IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mohamad Shwaike
1.Jan.2023

Outline

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

Executive Summary

Summary of methodologies:

This presentation will highlight on Data Collection with API and WebScraping and Data Wrangling additional to the Exploratory Data Analysis with SQL and Pandas. Then, how visualizae Exploratory Data Analysis using Matplotlib and using Interactive Visual Analytics and Dashboard. And also predictive Analysis with Machine Learning.

Summary of all results

Exploratory Data Analysis results

Predictive Analysis results

Introduction

Background:

In this capstone, we will predict if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

Problem to Solve:

In this project , our objective is to train machine learning models with publicly available data to predict successful Stage 1 recovery

Section 1

Methodology

Methodology

- The overall methodology includes:
- Data collection methodology:
  - Combined data from SpaceX public API and SpaceX Wikipedia page
- Perform data wrangling
  - Classifying true landings as successful and unsuccessful otherwise
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Tuned models using GridSearchCV

Data Collection

Data collection process involved a combination of API requests from Space X public API and web scraping data from a table in Space X's Wikipedia entry.

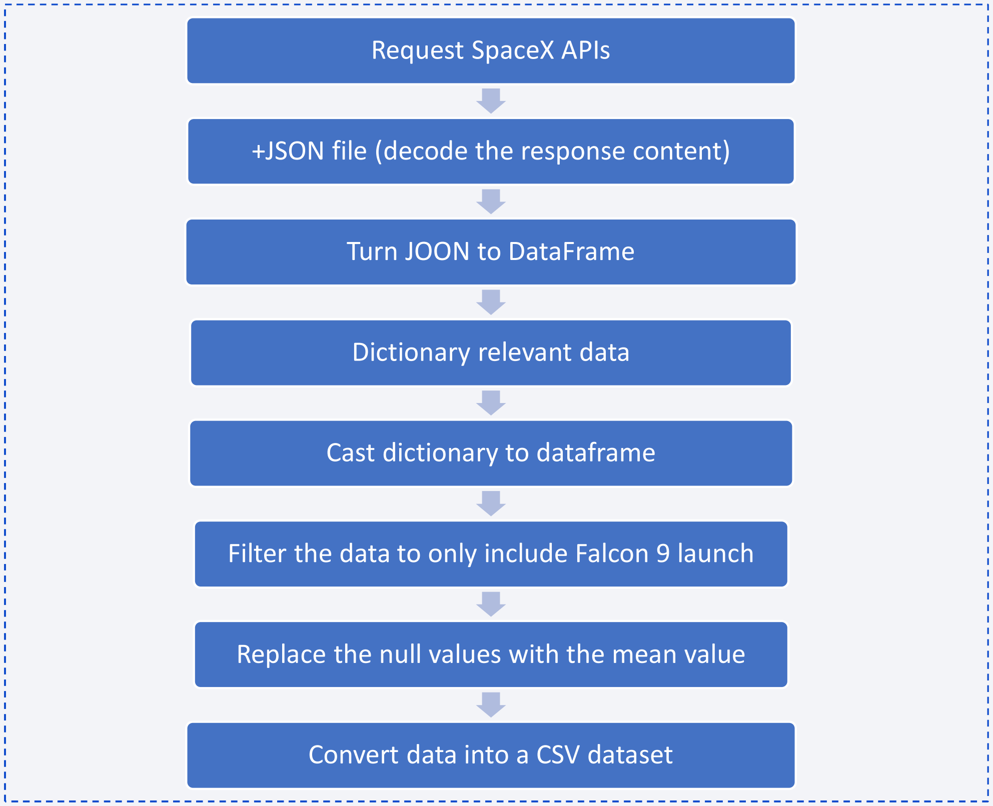
Using the SpaceX API , information was collected as JSON response which later on has been decoded into a Pandas DataFrame.

Data collection process involved a combination of API requests from Space X public API and web scraping data from a table in Space X's Wikipedia entry.

The next slide will show the flowchart of data collection from API and the one after will show the flowchart of data collection from webscraping.

Data Collection – SpaceX API

- Make a request to SpaceX API and make sure the data is in the correct format.
- Perform some basic data wrangling and formatting in order to clean the requested data.
- Convert our data frame into a CSV dataset.
- https://github.com/waelshouayki/DataScienceCapstone/blob/master/jupyter-labs-spacex-data-collection-api.ipynb.ipynb


Request SpaceX APIs

+JSON file (decode the response content)

Turn JOON to DataFrame

Dictionary relevant data

Cast dictionary to dataframe

Filter the data to only include Falcon 9 launch

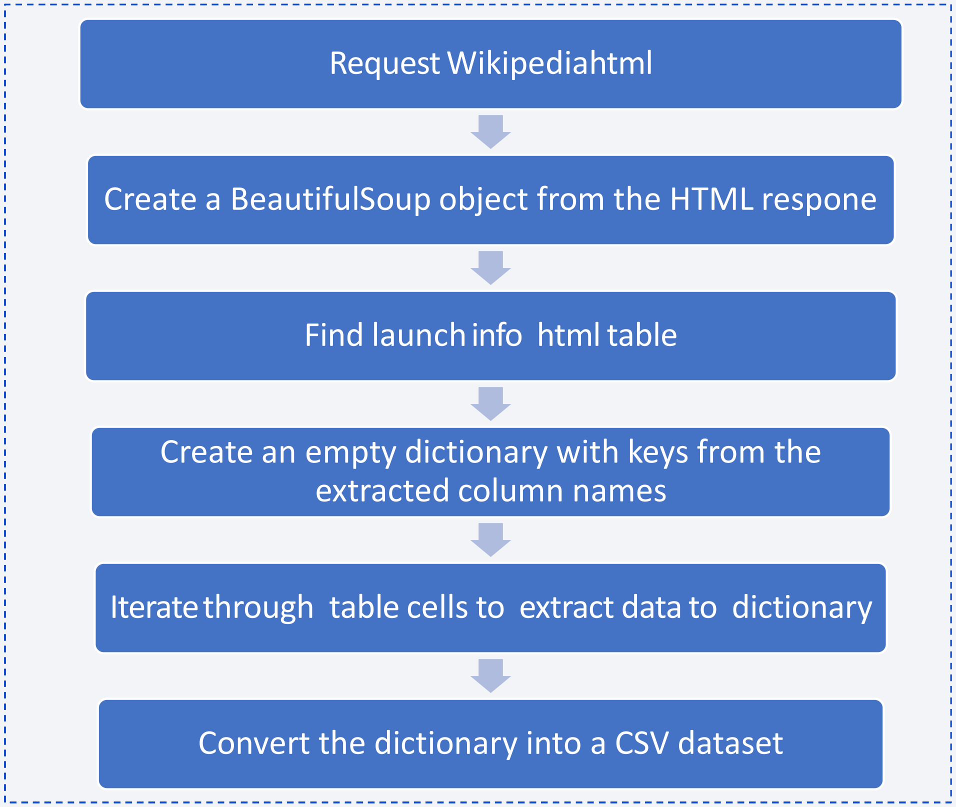
Replace the null values with the mean value

Convert data into a CSV dataset

Data Collection - Scraping

- Using BeautifulSoup, perform web scraping on the wikipedia page with title: List of Falcon 9 and Falcon Heavy launches
- Store the launch records in an HTML table.
- Parse the table and convert it into a CSV dataset Present your web scraping process using key phrases and flowcharts

- https://github.com/waelshouayki/DataScienceCapstone/blob/master/jupyter-labs-webscraping.ipynb



Request Wikipediahtml

Create a BeautifulSoup object from the HTML respone

Find launch info html table

Create an empty dictionary with keys from the extracted column names

Iterate through table cells to extract data to dictionary

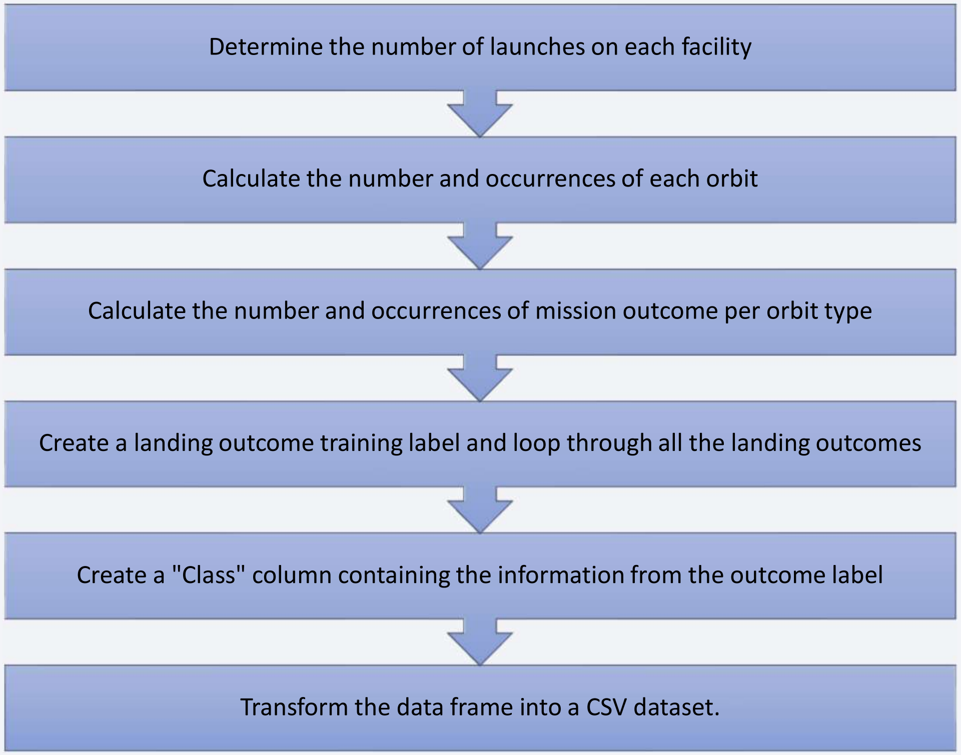
Convert the dictionary into a CSV dataset

Data Wrangling

The goal in this stage is to find patterns in the data and determine the label for training supervised machine learning models.

Create a training label with landing outcomes where successful = 1 & failure = 0.

https://github.com/waelshouayki/DataScienceCapstone/blob/master/labs-jupyter-spacex-Datawrangling.ipynb



Determine the number of launches on each facility

Calculate the number and occurrences of each orbit

Calculate the number and occurrences of mission outcome per orbit type

Create a landing outcome training label and loop through all the landing outcomes

Create a "Class" column containing the information from the outcome label

Transform the data frame into a CSV dataset.

