IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Fuge Zou
12/16/2021

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection
  - Data Wrangling
  - Exploratory Data Analysis (EDA) with data visualization
  - Exploratory Data Analysis (EDA) with SQL
  - Building an interactive map with Folium
  - Building a Dashboard with Plotly Dash
  - Predicted Analysis of Success Rate

- Summary of all results
  - Exploratory Data Analysis (EDA) results
  - Interactive analytics dashboard
  - Machine learning resutls

Introduction

- Project background and context
  - We get data from SpaceX website with all their launching and landing information. Our goal is trying to collect, clean, model the data so we can learn what makes a successful landing.

- Problems you want to find answers
  - What is the driver for a successful landing
  - Is there any trend or business insights on the landing performance
  - How can we make the landing more successful in the future

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Using the SpaceX Rest API and Wikipedia combined
- Perform data wrangling
  - We clean the data using Pandas library, and we made the categorical variables one-hot encoding by calling the pd.get_dummies() function
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - I have tested KNN, Decision Tree, SVM and Logistic Regression ‘
  - I have done hyperparameter tuning using GridSearchCV

Data Collection

- Describe how data sets were collected.
  - I downloaded data using SpaceX Rest API and Wikipedia
- You need to present your data collection process use key phrases and flowcharts

SpaceX Rest API

Space X API → JSON File → Pandas

Wikipedia

HTML → Beautiful Soup → Pandas

Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose

https://github.com/fenixchow/IBM_Final_Capstone_Project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

- Getting Response

```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"

In [7]: response = requests.get(spacex_url)
```

- Put the data into Pandas

```
response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose

https://github.com/fenixchow/IBM_Final_Capstone_Project/blob/main/jupyter-labs-webscraping.ipynb

- Using Beautiful Soup

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

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

```
200
```

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

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content

soup = BeautifulSoup(page.text,'html.parser')
```

Data Wrangling

https://github.com/fenixchow/IBM_Final_Capstone_Project/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

Perform Exploratory Data Analysis EDA on dataset

Calculate the number of launches at each site

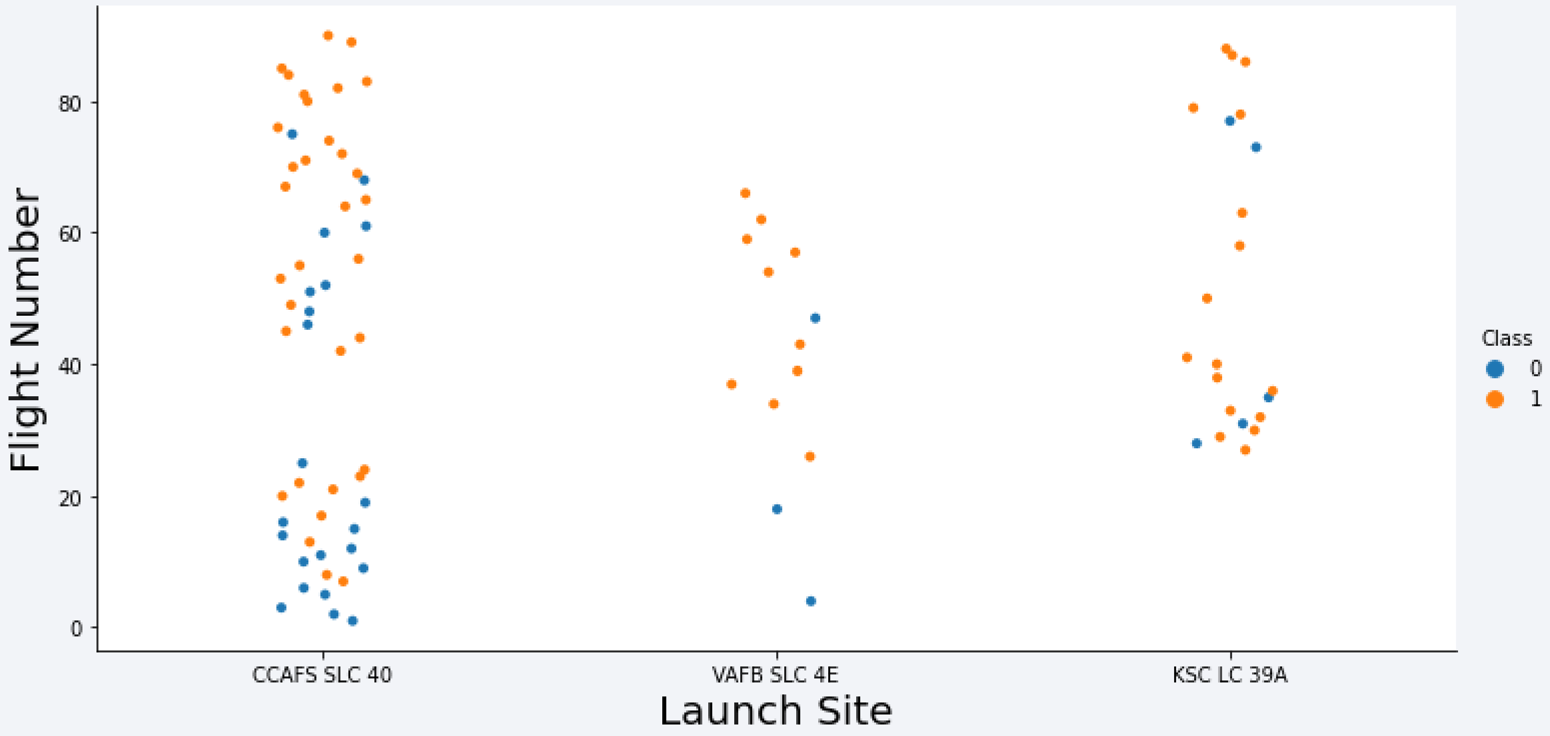
Calculate the number and occurrence of each orbit

Calculate the number and occurrence of mission outcome per orbit type

Create a landing outcome label from Outcome column

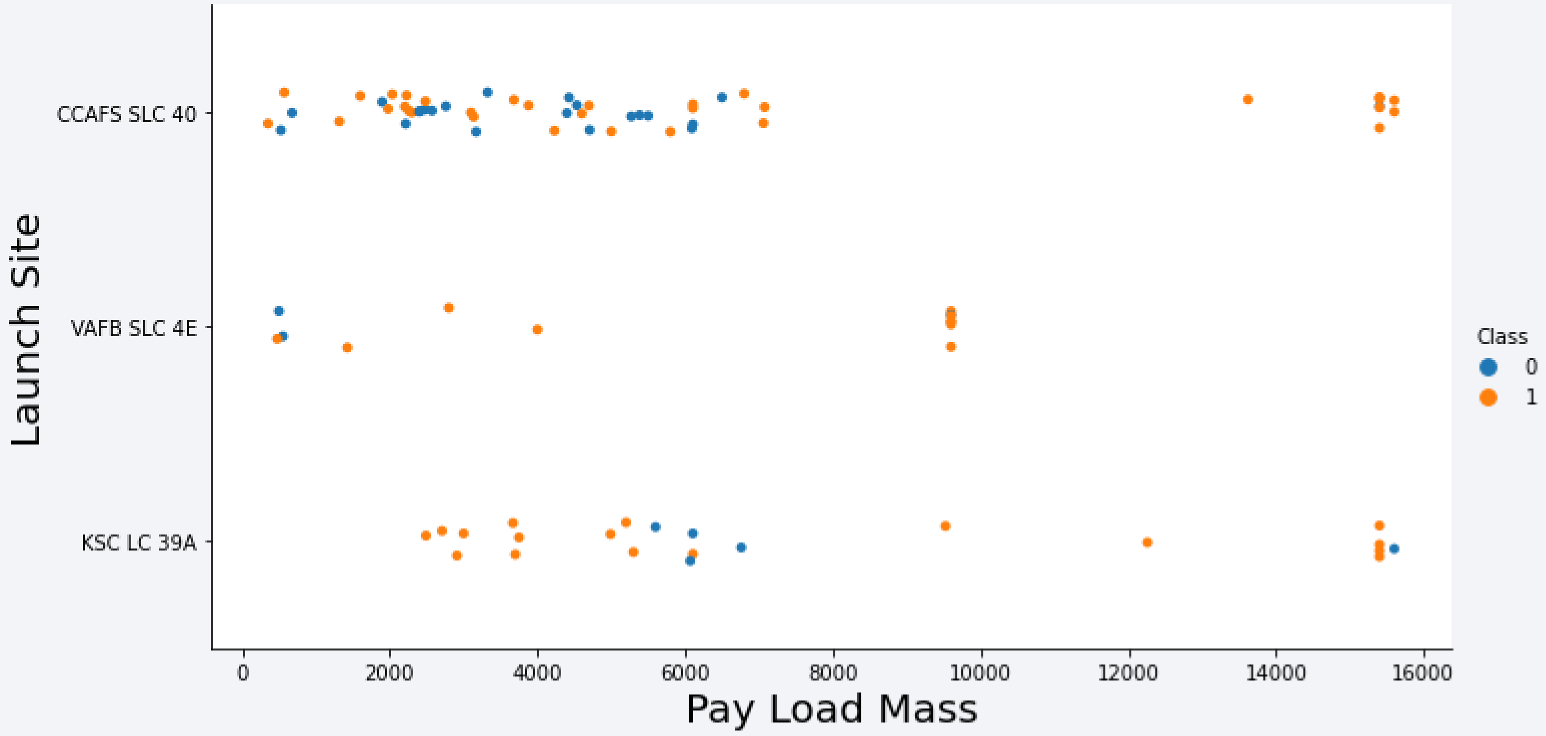
EDA with Data Visualization

https://github.com/fenixchow/IBM_Final_Capstone_Project/blob/main/jupyter-labs-eda-dataviz.ipynb



