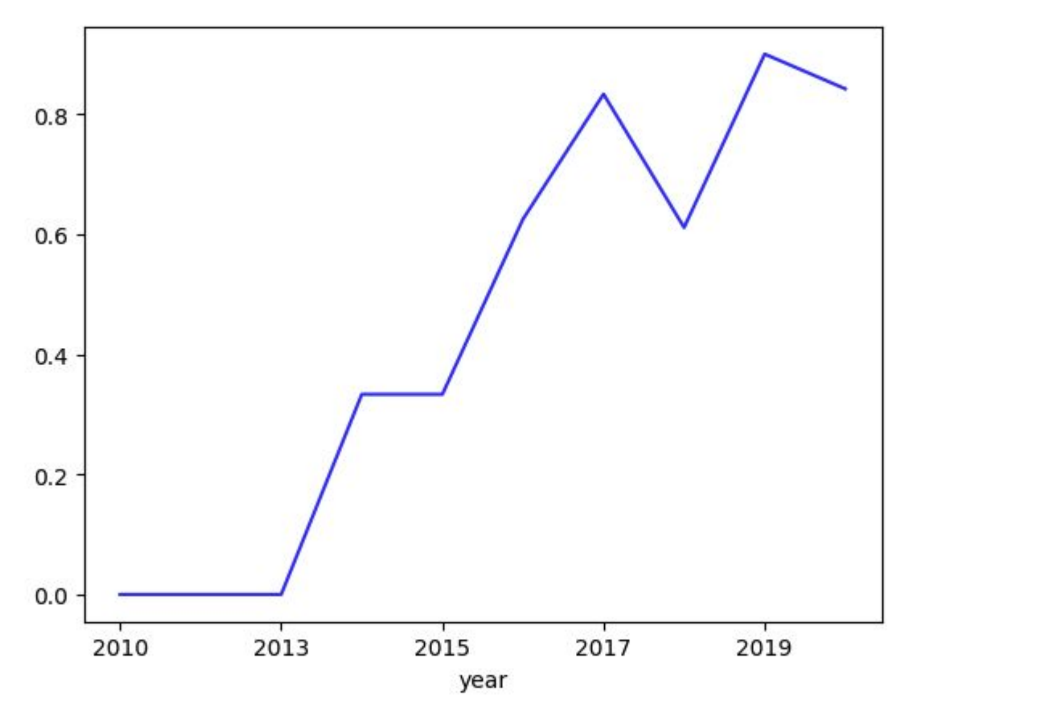
Payload vs. Orbit Type



- VLEO orbit has higher payload than others

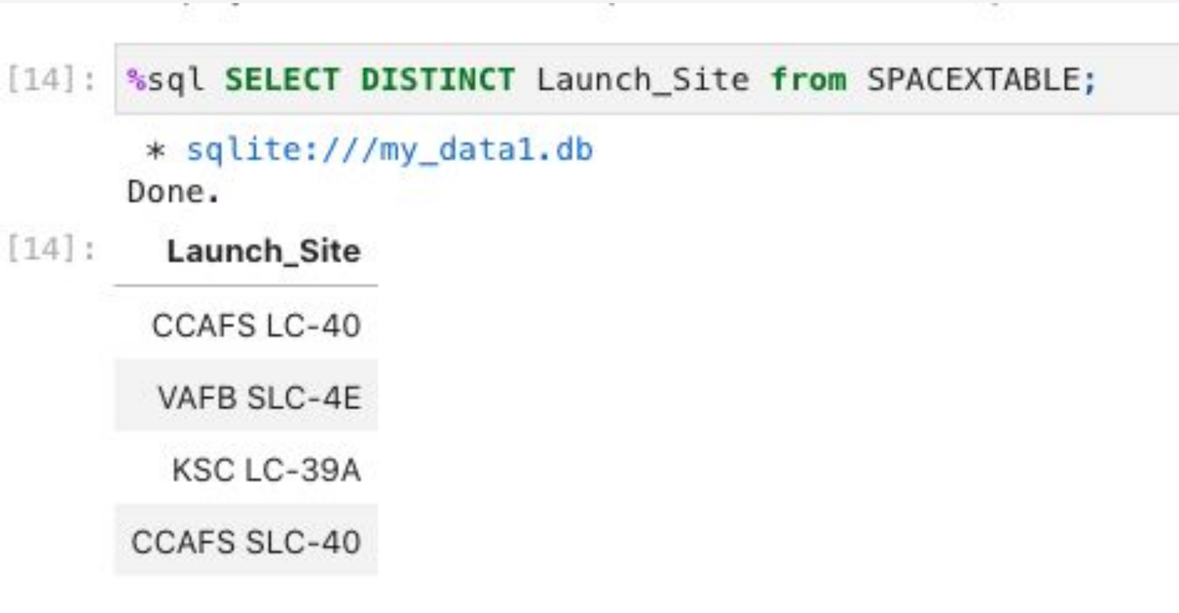
Launch Success Yearly Trend

- The success rate since 2013 kept increasing till 2017 (stable in 2014) and after 2015 it started increasing.



All Launch Site Names

- Use sql query to list the unique launch sites



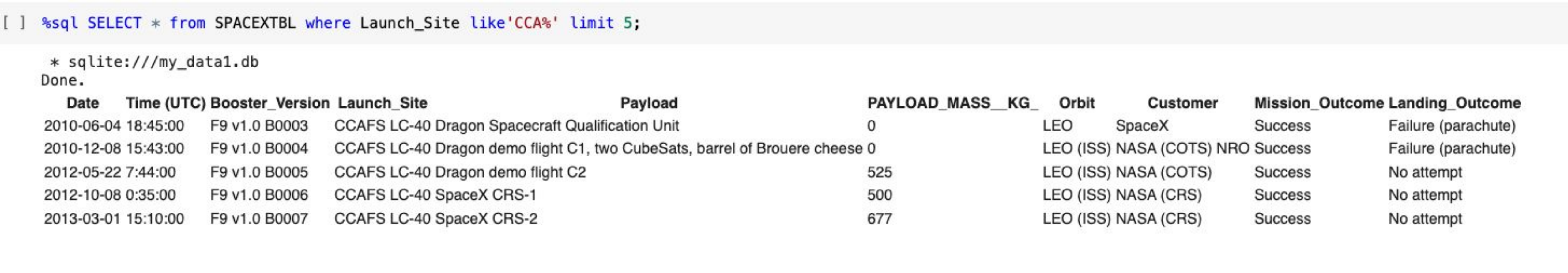
```
[14]: %sql SELECT DISTINCT Launch_Site from SPACEXTABLE;
```

