Winning Space Race with Data Science

Yeshwanth N

March 11th ,2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection Through API or Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL + Data Visualization
  - Interactive Visual Analysis with Plotly + Folium
  - Landing Prediction with Machine Learning.
- Summary of all results
  - EDA Results
  - Dashboard Screen shots of Folium
  - Predictive analytics results of different machine Learning Models.

Introduction

- Project background and context
  - On its website, Space X promotes Falcon 9 rocket launches for 62 million dollars whereas other companies charge upwards of 165 million dollars for each launch. A large portion of the savings is due to Space X's ability to reuse the first stage. Hence, if we can figure out whether the first stage will land and figure out how much a launch will cost. If another business wishes to submit a proposal for a rocket launch against space X, they can use this information. The project's whole objective is to build a pipeline for machine learning that can predict if the initial stage will land successfully.
- Problems you want to find answers
  - What Factors determine if the rocket with land successfully.
  - Conditions that are necessary for operation and determine the success rate of landing the rocket in Phase I.

Section 1

Methodology

Methodology

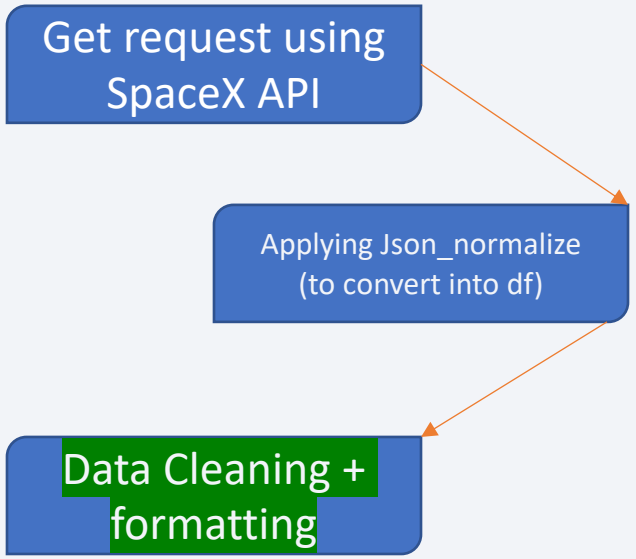
Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

Data Collection

- Describe how data sets were collected.
  - Two approaches were used one was by using get request from the SpaceX API and other approach was via web scraping from Wikipedia for Falcon 9 launch records with Beautiful Soup.
  - In the First Approach we used get request and decoded the response content as JSON and turn it into a pandas data frame using json_normalize() and cleaned the data, checked for missing values and deal with missing values.
  - Later Approach in web scrapping, we have applied Beautiful Soup to extract the launch record as HTML table, the later parse the table and covert into a pandas dataframe and use in analysis.

Data Collection – SpaceX API



Link:

- https://github.com/yesh2805/Falcon9_Landing-Prediction/blob/main/Data%20Collection.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 [9]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_a
```

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

In [10]:
```
response.status_code
```

Out[10]: 200

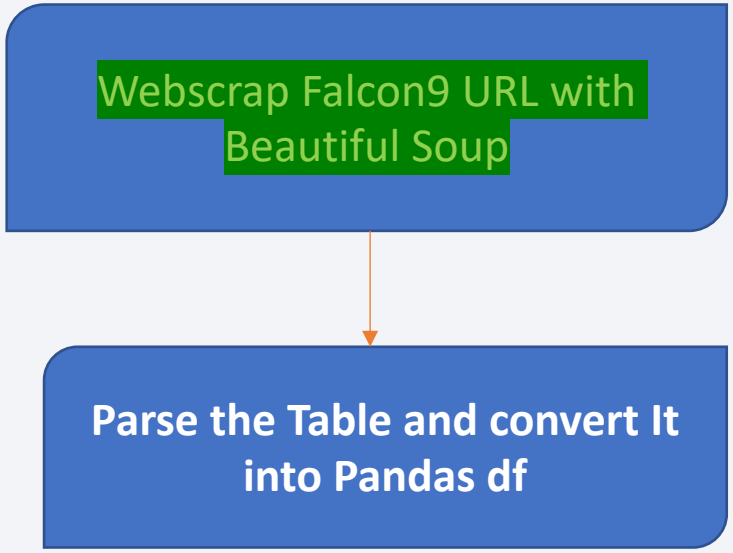
Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

In [14]:
```
# Use json_normalize meethod to convert the json result into a dataframe
response = requests.get(static_json_url)
response.json()
data = pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

In [15]:
```
# Get the head of the dataframe
data.head()
```

Data Collection – web Scraping



- Link: https://github.com/yesh2805/Falcon9_Landing-Prediction/blob/main/Data%20Collection-labs-webscraping.ipynb

In [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
import requests

response = requests.get(static_url).text
```

Create a `BeautifulSoup` object from the HTML `response`

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

In [7]:
```
# Use soup.title attribute
# Use soup.title attribute
soup.title
```

Out[7]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

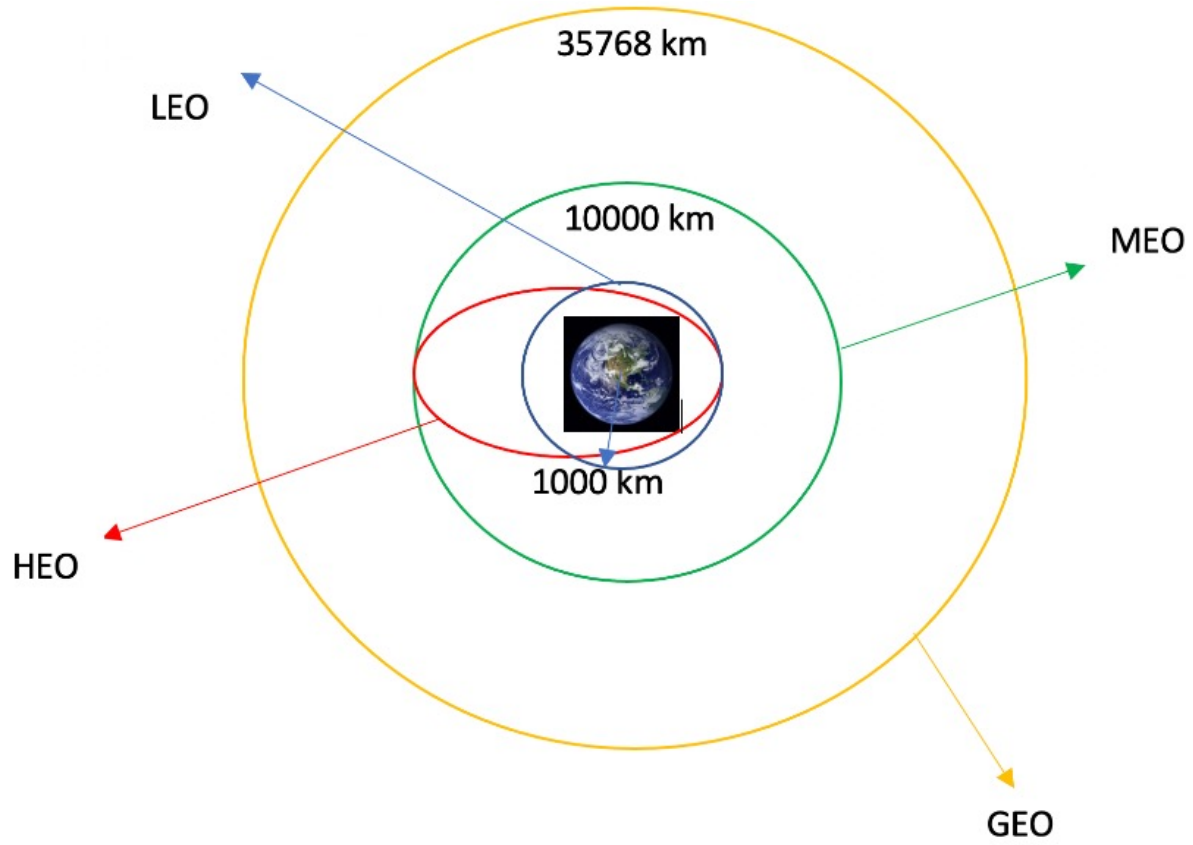
In [8]:
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`

html_tables = soup.find_all('table')
```

Starting from the third table is our target table contains the actual launch records.

