IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

ABHIJIT DAS
August 2024

Outline

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

Executive Summary

- Summary of methodologies: **In this project, we will predict if the Falcon 9 first stage will land successfully.**

The methodologies employed include:

- ❑ Data Sourcing using API calls and Web scraping
- ❑ Data Wrangling
- ❑ Exploratory Data Analysis using Visualization Libraries and SQL
- ❑ Building Dashboards using Folium, Dash and Plotly
- ❑ Machine Learning Algorithms for classification Problem in hand

- Summary of all results : **Decision Tree purportedly is the best ML Algorithm to predict the outcome. This follows from looking at the evaluation metrics of all ML Algorithms in the Jupyter Notebook.**

Introduction

- Project background and context

In this capstone, we will predict if the Falcon 9 first stage will land successfully. 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.

- Problems you want to find answers

Utilize machine learning algorithms to build a predictive model to help answer the above question in hand

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology: CRISP – DM (API & Web Scrapping)
- Perform data wrangling: Using Pandas and Numpy Library Functions
- Perform exploratory data analysis (EDA) using: Visualization and SQL
- Perform interactive visual analytics using: Folium, Plotly and Dash
- Perform predictive analysis using classification models: SVM, Decision Trees, KNN and Logistic Regression

Data Collection

- Describe how data sets were collected.

Data sets were primarily collected using API calls and Web scrapping

- You need to present your data collection process use key phrases and flowcharts

Convert a JSON file into a Python Pandas data frame

Develop Python code to manipulate data in a Pandas data frame

Utilize data analysis tools to load a dataset, clean it, and find insights from it using Pandas, Numpy , BeautifulSoup and SciPy Library functions

Data Collection – SpaceX API

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

❖ **Request to the SpaceX API**

❖ **Clean the requested data**

- 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/1abhijitdas1/IBMDataScience/blob/61c43b7231f5ca0c74a54ad678b1c5acc75d74a5/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts
  - ❖ **Extract a Falcon 9 launch records HTML table from Wikipedia**
  - ❖ **Parse the table and convert it into a Pandas data frame**
- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose

https://github.com/1abhijitdas1/IBMDataScience/blob/61c43b7231f5ca0c74a54ad678b1c5acc75d74a5/jupyter-labs-webscraping.ipynb

Data Wrangling

In this, 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.

In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.

We will mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.

https://github.com/1abhijitdas1/IBMDataScience/blob/61c43b7231f5ca0c74a54ad678b1c5acc75d74a5/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Visualize the relationship between Flight Number and Launch Site
- Visualize the relationship between Payload Mass and Launch Site
- Visualize the relationship between success rate of each orbit type
- Visualize the relationship between Flight Number and Orbit type
- Visualize the relationship between Payload Mass and Orbit type
- Visualize the launch success yearly trend
- Create dummy variables to categorical columns

https://github.com/1abhijitdas1/IBMDataScience/blob/61c43b7231f5ca0c74a54ad678b1c5acc75d74a5/edadataviz.ipynb

EDA with SQL

1.Understand the Spacex DataSet
2.Load the dataset into the corresponding table in a SQL Lite database
3.Use SQL Magic in Notebook

https://github.com/1abhijitdas1/IBMDataScience/blob/61c43b7231f5ca0c74a54ad678b1c5acc75d74a5/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

The launch success rate may depend on many factors such as payload mass, orbit type, and so on. It may also depend on the location and proximities of a launch site, i.e., the initial position of rocket trajectories. Finding an optimal location for building a launch site certainly involves many factors and hopefully we could discover some of the factors by analyzing the existing launch site locations. In exploratory data analysis labs, we have visualized the SpaceX launch dataset using matplotlib and seaborn and discovered some preliminary correlations between the launch site and success rates. Here, we will be performing more interactive visual analytics using Folium.

https://github.com/1abhijitdas1/IBMDataScience/blob/61c43b7231f5ca0c74a54ad678b1c5acc75d74a5/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- **Build an interactive dashboard that contains pie charts and scatter plots to analyze data with the Plotly Dash Python library.**

- **Build a dashboard to analyze launch records interactively with Plotly Dash.**

https://github.com/1abhijitdas1/IBMDataScience/blob/61c43b7231f5ca0c74a54ad678b1c5acc75d74a5/Build%20an%20Interactive%20Dashboard%20with%20Ploty%20Dash.txt

Predictive Analysis (Classification)

Perform exploratory Data Analysis and determine Training Labels

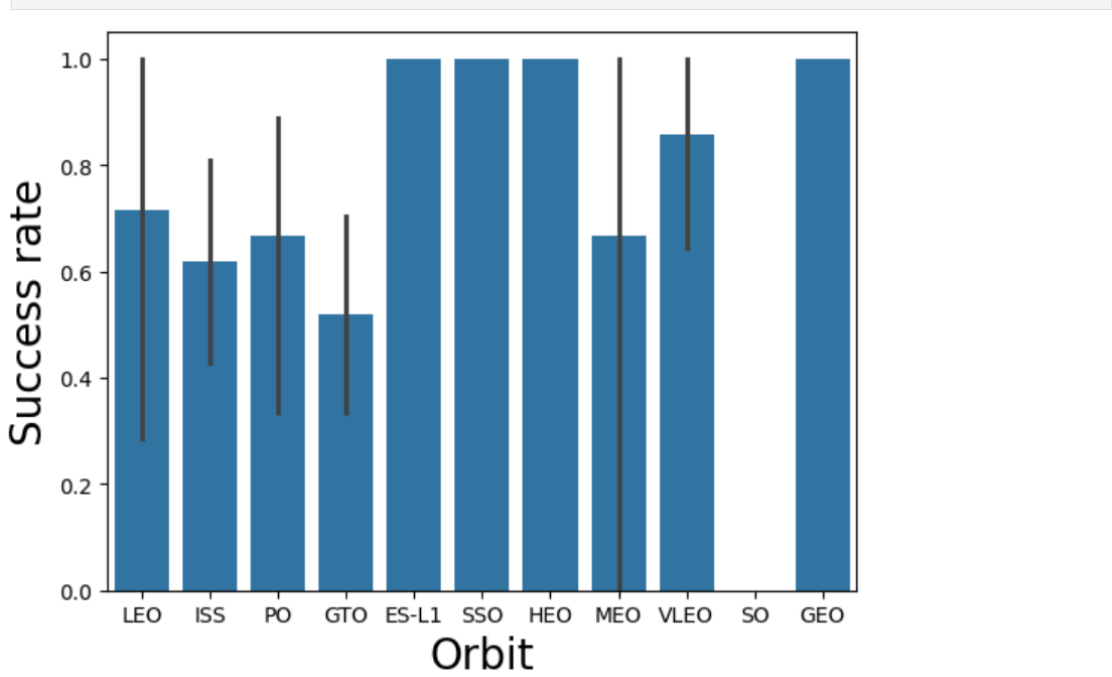
- **create a column for the class**
- **Standardize the data**
- **Split into training data and test data**

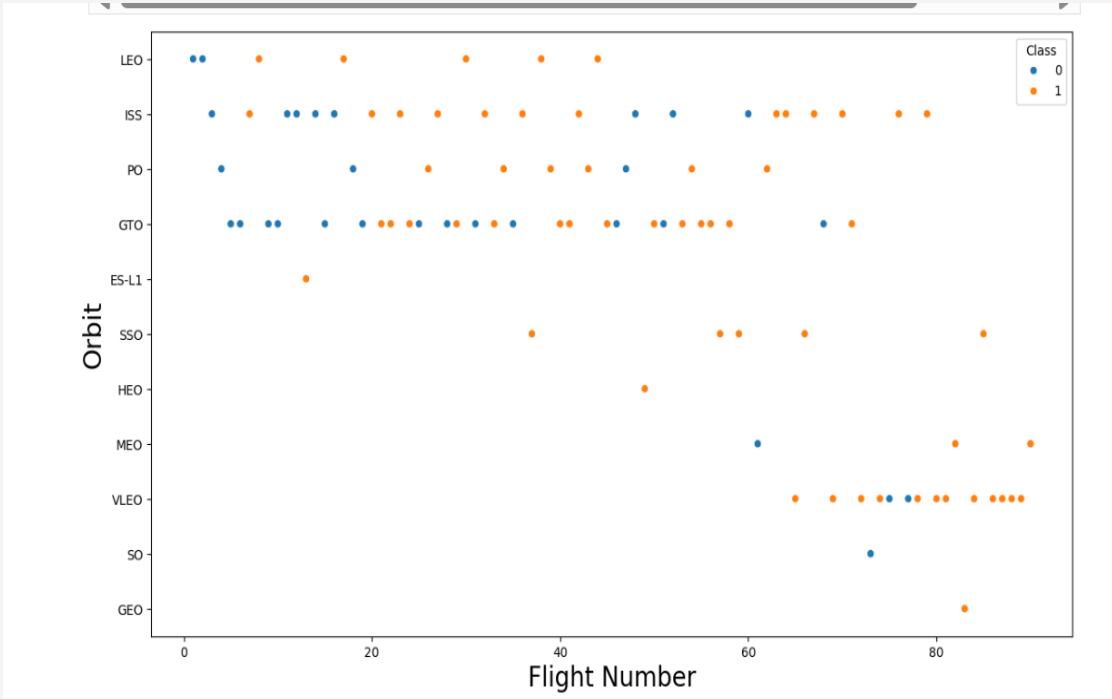
-Find best Hyperparameter for SVM, Classification Trees and Logistic Regression

