IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Farhanah Md Sam
22 June 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection via API and Web Scraping
  - Data wrangling
  - EDA with SQL
  - EDA with Data Visualization
  - Building an Interactive Map with Folium
  - Building a Dashboard with Plotly Dash
  - Machine Learning Prediction - Classification
- Summary of all results
  - EDA results
  - Interactive analytics
  - Predictive analysis

Introduction

- Project background and context
  - Space X 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 Space X can reuse the first stage.

- Problems you want to find answers
  - 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 space X for a rocket launch. In this lab, you will create a machine learning pipeline to predict if the first stage will land given the data from the preceding labs

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data were retrieved from SpaceX API and Web Scraping from Wikipedia
- Perform data wrangling
  - Data cleaning for missing values and one-hot encoding for categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models of Decision Tree, K-nearest Neighbors, Logistics Regression, Support Vector Machine to find the best classifier

Data Collection

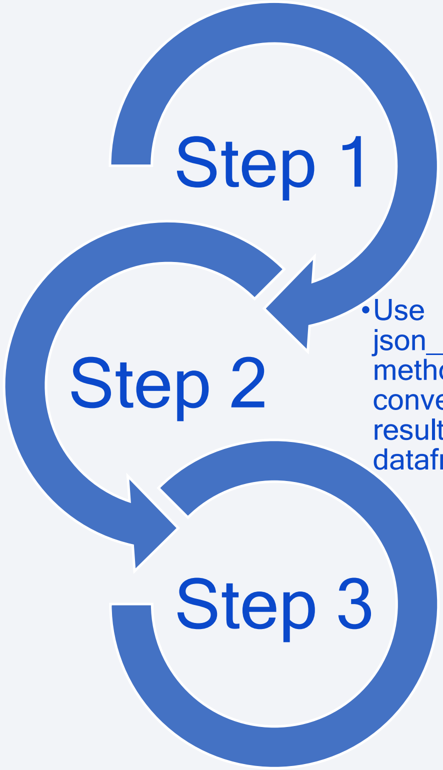
- Describe how data sets were collected.
  - There are two ways how the data sets were collected as follows:

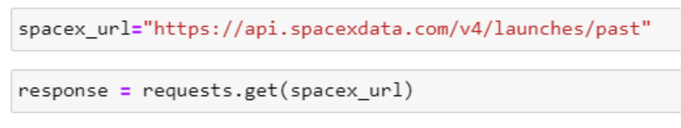
	SpaceX API	Web Scraping
Source	Space X	Wikipedia
Step 1	Requested rocket launch data from SpaceX API	Extracted a Falcon 9 launch records HTML table from Wikipedia
Step 2	Decoded the response content as a Json using .json() and turned it into a Pandas dataframe using .json_normalize()	Parsed the table and convert it into a Pandas data frame using BeautifulSoup
Step 3	Performed data cleaning for missing values by replacing it with the mean value of quantitative features	Extracted all column/variable names from the HTML table header

- You need to present your data collection process use key phrases and flowcharts

Data Collection - SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts
- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose
- Github: here



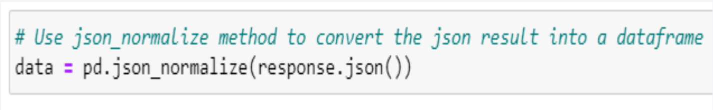


```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

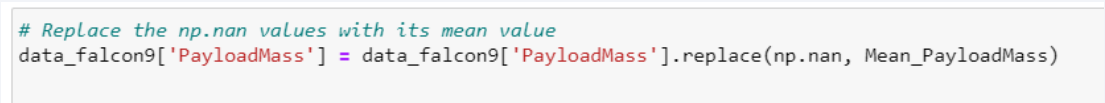
- Requested rocket launch data from SpaceX API

- Use json_normalize method to convert the json result into a dataframe



```
# Use json_normalize method to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

- Performed data cleaning for missing values



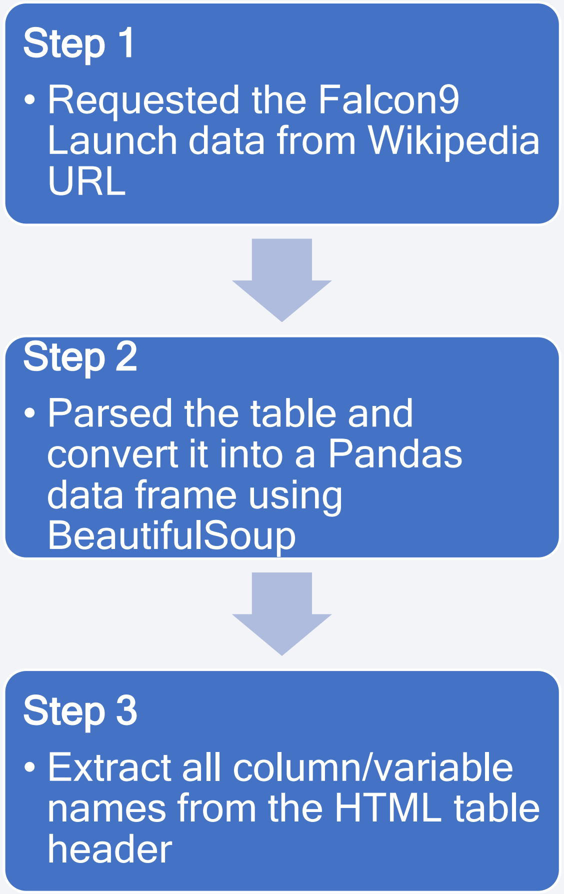
```
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, Mean_PayloadMass)
```

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose

- Github: [here]



Step 1

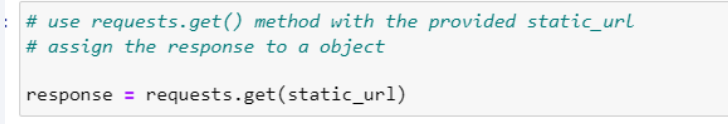
- Requested the Falcon9 Launch data from Wikipedia URL

Step 2

- Parsed the table and convert it into a Pandas data frame using BeautifulSoup

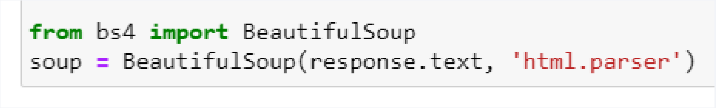
Step 3

- Extract all column/variable names from the HTML table header

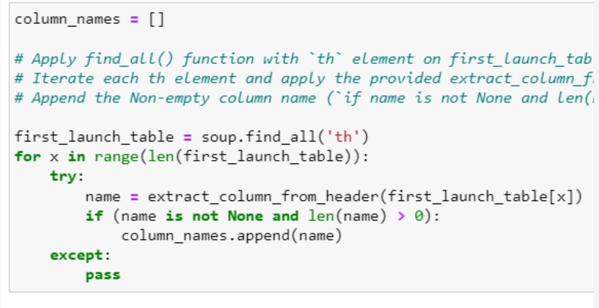


```
# use requests.get() method with the provided static_url
# assign the response to a object

response = requests.get(static_url)
```



```
from bs4 import BeautifulSoup
soup = BeautifulSoup(response.text, 'html.parser')
```



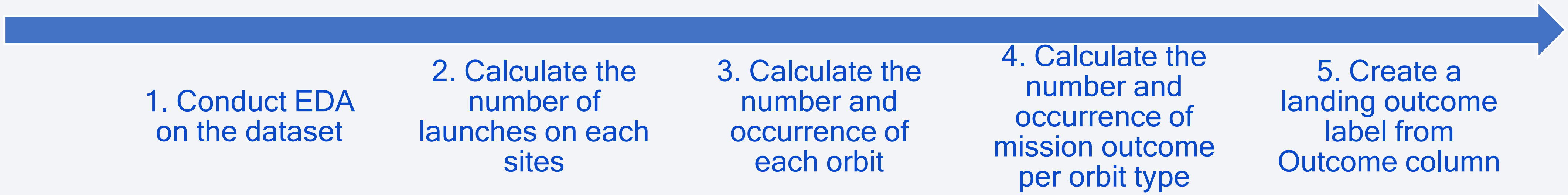
```
column_names = []

# Apply find_all() function with `th` element on first_launch_tab
# Iterate each th element and apply the provided extract_column_f
# Append the Non-empty column name (`if name is not None and len(

first_launch_table = soup.find_all('th')
for x in range(len(first_launch_table)):
    try:
        name = extract_column_from_header(first_launch_table[x])
        if (name is not None and len(name) > 0):
            column_names.append(name)
    except:
        pass
```

Data Wrangling

- Describe how data were processed
- You need to present your data wrangling process using key phrases and flowcharts



1. Conduct EDA on the dataset

2. Calculate the number of launches on each sites

3. Calculate the number and occurrence of each orbit

4. Calculate the number and occurrence of mission outcome per orbit type

5. Create a landing outcome label from Outcome column

- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose

Github: here

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose

No	Features	Type of Chart	Reason
1	Flight Number vs Launch Site	Scatterplot	To find relationship between these two numeric variables
2	Payload vs Launch Site		
3	FlightNumber vs Orbit type		
4	Payload vs Orbit type		
5	Success rate of each orbit type	Bar chart	To find the probability of success rate for each orbit type
6	Year vs Success Rate	Line chart	To observe the trend of success rate across a period of time (years)

Github: here

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
  1. Launch Sites – unique, string ‘CCA’
  2. Payload Mass (kg) – total, average
  3. Mission Outcome – total number of success and fail mission outcome
  4. Booster version - carried the maximum payload mass.
  5. Date – 1st successful landing outcome in ground pad
- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose

Github: here

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Explain why you added those objects

No	Item	Reason
1	Add each site's location, label by using site's latitude & longitude coordinates and circle marker with name of launch sites, respectively	Visualize all launch sites into an interactive map
2	Mark the success/failed launches for each site on the map	Assign feature of launch_outcomes with green marker for success launch site whereas red for failed launch site
3	Using Haversine's formula	Calculated distances to the closest coastline, city, railway and highway, represented by a blue line on the map

Github: here

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

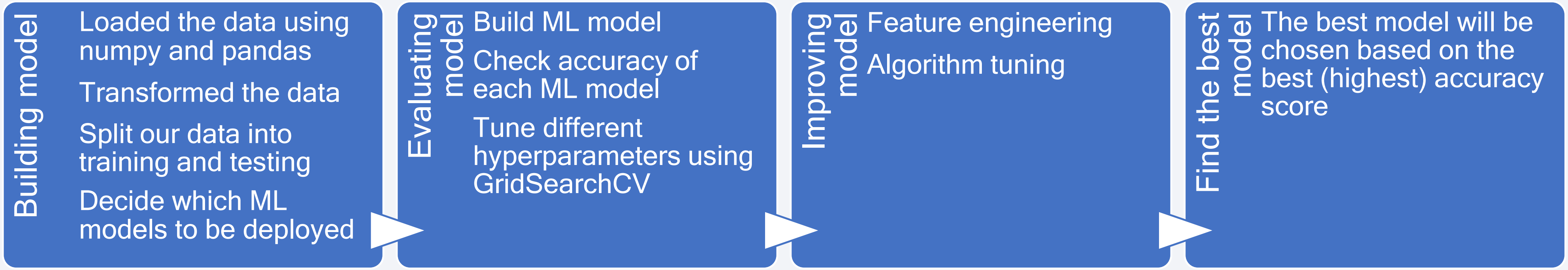
No	Item	Reason
1	Pie chart of total successful launches count for all sites	Represent the proportional i.e. percentage of the success for each sites
2	Scatter chart of between payload and launch success	Find relationship between Outcome and Payload Mass (Kg) for the different booster version.

