IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Mahdi Alimohammadi>
<4/9/2023>

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - EDA data visualization
  - EDA SQL
  - Interactive map
  - Predictive Analysis
- Summary of all results
  - Dashboard
  - Predictive results

Introduction

- Project background and context
- Problems you want to find answers

Section 1

Methodology

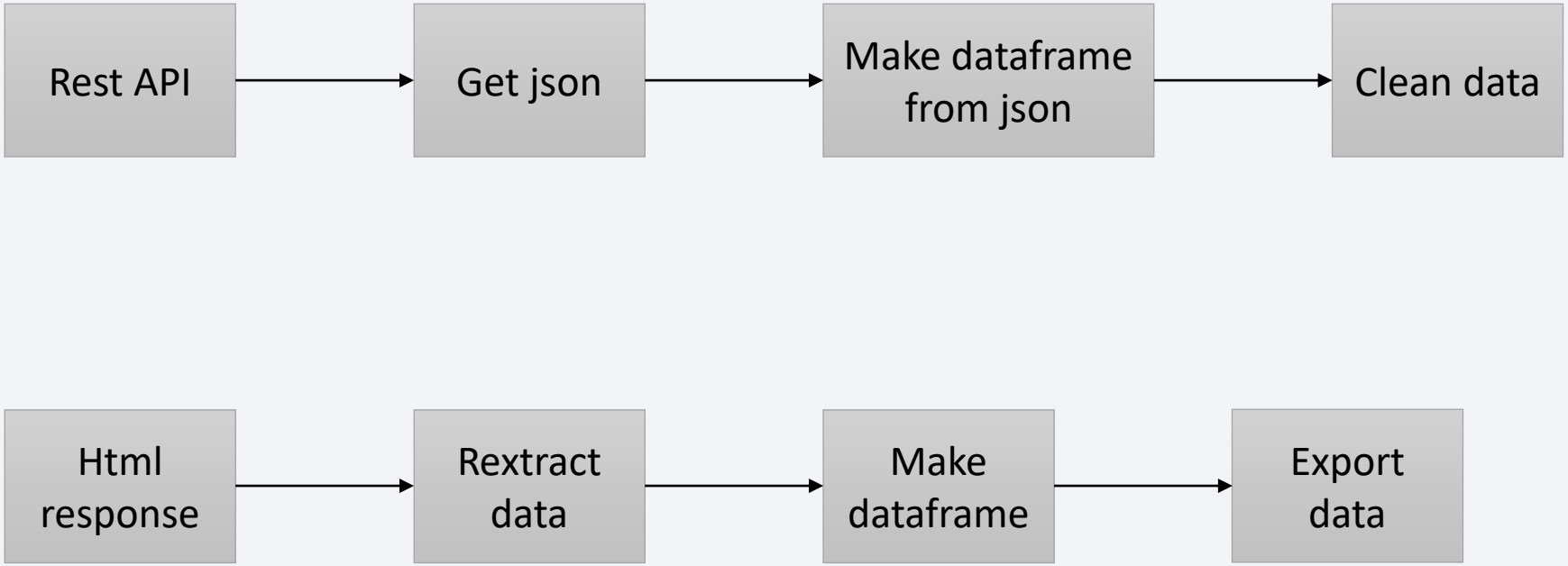
Methodology

Executive Summary

- Data collection methodology:
  - spaceX rest API & Web Scrapping
- Perform data wrangling
  - Hot encoding for classification & drop unimportant columns
- 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

- Datasets are collected from SpaceX Rest API and webscrapping Wikipedia



Data Collection – SpaceX API

1. Get response from API
2. Convert response to json
3. Transfrom data
4. Create dictionary
5. Create & filter dataframe
6. Export to file

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

response = requests.get(spacex_url)

data = response.json()
data = pd.json_normalize(data)

getBoosterVersion(data)

getLaunchSite(data)

getPayloadData(data)

getCoreData(data)
```

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

```
data = pd.DataFrame({key:pd.Series(value) for key, value in launch_dict.items()})
data_falcon9 = data[data['BoosterVersion']!='Falcon 1']

data_falcon9.to_csv('dataset_part_1.csv', index=False)
```


https://github.com/meti78/My-Files/blob/Data-Science-Capstone/DataCollection_API.ipynb

Data Collection - Scraping

1.Get response from Html

2.Create BeautifulSoup

3.Find all tables

4.Get column names

5.Create dictionary

6.Add data to keys

7.Create dataframe and export to file

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

```
soup = BeautifulSoup(response.text, "html.parser")
```

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

```
for i in first_launch_table.find_all('th'):
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0 :
        column_names.append(name)
```

```
launch_dict= dict.fromkeys(column_names)

del launch_dict['Date and time ( )']

launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

```
df=pd.DataFrame(launch_dict)
```

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

https://github.com/meti78/My-Files/blob/Data-Science-Capstone/DataCollection_Webscraping.ipynb

Data Wrangling

1.Calculate launch numbers

```
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

2.Calculate number of each orbit

```
df['Orbit'].value_counts()
```

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

3.Calculate number of mission outcome

```
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

4.Create landing outcome label

```
landing_class = []
for key,value in df["Outcome"].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