- **Find the method performs best using test data**

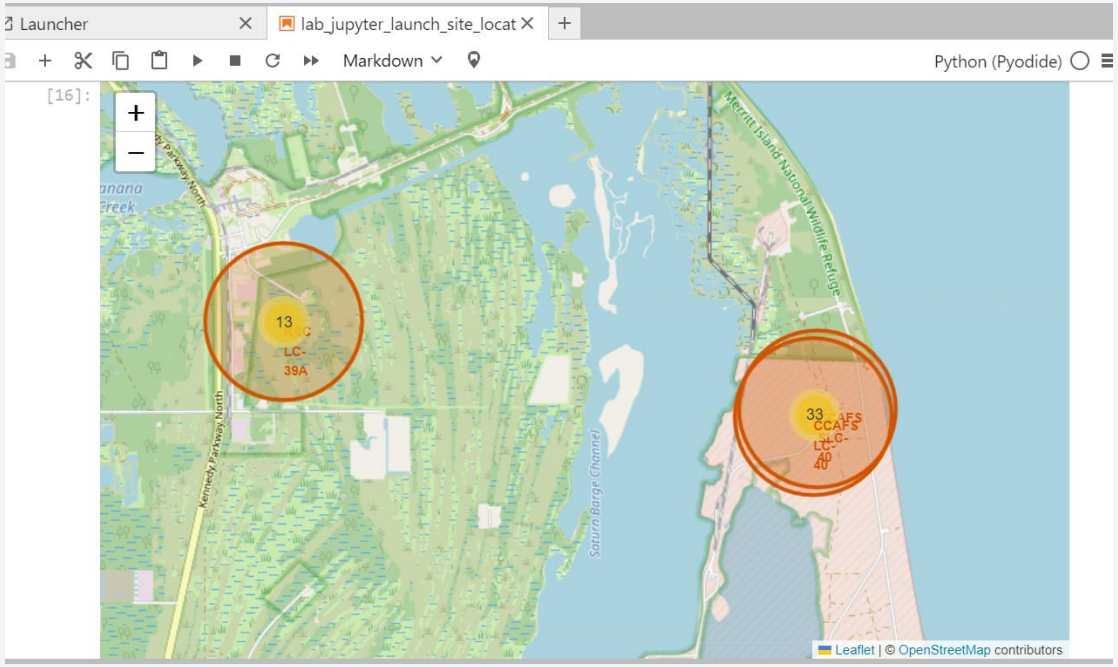
https://github.com/1abhijitdas1/IBMDataScience/blob/61c43b7231f5ca0c74a54ad678b1c5acc75d74a5/SpaceX_Machine%20Learning%20Prediction.ipynb

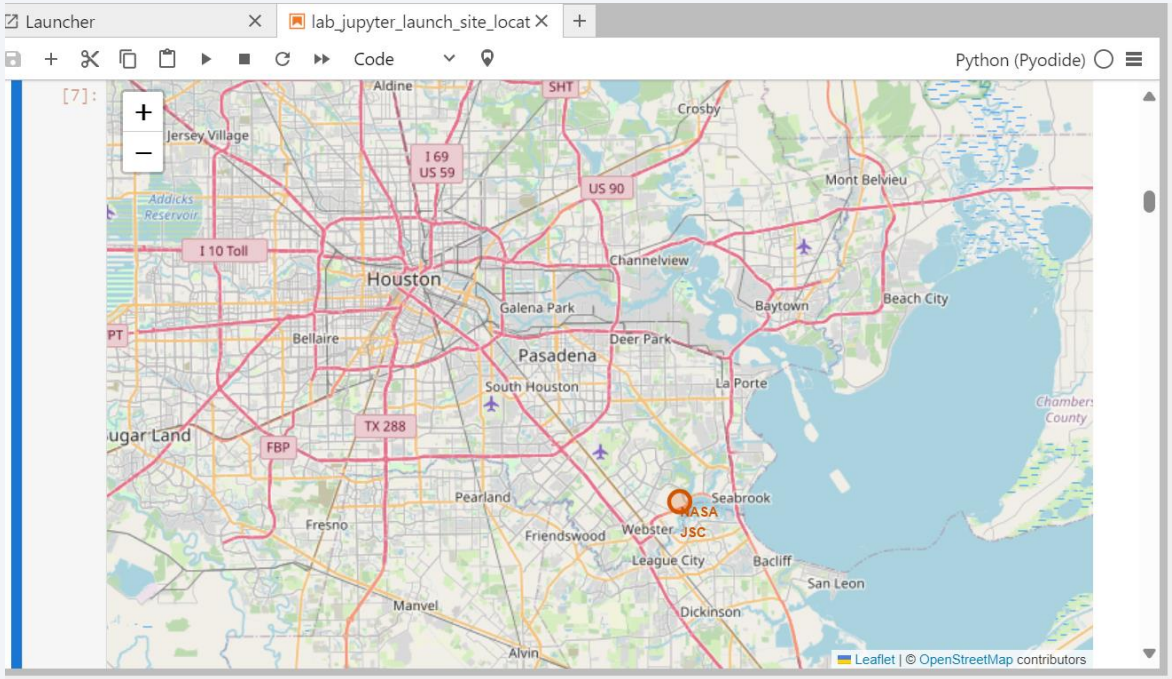
Results - EDA


Success rate
1.0
0.8
0.6
0.4
0.2
0.0
LEO
ISS
PO
GTO
ES-L1
SSO
HEO
MEO
VLEO
SO
GEO
Orbit

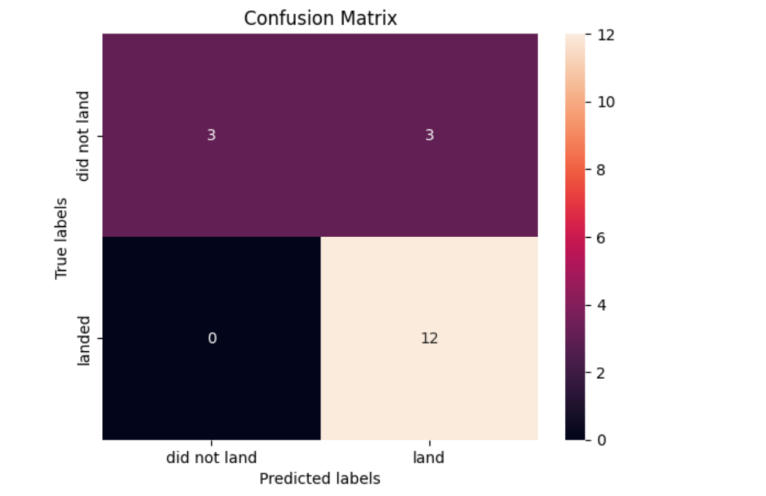

Class
0
1
Orbit
LEO
ISS
PO
GTO
ES-L1
SSO
HEO
MEO
VLEO
SO
GEO
0
20
40
60
80
Flight Number

