IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Abhishek
08/11/2023

Outline

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

Executive Summary

Methodologies used:

- Data collection using API call or web scraping
- Data wrangling
- EDA with SQL and visualization
- Building an interactive map with Folium
- Building a dashboard with Plotly Dash
- Predictive analytics (classification)

Results obtained:

- Exploratory data analysis
- Visual analytics
- Predictive analytics
- Conclusions

Introduction

- Project background and context:

Our client SpaceY, would like to bid against SpaceX for a rocket launch. SpaceX, founded by Elon Musk, is currently the leading company in this space. They advertise Falcon 9 rocket launches on their website at a cost of $62million. Other companies provide their cost upwards of $165million each. Much of the savings is because SpaceX can reuse the first stage.

- Problems we want to find answers of:

Can we predict if the first landing will be successful?

What factors influence a successful landing?

Is there a machine learning model we can apply to predict?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using API from SpaceX data and later converted to DataFrame.
- Perform data wrangling
  - Identified the null values and delt with that using mean values.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Used multiple machine learning models and tuned them to get optimum result.

Data Collection – SpaceX API

Data was collected using SpaceX web API.

- Used URL to get Data
- Then used response method to get data.
- After getting the data that data converted to pandas DataFrame so that we can use that data in our project.

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

response = requests.get(spacex_url)
```

```
json_data = response.json()

# Convert the JSON data to a Pandas dataframe using .json_normalize()
data = pd.json_normalize(json_data)
```

Using the dataframe `data` print the first 5 rows

```
# Get the head of the dataframe
data.head()
```

For more please refer to this GitHub file:
[IBMCapstoneFinalProject/week 1.1-jupyter-labs-spacex-data-collection-api.ipynb at main · Abhishek7415/IBMCapstoneFinalProject (github.com)]

Data Collection - Scraping

Web Scraping

- Created the ‘http response’ using get method.
- Then converted that response text to ‘Soup’ object using BeautifulSoup.
- Then found all the tables using ‘find_all’ method from soup.
- Converted that collected data to pandas DataFrame.

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

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

```
# Use BeautifulSoup() to create a BeautifulSoup obj
soup = BeautifulSoup(html_content, 'html.parser')
```

```
html_tables = soup.find_all('table')
# Assign the result to a list called `
```

Starting from the third table is our target tab

```
# Let's print the third table and chec
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
column_names = []

# Apply find_all() function with `th` element on
th_elements = first_launch_table.find_all('th')
# Iterate each th element and apply the provided
for th in th_elements:
    name = th.text.strip()
    if name is not None and len(name) > 0:
        column_names.append(name)
# Append the Non-empty column name (`if name is
```

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

For more please refer this Github file:

[IBMCapstoneFinalProject/week 1.2-jupyter-labs-webscraping.ipynb at main · Abhishek7415/IBMCapstoneFinalProject (github.com)]

Data Wrangling

- Data was loaded and then turned into pandas DataFrame.
- Found the percentage of null values from the data, data types.
- Calculated the number of launches on each sites and number of occurrence of each orbit.
- Manipulated the data for landing outcomes as for successful landing it was “1” as “0” for unsuccessful landing.

```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud
df.head(10)
```

```
df.isnull().sum()/len(df)*100
```

```
df.dtypes
```

```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

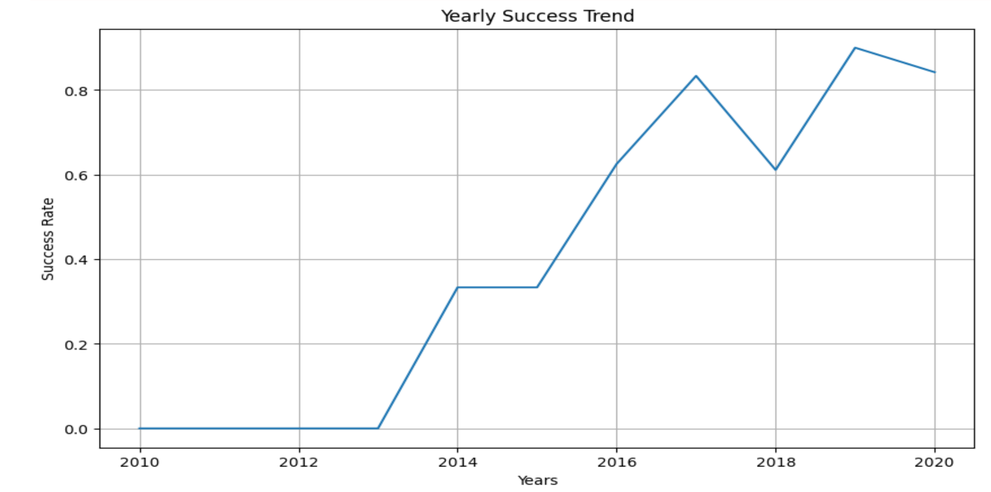
```
# Apply value_counts on Orbit
df['Orbit'].value_counts()
```

```
landing_class = [0 if outcome in bad_outcomes else 1 for outcome in df['Outcome']]
```

For more please refer to GitHub file:
[IBMCapstoneFinalProject/week 1.3-labs-jupyter-spacex-Data wrangling.ipynb at main · Abhishek7415/IBMCapstoneFinalProject (github.com)]()

EDA with Data Visualization

- To predict parameters for successful landing following graphs were created:
    1. Scatter chart on Flight Number vs. Payload
    2. Scatter chart on Flight Number vs. Launch Site
    3. Scatter chart on Flight Number vs. Orbit Type
    4. Scatter chart on Payload vs. Orbit Type
    5. Scatter chart on Payload vs. Launch Site
    6. Bar chart to represent Success Rate per Orbit Type
    7. Line chart to represent Yearly Success trend



For more refer to this GitHub file:
[IBMCapstoneFinalProject/week 2.2-IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb at main · Abhishek7415/IBMCapstoneFinalProject (github.com)]

EDA with SQL

- The following SQL queries were performed for project.
    1. Identify all unique launch sites
    2. Identify all launch sites beginning with ‘CCA’
    3. Calculate the total payload mass carried by boosters launched by NASA (CRS)
    4. Calculate average payload mass carried by booster version F9 v1.1
    5. Identify earliest date when first successful landing in ground pad was achieved
    6. List all boosters which have success in drone ship with payload mass between 4000 to 6000
    7. Calculate total number of successful vs. non-successful outcomes
    8. List all boosters which have the max payload mass
    9. Display all months where landing failed for drone ships in 2015
    10. Rank the count of landing outcomes between 2010-06-04 and 2017-03-20

For more please refer to this GitHub file:

[IBMCapstoneFinalProject/week 2.1-jupyter-labs-eda-sql-coursera_sqllite.ipynb at main · Abhishek7415/IBMCapstoneFinalProject (github.com)]

Build an Interactive Map with Folium

A Folium site map has been created in order to geographically explore the physical location of the launch sites. This will help to determine whether the geographic location contributes to a successful landing. The following objects have been added to the site map.