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

Github: here

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



Building model

- Loaded the data using numpy and pandas
- Transformed the data
- Split our data into training and testing
- Decide which ML models to be deployed

Evaluating model

- Build ML model
- Check accuracy of each ML model
- Tune different hyperparameters using GridSearchCV

Improving model

- Feature engineering
- Algorithm tuning

Find the best model

- The best model will be chosen based on the best (highest) accuracy score

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

Github: here

Results

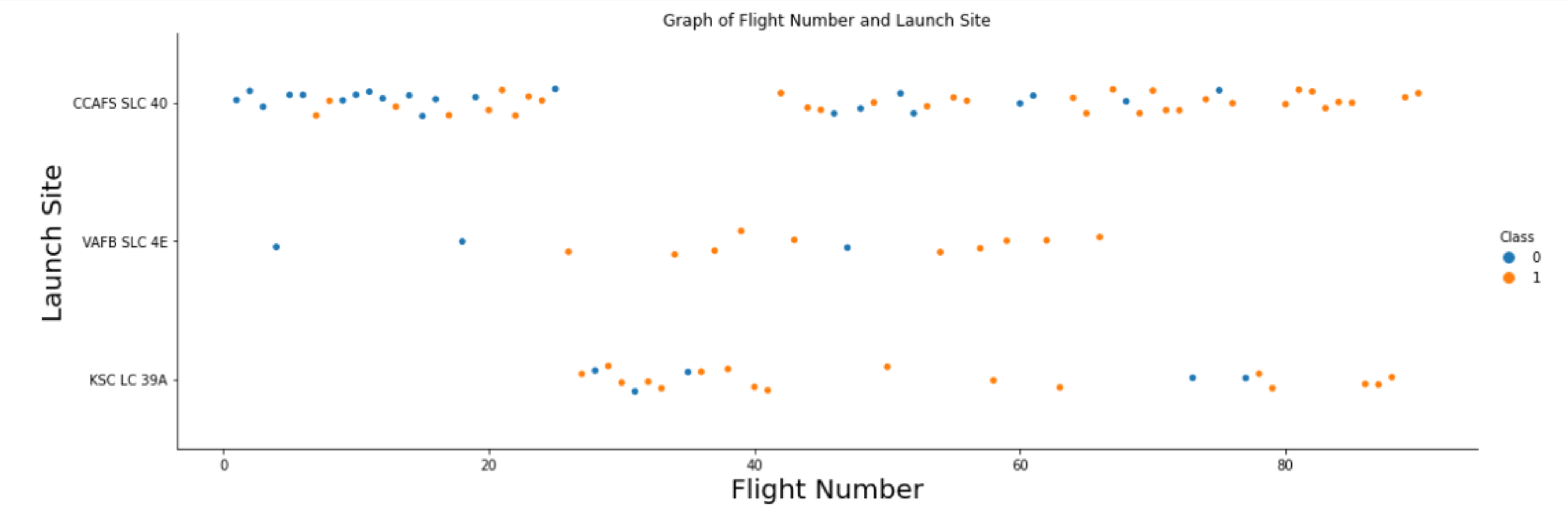
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

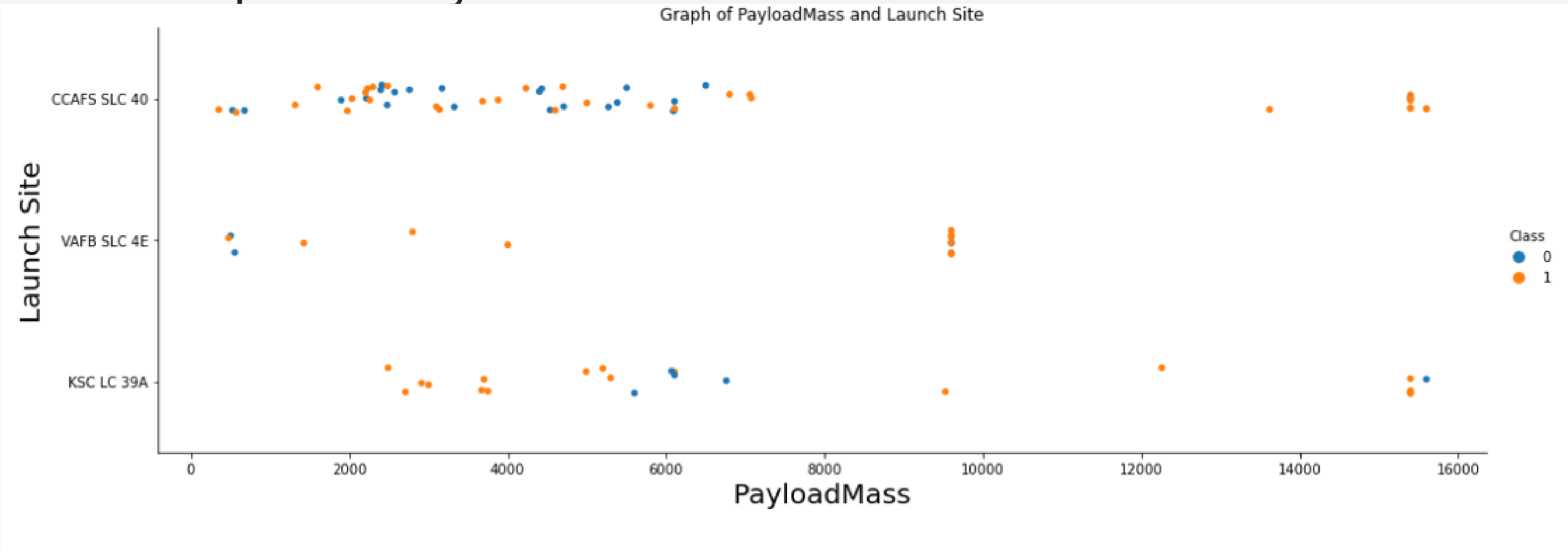
- Show a scatter plot of Flight Number vs. Launch Site



- Show the screenshot of the scatter plot with explanations
    - The larger amount of the flight number at the launch site, the greater the success rate at a launch site will be.

Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site



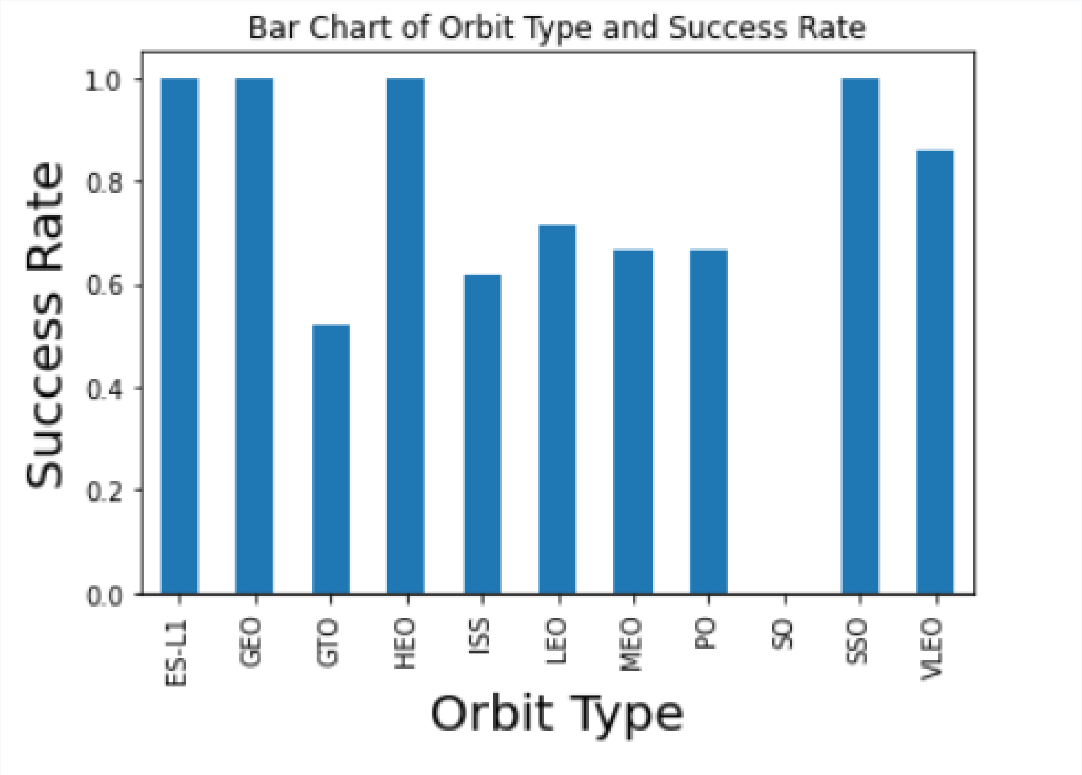
- Show the screenshot of the scatter plot with explanations

CCAFS SLC 40: The greater the payload mass at the launch site, the higher the success rate for the Rocket.
VAFB-SLC: No rockets launched for heavy payload mass (greater than 10,000 kg)

