IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Thakshila Herath
09/06/2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection through API
  - Data collection with web scraping
  - Exploratory Data Analysis (EDA) with SQL
  - Exploratory Data Analysis (EDA) with visualization
  - Interactive map with Folium
  - Plotly Dashboard results
  - Predictive analysis
- Summary of all results
  - EDA
  - Dashboard & interactive plots
  - Predictive results

Introduction

Project background and context

- SpaceX designs, manufactures and launches advanced rockets and spacecraft.
- SpaceX,
  - sends spacecraft to the international space station
  - provides satellite internet access
  - sends manned mission to space
- Rocket launches are relative inexpensive due to the reason that the stage 1 can be recovered
  - Ex : Falcon9 costs 62 million dollars (other providers cost greater than 165 million dollars each)
- Reason for the reduced cost : They reuse the first stage. So if we can determine if the first stage will land, we can determine the cost of the launch.
- Space Y is competing with Space X

Problems you want to find answers

- We want to find if the first stage can be reused using machine learning

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX launch data was gathered using SpaceX REST API
  - Falcon 9 launch records were collected by webscraping
- 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

- SpaceX launch data was gathered using SpaceX REST API

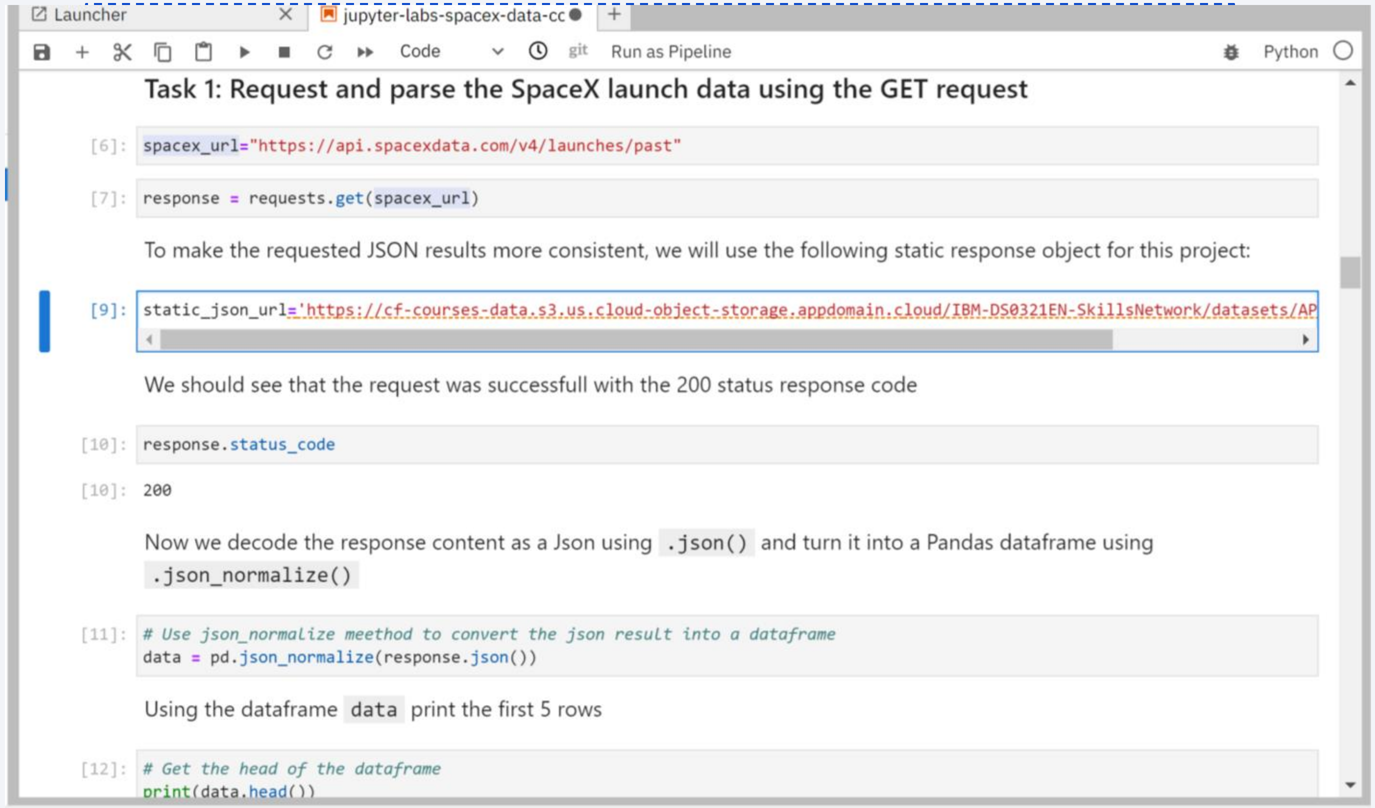
url	• https://api/spacexdata.com/v4/launches/past
get request	• requests.get(url)
Viewing data in json format	• response.json()
Convert json to a dataframe	• pd.json_normalize(response.json())

