IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Yaoshen Li>
<15.09.2022>

Outline

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

Executive Summary

Summary of methodologies

-Data Collection (API)

-Data Collection (Web Scraping)

-Data Wrangling

-EDA with SQL

-EDA with Data Visualization

-Launch Sites Locations Analysis 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
  - SpaceX 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 SpaceX 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 SpaceX for a rocket launch. In this project, we need to predict whether the first stage rocket will land successfully
- Problems you want to find answers
  - 啊手动阀手动阀

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Though API and Web Scraping do we collect the data
- Perform data wrangling
  - We dealt with the unmeaningful data through python
- 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.
  - API
    - We get data in form of JSON through API, we turned it into pandad dataframe and normalized it
    - We do the data Wrangling by check the missing value and filling the missing value
  - Web Scraping
    - Get data through tool of BeautifulSoup

Data Collection – SpaceX API

- With SpaceX API we collect data, clean the requested data and did some basic data wrangling and formatting.

- https://github.com/start0036/Applied-Data-Science-Capstone/blob/main/Data%20Collection.ipynb

[16]:
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

[17]:
```
response = requests.get(spacex_url)
```

Check the content of the response

[45]:
```
print(response.content)
```

```
b'[{"fairings":{"reused":false,"recovery_attempt":false,"r
5b/02/QcxHUb5V_o.png"},"reddit":{"campaign":null,"launch":
8","youtube_id":"0a_00nJ_Y88","article":"https://www.space
06-03-17T00:00:00.000Z","static_fire_date_unix":1142553600
ailure"}],"details":"Engine failure at 33 seconds and loss
r":1,"name":"FalconSat","date_utc":"2006-03-24T22:30:00.00
8033d3b2623","flight":1,"gridfins":false,"legs":false,"reu
y_id":null,"id":"5eb87cd9ffd86e000604b32a"},{"fairings":{"
png","large":"https://images2.imgbox.com/80/a2/bkWotCIS_o.
```

calls

Task 2: Filter the dataframe to only include `Falcon 9` launches

Finally we will remove the Falcon 1 launches keeping only the Falcon 9 launches. Filter data to a new dataframe called `data_falcon9`.

[42]:
```
# Hint data['BoosterVersion']!='Falcon 1'
falcon9 = data[data.BoosterVersion == 'Falcon 9']
falcon9
```

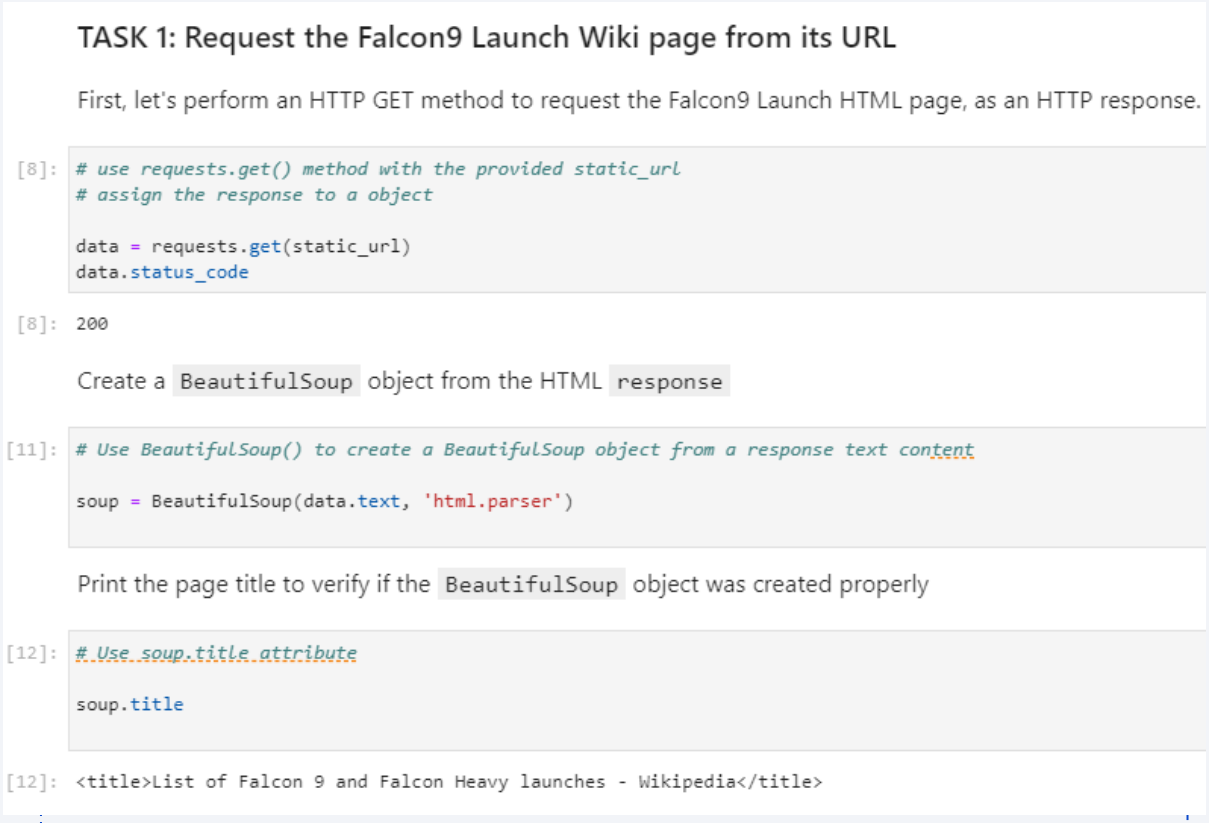
[42]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None
5	8	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None
6	10	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None

Data Collection - Scraping

- With the library of BeautifulSoup we collect the data on website

- https://github.com/start0036/Applied-Data-Science-Capstone/blob/main/Data%20Webscraping.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.

[8]:
```
# use requests.get() method with the provided static_url
# assign the response to a object

data = requests.get(static_url)
data.status_code
```

[8]:
```
200
```

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

[11]:
```
# 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

[12]:
```
# Use soup.title attribute

soup.title
```

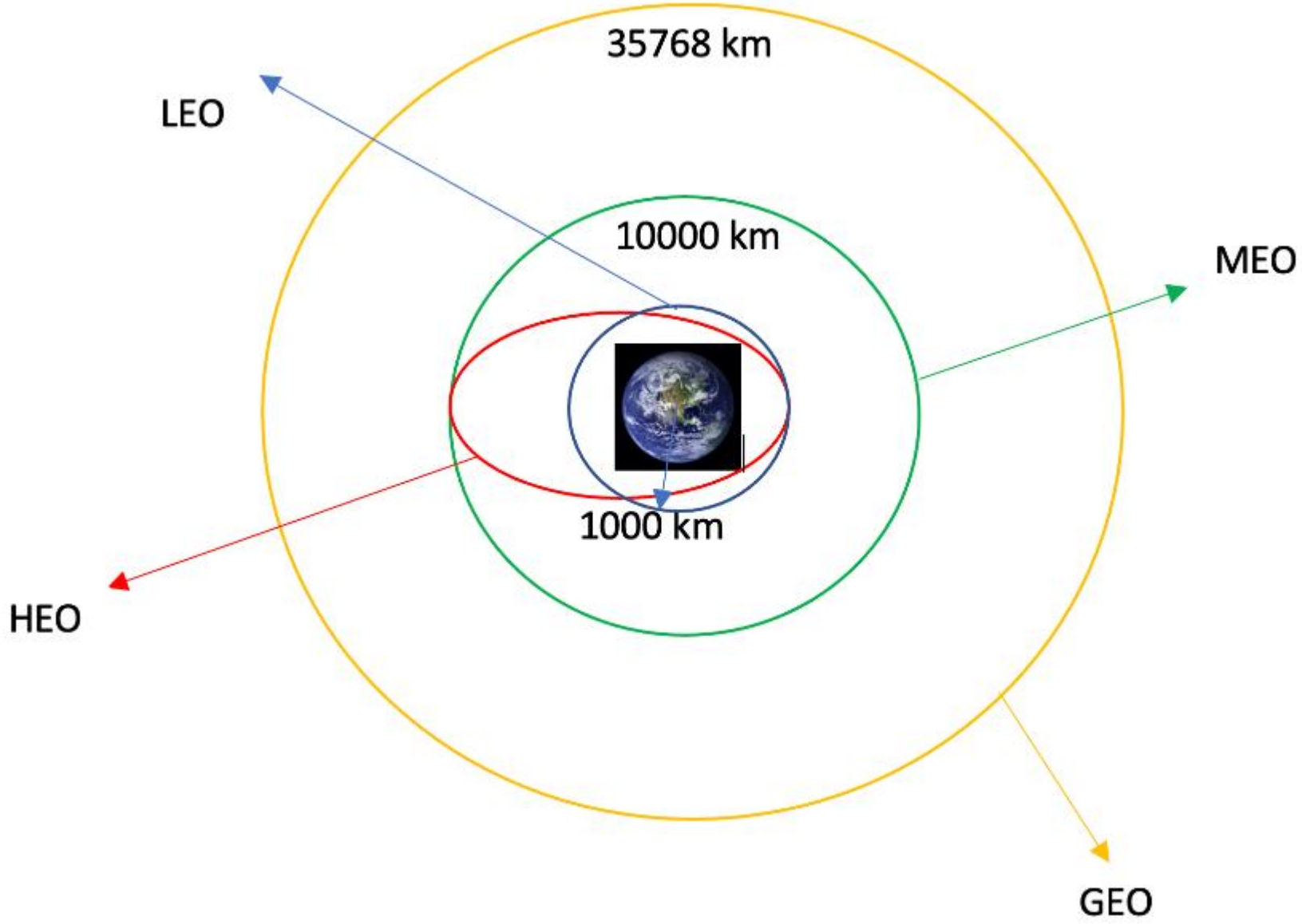
[12]:
```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