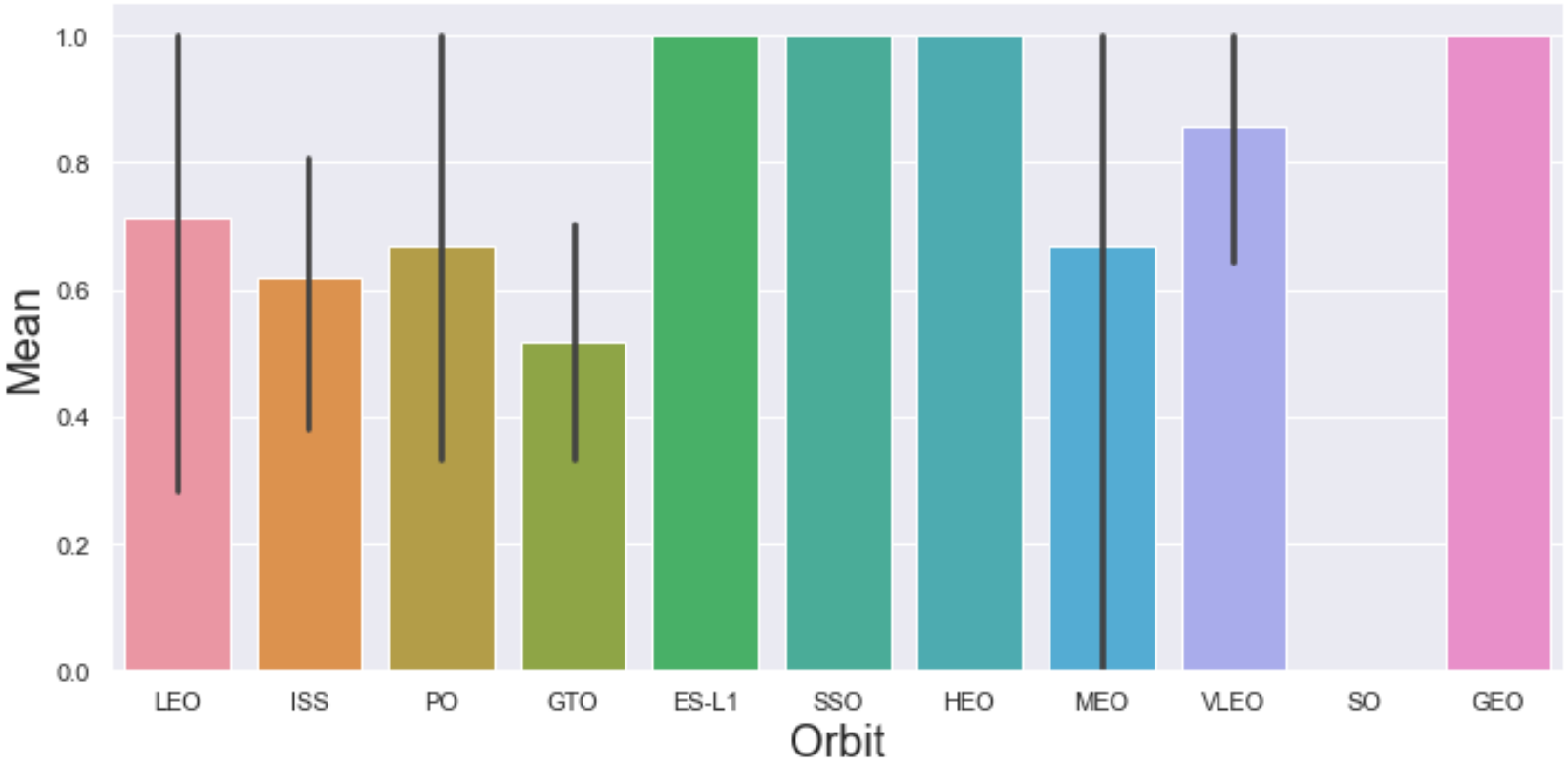
Data Wrangling

- Process of cleaning complex data sets for Exploratory Data Analysis and find training labels.
- Calculate the number of launches on each location and find out number of occurrence of mission outcome per orbit.
- Finally, we create the landing outcome label for further analysis and export end result to CSV.

Link:
https://github.com/yesh2805/Falcon9_Landing-Prediction/blob/main/Data%20Wrangling.ipynb

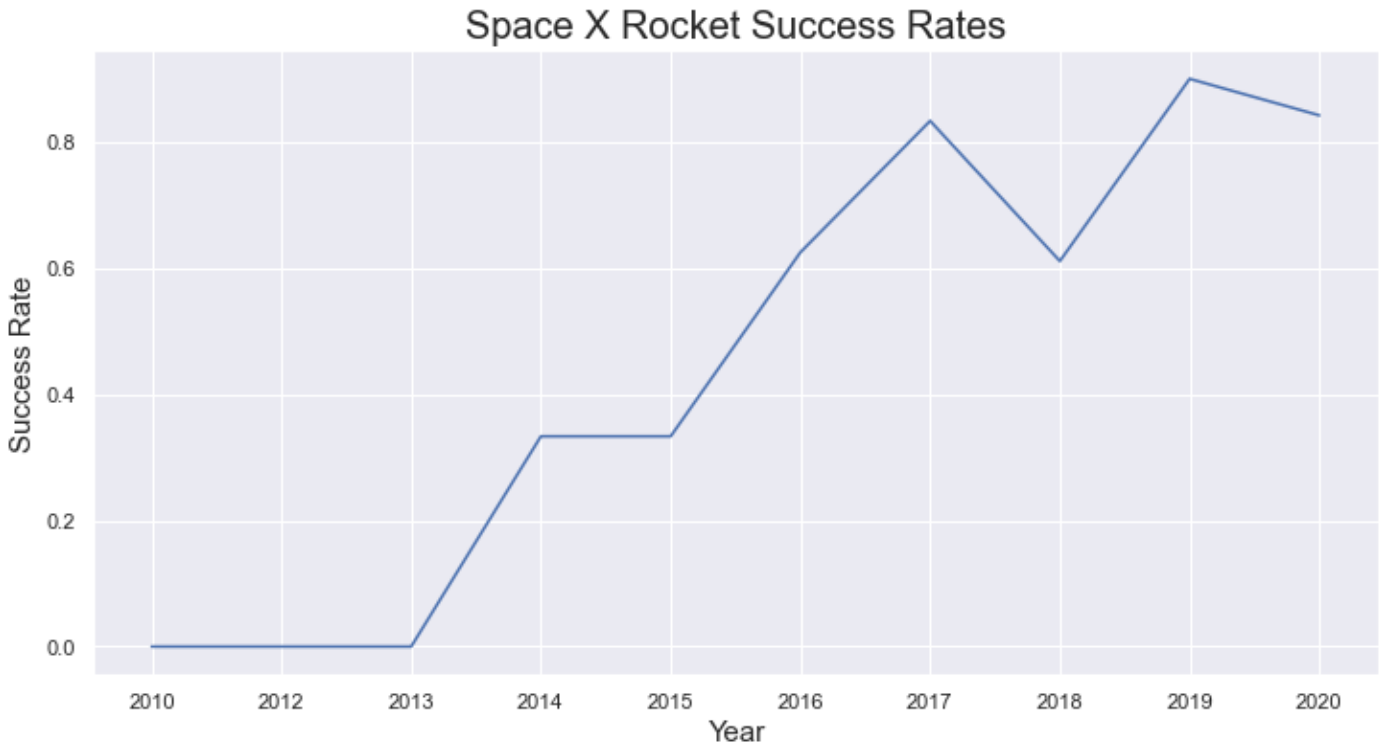


EDA with Data Visualization



Visualizing the relationship between success rate of each orbit with a Historgam.