Data Collection – SpaceX API

- Data collection with SpaceX REST API

- GitHub URL of the completed SpaceX API calls notebook :

https://github.com/hmtmherath85/EdX-Machine-Learning-Capstone/blob/main/SpaceXData.ipynb



Data Collection – Web Scraping Falcon 9 launch data

- Web scraping related wiki pages using BeautifulSoup python package
- The data was parsed from those tables to pandas dataframe
- https://github.com/hmtmherath85/EdX-Machine-Learning-Capstone/blob/main/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.

```python
# use requests.get() method with the provided static_url
# assign the response to a object
web_data = requests.get(static_url)
web_data.status_code
```

```
200
```

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

```python
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
webdata_soup = BeautifulSoup(web_data.text, 'html.parser')
```

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

```python
# Use soup.title attribute
webdata_soup.title
```

```
<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

```python
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = webdata_soup.find_all('table')
```

Data Wrangling

- Wrangling data using an API – need to use specific API targeting the missing data : getBoosterVersion, getLaunchSite, getPlayloadData, getCoreData
- Sampling Data – need to filter out data only for Falcon 9
- Dealing with Nulls – replacing nulls with the mean of the variable
- we will perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models
- we will mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.
- GitHub URL :

https://github.com/hmtmherath85/EdX-Machine-Learning-Capstone/blob/main/EDAlab.ipynb

EDA with Data Visualization

- We explored data by plotting flight number vs launch site, payload vs launch site, success rate of each orbit type and launch success trend
- GitHub URL :

https://github.com/hmtmherath85/EdX-Machine-Learning-Capstone/blob/main/Correct_EDA_Visualization.ipynb

EDA with SQL

- We performed the SQL queries to find out,
  - the names of the unique launch sites in the space mission
  - 5 records where launch sites begin with the string 'KSC'
  - the total payload mass carried by boosters launched by NASA (CRS)
  - the average payload mass carried by booster version F9 v1.1
  - the list the date where the first succesful landing outcome in drone ship was acheived
  - the list the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
- GitHub URL :

https://github.com/hmtmherath85/EdX-Machine-Learning-Capstone/blob/main/EDA_SQL.ipynb

Build an Interactive Map with Folium