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

[14]:

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

Launch Site Names Begin with 'CCA'



```
[ ] %sql SELECT * from SPACEXTBL where Launch_Site like'CCA%' limit 5;
```

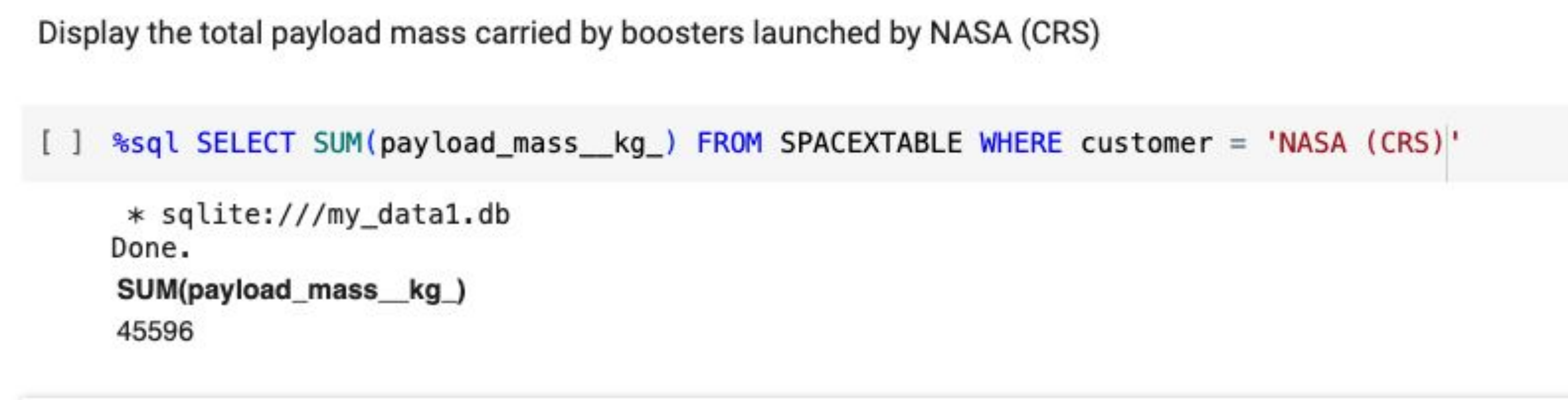
```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
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

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

Use sql query choose launch site name start with CCA

Total Payload Mass



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