Data Wrangling

- Wrangling Process
  - Calculate the number of launches on each site
  - Calculate the number and occurrence of each orbit
  - Calculate the number and occurence of mission outcome per orbit type
  - Create a landing outcome label from Outcome column

- https://github.com/start0036/Applied-Data-Science-Capstone/blob/main/Data%20Wangling.ipynb

Data Wrangling

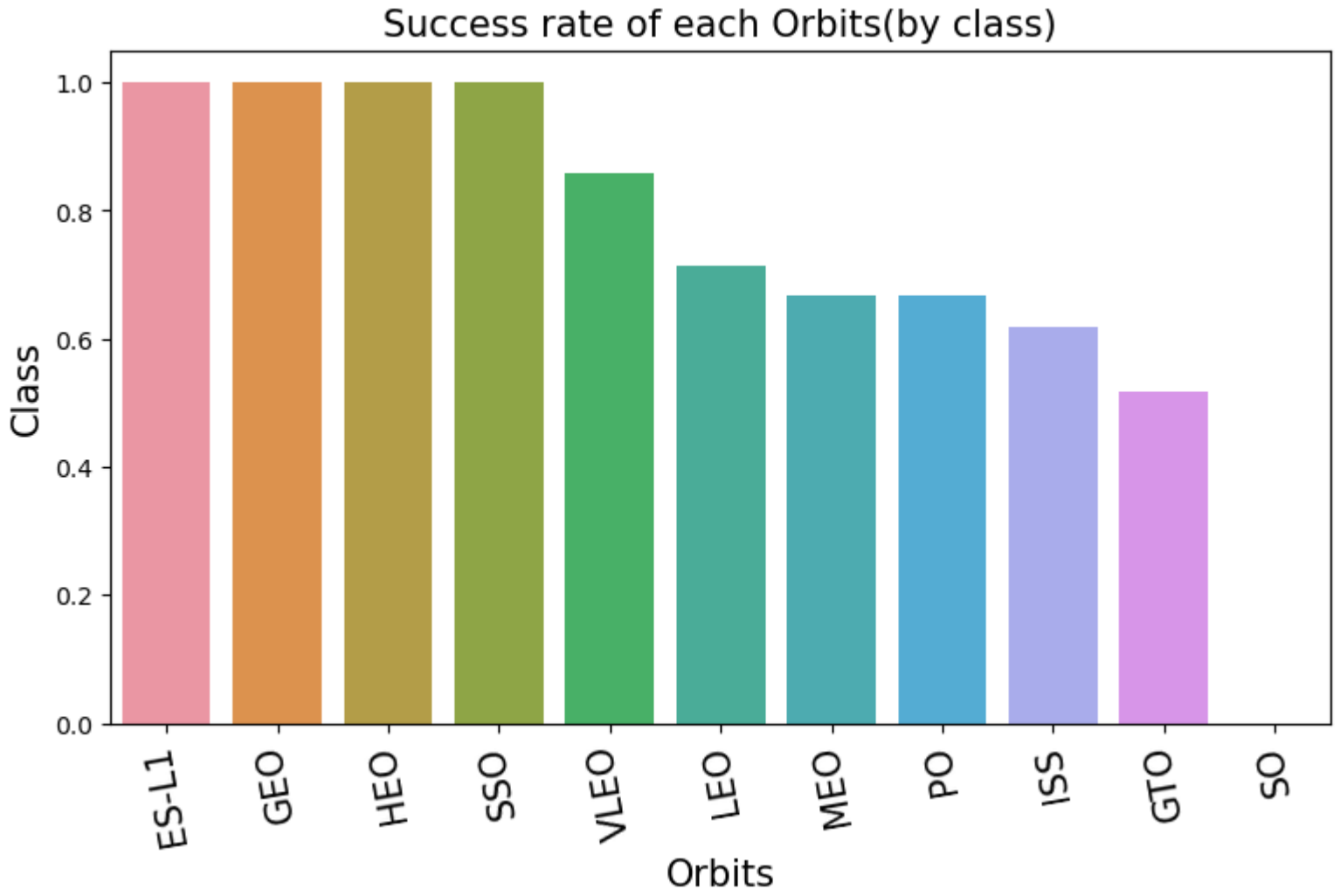


EDA with Data Visualization

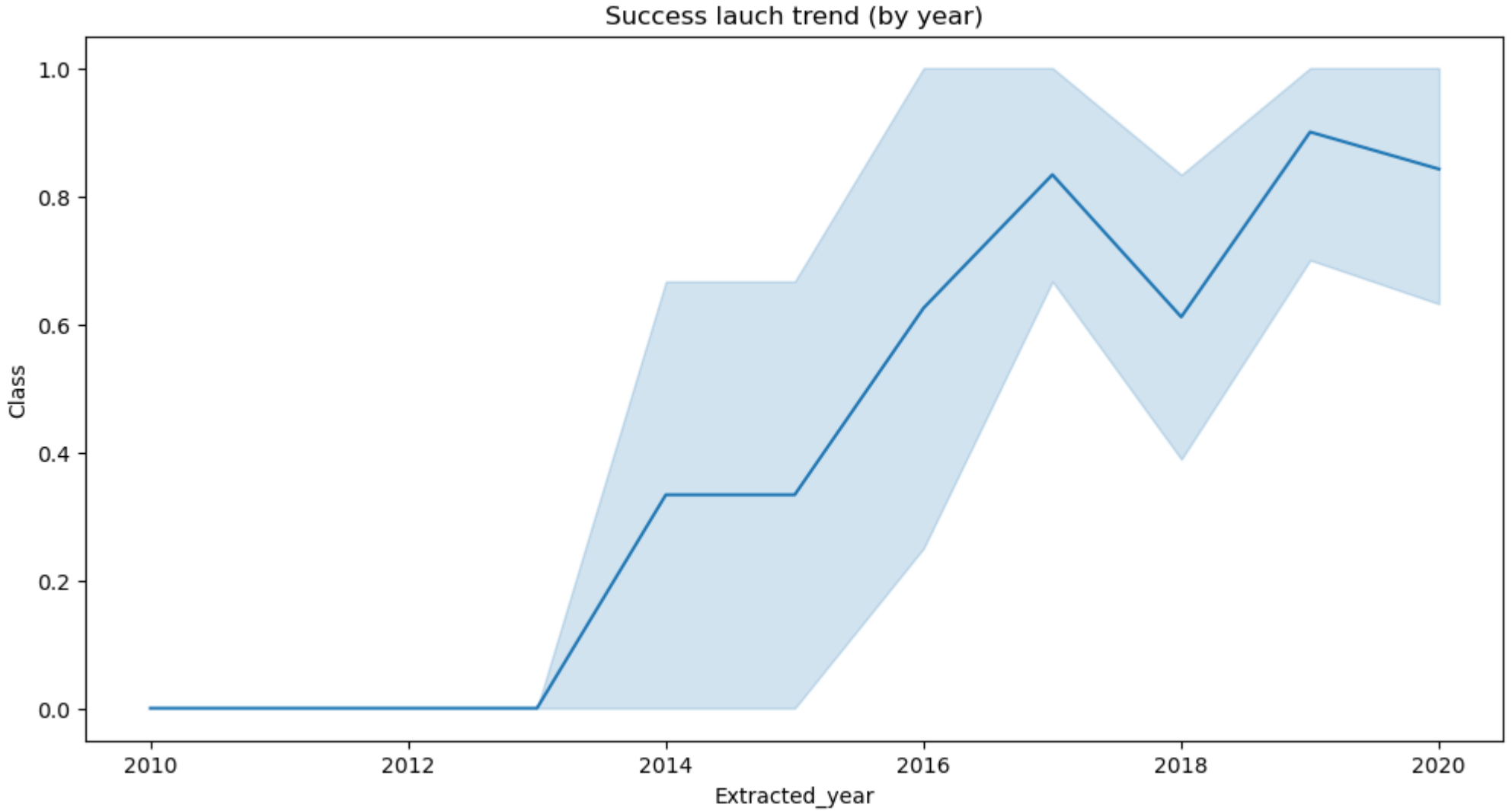
- We did research to find out the relationship among flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend. (Chart see below)