- We added map objects such as markers, circles, lines, etc. to folium maps to,
  - Mark all launch sites on a map
  - Mark the success/failed launches for each site on the map
  - Calculate the distances between a launch site to its proximities
- We find some geographical patterns about launch sites.

- GitHub URL :

https://github.com/hmtmherath85/EdX-Machine-Learning-Capstone/blob/main/Dashboard%20with%20Folium%20(1).ipynb

Some plots may not see in github. But they should be visible in cloud pak. Here is the link for IBM cloud pak file

https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/6d598143-48bb-4388-b8e4-4ec556867182/view?access_token=e0a64303c76b640f72de9dfa38f6a48c8e880bf6d4583e6482ee164de0648d6f

Build a Dashboard with plotly Dash

- We created an interactive dashboard with plotly Dash that includes pie chart and etc.

- GitHub URL :

https://github.com/hmtmherath85/EdX-Machine-Learning-Capstone/blob/main/EdX_DashTutorial.ipynb

Predictive Analysis (Classification)

- Perform exploratory Data Analysis and determine Training Labels
- Create a column for the class
- Standardize the data
- Split into training data and test data
- Find best Hyperparameter for SVM, Classification Trees and Logistic Regression
- Find the method performs best using test data

- GitHub URL :

https://github.com/hmtmherath85/EdX-Machine-Learning-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

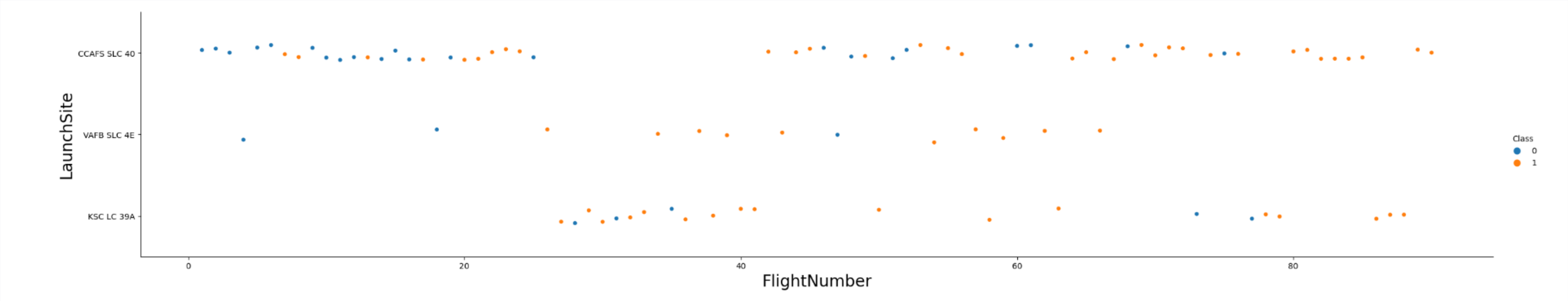
Results

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

Section 2

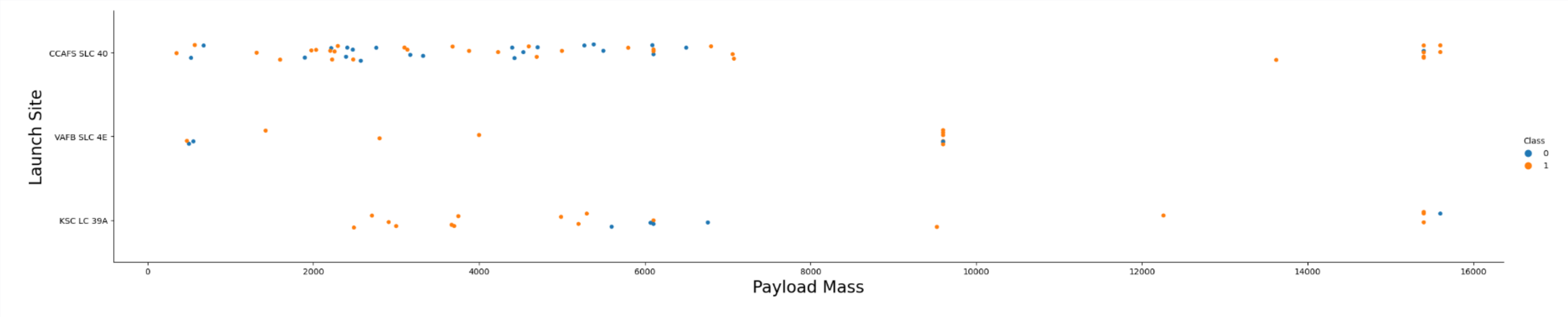
Insights drawn from EDA

Flight Number vs. Launch Site



- We see that the success rate has increased for all launch sites as the flight number increases.

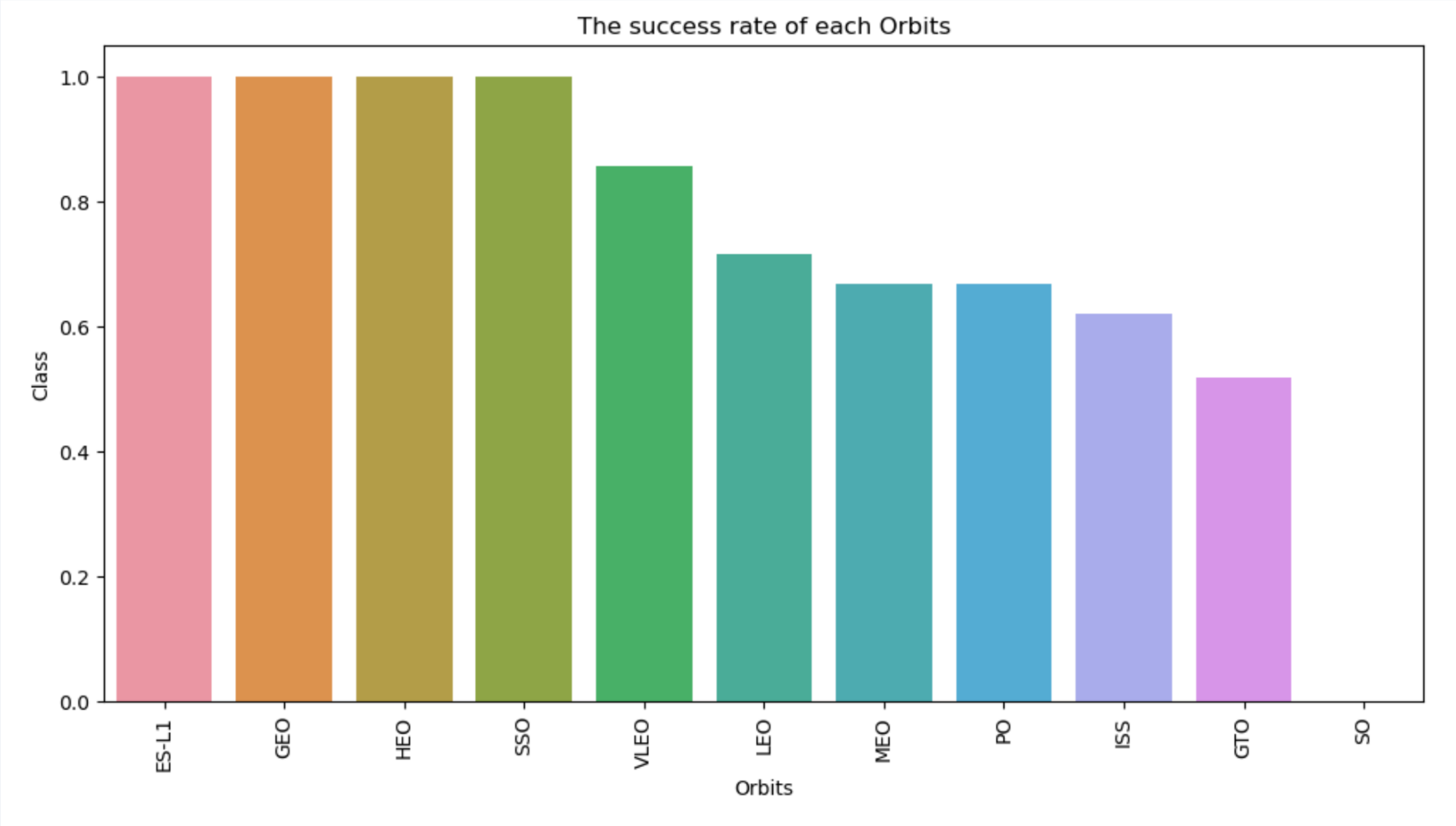
Payload vs. Launch Site



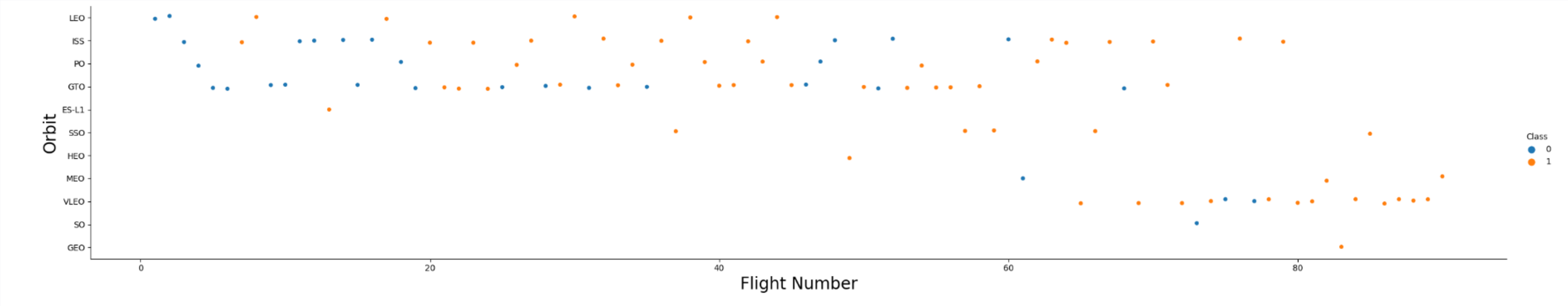
- Launch site KSC LC 39A has more success rate with small payload mass.
- there are no rockets launched for heavy payload mass(greater than 10000) at VAFB-SLC launchsite
- Most of the sites which launch rockets with heavy payload mass have higher success rates

Success Rate vs. Orbit Type

- ES-L1, GEO, HEO and SSO show great success rate of each orbits

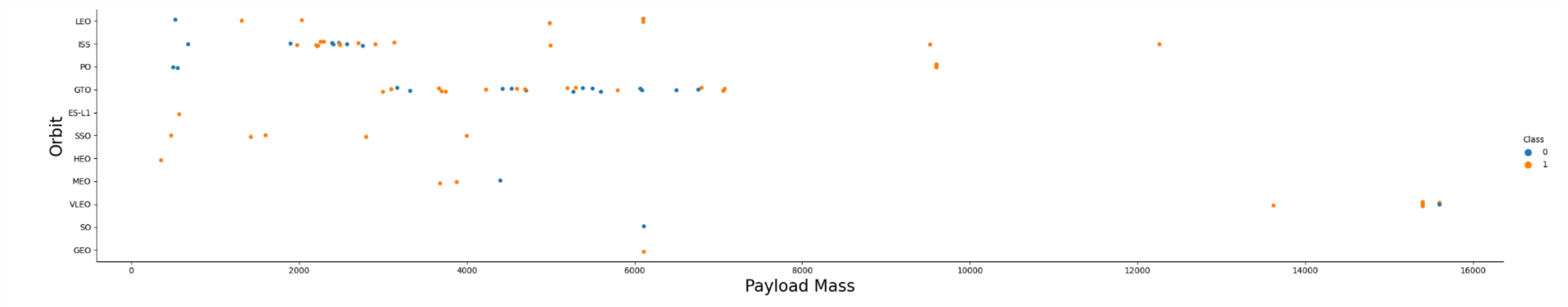


Flight Number vs. Orbit Type



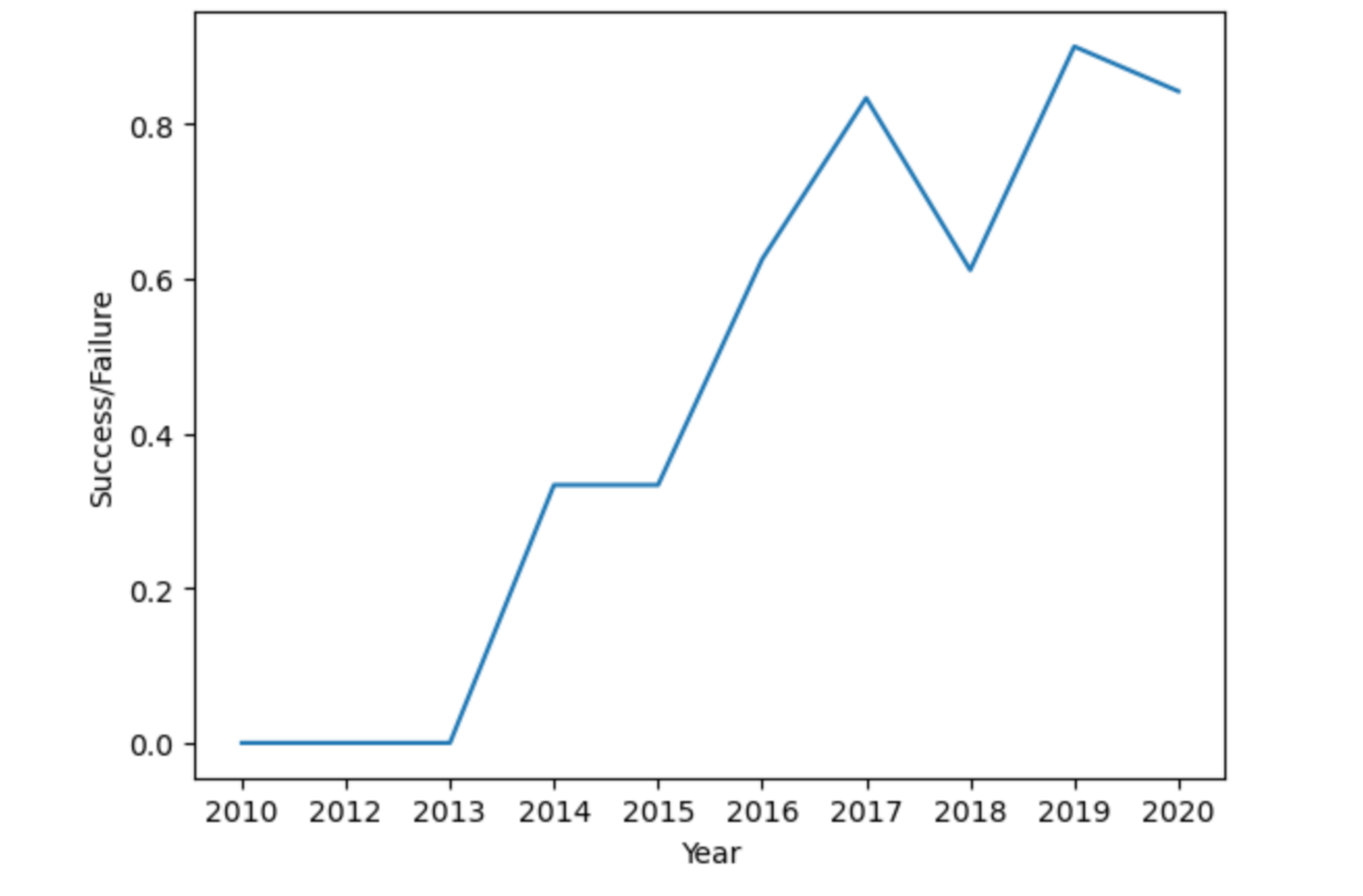
- You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend



- the sucess rate keep constant since 2013 and kept increasing till 2020

All Launch Site Names

Select DISTINCT is used to find the list of unique launch site names

In [20]:
```
%sql SELECT DISTINCT launch_site from SPACEXTBL
```