```
df['Class']=landing_class
df[['Class']].head(8)
```

5.Export to file

```
df.to_csv("dataset_part_2.csv", index=False)
```

https://github.com/meti78/My-Files/blob/Data-Science-Capstone/DataWrangling.ipynb

EDA with Data Visualization

- Bar graph
  - Success rate vs orbit
- Line graph
  - Success rate vs year
- Scatter plot
  - Payload mass vs flight number
  - Launch site vs flight number
  - Launch site vs payload mass
  - Orbit vs success rate
  - Orbit vs flight number
  - Orbit vs payload mass

https://github.com/meti78/My-Files/blob/Data-Science-Capstone/EDA_DataVisualization.ipynb

EDA with SQL

We load the dataset into a database. Then we apply EDA with SQL to know the data better such as:

- Name of unique sites
- Total payload masses
- and …

https://github.com/meti78/My-Files/blob/Data-Science-Capstone/EDA_SQL.ipynb

Build an Interactive Map with Folium

https://github.com/meti78/My-Files/blob/Data-Science-Capstone/LaunchSiteLocation.ipynb

Build a Dashboard with Plotly Dash

Build a dashboard to plot pie charts showing total launches for each sites and scatter plots showing the relationship with outcome and payload nass for different booster versions

https://github.com/meti78/My-Files/blob/Data-Science-Capstone/spacex_dash_app.py

Predictive Analysis (Classification)

- Data preparation
  - Normalize data and split into train and test sets
- Model preparation
  - Train with different machine learning algorithms
- Model evaluation
  - Compute accuracy for each model
  - Plot confusion matrix
- Model comparison
  - Compare different models and chose the model with best accuracy

https://github.com/meti78/My-Files/blob/Data-Science-Capstone/Machine%20Learning%20Prediction.ipynb

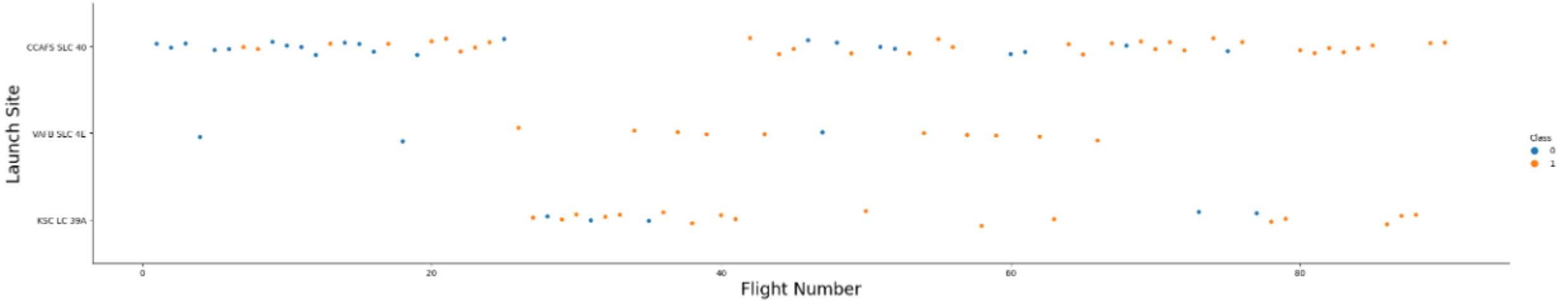
Results

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

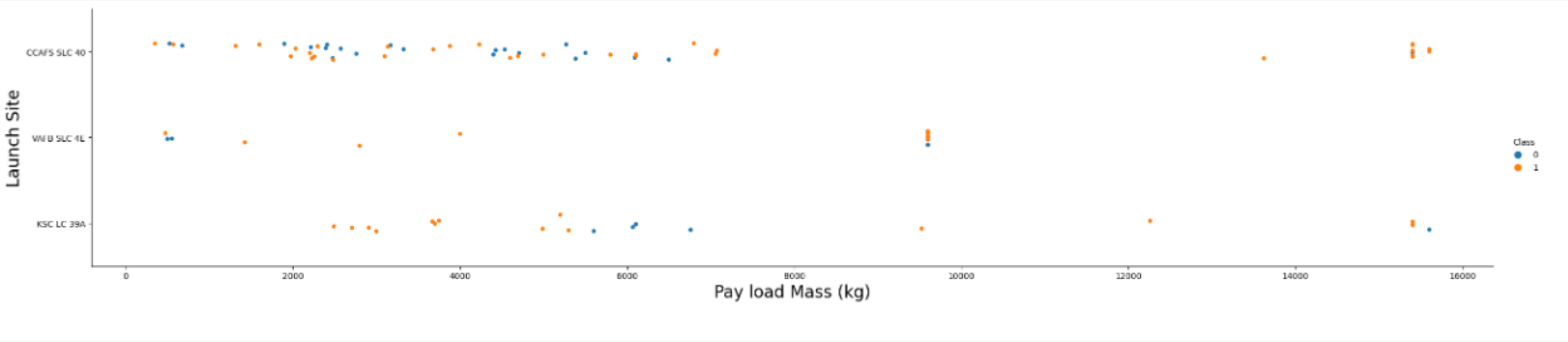
Section 2

Insights drawn from EDA

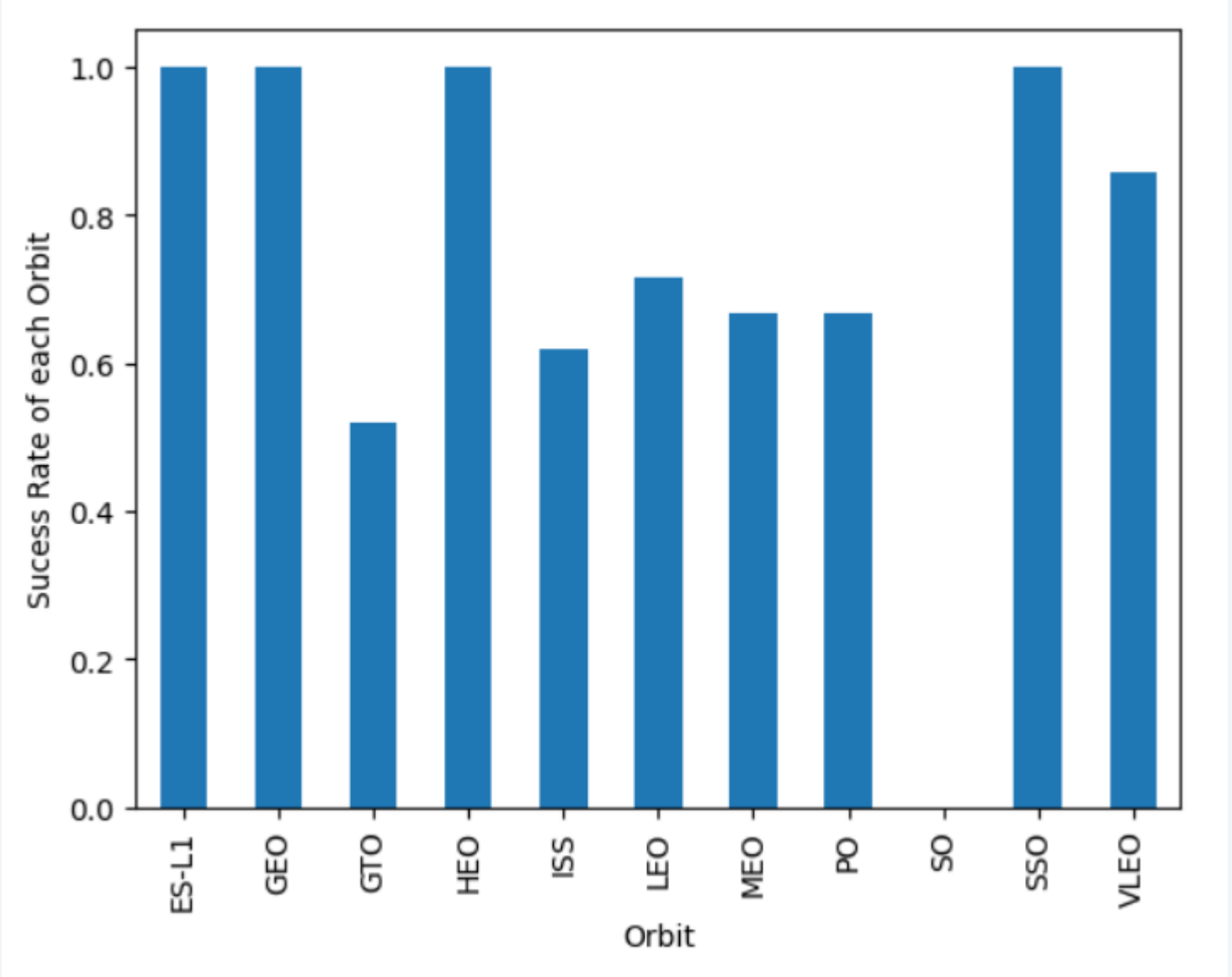
Flight Number vs. Launch Site



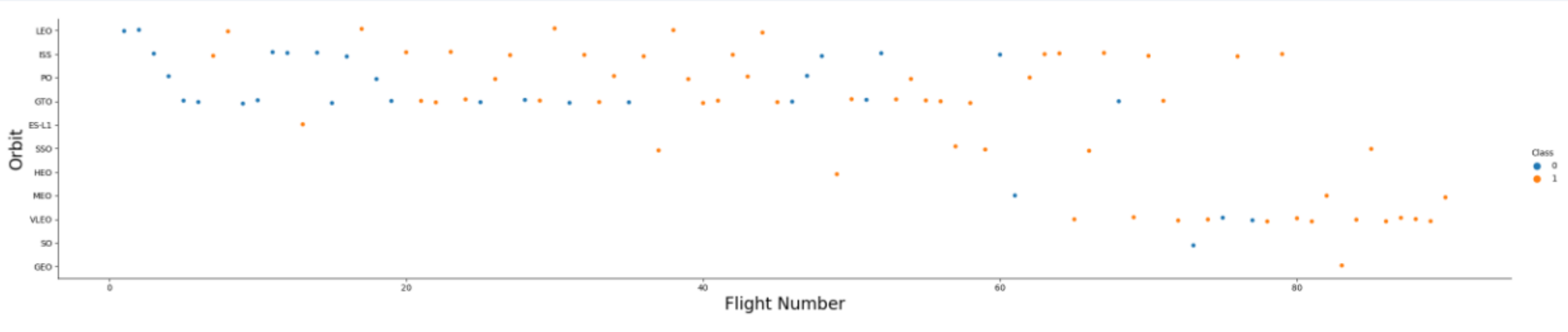
Payload vs. Launch Site



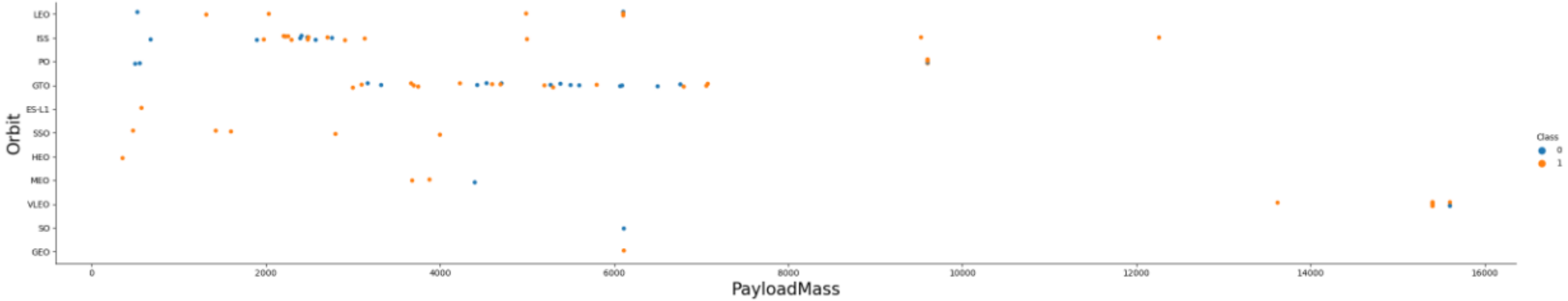
Success Rate vs. Orbit Type



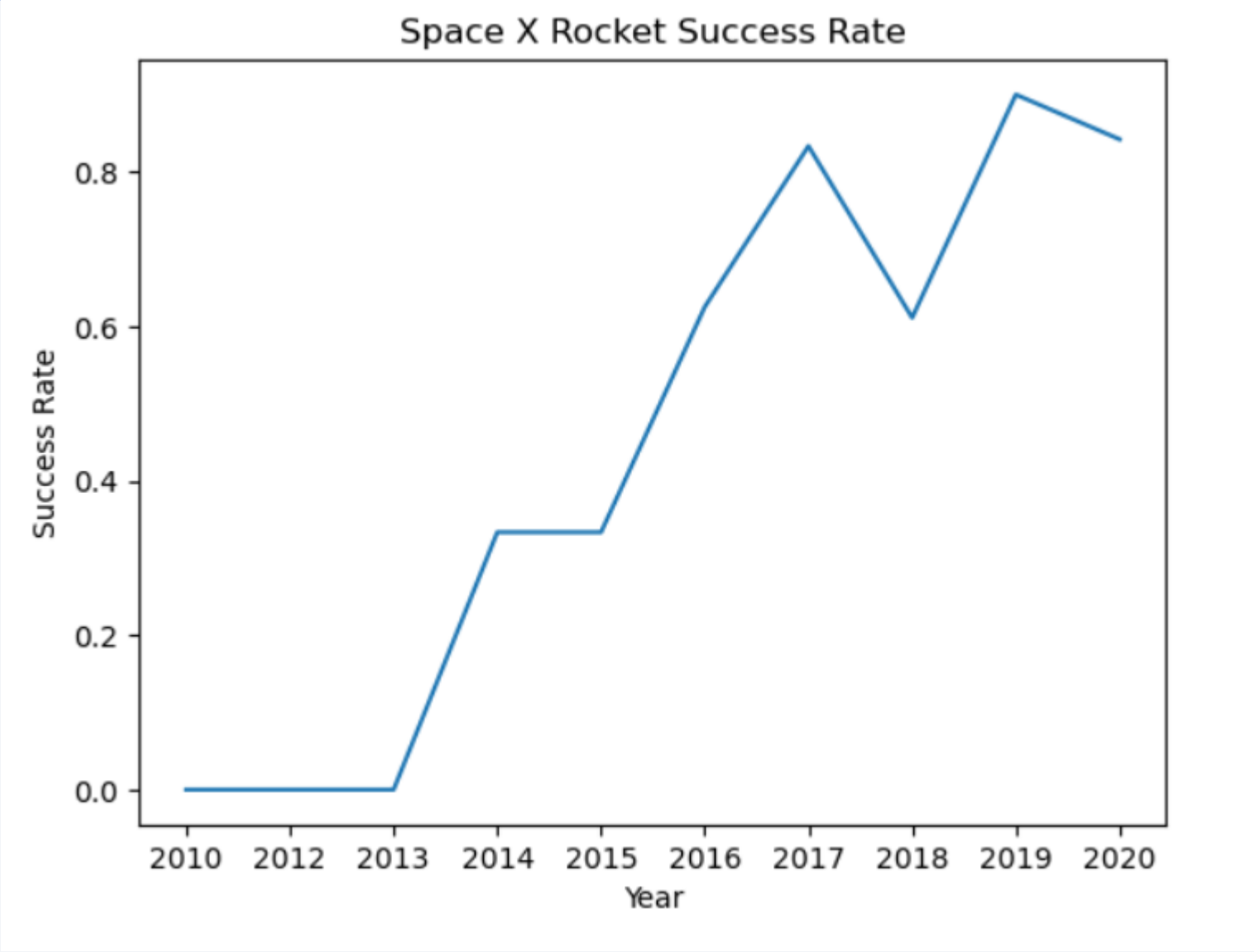
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

We use “DISTINCT” expression to get each unique launch site name

