IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Koa Chang
12/31/21

Outline

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

Executive Summary

- The data is from Space X API and a Wikipedia Page and the goal is to cleanse this data and use it to build a machine learning model. Many different models are tested with Grid Search CV to obtain the best hyperparameters for each model.
- The 4 specific models used was K Nearest Neighbors, Decision Tree, Support Vector Machine, and Logistic Regression.
- As seen in the data presented later most models preformed the same.

Introduction

- Space is a growing industry
- Costs need to come down in order to democratize space
- Space X is a leading company in this goal as they reuse their first stage falcon 9 to save money
- Our goal is to be able to predict given certain data whether or not the first stage can be recovered or not
    - We will figure this out through a machine learning model
    - Data will be gathered from previous launches

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data collected from SpaceX API and WIkipedia
- Perform data wrangling
  - Classify 1 as successful and 0 as failure.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Use GridSearchCv to find hyperparameters.

Data Collection – SpaceX API

- GIthub url: https://github.com/MalamaPono/IBM-Data-Science-Specialization/blob/main/Capstone%20Project%20(Course%2010)/Data%20Collection%20API.ipynb

- 1. Get data from SpaceX API's
- 2.Filter out only Falcon 9 Launches
- 3. Get Data from Wikipedia
- 4.Handle Missing Values

Data Collection - Scraping

- Github URL: https://github.com/MalamaPono/IBM-Data-Science-Specialization/blob/main/Capstone%20Project%20(Course%2010)/Data%20Collection%20Web%20Scraping.ipynb

1. Request the Wikipedia page
2. Make beautiful soup object
3. Locate table
4. Scrape table into dataframe

Data Wrangling

- Github URL: https://github.com/MalamaPono/IBM-Data-Science-Specialization/blob/main/Capstone%20Project%20(Course%2010)/Data%20Wrangling.ipynb

- 1.Label success with 1 and failure with 0
- 2.Map the different categorical variables to numbers with one hot encoding

EDA with Data Visualization

- Github URL: https://github.com/MalamaPono/IBM-Data-Science-Specialization/blob/main/Capstone%20Project%20(Course%2010)/EDA%20with%20Data%20Visualization.ipynb

- Used many different plots to compare relationships between variables to see how they would affect our training label of 'Class'

EDA with SQL

- 1.Queried unique station sites
- 2.Queries payload mass
- 3.Queried booster version value counts
- 4.Queried most recent launches
- 5.Queried average values
- Github URL: https://github.com/MalamaPono/IBM-Data-Science-Specialization/blob/main/Capstone%20Project%20(Course%2010)/EDA%20With%20SQL.ipynb

Build an Interactive Map with Folium

- I used a distance line to the city of Lombon
- I used a distance line to a coastline
- I used a distance line to highway
- I used a distance line to a railway
- Github URL: https://github.com/MalamaPono/IBM-Data-Science-Specialization/blob/main/Capstone%20Project%20(Course%2010)/Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Build a Dashboard with Plotly Dash

- Github URL: https://github.com/MalamaPono/IBM-Data-Science-Specialization/blob/main/Capstone%20Project%20(Course%2010)/Dataset%20Graph%20Dashboard.ipynb
- This notebook uses an interactive dropdown menu to pick a specific launch location for the rocket. It also uses a slider to select a range of payload mass. It then displays this information in a pie chart and a scatter plot to visualize and make conclusions about variables.

Predictive Analysis (Classification)

- Github URL: https://github.com/MalamaPono/IBM-Data-Science-Specialization/blob/main/Capstone%20Project%20(Course%2010)/Machine%20Learning%20Prediction.ipynb