EDA with Data Visualization



- Line graph which shows the Space X Rocket Success Rates over the years from 2010 to 2020.
- All the further EDA can be found at the below link:
- Link:
- https://github.com/yesh2805/Falcon9_LandingPrediction/blob/main/EDA%20Part%203.ipynb

EDA with SQL

- By Applying EDA with SQL to get into the insights from the data and by using following SQL queries.
- Finding out Unique launch sites names and displayed 5 launch sites which begins with ‘CCA’
- Displaying average payload mass carried by boosters launched by NASA (CRS) + booster version F9 v1.1.
- Finding the data when first successful landing outcome in ground pad was achieved.
- Displaying the names of boosters which have success in drone ship and have payload mass greater than 400 but less than 6000.
- Finding out the total number of successful and failure mission outcomes.
- Listing out the failed landing outcomes in drone ship, their booster versions, and launch sites names for in year 2015.
- Rank the count of landing outcomes or success between a given dates in descending order.

Link:

https://github.com/yesh2805/Falcon9_Landing-Prediction/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- With an Interactive Map marked all launch sites and using latitude and longitudes of each launch sites added marker objects such circles, lines to identify the success or failure of launches for each site. Necessary labels has been added.
- Assigned launch outcomes to ‘0’ or ‘1’ classes with different colors on the map and with these color labeled makers we have identified launch sites possessing high success rate.
- Finally, we have used Haversine’s formula to calculate the distance of launch sites to various proximity subject and answered the following questions:
  - To find out how close the launch sites with railways, highways and coastlines?
  - How close the launch sites with near cities?

Link: https://github.com/yesh2805/Falcon9_Landing-Prediction/blob/main/Folium%20Dashboard.ipynb

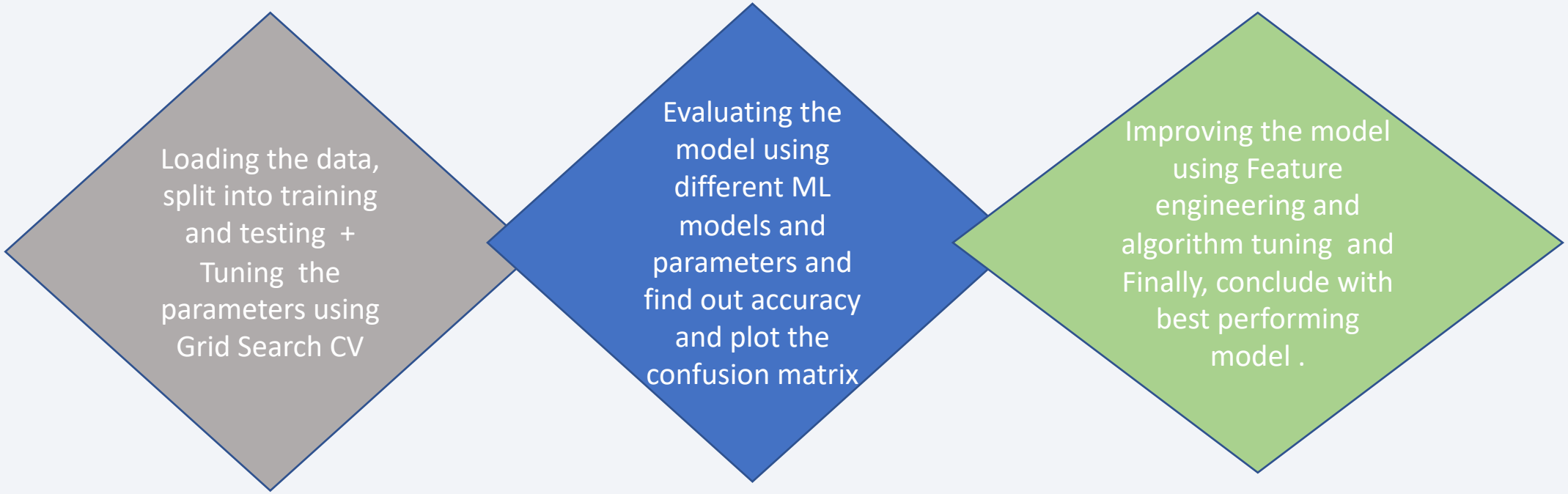
Build a Dashboard with Plotly Dash

- Built an interactive dashboard with Plotly dash and plotted pie charts which shows the total launches by a certain sites.

- Plotted a scatter graph showing the relationship with outcome and Payload Mass(kg) for a different booster version.

- Link: https://github.com/yesh2805/Falcon9_Landing-Prediction/blob/main/Plotlydash.app

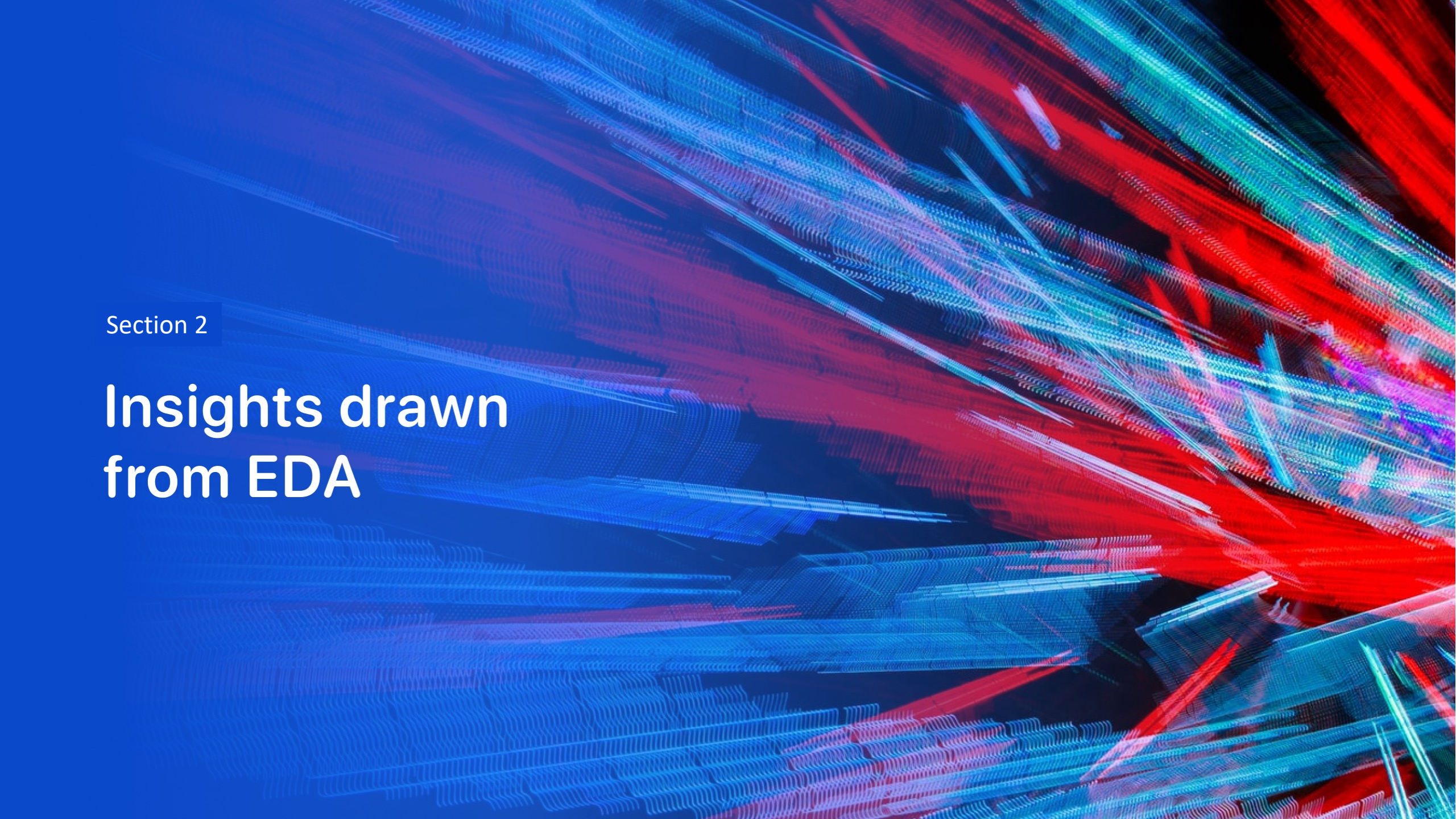
Predictive Analysis (Classification)