EDA with Data Visualization

- Data visualization helps us understand data by curating it into a form that's easier to understand, highlighting the trends and outliers. Several types of charts were used in the visualization of the data:
- A bar chart was used to visualize the success rate of each orbit type.
- A plot chart was used to visualize the relationship between FlightNumber and Orbit type
- A line chart was used to visualize the launch success yearly trend https://github.com/waelshouayki/DataScienceCapstone/blob/master/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Loaded data set into IBM DB2 Database.
- Queried using SQL Python integration.
- Queries were made to get a better understanding of the dataset.
- Queried information about launch site names, mission outcomes, various pay load sizes of customers and booster versions, and landing outcomes

Build an Interactive Map with Folium

- Folium Markers were used to show the SpaceX launch sites and their nearest important landmarks like railways, highways, cities and coastlines.
- Polylines were used to connect the launch sites to their nearest land marks.
- Furthermore, Folium Circles were used to highlight circle area of launch sites.
- In order to mark the success/failed launches for each site, marker clusters were used on the map. Whereby Red represents rocket launch failures while Green represents the successes.
- https://github.com/waelshouayki/DataScienceCapstone/blob/master/launch_site_location_with_Folium.ipynb

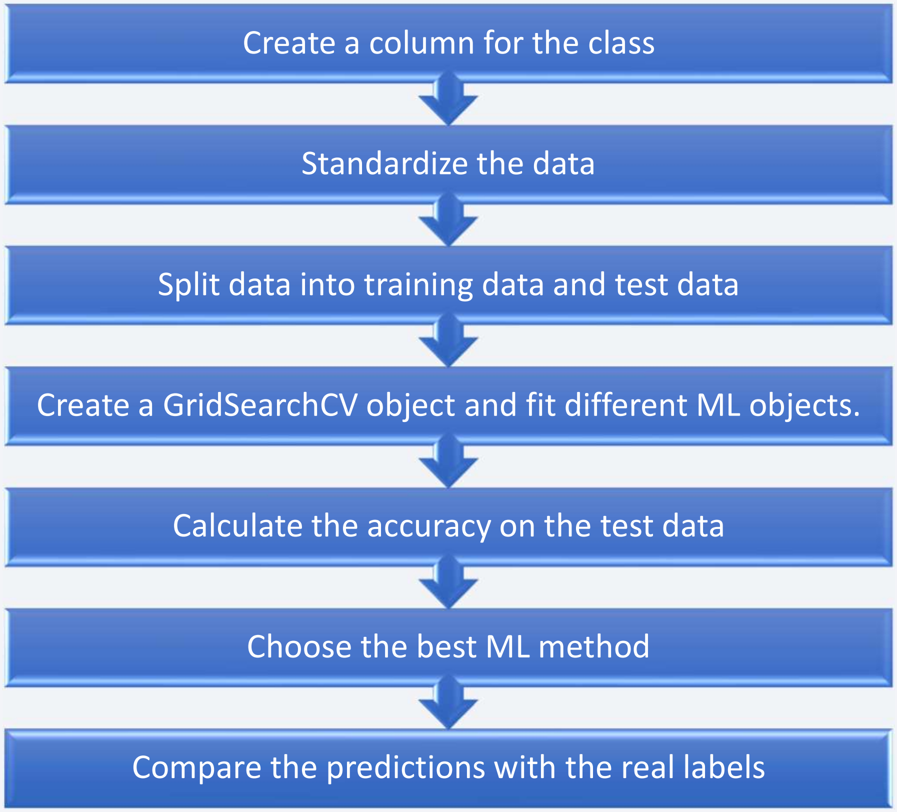
Build a Dashboard with Plotly Dash

- Dashboard includes a pie chart and a scatter plot.
- Pie chart can be selected to show distribution of successful landings across all launch sites and can be selected to show individual launch site success rates.
- Scatter plot takes two inputs: All sites or individual site and payload mass on a slider between 0 and 10000 kg.
- The pie chart is used to visualize launch site success rate.
- The scatter plot can help us see how success varies across launch sites, payload mass, and
- booster version category.

- https://github.com/waelshouayki/DataScienceCapstone/blob/master/spacex_dash_app.py

Predictive Analysis (Classification)

- https://github.com/waelshouayki/DataScienceCapstone/blob/master/Machine_Learning_Prediction.ipynb



Create a column for the class

Standardize the data

Split data into training data and test data

Create a GridSearchCV object and fit different ML objects.

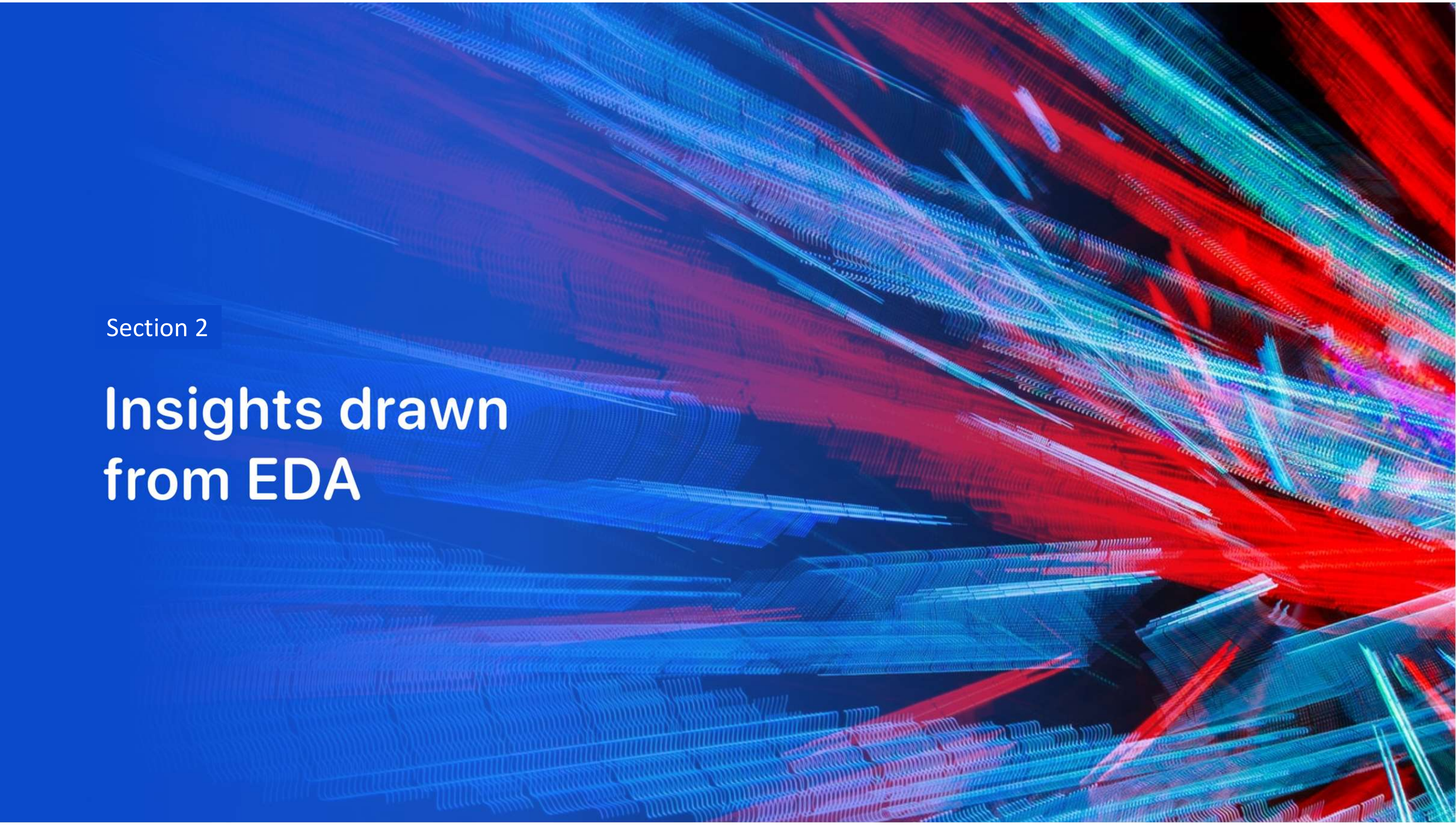
Calculate the accuracy on the test data

Choose the best ML method

Compare the predictions with the real labels

Results

- The exploratory data analysis has shown us that successful landing outcomes are somewhat correlated with flight number. It was also apparent that successful landing outcomes have had a significant increase since the year 2015.
- All launch sites are located near the coast line. Perhaps, this makes it easier to test rocket landings in the water.
- Furthermore, the sites are also located near highways and railways. This may facilitate transportation of equipment and research material.
- The machine learning models that were built, were able to predict the landing success of rockets with an accuracy score of 83.33%. This accuracy can be increased in future projects with more data.

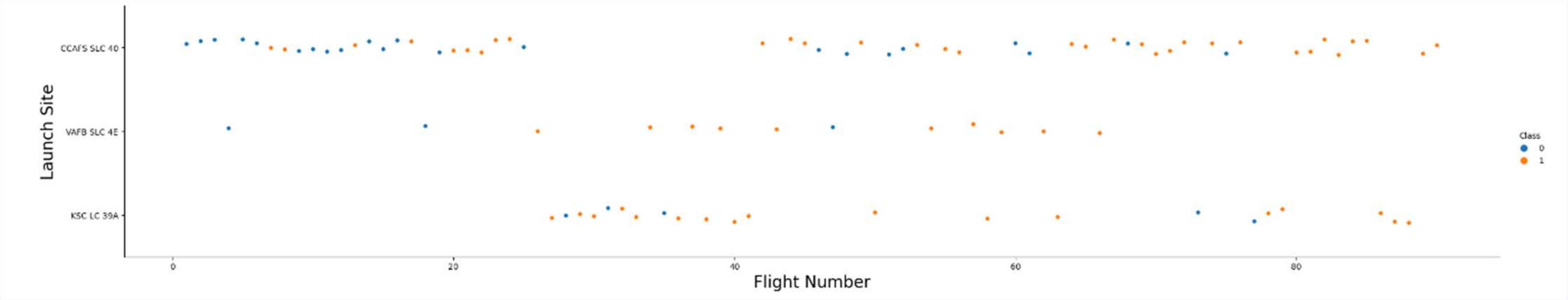


Section 2

Insights drawn from EDA

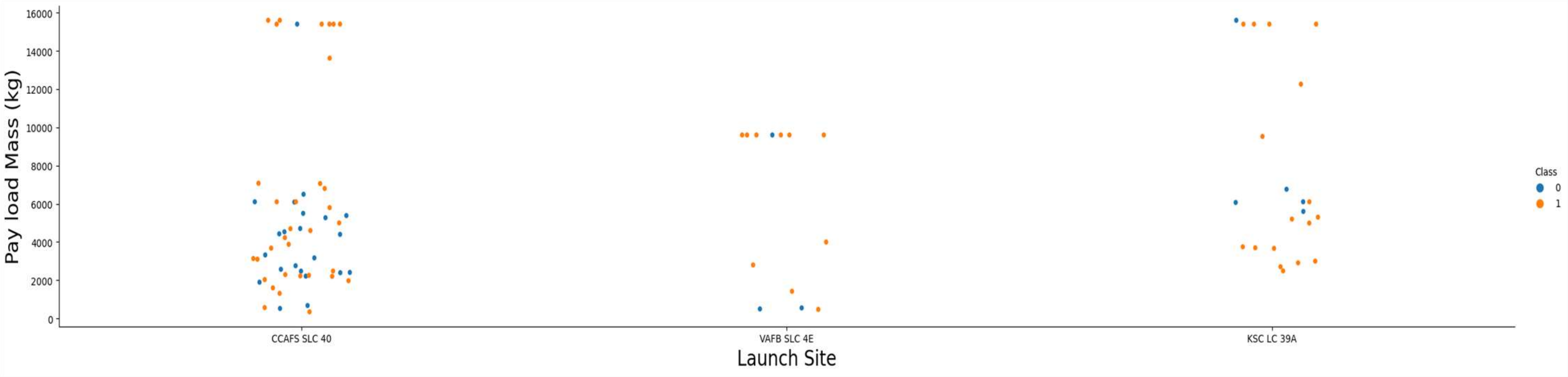
Flight Number vs. Launch Site

- Green indicates successful launch; Purple indicates unsuccessful launch.
- Graphic suggests an increase in success rate over time (indicated in Flight Number).
- Likely a big breakthrough around flight 20 which significantly increased success rate.
- CCAFS appears to be the main launch site as it has the most volume.