Markers were placed on the map using the mentioned sites' latitudes and longitudes

1. A **blue** circle to highlight the NASA Johnson Space center
2. **Orange** circles for all launch sites with name labels
3. **Green** and **red** pop-up markers to differentiate between successful (green) and failed (red) landings at each launch site.

Plot lines to highlight the proximity of the launch sites to some landmarks

1. Plot lines with distance displayed to the nearest coast or highway
2. Plot lines with the distance displayed to the nearest city, airport or railway station

For more please refer this GitHub file:

IBMCapstoneFinalProject/week 3.1-IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb at main · Abhishek7415/IBMCapstoneFinalProject (github.com)

Build a Dashboard with Plotly Dash

A dashboard has been created to **allow the client to view dynamic data on successful vs. failed landings per launch site**. It also provides a **scatter chart on payload mass which can be analyzed to check whether it contributes to a successful landing**. The following features / charts are in the dashboard. The dashboard can be further enhanced should the client require it.

1. Dropdown list showing all unique launch sites as well as 'All sites'
2. Pie chart showing the % rate of successful vs. failed launches of the selected dropdown option
3. Range slider for payload mass range to allow users to select the payload range
4. Scatter chart to show the correlation between payload and success rate by booster version

For more please refer to this GitHub file:

IBMCapstoneFinalProject/week 3.2-spacex_dash_app.py at main · Abhishek7415/IBMCapstoneFinalProject (github.com)

Predictive Analysis (Classification)

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

Results

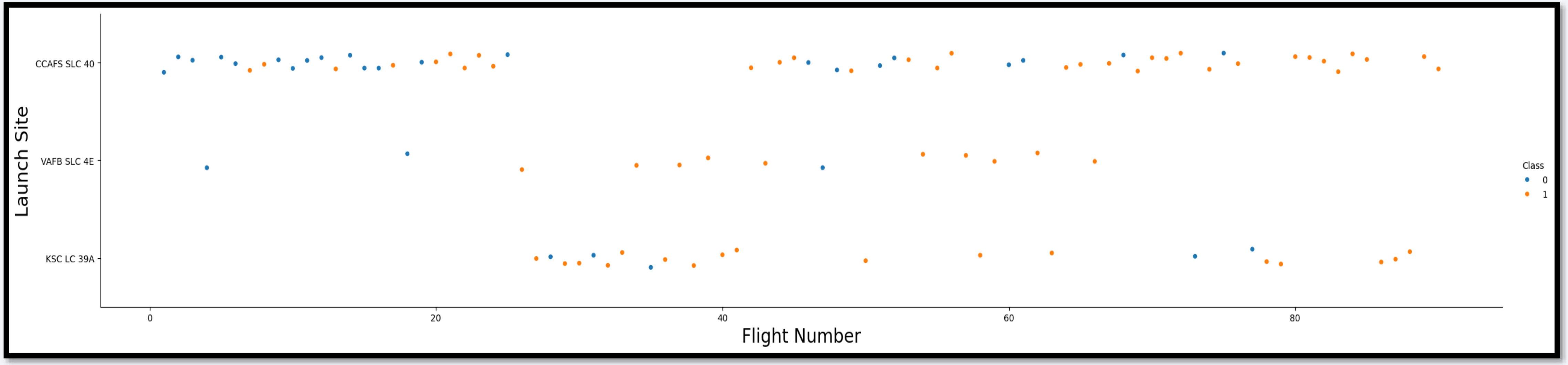
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

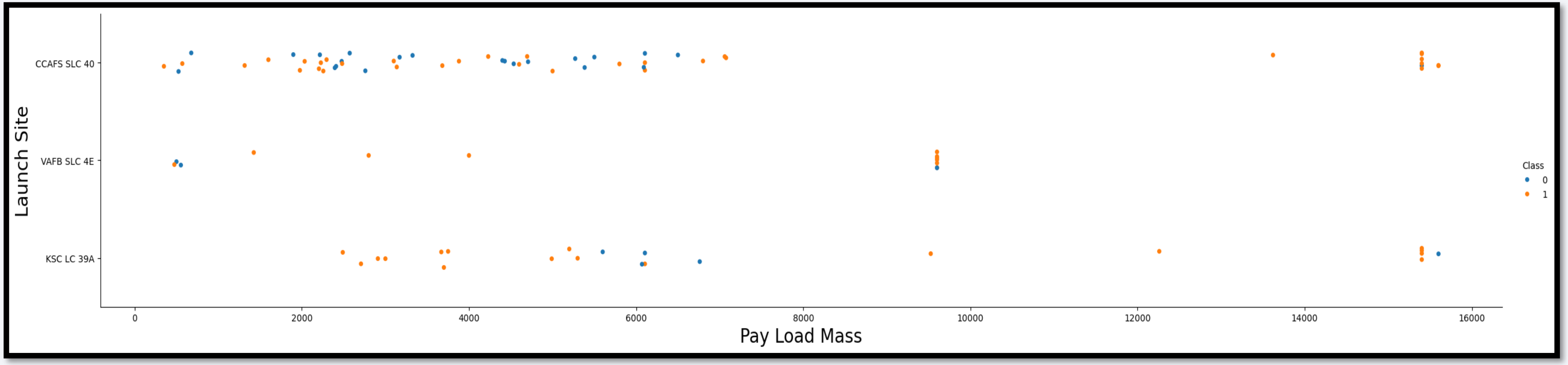
- Scatter plot of Flight Number vs. Launch Site



- We can see that CCAFS SLC 40 have most failure rate than other two launch sites as we have most of launch trials from this site.
- VAFB SLC 4E have least number of launches but most of them are successful.