```
[ ] %sql SELECT SUM(payload_mass__kg_) FROM SPACEXTABLE WHERE customer = 'NASA (CRS)'
```

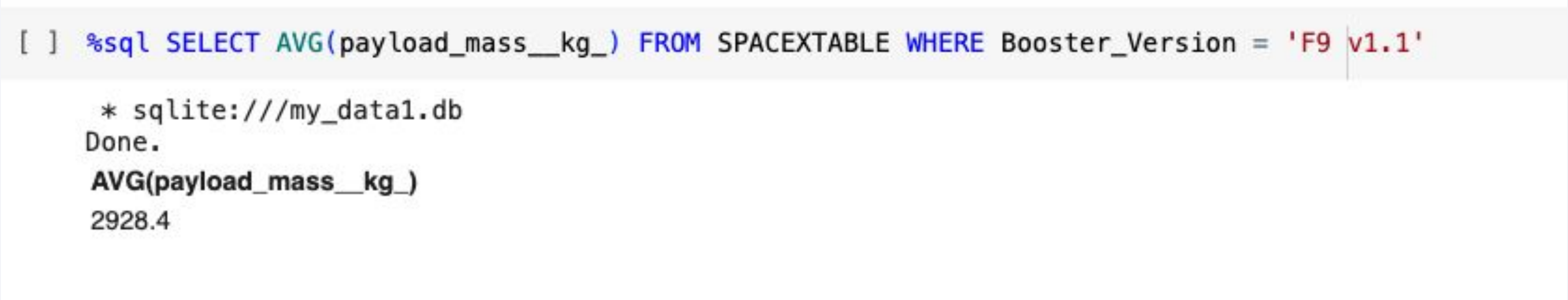
```
 * sqlite:///my_data1.db
Done.
```

SUM(payload_mass__kg_)
45596

- Calculate the total payload carried by boosters from NASA

total mass is 45596 kg,

Average Payload Mass by F9 v1.1



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

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

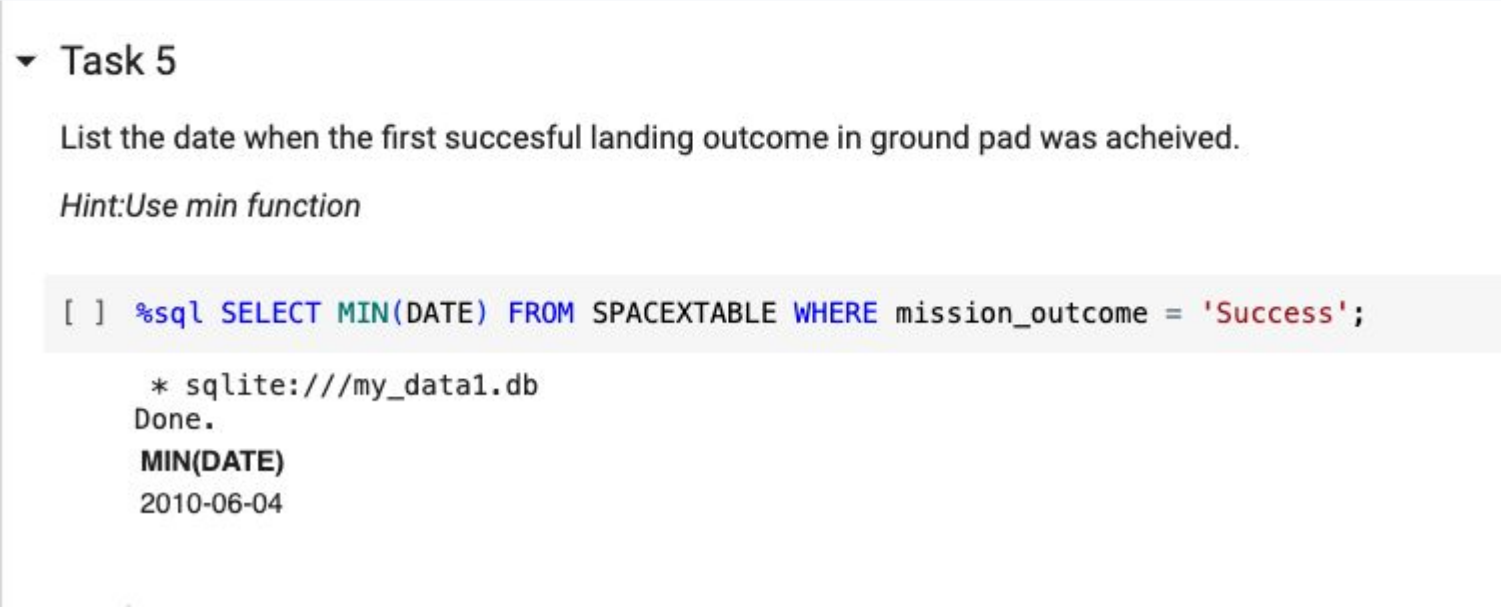
AVG(payload_mass__kg_)

2928.4

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

  2928.4 kg

First Successful Ground Landing Date



Task 5

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

Hint:Use min function

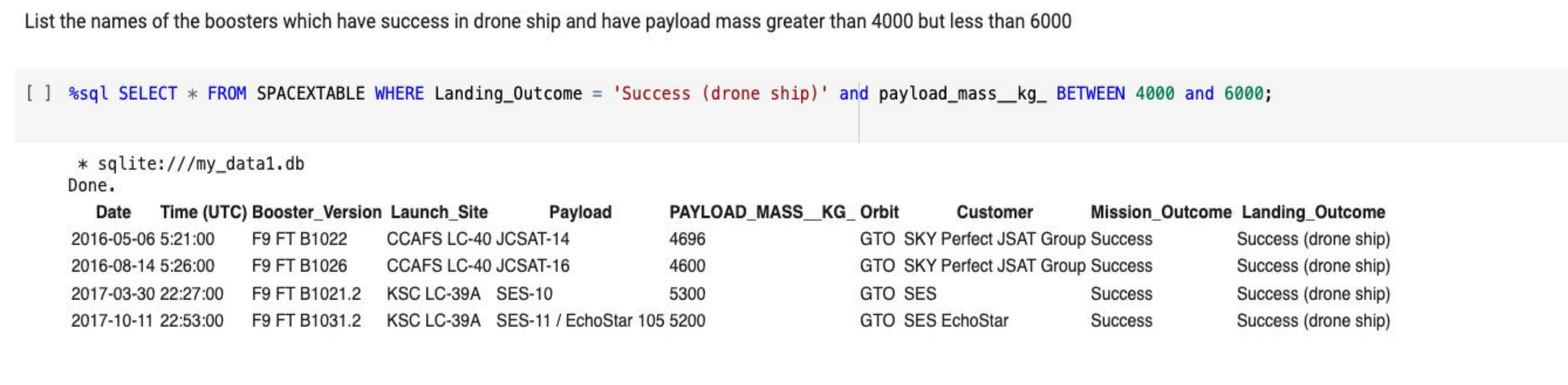
```
[ ] %sql SELECT MIN(DATE) FROM SPACEXTABLE WHERE mission_outcome = 'Success';
```

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

MIN(DATE)
2010-06-04

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

Successful Drone Ship Landing with Payload between 4000 and 6000



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

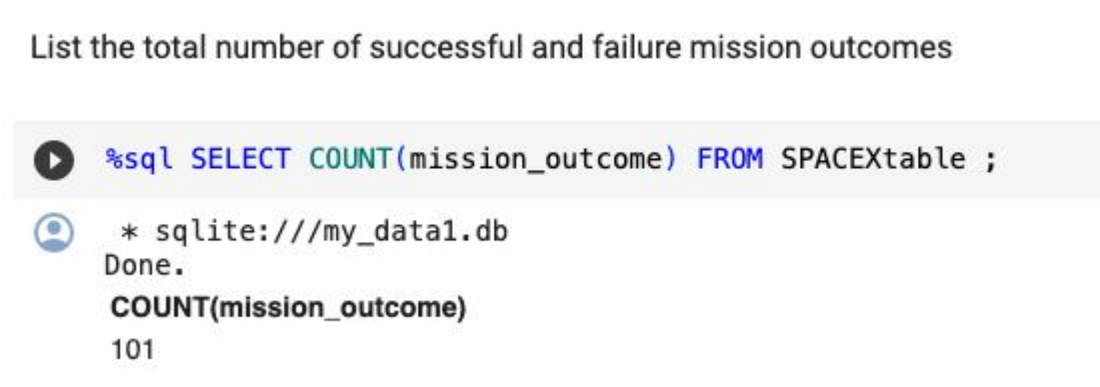
```
[ ] %sql SELECT * FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' and payload_mass__kg_ BETWEEN 4000 and 6000;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2016-05-06	5:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-08-14	5:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-10-11	22:53:00	F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105	5200	GTO	SES EchoStar	Success	Success (drone ship)