- https://github.com/start0036/Applied-Data-Science-Capstone/blob/main/EDA%20with%20Visualization.ipynb

EDA with Data Visualization



EDA with Data Visualization



EDA with SQL

- We used library sqlalchemy to handle with the data, and we….
  - Display the names of the unique launch sites in the space mission
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - List the date when the first succesful landing outcome in ground pad was acheived.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - List the total number of successful and failure mission outcomes
  - List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - 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.
  - Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.
- https://github.com/start0036/Applied-Data-Science-Capstone/blob/main/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- We point out all the information such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We point out which launch sites has higher successful launch rate.
- We answered questions like:
  - Are all launch sites in proximity to the Equator line?
  - Are all launch sites in very close proximity to the coast?

- https://github.com/start0036/Applied-Data-Science-Capstone/blob/main/Launch%20Sites%20Locations%20Analysis%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- We draw the dashboard with plotly dash and a chart about total launches by a certain sites and revealed the relationship with Outcome and Payload Mass (Kg) for the different booster version

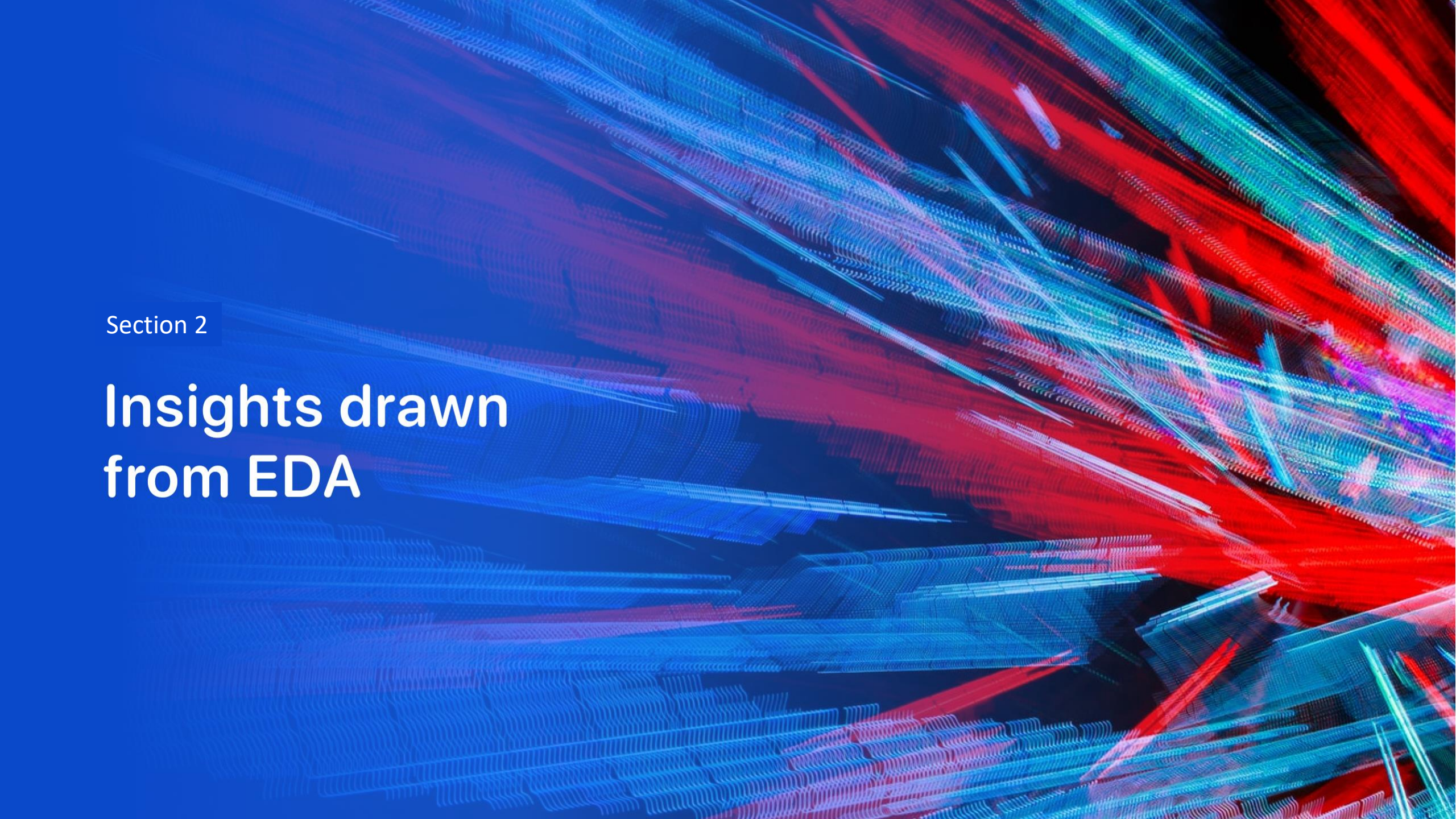
- https://github.com/start0036/Applied-Data-Science-Capstone/blob/main/Launch%20Sites%20Locations%20Analysis%20with%20Folium.ipynb

Predictive Analysis (Classification)

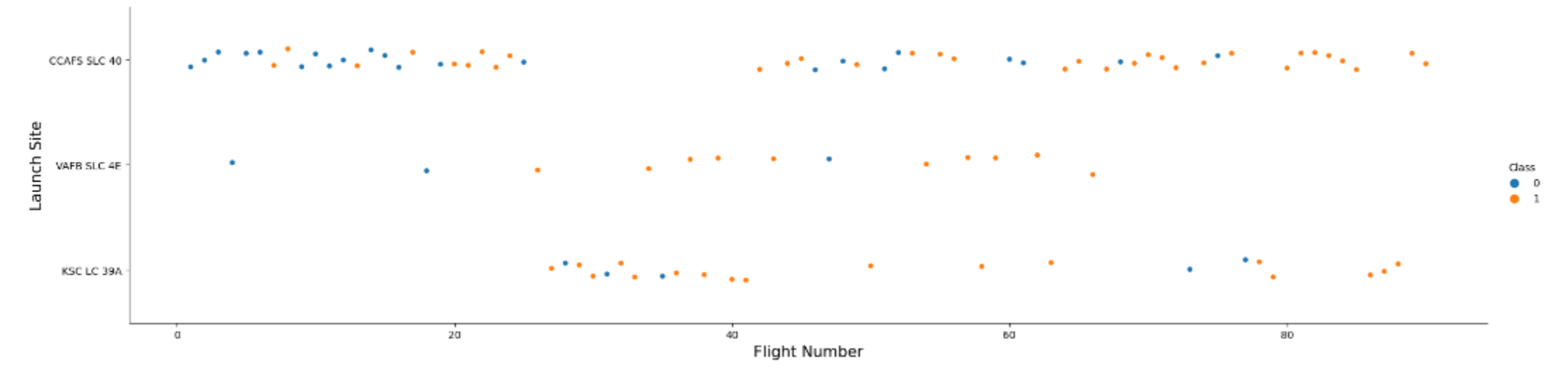
- We use numpy and pandas to deal with the data and divide them into traiding part and test part
- We built a different model
- We calculate the accuracy of each model and chose the best one
- https://github.com/start0036/Applied-Data-Science-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5%20(1).ipynb