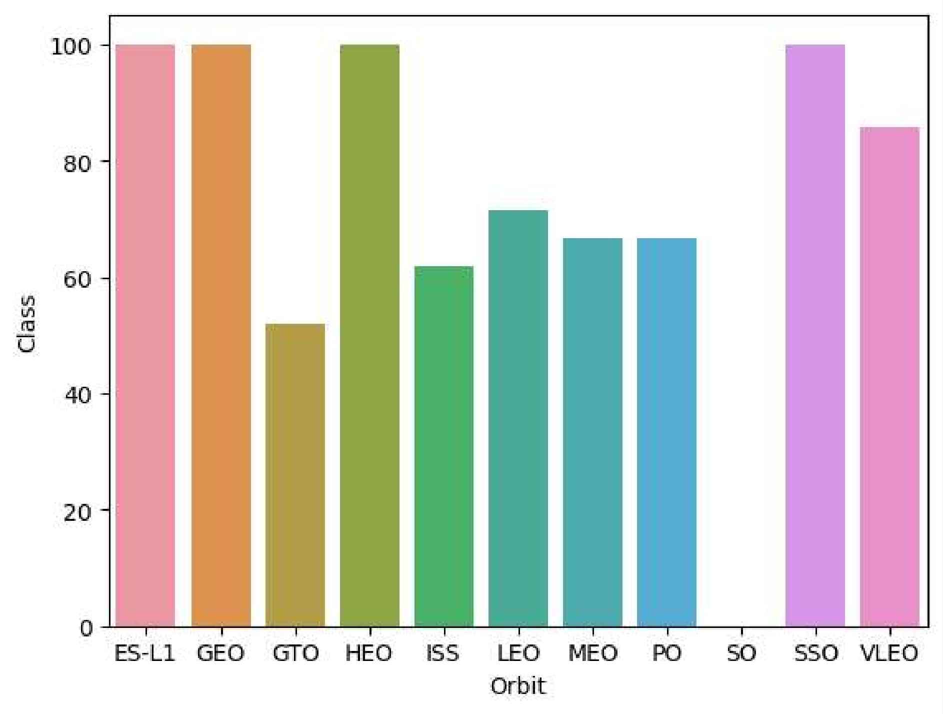
Payload vs. Launch Site

- Green indicates successful launch; Purple indicates unsuccessful launch
- No rocket launched with heavy payload mass in WAFB SLC



Success Rate vs. Orbit Type

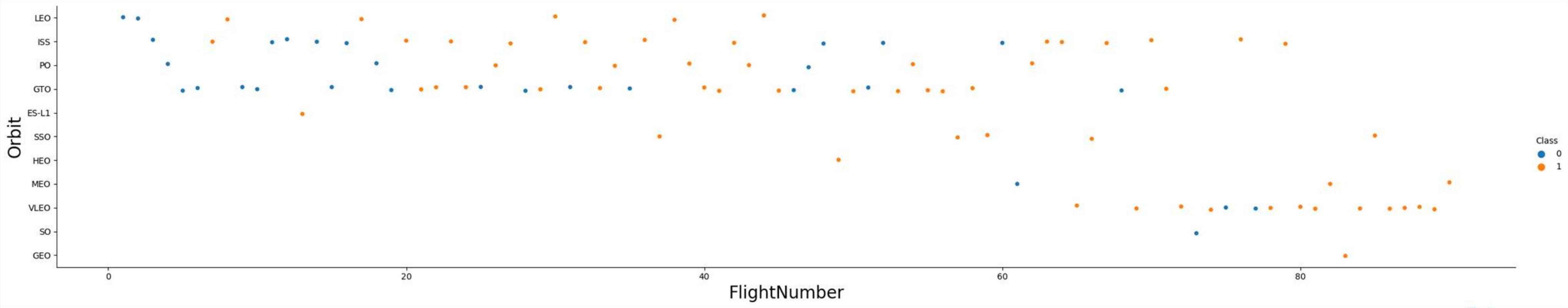
- ES-L1 , GEO , HEO, SSO have highest success rate (sample sizes in parenthesis)
- SO has lowest success rate



Flight Number vs. Orbit Type

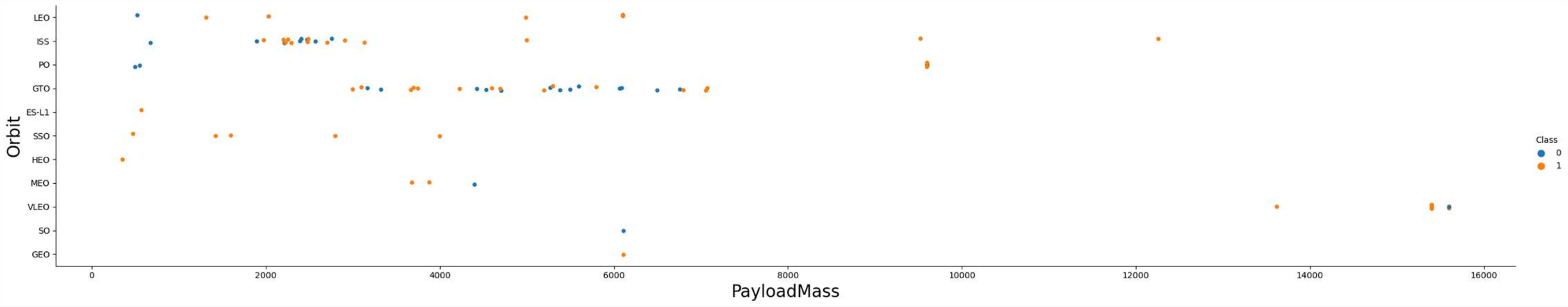
- Launch Orbit preferences changed over Flight Number.

  Launch Outcome seems to correlate with this preference.
- SpaceX started with LEO orbits which saw moderate success LEO and returned to VLEO in recent launches SpaceX appears to perform better in lower orbits or Sun-synchronous orbits



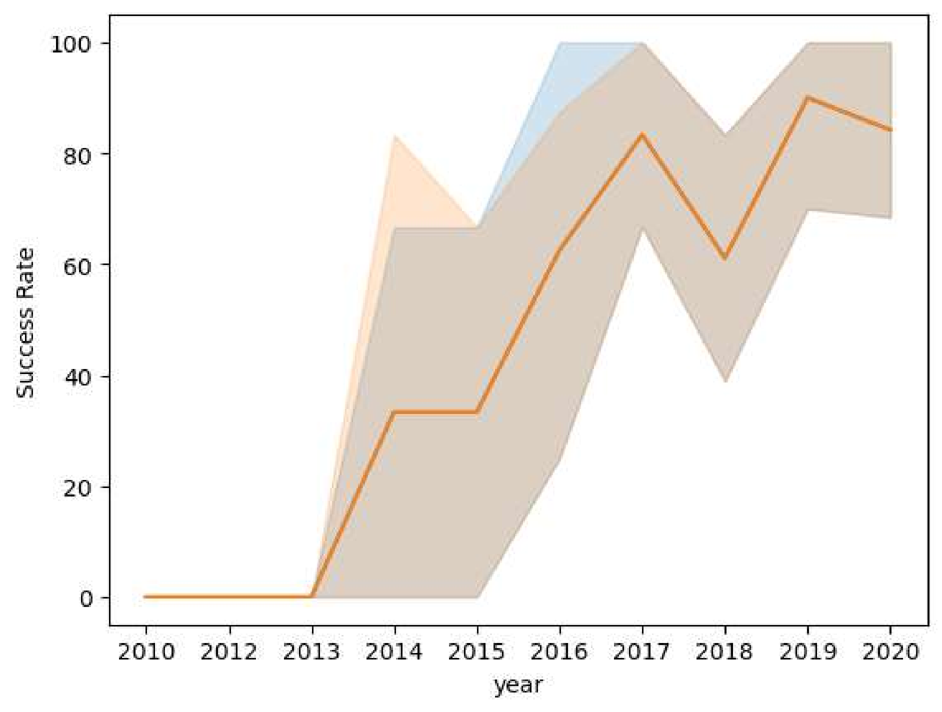
Payload vs. Orbit Type

- 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.



Launch Success Yearly Trend

- Success generally increases over time since 2013 with a slight dip in 2018
- Success in recent years at around 80%

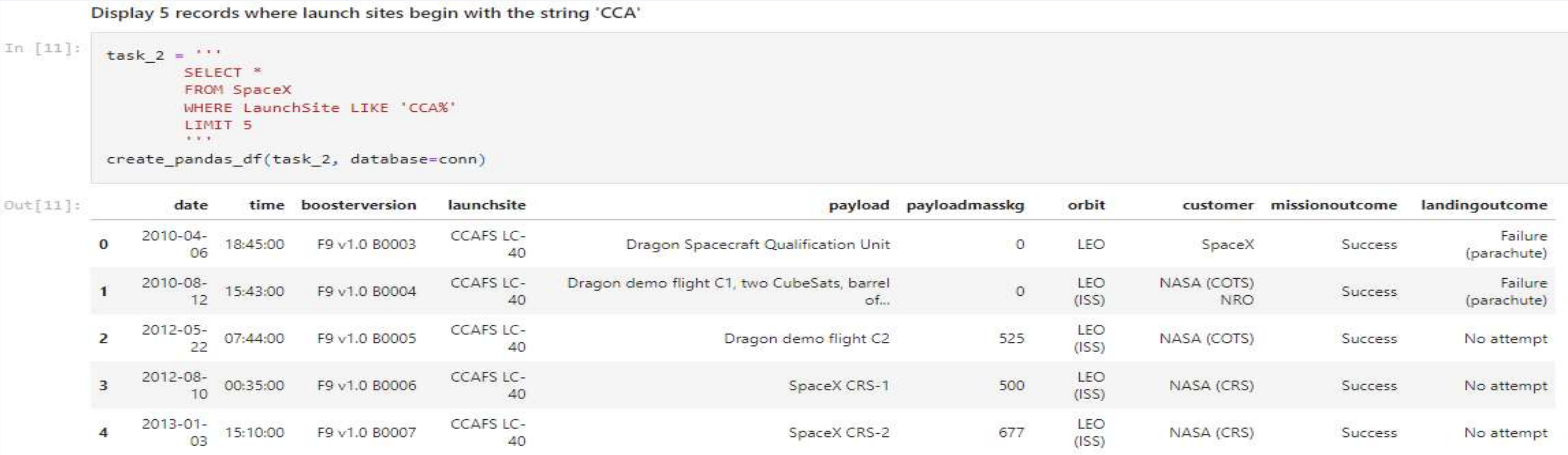


All Launch Site Names

Given the data, these are the names of the launch sites where different rocket landings where attempted:

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

Launch Site Names Begin with 'CCA'



