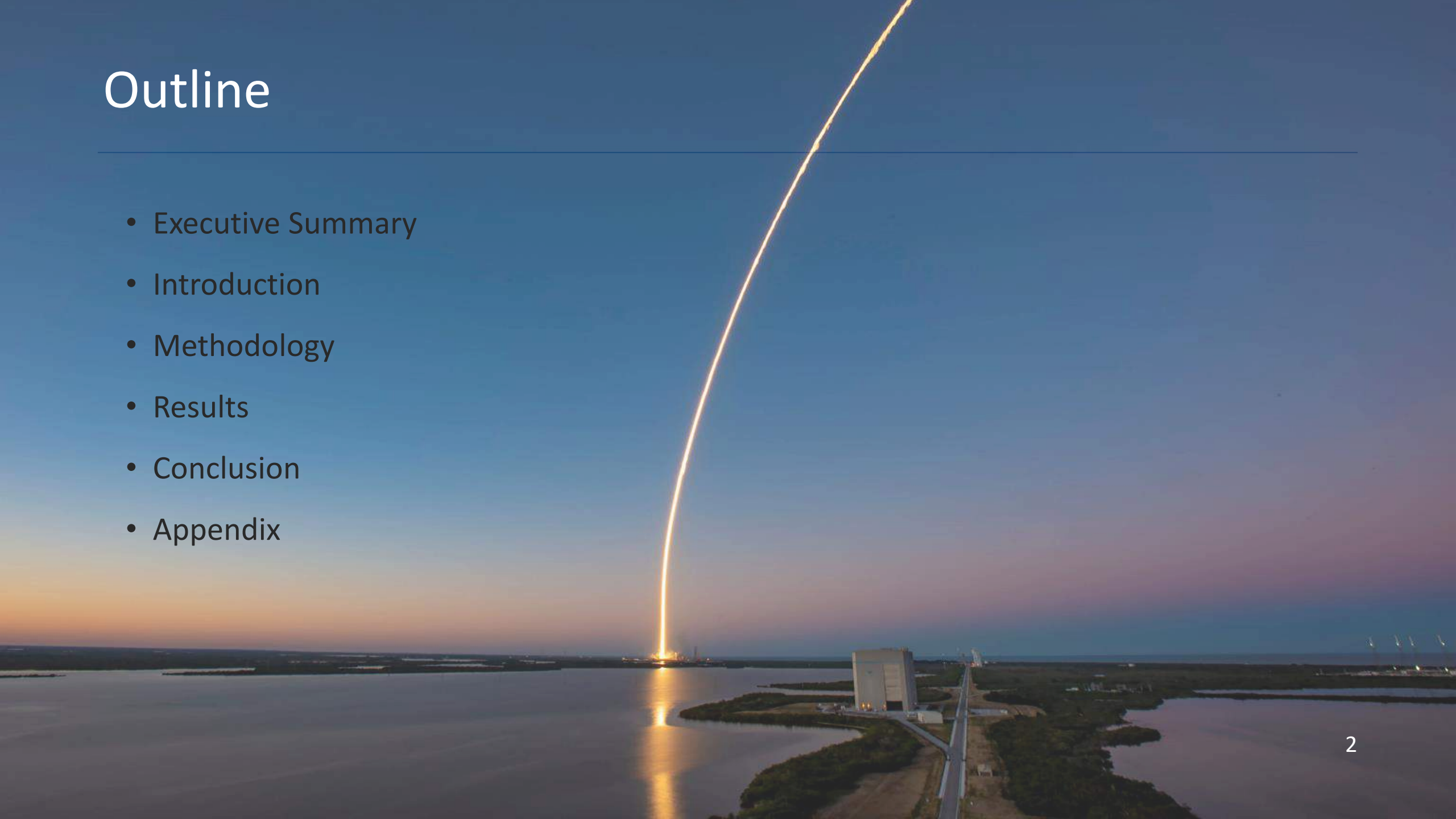
IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Udit Jadli
25/03/2023

Outline

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



Executive Summary

- Summary of methodologies

  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction

- Summary of all results

  * Exploratory Data Analysis result
  * Interactive analytics in screenshots
  * Predictive Analytics result

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. 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. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Problems you want to find answers

  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - What operating conditions needs to be in place to ensure a successful landing program.

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

❖The data was collected using various methods:

- Data collection was done using get request to the SpaceX API.
- Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
- We then cleaned the data, checked for missing values and fill in missing values where necessary.
- In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
- The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.

- The link to the notebook is https://github.com/jadliudit/testrepo/blob/master/jupyter-labs-spacex-data-collection-api.ipynb

Now let's start requesting rocket launch data from SpaceX API with the following URL:

In [49]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

In [50]:
```
response = requests.get(spacex_url)
```

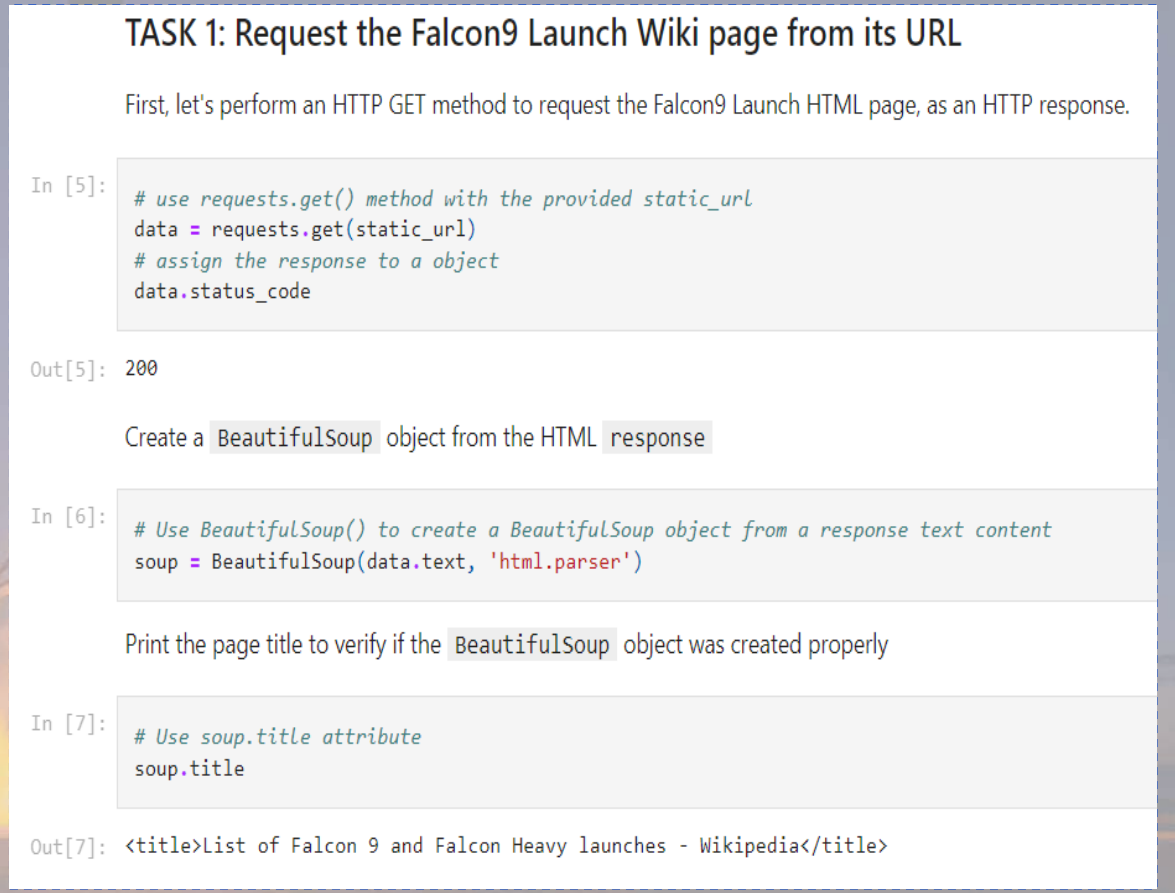
Check the content of the response

In [51]:
```
print(response.content)
```


```
b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https://images2.imgbox.com/94/f2/NN6Ph
45r_o.png","large":"https://images2.imgbox.com/5b/02/QcxHUb5V_o.png"},"reddit":{"campaign":null,"launch":null,"media":null,"recovery":null},"flickr":
{"small":[],"original":[]},"presskit":null,"webcast":"https://www.youtube.com/watch?v=0a_00nJ_Y88","youtube_id":"0a_00nJ_Y88","article":"https://www.sp
ace.com/2196-spacex-inaugural-falcon-1-rocket-lost-launch.html","wikipedia":"https://en.wikipedia.org/wiki/DemoSat"},"static_fire_date_utc":"2006-03-17
T00:00:00.000Z","static_fire_date_unix":1142553600,"net":false,"window":0,"rocket":"5e9d0d95eda69955f709d1eb","success":false,"failures":[{"time":33,"a
ltitude":null,"reason":"merlin engine failure"}],"details":"Engine failure at 33 seconds and loss of vehicle","crew":[],"ships":[],"capsules":[],"paylo
ads":["5eb0e4b5b6c3bb0006eeb1e1"],"launchpad":"5e9e4502f5090995de566f86","flight_number":1,"name":"FalconSat","date_utc":"2006-03-24T22:30:00.000Z","da
te_unix":1143239400,"date_local":"2006-03-25T10:30:00+12:00","date_precision":"hour","upcoming":false,"cores":[{"core":"5e9e289df35918033d3b2623","flig
ht":1,"gridfins":false,"legs":false,"reused":false,"landing_attempt":false,"landing_success":null,"landing_type":null,"landpad":null}],"auto_update":tr
ue,"tbd":false,"launch_library_id":null,"id":"5eb87cd9ffd86e000604b32a"},{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ship
s":[]},"links":{"patch":{"small":"https://images2.imgbox.com/f9/4a/ZboXReNb_o.png","large":"https://images2.imgbox.com/80/a2/bkWotCIS_o.png"},"reddit":
{"campaign":null,"launch":null,"media":null,"recovery":null},"flickr":{"small":[],"original":[]},"presskit":null,"webcast":"https://www.youtube.com/wat
ch?v=Lk4zQ2wP-Nc","youtube_id":"Lk4zQ2wP-Nc","article":"https://www.space.com/3590-spacex-falcon-1-rocket-fails-reach-orbit.html","wikipedia":"https://
en.wikipedia.org/wiki/DemoSat"},"static_fire_date_utc":null,"static_fire_date_unix":null,"net":false,"window":0,"rocket":"5e9d0d95eda69955f709d1eb","su
```