Results

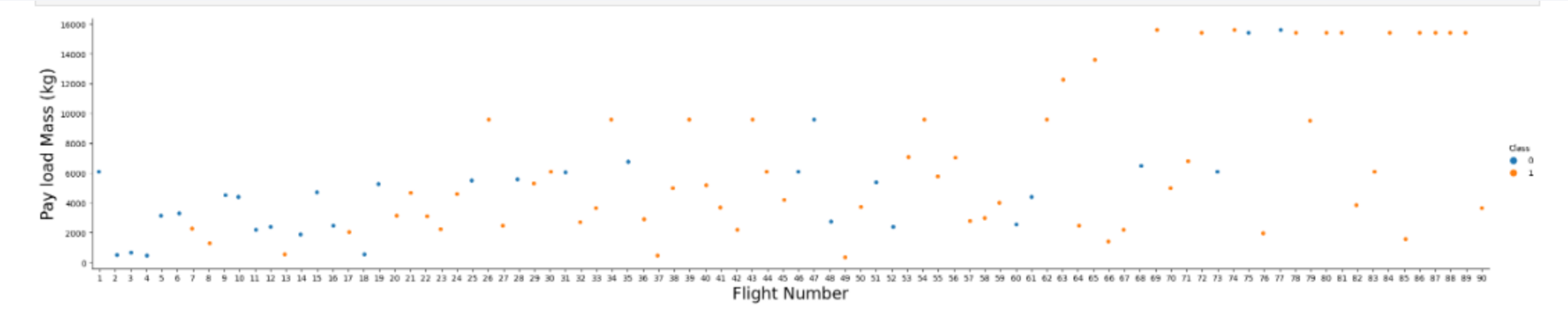
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results