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

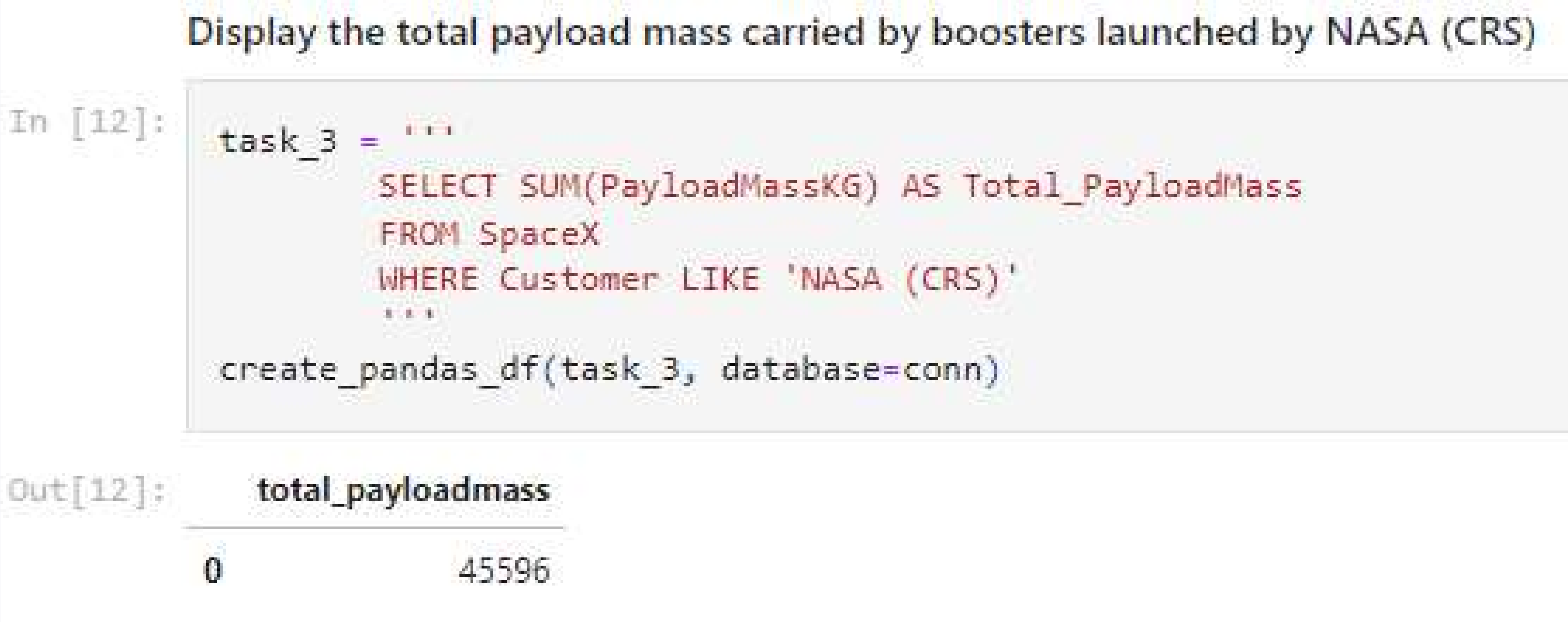
In [11]:

```
task_2 = '''
        SELECT *
        FROM SpaceX
        WHERE LaunchSite LIKE 'CCA%'
        LIMIT 5
        '''
create_pandas_df(task_2, database=conn)
```

Out[11]:

	date	time	boosterversion	launchsite	payload	payloadmasskg	orbit	customer	missionoutcome	landingoutcome
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	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
3	2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	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



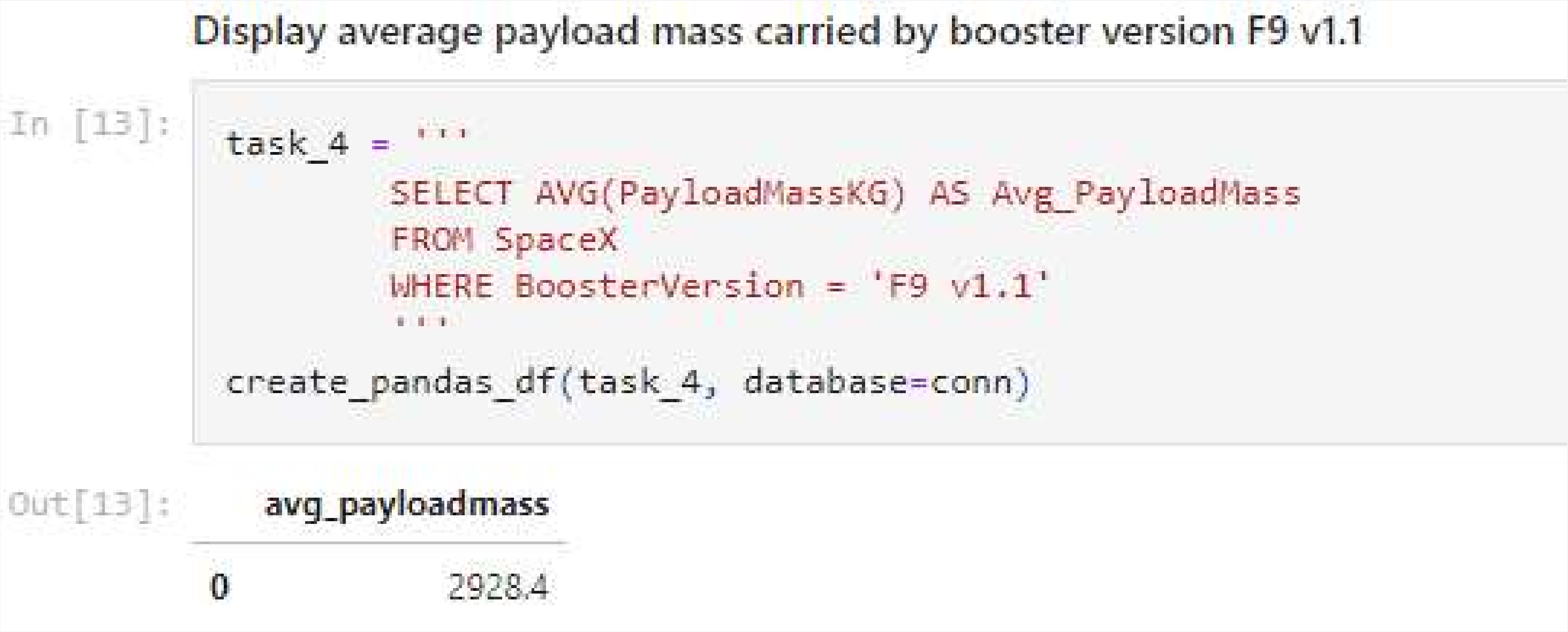
Display the total payload mass carried by boosters launched by NASA (CRS)

```
In [12]:

task_3 = '''
        SELECT SUM(PayloadMassKG) AS Total_PayloadMass
        FROM SpaceX
        WHERE Customer LIKE 'NASA (CRS)'
        '''
create_pandas_df(task_3, database=conn)
```

```
Out[12]:
    total_payloadmass
0               45596
```

Average Payload Mass by F9 v1.1



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

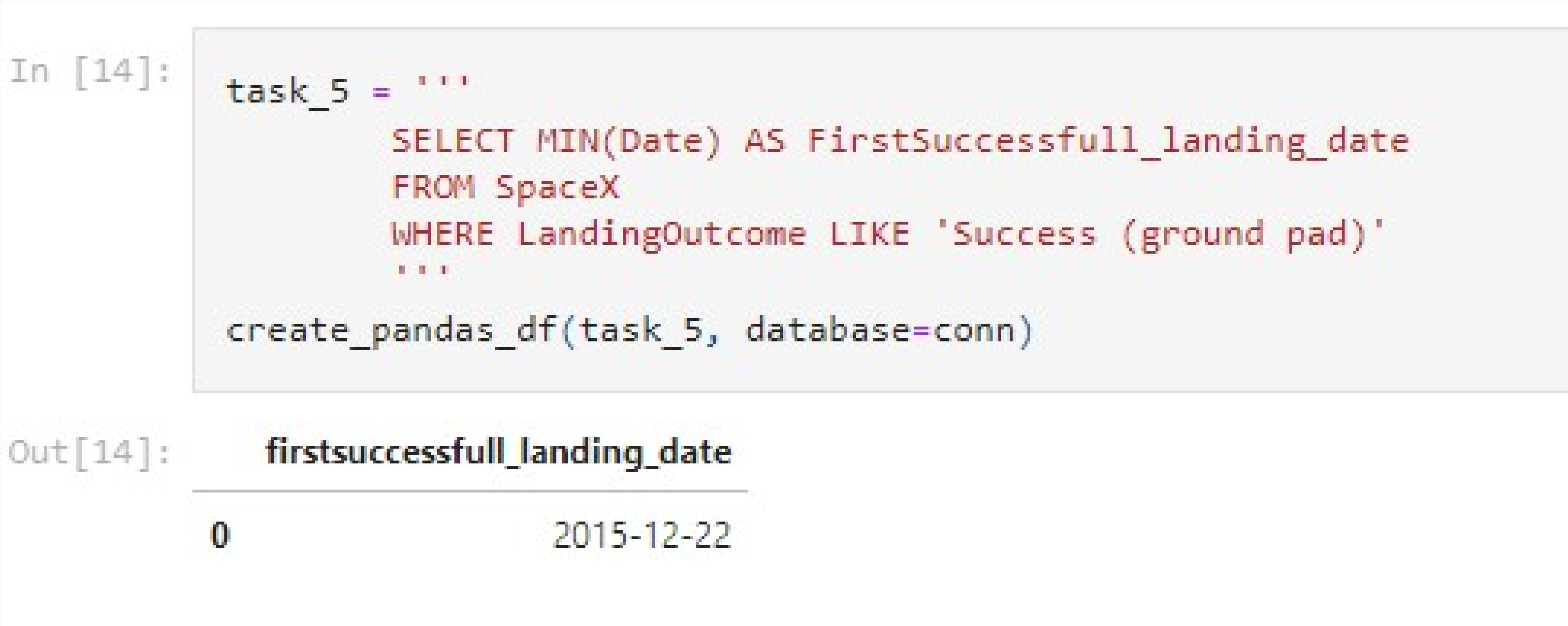
In [13]:

```
task_4 = '''
        SELECT AVG(PayloadMassKG) AS Avg_PayloadMass
        FROM SpaceX
        WHERE BoosterVersion = 'F9 v1.1'
        '''
create_pandas_df(task_4, database=conn)
```