Payload vs. Launch Site

- Scatter plot of Payload vs. Launch Site

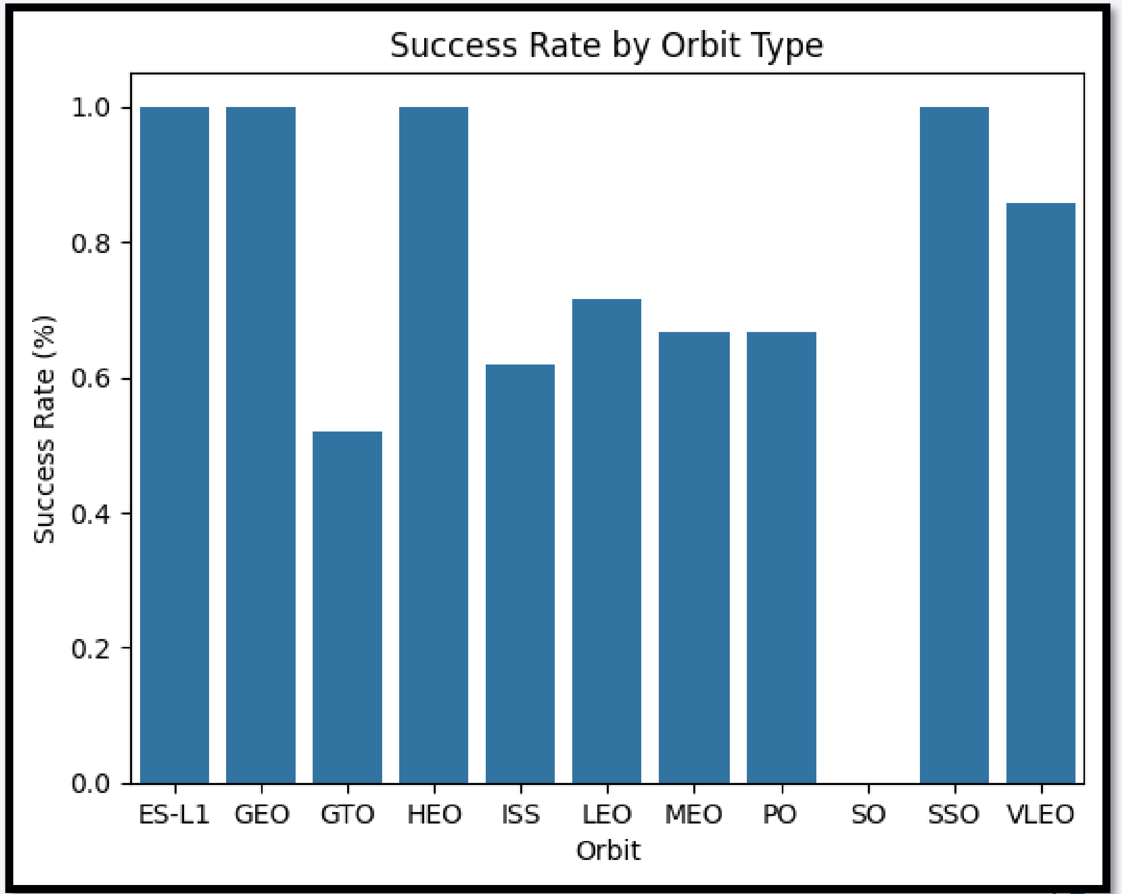


- VAFB SLC 4E site launches rockets less than 10000kg
- CCAFS SLC 40 and KSL LC 39A sites have launches heaviest rockets also

Success Rate vs. Orbit Type

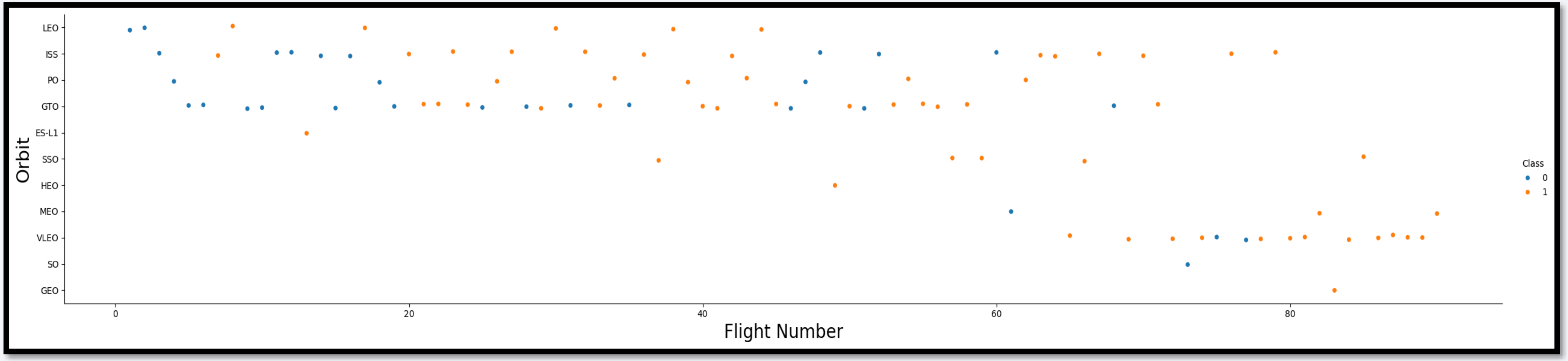
- Bar chart for the success rate of each orbit type

- Orbit SO have zero chances of success rate.
- ES-L1, GEO, HEO, SSO have mostly successful chances.



Flight Number vs. Orbit Type

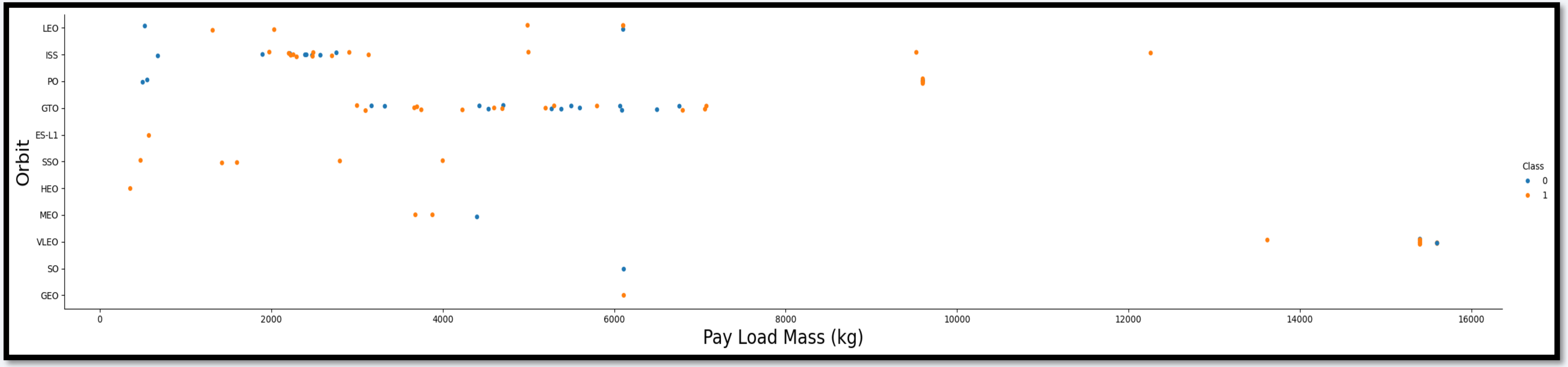
- Show a scatter point of Flight number vs. Orbit type



- GEO have only one flight and it was successful
- Despite of SO have no success there are not enough trails to support the claim of failure of success rate.
- GTO have maximum number of flights.

Payload vs. Orbit Type

- Scatter point of payload vs. orbit type.