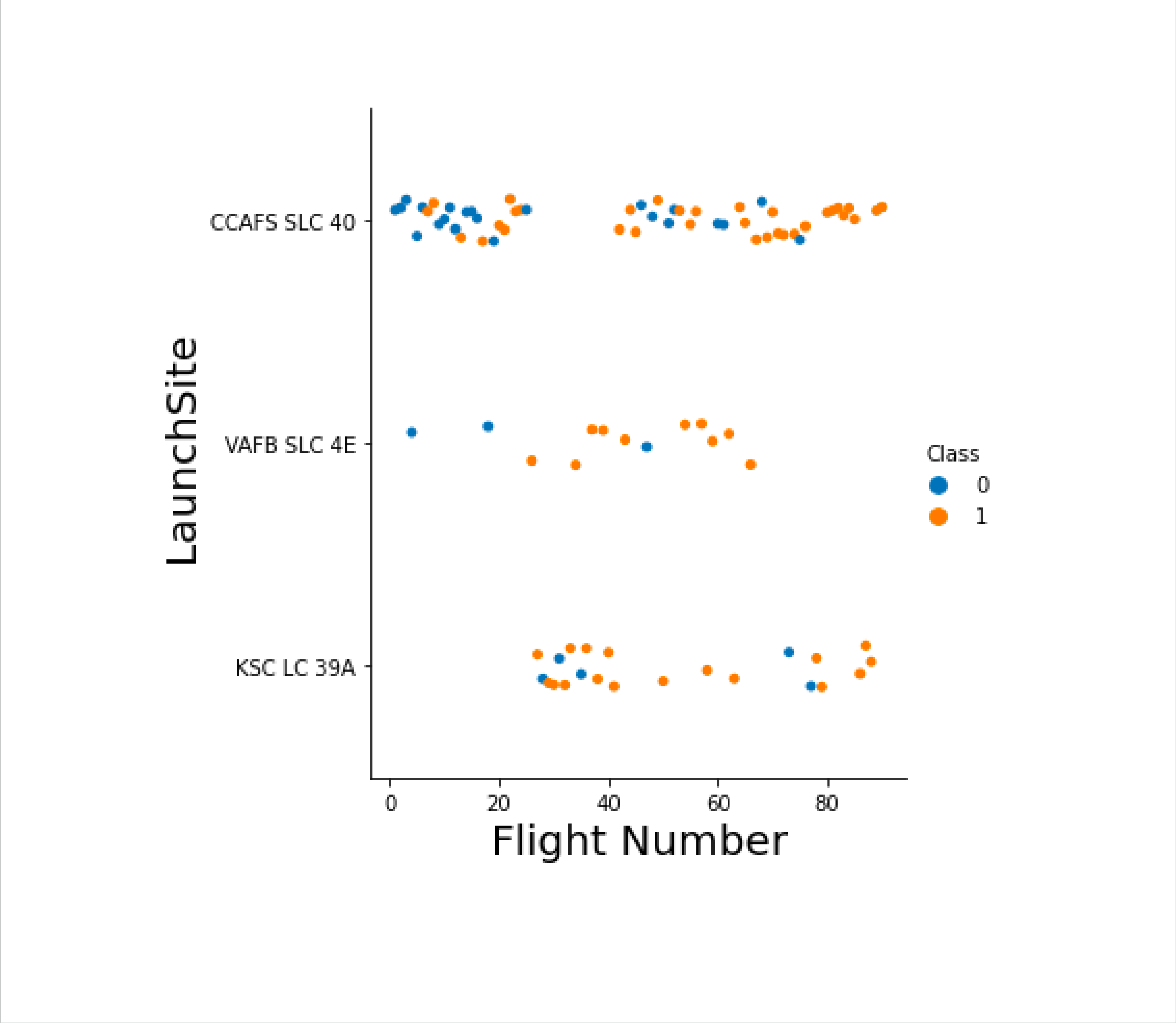
- Perform exploratory Data Analysis and determine Training Labels
- create a column for the class
- Standardize the data
- Split into training data and test data
- -Find best Hyperparameter for SVM, Classification Trees and Logistic Regression
- Find the method performs best using test data