Out[13]:

	avg_payloadmass
0	2928.4

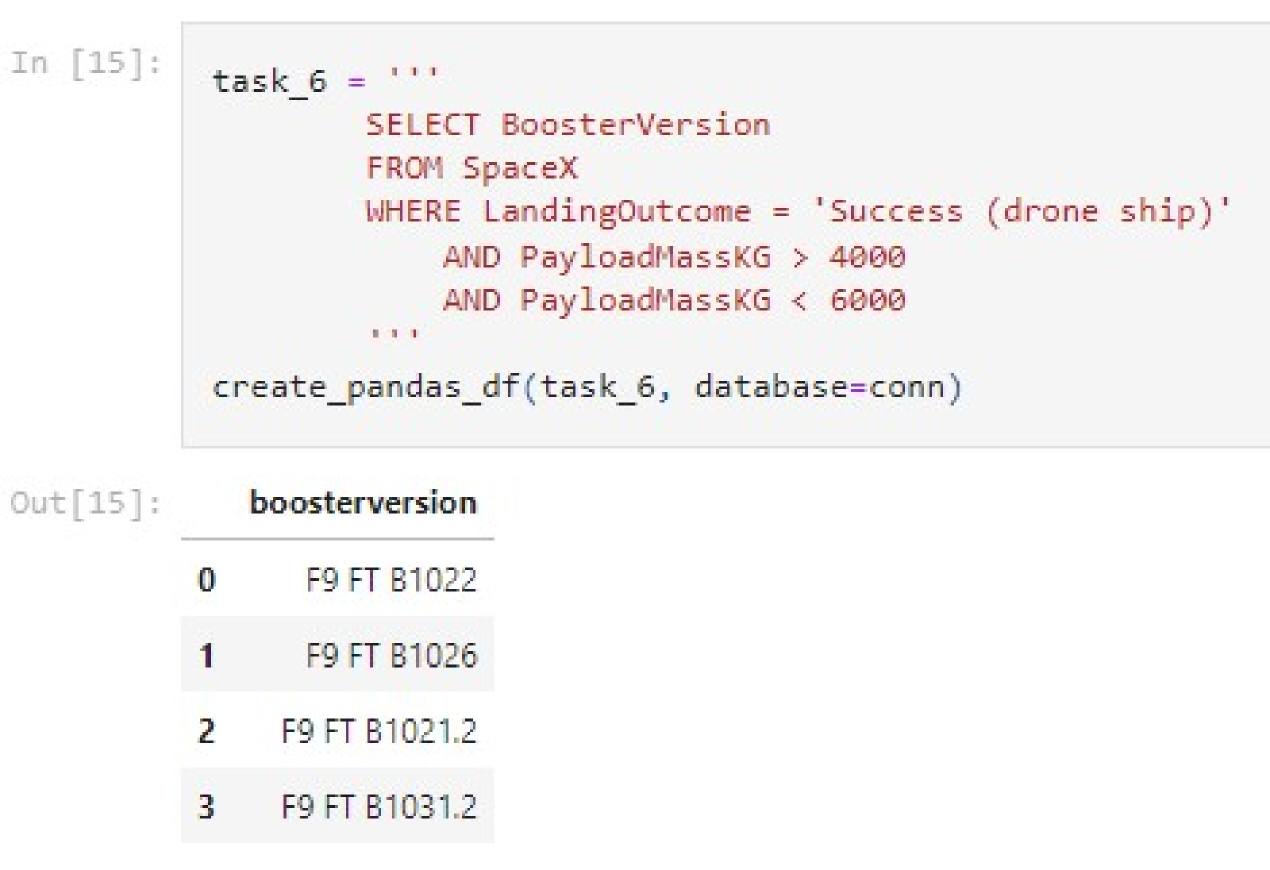
First Successful Ground Landing Date



```
In [14]: task_5 = '''
                 SELECT MIN(Date) AS FirstSuccessfull_landing_date
                 FROM SpaceX
                 WHERE LandingOutcome LIKE 'Success (ground pad)'
                 '''
         create_pandas_df(task_5, database=conn)
```

```
Out[14]:    firstsuccessfull_landing_date
         0                     2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000



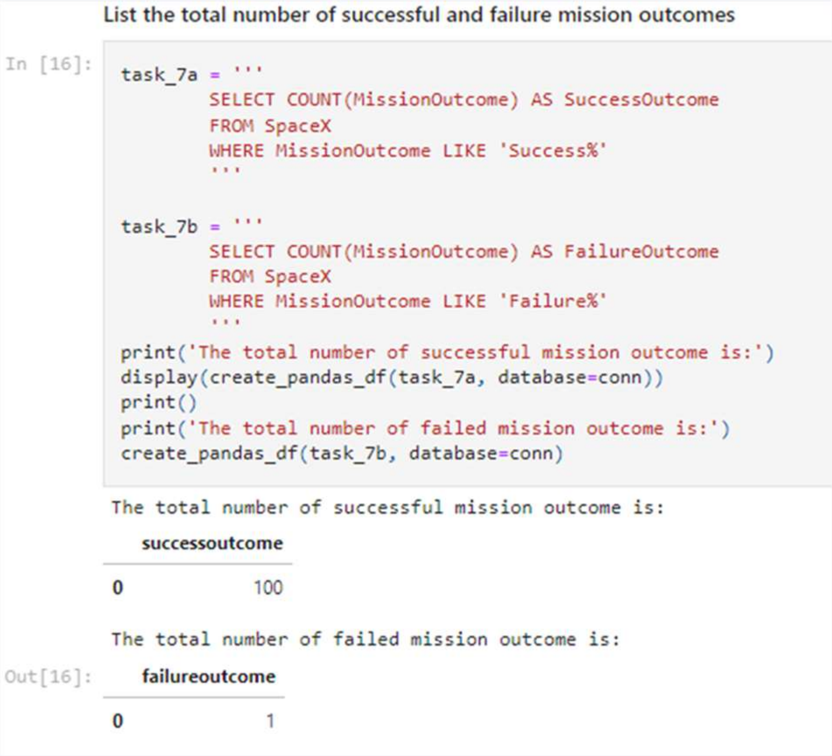
In [15]:

```
task_6 = '''
        SELECT BoosterVersion
        FROM SpaceX
        WHERE LandingOutcome = 'Success (drone ship)'
            AND PayloadMassKG > 4000
            AND PayloadMassKG < 6000
        '''
create_pandas_df(task_6, database=conn)
```

Out[15]:

	boosterversion
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes



List the total number of successful and failure mission outcomes

In [16]:

```
task_7a = '''
        SELECT COUNT(MissionOutcome) AS SuccessOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Success%'
        '''

task_7b = '''
        SELECT COUNT(MissionOutcome) AS FailureOutcome
        FROM SpaceX
        WHERE MissionOutcome LIKE 'Failure%'
        '''
print('The total number of successful mission outcome is:')
display(create_pandas_df(task_7a, database=conn))
print()
print('The total number of failed mission outcome is:')
create_pandas_df(task_7b, database=conn)
```

```
The total number of successful mission outcome is:
```

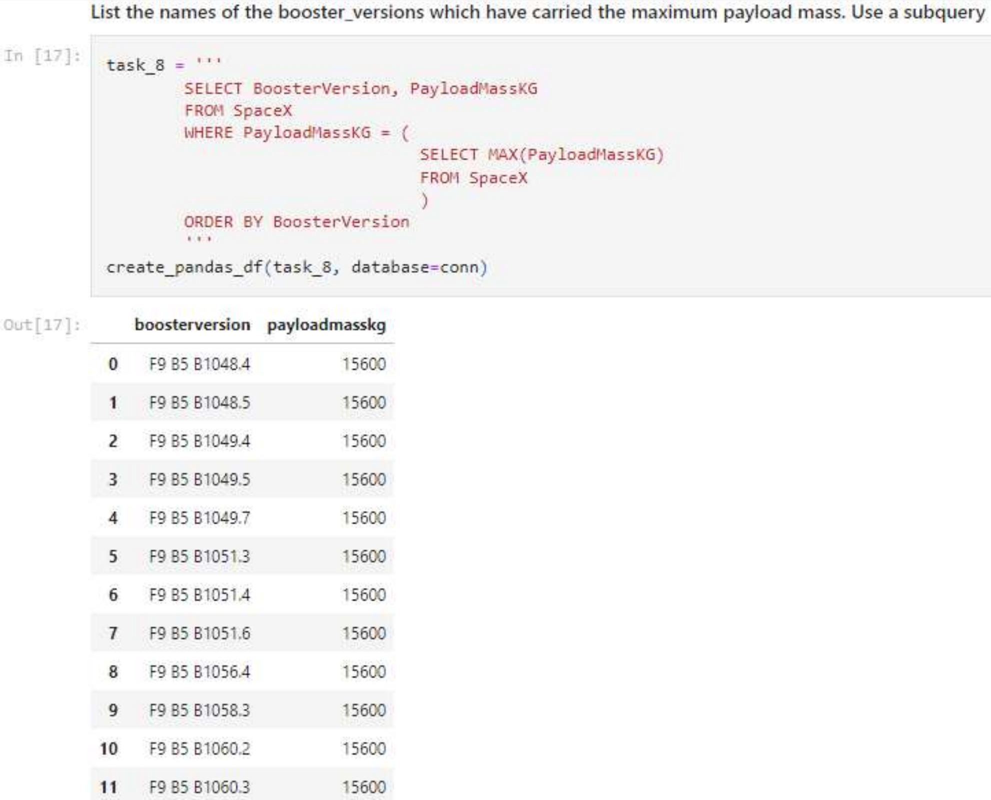
	successoutcome
0	100

```
The total number of failed mission outcome is:
```

Out[16]:

	failureoutcome
0	1

Boosters Carried Maximum Payload



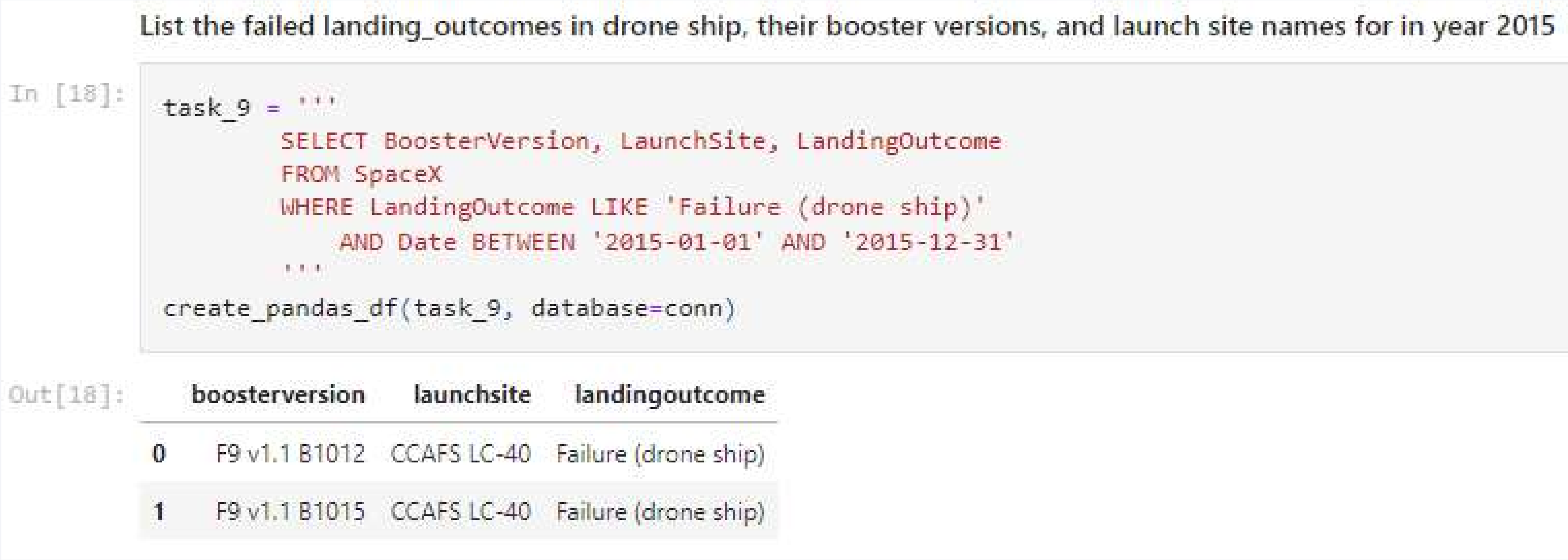
List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
In [17]:
task_8 = '''
          SELECT BoosterVersion, PayloadMassKG
          FROM SpaceX
          WHERE PayloadMassKG = (
                                  SELECT MAX(PayloadMassKG)
                                  FROM SpaceX
                                  )
          ORDER BY BoosterVersion
          '''
create_pandas_df(task_8, database=conn)
```