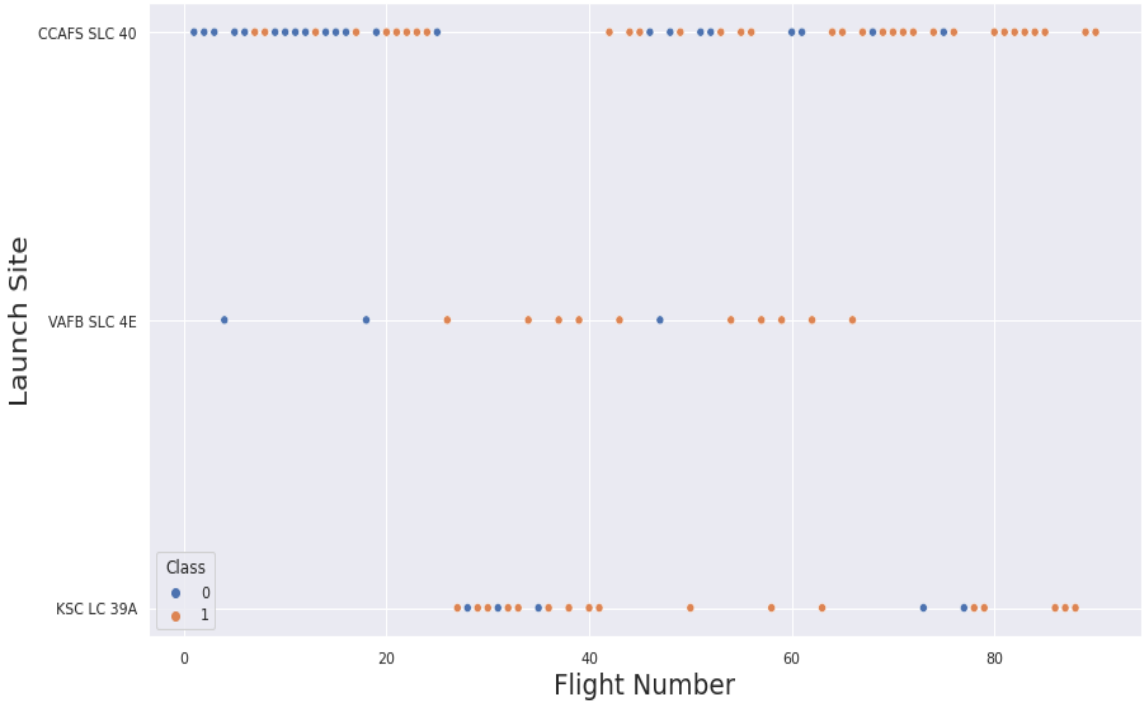


- Link: https://github.com/yesh2805/Falcon9_Landing-Prediction/blob/main/Landing%20Prediction%20using%20Machine%20Learning.ipynb

Results

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


Section 2
Insights drawn from EDA



Flight Number vs. Launch Site

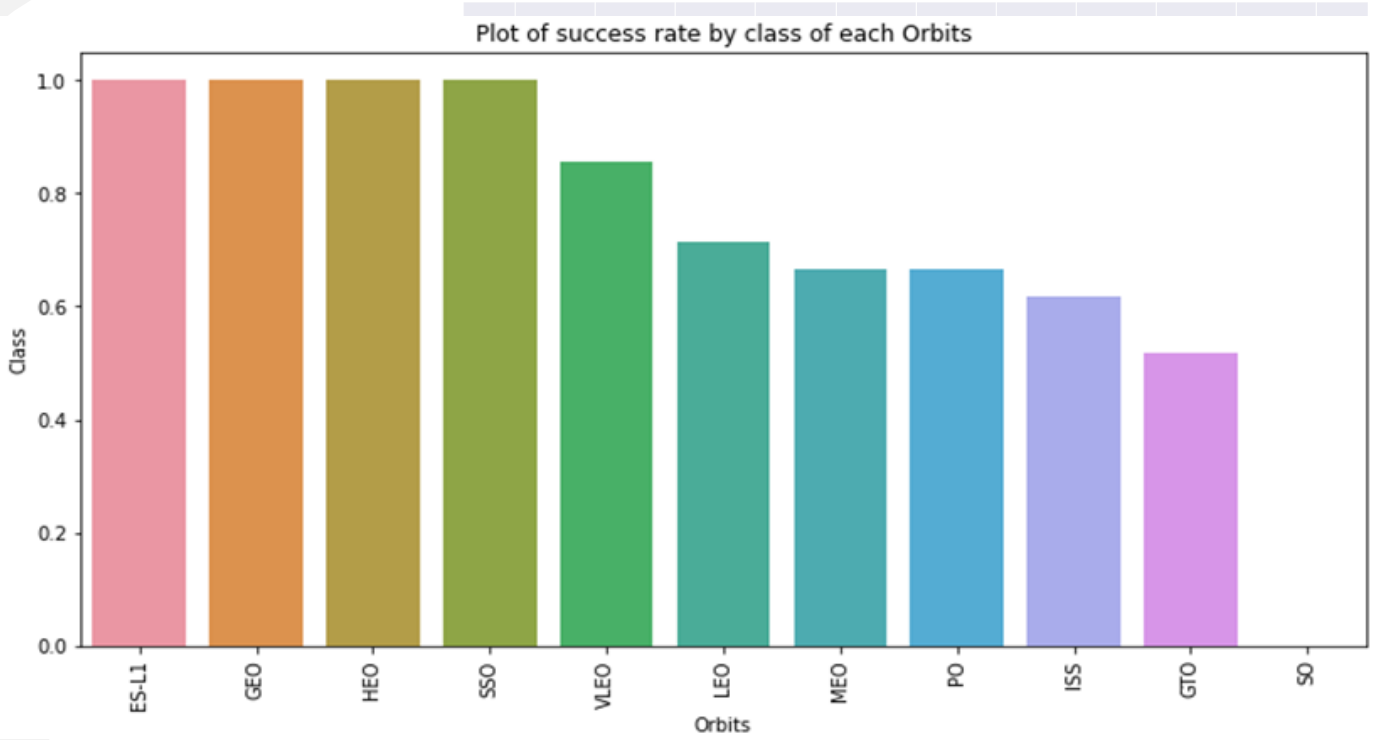
- Scatter plot shows that Number of flights of the launch sites which shows that CCAFS SLC 40 has the greatest number of launches when compared to the rest two.