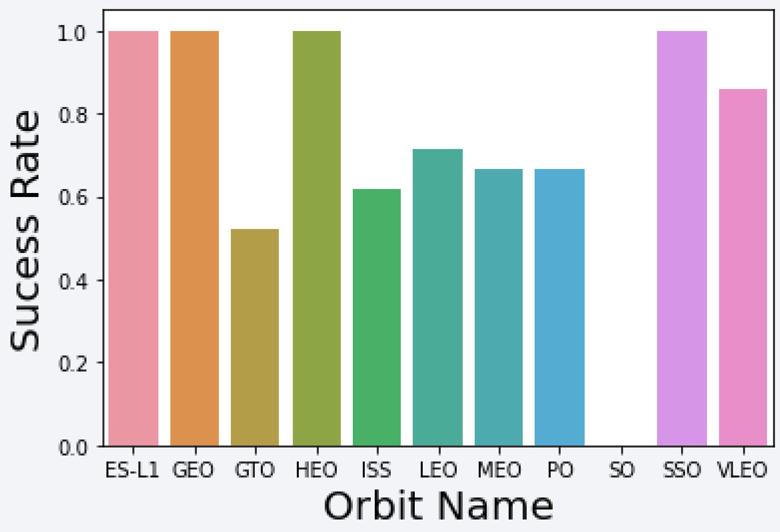
EDA with Data Visualization

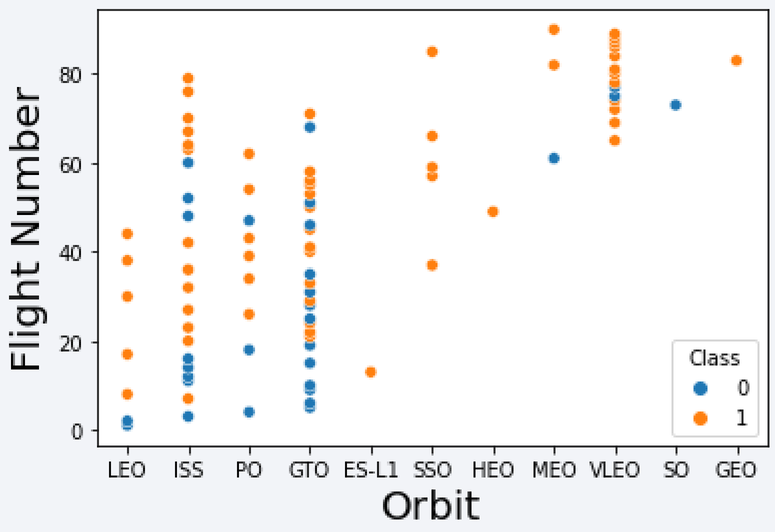
https://github.com/fenixchow/IBM_Final_Capstone_Project/blob/main/jupyter-labs-eda-dataviz.ipynb



EDA with Data Visualization

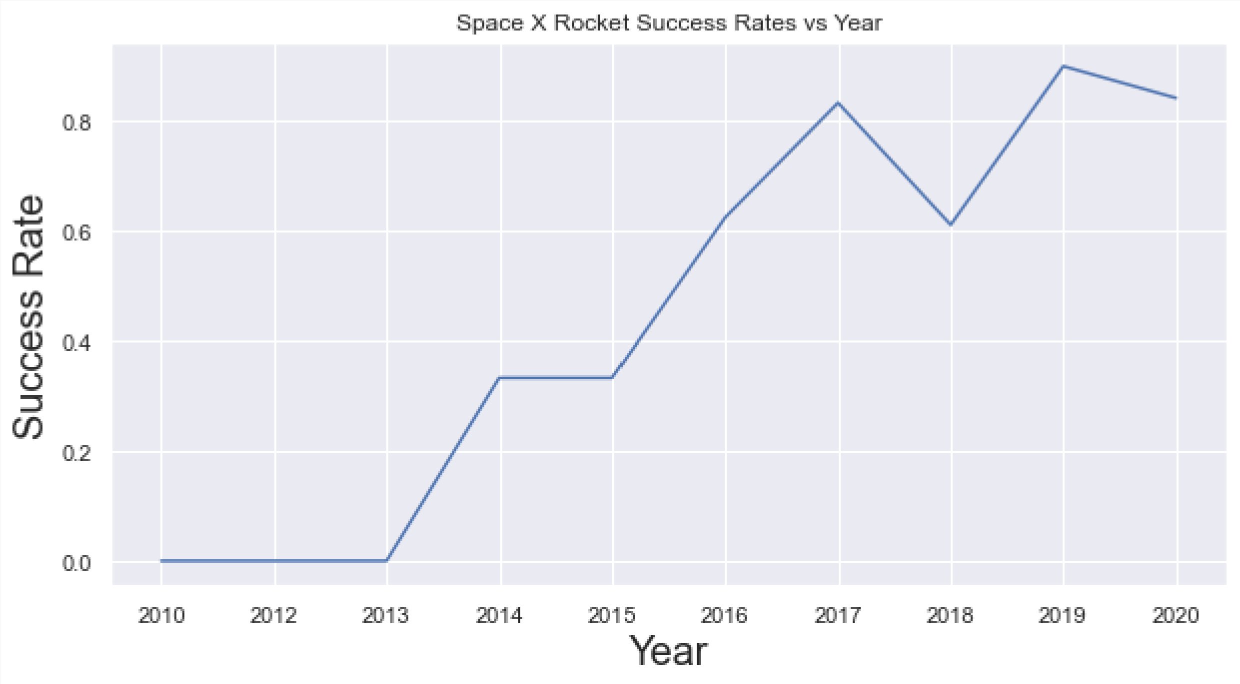
https://github.com/fenixchow/IBM_Final_Capstone_Project/blob/main/jupyter-labs-eda-dataviz.ipynb





EDA with Data Visualization

https://github.com/fenixchow/IBM_Final_Capstone_Project/blob/main/jupyter-labs-eda-dataviz.ipynb