Out[17]:

	boosterversion	payloadmasskg
0	F9 B5 B1048.4	15600
1	F9 B5 B1048.5	15600
2	F9 B5 B1049.4	15600
3	F9 B5 B1049.5	15600
4	F9 B5 B1049.7	15600
5	F9 B5 B1051.3	15600
6	F9 B5 B1051.4	15600
7	F9 B5 B1051.6	15600
8	F9 B5 B1056.4	15600
9	F9 B5 B1058.3	15600
10	F9 B5 B1060.2	15600
11	F9 B5 B1060.3	15600

2015 Launch Records



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

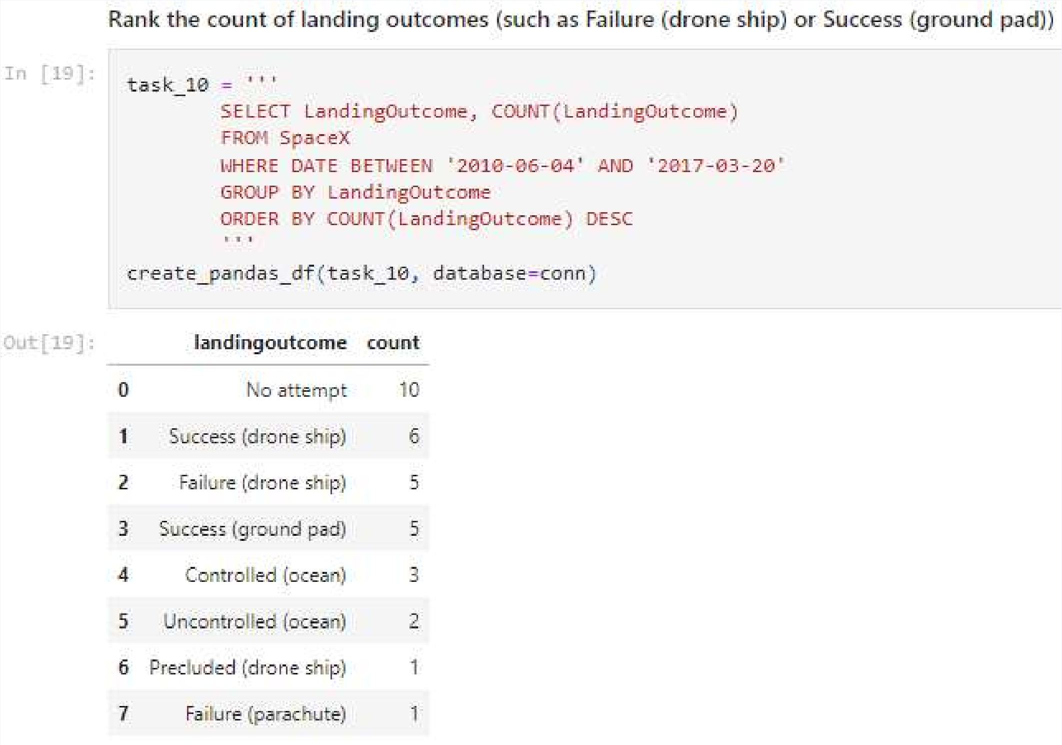
```
In [18]:  task_9 = '''
                  SELECT BoosterVersion, LaunchSite, LandingOutcome
                  FROM SpaceX
                  WHERE LandingOutcome LIKE 'Failure (drone ship)'
                      AND Date BETWEEN '2015-01-01' AND '2015-12-31'
                  '''
          create_pandas_df(task_9, database=conn)
```

Out[18]:

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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
-



Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad))

In [19]:
```
task_10 = '''
        SELECT LandingOutcome, COUNT(LandingOutcome)
        FROM SpaceX
        WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY LandingOutcome
        ORDER BY COUNT(LandingOutcome) DESC
        '''
create_pandas_df(task_10, database=conn)
```

Out[19]:

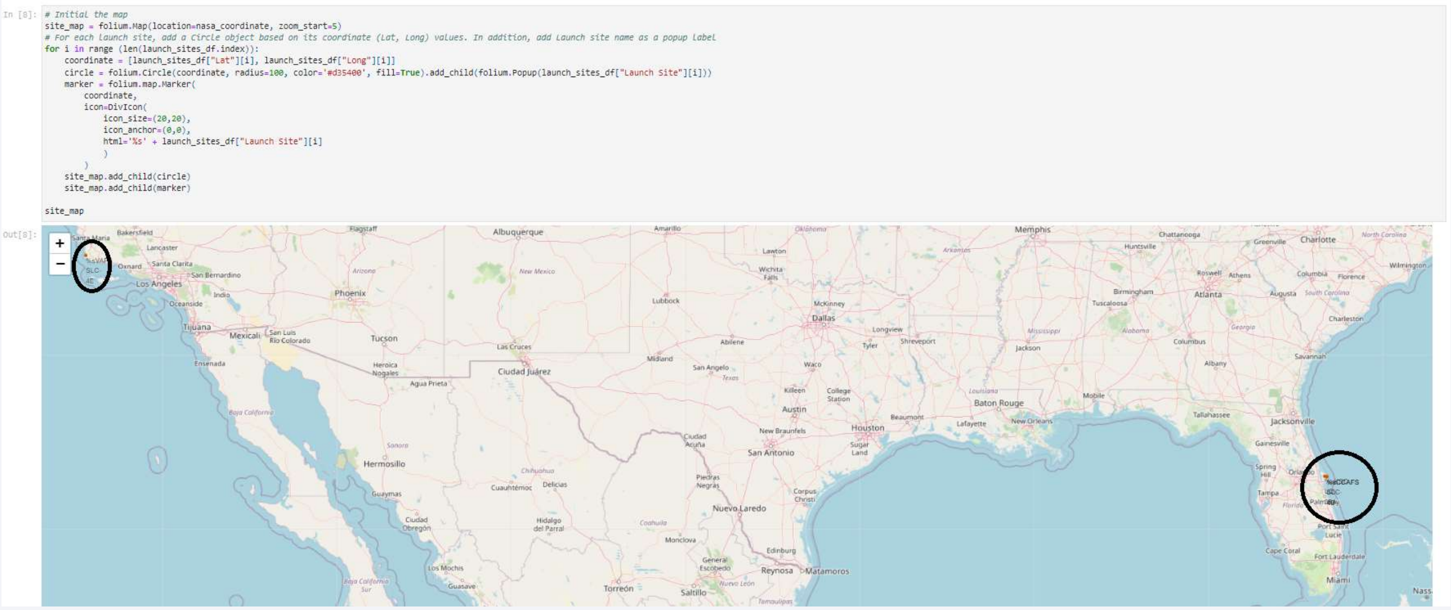
	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

Section 3

Launch Sites Proximities Analysis

Launch Site Locations

- We can see that all launch sites are in very close proximity to the coast and they are also a couple thousand kilometers away from the equator line.
- It is interesting to see that most launch sites are concentrated near Miami.



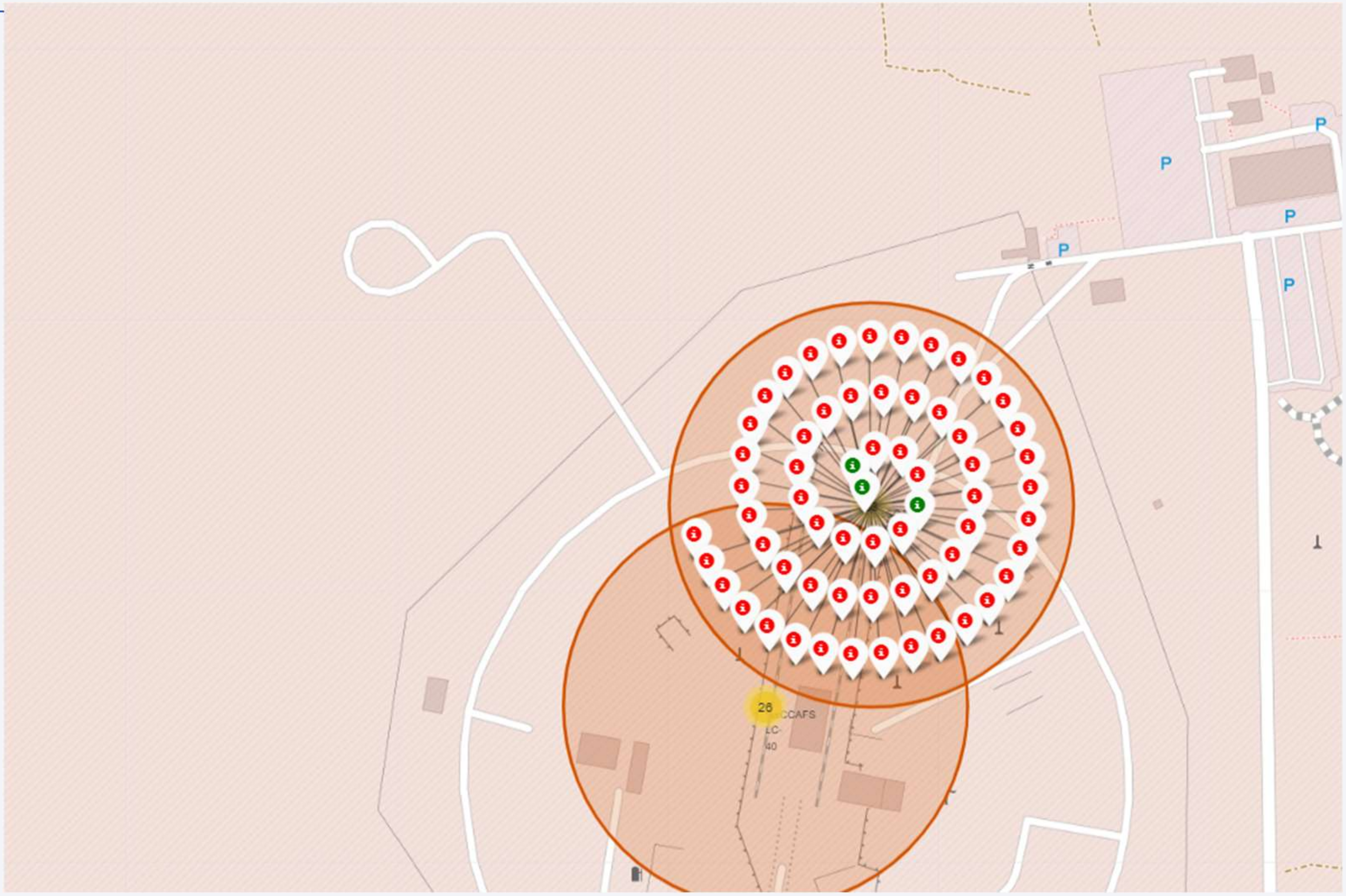
In [8]:

```
# Initial the map
site_map = folium.Map(location=nasa_coordinate, zoom_start=5)
# For each launch site, add a Circle object based on its coordinate (Lat, Long) values. In addition, add Launch site name as a popup label
for i in range (len(launch_sites_df.index)):
    coordinate = [launch_sites_df["Lat"][i], launch_sites_df["Long"][i]]
    circle = folium.Circle(coordinate, radius=100, color='#d35400', fill=True).add_child(folium.Popup(launch_sites_df["Launch Site"][i]))
    marker = folium.map.Marker(
        coordinate,
        icon=DivIcon(
            icon_size=(20,20),
            icon_anchor=(0,0),
            html='%s' + launch_sites_df["Launch Site"][i]
            )
        )
    site_map.add_child(circle)
    site_map.add_child(marker)

site_map
```