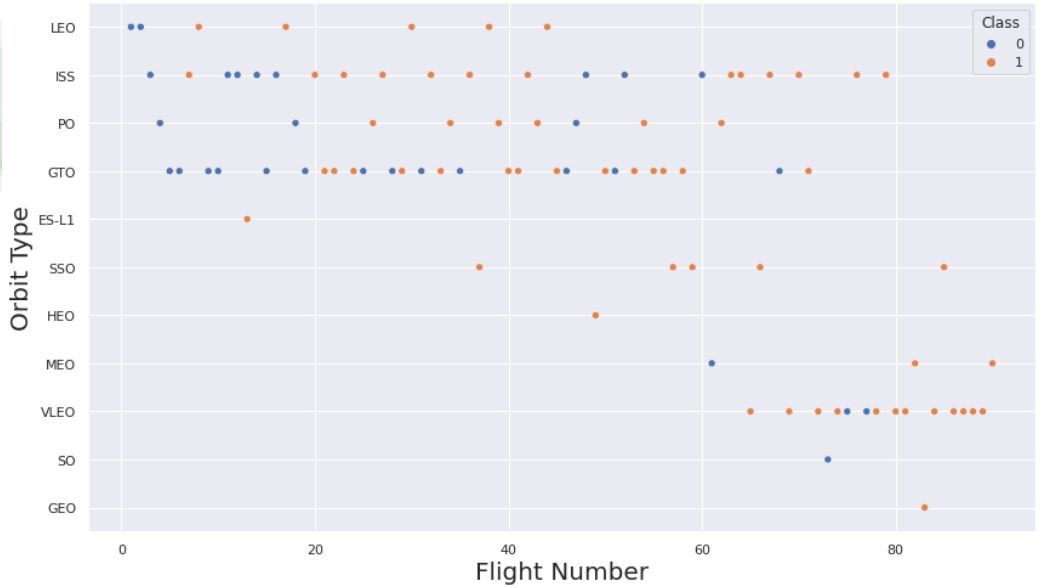
Payload vs. Launch Site

- This scatter plot even though show that the greater the payload the higher chance of success but this cannot be the only one factor which determines the success entirely.

Success Rate vs. Orbit Type

- Some orbits has higher chance of success that are ES-L1, GEO, HEO , SSO, VLEO has most success rate compared to the rest.



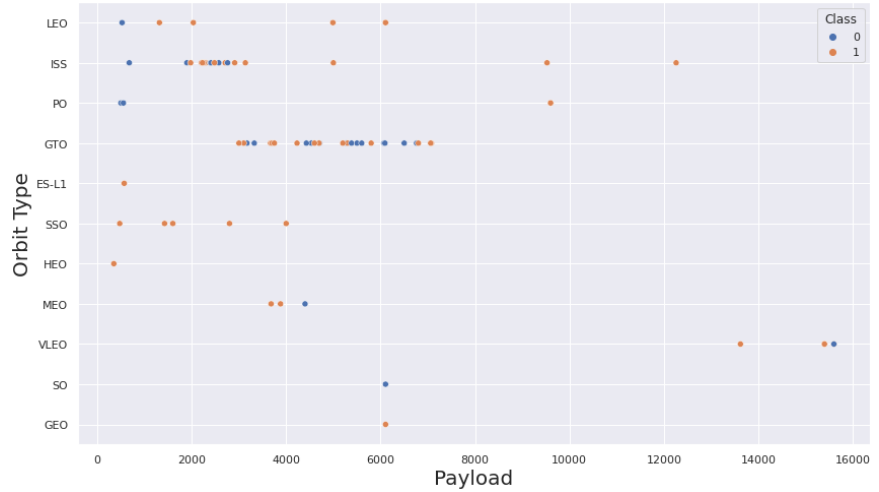


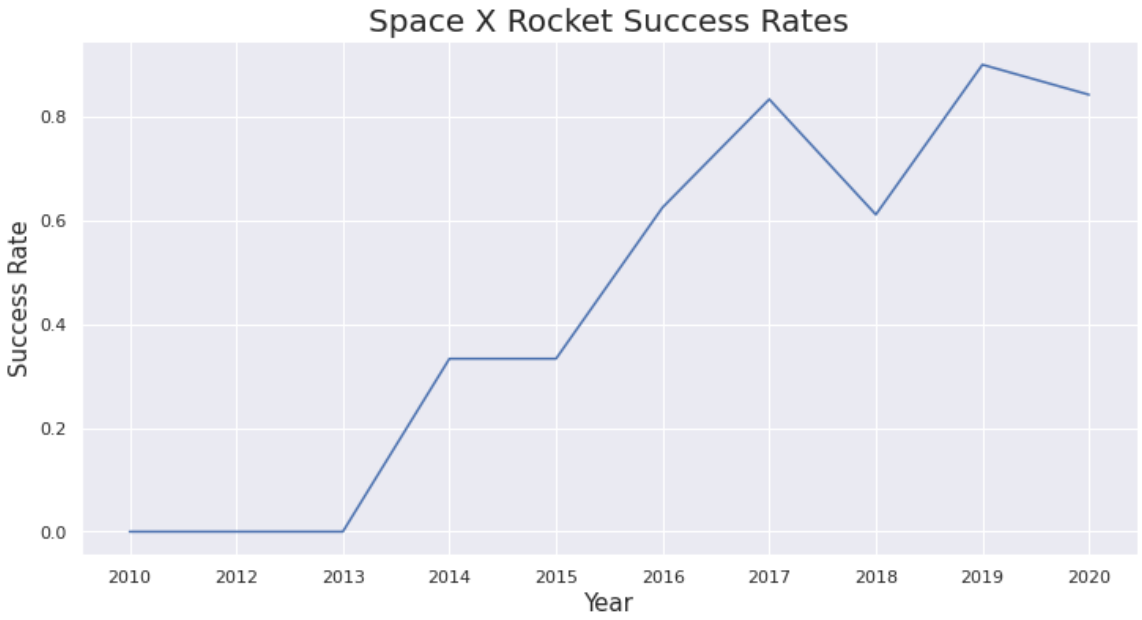
- The scatter plot shows that the in the LEO orbit success is related to the number of flights in the GTO orbit, there is no relationship between flight number and the orbit.

Flight Number vs. Orbit Type

Payload vs. Orbit Type

- The Heavier the payload the more successful chance of landing in PO, LEO and ISS orbits however GTO orbit seem to depict no relations between attributes.





Launch Success Yearly Trend

- From the line graph success rate from 2013 has been increased significantly till 2020.

All Launch Site Names

Task 1

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

In [17]:
```
%sql select distinct Launch_Site FROM SPACEXTBL;
```

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

Out[17]:

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

- We used Keyword DISTINCT to show only unique launch sites from the data(SPACEXTBL).

Launch Site Names Begin with 'CCA'

- Use of query with keyword LIKE which begin with ‘CCA’ for the launch sites.
- SQL query:

%sql SELECT LAUNCH_SITE FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;

Task 2

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

In [18]:
```
%sql select * from SPACEXTBL where Launch_Site LIKE 'CCA%'
```

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

Out[18]:

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
03-12-2013	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES	Success	No attempt
06-01-2014	22:06:00	F9 v1.1	CCAFS LC-40	Thaicom 6	3325	GTO	Thaicom	Success	No attempt
18-04-2014	19:25:00	F9 v1.1	CCAFS LC-40	SpaceX CRS-3	2296	LEO (ISS)	NASA (CRS)	Success	Controlled (ocean)

Task 3

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

In [23]:
```
%sql select sum(PAYLOAD_MASS__KG_) FROM SPACEXTBL where Customer = 'NASA (CRS)';
```

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

Out[23]:

sum(PAYLOAD_MASS__KG_)
45596

Total Payload Mass

- Calculated the total payload carried by boosters from NASA

Task 4

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

In [30]:
```
#%sql select * from SPACEXTBL;
%sql select avg(PAYLOAD_MASS__KG_) FROM SPACEXTBL where Booster_version LIKE 'F9 v1.1%';
#%sql select avg(PAYLOAD_MASS__KG_) from SPACEXDTASET where booster_version like 'F9 v1.1%'
```

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

Out[30]:

avg(PAYLOAD_MASS__KG_)
2534.6666666666665

Average Payload Mass by F9 v1.1

- Calculated the average payload by booster version F9 v1.1

Task 5

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

Hint:Use min function

```
[16]:
#%sql PRAGMA table_info(SPACEXTBL);
%sql select MIN(Date) from SPACEXTBL where [Landing _Outcome] LIKE 'Success (ground pad)';
#%sql select distinct Landing_Outcome from SPACEXTBL;
#%sql select * from  information_schema.columns where table_name = 'SPACEXTBL';
#%sql Select * from SPACEXTBL;
#%sql show columns from my_data1.db.SPACEXTBL;
#%sql USE my_data1.db DESCRIBE SPACEXTBL;
#%sql SELECT * FROM SPACEXTBL WHERE OBJECT_ID = OBJECT_ID('SPACEXTBL');
#%sql SELECT MIN(DATE) FROM SPACEXTBL WHERE Landing _Outcome = 'Success (ground pad)'
#%sql SELECT MIN(DATE) AS "First Successful Landing" FROM SPACEXTBL WHERE LANDING _OUTCOME = 'Success (ground pad)';
```

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

[16]:

Date
22-12-2015

First Successful Ground Landing Date

- First Success Landing date has been found using MIN() function as 22nd December 2015

Successful Drone Ship Landing with Payload between 4000 and 6000

- WHERE clause has been used to limit landing outcomes only to success ground pad and defined payload mass of 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

```
[20]: %sql select BOOSTER_VERSION from SPACEXTBL where [Landing _Outcome] = 'Success (ground pad)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
 * sqlite:///my_data1.db
Done.
```

[20]:

Booster_Version
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1

Total Number of Successful and Failure Mission Outcomes

- Find the total number of outcomes for success and failure and Final result displaying Successful Mission and failure mission count.