Data Collection - Scraping

- Applied web scrapping to web scrap Falcon 9 launch records with BeautifulSoup
- Parsed the table and converted it into a pandas dataframe.
- The link to my notebook is https://github.com/jadliudit/testrepo/blob/master/jupyter-labs-webscraping.ipynb



TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

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

Out[5]: 200

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

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

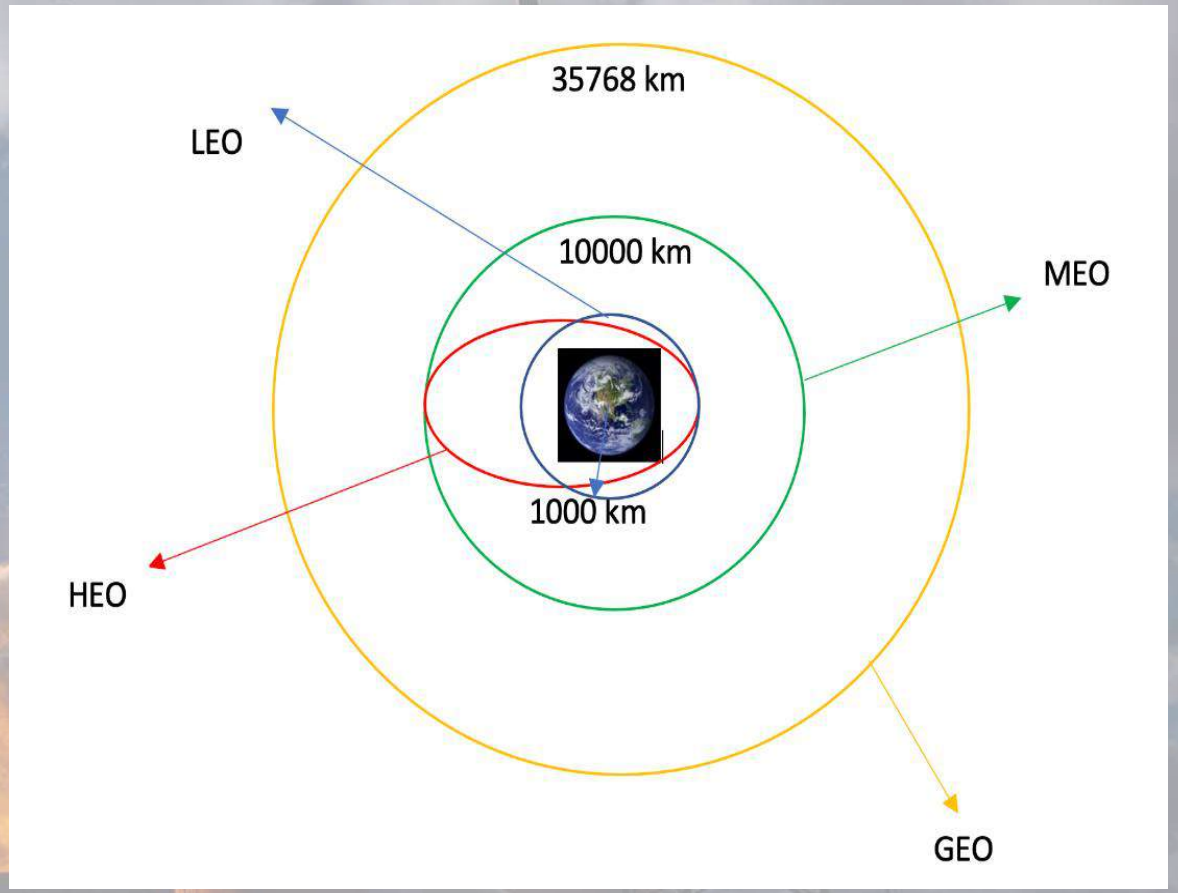
Print the page title to verify if the `BeautifulSoup` object was created properly

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

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

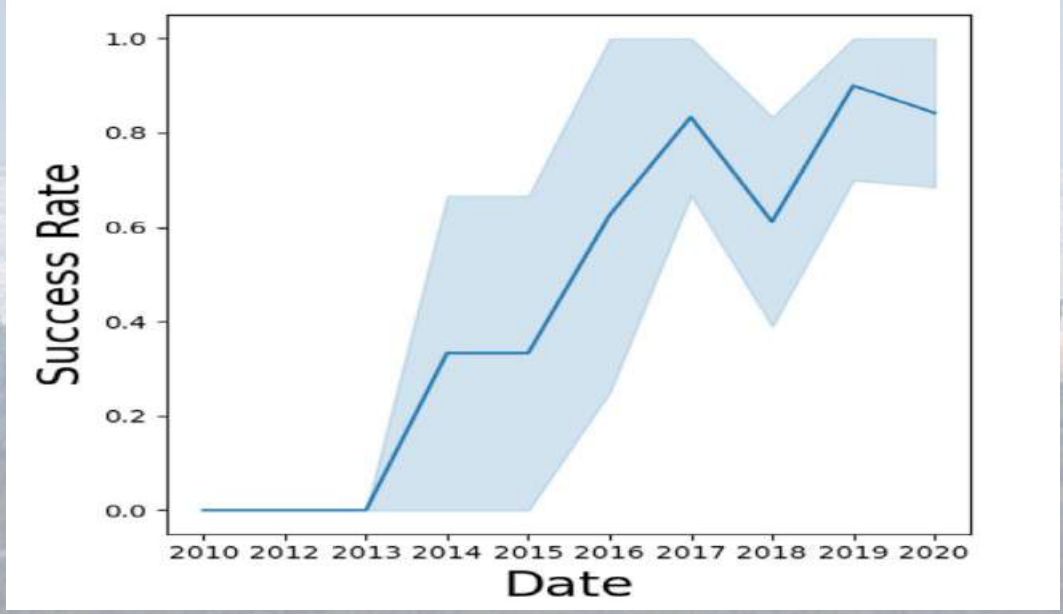
Data Wrangling

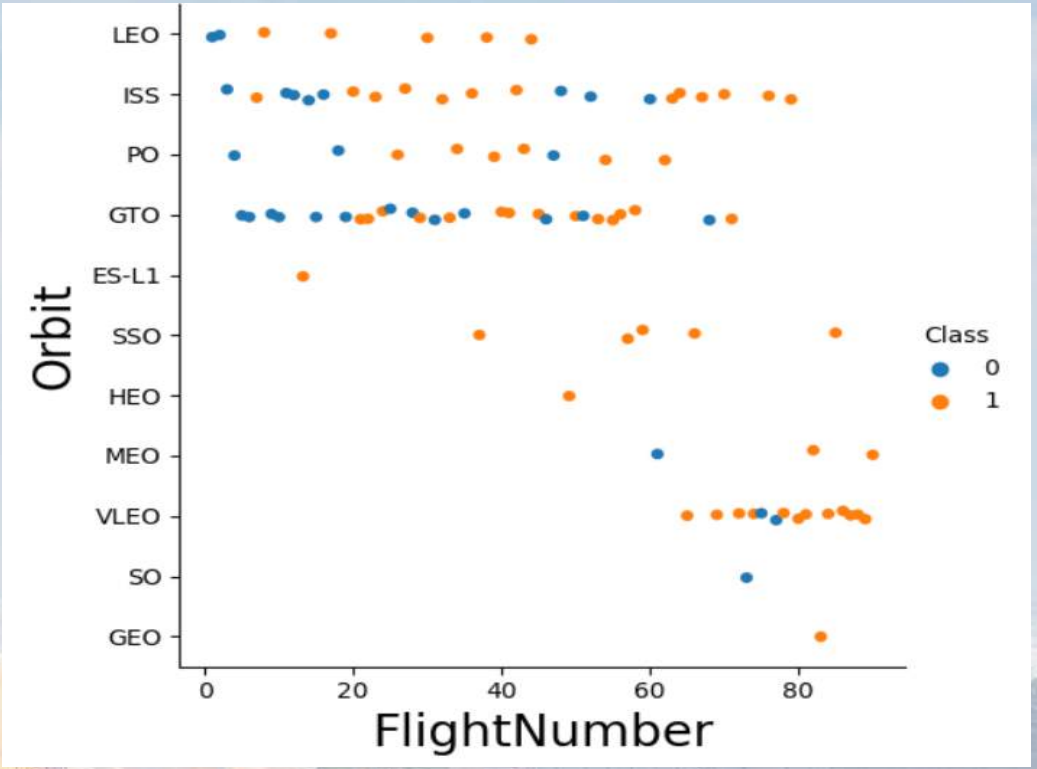
- We performed exploratory analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits.
- We created landing outcome label from outcome column and exported the results to csv.
- The link to the notebook is https://github.com/jadliudit/testrepo/blob/master/jupyter-labs-data%20wrangling.ipynb



EDA with Data Visualization

- Explored the data by visualizing the relationship between flight number and launch site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.