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

Total Number of Successful and Failure Mission Outcomes



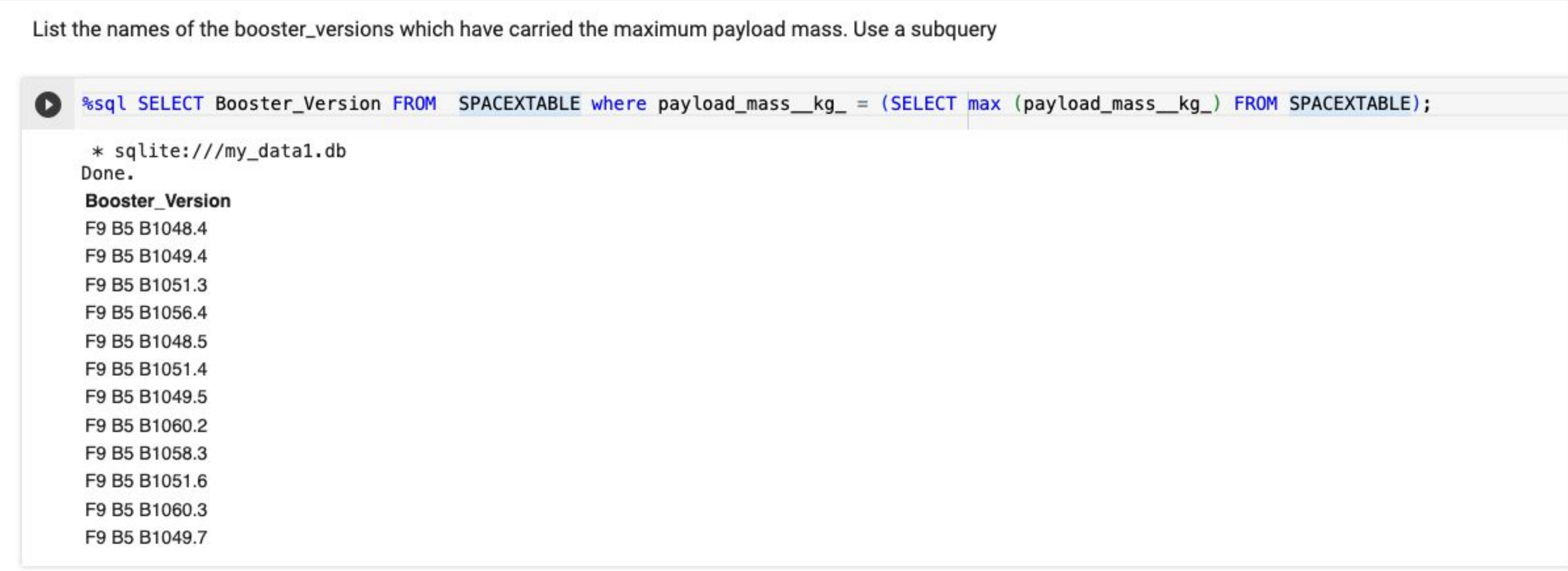
List the total number of successful and failure mission outcomes

```
%sql SELECT COUNT(mission_outcome) FROM SPACEXtable ;
```

```
 * sqlite:///my_data1.db
Done.
COUNT(mission_outcome)
101
```

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here

Boosters Carried Maximum Payload



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

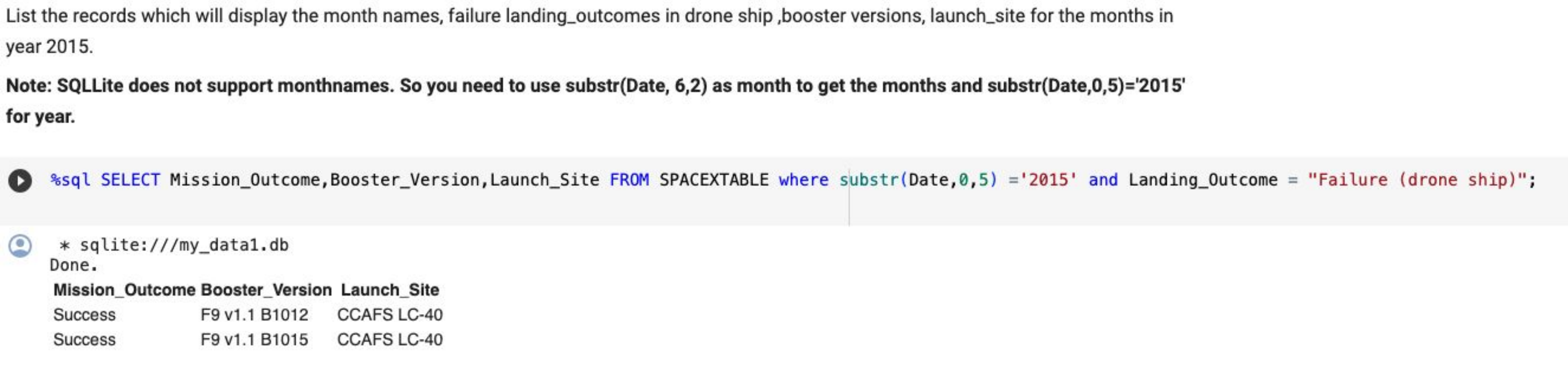
```
%sql SELECT Booster_Version FROM  SPACEXTABLE where payload_mass__kg_ = (SELECT max (payload_mass__kg_) FROM SPACEXTABLE);
```

```
 * 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

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

2015 Launch Records



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.

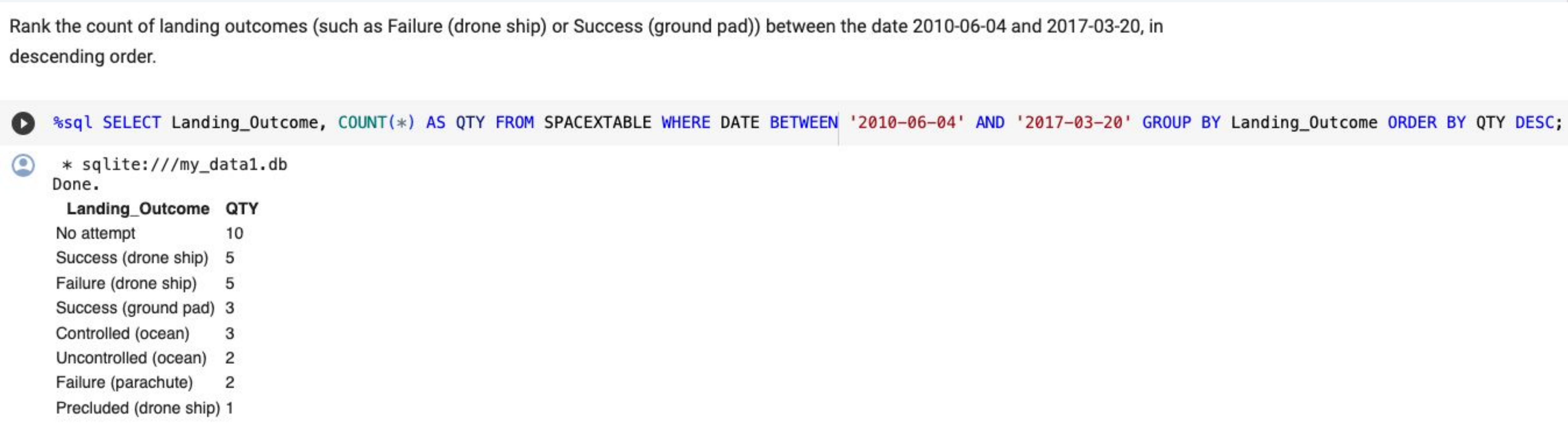
Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%sql SELECT Mission_Outcome,Booster_Version,Launch_Site FROM SPACEXTABLE where substr(Date,0,5) ='2015' and Landing_Outcome = "Failure (drone ship)";
```

```
 * sqlite:///my_data1.db
Done.
Mission_Outcome Booster_Version Launch_Site
Success         F9 v1.1 B1012   CCAFS LC-40
Success         F9 v1.1 B1015   CCAFS LC-40
```