```
%sql SELECT DISTINCT "LAUNCH_SITE" FROM SPACEXTBL
```

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

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

Launch Site Names Begin with 'CCA'

We use “WHERE” and “LIKE” expression to find sites that begin with “CCA”.Then “LIMIT” helps us to show just 5 sites.

```
%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

Total Payload Mass

This command returns sum of all payload masses where the customer is NASA.

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "CUSTOMER" = 'NASA (CRS)'
```

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

This query calculate the average payload mass carried by booster version F9 v1.1

```
%sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "BOOSTER_VERSION" LIKE '%F9 v1.1%'
```

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

This query find the dates of the first successful landing outcome on ground pad

```
%sql SELECT MIN("DATE") FROM SPACEXTBL WHERE "Landing_Outcome" LIKE '%Success%';
```

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

MIN(DATE)
1/5/2017

Successful Drone Ship Landing with Payload between 4000 and 6000

This query lists the 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" FROM SPACEXTBL WHERE "Landing_Outcome" = 'Success (drone ship)' \
AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000;
```

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

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

Total Number of Successful and Failure Mission Outcomes

This query calculatec the total number of successful and failure mission outcomes

```
%sql SELECT (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Success%') AS SUCCESS, \
(SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Failure%') AS FAILURE
```

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

SUCCESS	FAILURE
100	1

Boosters Carried Maximum Payload

This query lists the names of the booster which have carried the maximum payload mass

```
%sql SELECT DISTINCT "BOOSTER_VERSION" FROM SPACEXTBL \
WHERE "PAYLOAD_MASS__KG_" = (SELECT max("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

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

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

2015 Launch Records

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