- The link to my notebook is https://github.com/jadliudit/testrepo/blob/master/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- We loaded the SpaceX dataset into a PostgreSQL database without leaving the jupyter notebook.
- We loaded the SpaceX dataset into a PostgreSQL database without leaving the jupyter notebook.
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The successful landing outcomes in ground pad, their booster version and launch site names.
- The link to my notebook is https://github.com/jadliudit/testrepo/blob/master/jupyter-labs-eda-sql-edx_sqllite.ipynb

Build an Interactive Map with Folium

- Marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- Assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, identified which launch sites have relatively high success rate.
- Calculated the distances between a launch site to its proximities. Answered some question for instance:
  - Are launch sites near railways, highways and coastlines.
  - Do launch sites keep certain distance away from cities.
- The link to my notebook is https://github.com/jadliudit/testrepo/blob/master/jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash.
- We plotted pie charts showing the total launches by a certain sites.
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- The link to the notebook is: https://github.com/jadliudit/testrepo/blob/master/Dash Interactive.py

Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- The link to the notebook is https://github.com/jadliudit/testrepo/blob/master/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

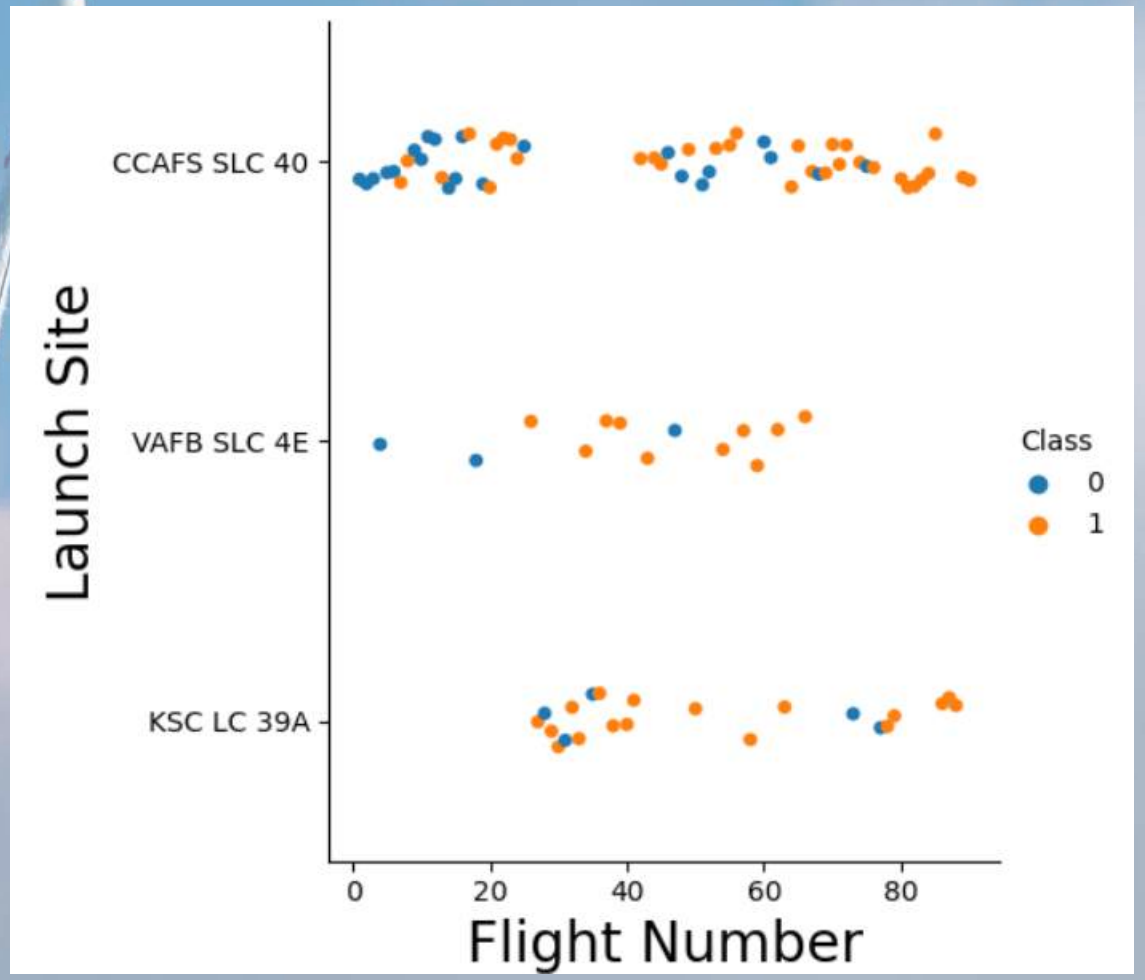
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

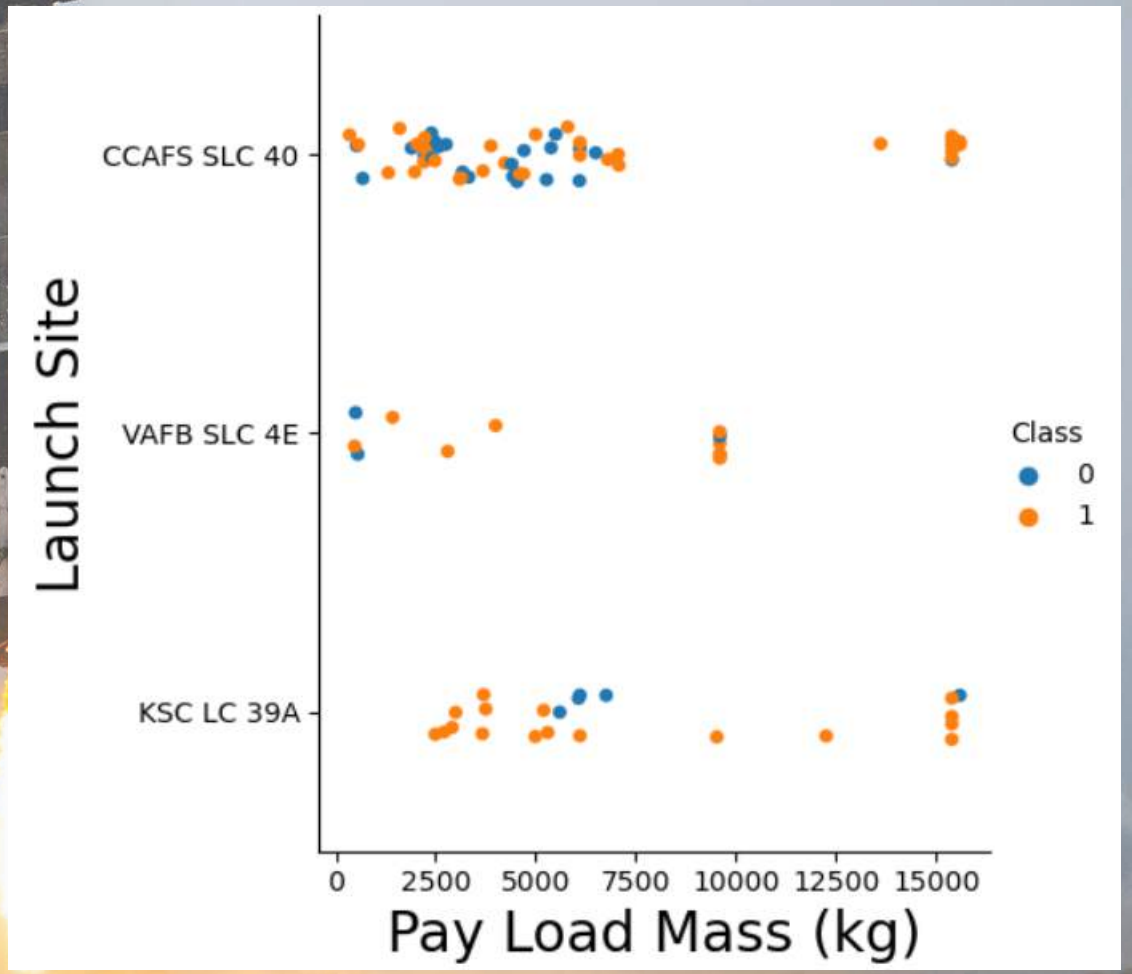
Flight Number vs. Launch Site

- According to the plot above, the most successful launch site is CCAF5 SLC 40.
- In second place VAFB SLC 4E and third place KSC LC 39A.
- Also, the overall success rate is improving over time.



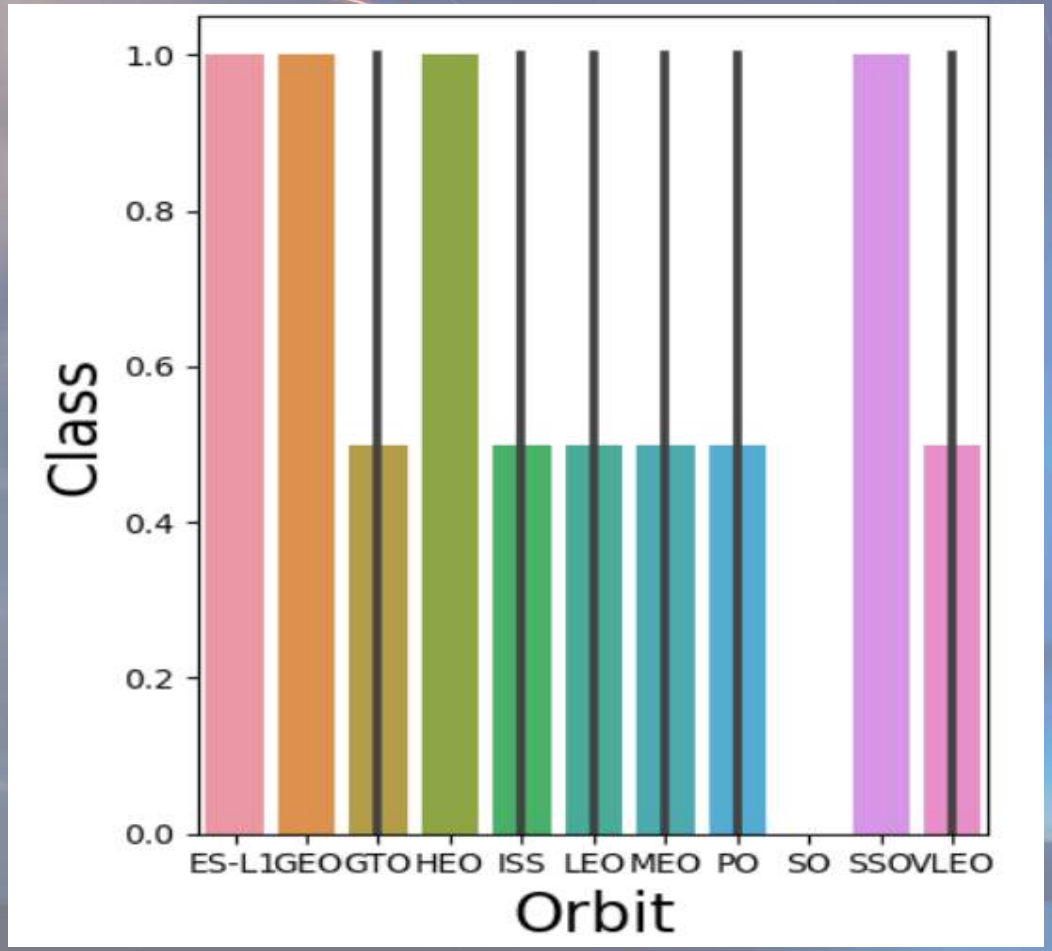
Payload vs. Launch Site

- Higher payloads (over 12,000kg) have a better success rate at CCAFS SLC 40 and KSC LC 39A launch sites.