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

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 QTY FROM SPACEXTABLE WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY QTY DESC;
```

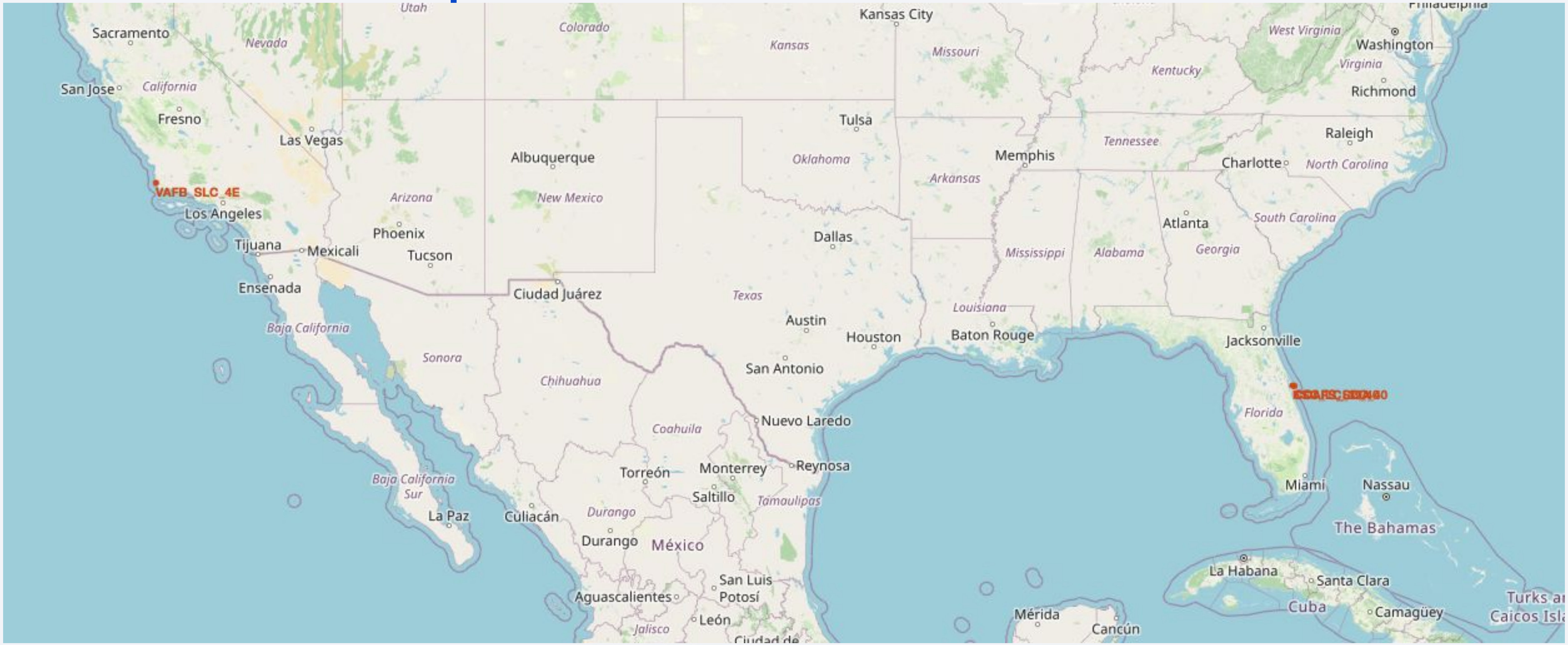
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	QTY
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

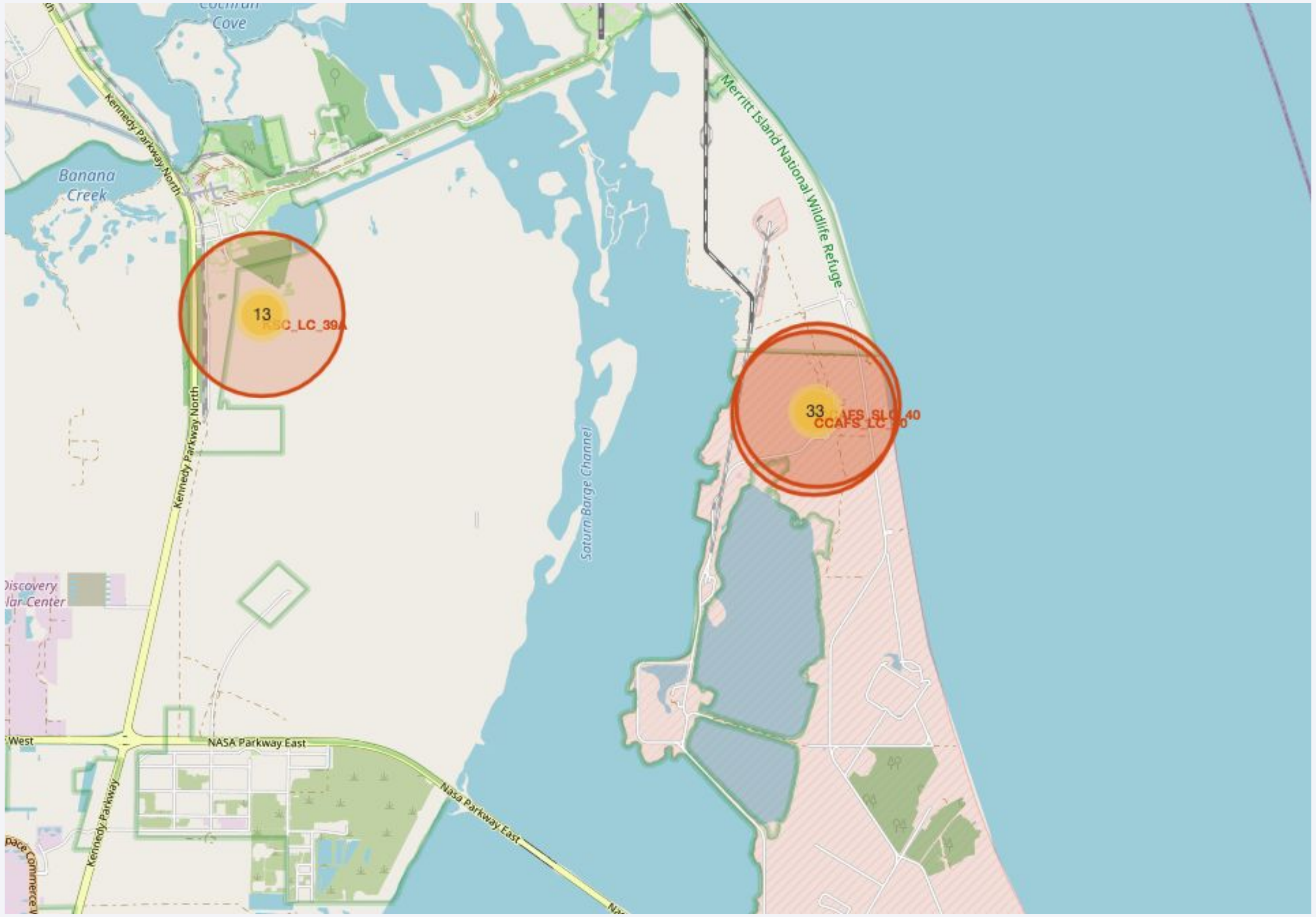
Section 3

Launch Sites Proximities Analysis

<Folium Map : Mark all launch sites>

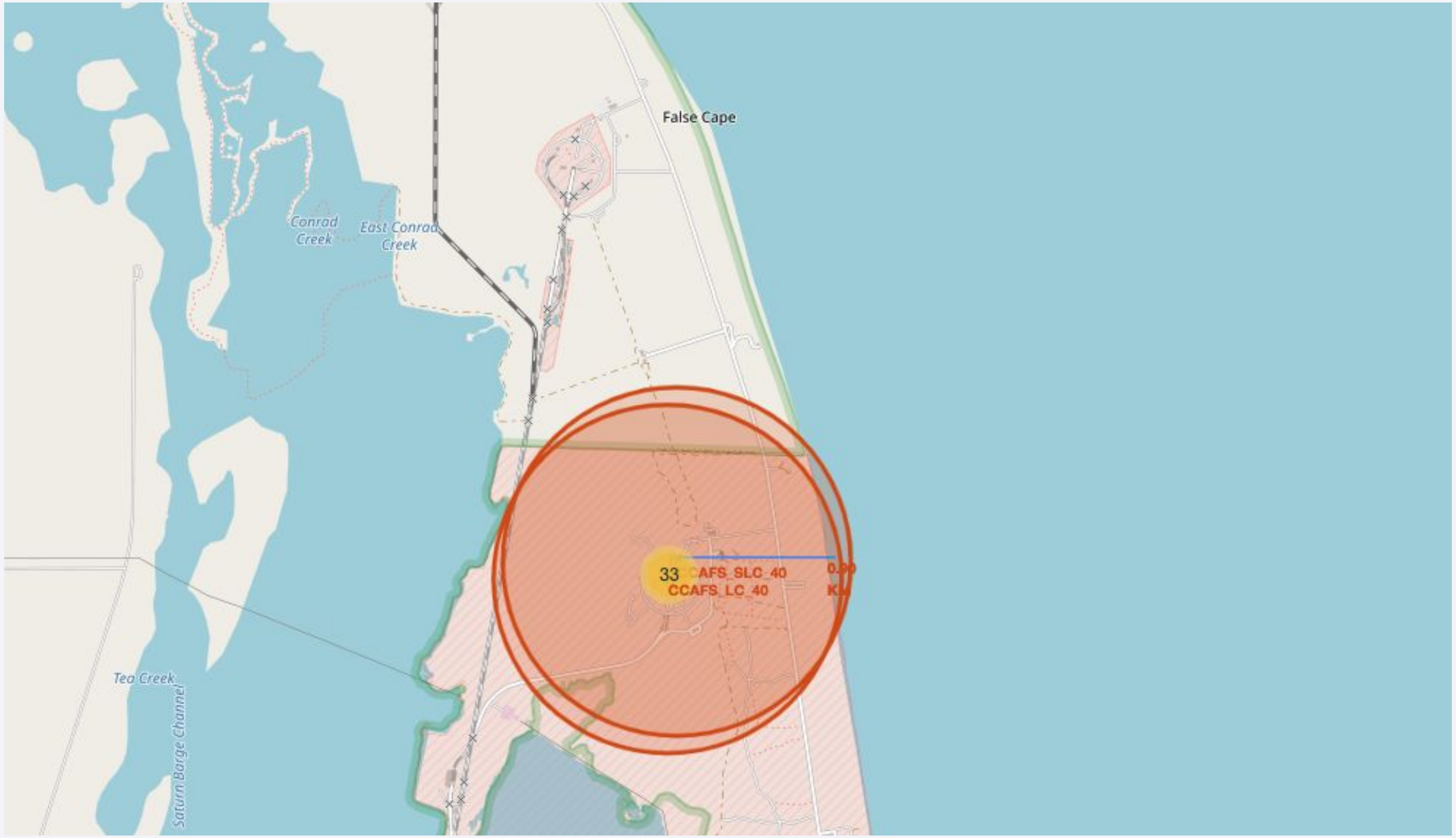

VAFB SLC 4E
Sacramento
San Jose
California
Fresno
Nevada
Utah
Colorado
Kansas
Kansas City
Missouri
West Virginia
Washington
Virginia
Kentucky
Richmond
Las Vegas
Tulsa
Albuquerque
Oklahoma
Memphis
Tennessee
Raleigh
Charlotte