Results - Folium





Results – ML Classification



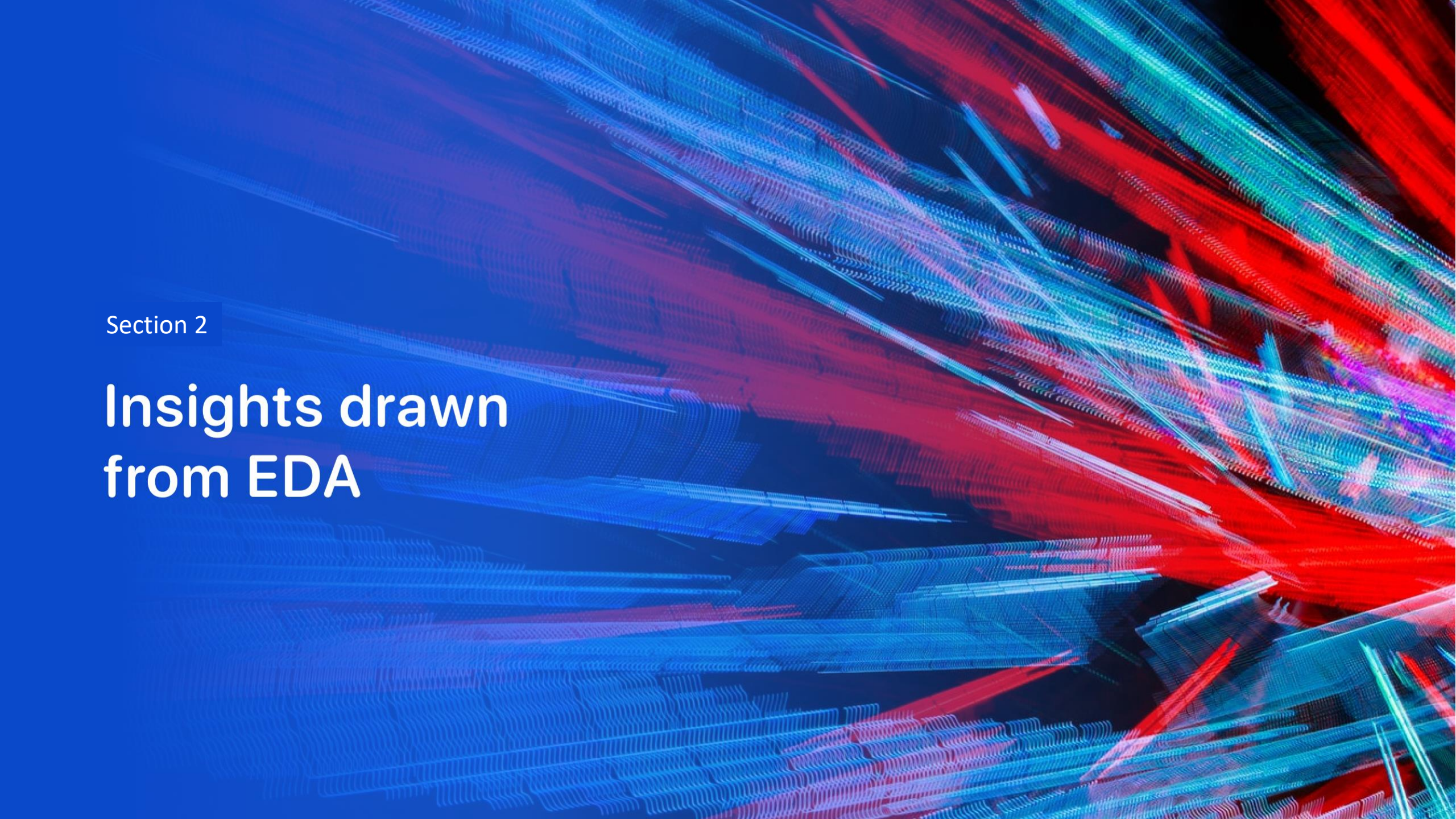
Find the method performs best:

In [39]:

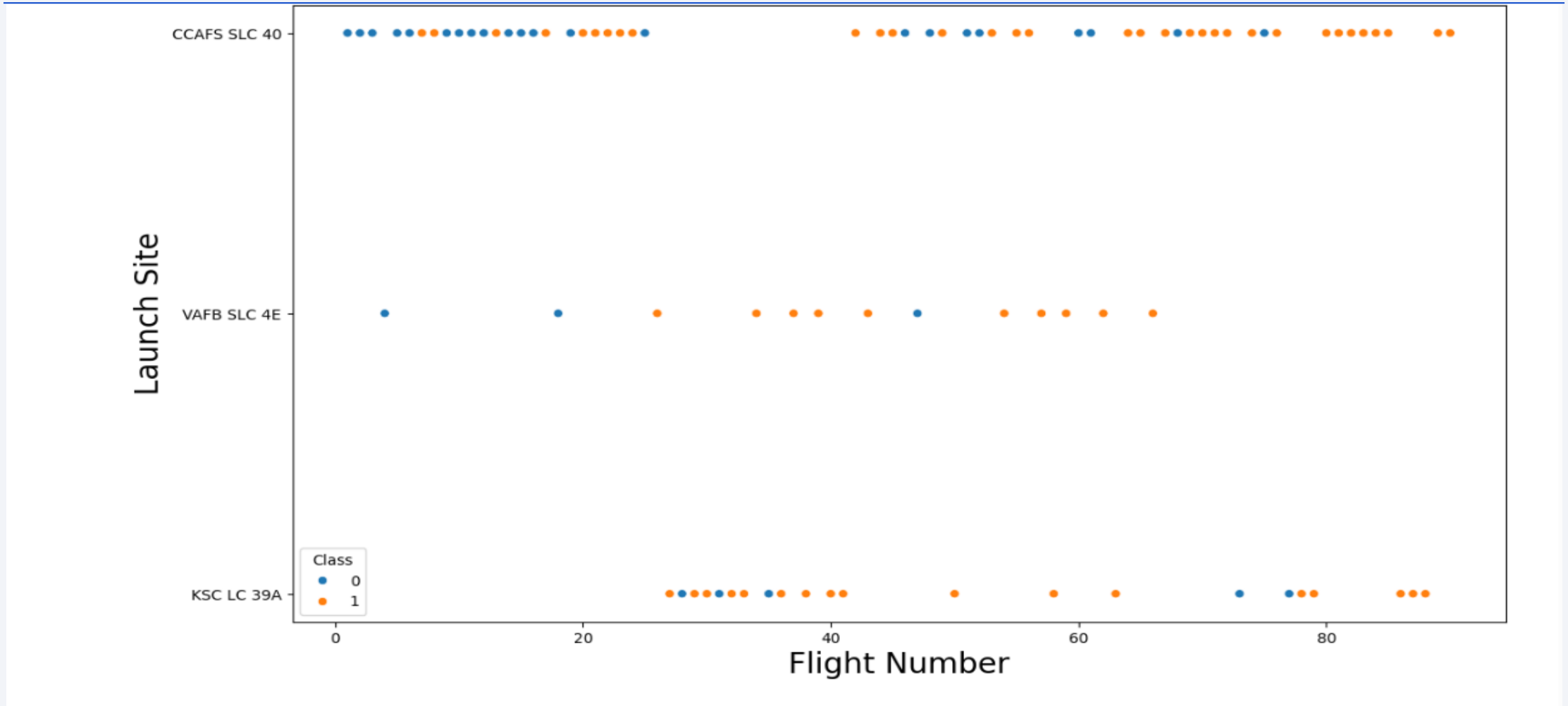
```
# Since their accuracies are all the same, we pick based on their best scores
# Since their accuracies are all the same, we pick based on their best scores
alg_score = {'Logistic regresssion': [logreg_cv.best_score_], 'SVM': [svm_cv.best_score_], 'Decision tr
df = pd.DataFrame.from_dict(alg_score, orient='index', columns=['Best scores'])
df
```

Out[39]:

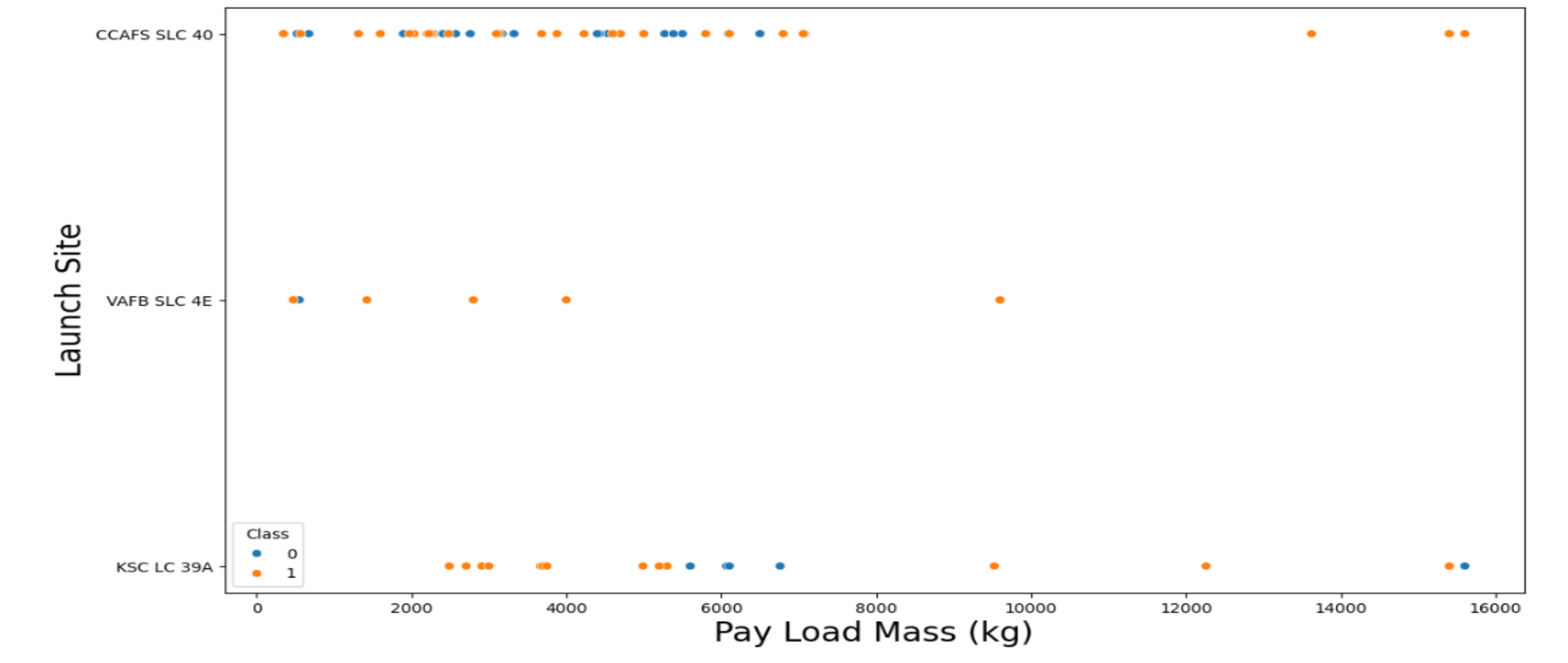
	Best scores
Logistic regresssion	0.846429
SVM	0.848214
Decision tree	0.876786
KNN	0.848214