```
%sql SELECT substr("DATE", 4, 2) AS MONTH, "BOOSTER_VERSION", "LAUNCH_SITE" FROM SPACEXTBL\
WHERE "LANDING_OUTCOME" = 'Failure (drone ship)' and substr("DATE",7,4) = '2015'
```

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

MONTH	Booster_Version	Launch_Site
04	F9 v1.1 B1015	CCAFS LC-40

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

This query 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

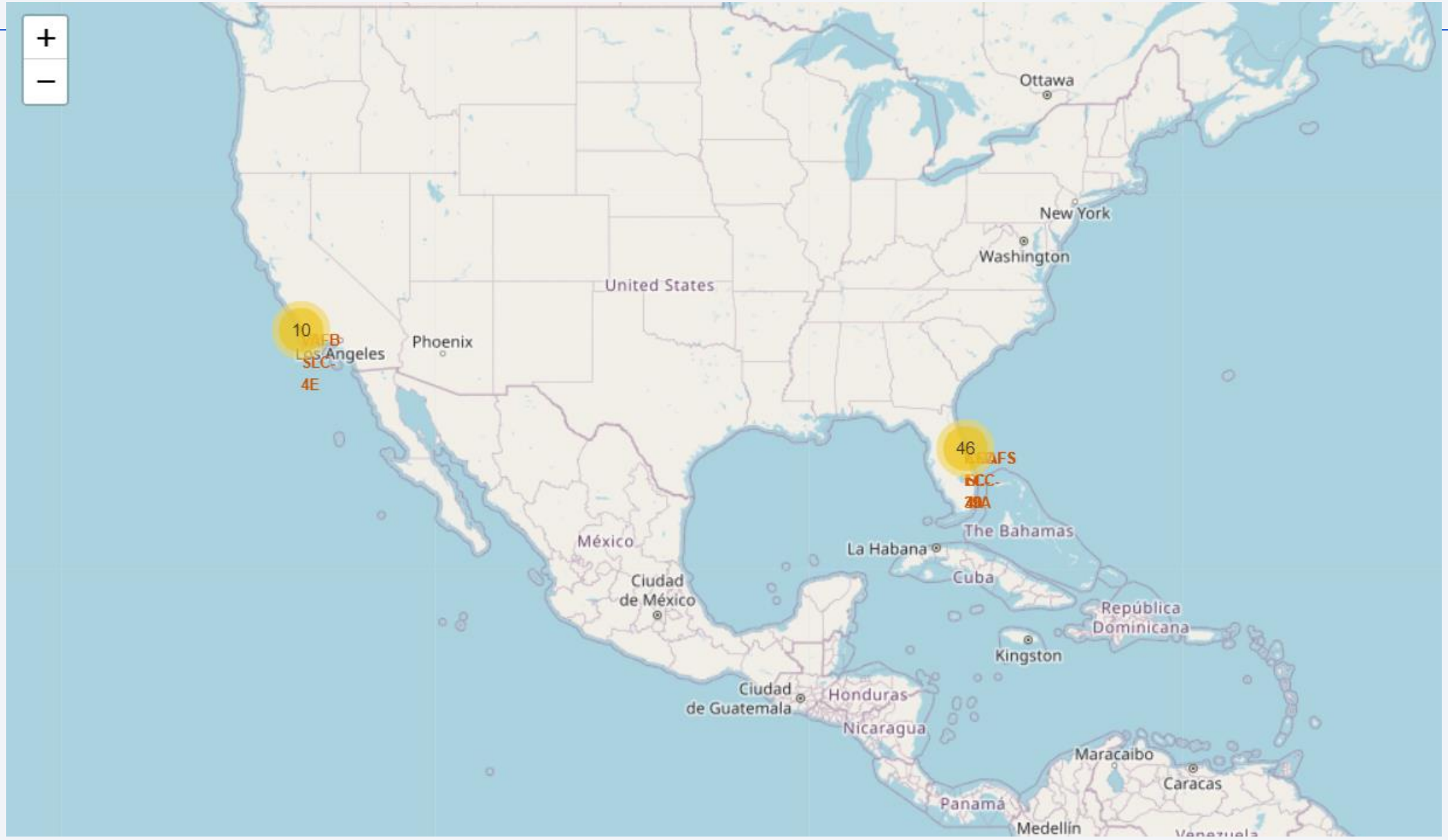
```
%sql SELECT "LANDING_OUTCOME", COUNT("LANDING_OUTCOME") FROM SPACEXTBL\
WHERE "DATE" >= '04-06-2010' and "DATE" <= '20-03-2017' and "LANDING_OUTCOME" LIKE '%Success%'\
GROUP BY "LANDING_OUTCOME" \
ORDER BY COUNT("LANDING_OUTCOME") DESC ;
```

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