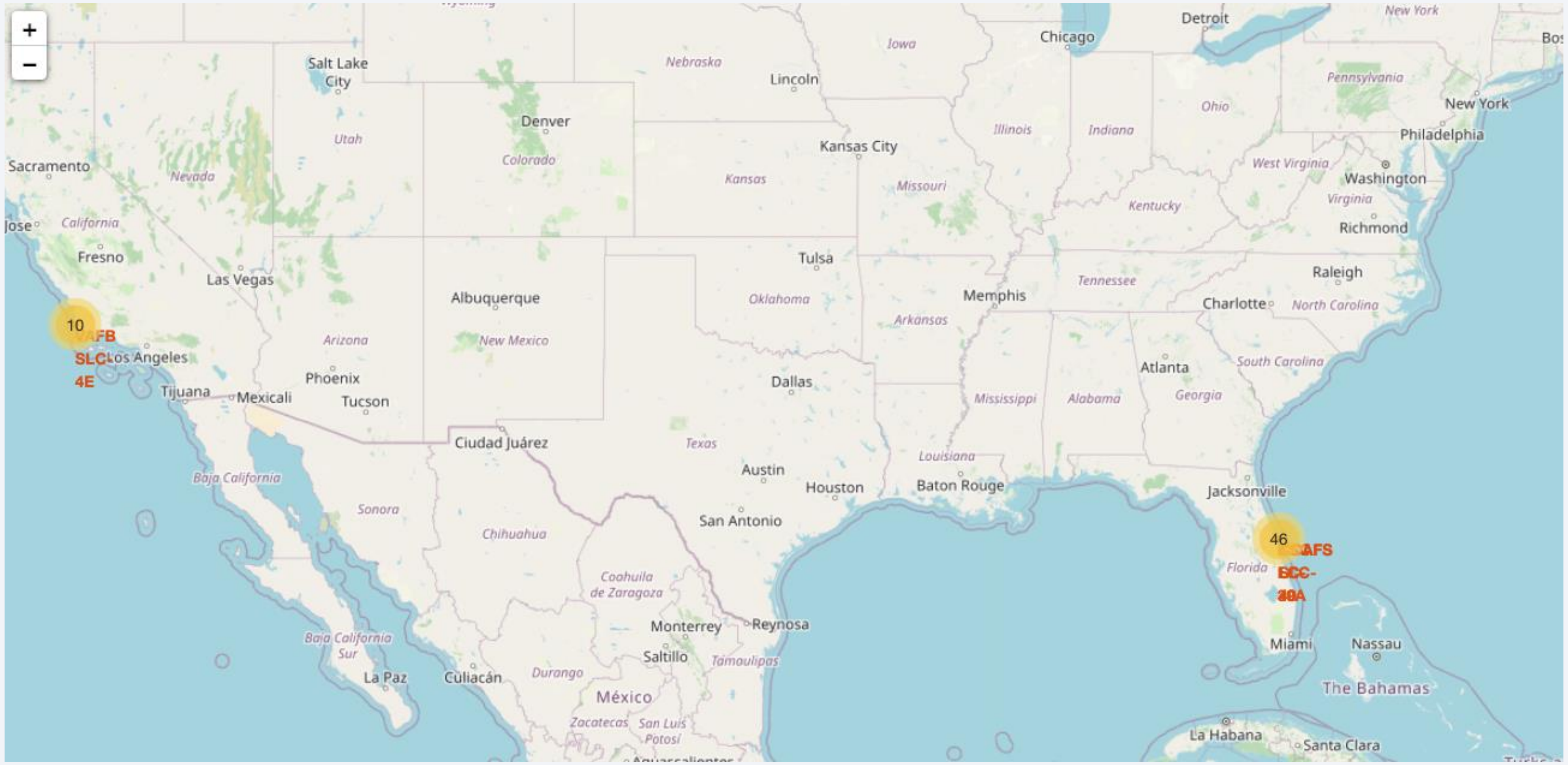
EDA with SQL

https://github.com/fenixchow/IBM_Final_Capstone_Project/blob/main/jupyter-labs-eda-sql-coursera.ipynb

Finished 10 Task on SQL to further explore the SpaceX Dataset
Used IBM DB2 as the cloud database
Used Python to call IBM DB2 using the magic SQL command
Check the GitHub for details

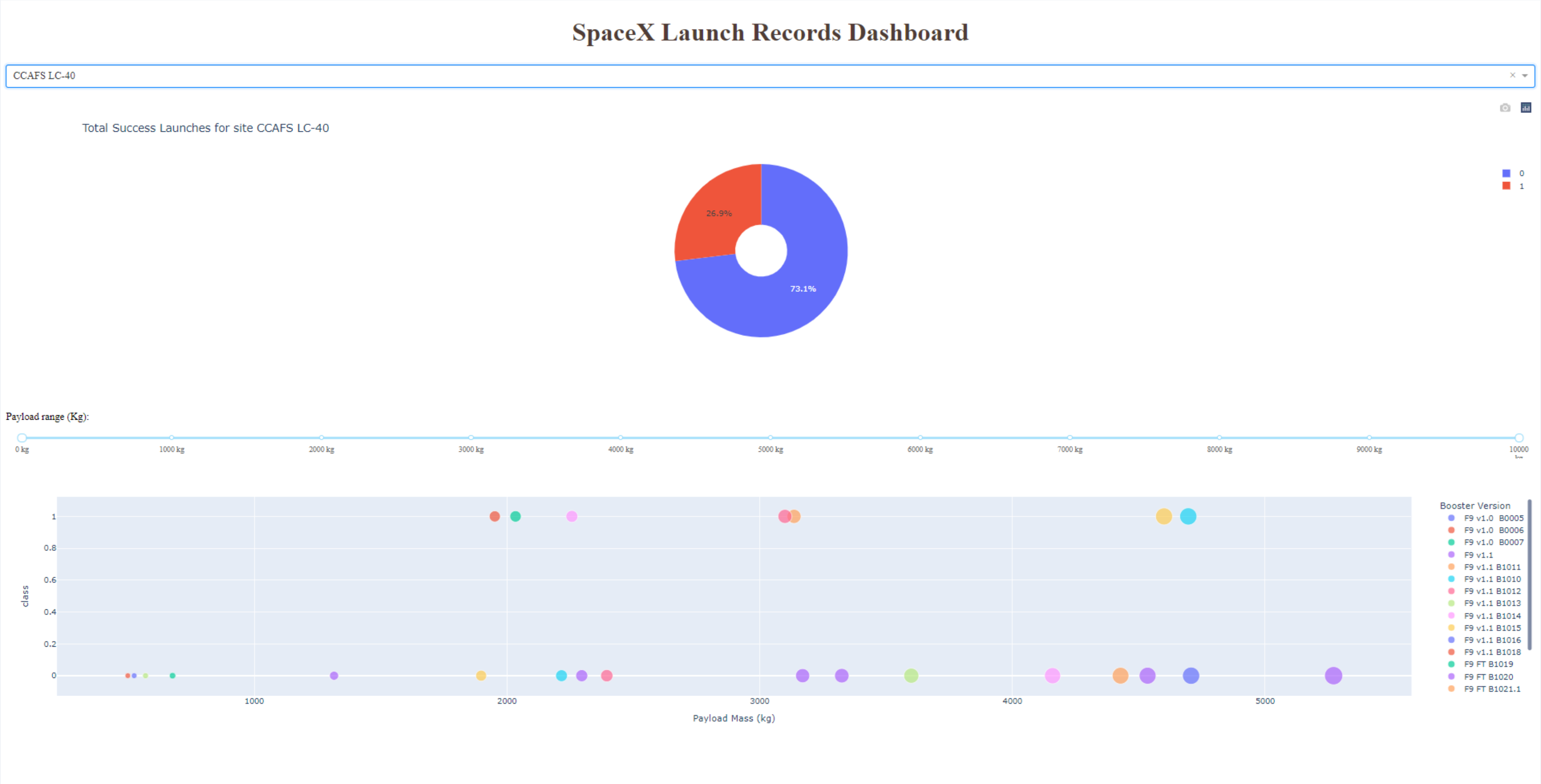
Build an Interactive Map with Folium

https://github.com/fenixchow/IBM_Final_Capstone_Project/blob/main/lab_jupyter_launch_site_location.ipynb



Build a Dashboard with Plotly Dash

https://github.com/fenixchow/IBM_Final_Capstone_Project/blob/main/Interactive_Dashboard_w_Ploty_Dash.ipynb



Predictive Analysis (Classification)

https://github.com/fenixchow/IBM_Final_Capstone_Project/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

```
Best Algorithm is Tree with a score of 0.8767857142857143
Best Params is : {'criterion': 'entropy', 'max_depth': 4, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_split': 5, 'splitter': 'random'}
```

I used SVM, Logistic Regression, Decision Tree and KNN model, the best results on test data is decision tree with a score of 87.7%