Section 2
Insights drawn from EDA

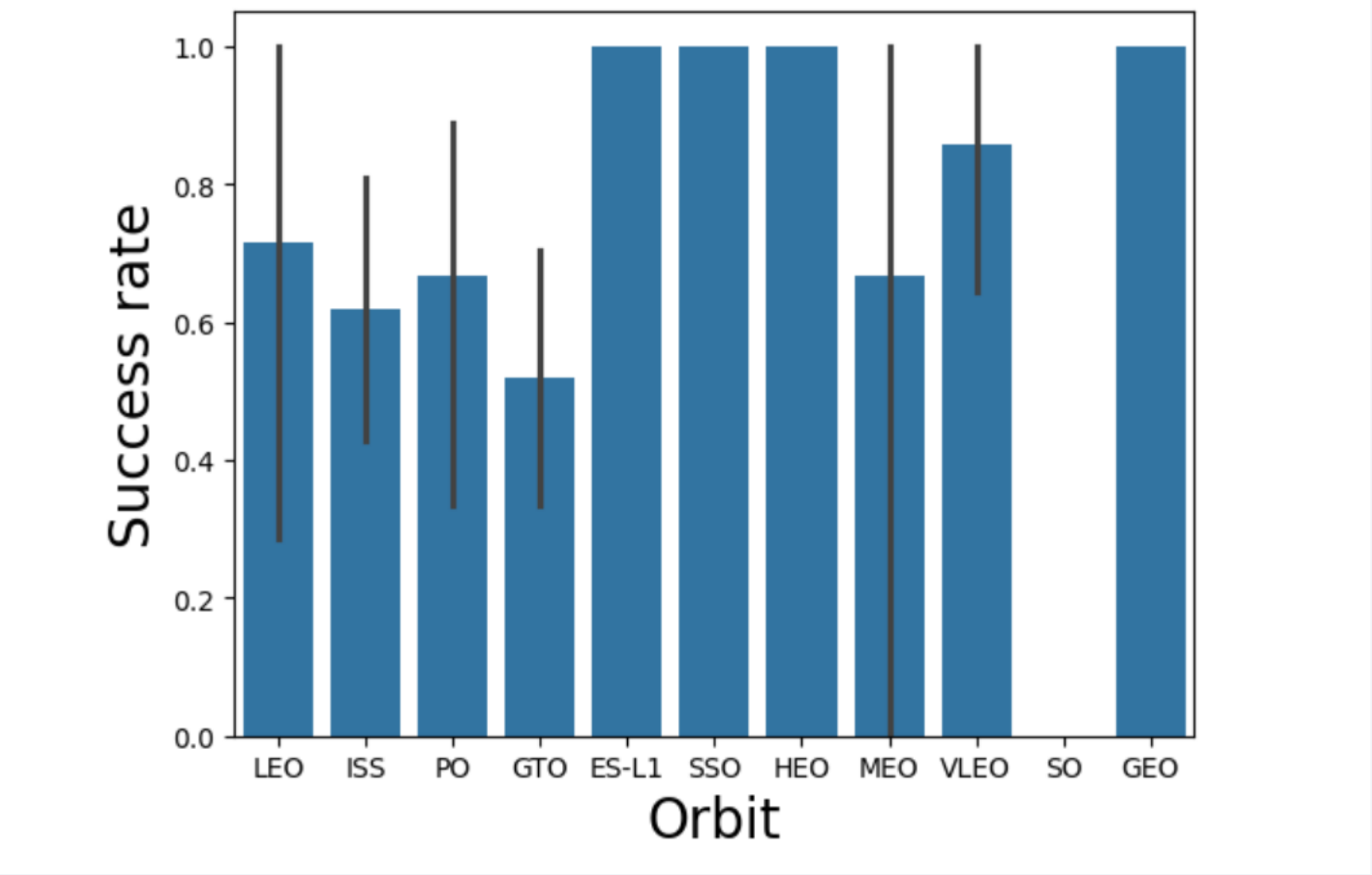
Flight Number vs. Launch Site



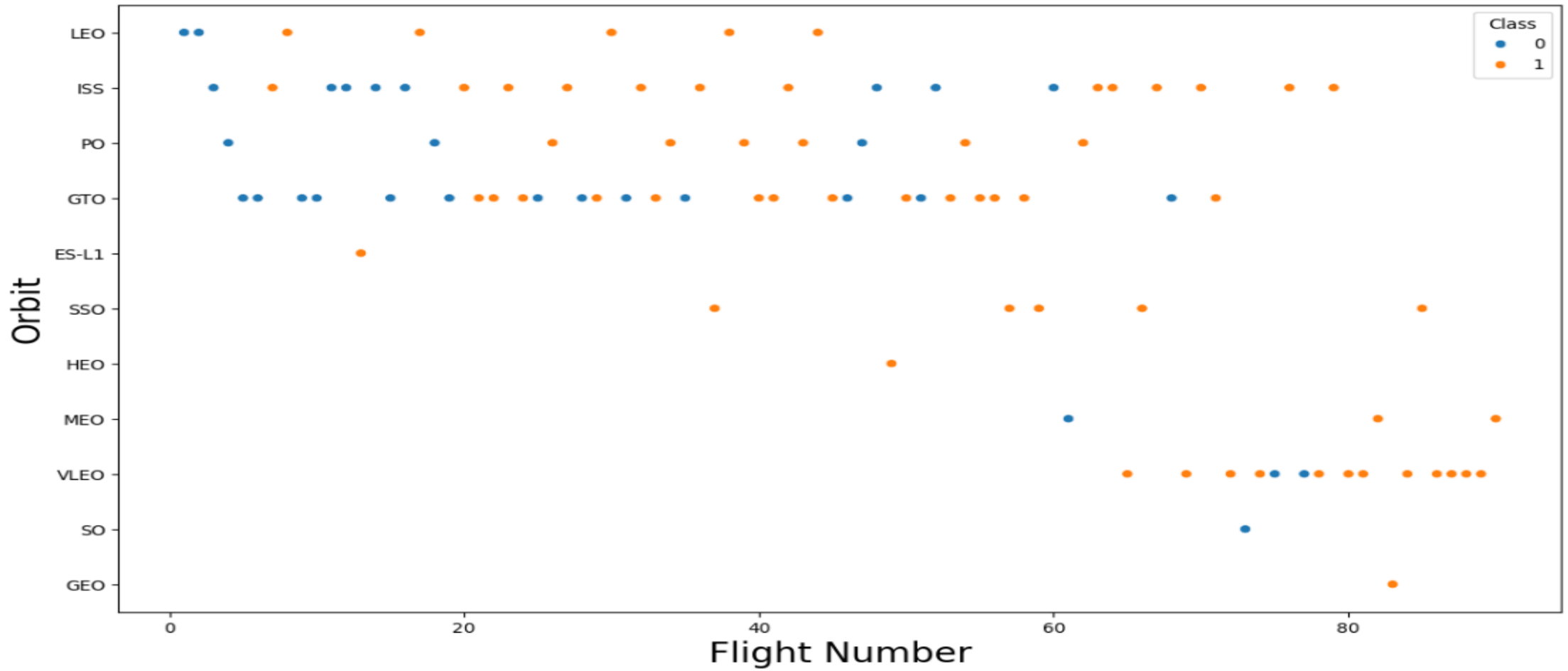
Payload vs. Launch Site



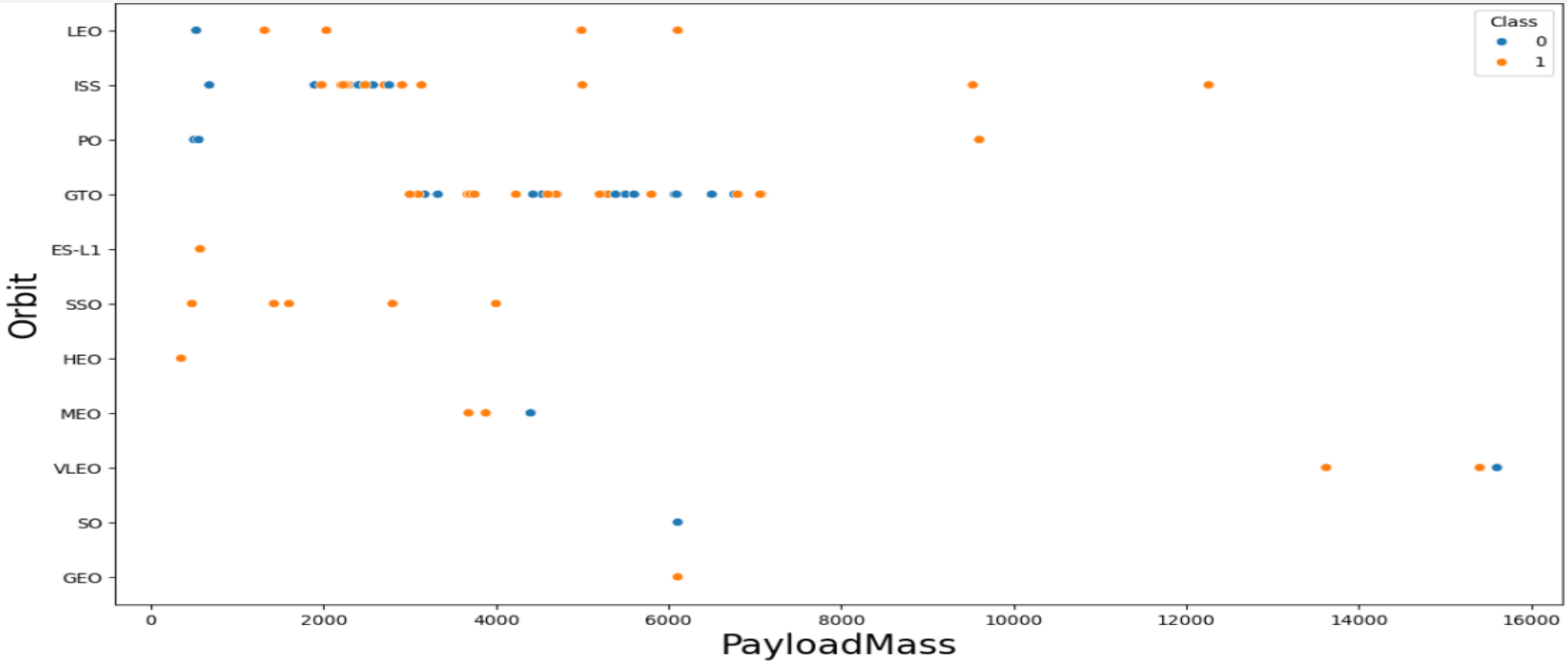
Success Rate vs. Orbit Type



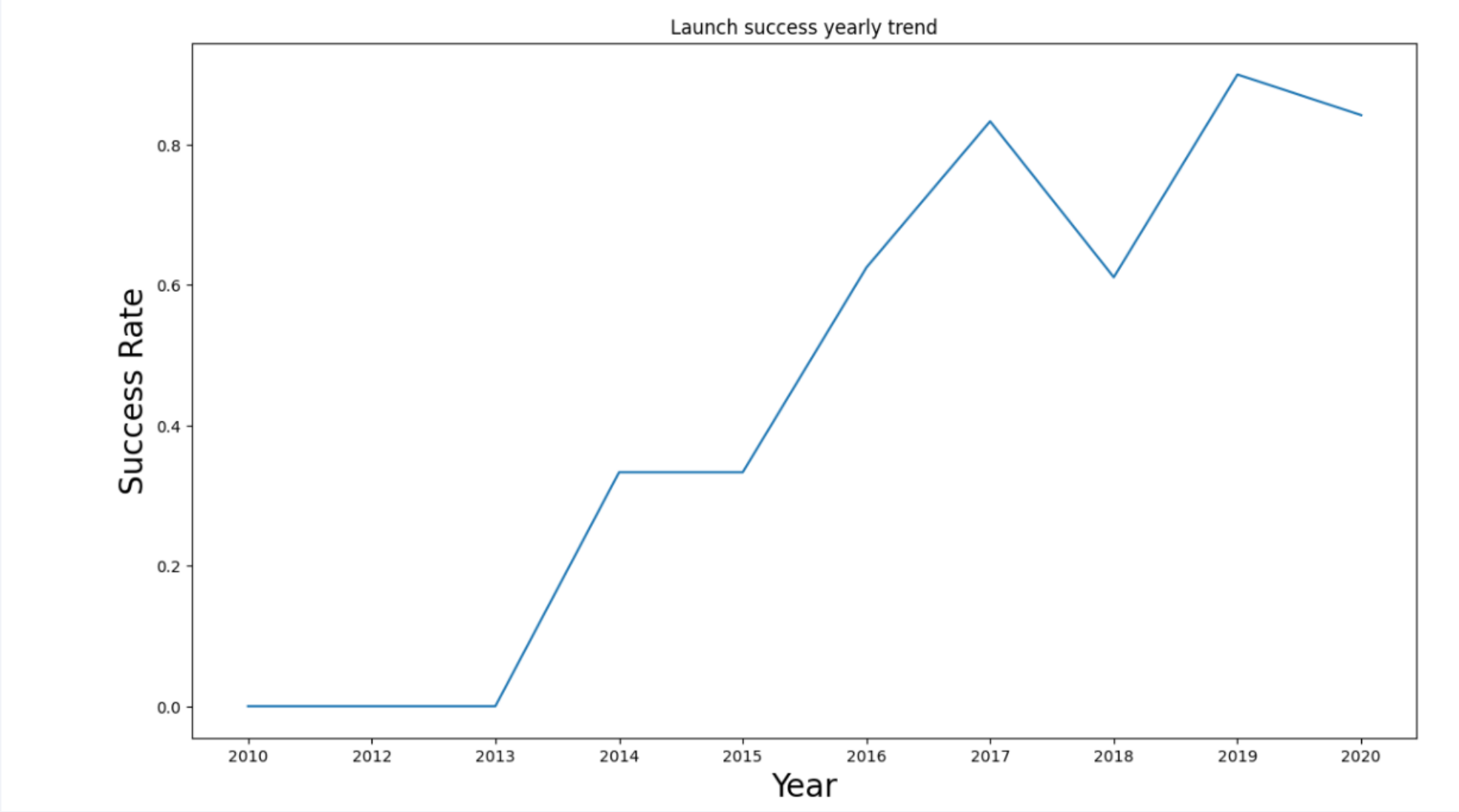
Flight Number vs. Orbit Type



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 [24]:
```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

Out[24]:

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

Launch Site Names Begin with 'CCA'

Out[25]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcc
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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Succ
2012-10-08	0:35:00	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 [26]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total payload mass by NASA (CRS)" FROM SPACEXTBL WHERE CUSTOMER
```

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

Out[26]:

Total payload mass by NASA (CRS)
45596

Average Payload Mass by F9 v1.1

Task 4

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

In [27]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average payload mass by Booster Version F9 v1.1" FROM SPACEXTBL
```

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

Out[27]:

Average payload mass by Booster Version F9 v1.1
2928.4

First Successful Ground Landing Date

Task 5

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

Hint:Use min function

In [30]:
```
%sql SELECT MIN(DATE) AS "Date of first successful landing outcome in ground pad" FROM SPACEXTBL WHERE
```

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

Out[30]:

Date of first successful landing outcome in ground pad
2015-12-22

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 [32]:
```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (drone ship)' AND PAYLOAD_M
```

```
 * sqlite:///my_data1.db
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 [33]:
```
%sql SELECT number_of_success_outcomes, number_of_failure_outcomes FROM (SELECT COUNT(*) AS number_of_s
```

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

Out[33]:

number_of_success_outcomes	number_of_failure_outcomes
100	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 [34]:
```
%sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__
```