Section 2

Insights drawn from EDA

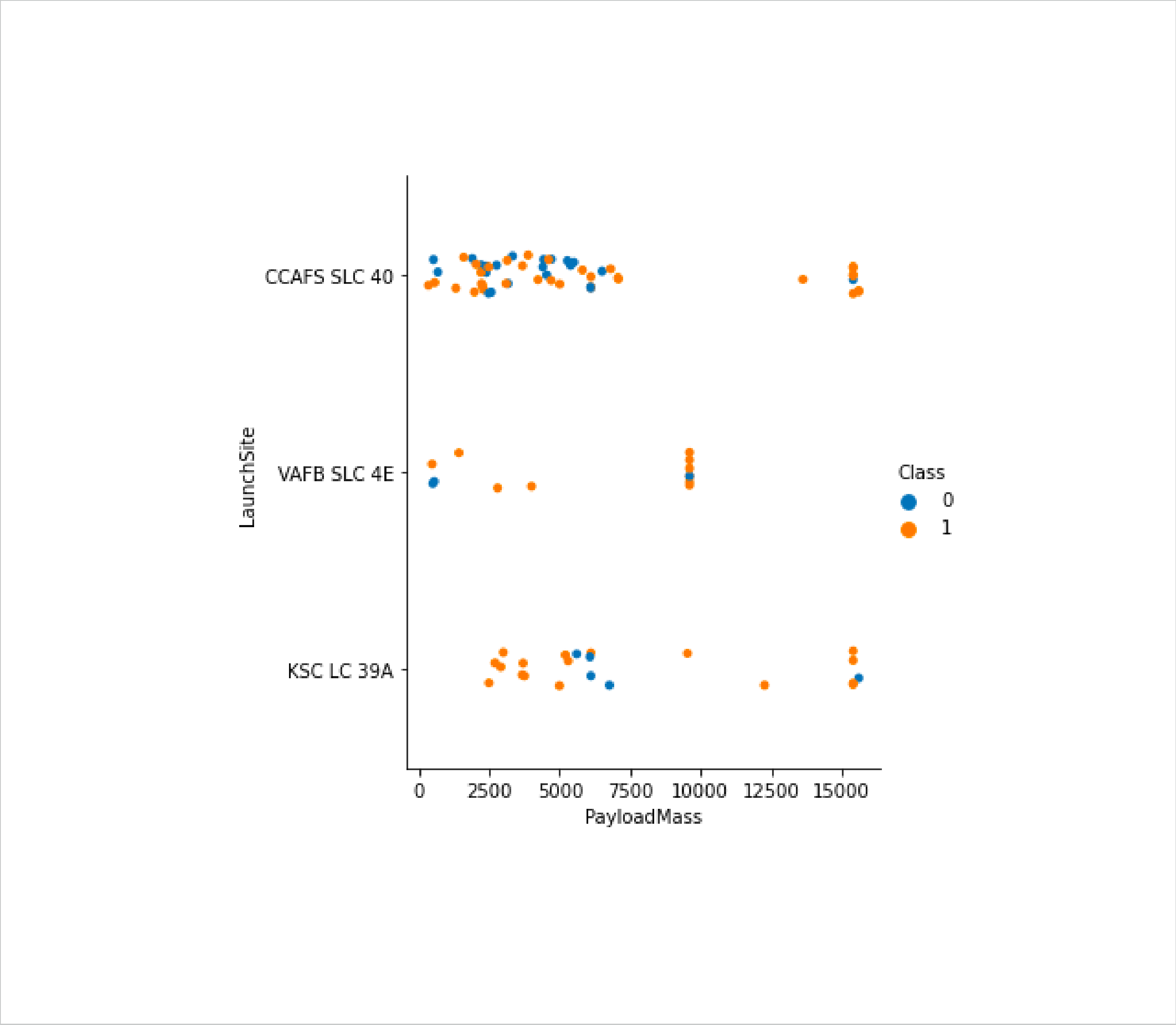
Flight Number vs. Launch Site

- As flight number goes up the launch site and its outcome varies



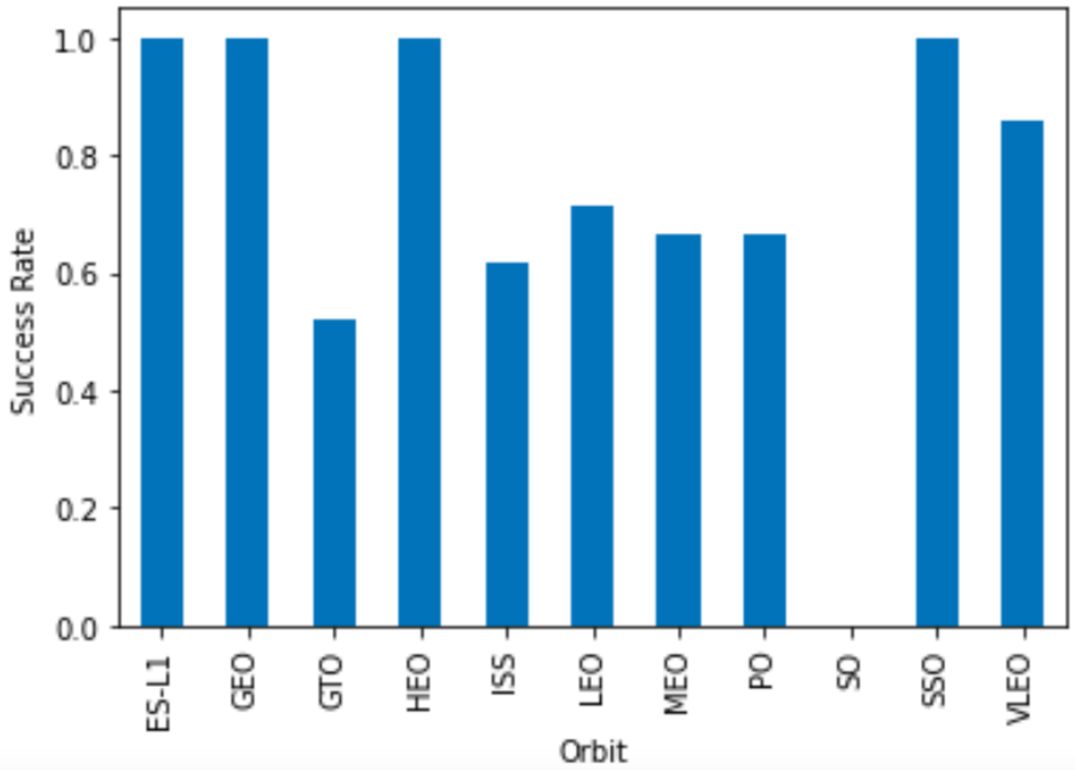
Payload vs. Launch Site

- Most payloads are between 0 and 6000 and as the payload goes up there are usually more failures