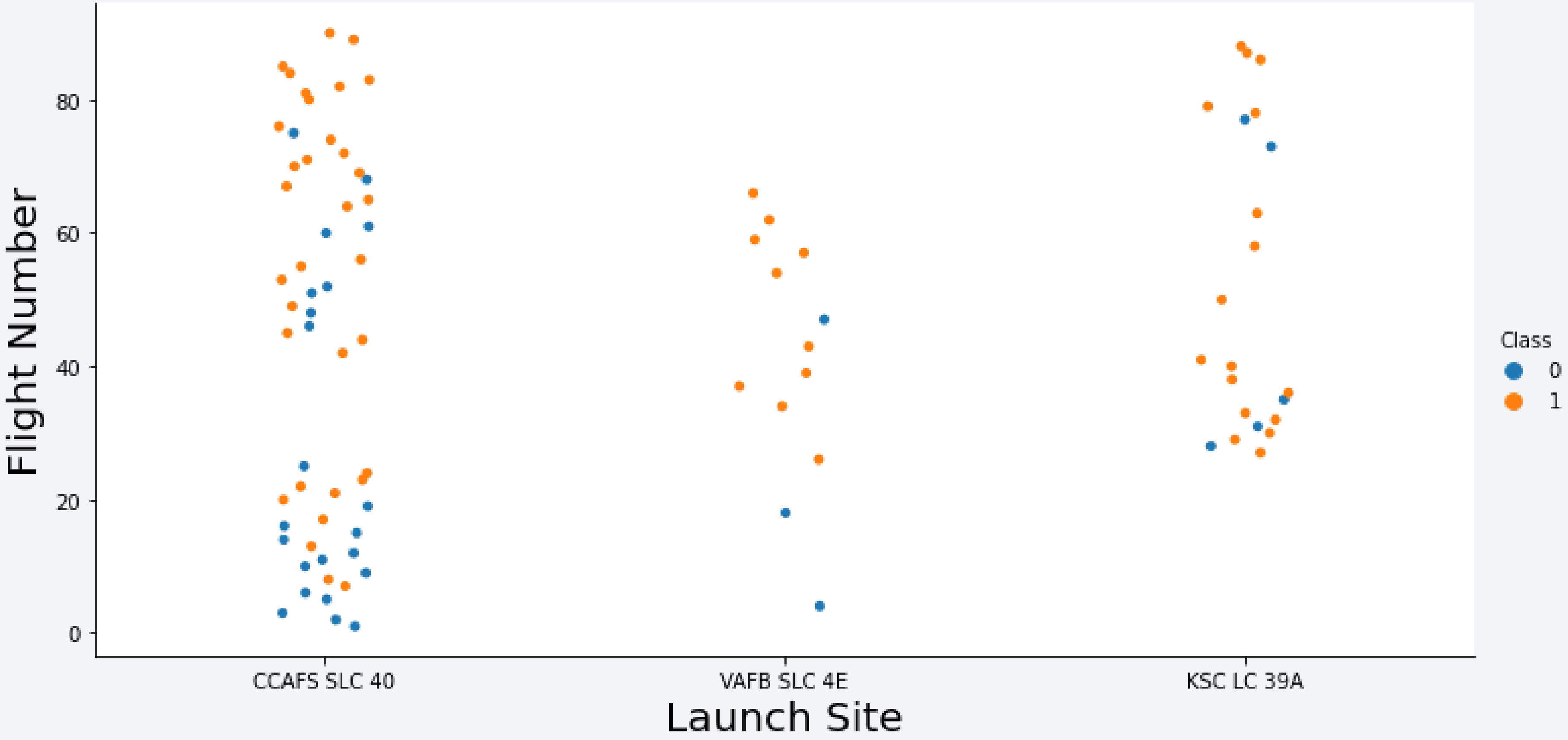
Results

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

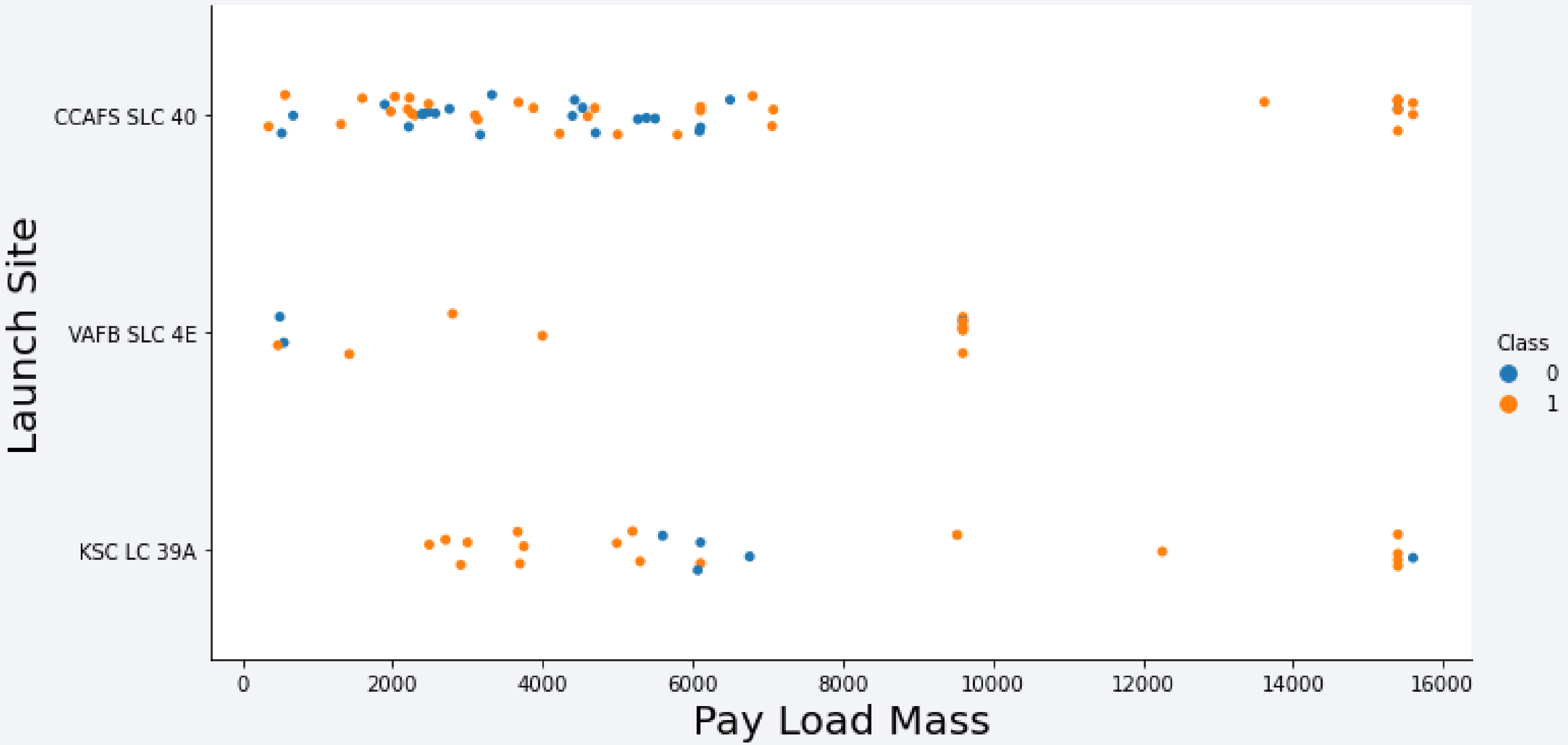
Section 2

Insights drawn from EDA

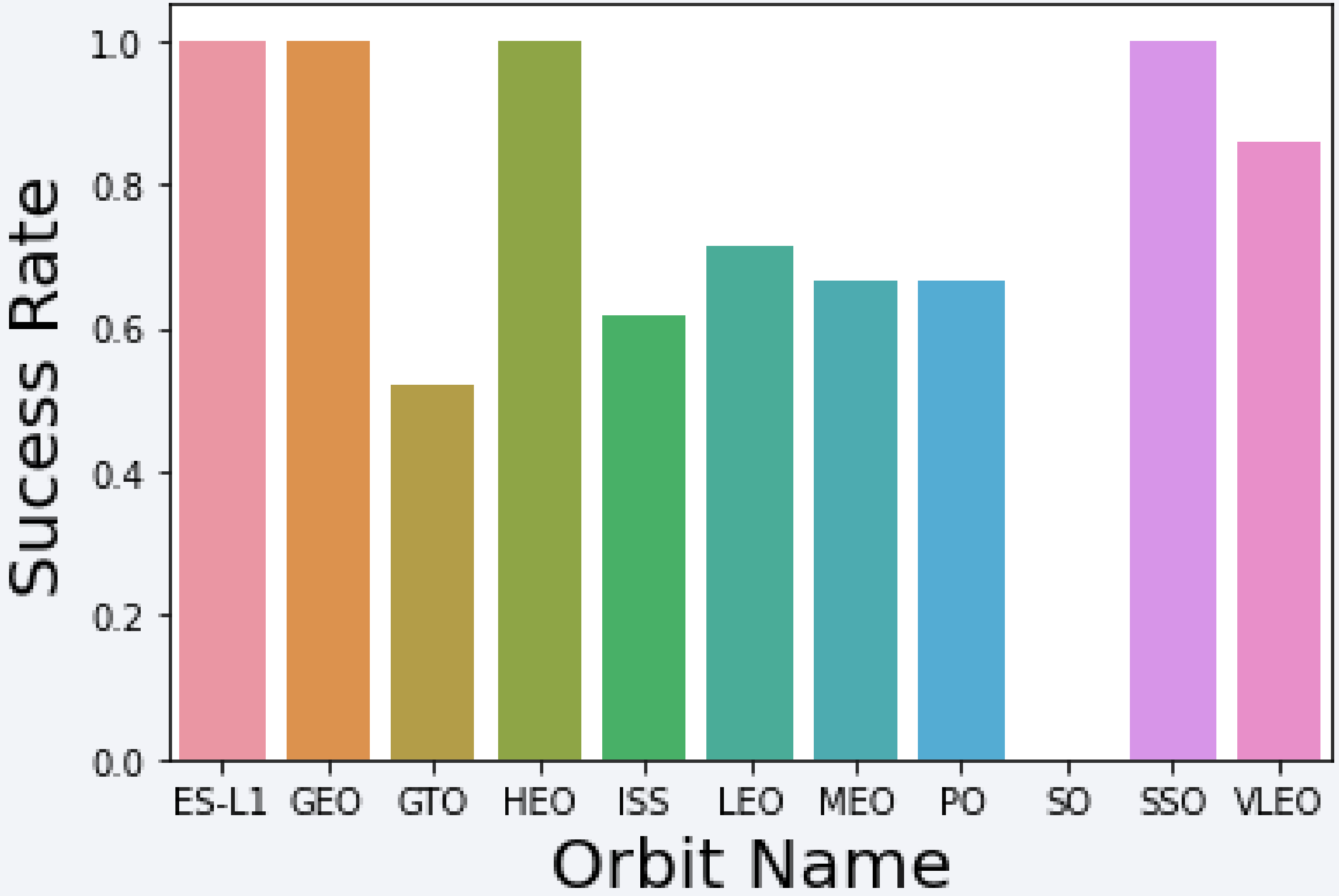
Flight Number vs. Launch Site



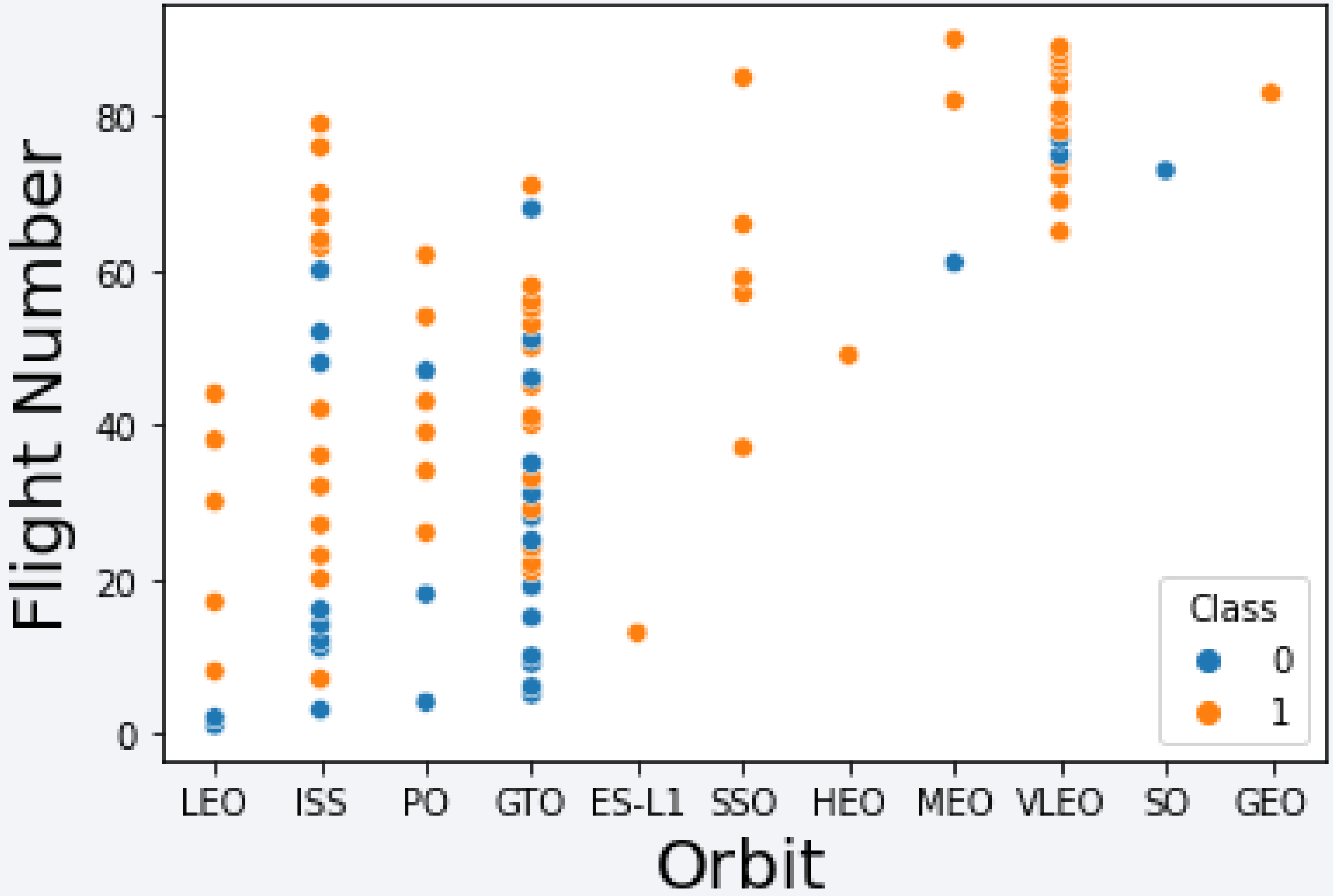
Payload vs. Launch Site



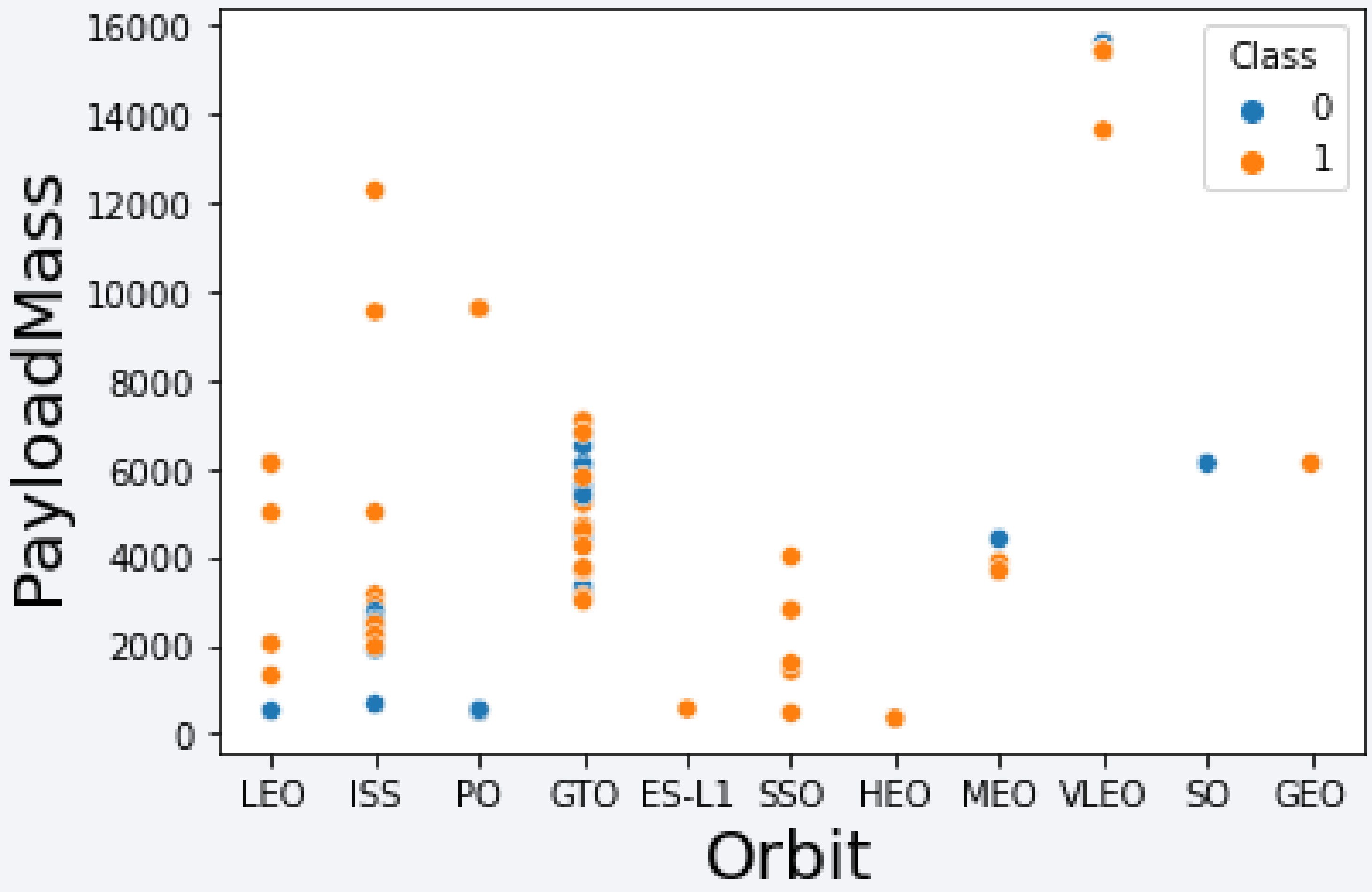
Success Rate vs. Orbit Type



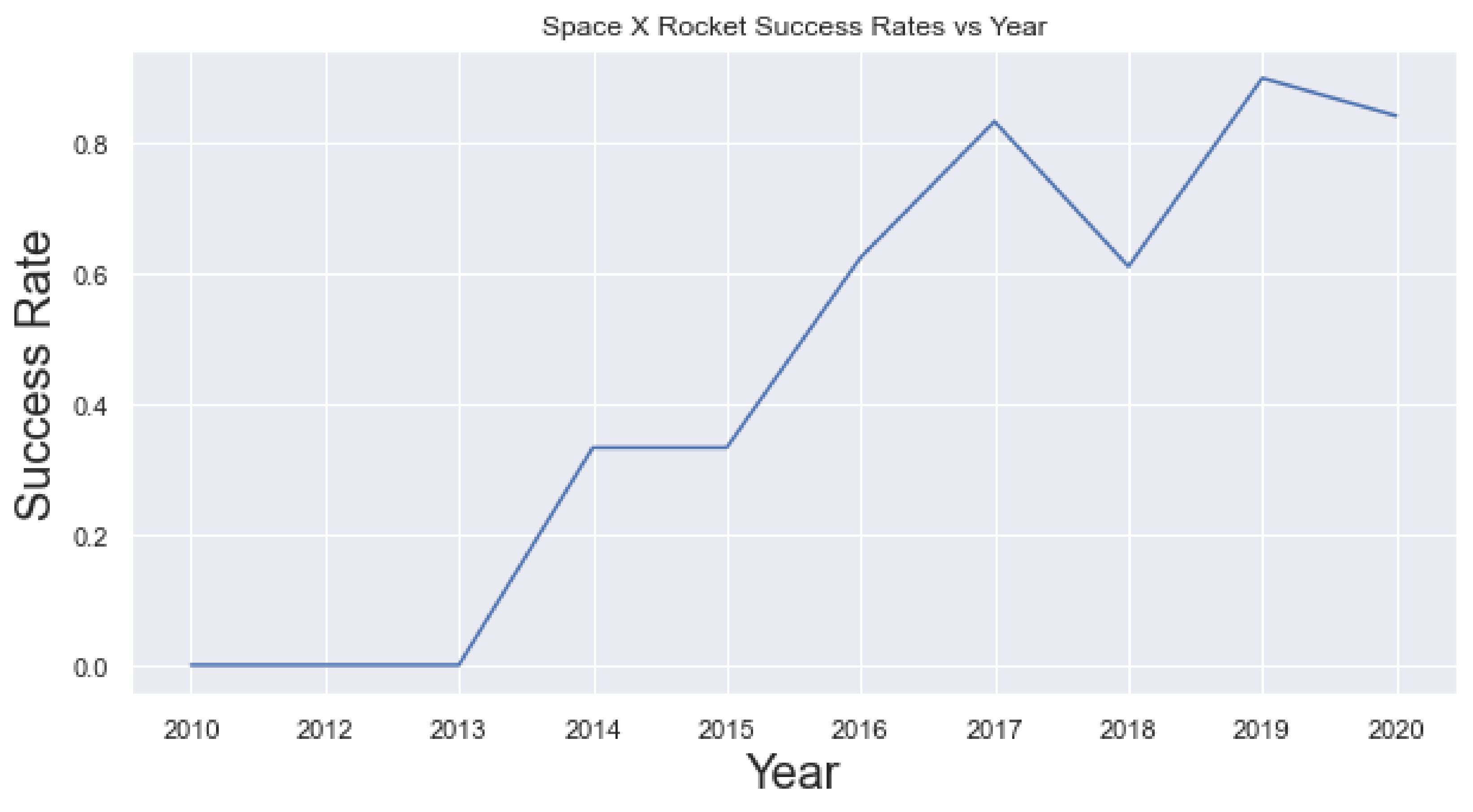
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

Task 1

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

In [17]:
```
%%sql

SELECT DISTINCT Launch_Site FROM SPACEXTBL
```

```
 * ibm_db_sa://drd98900:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4
Done.
```

Out[17]:

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

Launch Site Names Begin with 'CCA'

Task 2

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

In [20]:
```
%%sql

SELECT * FROM SPACEXTBL
WHERE Launch_Site LIKE 'CCA%'
LIMIT 5
```

```
 * ibm_db_sa://drd98900:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.clo
Done.
```

Out[20]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outc
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Succ
2010-12-08	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	Succ
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Succ
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Succ
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Succ

Total Payload Mass

Task 3

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

In [23]:

```
%%sql

SELECT SUM(payload_mass__kg_) AS total_payload_mass FROM SPACEXTBL