Task 7

List the total number of successful and failure mission outcomes

```
[73]: #%sql select mission_outcome, count(*) from SPACEXTBL group by mission_outcome
      %sql SELECT sum(case when MISSION_OUTCOME LIKE '%Success%' then 1 else 0 end) AS "Successful Mission", \
          sum(case when MISSION_OUTCOME LIKE '%Failure%' then 1 else 0 end) AS "Failure Mission" \
          FROM SPACEXTBL;
```

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

[73]:

Successful Mission	Failure Mission
100	1

Boosters Carried Maximum Payload

- Listed the names of the boosters that carried maximum payload and use of subquery has been shown.

Task 8

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

[83]:
```
%sql select distinct Booster_Version From SPACEXTBL where Payload_Mass__Kg_ >= (select Max(Payload_Mass__Kg_) from SPACEXTBL);
#%sql SELECT DISTINCT BOOSTER_VERSION AS "Booster_Versions which carried the Maximum Payload Mass" FROM SPACEX \
#WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEX);
```

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

[83]:

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

Task 9

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

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

[8]:
```
%sql SELECT substr(Date, 4, 2) as 'month' , BOOSTER_VERSION, LAUNCH_SITE, [Landing _Outcome] FROM SPACEXTBL WHERE substr(Date, 7,4)='2015' and [Landing _Outcome] = 'Failure (drone ship)';
```

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

[8]:

month	Booster_Version	Launch_Site	Landing _Outcome
01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

2015 Launch Records

- Substring selects out of date and WHERE clause selects Landing outcome for failure drone ship. ‘01’ and ‘04’ are the month names which we are looking for.

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

Task 10

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

```
[47]: %sql SELECT [Landing _Outcome], COUNT([Landing _Outcome]) as "Total_Count" from SPACEXTBL WHERE [Landing _Outcome] LIKE 'Success%' and Date between '04-06-2010' and '20-03-2017'\
group by [Landing _Outcome]order by "Total_Count" desc;
```

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

[47]:

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

Rank the count of landing outcomes such as Failure(drone ship) or Success between ‘2010-06-04’ and ‘2017-03-20’ in descending order and use of group by clause was used to group the landing outcome in descending order.

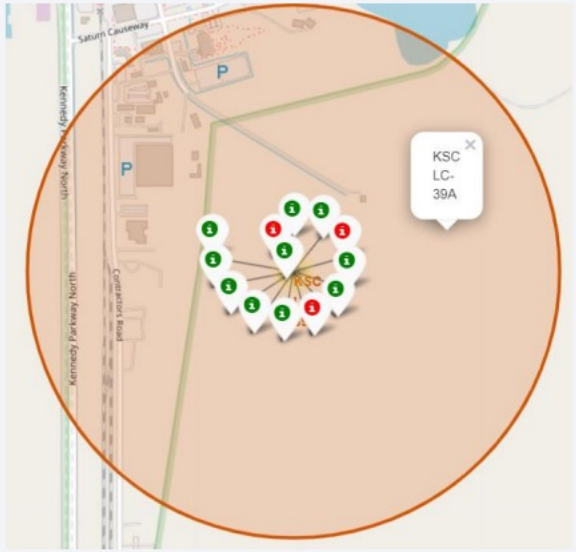

Section 3
Launch Sites
Proximities Analysis

Location of Launch sites

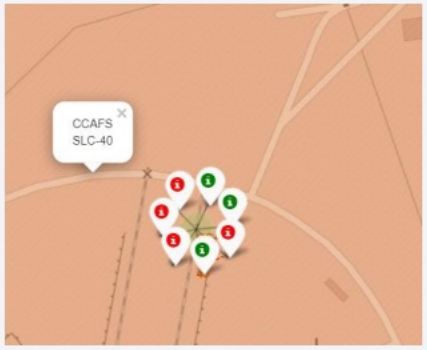
SpaceX Launch Sites on the Global Map

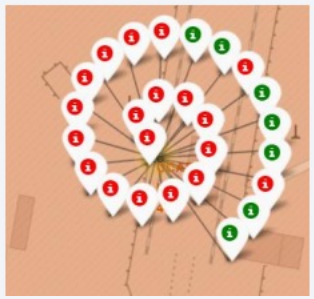
- The SpaceX Launch Sites are on the coasts Florida and California in America
- Only the launch site VAFB SLC – 4E is in California; the rest are in Florida