```
 * ibm_db_sa://tzg42824:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[20]:

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

Launch Site Names Begin with 'KSC'

- "LIKE 'KSC%'" is used to find the names start with `KSC`
- "LIMIT 5" is used to get only 5 records

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

In [21]:
```
%sql SELECT launch_site from SPACEXTBL where launch_site LIKE'KSC%' LIMIT 5
```

```
 * ibm_db_sa://tzg42824:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

Out[21]:

launch_site
KSC LC-39A
KSC LC-39A
KSC LC-39A
KSC LC-39A
KSC LC-39A

Total Payload Mass

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

```
%sql SELECT SUM(payload_mass__kg_) from SPACEXTBL where customer LIKE 'NASA%'
```

```
 * ibm_db_sa://tzg42824:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases
Done.
```

1
99980

Average Payload Mass by F9 v1.1

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

```
%sql SELECT AVG(payload_mass__kg_) from SPACEXTBL where booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://tzg42824:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.
Done.
```

1
2928

First Successful Ground Landing Date

- The dates of the first successful landing outcome on drone ship.

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

Hint:Use min function

```
%sql SELECT MIN(Date) from SPACEXTBL where landing__outcome= 'Success (drone ship)'
```

```
 * ibm_db_sa://tzg42824:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.da
Done.
```

1
2016-04-08

Successful Drone Ship Landing with Payload between 4000 and 6000

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

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

```
%sql SELECT booster_version,payload_mass__kg_ from SPACEXTBL where landing__outcome='Success (ground pad)'
```

```
 * ibm_db_sa://tzg42824:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/b
Done.
```

booster_version	payload_mass__kg_
F9 FT B1019	2034
F9 FT B1025.1	2257
F9 FT B1031.1	2490
F9 FT B1032.1	5300
F9 FT B1035.1	2708
F9 B4 B1039.1	3310
F9 B4 B1040.1	4990
F9 FT B1035.2	2205
F9 B4 B1043.1	5000

Total Number of Successful and Failure Mission Outcomes

- Total number of successful and failure mission outcomes

List the total number of successful and failure mission outcomes

```
%sql SELECT COUNT(mission_outcome) from SPACEXTBL WHERE mission_outcome='Success'
```

```
 * ibm_db_sa://tzg42824:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde0
Done.
```

1
99

Boosters Carried Maximum Payload

- The names of the booster which have carried the maximum payload mass

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

```
#%sql SELECT MAX(payload_mass__kg_) as max_mass from SPACEXTBL
%sql SELECT booster_version from SPACEXTBL WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) as max_mass from SPACEXTBL)
```

```
 * ibm_db_sa://tzg42824:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
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

2015 Launch Records

- 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

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

```
%sql SELECT MONTHNAME(DATE),booster_version,launch_site from SPACEXTBL WHERE landing__outcome = 'Success (ground pad)' AND YEAR(DATE)=2017
```

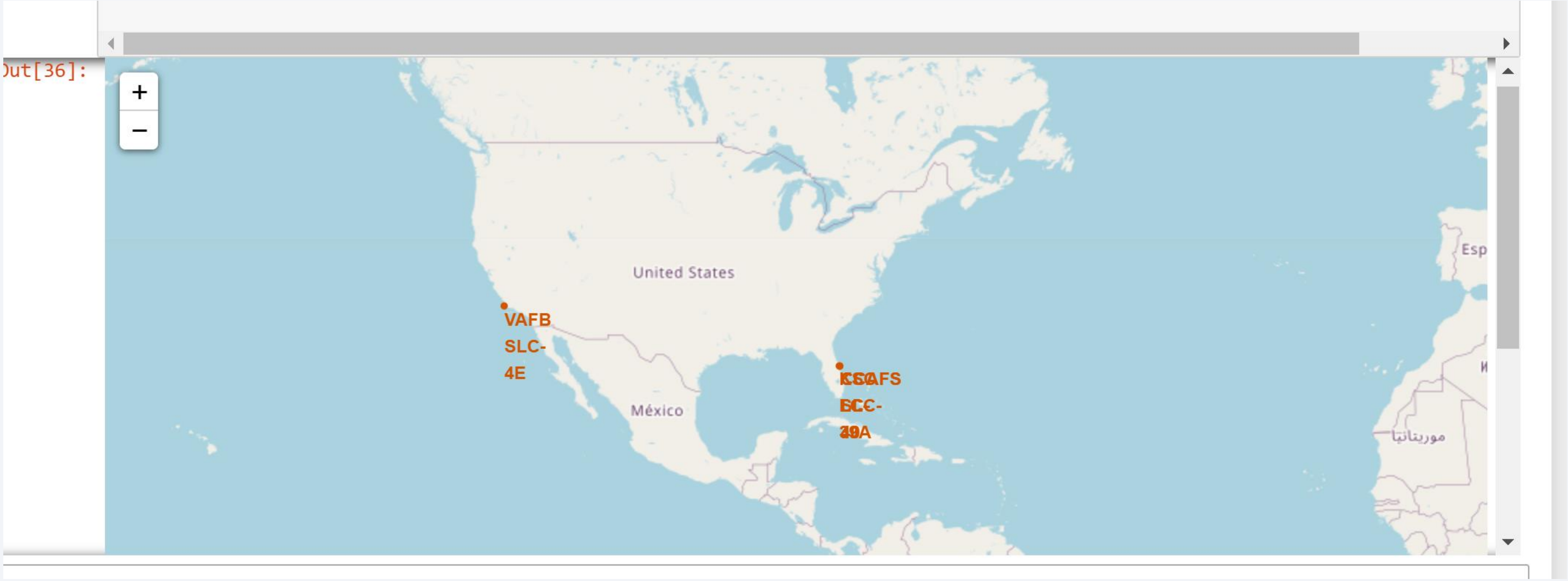
```
 * ibm_db_sa://tzg42824:***@6667d8e9-9d4d-4ccb-ba32-21da3bb5aafc.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30376/bludb
Done.
```