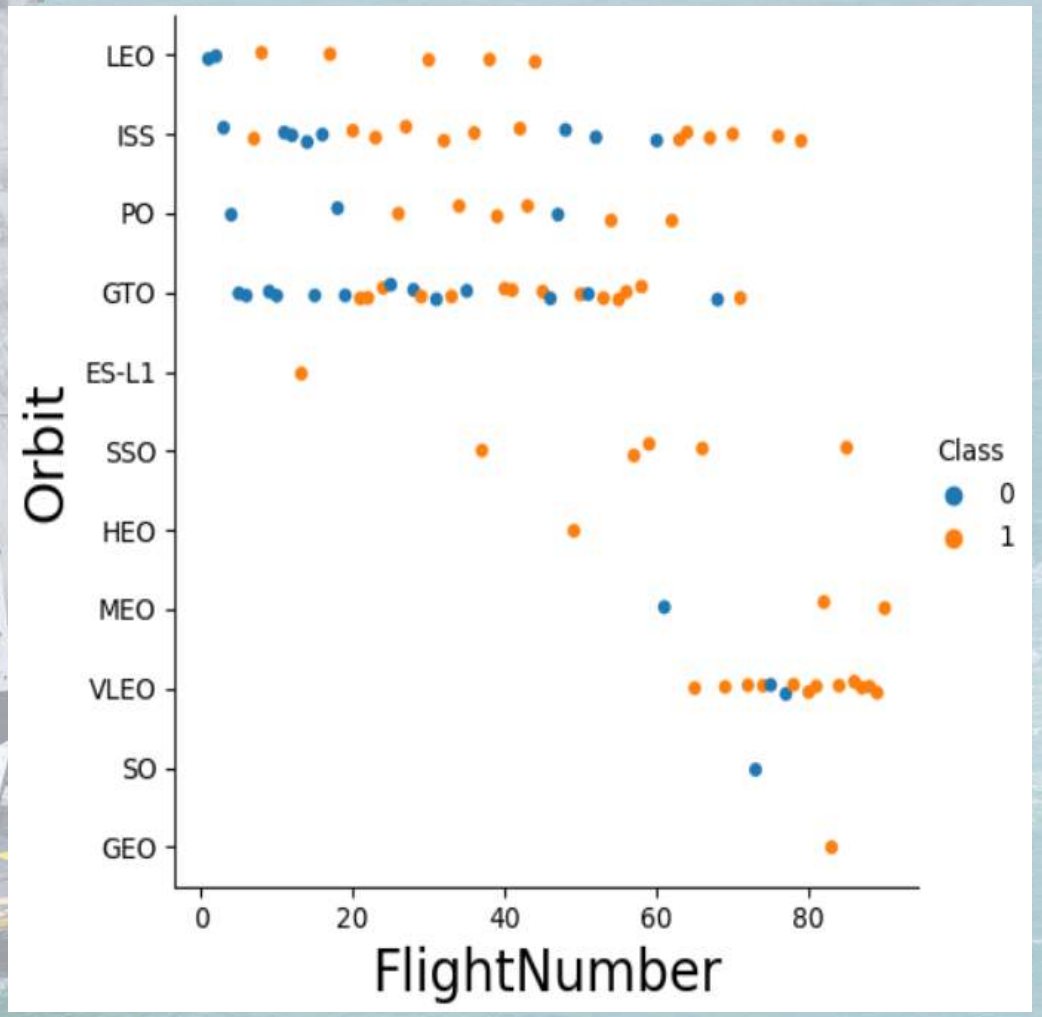
Success Rate vs. Orbit Type

- The following orbits have the highest success rates:
  - ES-L1
  - GEO
  - HEO
  - SSO