Landing_Outcome	COUNT(LANDING_OUTCOME)
Success	11
Success (ground pad)	5
Success (drone ship)	5


Section 3
Launch Sites
Proximities Analysis

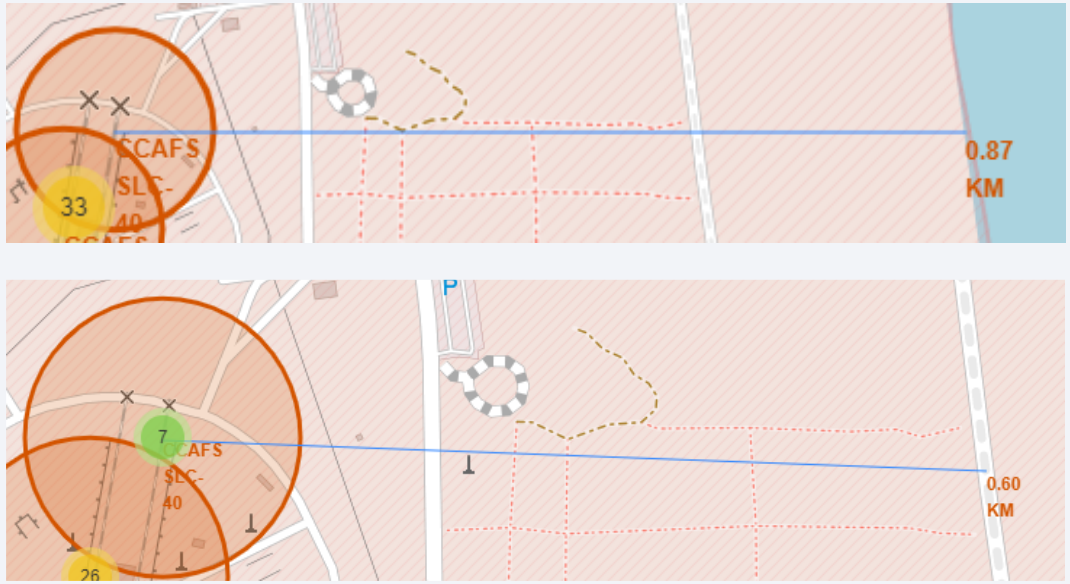
Ground stations

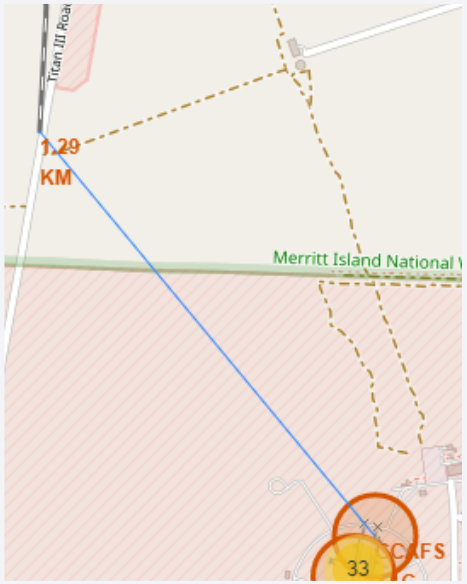


<Folium Map Screenshot 2>

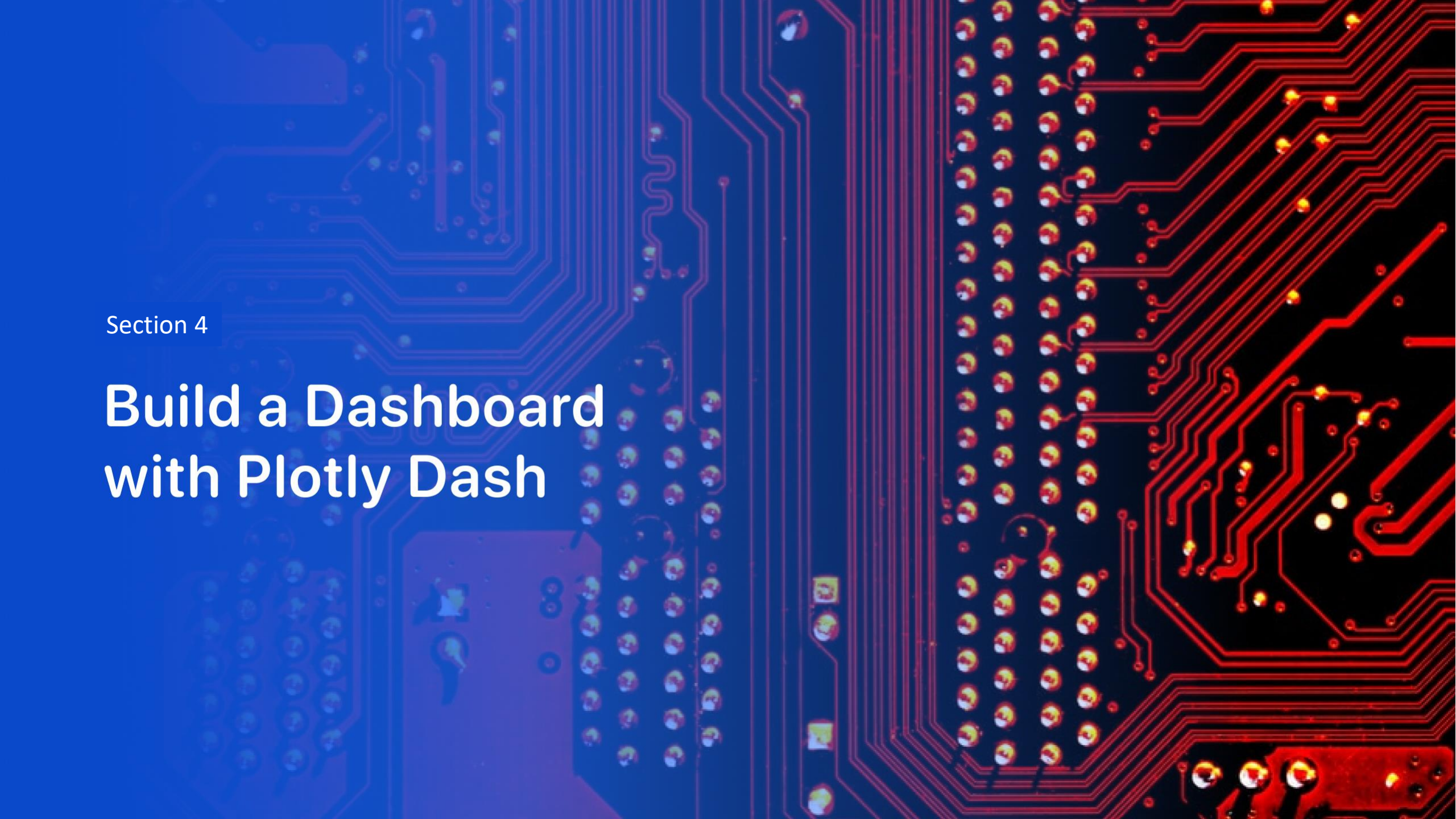
- Replace <Folium map screenshot 2> title with an appropriate title
- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map
- Explain the important elements and findings on the screenshot

Distances

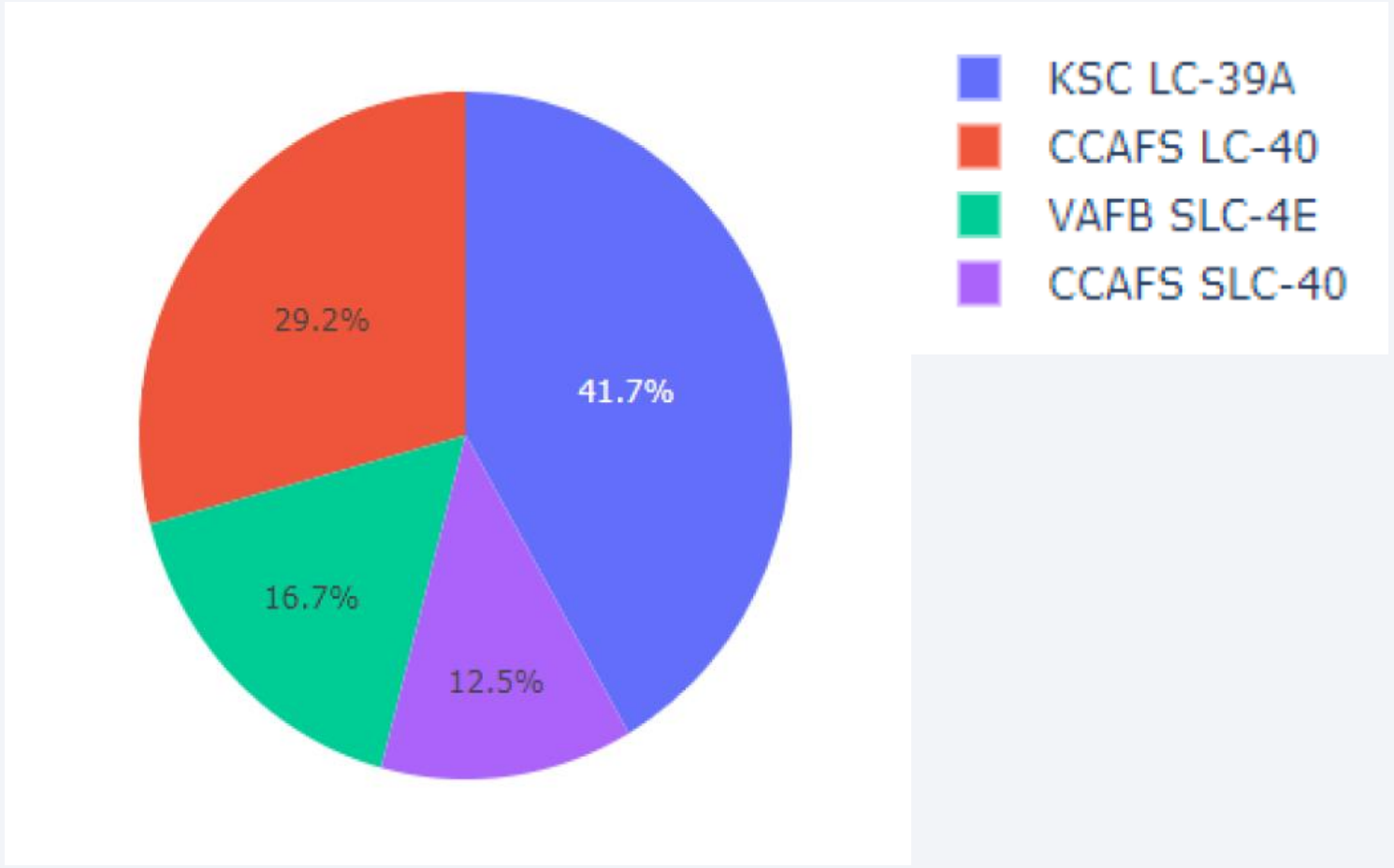

CCAFS
SLC-
40
33
0.87
KM
7
CCAFS
SLC-
40
26
0.60
KM


1.29
KM
Merritt Island National
33
CCAFS


Island National
Wildlife
Refuge
Titusville
23.18
KM
Shuttle
Landing
Facility
13
KSC
LC-