North Carolina
Arkansas
Arizona
New Mexico
Los Angeles
Phoenix
Tijuana
Mexicali
Tucson
Dallas
Atlanta
South Carolina
Mississippi
Alabama
Georgia
Ensenada
Ciudad Juárez
Texas
Louisiana
Baja California
Austin
Houston
Baton Rouge
Jacksonville
Sonora
San Antonio
Chihuahua
Florida
Nuevo Laredo
Coahuila
Torreón
Monterrey
Reynosa
Baja California Sur
Saltillo
Tamaulipas
Miami
Nassau
La Paz
Culiacán
Durango
The Bahamas
Durango
México
La Habana
San Luis Potosí
Santa Clara
Aguascalientes
Cuba
Mérida
Camagüey
León
Cancún
Jalisco
Turks ar
Caicos Isla

<Folium Map: color labeled launch sites>

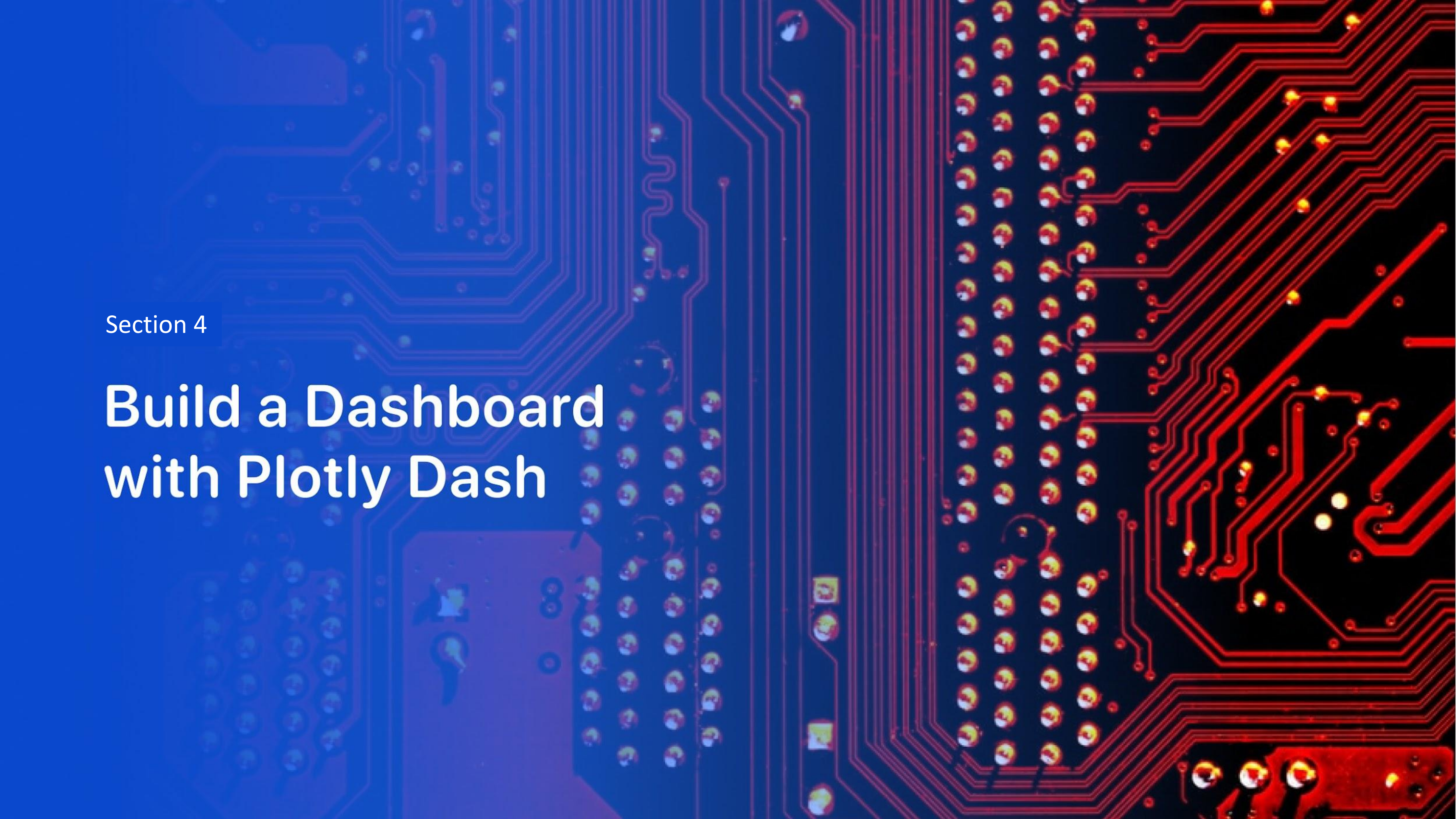


Cochran Cove
Banana Creek
Kennedy Parkway North
13
KSC_LC_39A
Merritt Island National Wildlife Refuge
33
CCAFS_SLC_40