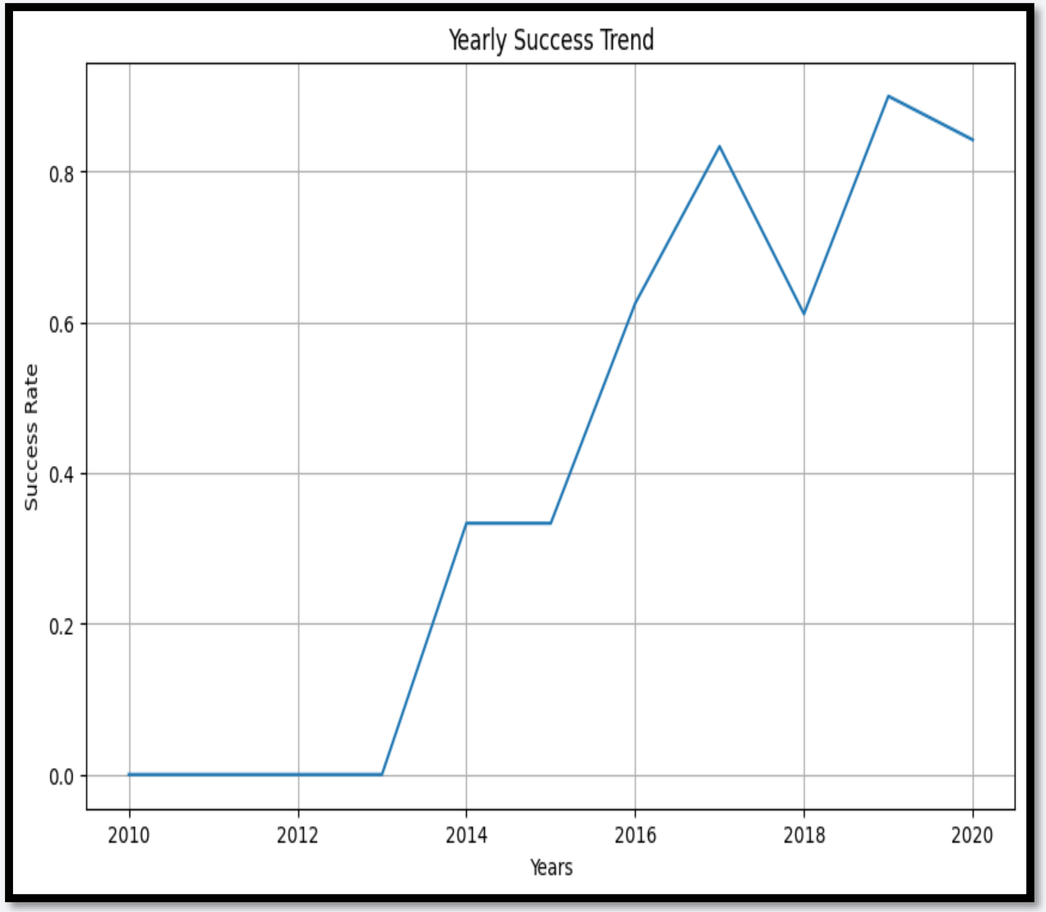


- Mostly launched rockets are less than 8000kg.
- SSO have maximum success rate.
- The heaviest rocket launched was a failure in VLEO orbit.

Launch Success Yearly Trend

- Line chart of yearly average success rate

- As we can see over the year till 2013 there was no success.
- Then success emerged all of a sudden.
- There was a drop in success rate in 2018.



All Launch Site Names

- Unique launch sites

There were only four launch sites

Distinct method was used to retrieve those names

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

Launch Site Names Begin with 'CCA'

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

These are the sites which are starting with CCA.
We used like “CCA%%” method to find these names.

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

```
%sql select sum(PAYLOAD_MASS__KG_)from spacextbl where customer = 'NASA (CRS)';
```

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

sum(PAYLOAD_MASS__KG_)
45596

The total payload mass was 45596kg upto yet.

Average Payload Mass by F9 v1.1

- 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

Average Pay Load Mass is 2928.4kg.

First Successful Ground Landing Date

- First successful landing outcome on ground pad

```
%sql select min(Date) from spacextbl limit 1;
```

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

min(Date)
1/10/2015

First successful landing was on 1/10/2015.
We used min method to extract first date.

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
%%sql

select Booster_Version, Landing_Outcome, PAYLOAD_MASS__KG_
from SPACEXTABLE
where Landing_Outcome = 'Success (drone ship)'
and PAYLOAD_MASS__KG_ between 4000 and 6000
order by PAYLOAD_MASS__KG_
```

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

Booster_Version	Landing_Outcome	PAYLOAD_MASS__KG_
F9 FT B1026	Success (drone ship)	4600
F9 FT B1022	Success (drone ship)	4696
F9 FT B1031.2	Success (drone ship)	5200
F9 FT B1021.2	Success (drone ship)	5300

Total Number of Successful and Failure Mission Outcomes

- Total number of successful and failure mission outcomes

```
]: %sql SELECT "Mission_Outcome", COUNT(*) as "Total_Count" FROM spacextbl GROUP BY "Mission_Outcome";
```

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

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

Boosters Carried Maximum Payload

- Names of the booster which have carried the maximum payload mass

```
%%sql

select Booster_Version, PAYLOAD_MASS__KG_
from SPACEXTABLE
where PAYLOAD_MASS__KG_ in (select max(PAYLOAD_MASS__KG_)
                            from SPACEXTABLE)
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

The SUBQUERY contains the query to search for the MAX Payload Mass from the same database table. The subquery limits the results from the outside query.

Here, you can see that the MAX Payload Mass is 15,600 kg. Only Boosters which have carried this weight are outputted by the query.

2015 Launch Records

- Failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015.

```
[24]: %%sql

select substr(Date,6,2) as 'Month', substr(Date,0,5) as 'Year', Landing_Outcome, Booster_Version, Launch_Site
from SPACEXTABLE
where Landing_Outcome = 'Failure (drone ship)'
and substr(Date,0,5) = '2015'

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

[24]:

Month	Year	Landing_Outcome	Booster_Version	Launch_Site
10	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