Success and Failure for each Launch site

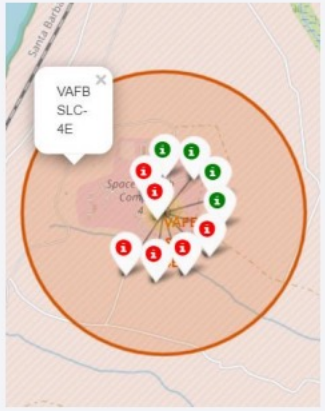


Florida Launch Sites



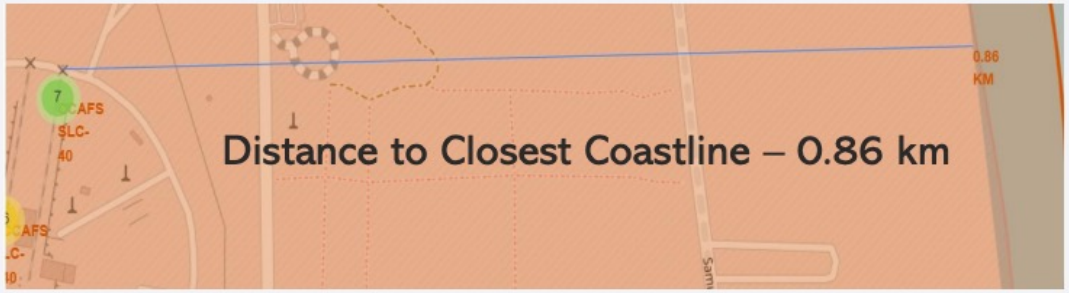


CCAFS LC-40

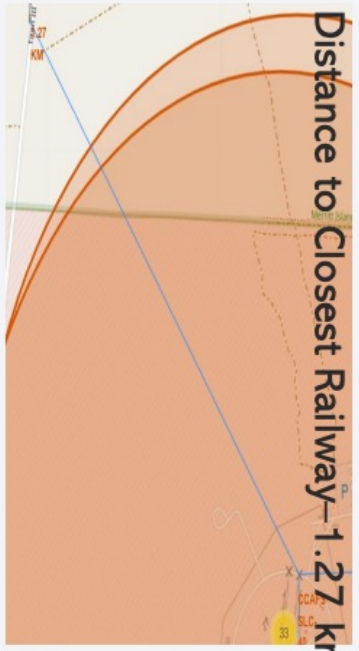


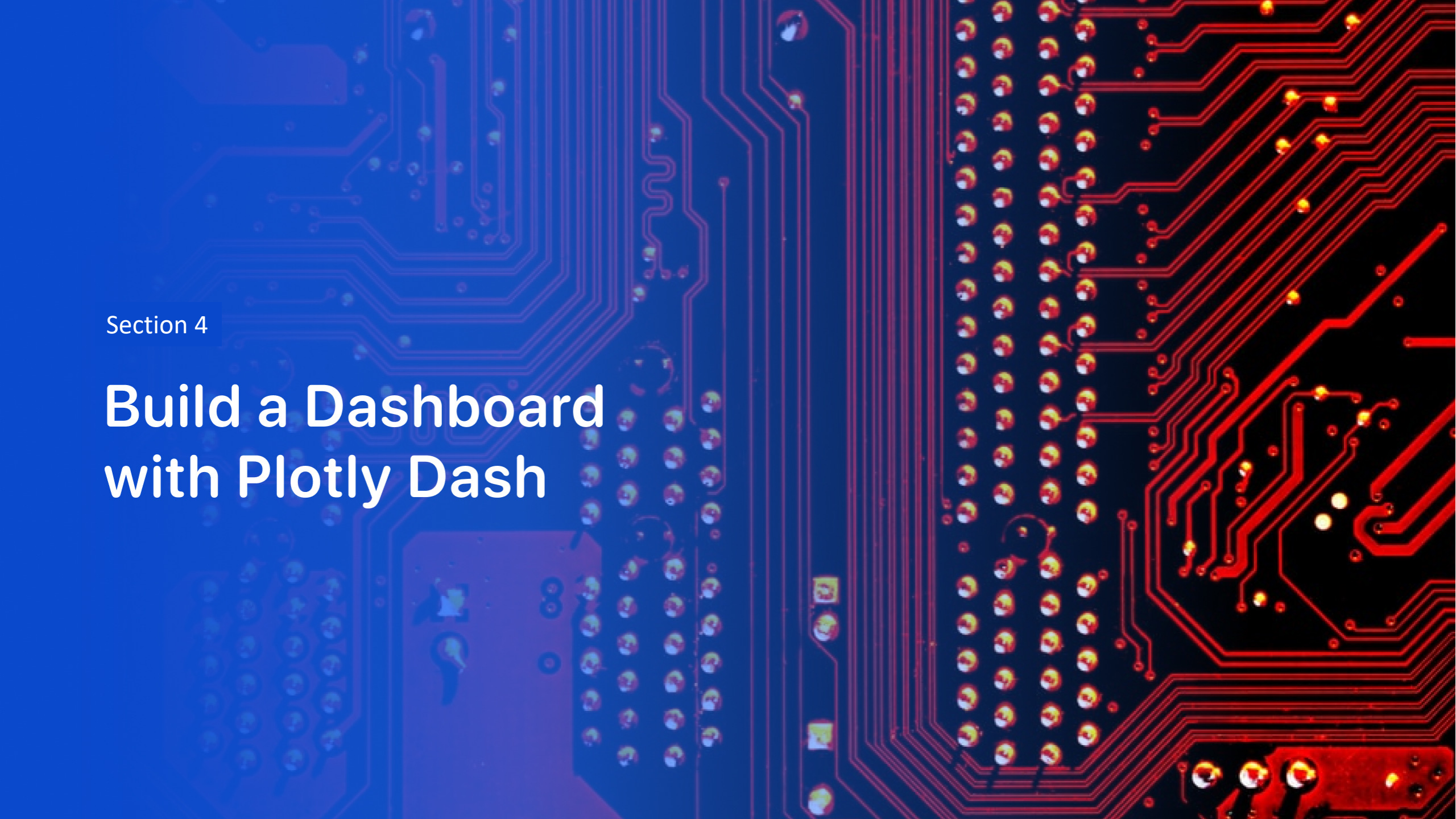
California Launch Site

Distance of launch sites from Proximities.

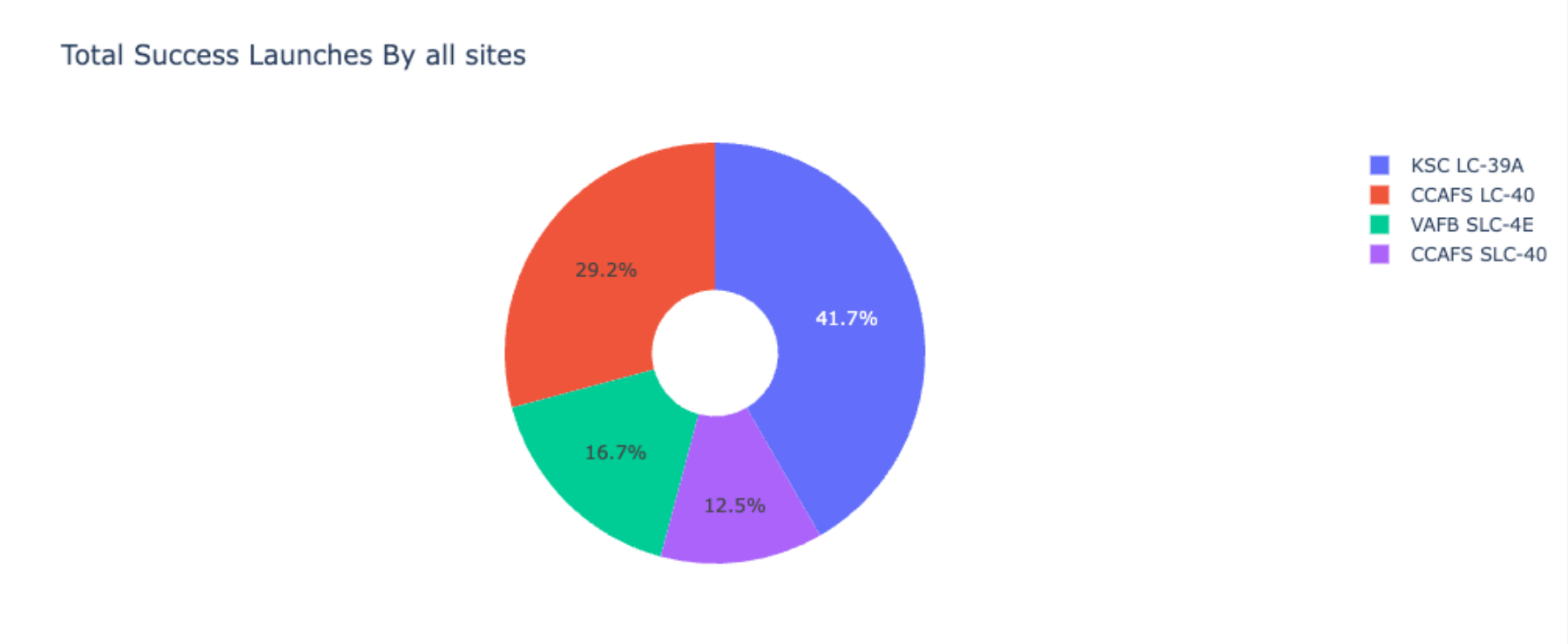


Are launch sites in close proximity to railways? – Yes (1.27 km)
Are launch sites in close proximity to highways? Yes (0.58 km)