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

Out[34]:

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

2015 Launch Records

Task 9

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

In [37]:
```
%sql SELECT substr(Date, 6,2) as month, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE substr(Date,6
```

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

Out[37]:

month	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	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 [41]:
```
%sql SELECT LANDING_OUTCOME, COUNT(LANDING_OUTCOME) AS Landing_Count FROM SPACEXTBL WHERE DATE BETWEEN
```

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

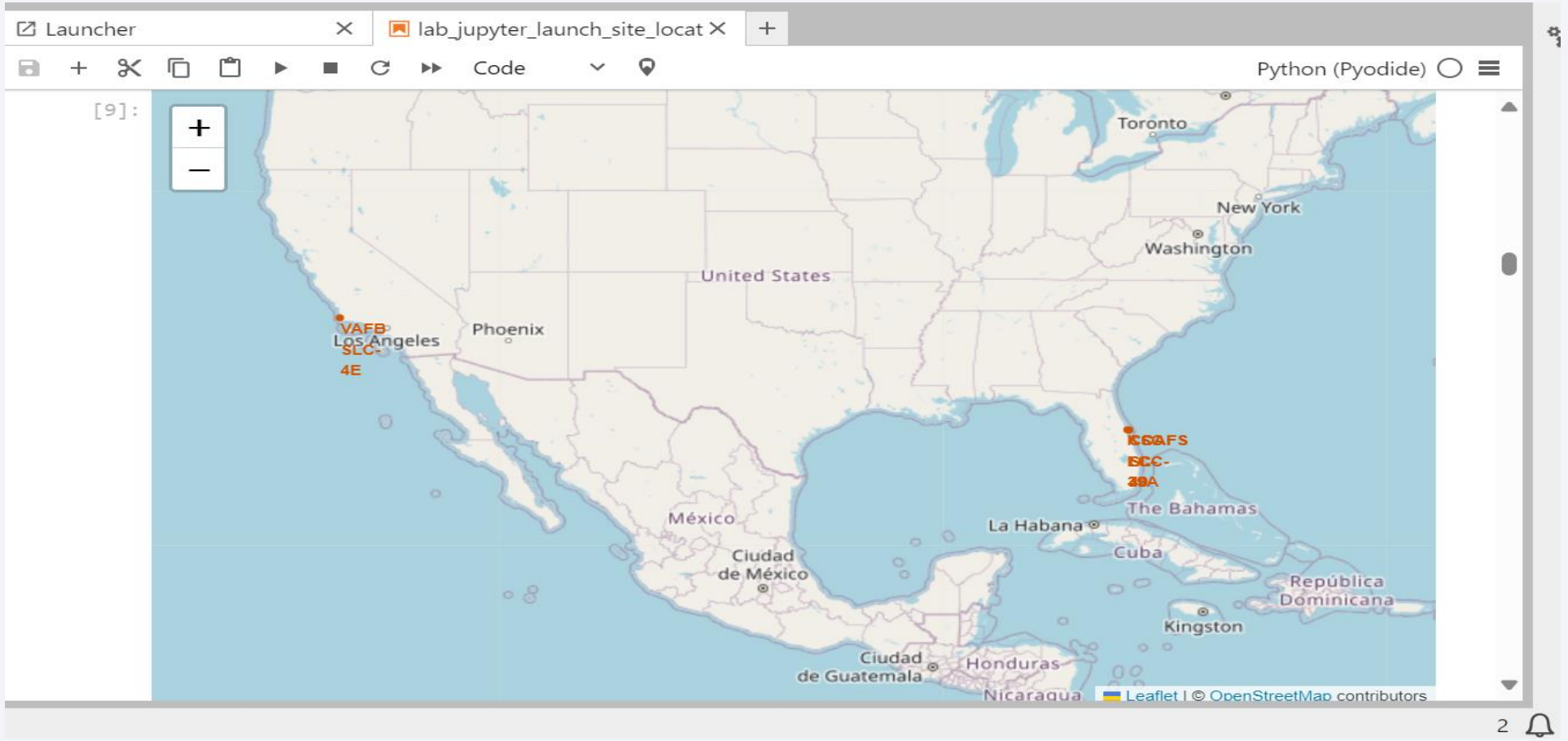
Out[41]:

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

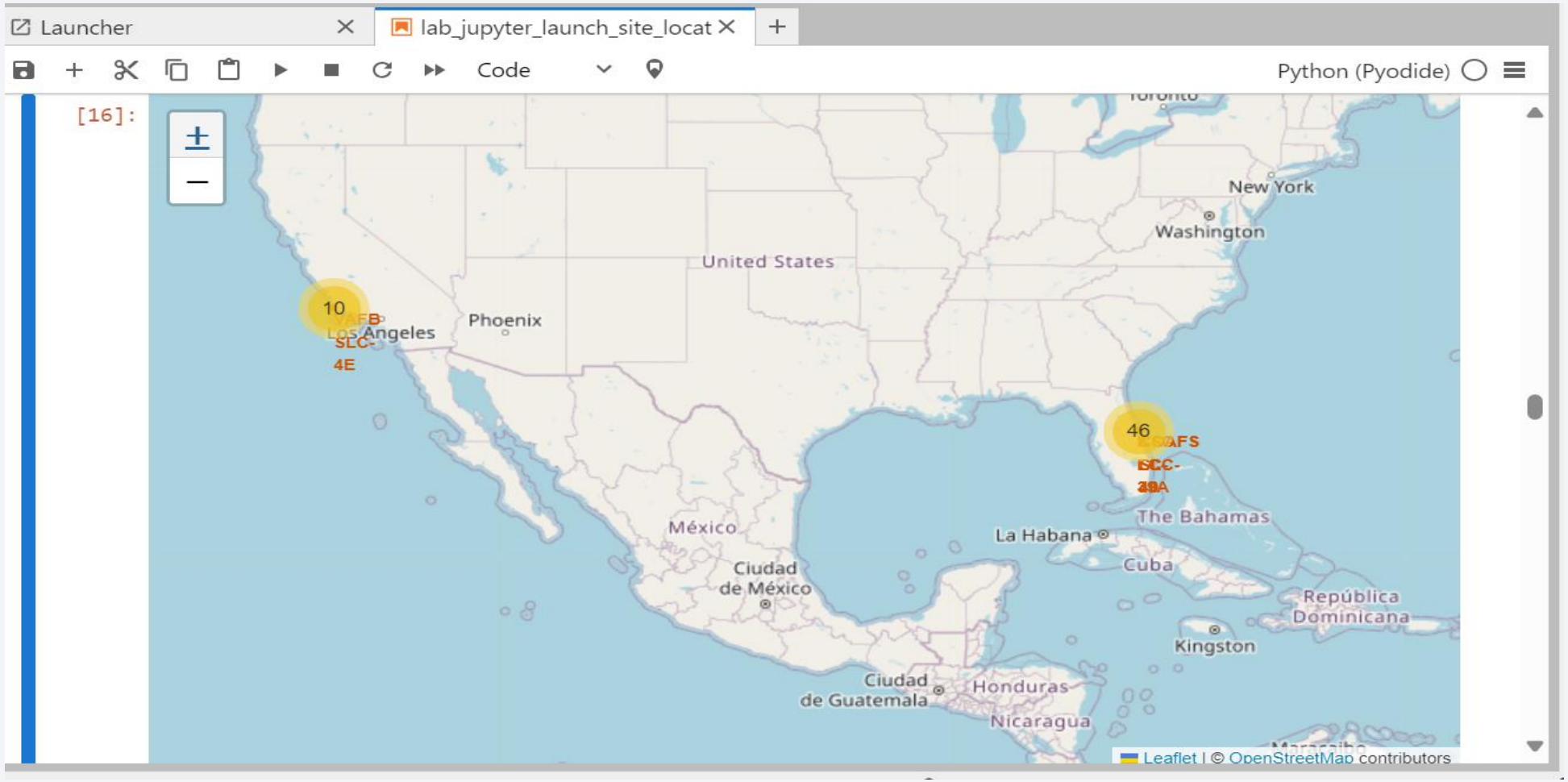
Section 3

Launch Sites Proximities Analysis

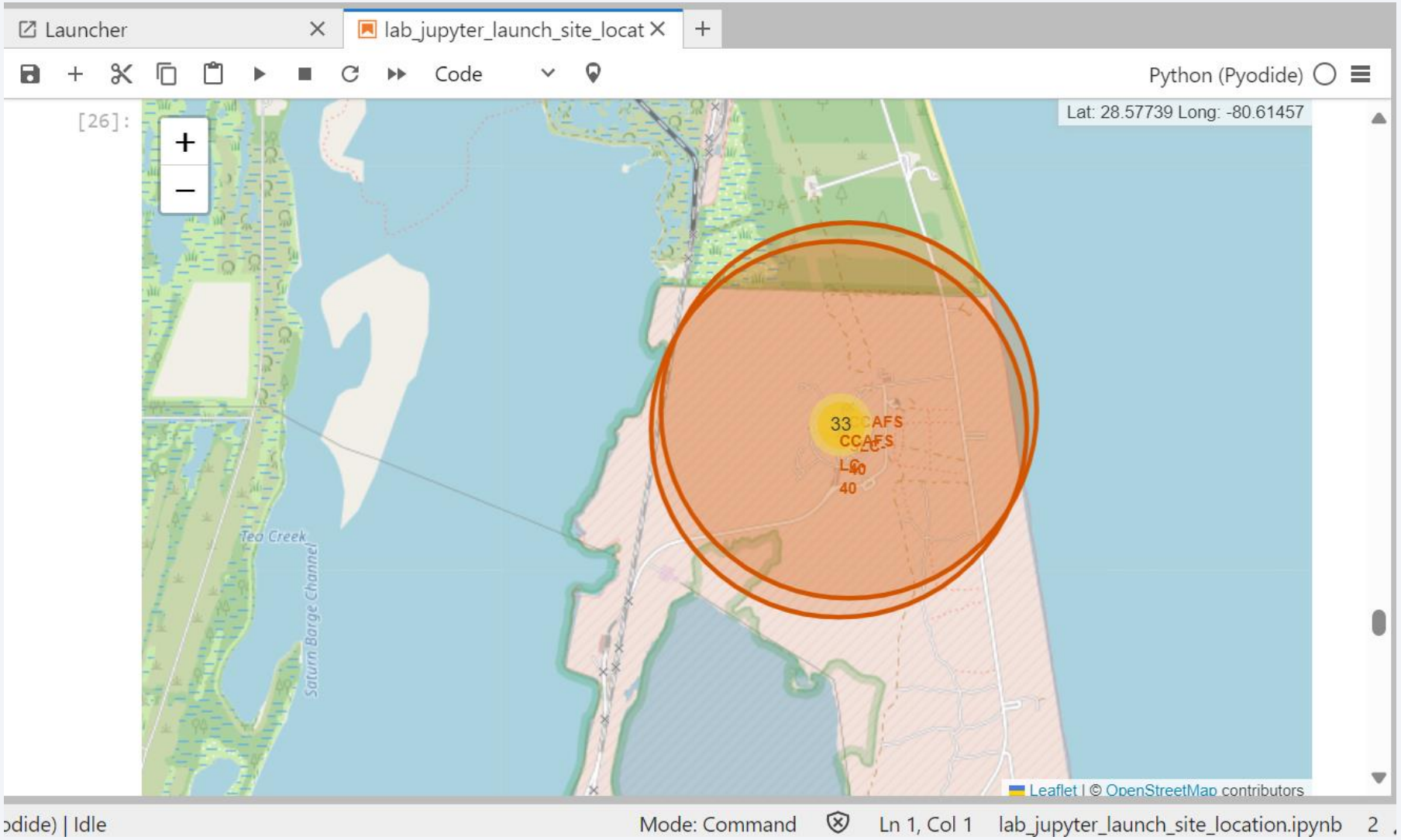
<Folium Map Screenshot 1>

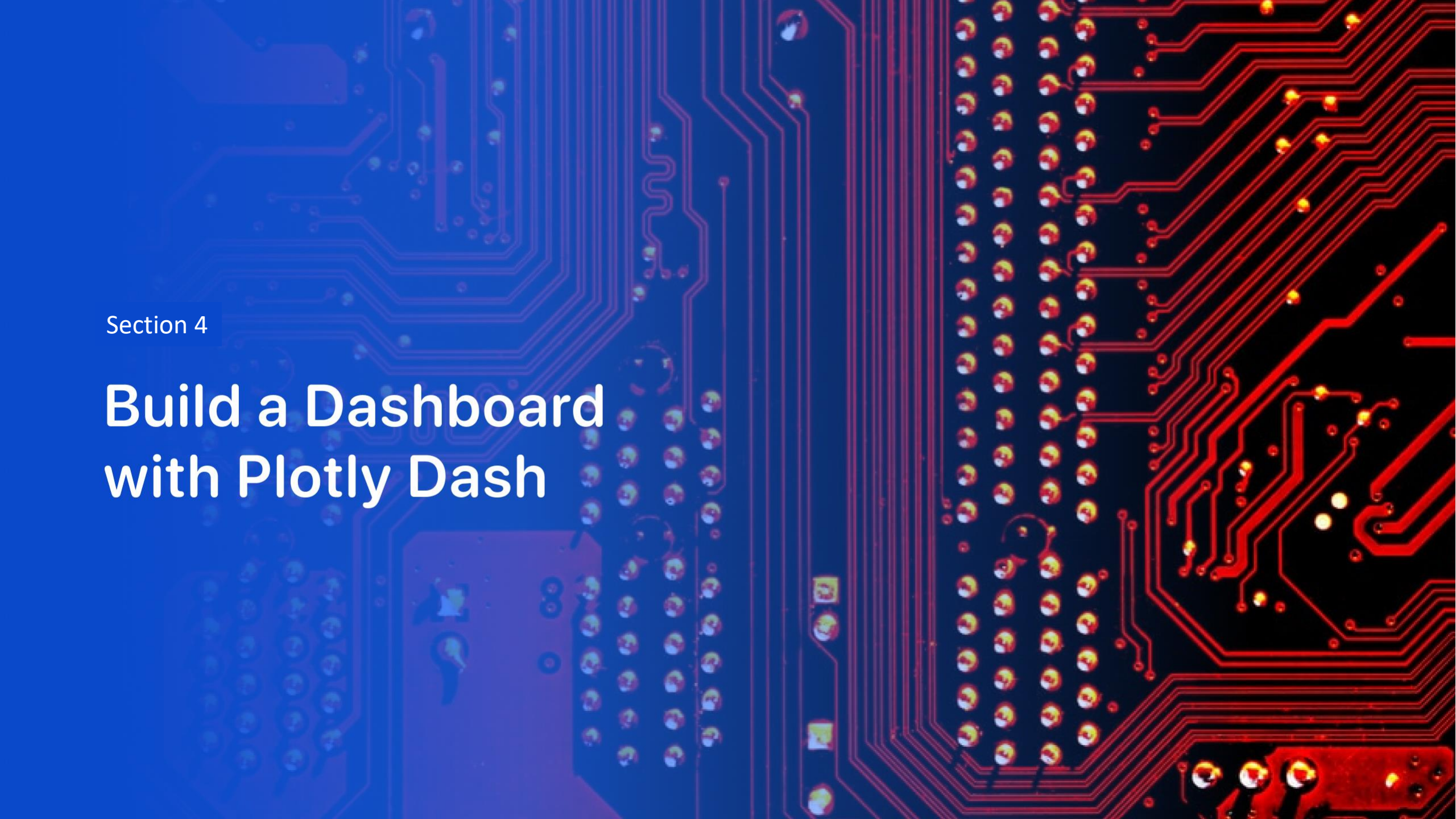


<Folium Map Screenshot 2>

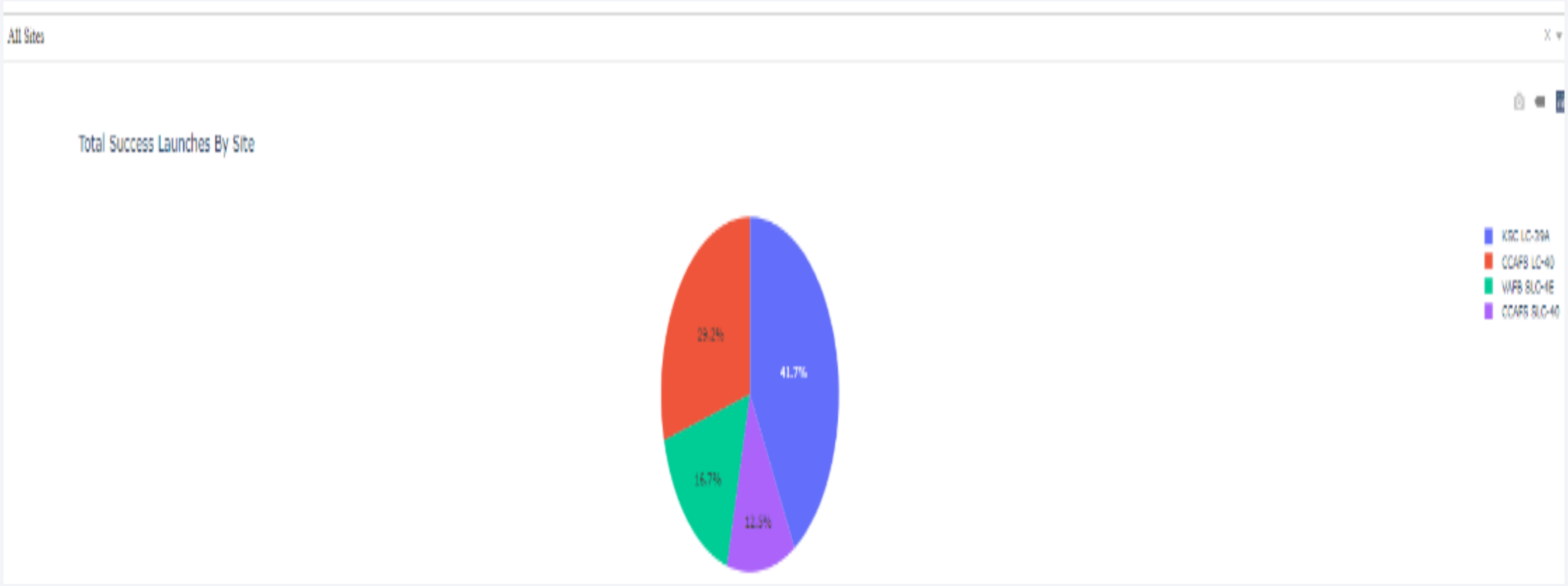


<Folium Map Screenshot 3>

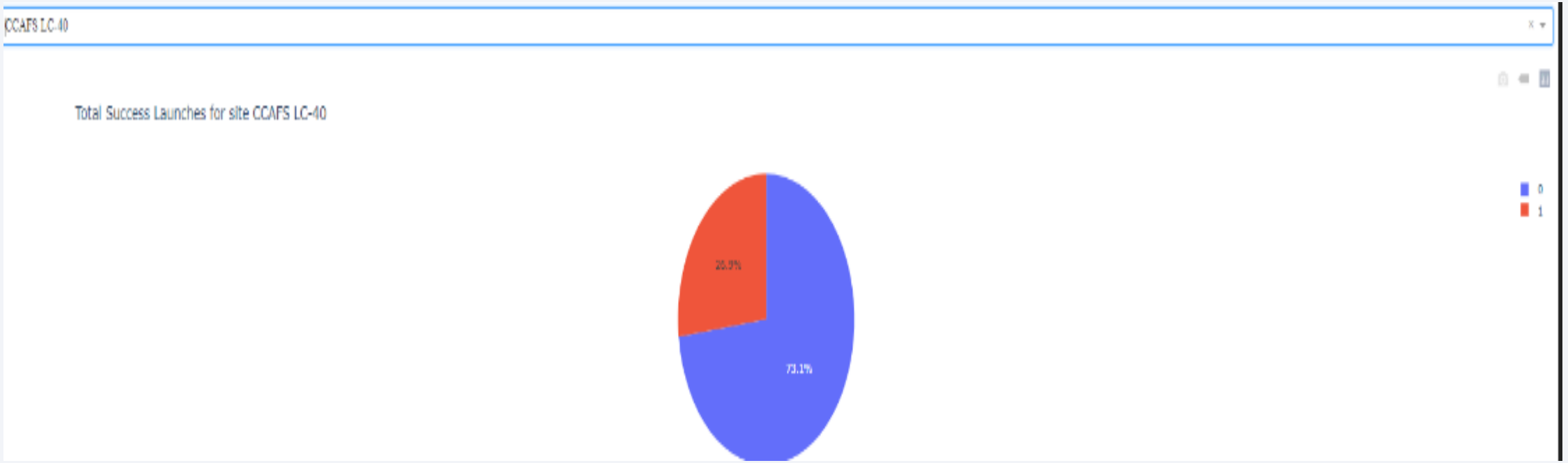



Section 4
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>



<Dashboard Screenshot 2>



<Dashboard Screenshot 3>

Section 5

Predictive Analysis (Classification)

Classification Accuracy

TASK 12

Find the method performs best:

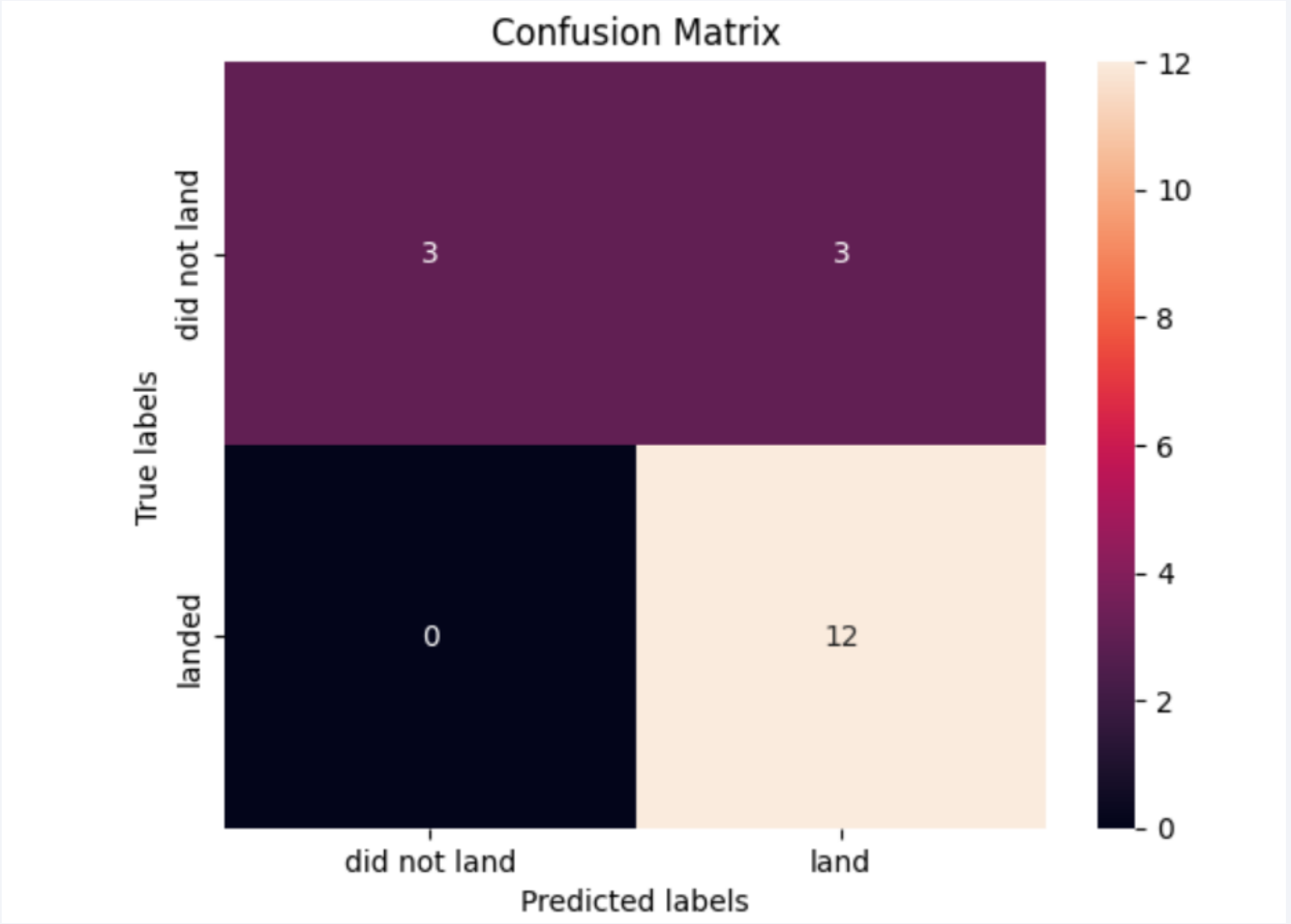
In [39]:

```
# Since their accuracies are all the same, we pick based on their best scores
# Since their accuracies are all the same, we pick based on their best scores
alg_score = {'Logistic regresssion': [logreg_cv.best_score_], 'SVM': [svm_cv.best_score_], 'Decision tr
df = pd.DataFrame.from_dict(alg_score, orient='index', columns=['Best scores'])
df
```

Out[39]:

	Best scores
Logistic regresssion	0.846429
SVM	0.848214
Decision tree	0.876786
KNN	0.848214

Confusion Matrix for Decision Tree



Conclusions

“Decision Tree purportedly is the best ML Algorithm to predict the outcome. This follows from looking at the evaluation metrics of all ML Algorithms in the Jupyter Notebook”

Appendix

- GIT HUB link to Labs: [1abhijitdas1/IBMDataScience: CapstoneProjectFiles (github.com)]

Thank you!