Success Rate vs. Orbit Type

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

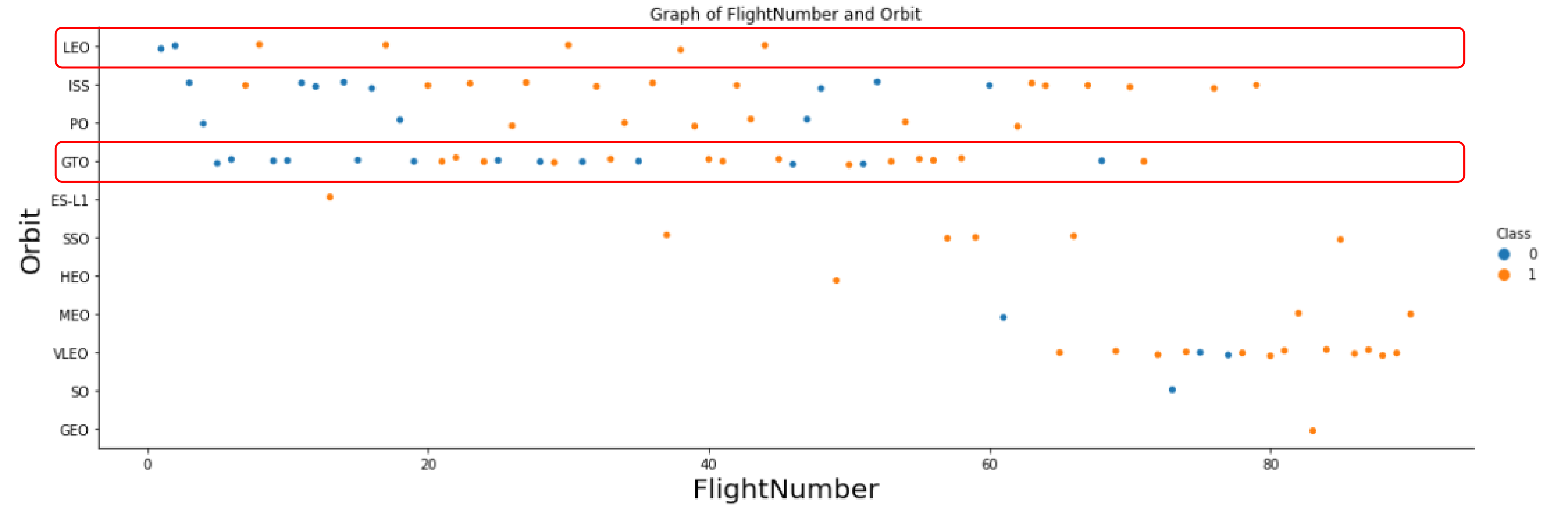
- Show the screenshot of the scatter plot with explanations



It shows that Orbit ES-L1, GEO, HEO and SSO have 100% success rate at the landing outcomes whereas Orbit SO recorded 0% of success rate at the landing outcome.

Flight Number vs. Orbit Type

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

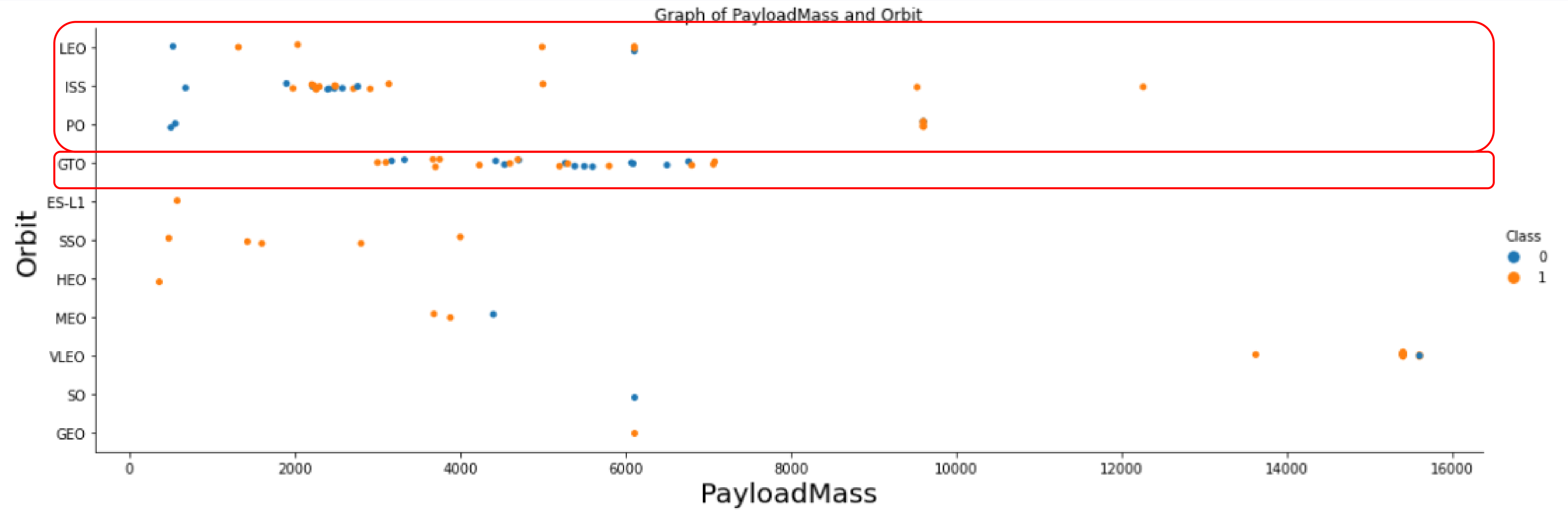


- Show the screenshot of the scatter plot with explanations

  - In LEO orbit, the Success appears related to the number of flights
  - However, there exists no relationship between flight number when in GTO orbit

Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type

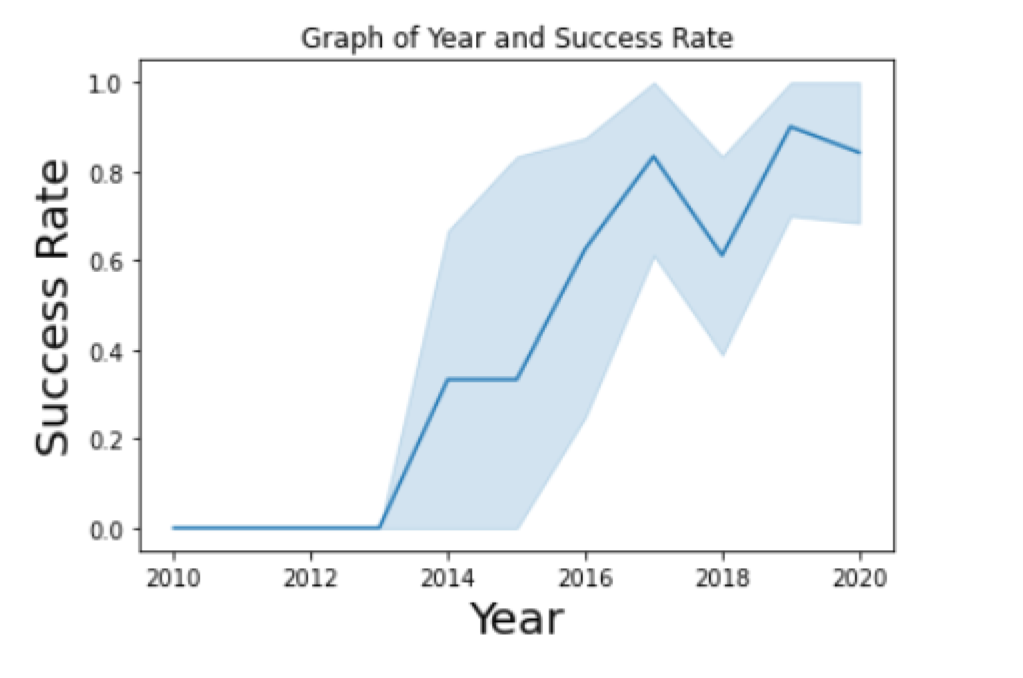


- Show the screenshot of the scatter plot with explanations
  - Polar, LEO and ISS Orbit: Heavy payloads contributed to the successful landing
  - GTO orbit: It seems like there exists no relationship between payloadmass and landing
  - outcomes at the site

Launch Success Yearly Trend

- Show a line chart of yearly average success rate

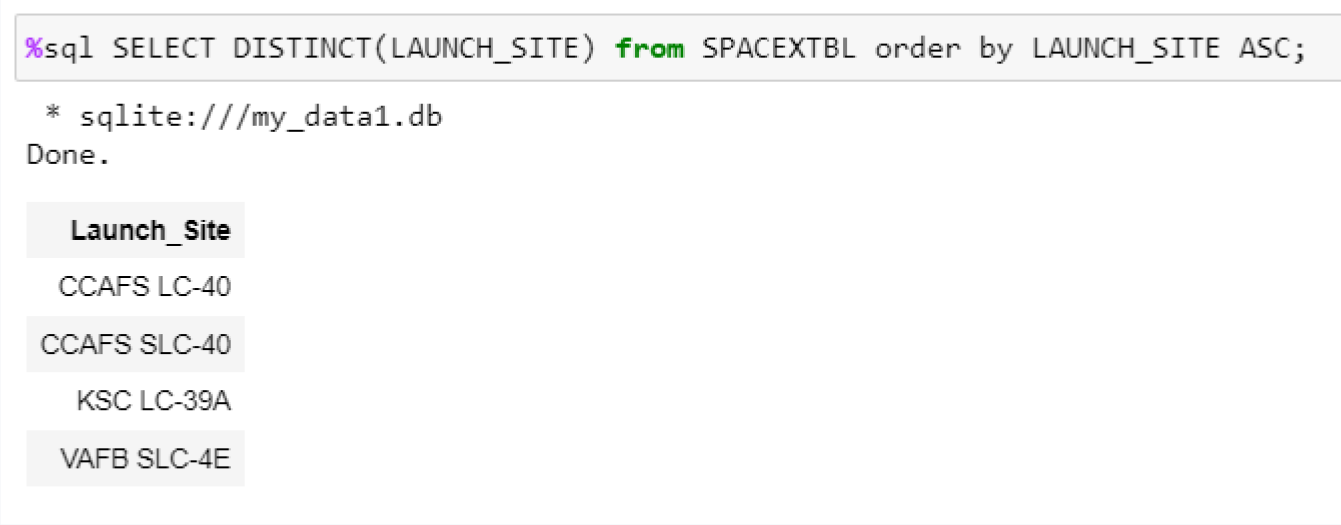
- Show the screenshot of the scatter plot with explanations



It is observed that the success rate is on an increasing trend, from 2013 to 2020

All Launch Site Names

- Find the names of the unique launch sites
- Present your query result with a short explanation here



```
%sql SELECT DISTINCT(LAUNCH_SITE) from SPACEXTBL order by LAUNCH_SITE ASC;
```