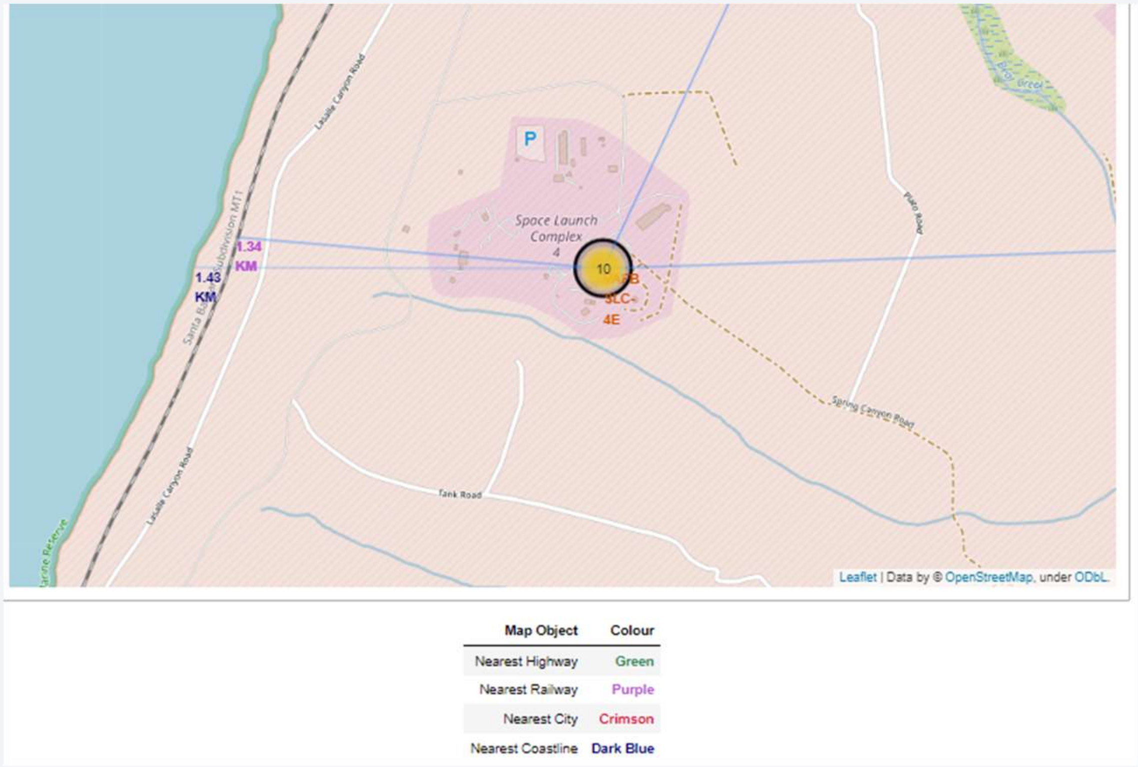
Successful Rocket Launches

- The successful launches are represented by a green marker while the red marker represents failed rocket launches.

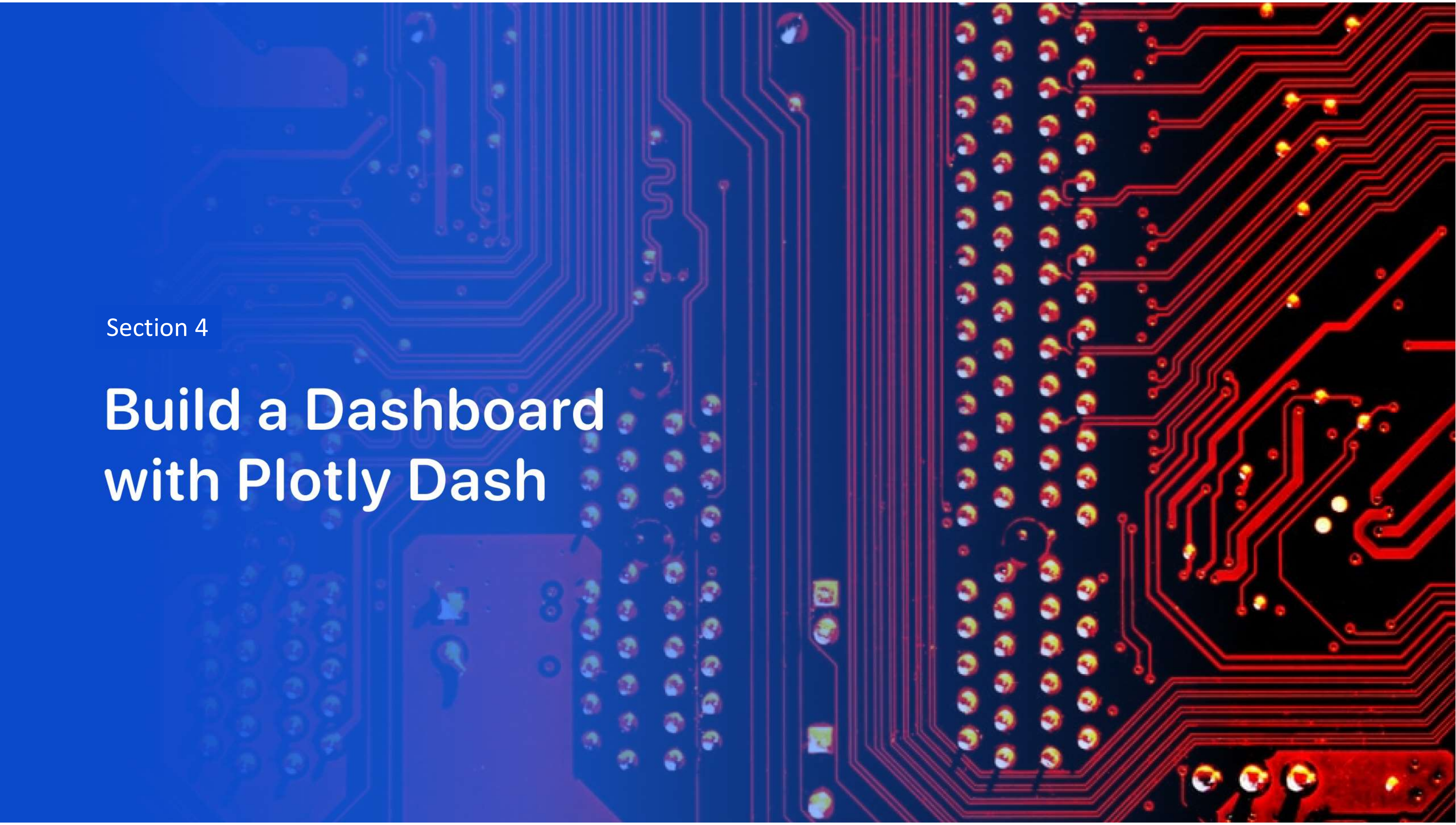


Surrounding Landmarks

- It appears that launch sites are usually set up at least 18 km away from cities. This may be because of the desire to prevent any crashes near populated areas.
- It is also apparent that launch sites are in very close proximity to railways and highways. Perhaps, due to the necessary transportation requirements for rocket parts



Map Object	Colour
Nearest Highway	Green
Nearest Railway	Purple
Nearest City	Crimson
Nearest Coastline	Dark Blue

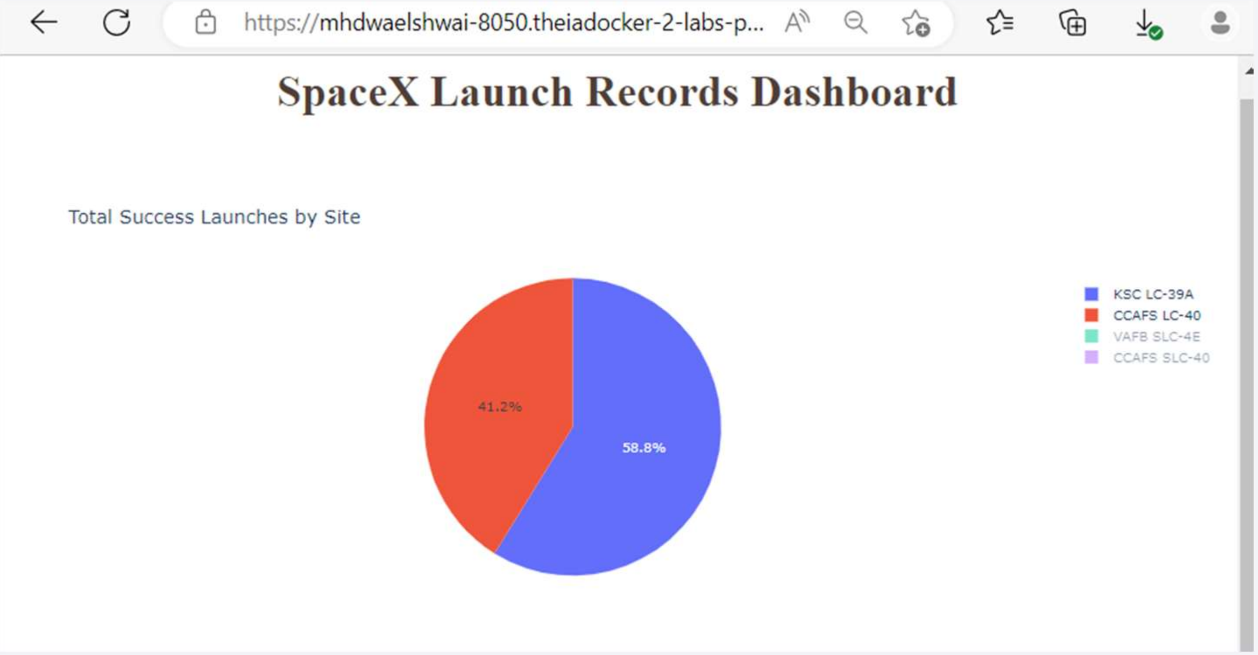


Section 4

Build a Dashboard with Plotly Dash

Successful Launches ite

- Site KSC LC-39A has the largest successful launches as well the highest launch success rate.
- More investigation may be needed to determine why KSC LC-39A is the preferred launch site.

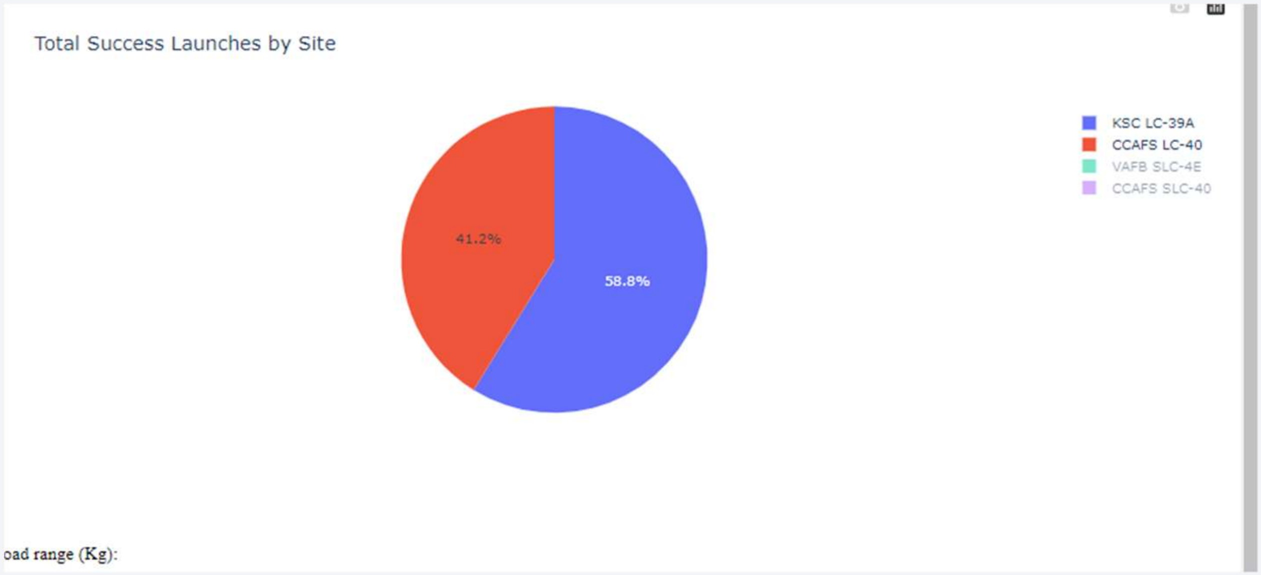


https://mhdwaelshwai-8050.theiadocker-2-labs-p...