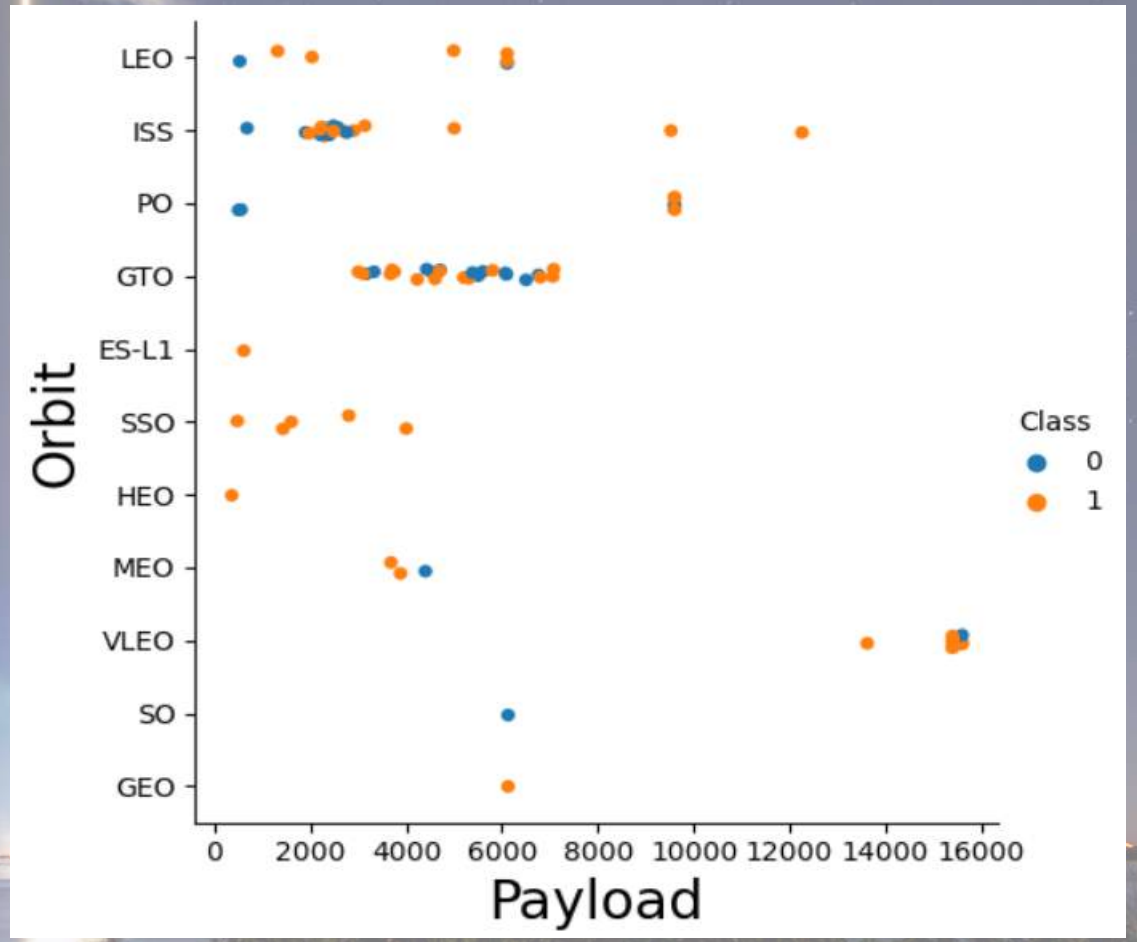
Flight Number vs. Orbit Type

- We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



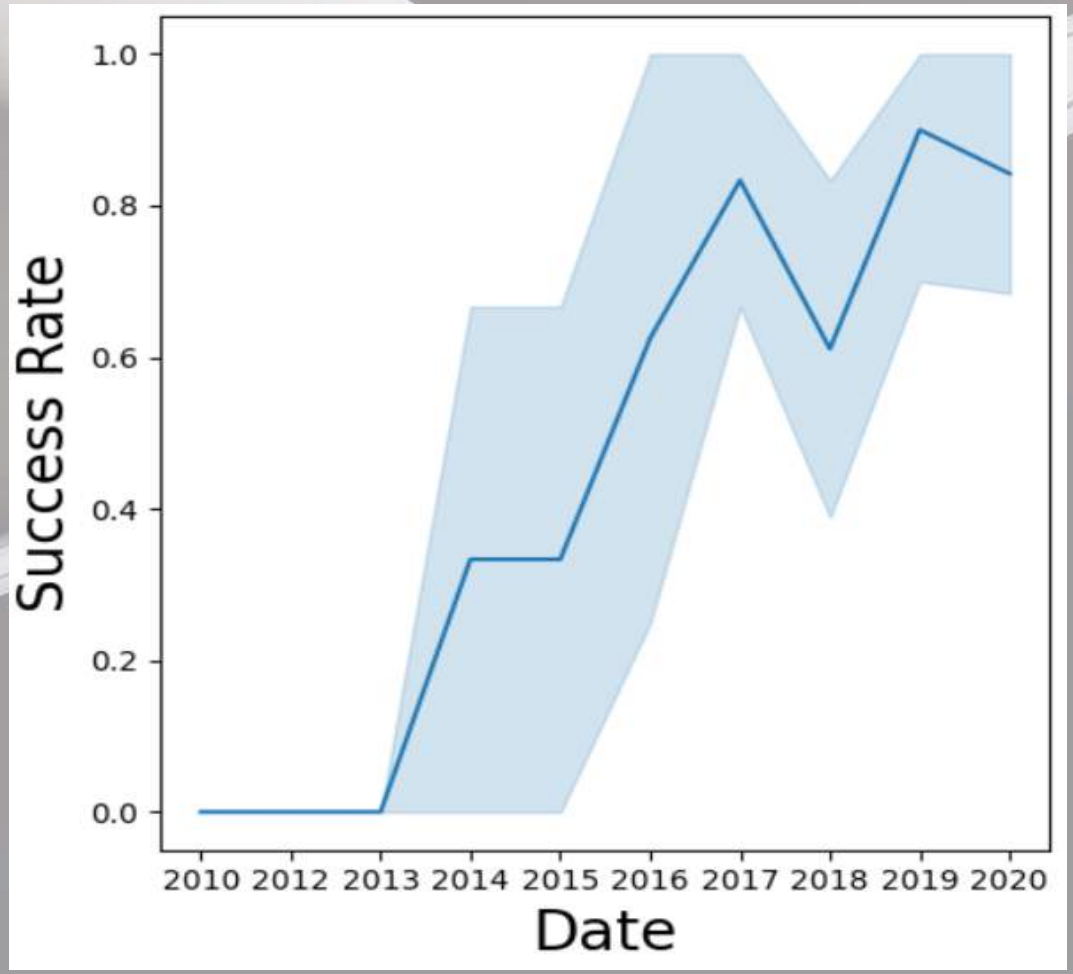
Payload vs. Orbit Type

- We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



Launch Success Yearly Trend

- From the plot, we can observe that success rate since 2013 kept on increasing till 2020.



All Launch Site Names

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

Task 1

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

In [7]:
```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

Out[7]:

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

Launch Site Names Begin with 'KSC'

- Used the query above to display 2 records where launch sites begin with `KSC`.

Task 2

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

In [11]:
```
%sql select * from SPACEXTBL where launch_site like 'KSC%' limit 5
```

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

Out[11]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
19-02-2017	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
16-03-2017	06:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
30-03-2017	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
01-05-2017	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
15-05-2017	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

Total Payload Mass

- We calculated the total payload carried by boosters from NASA with the below query.

Task 3

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

In [12]:
```
%sql select sum(payload_mass__kg_) as sum from SPACEXTBL where customer like 'NASA (CRS)'
```

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

Out[12]:

sum
45596

Average Payload Mass by F9 v1.1

- We calculated the average payload mass carried by booster version F9 v1.1 using the below query.

Task 4

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

In [13]:
```
%sql select avg(payload_mass__kg_) as average from SPACEXTBL where booster_version like 'F9 v1.1%'
```

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

Out[13]:

average
2534.6666666666665

First Successful Ground Landing Date

- The date of the first successful landing outcome on ground pad can be fetched using the below query.

Task 5

List the date where the succesful landing outcome in drone ship was acheived.

Hint:Use min function

In [14]:
```
%sql select min(date) as Date from SPACEXTBL where mission_outcome like 'Success'
```

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

Out[14]:

Date
01-03-2013

Successful Drone Ship Landing with Payload between 4000 and 6000

- We used the WHERE clause to filter for boosters which have successfully landed on drone ship and applied the AND condition to determine successful landing with payload mass greater than 4000 but less than 6000.

Task 6

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

In [25]:
```
%sql SELECT booster_version FROM spacextbl \
where "landing _outcome" = 'Success (ground pad)' \
AND 4000 < payload_mass__kg_ < 6000;
```

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

Out[25]:

Booster_Version
F9 FT B1019
F9 FT B1025.1
F9 FT B1031.1
F9 FT B1032.1
F9 FT B1035.1
F9 B4 B1039.1
F9 B4 B1040.1
F9 FT B1035.2
F9 B4 B1043.1

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failure mission outcomes can be fetched using the below query.

Task 7

List the total number of successful and failure mission outcomes

In [20]:
```
%sql SELECT mission_outcome, COUNT(*) FROM spacextbl GROUP BY mission_outcome
```

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

Out[20]:

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

Boosters Carrying Maximum Payload

- Determined the booster that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX()** function.

Task 8

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

In [21]:
```
%sql SELECT DISTINCT(booster_version) FROM spacextbl \
WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) from spacextbl)
```

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

Out[21]:

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

2017 Launch Records

- We used a combinations of the WHERE clause, LIKE, AND, and BETWEEN conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2017.

Task 9

List the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017

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)='2017' for year.

[24]:
```
%%sql select

case cast (SUBSTR(Date,4,2) as integer)

when 1 then 'Jan'

when 2 then 'Feb'

when 3 then 'March'

when 4 then 'April'

when 5 then 'May'

when 6 then 'June'

when 7 then 'July'

when 8 then 'Aug'

when 9 then 'Sept'

when 10 then 'Oct'

when 11 then 'Nov'

else 'Dec' end AS "Month", Booster_Version, Launch_Site FROM SPACEXTBL

WHERE "Landing _Outcome" = 'Success (ground pad)' AND Substr(Date,7,4)='2017'
```

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

[24]:

Month	Booster_Version	Launch_Site
Feb	F9 FT B1031.1	KSC LC-39A
May	F9 FT B1032.1	KSC LC-39A
June	F9 FT B1035.1	KSC LC-39A
Aug	F9 B4 B1039.1	KSC LC-39A
Sept	F9 B4 B1040.1	KSC LC-39A
Dec	F9 FT B1035.2	CCAFS SLC-40

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

- Showing some error.

Task 10

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

In [29]:

```
%sql SELECT landing__outcome, COUNT(landing__outcome) AS COUNT FROM SPACEXDATASET WHERE (DATE between '2010-06-04' AND '2017-03-20') GROUP BY landing_
```

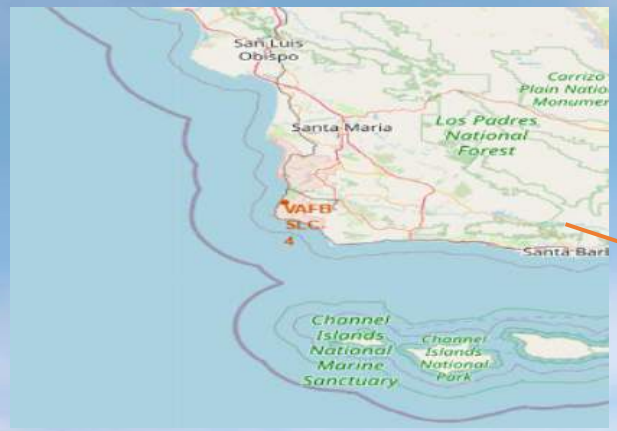
```
  File "/tmp/ipykernel_69/1758027512.py", line 2
    COUNT([Landing _Outcome]) as [Successful Landing Outcomes]
    ^
IndentationError: unexpected indent
```

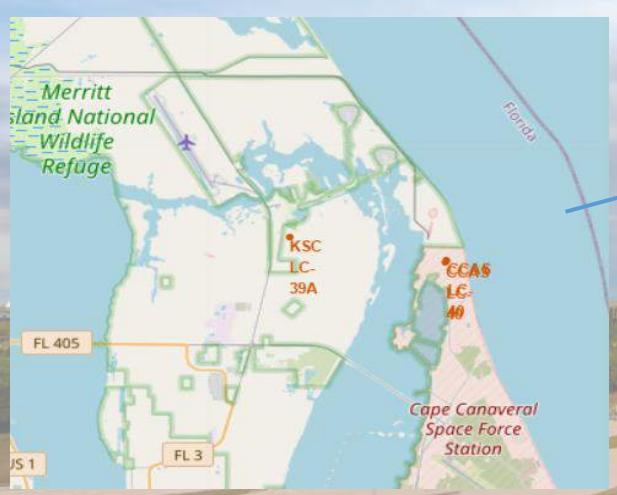
Section 3

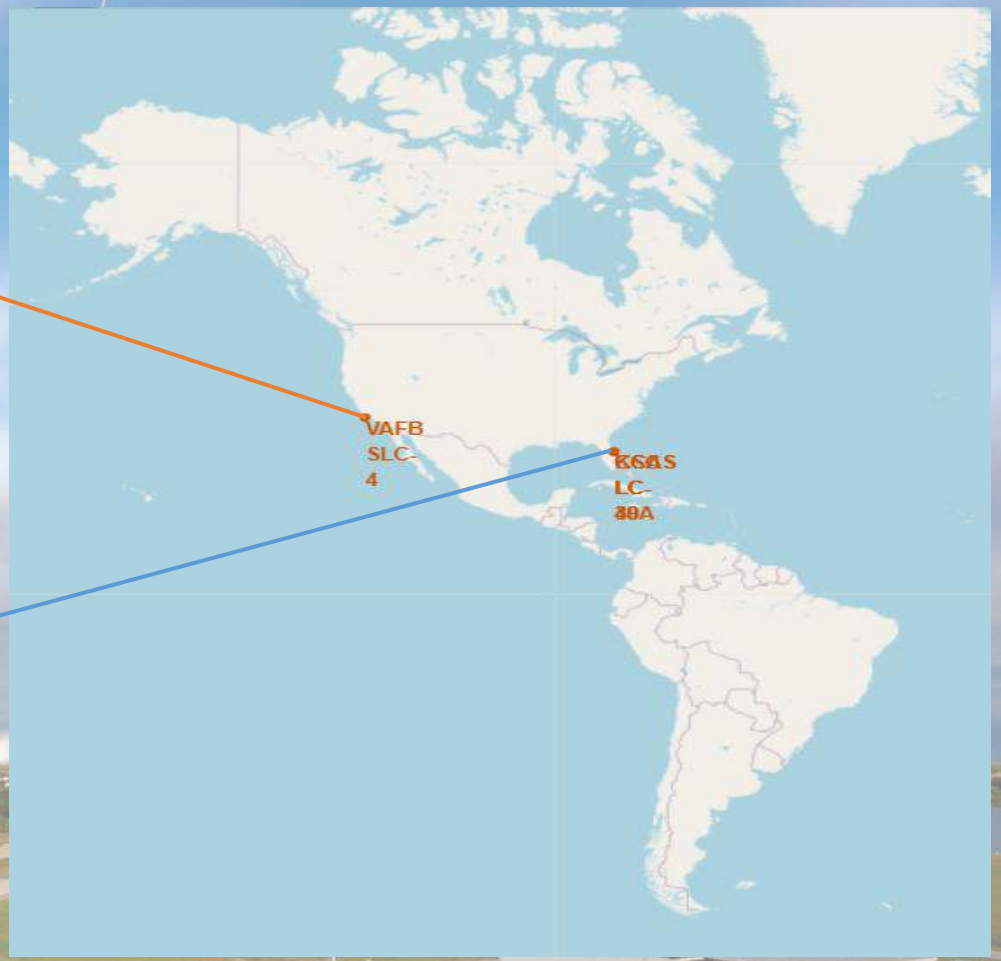
Launch Sites Proximities Analysis

All Launch Sites on Folium Map

- We can see that the SpaceX launch sites are near to the United States of America coasts i.e., Florida and California Regions.

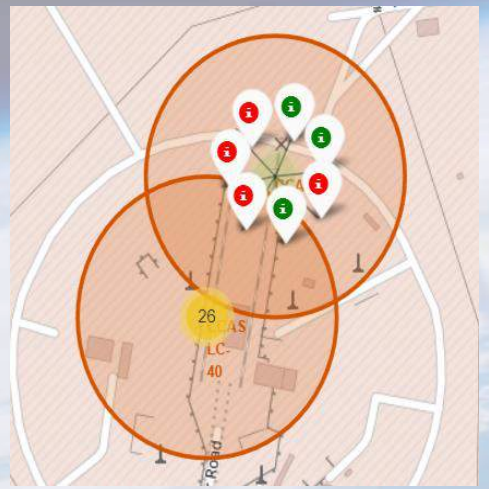