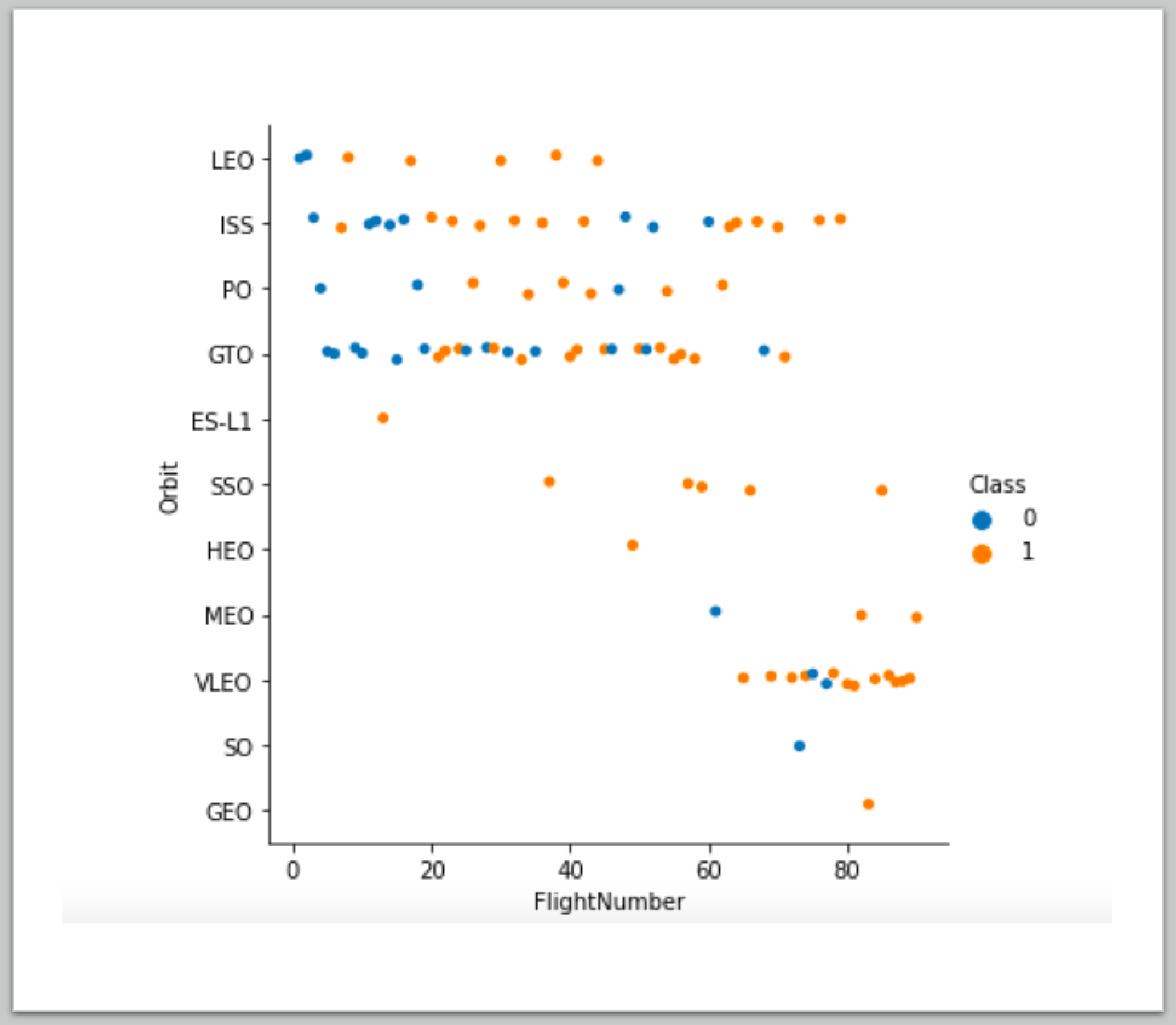
Success Rate vs. Orbit Type

- Different orbit types have different success rates



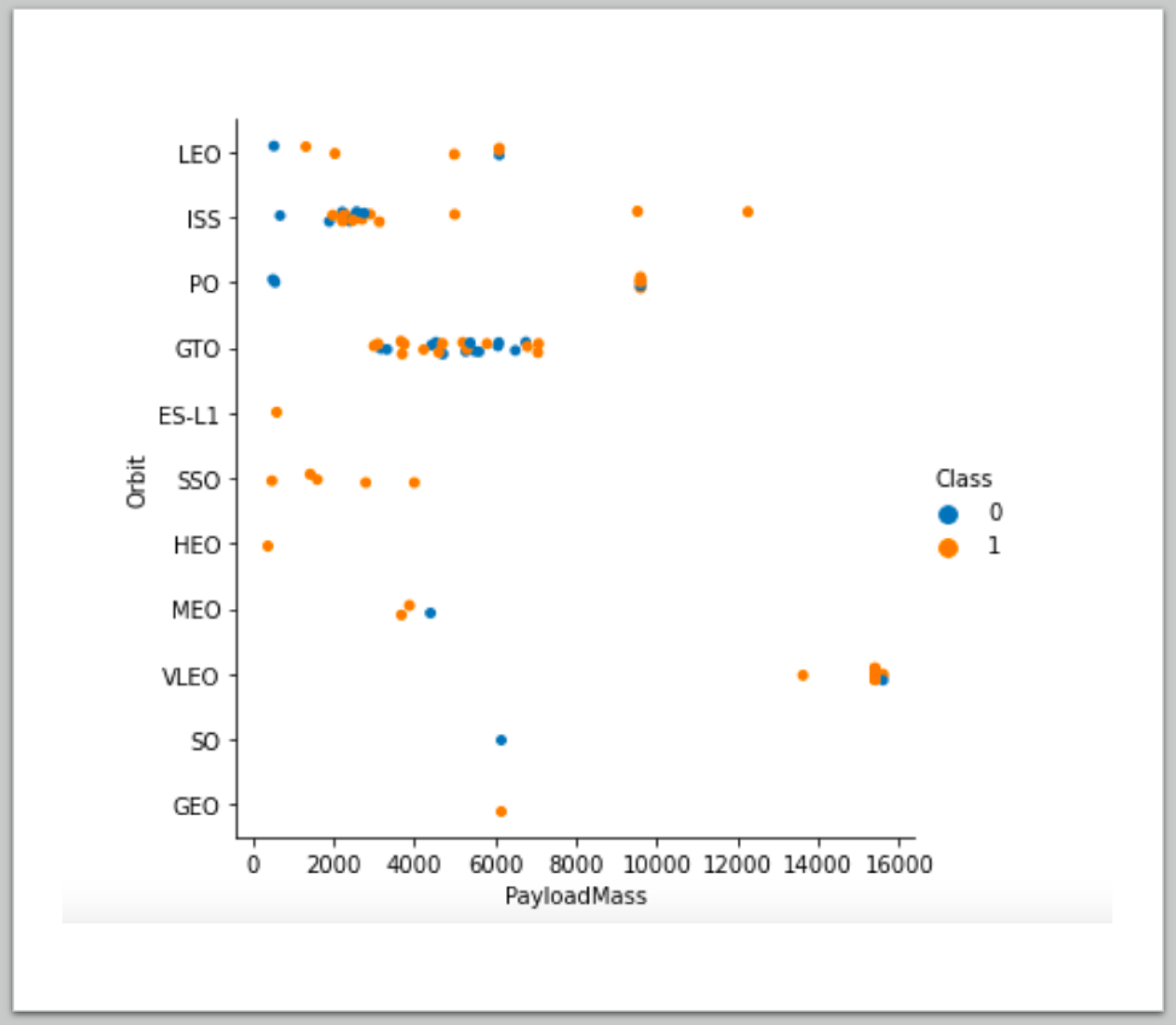
Flight Number vs. Orbit Type

- Seems to have not much correlation



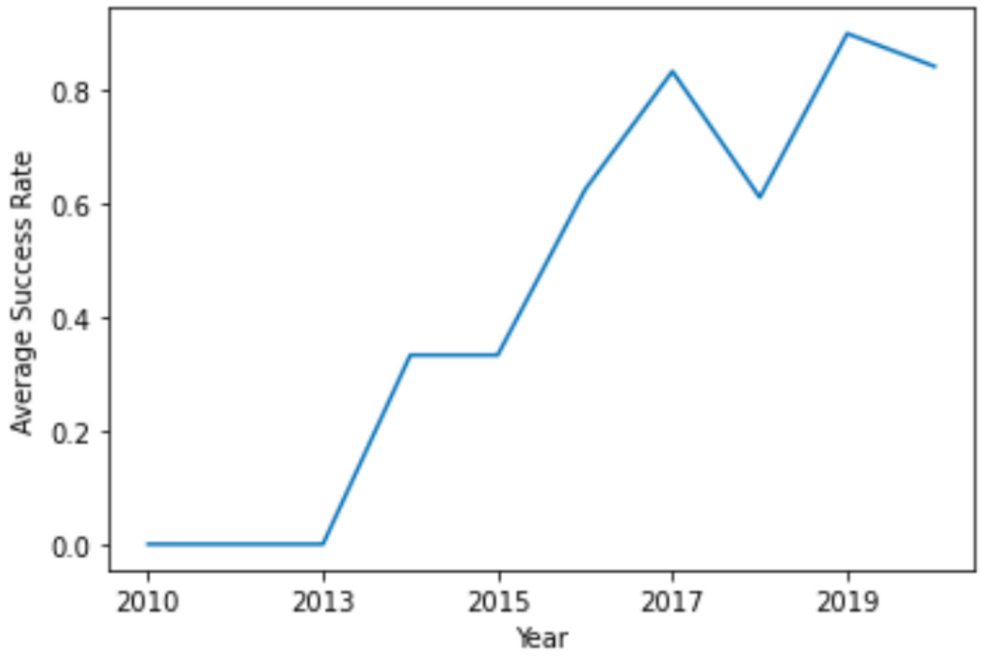
Payload vs. Orbit Type

- Some orbit types are specifically for higher masses and vise versa



Launch Success Yearly Trend

- A the years go on the success rate increases



All Launch Site Names

- Find the names of the unique launch sites

Task 1 ¶

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

```
In [6]: %sql Select Distinct Launch_Site From SPACEXDATASET

 * ibm_db_sa://sxs86238:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:3087
5/BLUDB
Done.
```

Out[6]:

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

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

In [7]:
```
%sql Select * From SPACEXDATASET Where Launch_Site Like 'CCA%' Limit 5
```

```
 * ibm_db_sa://sxs86238:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:3087
5/BLUDB
Done.
```

Out[7]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
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	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

In [8]:
```
%sql Select Payload_mass__kg_ From SPACEXDATASET Where Customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://sxs86238:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:3087
5/BLUDB
Done.
```

Out[8]:

payload_mass__kg_
500
677
2296
2216
2395
1898
1952
3136
2257
2490
2708
3310
2205
2647
2697
2500
2495
2268
1977
2972

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here

Task 4

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

```
In [9]: %sql Select AVG(Payload_mass__kg_) From SPACEXDATASET Where Booster_Version = 'F9 v1.1'

 * ibm_db_sa://sxs86238:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:3087
5/BLUDB
Done.
```

Out[9]:

1
2928

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here

Task 5

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

Hint:Use min function

In [10]:
```
%sql Select Min(Date) From SPACEXDATASET Where Landing__Outcome = 'Success (ground pad)'
```

```
 * ibm_db_sa://sxs86238:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:3087
5/BLUDB
Done.
```