```
%%sql

select count(*) as 'Count', Landing_Outcome
from SPACEXTABLE
where Date between '2010-06-04' and '2017-03-20'
group by Landing_Outcome
order by count(*) desc
```

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

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

COUNT counts the records per group / landing outcome.

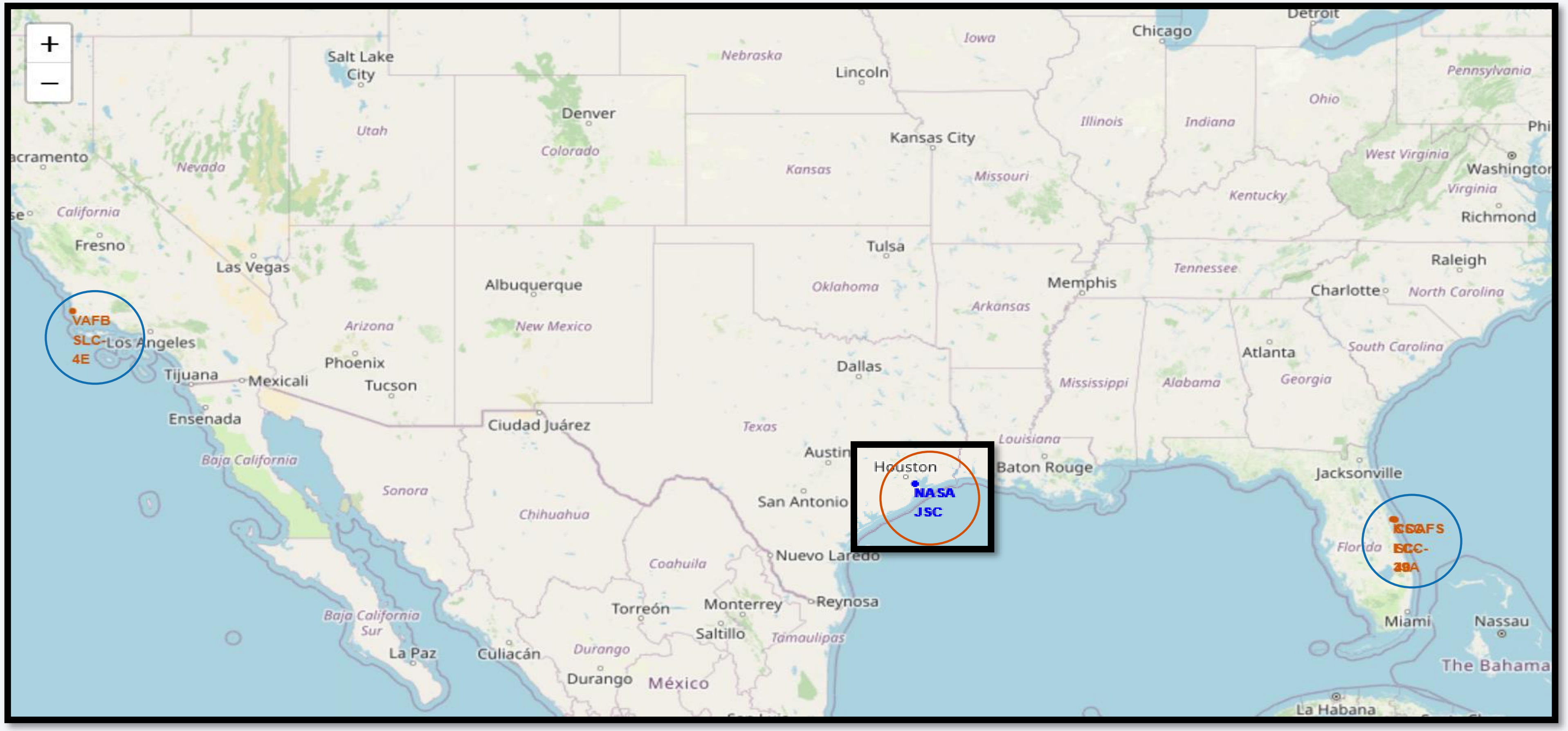
WHERE restricts the records to the date range specified.

ORDER BY sorts out the count of landing outcomes in descending order.

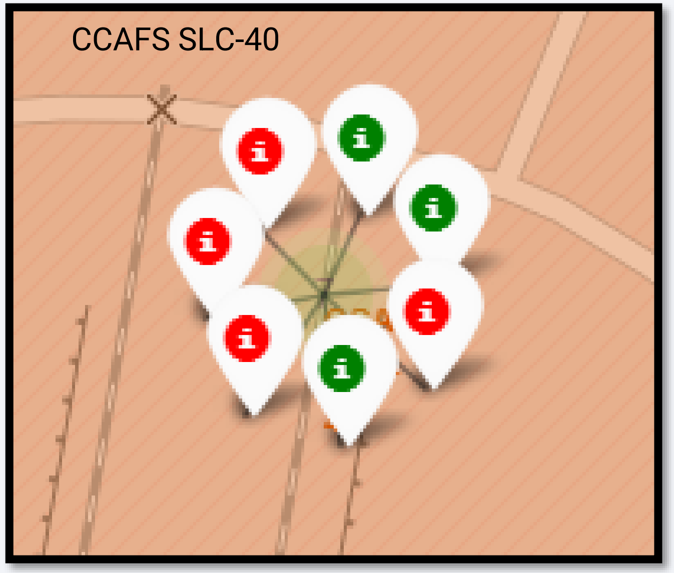
Section 3

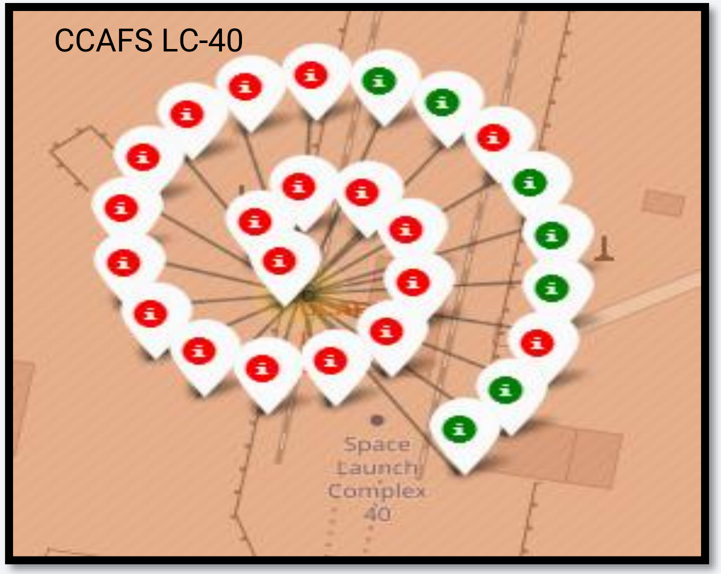
Launch Sites Proximities Analysis

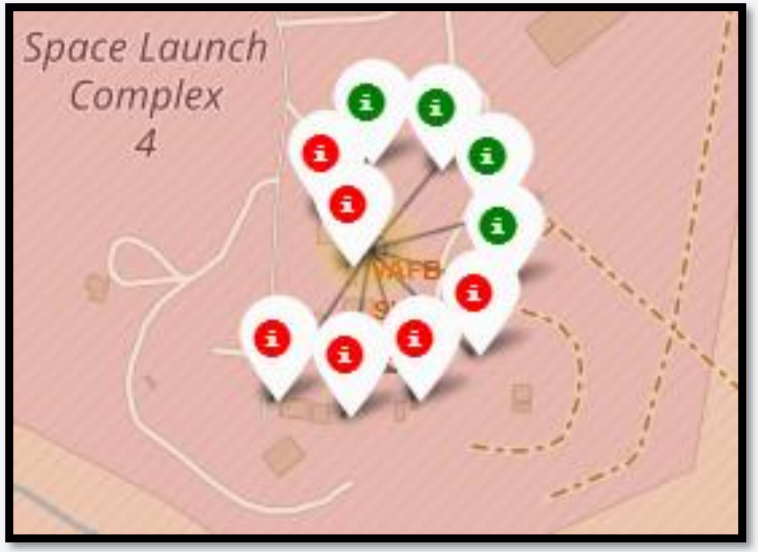