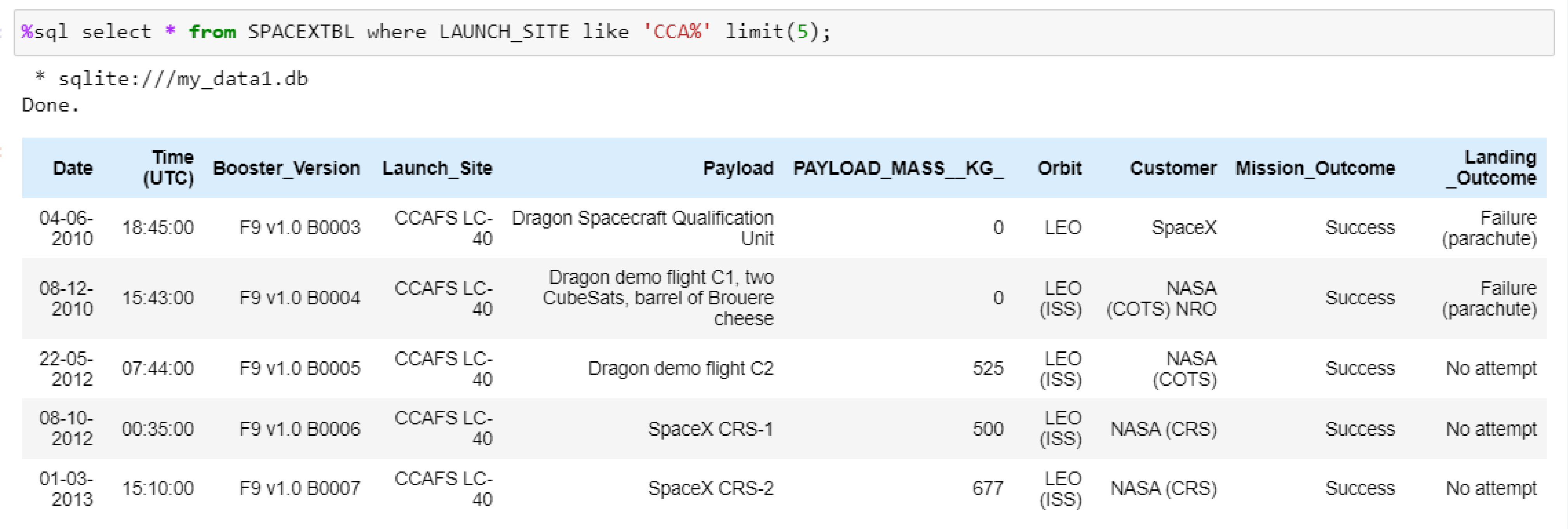
```
 * sqlite:///my_data1.db
Done.
```

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

We used the key word DISTINCT to show only unique launch sites from the SpaceX data.

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Present your query result with a short explanation here



```
%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit(5);
```

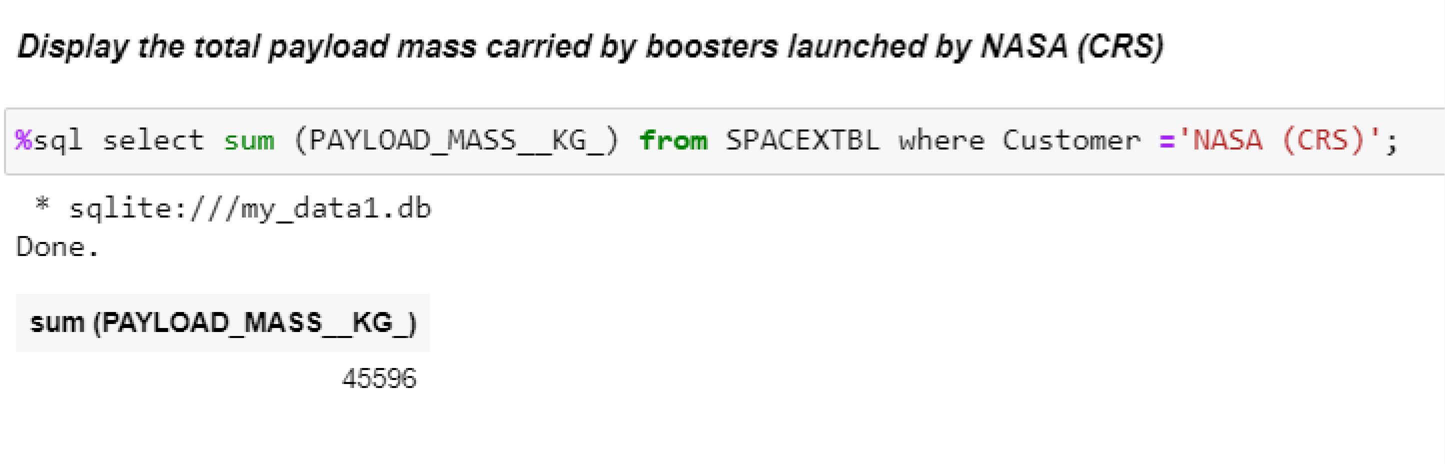
```
 * sqlite:///my_data1.db
Done.
```

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

- We used the word like ‘CCA%’ to find the launch site begin with CCA
- We use the word limit (5) to only show 5 records from SpaceX data

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here



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

```
%sql select sum (PAYLOAD_MASS__KG_) from SPACEXTBL where Customer ='NASA (CRS)';
```

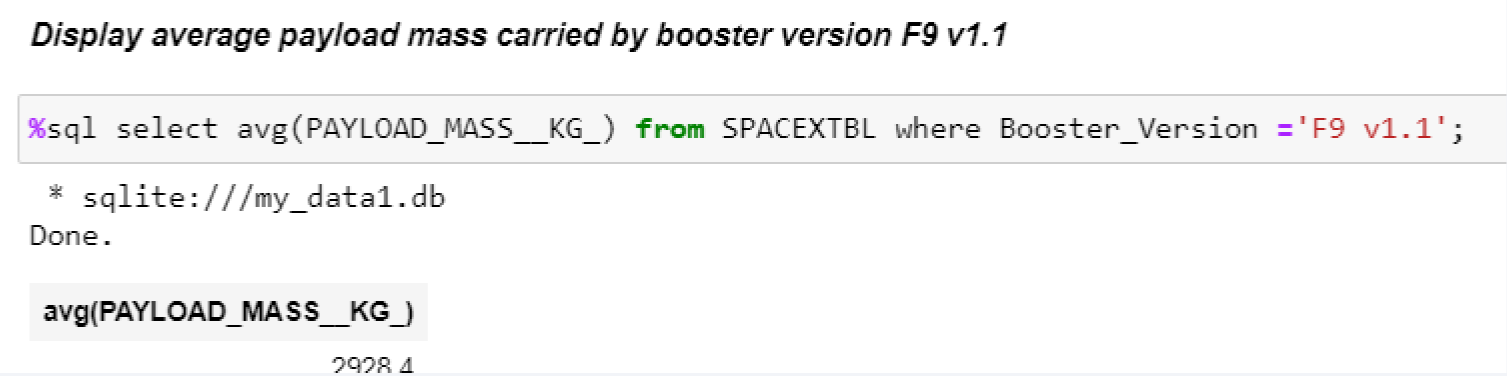
```
 * sqlite:///my_data1.db
Done.
```

sum (PAYLOAD_MASS__KG_)
45596

- We use function of SUM to find the total in column PAYLOAD_MASS_KG_
- We use the WHERE clause to perform calculation for Customer NASA (CRS) from the SpaceX data

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here



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

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version ='F9 v1.1';
```

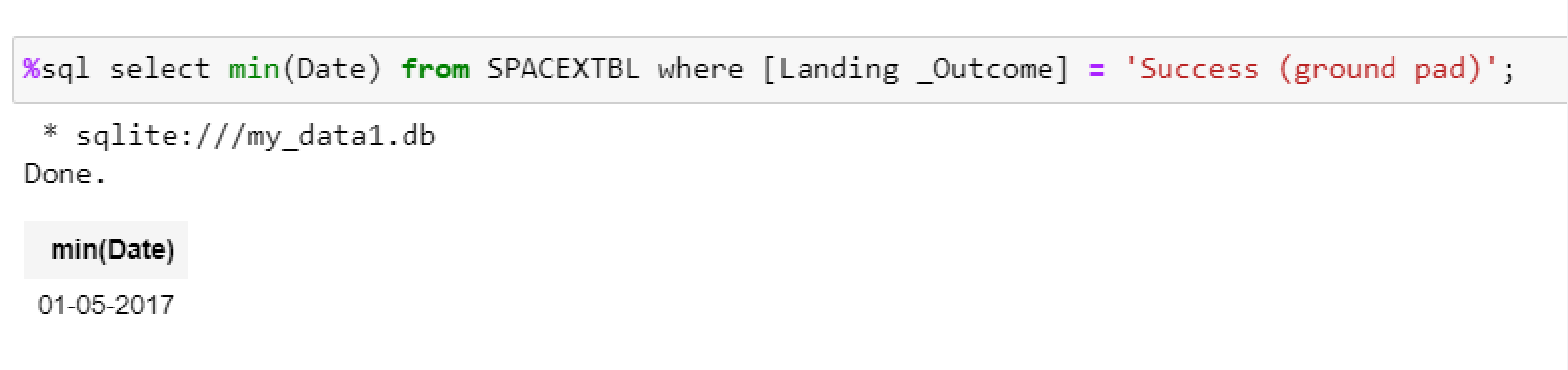
```
 * sqlite:///my_data1.db
Done.
```

avg(PAYLOAD_MASS__KG_)
2928.4

- We use function of AVG to find the average in column PAYLOAD_MASS_KG_
- We use the WHERE clause to perform calculation for Booster Version F9 v1.1 from the SpaceX data

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here



```
%sql select min(Date) from SPACEXTBL where [Landing _Outcome] = 'Success (ground pad)';
```

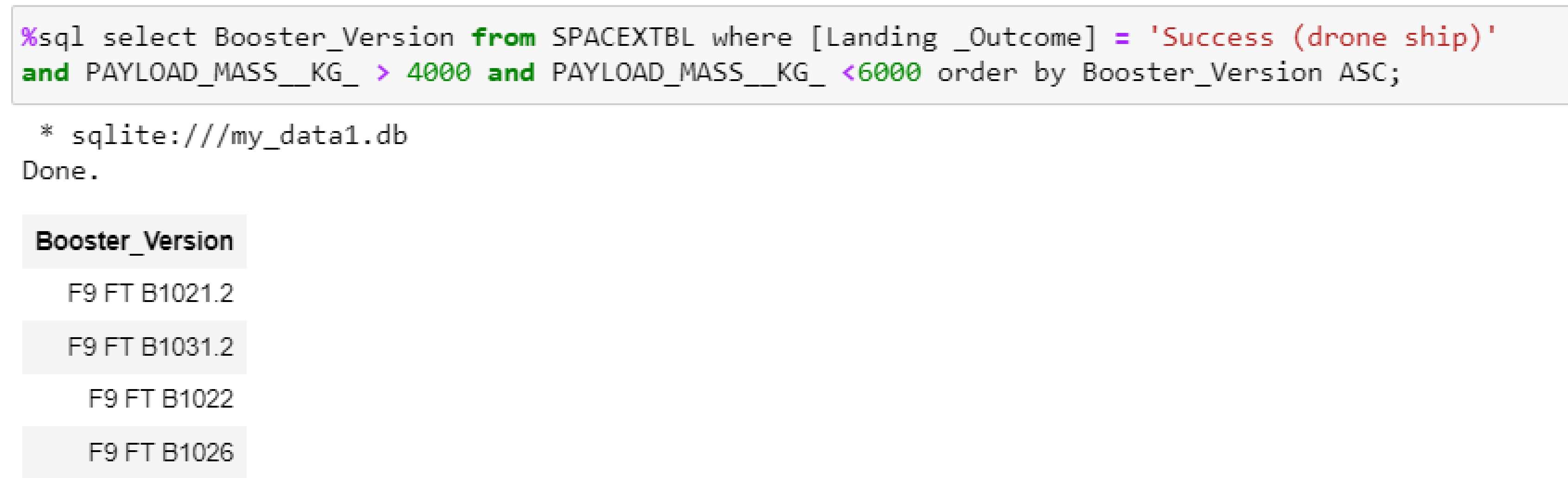
```
 * sqlite:///my_data1.db
Done.
```

min(Date)
01-05-2017

- We use function of MIN to find the earliest date in column DATE
- We use the WHERE clause to find from column Landing Outcome = 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
- Present your query result with a short explanation here