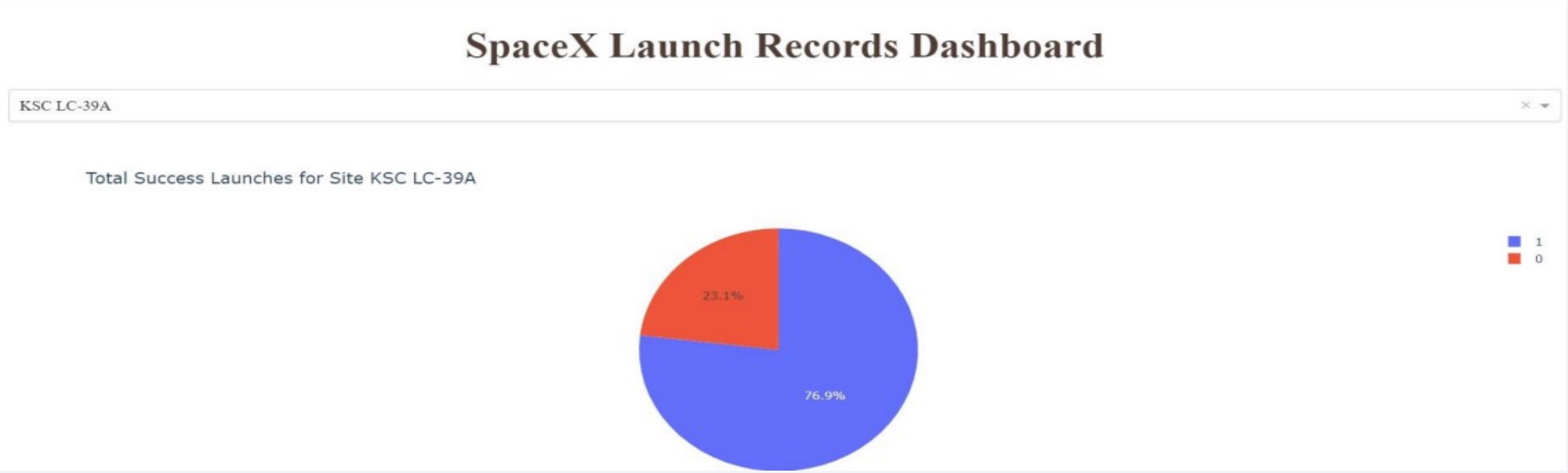

Section 4
Build a Dashboard with Plotly Dash

Percentage of each sites based on success.



We can Note that KSC LC -39A has most successful launches among other launch sites.

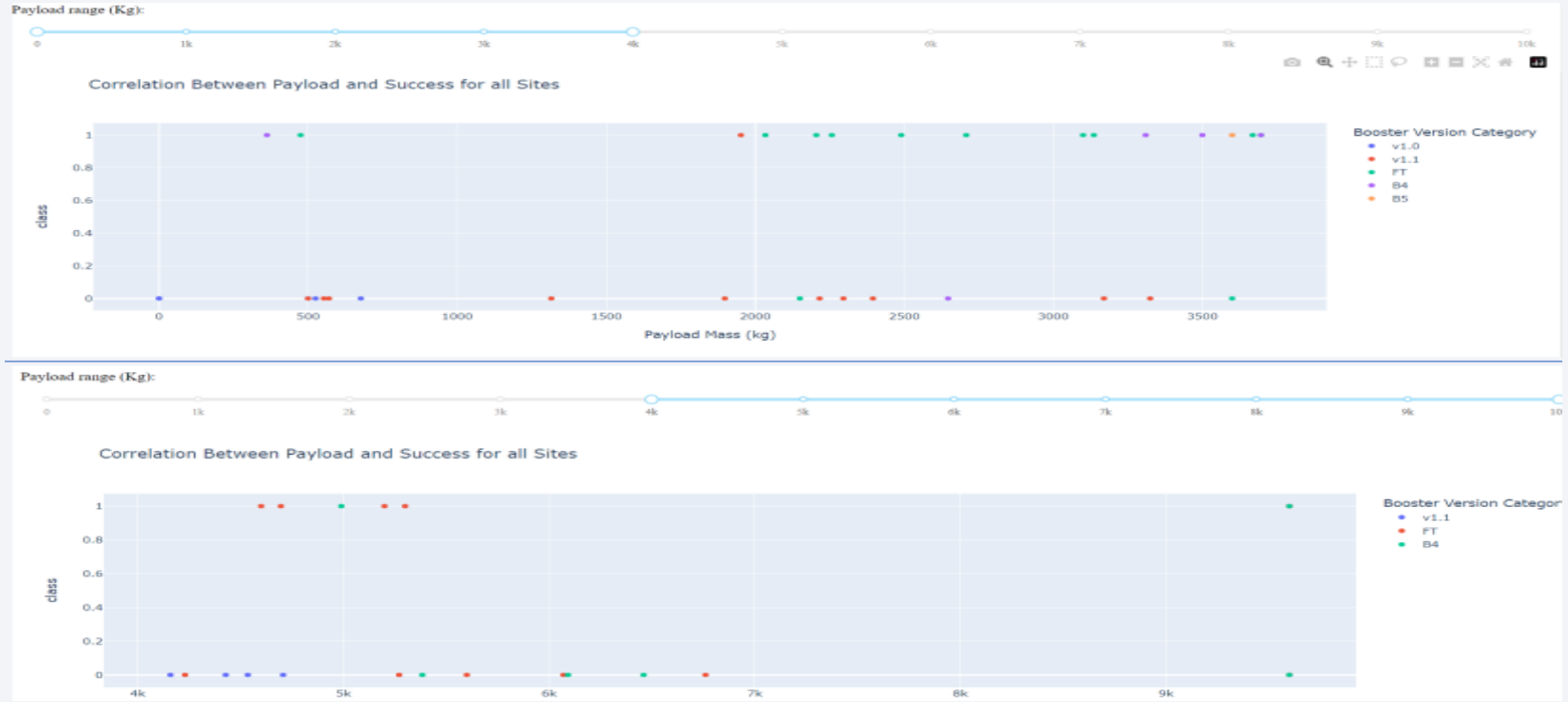
Success ration using Pie chart of KSC LC-39A



KSC LC-39A has attained a success rate of 76.9% and 23.1% failure rate.

Scatter plot of Payload vs Launch outcome for all

- Success ratio of 0-4000kg and >4000 kg(till 10000kg) w.r.t to payload.



Section 5

Predictive Analysis (Classification)

Classification Accuracy

TASK 12

Find the method performs best:

In [37]:
```
algorithms = {'KNN':knn_cv.best_score_,'Decision Tree':tree_cv.best_score_,'Logistic Regression':logreg_cv.best_score_,'SVM':svm_cv.best_s
best_algorithm = max(algorithms, key= lambda x: algorithms[x])

print('The method which performs best is \"',best_algorithm,'\" with a score of',algorithms[best_algorithm])
```

```
The method which performs best is " Decision Tree " with a score of 0.8892857142857142
```

In [38]:
```
algo_df = pd.DataFrame.from_dict(algorithms, orient='index', columns=['Accuracy'])
```

In [39]:
```
algo_df.head()
```

Out[39]:

	Accuracy
KNN	0.833929
Decision Tree	0.889286
Logistic Regression	0.821429
SVM	0.848214

From the models used ‘Decision Tree Classifier’ has the highest classification accuracy.

Confusion Matrix

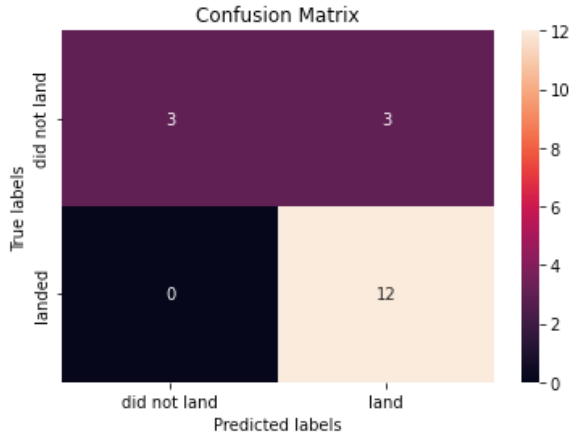
TASK 9

Calculate the accuracy of tree_cv on the test data using the method score:

In [30]:
```
print("Accuracy for decision tree classifier on the test data using the method score:", tree_cv.score(X_test, Y_test))
```

```
Accuracy for decision tree classifier on the test data using the method score: 0.9444444444444444
```

In [31]:
```
yhat = svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- From the above we can conclude that:
- Launch success rate has been increased significantly from 2013 till 2020.
- KSC-LC-39A have the most successful launches compared to rest sites, i.e. 77%.
- Orbits ES-L1, SSO have the most success rate.
- Decision Tree classifier is the Top accurate Machine Learning algorithm for this problem to prediction landing outcomes.
- Low weighted payloads under 4000kg performed better compared to heavy weighted payloads.

Thank you!