39A
33
US 1
FL 405
Space Coast
Regional
Florida


Section 4
Build a Dashboard with Plotly Dash

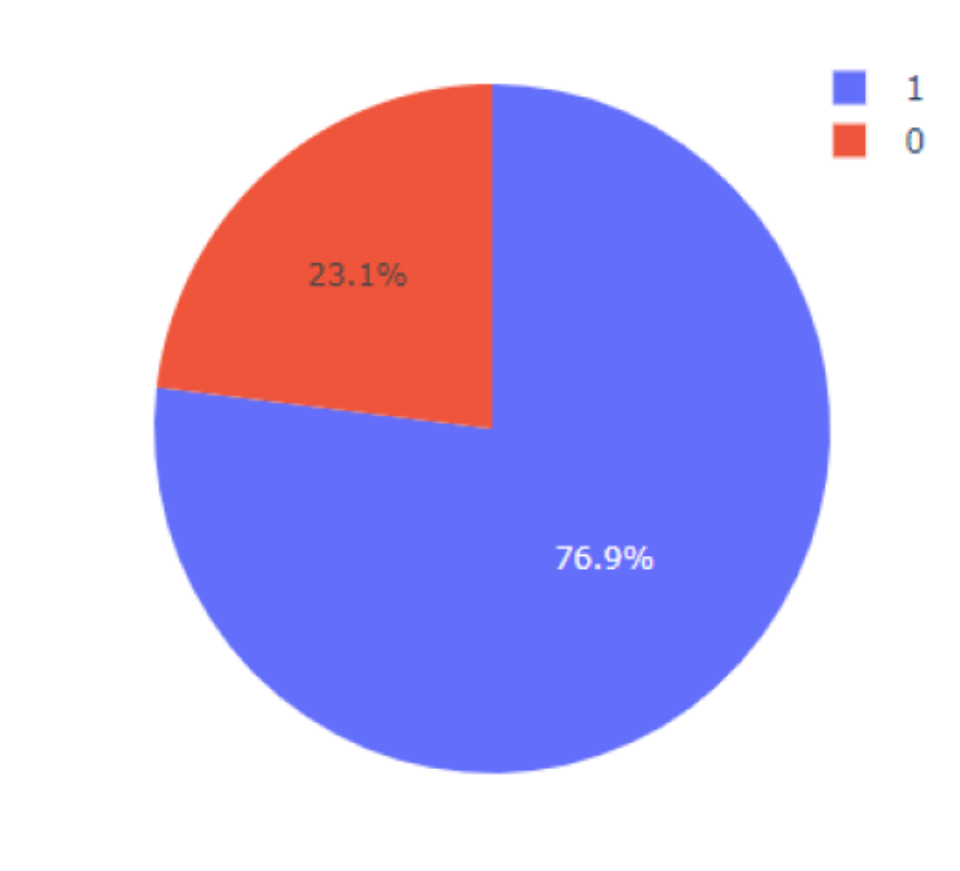
Success count for each site



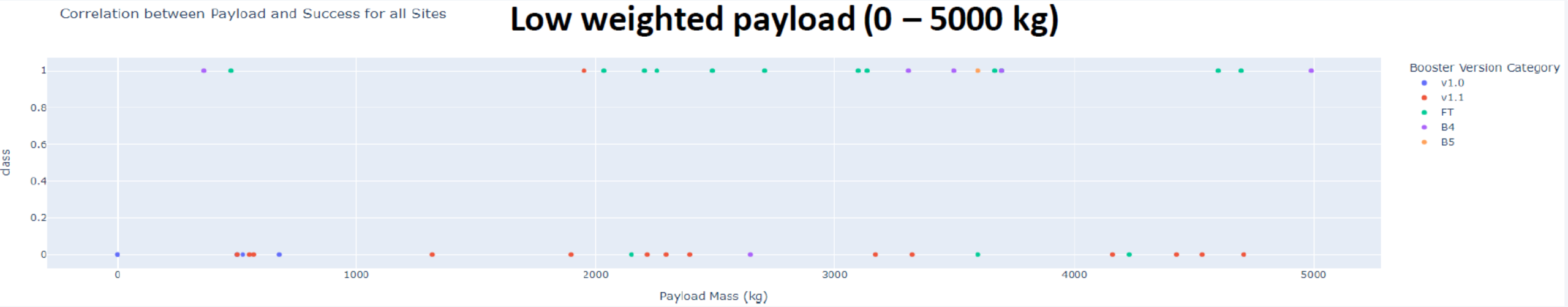
The chart indicates that the KSC LC-39A has the best success rate.

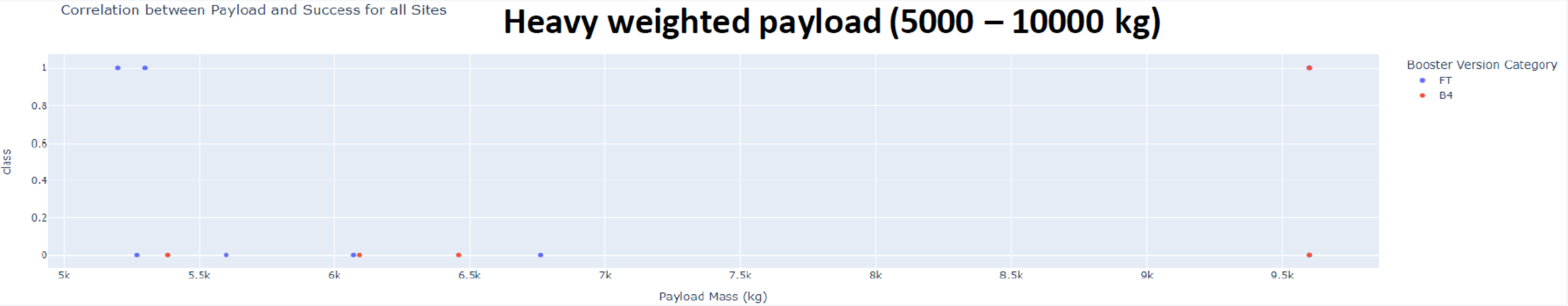
highest launch success ratio

The chart indicates that KSC site has 76.9% success rate and 23.1% failure rate



Payload mass vs launch outcome

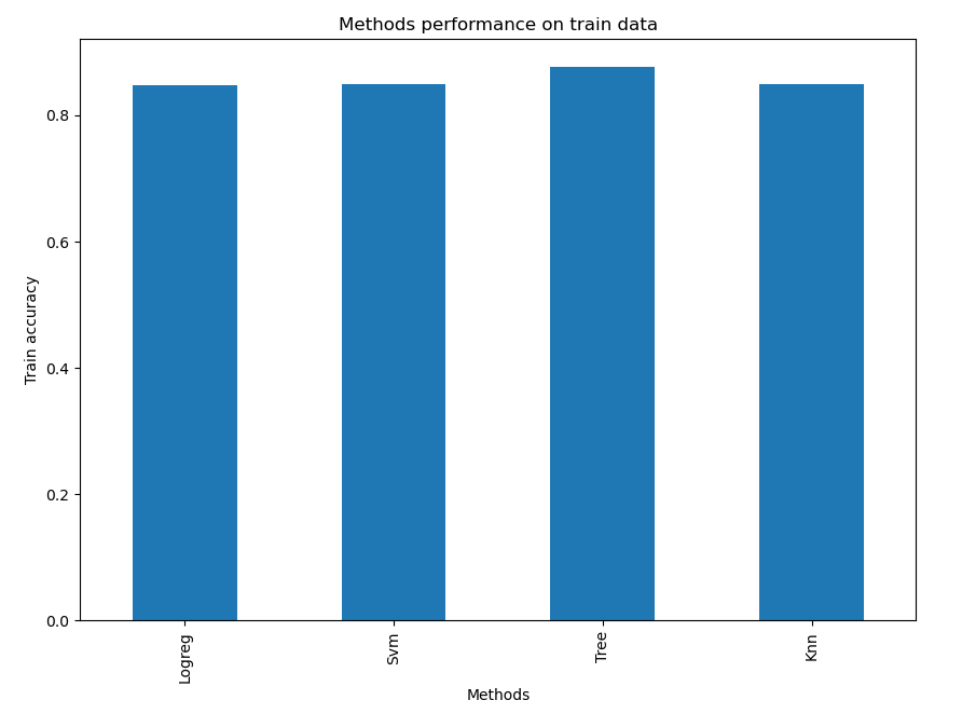




Section 5

Predictive Analysis (Classification)

Classification Accuracy

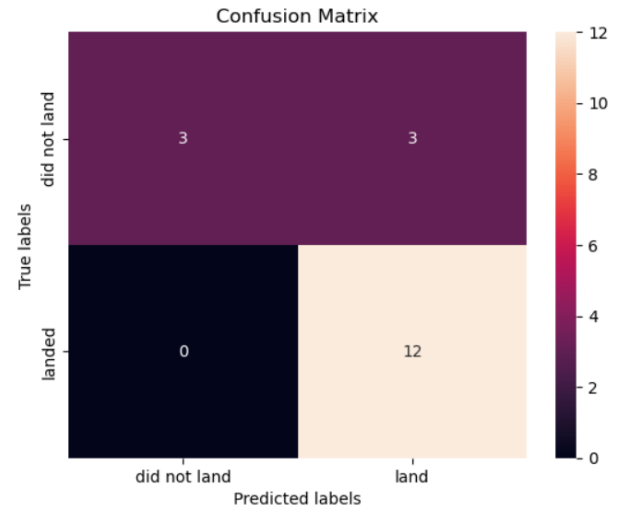


	Accuracy Train	Accuracy Test
Logreg	0.846429	0.833333
Svm	0.848214	0.833333
Tree	0.876786	0.833333
Knn	0.848214	0.833333