WHERE Customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://drd98900:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnr
Done.
```

Out[23]:

total_payload_mass
45596

Average Payload Mass by F9 v1.1

Task 4

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

In [26]:
```
%%sql

SELECT AVG(payload_mass__kg_) AS average_payload_mass FROM SPACEX
WHERE booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://drd98900:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41
Done.
```

Out[26]:

average_payload_mass
2928

First Successful Ground Landing Date

List the date when the first successful landing outcome in ground pad was acheived.

Hint:Use min function

In [30]:
```
%%sql
SELECT DISTINCT landing__outcome FROM SPACEXTBL
```

```
 * ibm_db_sa://drd98900:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqr
Done.
```

Out[30]:

landing__outcome
Controlled (ocean)
Failure
Failure (drone ship)
Failure (parachute)
No attempt
Precluded (drone ship)
Success
Success (drone ship)
Success (ground pad)
Uncontrolled (ocean)

Successful Drone Ship Landing with Payload between 4000 and 6000

In [32]:

```
%%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://drd98900:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:32286/BLUDB
Done.
```

Out[32]:

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

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

In [36]:

```
%%sql

SELECT mission_outcome, COUNT(*) FROM SPACEXTBL
GROUP BY mission_outcome
```

```
 * ibm_db_sa://drd98900:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:32286/BLUDB
Done.
```

Out[36]:

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	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 [39]:
```
# %%sql

# SELECT DISTINCT booster_version, MAX(payload_mass__kg_) AS maximum_payload_mass FROM SPACEXTBL
# GROUP BY booster_version ORDER BY maximum_payload_mass DESC
```

In [41]:
```
%%sql

SELECT DISTINCT booster_version FROM SPACEXTBL
WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM SPACEXTBL)

```

```
 * ibm_db_sa://drd98900:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:32286/BLUDB
Done.
```

Out[41]:

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6

2015 Launch Records

Task 9

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

In [46]:

```
%%sql


SELECT landing__outcome, booster_version, launch_site FROM SPACEXTBL
WHERE YEAR(DATE) = 2015 AND landing__outcome = 'Failure (drone ship)'
```

```
 * ibm_db_sa://drd98900:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.
Done.
```

Out[46]:

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

Task 10

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

In [64]:

```
%%sql

SELECT landing__outcome, COUNT(*) AS COUNT FROM (SELECT * FROM SPACEXTBL WHERE DATE BETWEEN '2011-06-04' AND '2017-03-20')
GROUP BY landing__outcome
ORDER BY COUNT DESC
```

```
 * ibm_db_sa://drd98900:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:32286/BLUDB
Done.
```

Out[64]:

landing__outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1

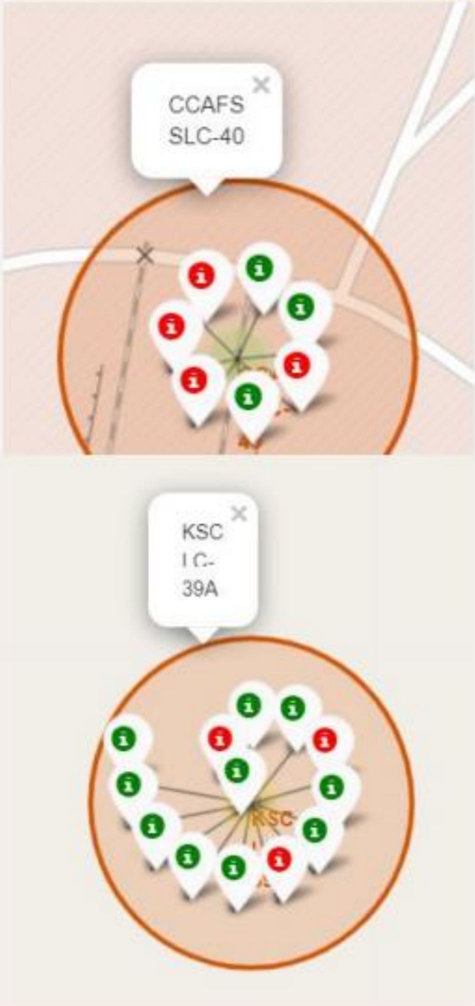
Section 4

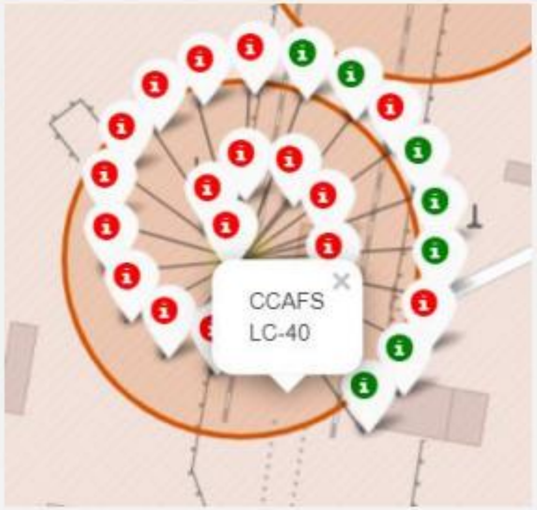
Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