SpaceX Launch Records Dashboard

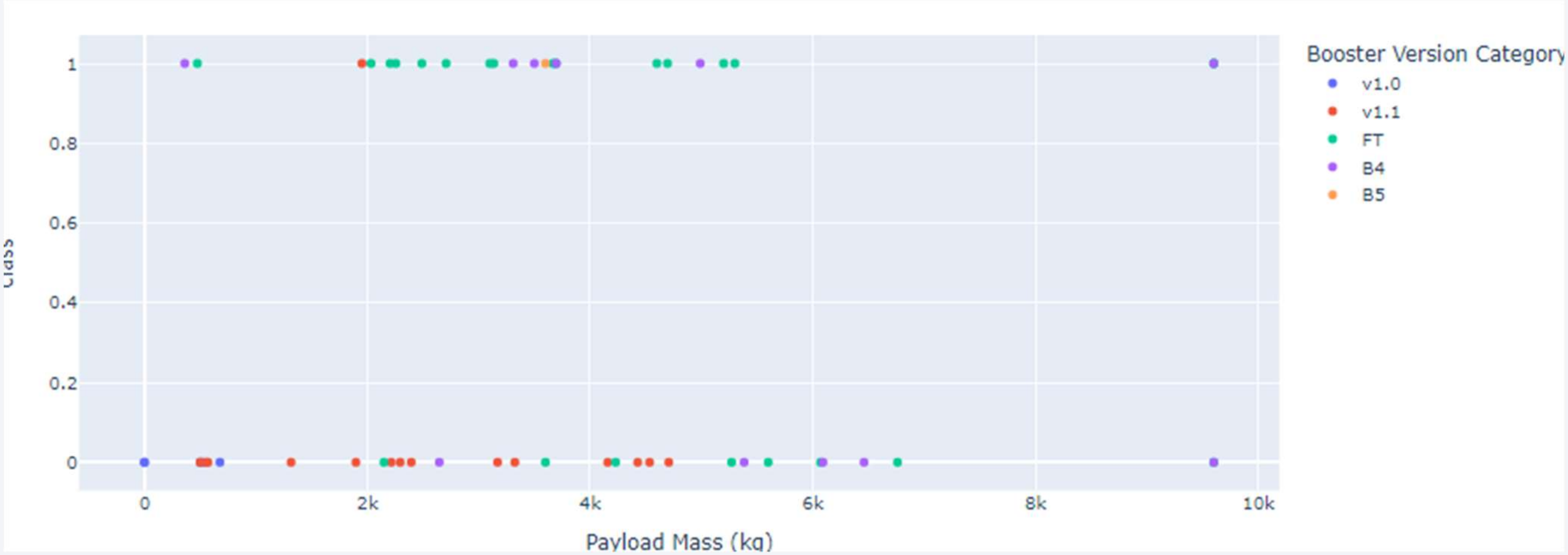
Total Successful Launches for Site KSC LC-39A

- As we can see, 76.9% of the total launches at site KSC LC-39A were successful. This is a the highest success rate of all the different launch sites.
- However, this success rate was only around 3% higher than the runner up; site CCAFS LC-40.



Launch Success and Payload Mass for All Sites

- highest success rate is for payload between 2000 kg and 4000 kg

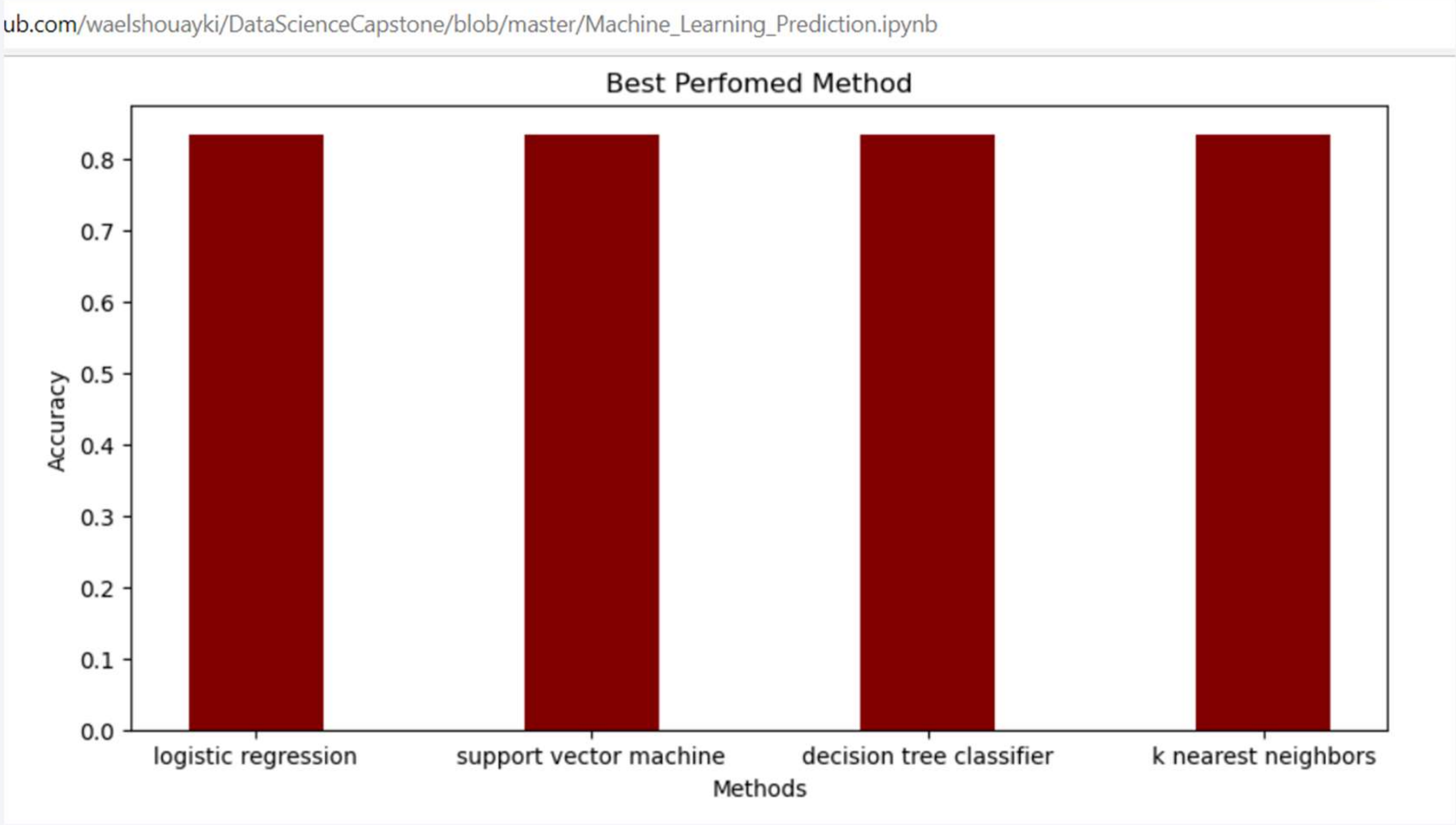


Section 5

Predictive Analysis (Classification)

Classification Accuracy

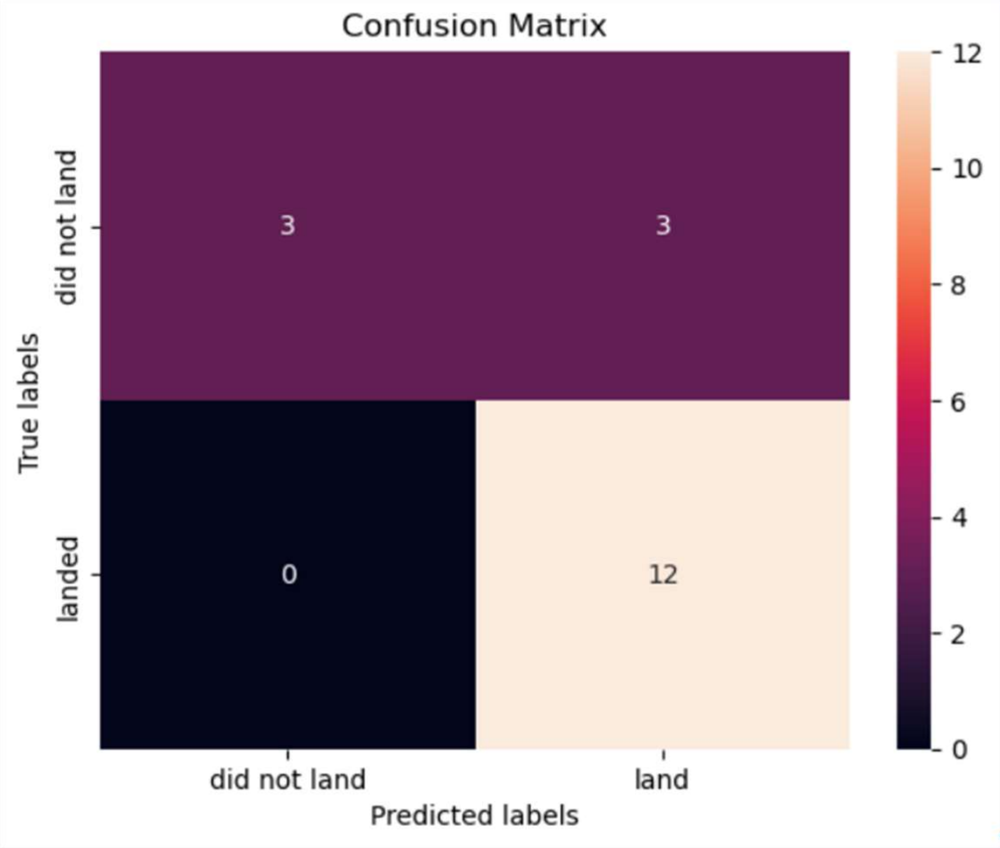
- All method having same accuracy, so logistic regression is required for the classification



ub.com/waelshouayki/DataScienceCapstone/blob/master/Machine_Learning_Prediction.ipynb

Confusion Matrix

- The model only failed to accurately predict 3 labels.



Conclusions

- Develop a machine learning model for Space Y who wants to bid against SpaceX
- The goal of model is to predict when Stage 1 will successfully land to save ~$100 million USD
- Use data from a public SpaceX API and web scraping SpaceX Wikipedia page
- Create a dashboard for visualization
- Create a machine learning model with an accuracy of 83%

Thank you!