1	booster_version	launch_site
February	F9 FT B1031.1	KSC LC-39A
May	F9 FT B1032.1	KSC LC-39A
June	F9 FT B1035.1	KSC LC-39A
August	F9 B4 B1039.1	KSC LC-39A
September	F9 B4 B1040.1	KSC LC-39A
December	F9 FT B1035.2	CCAFS SLC-40

Section 3

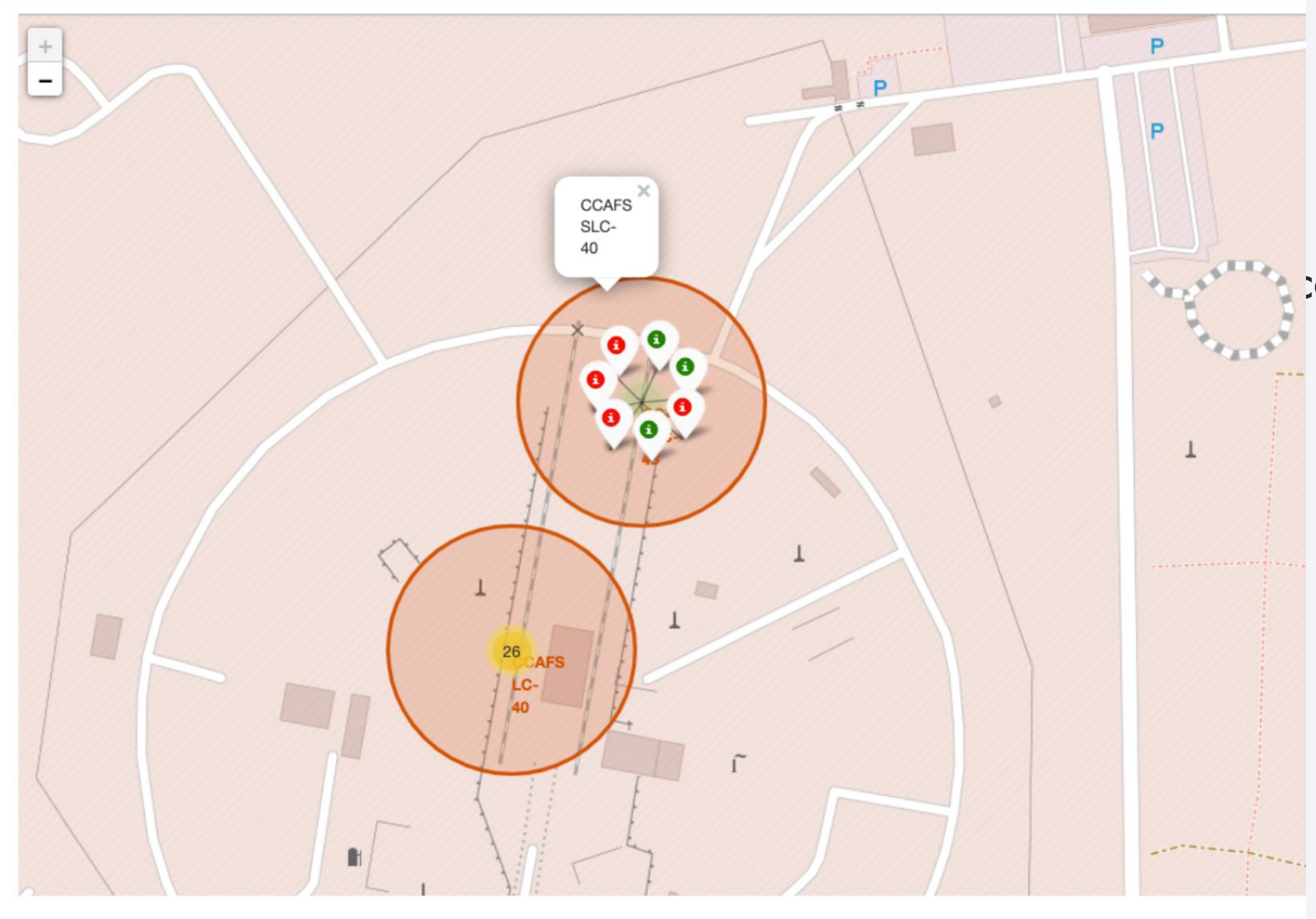
Launch Sites Proximities Analysis

Folium Map : Global Launch Sites



<Folium Map Screenshot 2>

- Rep
- Exp color-
  lab
- Exp



Section 4

Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

- Skills Networks Labs didn’t work

Section 5

Predictive Analysis (Classification)