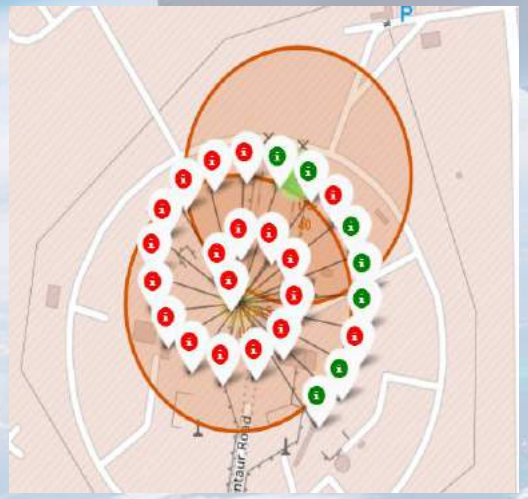


Color Labeled Launch Records

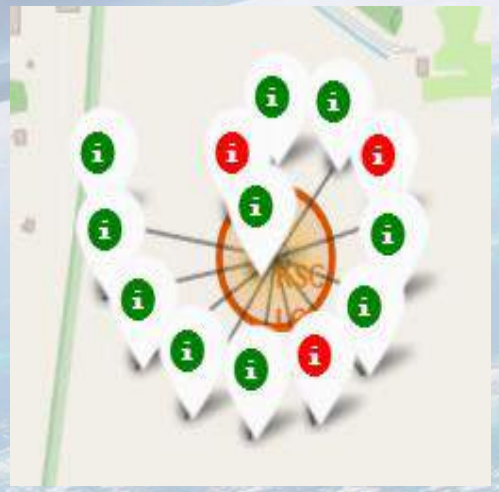
- Green Marker shows successful launches and Red Marker shows Failures.
- From these screenshots its easily understandable that KSC LC-39A has the maximum probability of success.



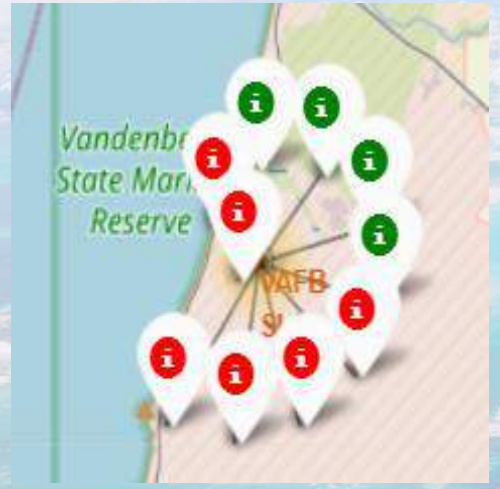
CCAFS SLC-40



CCAFSLC-40



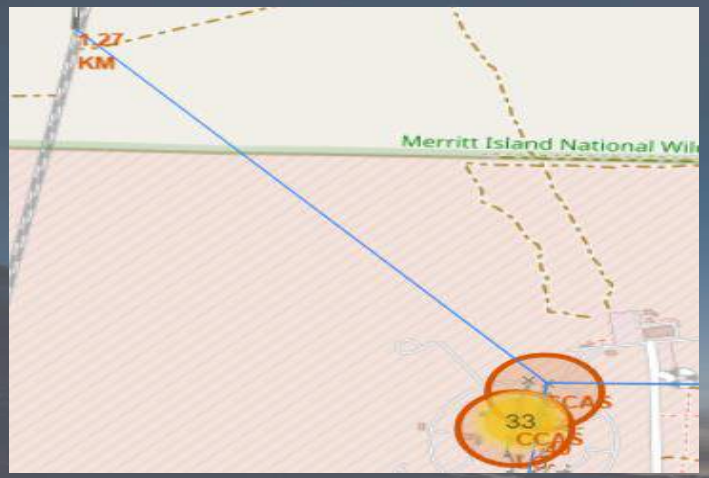
KSC LC-39A



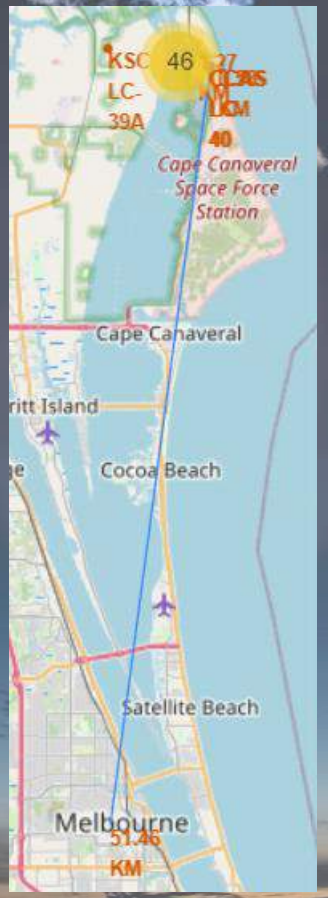
VAFB SLC-4E

Launch Site Distances from Railway, Highway, Coastline & City

Distance to highway



Distance to railroad

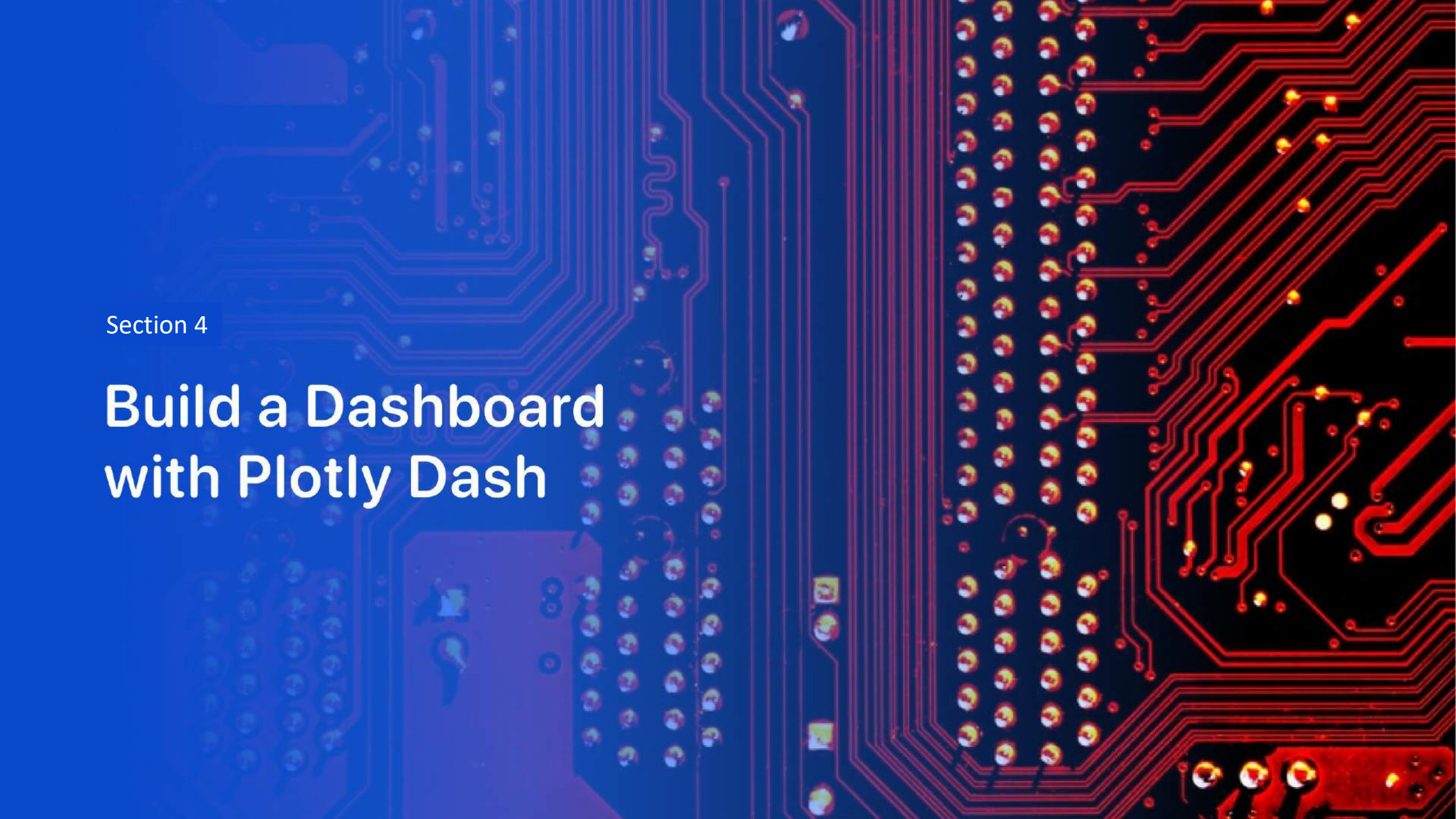


Distance to City

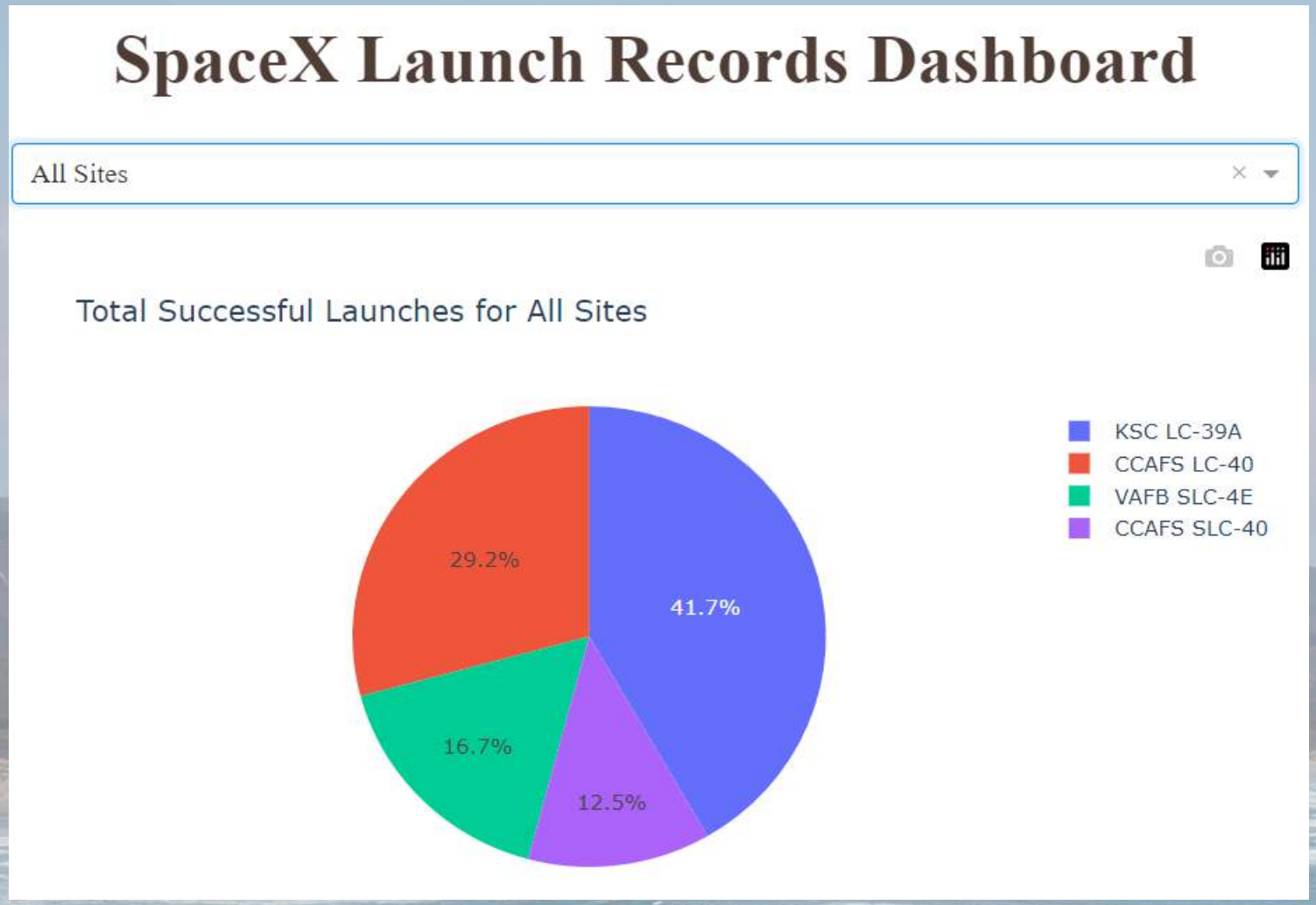
Distance to coastline