```
%sql select Booster_Version from SPACEXTBL where [Landing _Outcome] = 'Success (drone ship)'
and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ <6000 order by Booster_Version ASC;
```

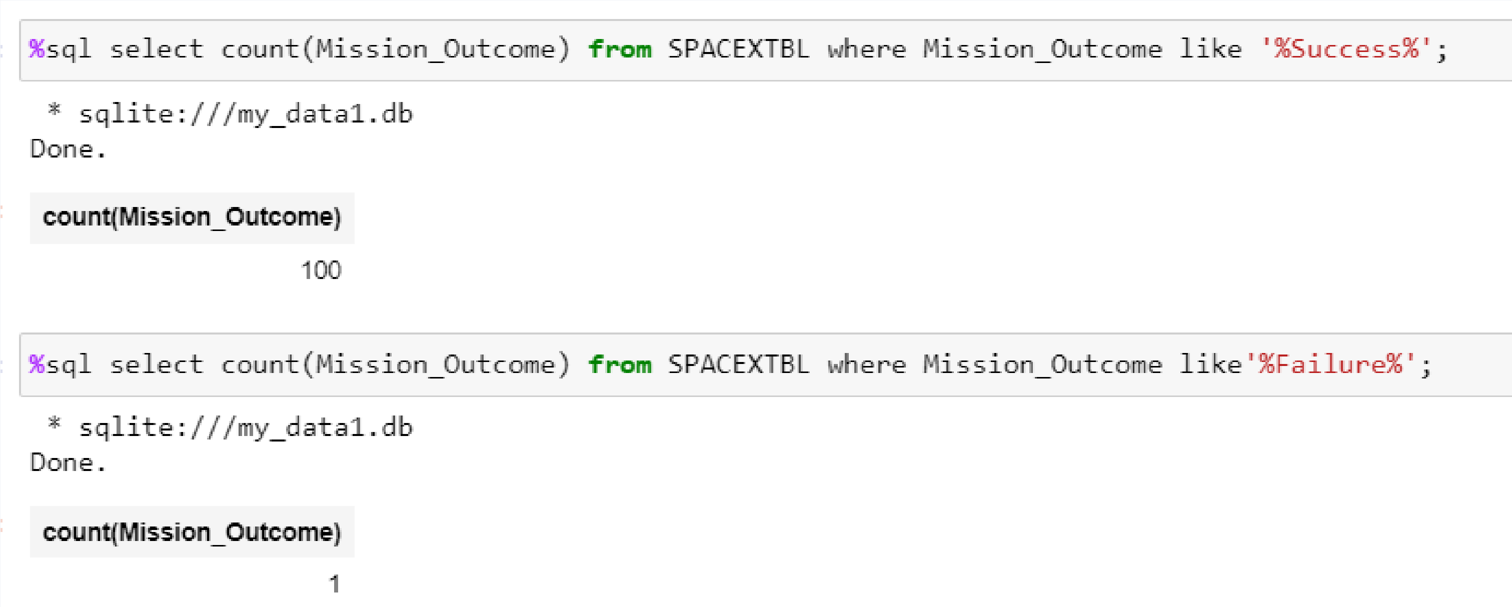
```
 * sqlite:///my_data1.db
Done.
```

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

- We used the WHERE clause to filter for boosters which have successfully landed on drone ship
- We used AND condition to find successful landing withpayload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

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



```
%sql select count(Mission_Outcome) from SPACEXTBL where Mission_Outcome like '%Success%';
```

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

count(Mission_Outcome)
100

```
%sql select count(Mission_Outcome) from SPACEXTBL where Mission_Outcome like'%Failure%';
```

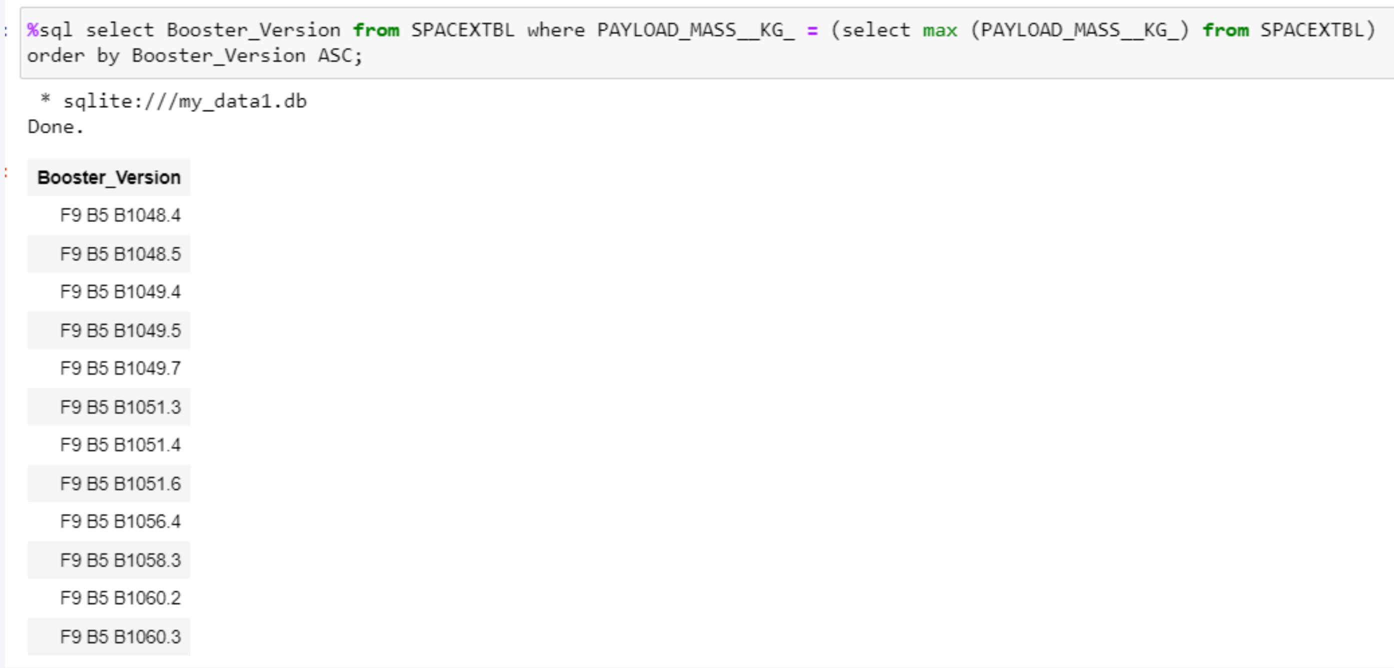
```
 * sqlite:///my_data1.db
Done.
```

count(Mission_Outcome)
1

- We used the COUNT function to count the number of failure and success in column Mission Outcomes
- We used wildcard like % to filter for WHERE Mission_Outcome was a success or failure

Boosters Carried Maximum Payload

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



```
%sql select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max (PAYLOAD_MASS__KG_) from SPACEXTBL)
order by Booster_Version ASC;
```

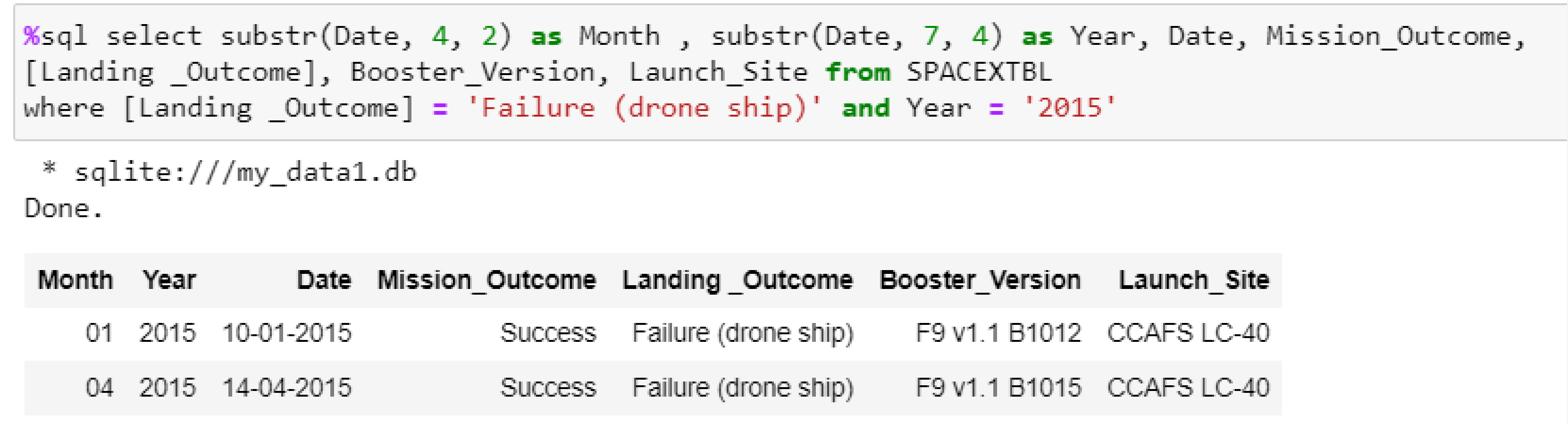
```
 * sqlite:///my_data1.db
Done.
```

Booster_Version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

We used WHERE clause and MAX function to find the booster have carried the maximum payload mass

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Present your query result with a short explanation here