Classification Accuracy

Create a decision tree classifier object then create a `GridSearchCV` object `tree_cv` with cv = 10. Fit the object to find the best parameters from the dictionary `parameters`.

```
]: parameters = {'criterion': ['gini', 'entropy'],
        'splitter': ['best', 'random'],
        'max_depth': [2*n for n in range(1,10)],
        'max_features': ['auto', 'sqrt'],
        'min_samples_leaf': [1, 2, 4],
        'min_samples_split': [2, 5, 10]}

   tree = DecisionTreeClassifier()
```

```
]: tree_cv = GridSearchCV(tree, parameters, cv=10)
```

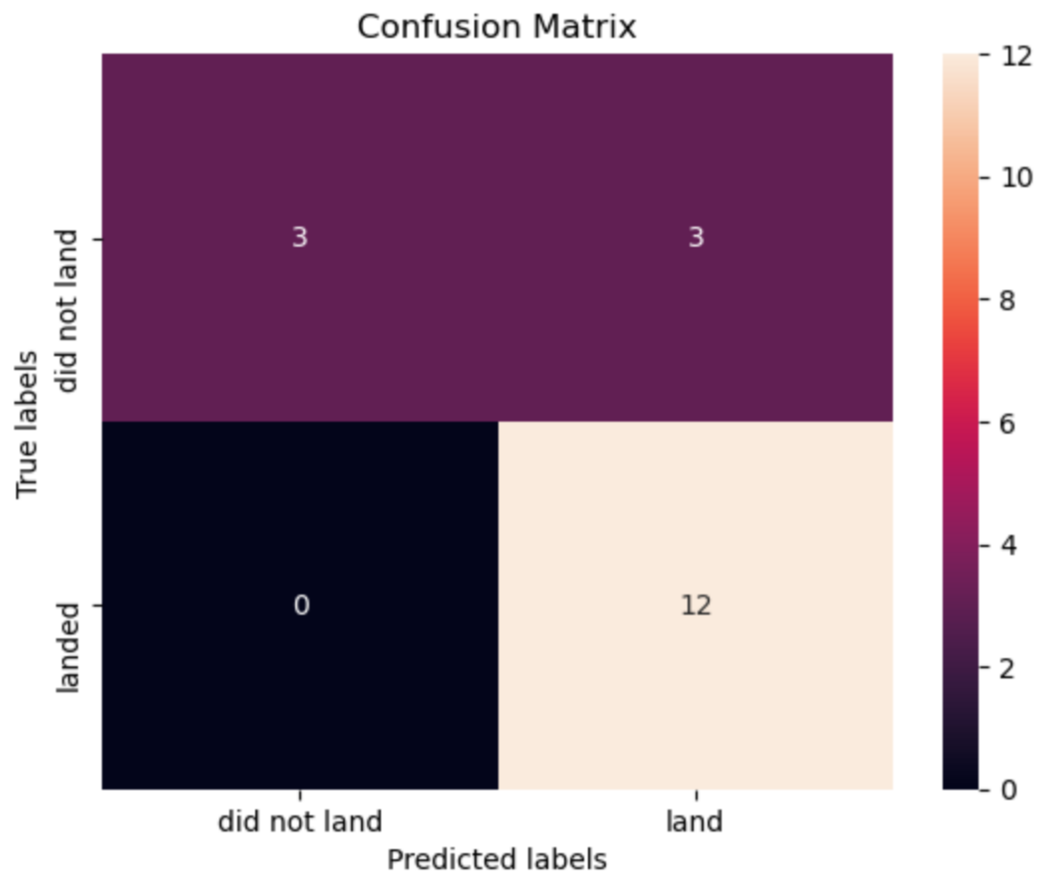
```
]: print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
   print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 18, 'max_features': 'auto', 'min_samples_leaf': 1,
'min_samples_split': 10, 'splitter': 'random'}
accuracy : 0.875
```

- Decision Tree model has the highest classification accuracy

Confusion Matrix

```
yhat = svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



- Show the confusion matrix of the best performing model with an explanation

Conclusions

- the sucess rate keep constant since 2013 and kept increasing till 2020
- Most of the sites which launch rockets with heavy payload mass have higher success rates
- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- Decision Tree is the best machine learning algorithm for this task

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!