Out[10]:

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- Present your query result with a short explanation here

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 [11]:

```
%sql Select Booster_Version From SPACEXDATASET Where (Landing__Outcome like 'Success (drone ship)') And (payload_mass_
_kg_ between 4000 and 6000)
```

```
 * ibm_db_sa://sxs86238:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:3087
5/BLUDB
Done.
```

Out[11]:

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here

In [13]:
```
%sql SELECT mission_outcome, count(*) as Count FROM SPACEXDATASET GROUP by mission_outcome ORDER BY mission_outcome
```

```
 * ibm_db_sa://sxs86238:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:3087
5/BLUDB
Done.
```

Out[13]:

mission_outcome	COUNT
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

In [15]:

```
maxm = %sql select max(payload_mass__kg_) from SPACEXDATASET
maxv = maxm[0][0]
%sql select booster_version from SPACEXDATASET where payload_mass__kg_=(select max(payload_mass__kg_) from SPACEXDATAS
ET)
```

```
 * ibm_db_sa://sxs86238:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:3087
5/BLUDB
Done.
 * ibm_db_sa://sxs86238:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:3087
5/BLUDB
Done.
```

Out[15]:

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

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

- Present your query result with a short explanation here

In [17]:
```
%sql select MONTHNAME(DATE) as Month, landing__outcome, booster_version, launch_site from SPACEXDATASET where DATE lik
e '2015%' AND landing__outcome like 'Failure (drone ship)'
```

```
 * ibm_db_sa://sxs86238:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:3087
5/BLUDB
Done.
```

Out[17]:

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

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

- 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 [19]:
```
%sql select landing__outcome, count(*) as count from SPACEXDATASET where Date >= '2010-06-04' AND Date <= '2017-03-20'
GROUP by landing__outcome ORDER BY count Desc
```

```
 * ibm_db_sa://sxs86238:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:3087
5/BLUDB
Done.
```

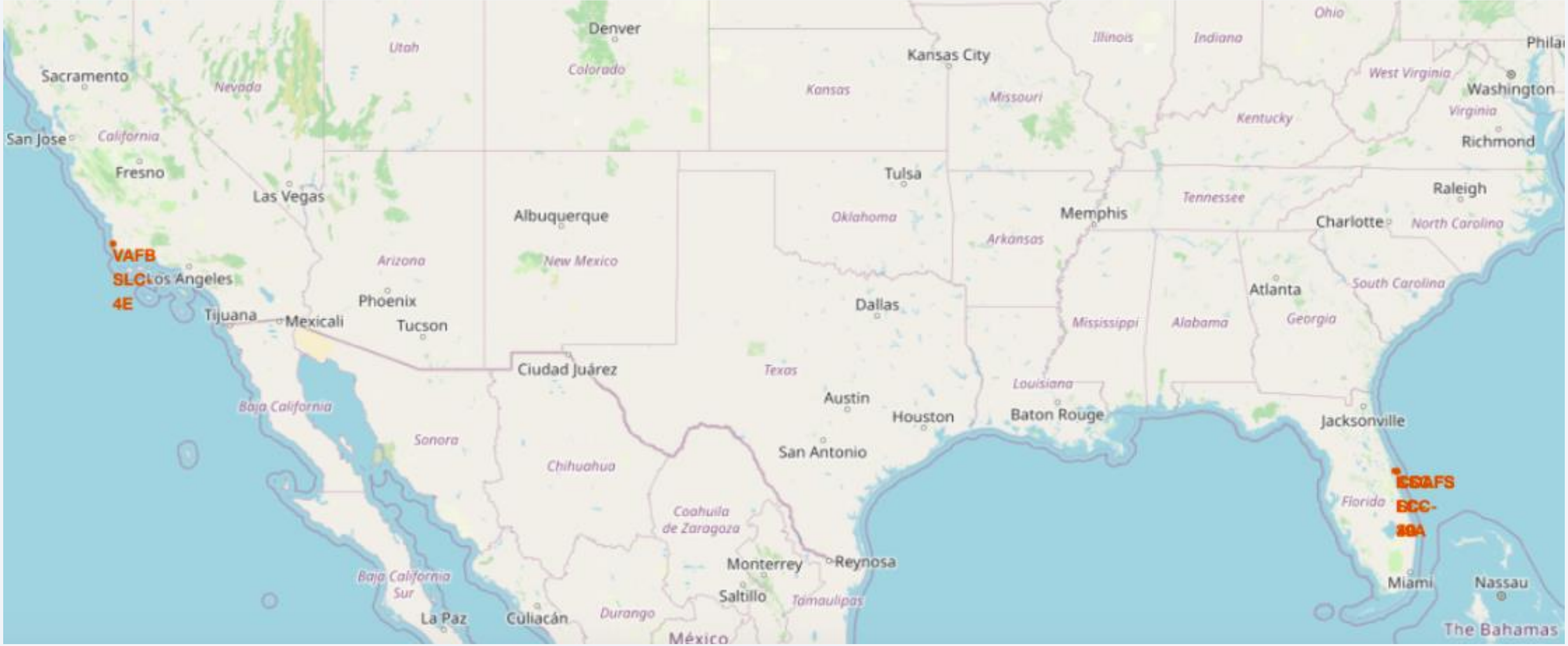
Out[19]:

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

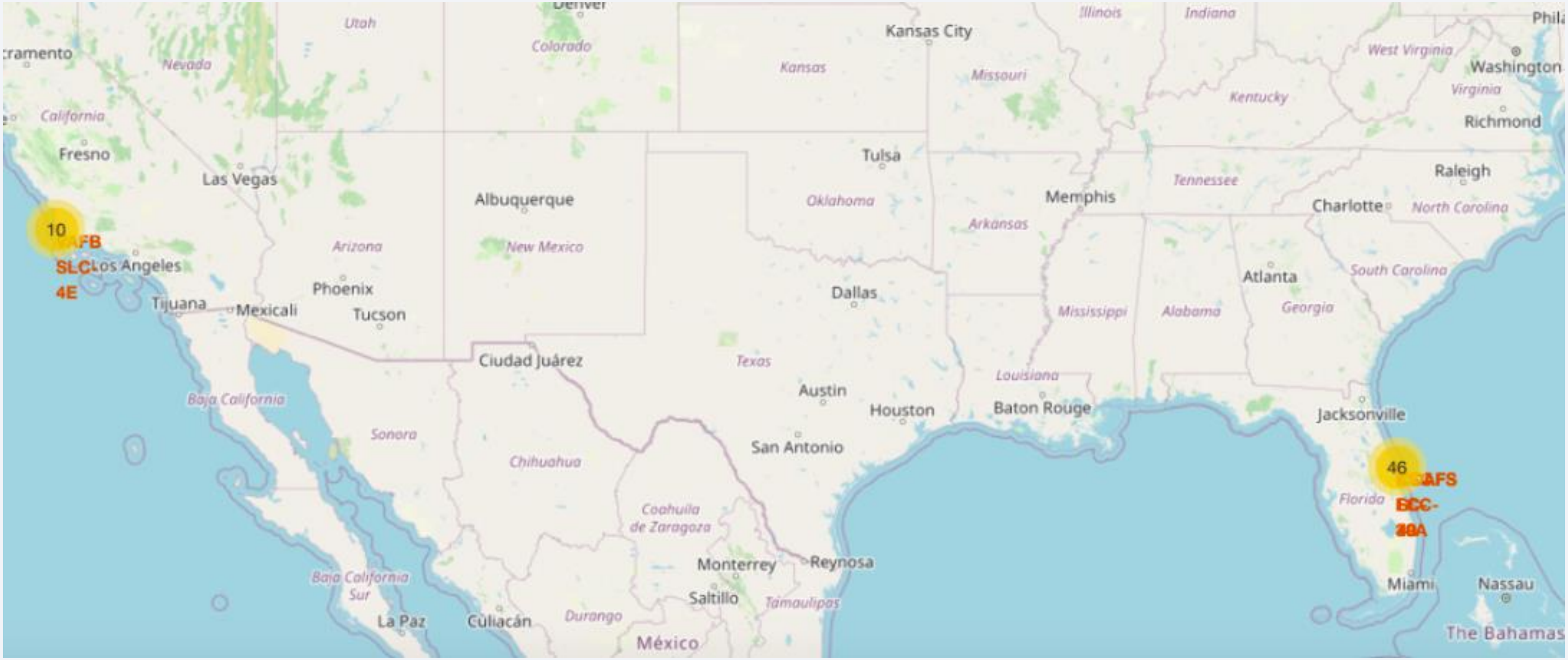
Section 4

Launch Sites Proximities Analysis