<Folium Map Screenshot 1>

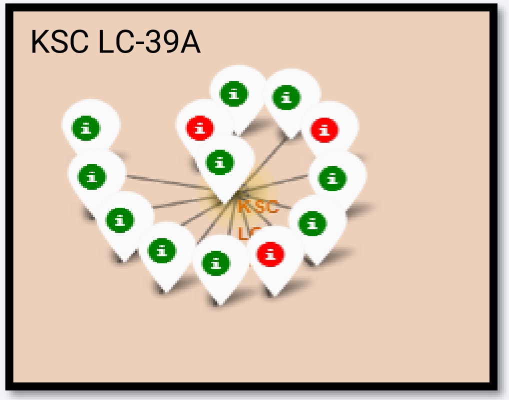


<Folium Map Screenshot 2>



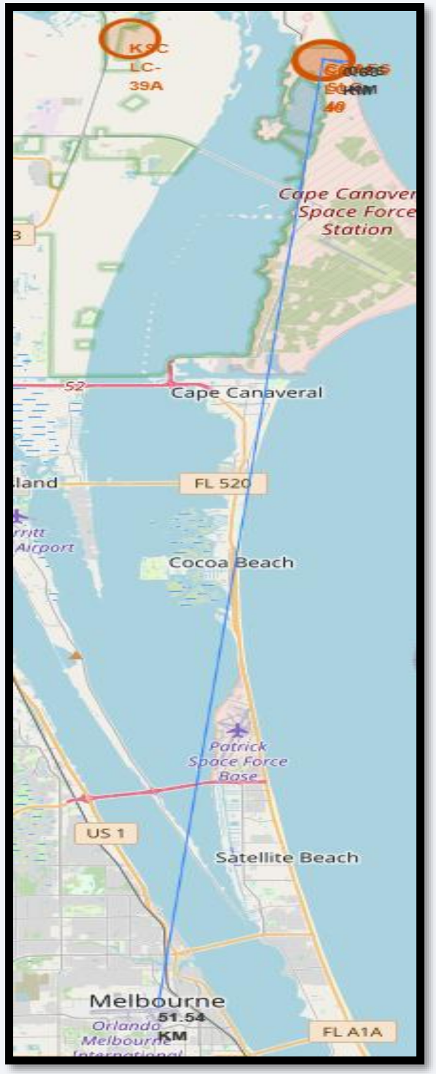


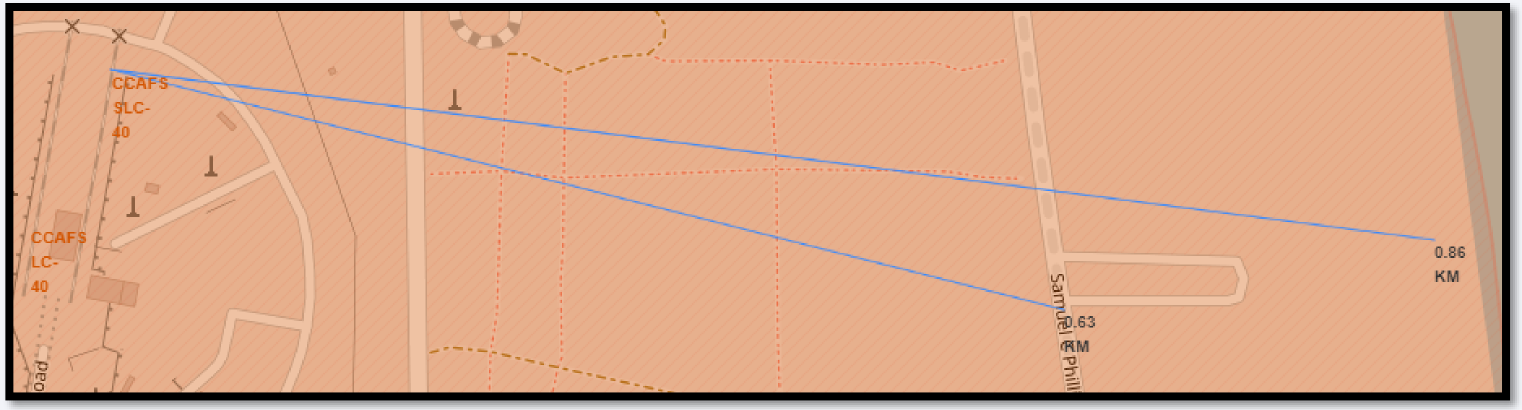




On average, there is a 43.5% success rate across all 3 launch sites in Florida. VAFB, the site in Cali, has a 40% success rate. There's also more launch attempts from Florida (ratio 23:5). **This is because rockets launched closer to the equator can take optimum advantage of the earth's substantial rotational speed. We can infer that equatorial launches help contribute to a successful landing.**

<Folium Map Screenshot 3>

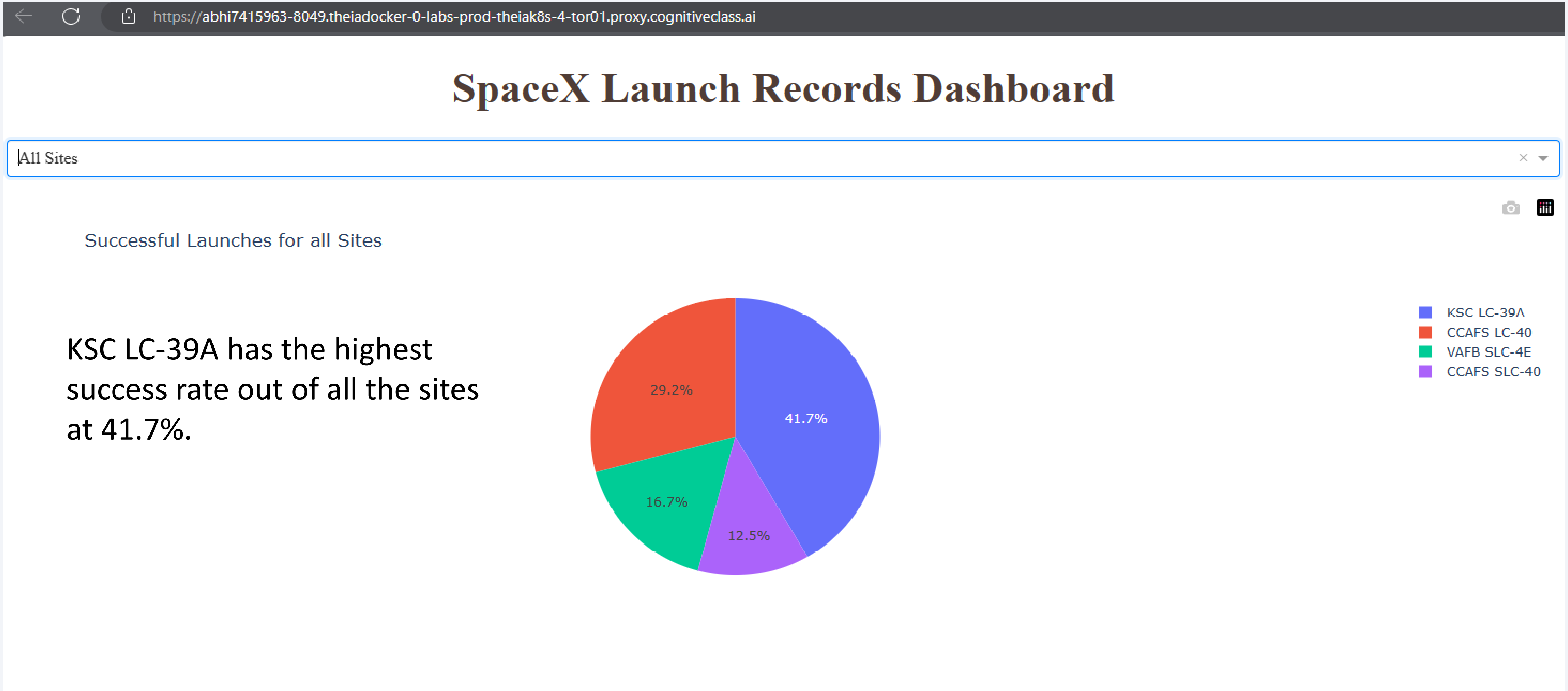




Section 4