Section 2
Insights drawn from EDA

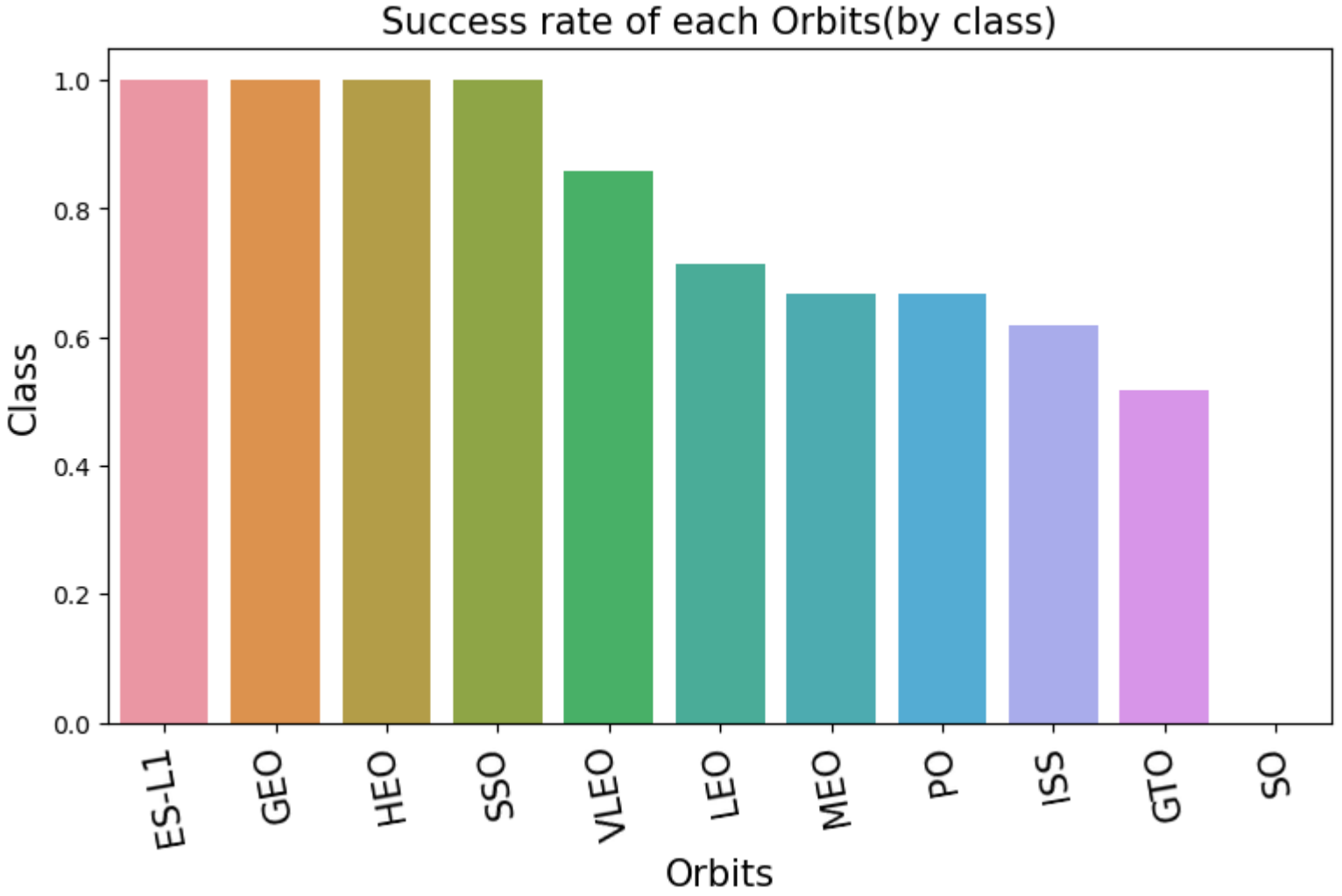
Flight Number vs. Launch Site



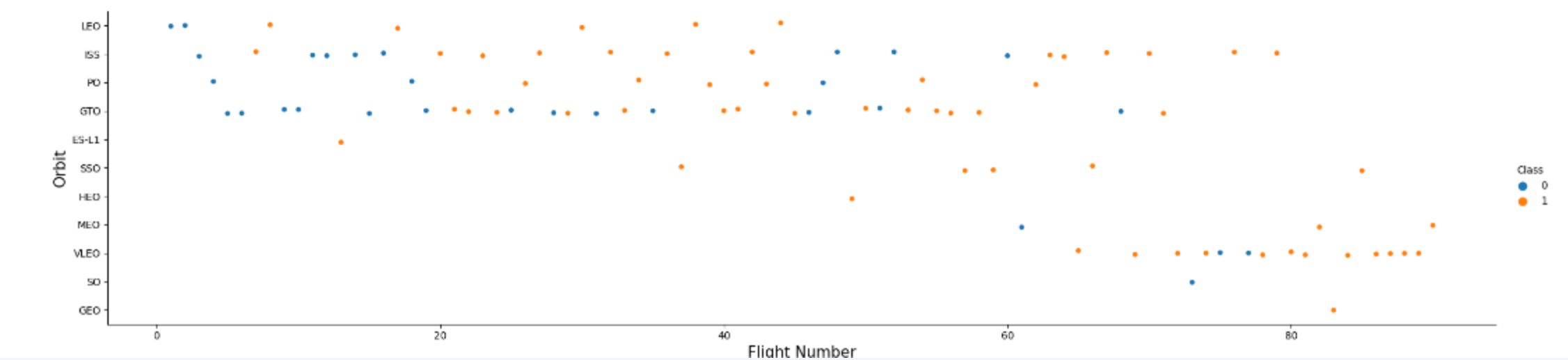
Payload vs. Flight Number



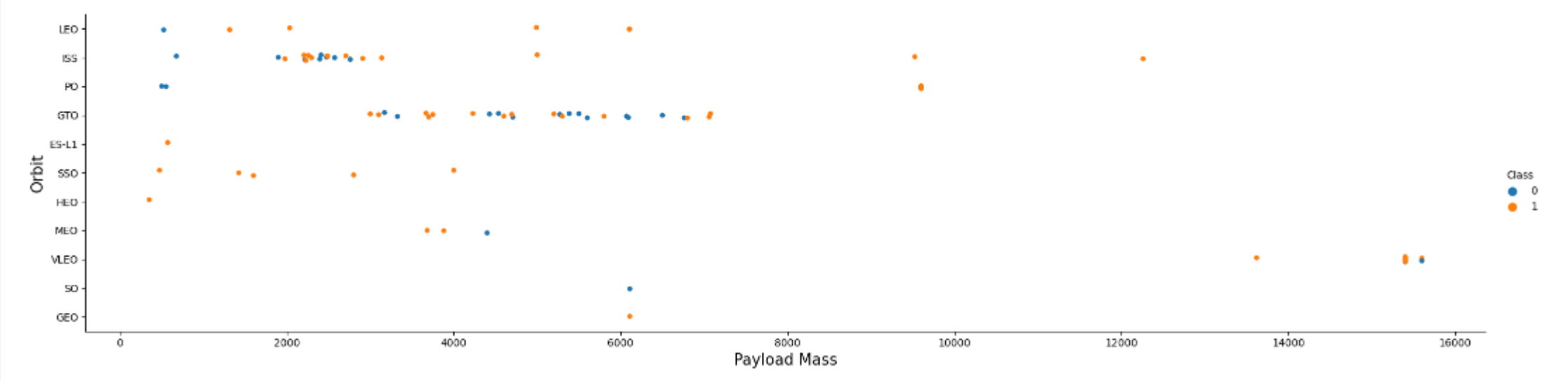
Success Rate vs. Orbit Type



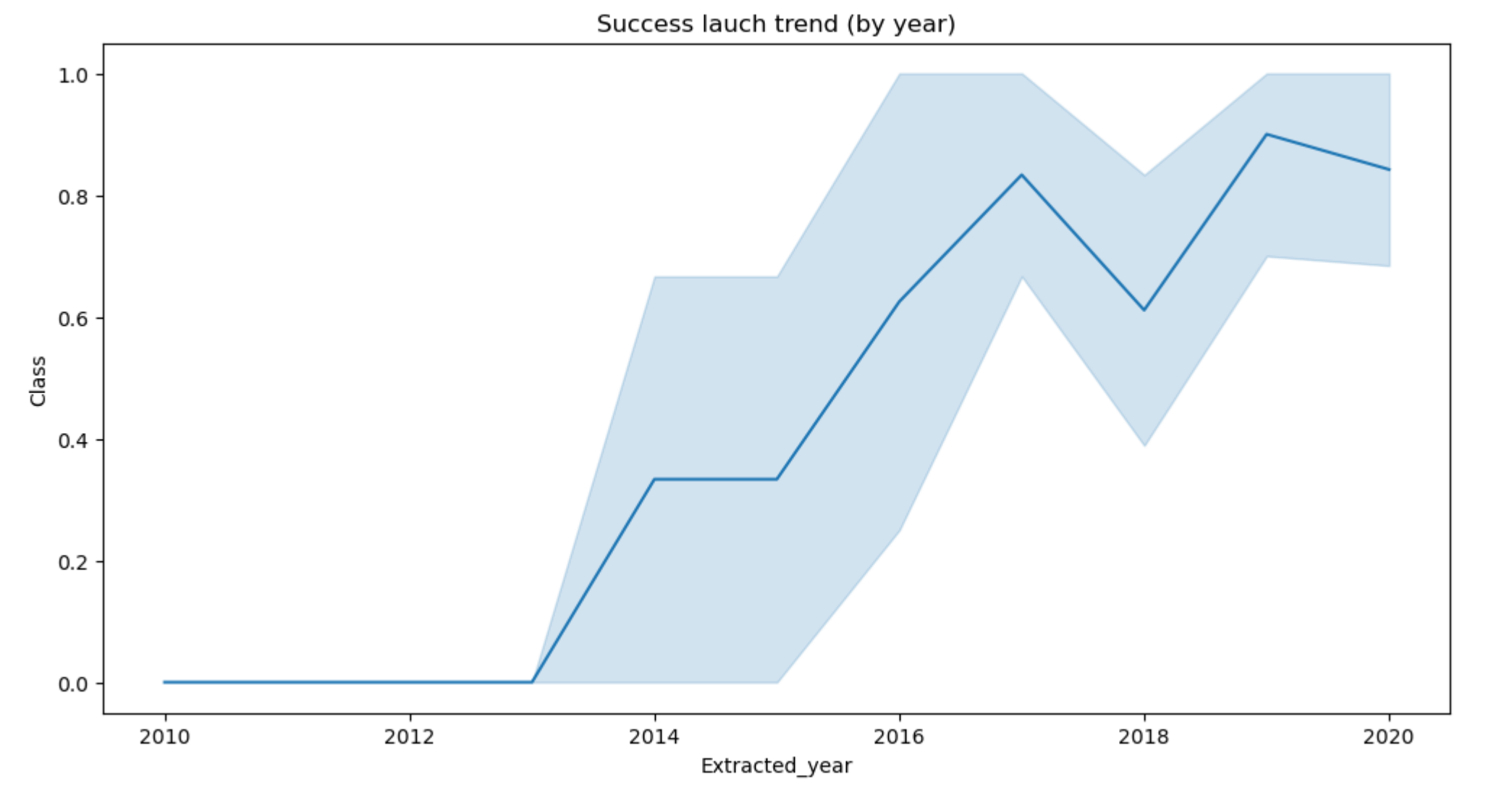
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

In [39]:
```
%sql select distinct launch_site from SPACEXTBL;
```

```
 * ibm_db_sa://cxj24332:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[39]:

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

Launch Site Names Begin with 'CCA'

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

In [44]:
```
%sql SELECT * FROM SPACEXTBL WHERE launch_site LIKE 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://cxj24332:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[44]:

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

Task 3

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

In [46]:
```
%sql SELECT SUM(payload_mass__kg_) AS Total_PayloadMass FROM SPACEXTBL WHERE Customer LIKE 'NASA (CRS)'
```

```
 * ibm_db_sa://cxj24332:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[46]:

total_payloadmass
45596

Average Payload Mass by F9 v1.1

Task 4

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

In [49]:
```
%sql SELECT AVG(payload_mass__kg_) AS Avg_PayloadMass FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1'
```

```
 * ibm_db_sa://cxj24332:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[49]:

avg_payloadmass
2928

Successful Drone Ship Landing with Payload 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

In [52]:
```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE Landing__Outcome = 'Success (drone ship)' AND payload_mass__kg_ > 4000 AND payload_mass__kg_ < 6000
```

```
 * ibm_db_sa://cxj24332:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[52]:

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

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

In [64]:
```
%sql SELECT COUNT (*) FROM SPACEXTBL where Mission_Outcome LIKE 'Success%'
```

```
 * ibm_db_sa://cxj24332:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[64]:

1
100

In [65]:
```
%sql SELECT COUNT (*) FROM SPACEXTBL where Mission_Outcome LIKE 'Failure%%'
```

```
 * ibm_db_sa://cxj24332:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[65]:

1
1

Boosters Carried Maximum Payload

Task 8

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

In [67]:
```
%sql SELECT Booster_Version, payload_mass__kg_ FROM SPACEXTBL WHERE payload_mass__kg_ = ( SELECT MAX(payload_mass__kg_)  FROM SPACEXTBL )  ORDER BY Bo
```

```
 * ibm_db_sa://cxj24332:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