```
[15]: # find coordinate of the closet coastline
      # e.g.,: Lat: 28.56367  Lon: -80.57163
      # distance_coastline = calculate_distance(launch_site_lat, launch_site_lon, coastline_lat, coastline_lon)
      launch_site_lat = 28.56341
      launch_site_lon = -80.57678
      launch_site = [launch_site_lat, launch_site_lon]
      coastline_lat = 28.56335
      coastline_lon = -80.56755
      coordinates = [coastline_lat, coastline_lon]
      distance_coastline = calculate_distance(launch_site_lat, launch_site_lon, coastline_lat, coastline_lon)
      distance_coastline

[15]: 0.9017210342321572
```


Section 4
Build a Dashboard with Plotly Dash

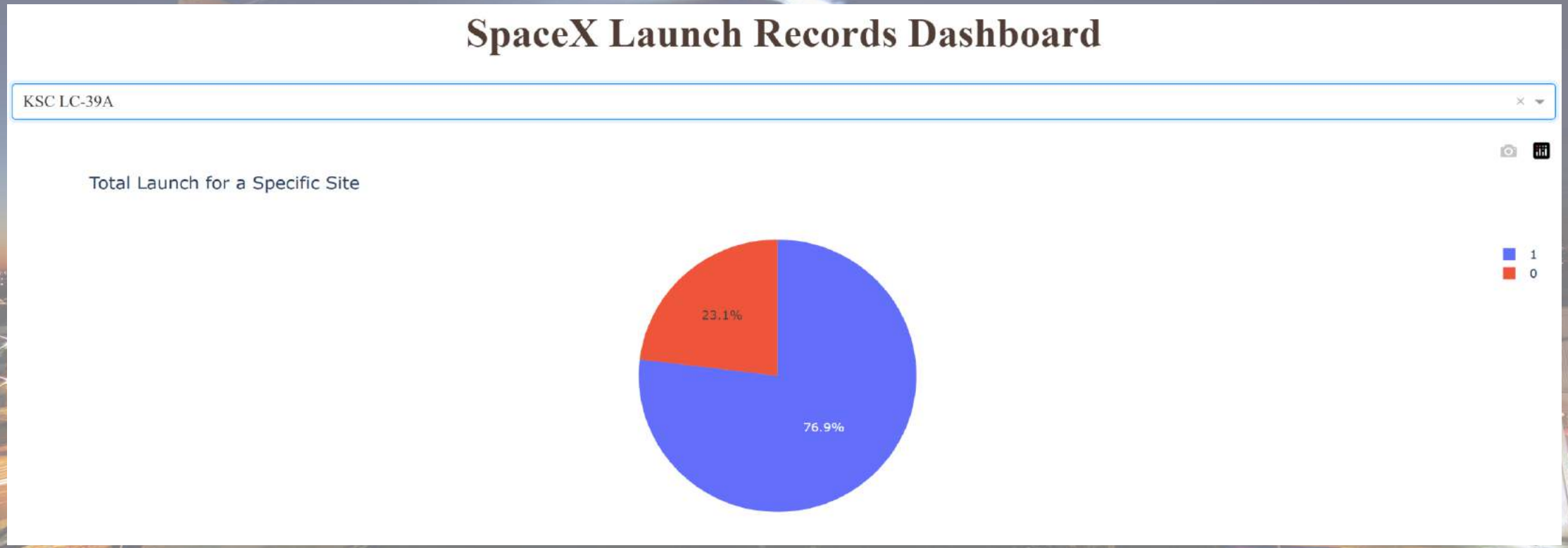
Successful Launches Across Launch Sites



- This is the distribution of successful landings across all launch sites. CCAFS and KSC have the same amount of successful landings. VAFB has the smallest percentage of successful landings. This may be due to smaller sample and increase in difficulty of launching in the west coast.

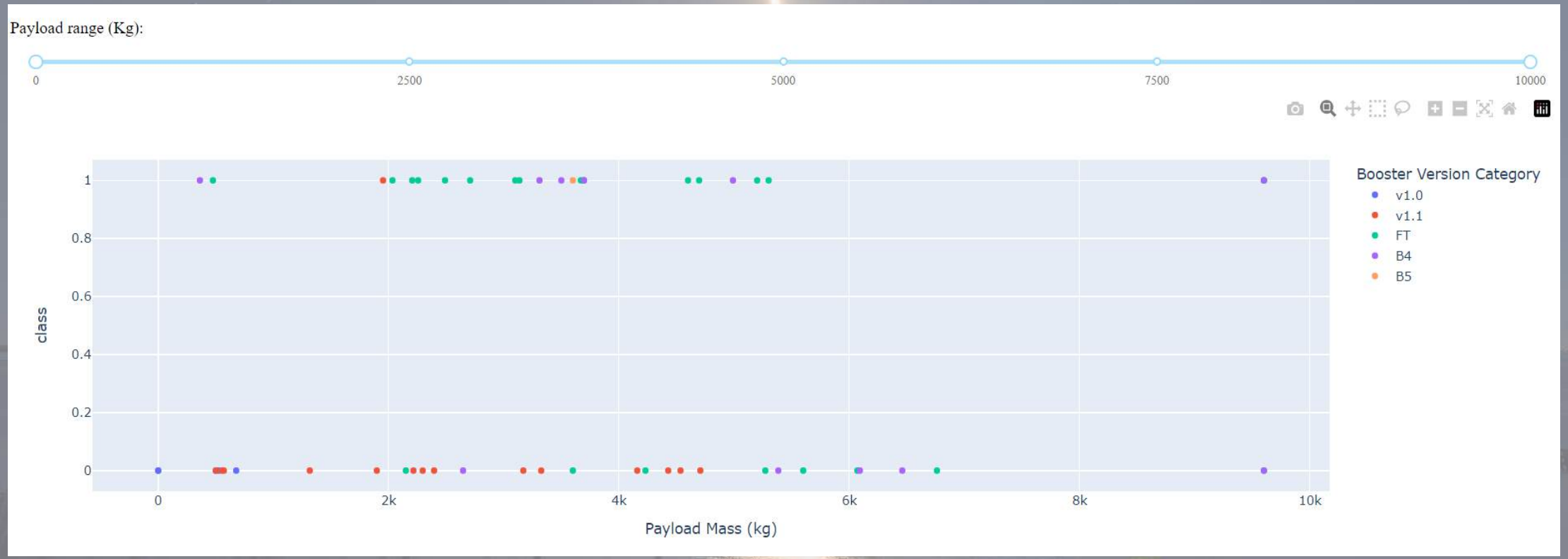
Highest Success Rate Launch Site

- KSC LC-39A has the highest success rate of 76.9% among all the other launch sites.



Payload Vs Launch Outcome

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



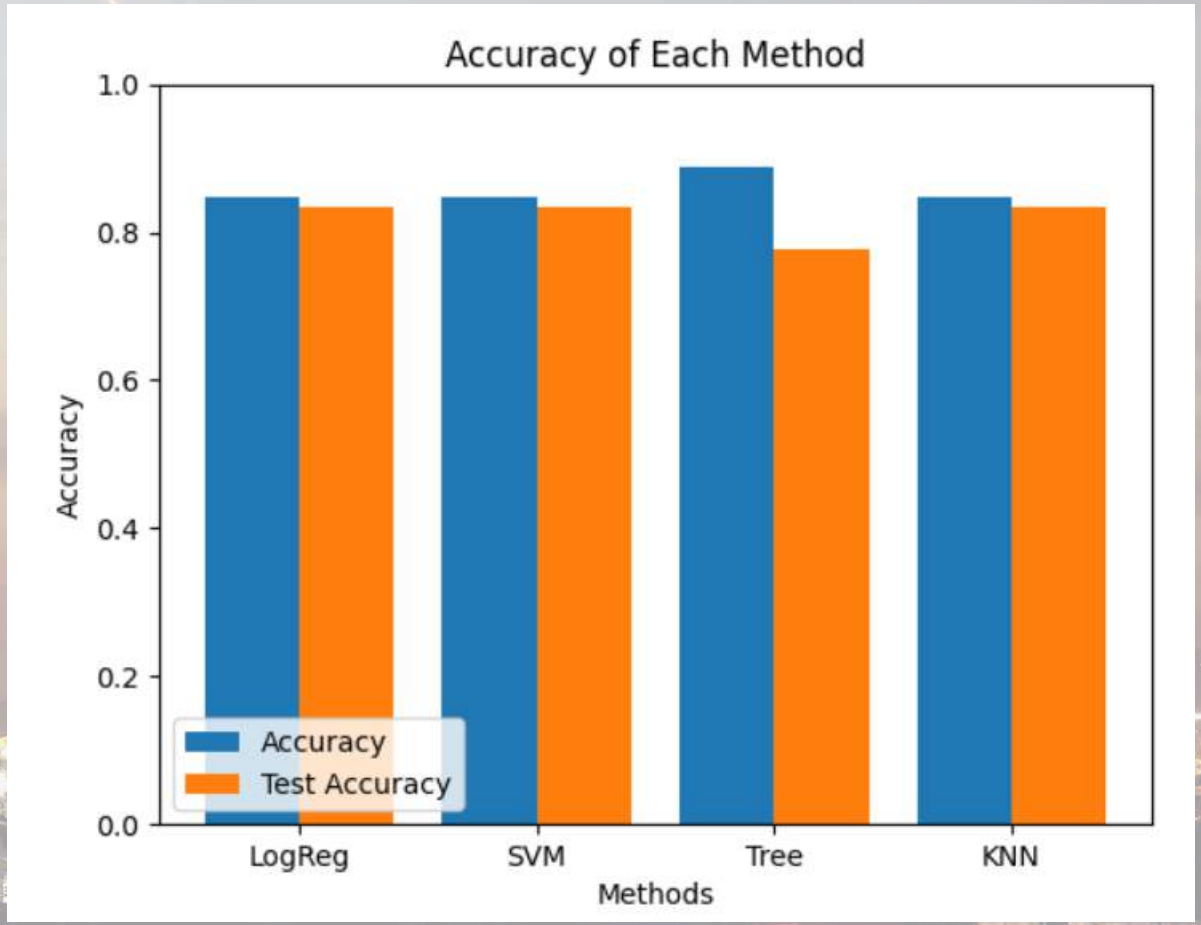
Section 5

Predictive Analysis (Classification)

Classification Accuracy

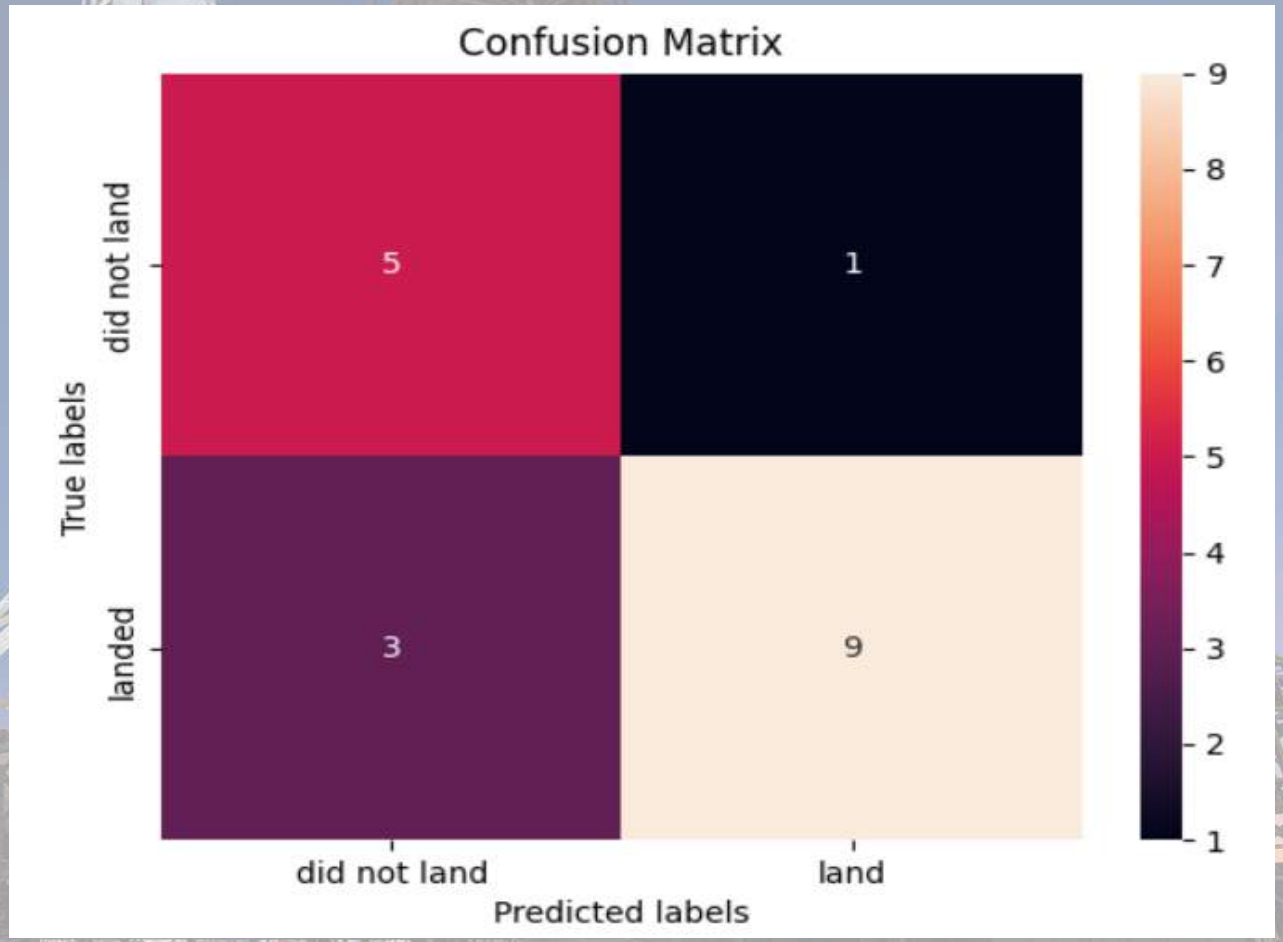
- Four classification models were tested, and their accuracies are plotted beside.

- The classifier model with the highest accuracy is the ‘Decision Tree’.



Confusion Matrix

- The Decision Tree confusion matrix shows the largest number of true positive and true negative compared to other models.



Conclusions

- The larger the flight amount at a launch site, the greater the success rate at a launchsite.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO and SSO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

Appendix

- GitHub repository URL: https://github.com/jadliudit/testrepo

Thank you!