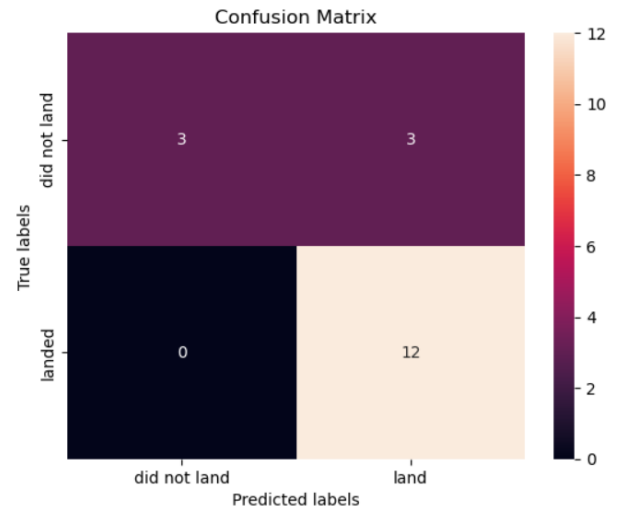
Accuracy of models are very close but decision tree has the most accuracy of all.

Confusion Matrix

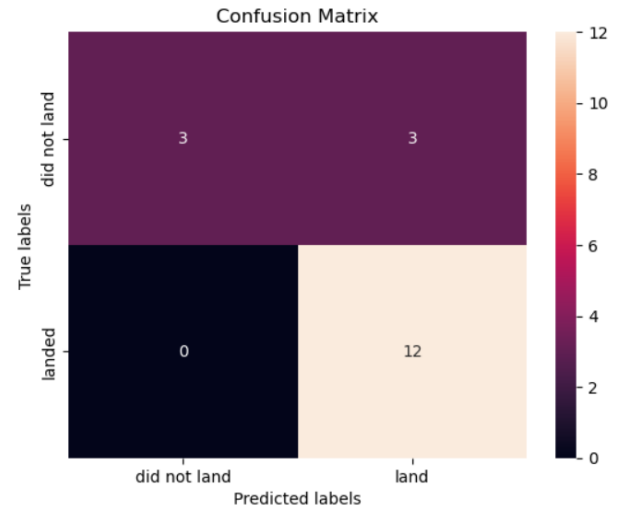
Logistic Regression



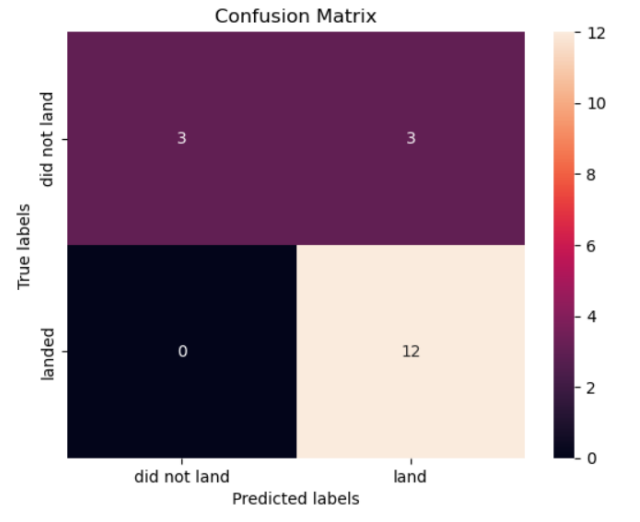
KNN



Decision Tree



SVM



As we can see, accuracy of each model is equal based on their confusion matirx

Conclusions

Orbits with the best success rates are GEO, HEO, SSO, ES-L1

Decision Tree algorithm has the best accuracy among other algorithms. Also other algorithms have very great accuracies.

The success of a mission can be explained by several factors such as the launch site, the orbit and especially the number of previous launches. Indeed, we can assume that there has been a gain in

Thank you!