- Repl
- Explo
  all la
- Expla

<Folium Map Screenshot 2>



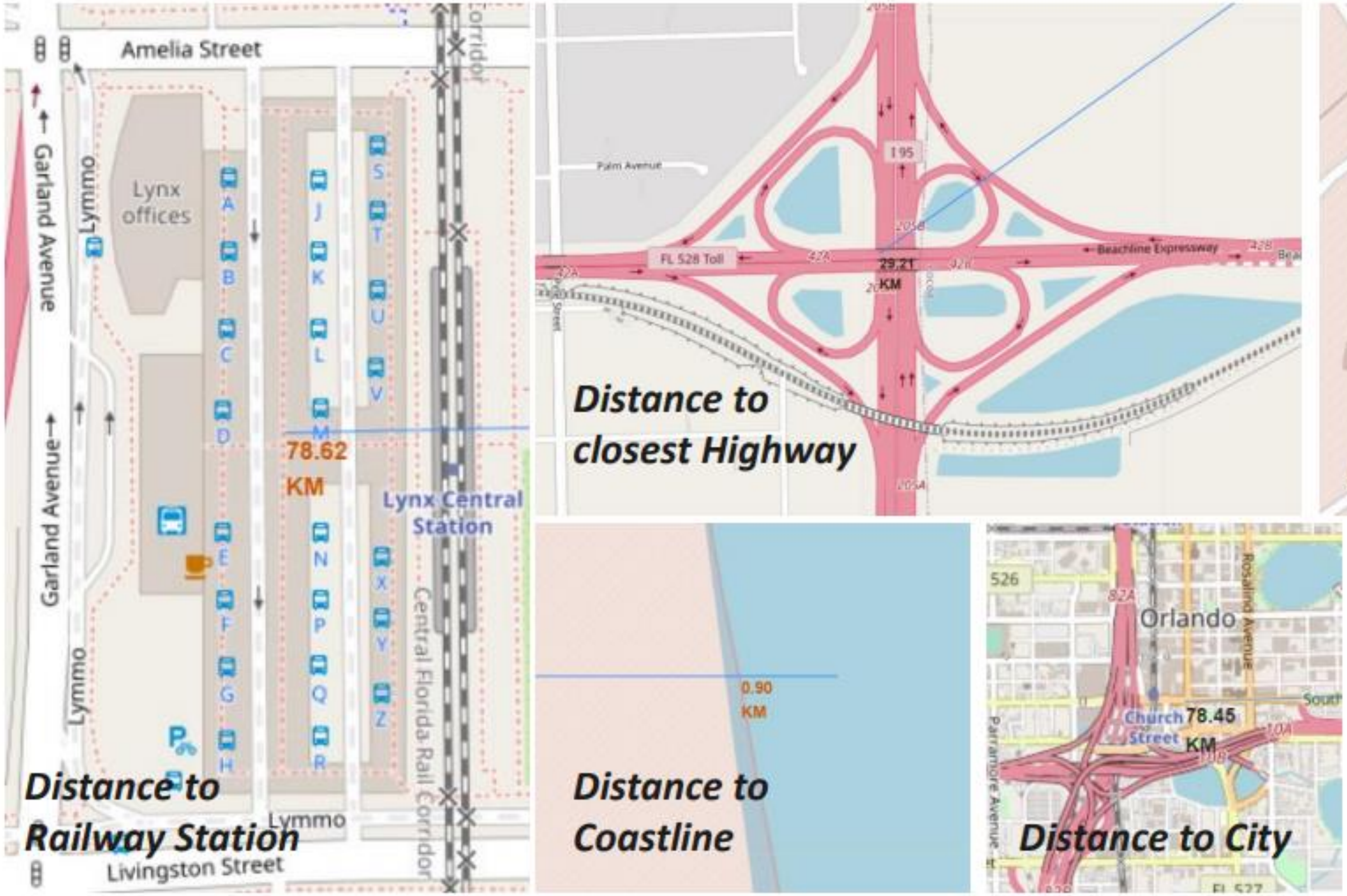


Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

<Folium Map Screenshot 3>

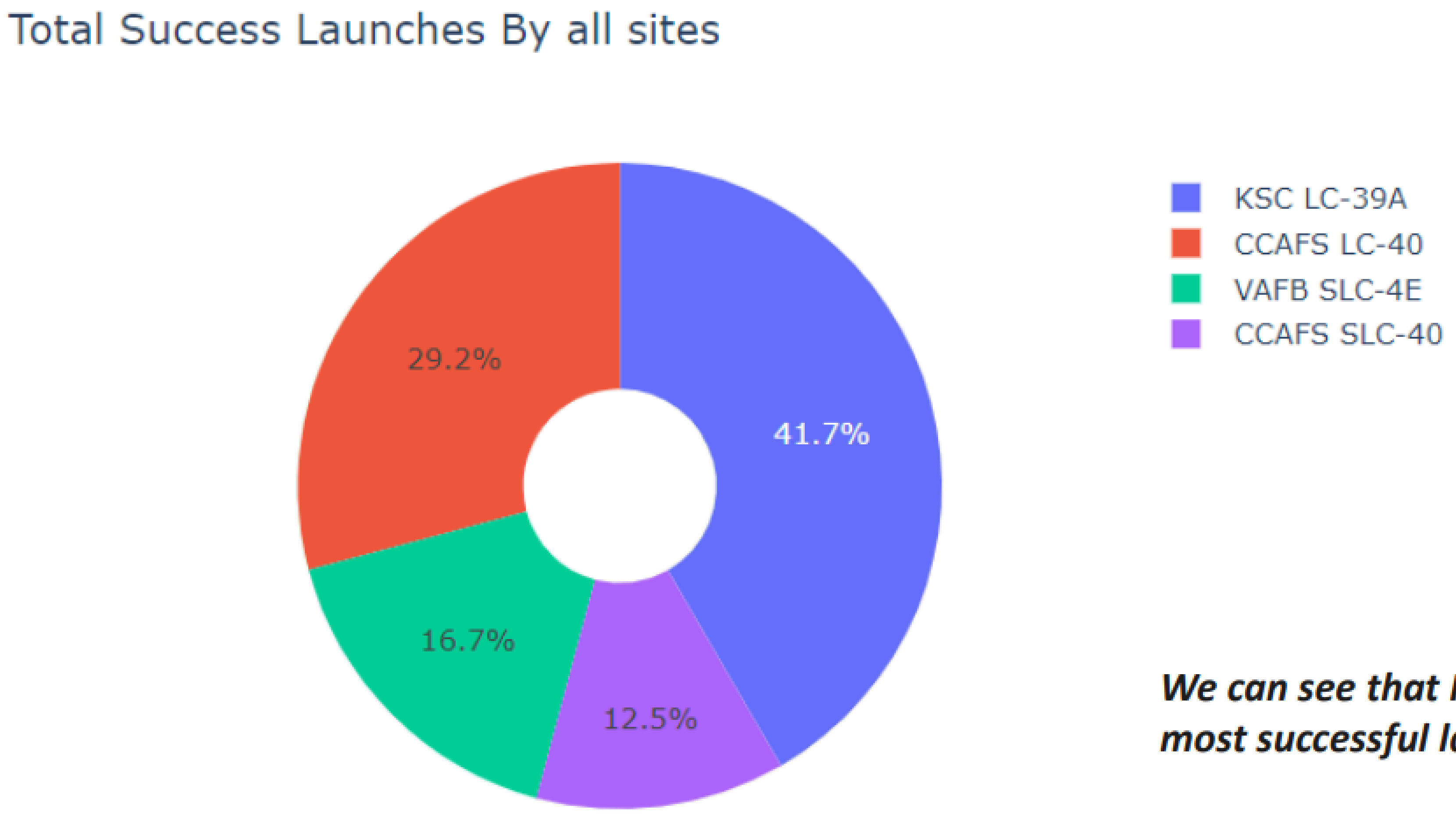


- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

Section 5

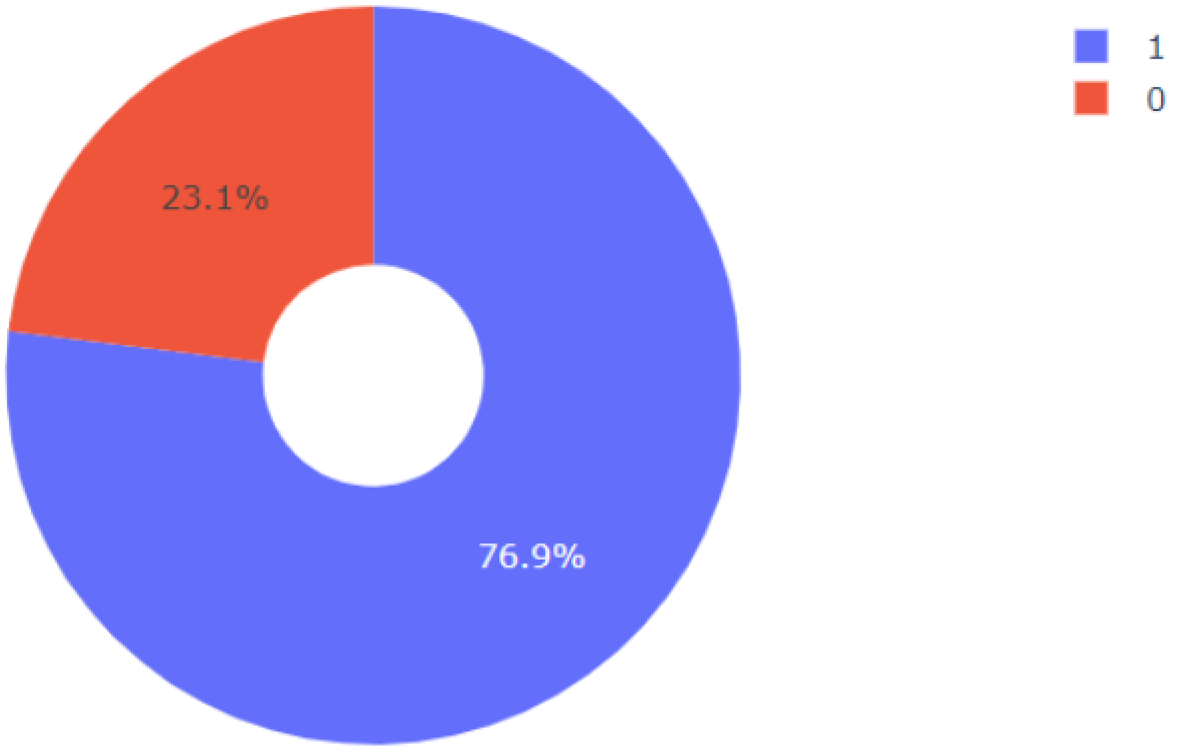
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>



We can see that KSC LC-39A had the most successful launches from all the sites

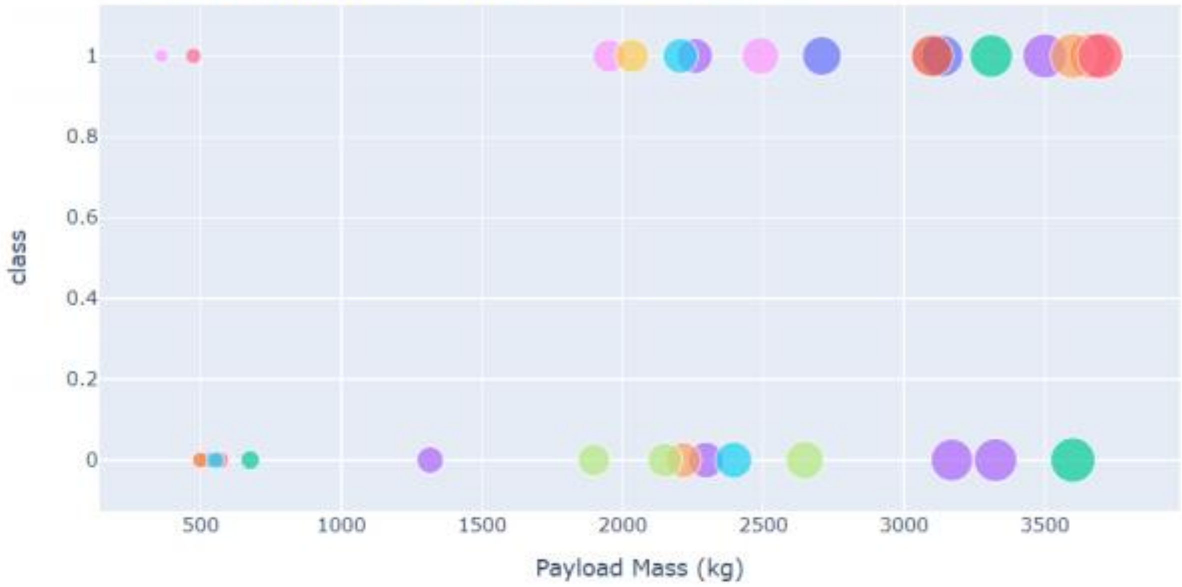
<Dashboard Screenshot 2>



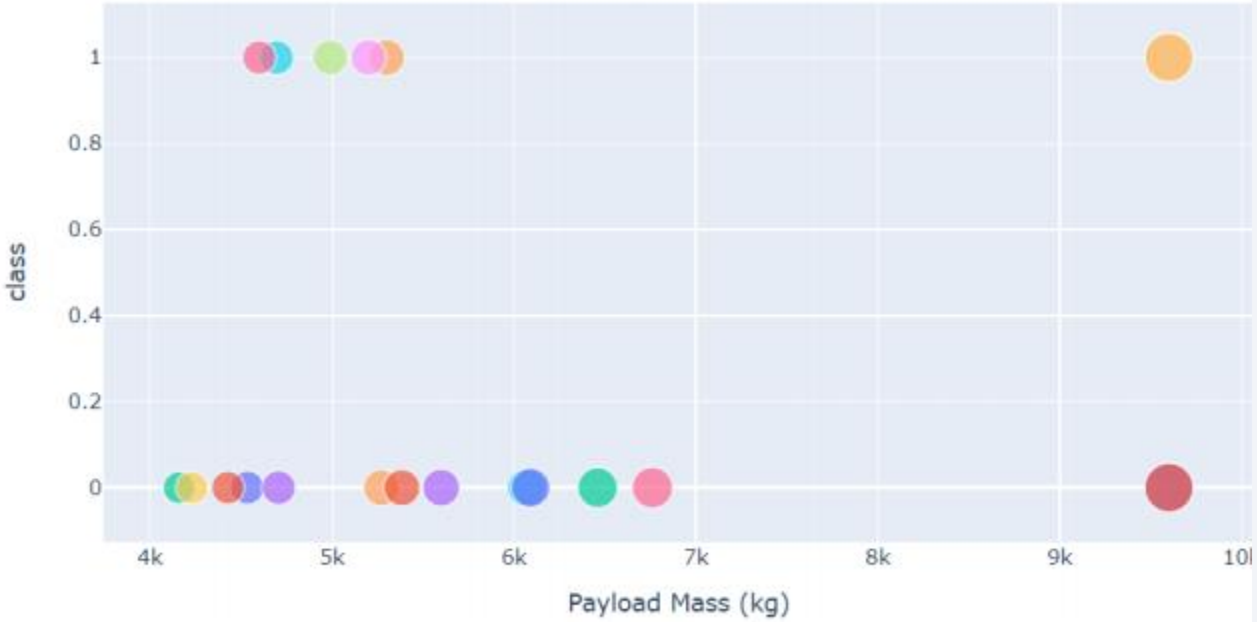
KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

<Dashboard Screenshot 3>

Low Weighted Payload 0kg – 4000kg



Heavy Weighted Payload 4000kg – 10000kg



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

Section 6

Predictive Analysis (Classification)

Classification Accuracy

Method	Accuracy
Logistic Regression	0.8333333333333334
SVM	0.8482142857142856
Decision Tree	0.8767857142857143
KNN	0.8333333333333334

Confusion Matrix

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

The TP / TP + FP = 12 / 15 = 0.8 (Precision)

The TP / TP + FN = 1 (Recall)

The f1-score is pretty good in this case, with only 3 false positive, where the model predicted successful landing, but in reality it failed

Conclusions

- Decision Tree in this example is the best ML with an accuracy of 87.7% on test dataset
- Low weighted payloads perform better than heavier payloads
- The success rate of SpaceX improve with time from 2010-2020
- Orbit GEO, HEO,SSO and ES-L1 has the best successful rate
- KSC LC-39A had the most successful launches

Appendix

- They are all in the GitHub

Thank you!