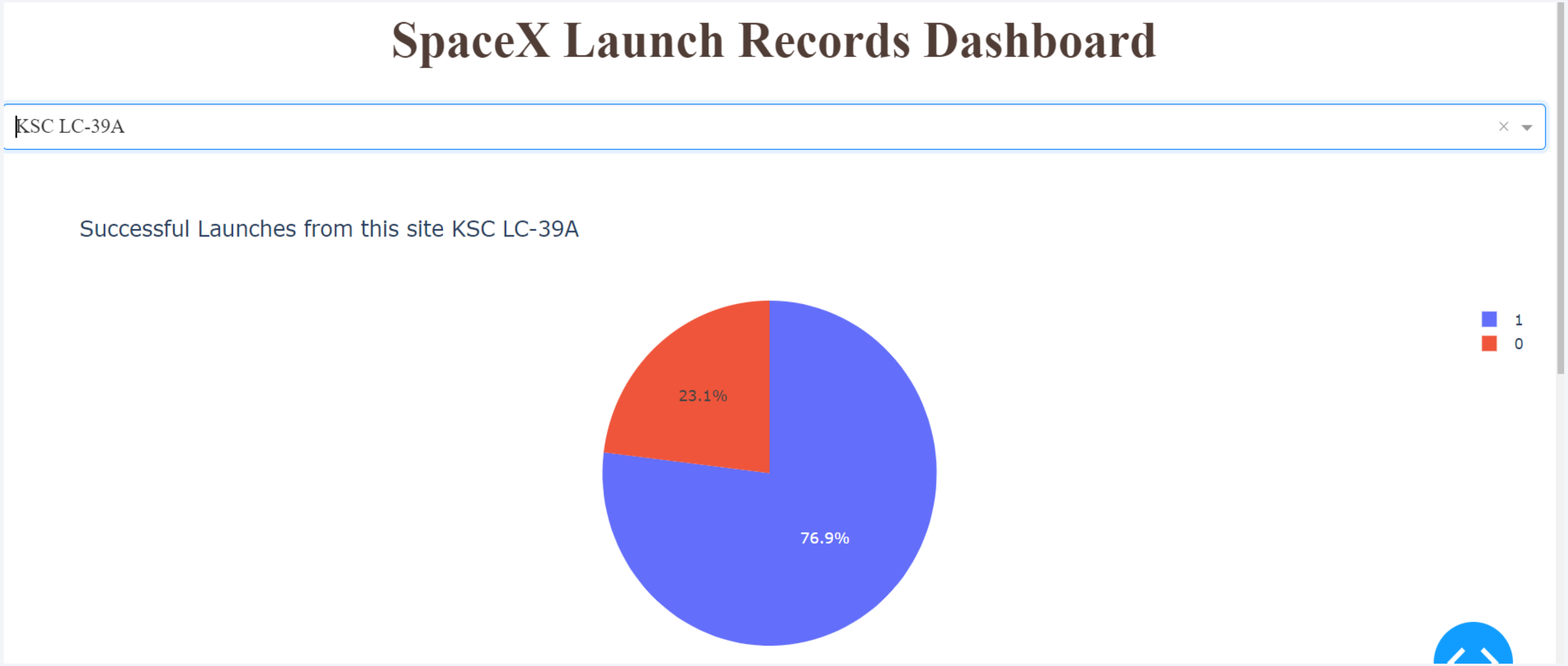
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>



KSC LC-39A has the highest success rate out of all the sites at 41.7%.

<Dashboard Screenshot 2>

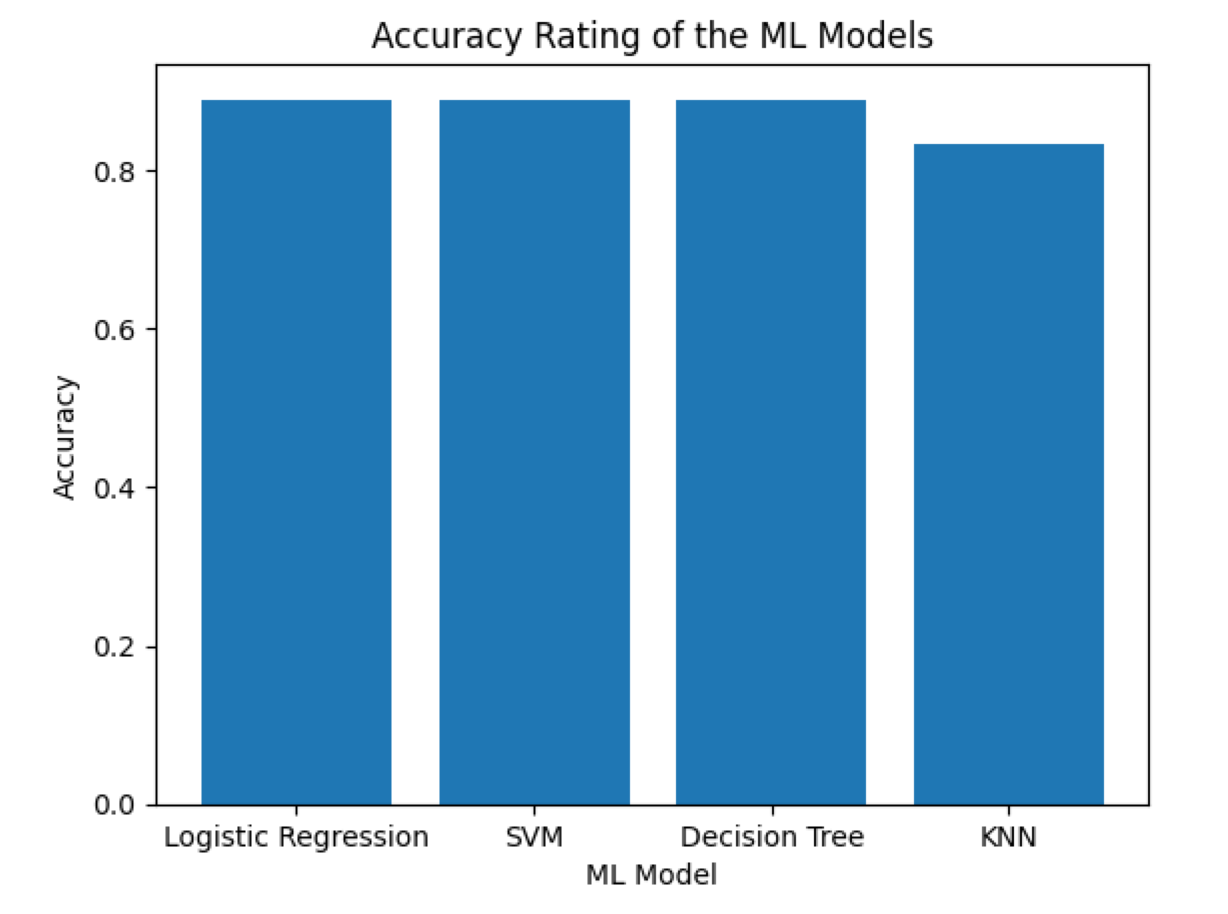


<Dashboard Screenshot 3>

Section 5

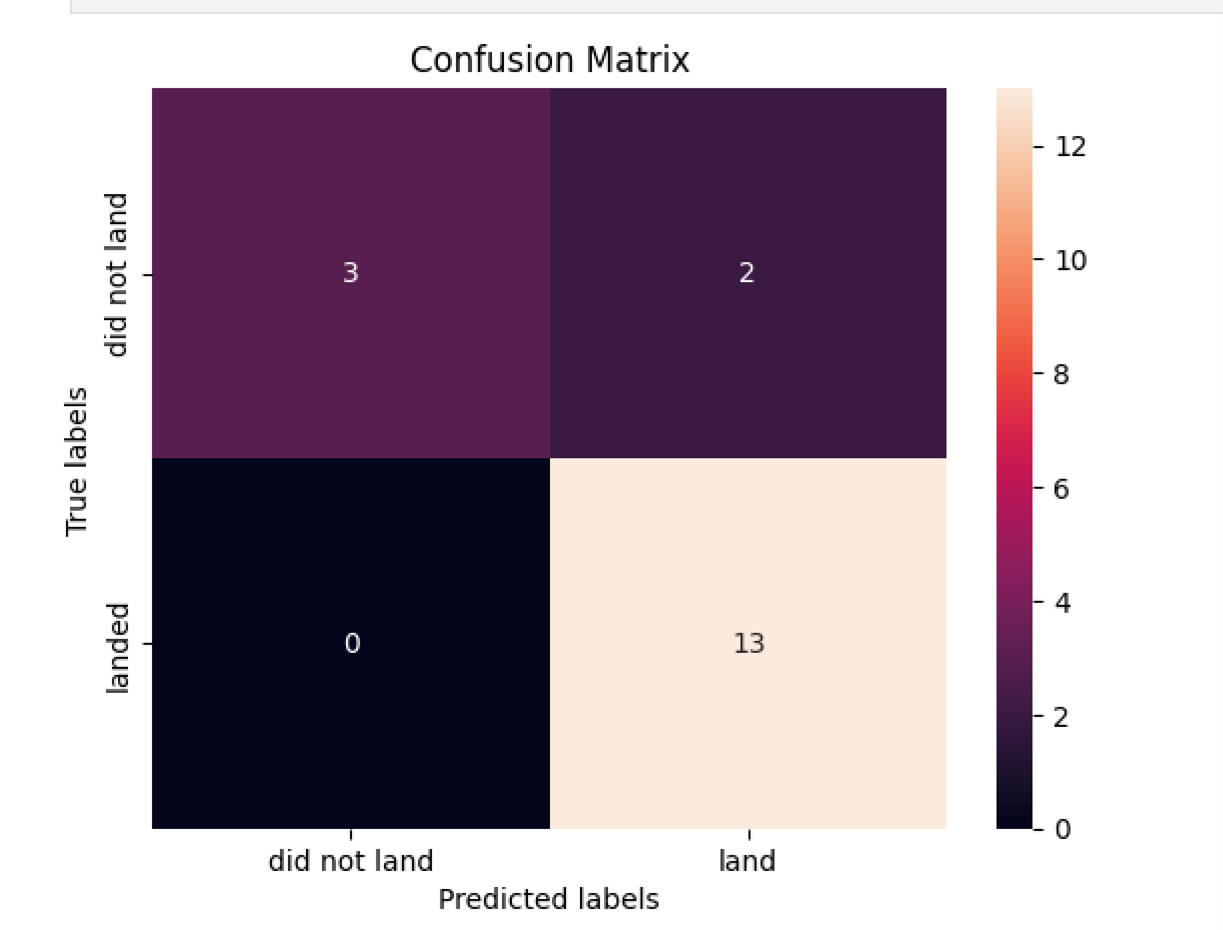
Predictive Analysis (Classification)

Classification Accuracy



- KNN have least accuracy.
- Other 3 have same accuracy.
- Maybe data is not enough available.

Confusion Matrix



All 3 best performing models have same confusion matrix.

Conclusions

- Based on our analyses, we are able to make predictions on whether the first stage can be reused. If the first stage can be reused, this will cost the company less.
- We know that launches are more successful when:
    1. Launches happen from KSC LC-39A, with payload masses above 5,500 kg.
    2. We apply the lessons learned from the previous years as exhibited by the sharp increase yearly.
    3. Launches in orbits ES-L1, GEO, HEO and SSO.
    4. Launched in a site closer to the equator.
- We created 4 models. Based on our initial evaluation, all 4 models: logistic regression, KNN, decision tree and SVM, can all be used to predict but we can loose KNN, with the possibility that we can train and test better with a bigger sample data.

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!