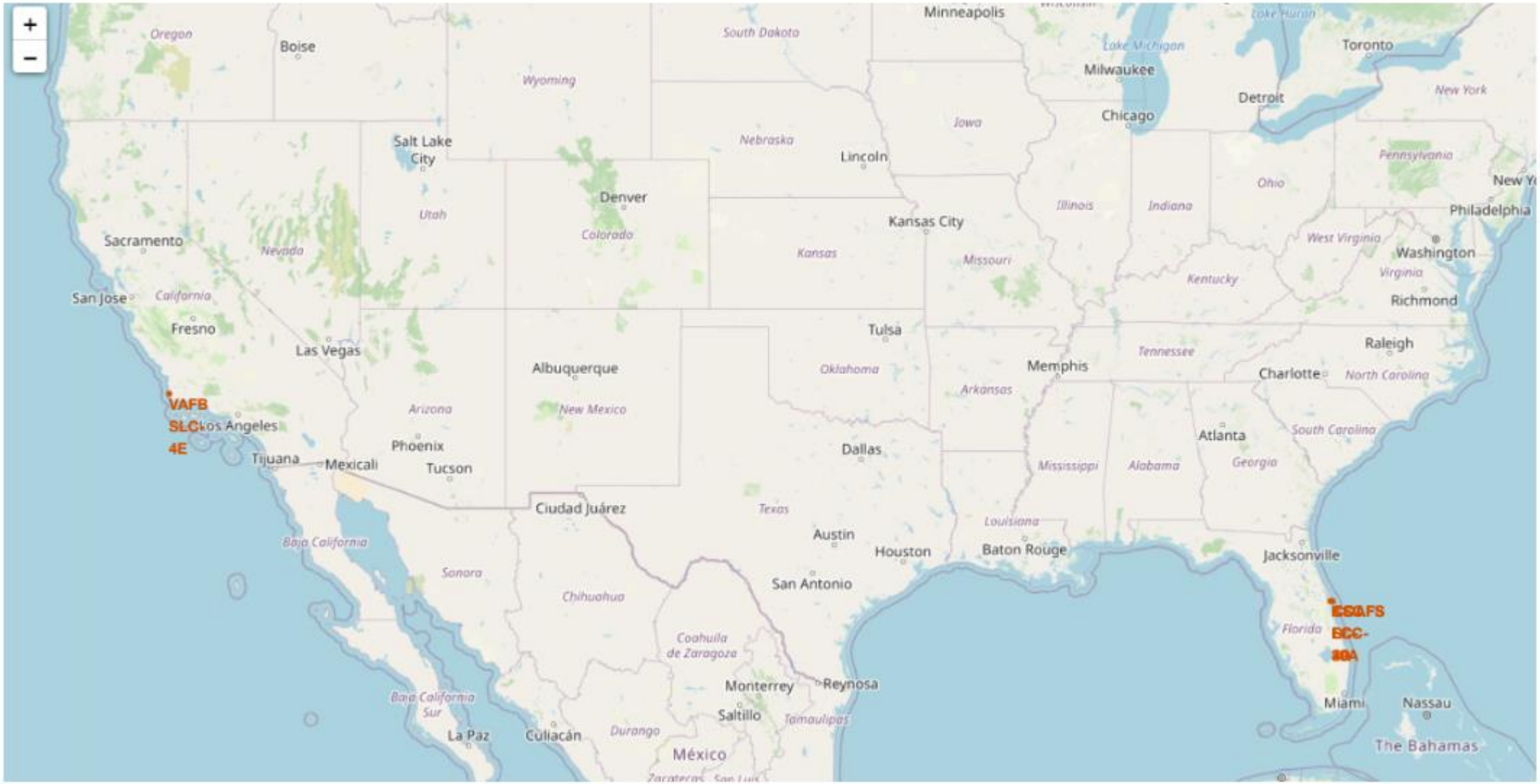
Out[67]:

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

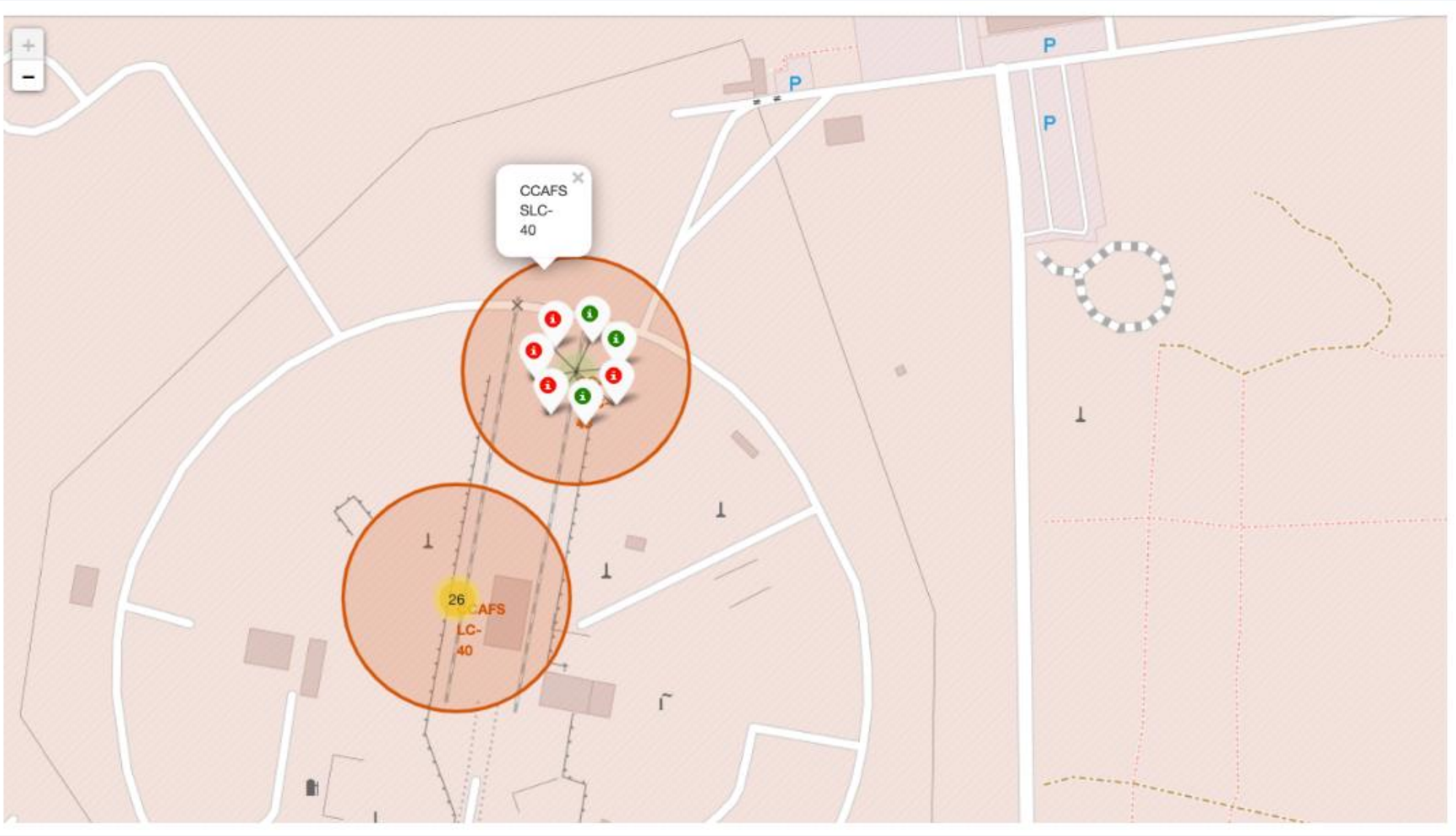
Section 3

Launch Sites Proximities Analysis

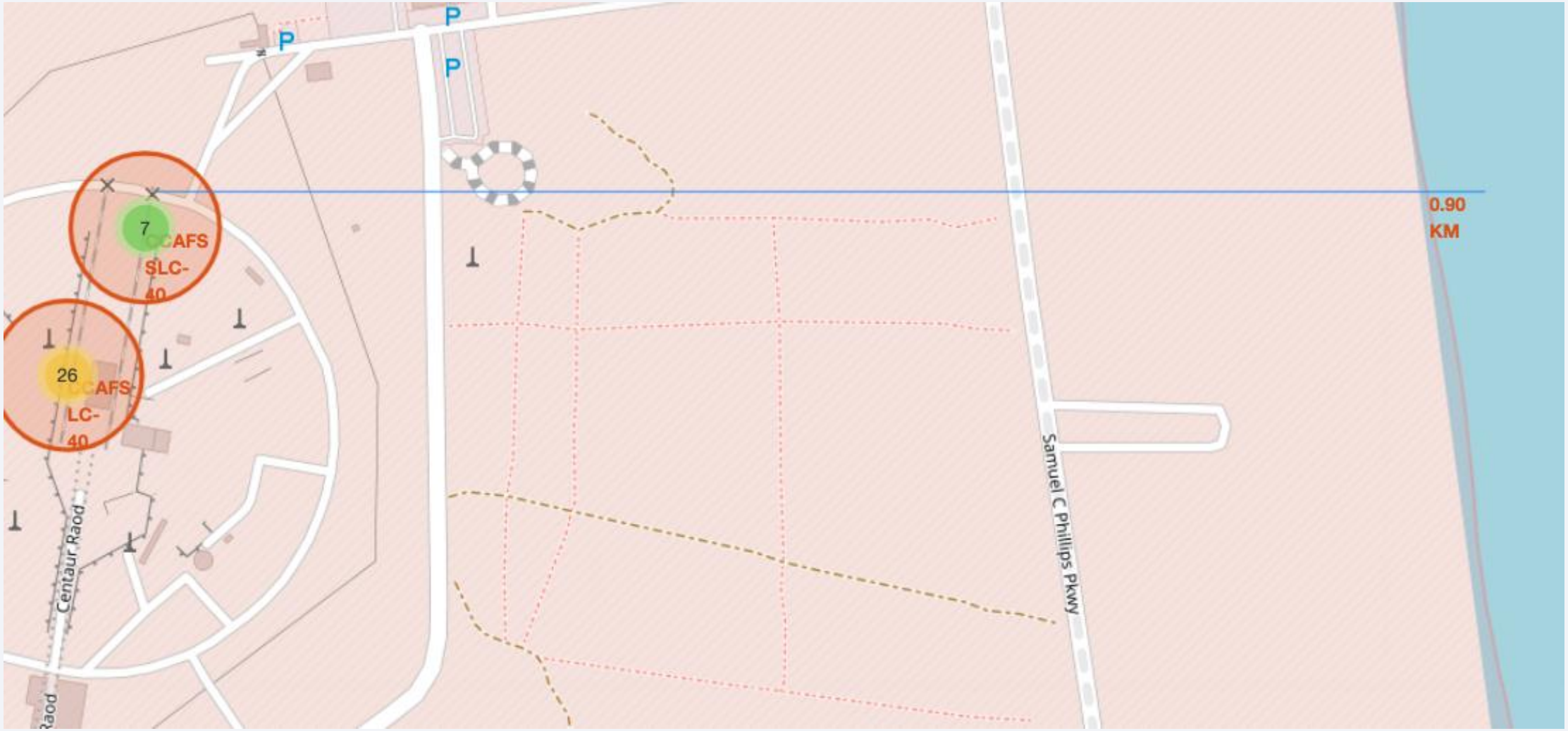
global map markers

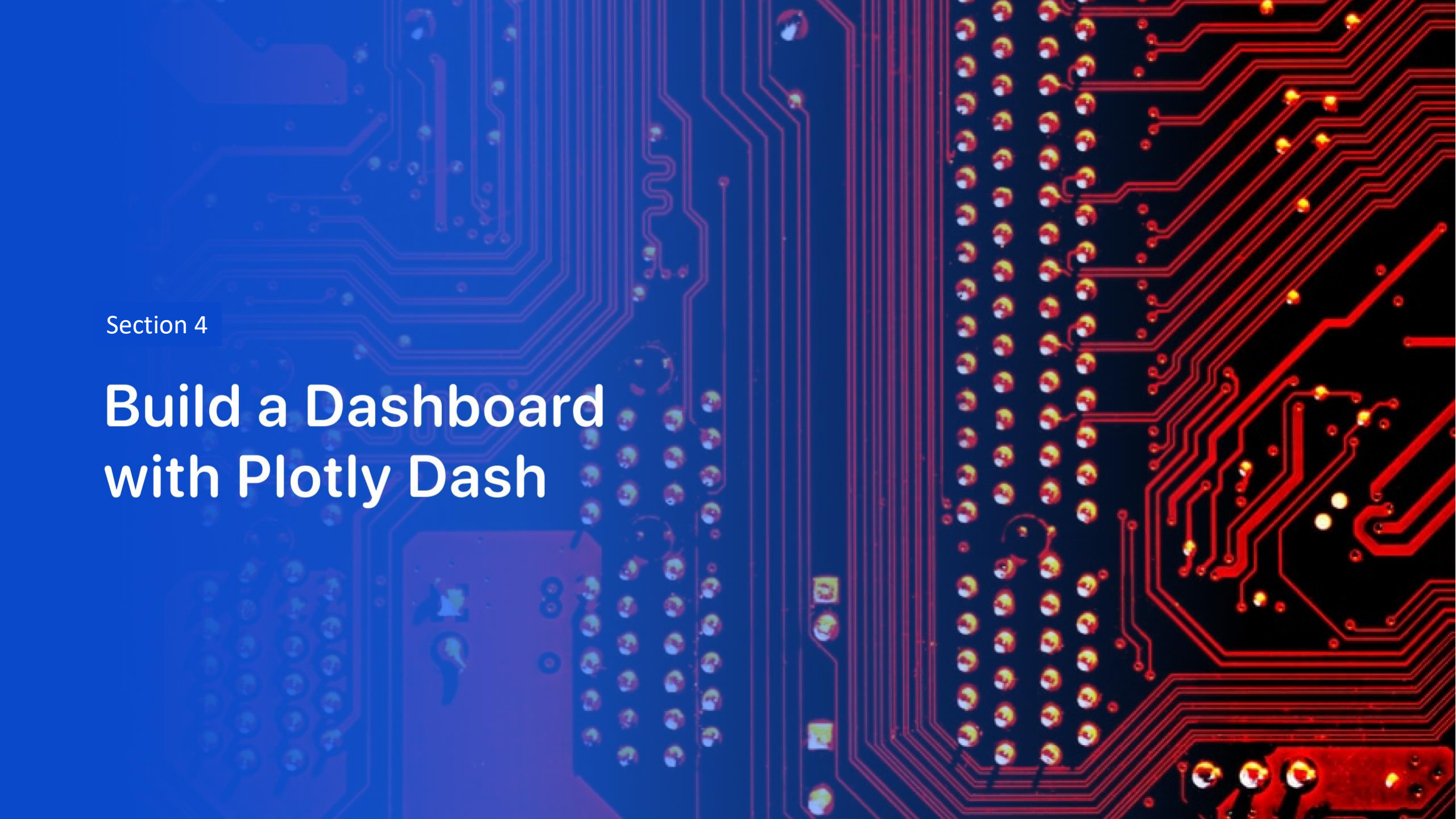


launch sites

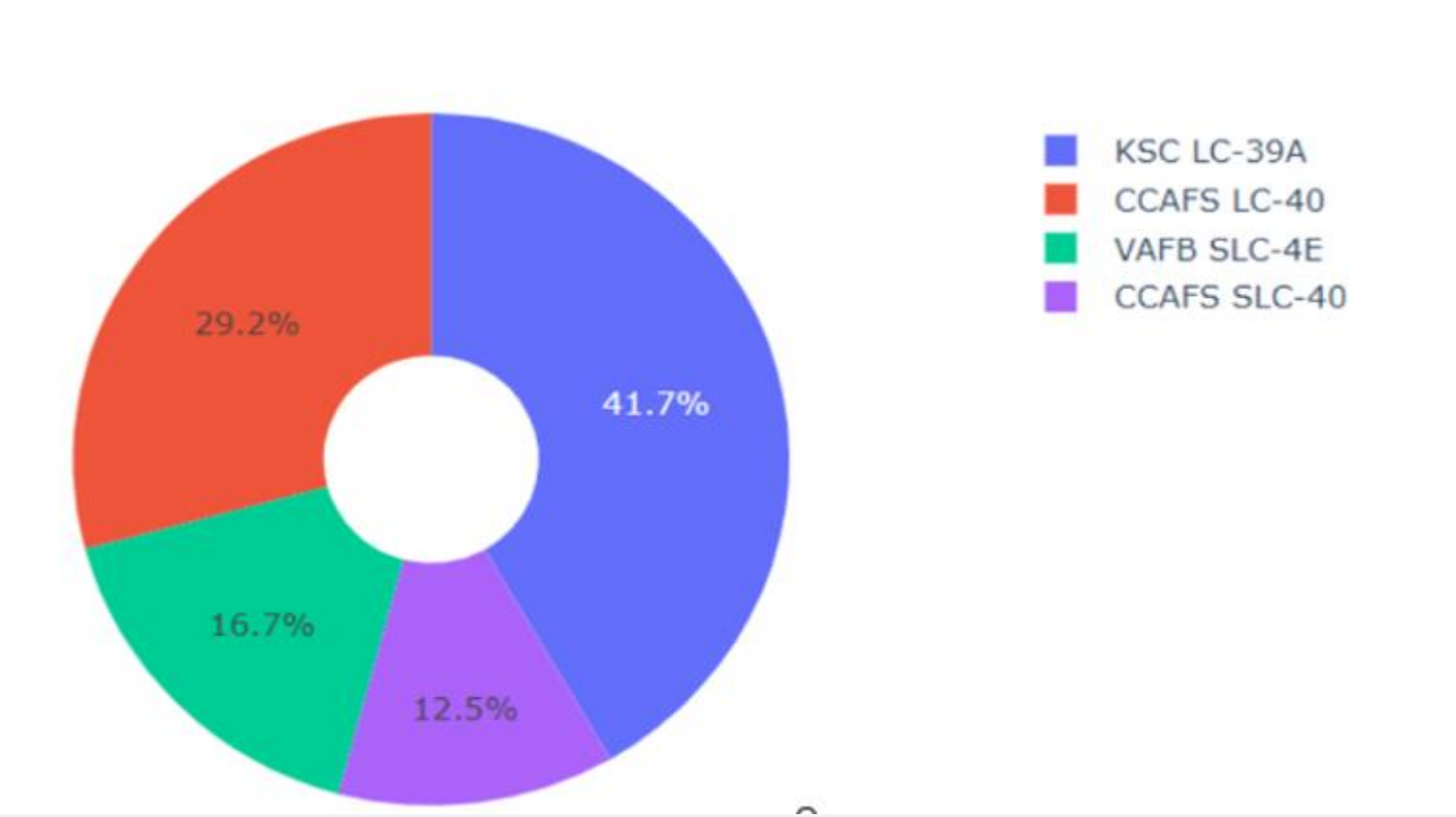


distance to landmarks

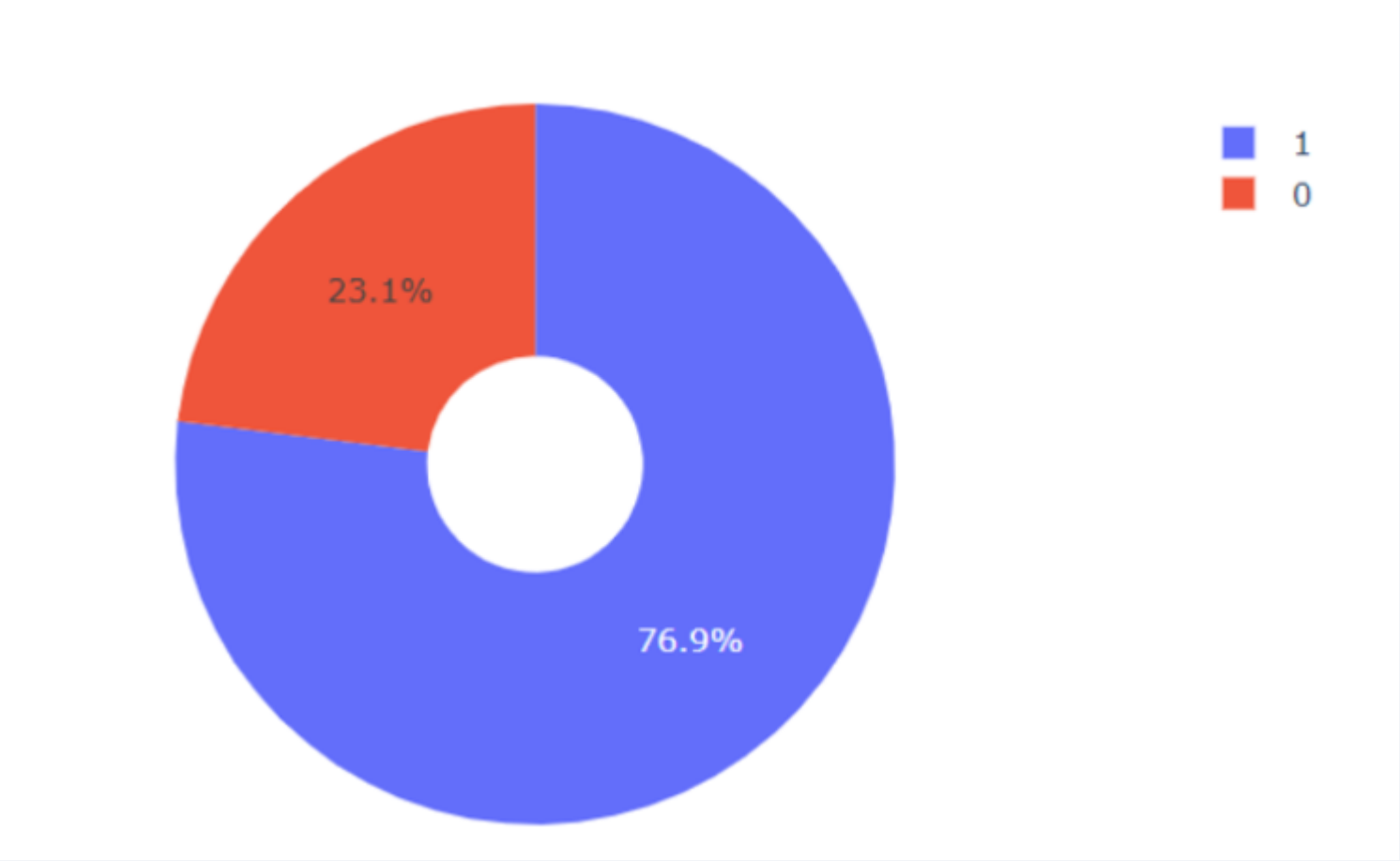



Section 4
Build a Dashboard with Plotly Dash

Success percentage of all sites



Success rate



Section 5

Predictive Analysis (Classification)

Classification Accuracy