```
%sql select substr(Date, 4, 2) as Month , substr(Date, 7, 4) as Year, Date, Mission_Outcome,
[Landing _Outcome], Booster_Version, Launch_Site from SPACEXTBL
where [Landing _Outcome] = 'Failure (drone ship)' and Year = '2015'
```

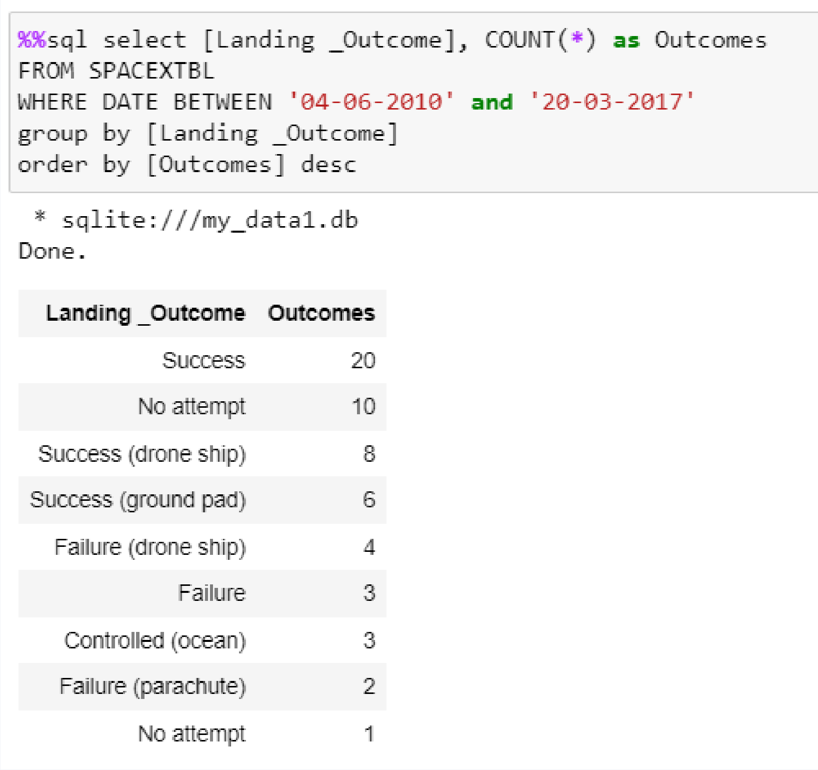
```
 * sqlite:///my_data1.db
Done.
```

Month	Year	Date	Mission_Outcome	Landing _Outcome	Booster_Version	Launch_Site
01	2015	10-01-2015	Success	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	2015	14-04-2015	Success	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

We used WHERE clause to find the failed landing outcomes in drone ship 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
- Present your query result with a short explanation here