CCAFS_LC_40
Saturn Barge Channel
Discovery
lar Center
West
NASA Parkway East
Nasa Parkway East
Kennedy Parkway

<Folium Map: launch site to coast line>



False Cape
Conrad Creek
East Conrad Creek
Tea Creek
Saturn Barge Channel
33
CCAFS_SLC_40
CCAFS_LC_40
0.90
KM



Section 4

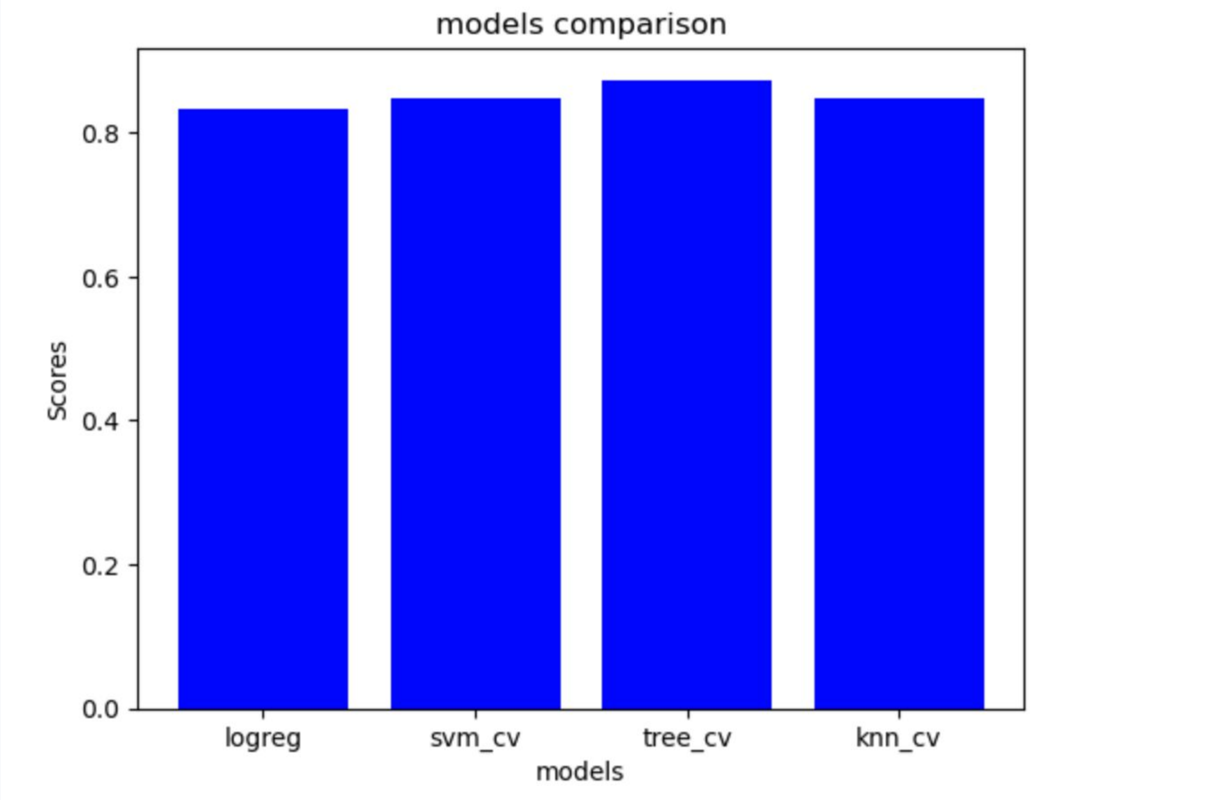
Build a Dashboard with Plotly Dash

Section 5

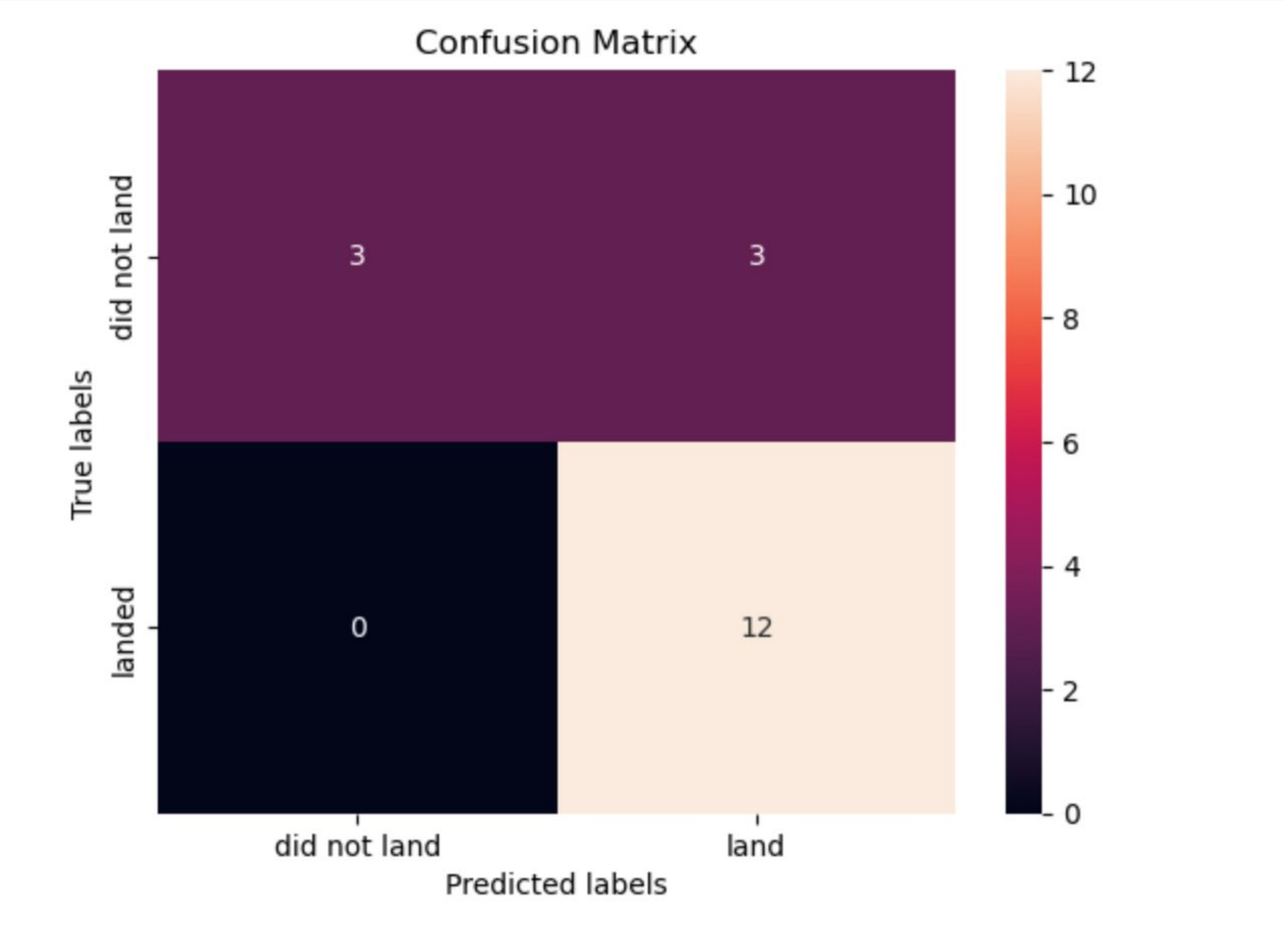
Predictive Analysis (Classification)

Classification Accuracy

- decision tree model has hughes accuracy



Confusion Matrix



In this model, 3 failed landing out of 6 was predicted right, 12 success landings out of 12 was predicted right. total accuracy is 15/18, is the best accuracy in all four models

Conclusions

- Four models were built
- The decision tree model gives the best accuracy

Appendix

- https://github.com/cahzheng/capstone_spaceX_hz/blob/main/week1_lab1_jupyter_labs_spacex_data_collection_api.ipynb
-
- https://github.com/cahzheng/capstone_spaceX_hz/blob/main/week1_lab1_2_jupyter_spacex_webscraping.ipynb
- https://github.com/cahzheng/capstone_spaceX_hz/blob/main/week1_lab2_jupyter_spacex_Data_wrangling.ipynb
- https://github.com/cahzheng/capstone_spaceX_hz/blob/main/week2_lab4_1_jupyter_labs_eda_dataviz.ipynb
- https://github.com/cahzheng/capstone_spaceX_hz/blob/main/week2_jupyter_labs_eda_sql_coursera_sqllite_solution.ipynb
- https://github.com/cahzheng/capstone_spaceX_hz/blob/main/week3_lab3_jupyter_Launch_Sites_Locations_Analysis_with_Folium.ipynb

Thank you!