Find the method performs best:

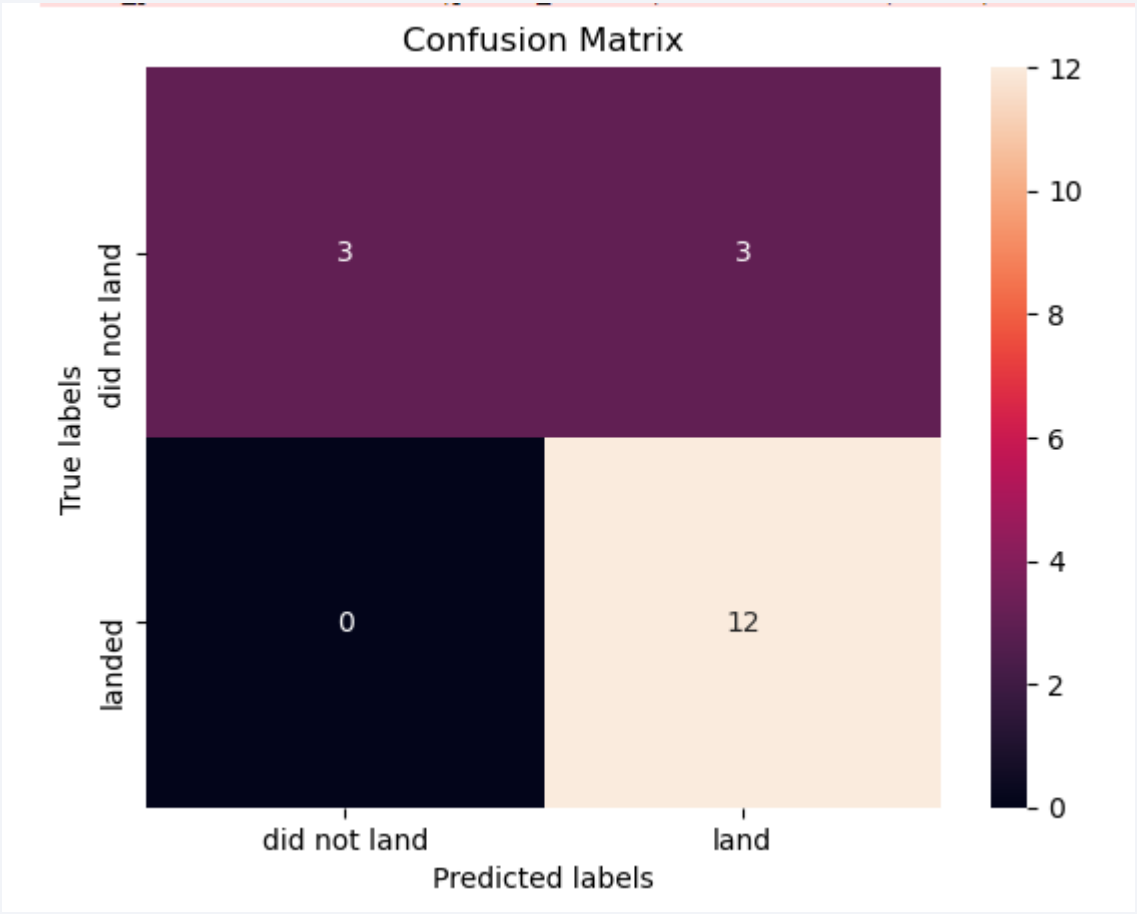
In [162…

```
models = {'KNeighbors':knn_cv.best_score_,
               'DecisionTree':tree_cv.best_score_,
               'LogisticRegression':logreg_cv.best_score_,
               'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
```

```
Best model is DecisionTree with a score of 0.875
```

Confusion Matrix



Conclusions

- Launch success rate increase in 2013 till 2020
- The Decision tree is the best algorithm
- KSC LC-39A is best

Thank you!