```
%%sql select [Landing _Outcome], COUNT(*) as Outcomes
FROM SPACEXTBL
WHERE DATE BETWEEN '04-06-2010' and '20-03-2017'
group by [Landing _Outcome]
order by [Outcomes] desc
```

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

Landing _Outcome	Outcomes
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Failure	3
Controlled (ocean)	3
Failure (parachute)	2
No attempt	1

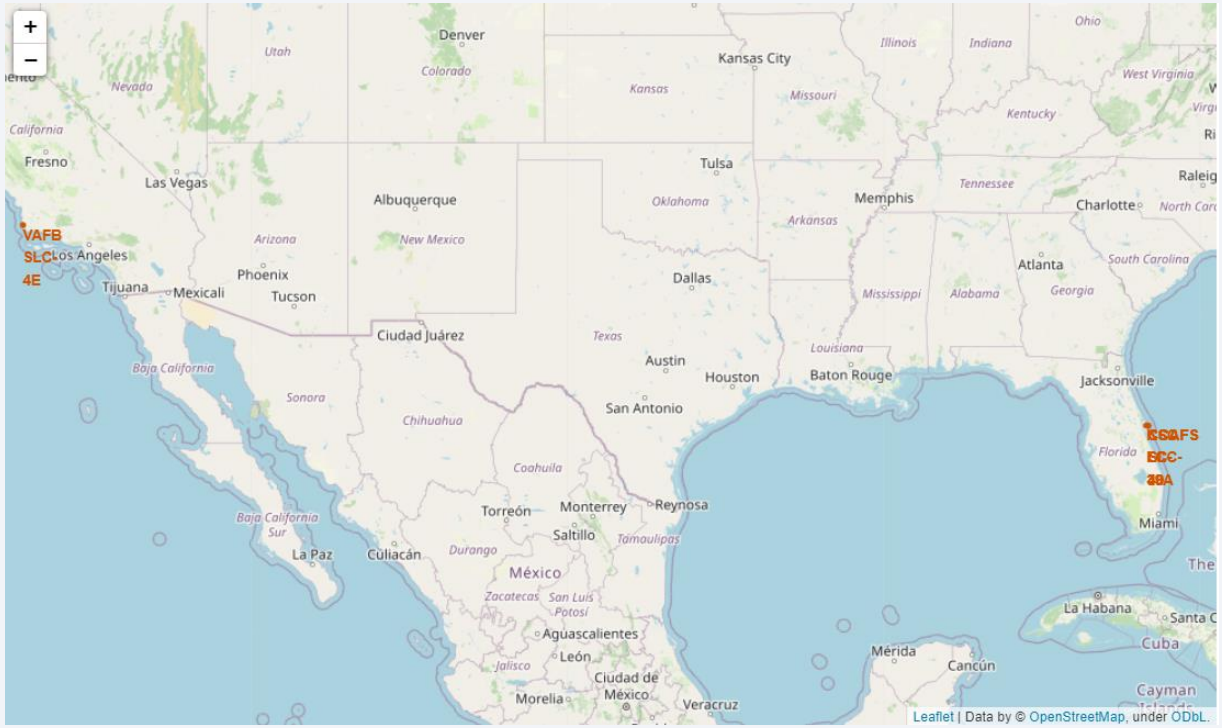
- We used COUNT function to count the frequency for each landing outcomes
- We used WHERE clause to find the data between the mentioned data
- We used ORDER BY DESC function to sort the number of outcomes in descending order

Section 3

Launch Sites Proximities Analysis

All launch sites global map markers

- 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



All SpaceX launch sites are loctated in United States of America coasts, Florida and California

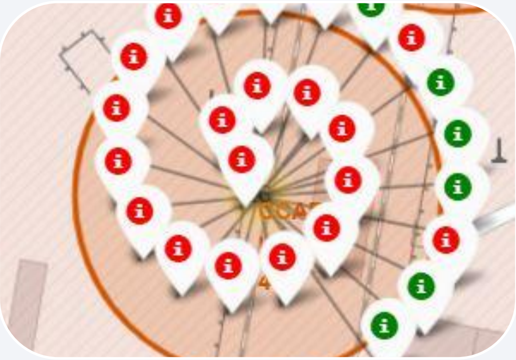
Status of each launch sites by markers

- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map

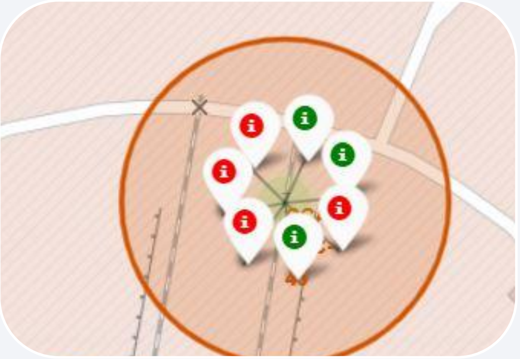
- Explain the important elements and findings on the screenshot

CALIFORNIA LAUNCH SITES

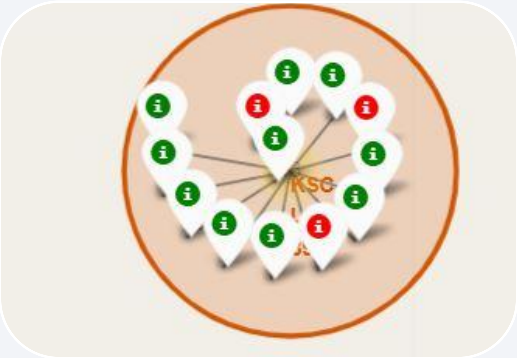
FLORIDA LAUNCH SITES



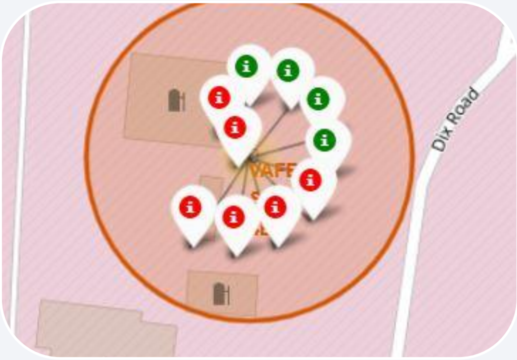
CCAFS LC 40



CCAFS SLC 40



KSC LC 39

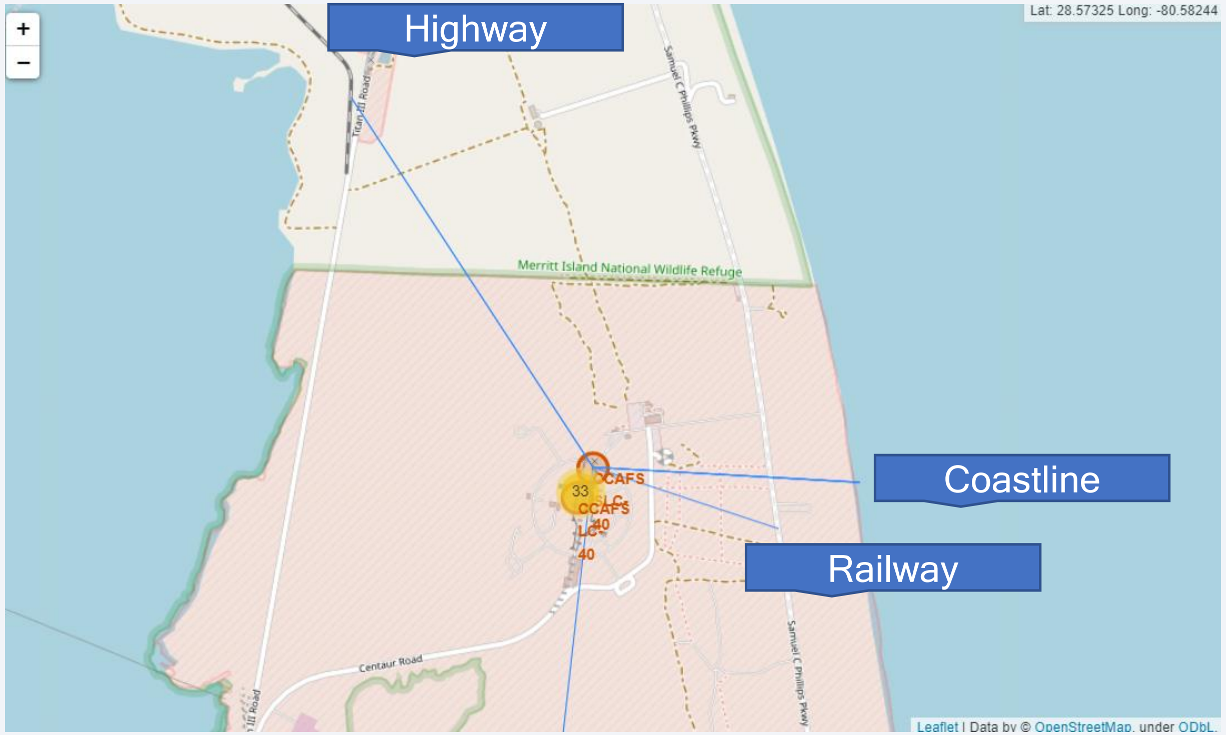


VAFB SLC 4E

Green Marker - Successful launches
Red Marker - Failed launches

Launch Site distance to its proximities

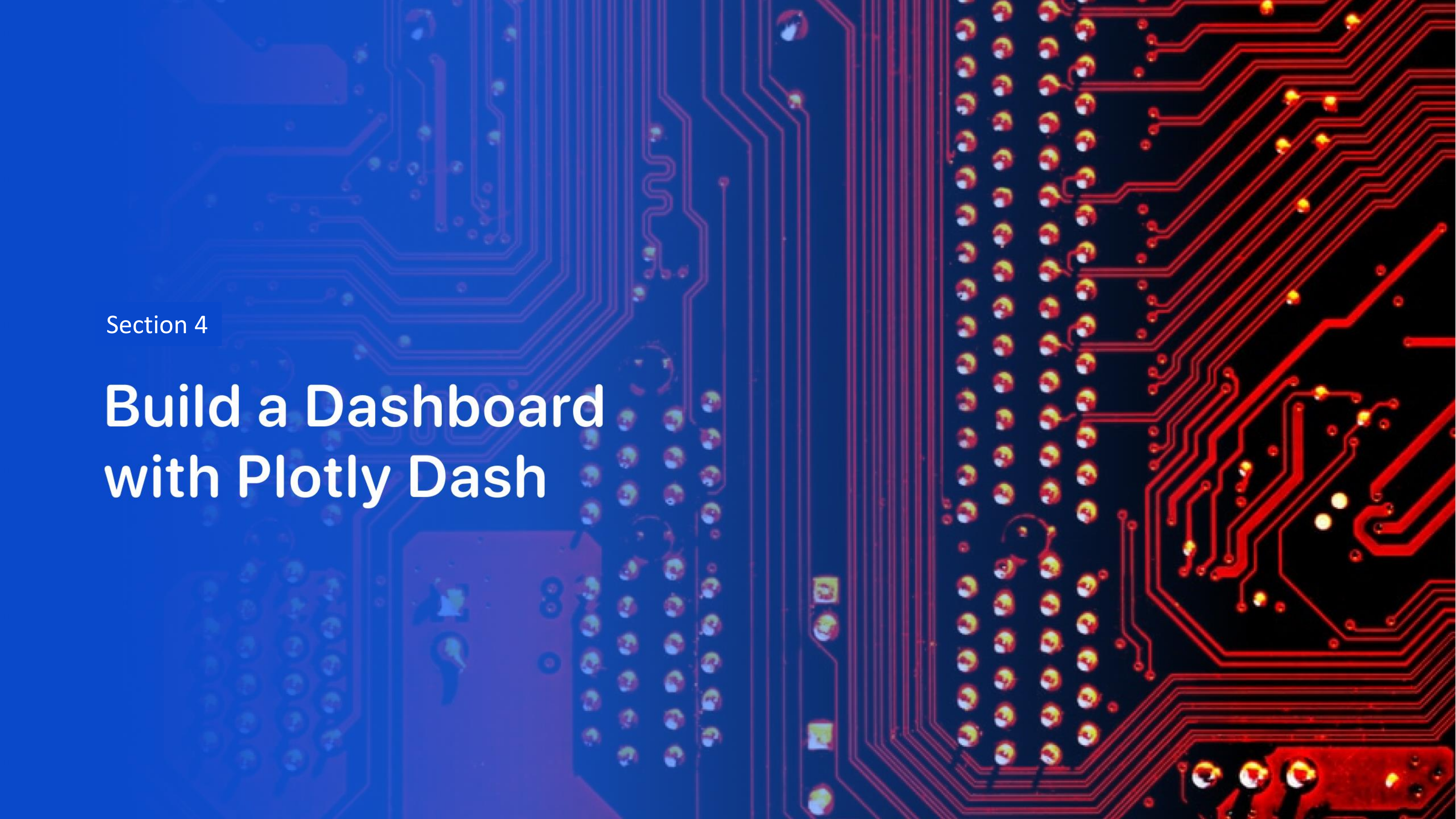
- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed
- Explain the important elements and findings on the screenshot


Lat: 28.57325 Long: -80.58244
Highway
Coastline
Railway
Merritt Island National Wildlife Refuge
Samuel C Phillips Pkwy
Titan III Road
Centaur Road
33
CCAFS
CCAFS LC-40
Leaflet | Data by © OpenStreetMap, under ODbL.

Questions	Answer
Are launch sites in close proximity to railways?	Yes
Are launch sites in close proximity to highways?	No
Are launch sites in close proximity to coastline?	Yes
Do launch sites keep certain distance away from cities?	Yes

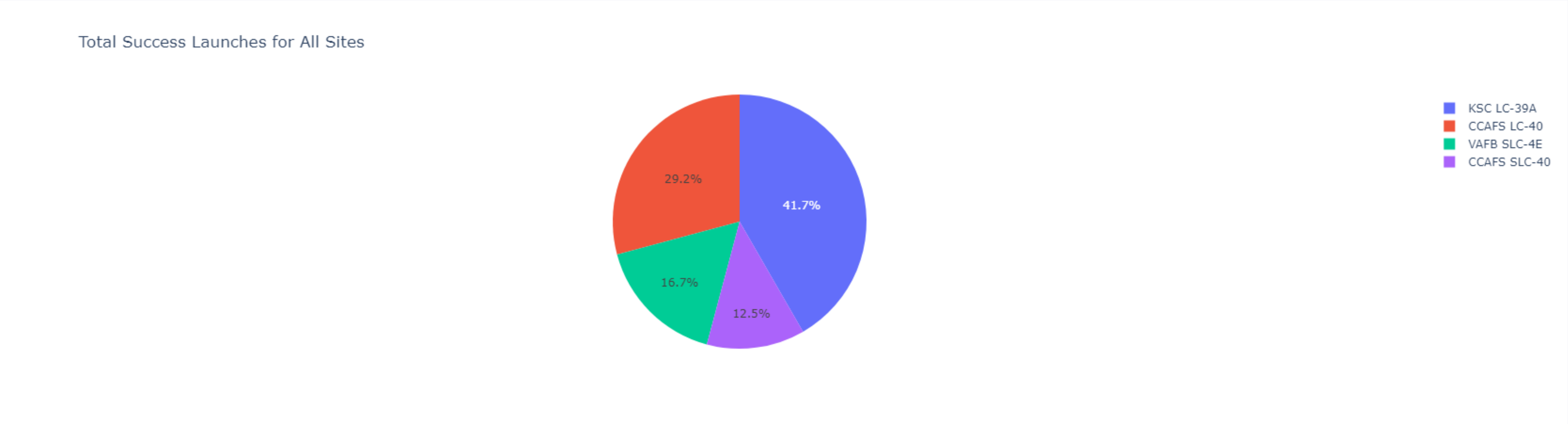
Section 4

Build a Dashboard with Plotly Dash



Launch success count for all sites

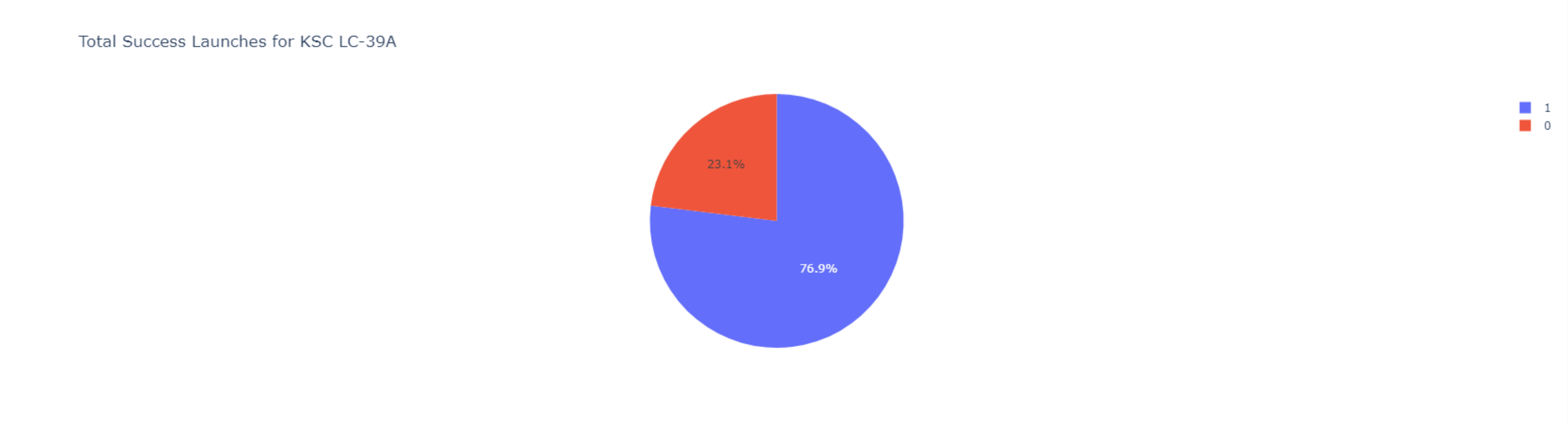
- Show the screenshot of launch success count for all sites, in a piechart
- Explain the important elements and findings on the screenshot



KSC LC-39A has the highest launch success from all sites.

Launch site with highest launch success ratio

- Show the screenshot of the piechart for the launch site with highest launch success ratio
- Explain the important elements and findings on the screenshot



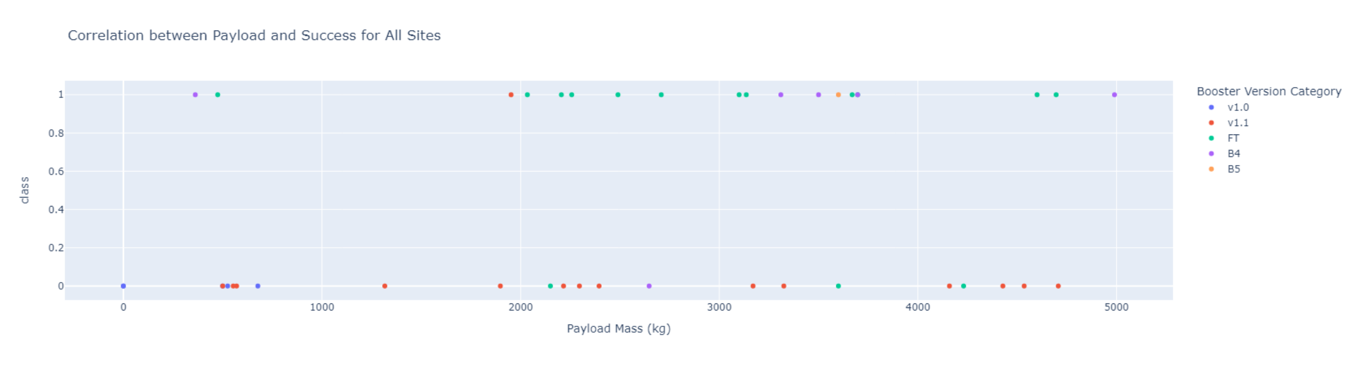
KSC LC-39A has a record of 76.9% of success rate whereas failure rate at 23.1%

Scatter plot of Payload vs. Launch Outcome

- 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.

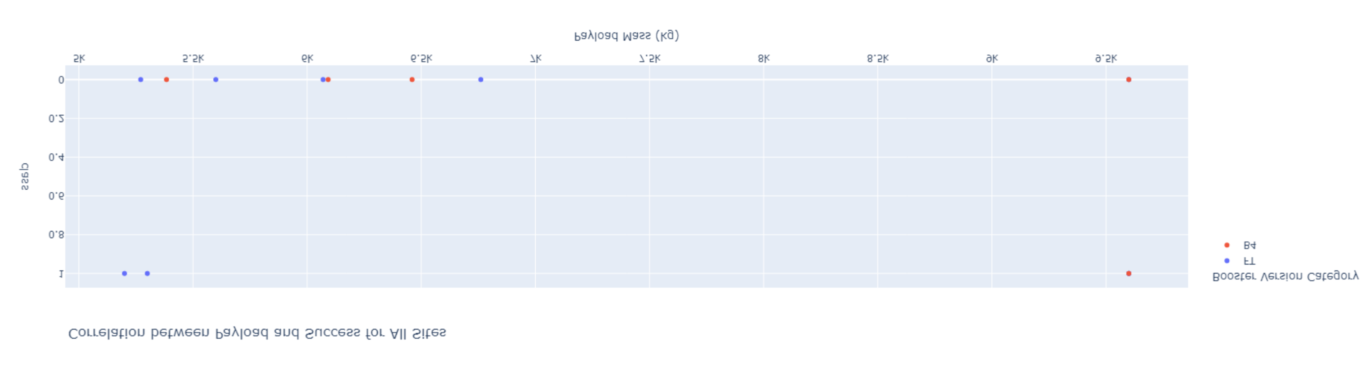
Payload mass,

0 kg - 5000 kg



Payload mass,

5000 kg – 10000 kg



It is observed that low weighted payload mass have largest success rate as compared to high weighted payload mass.

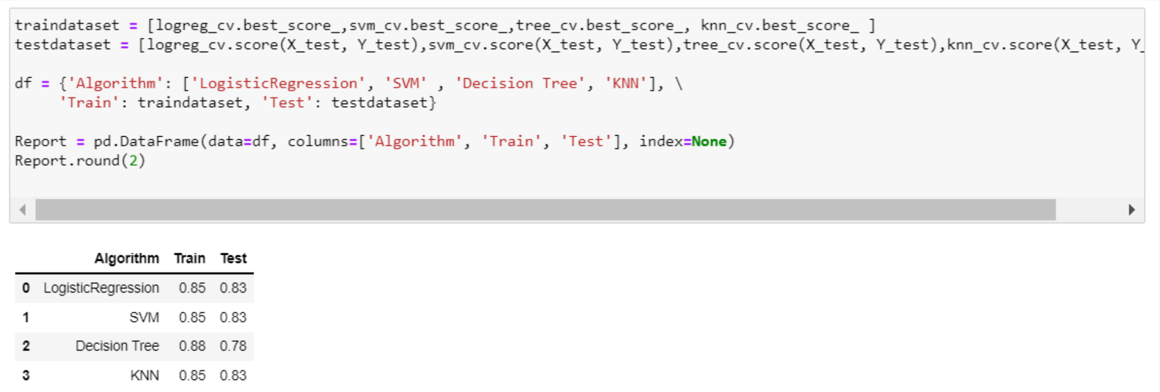
Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart

- Find which model has the highest classification accuracy

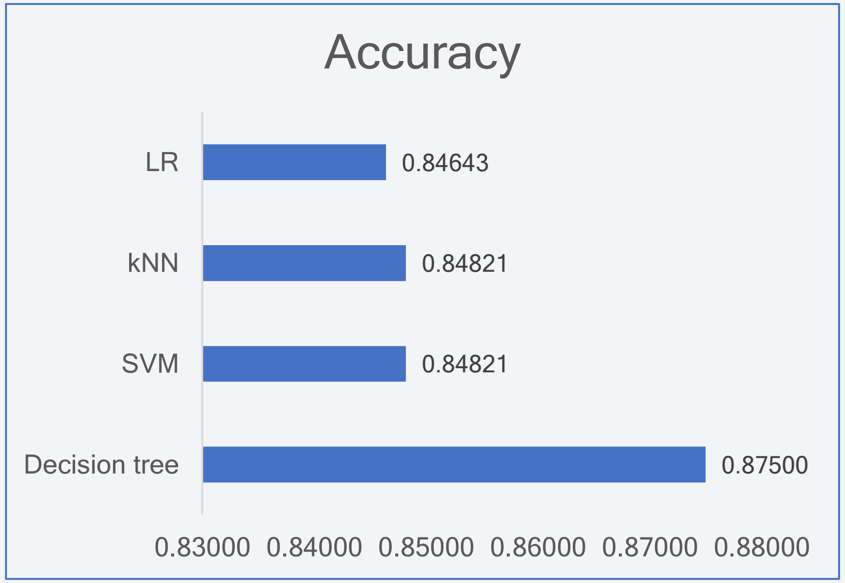


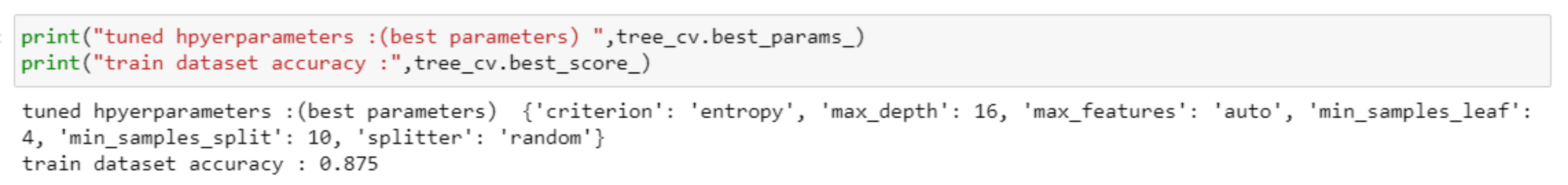
```
traindataset = [logreg_cv.best_score_,svm_cv.best_score_,tree_cv.best_score_, knn_cv.best_score_ ]
testdataset = [logreg_cv.score(X_test, Y_test),svm_cv.score(X_test, Y_test),tree_cv.score(X_test, Y_test),knn_cv.score(X_test, Y_

df = {'Algorithm': ['LogisticRegression', 'SVM' , 'Decision Tree', 'KNN'], \
     'Train': traindataset, 'Test': testdataset}

Report = pd.DataFrame(data=df, columns=['Algorithm', 'Train', 'Test'], index=None)
Report.round(2)
```

	Algorithm	Train	Test
0	LogisticRegression	0.85	0.83
1	SVM	0.85	0.83
2	Decision Tree	0.88	0.78
3	KNN	0.85	0.83



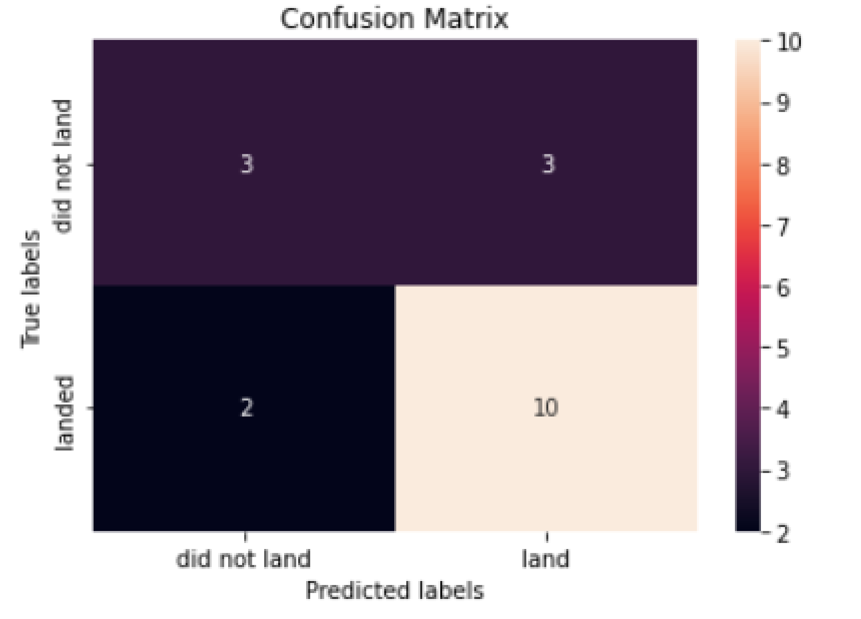


```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("train dataset accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 16, 'max_features': 'auto', 'min_samples_leaf':
4, 'min_samples_split': 10, 'splitter': 'random'}
train dataset accuracy : 0.875
```

Confusion Matrix

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



The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes.

Issue:

- False positive i.e. unsuccessful landing marked as successful landing by the classifier.
- False negative i.e. successful landing marked as unsuccessful landing by classifier

Conclusions

- The larger amount of the flight number at the launch site, the greater the success rate at a launch site will be.
- Orbit ES-L1, GEO, HEO and SSO have 100% success rate at the landing outcomes.
- Success rate is on an increasing trend from 2013 till 2020.
- Low weighted payload mass have largest success rate as compared to high weighted payload mass.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this project.

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!