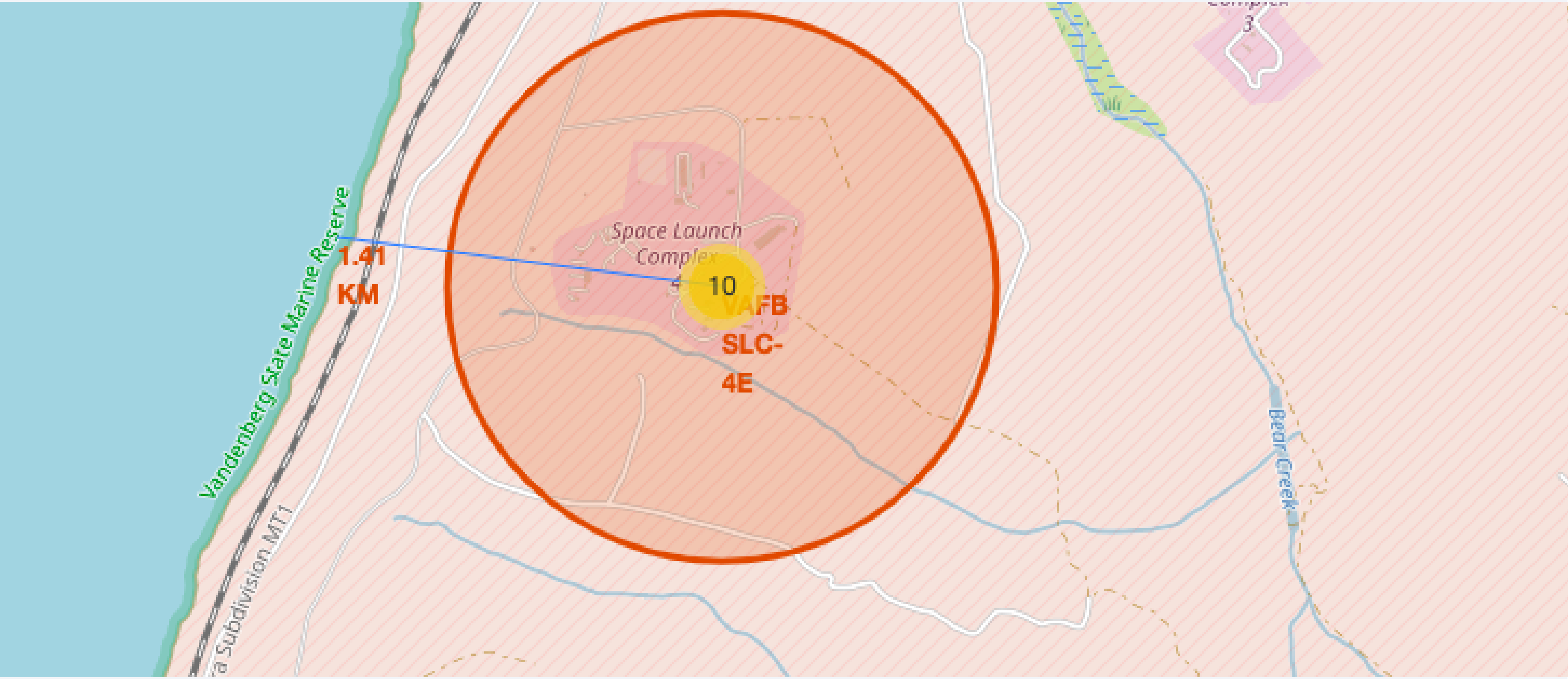
Launch Site Locations



Color Coded Markers



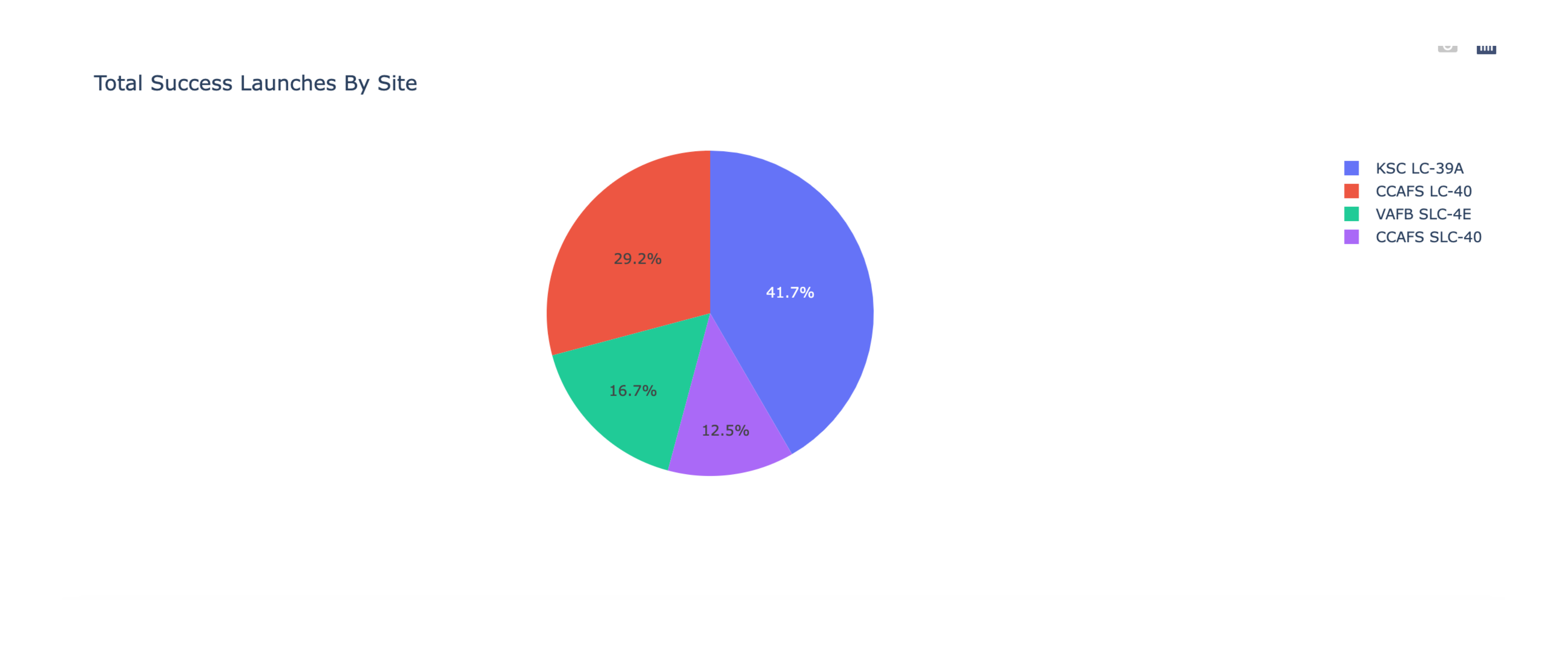
Locations Nearby



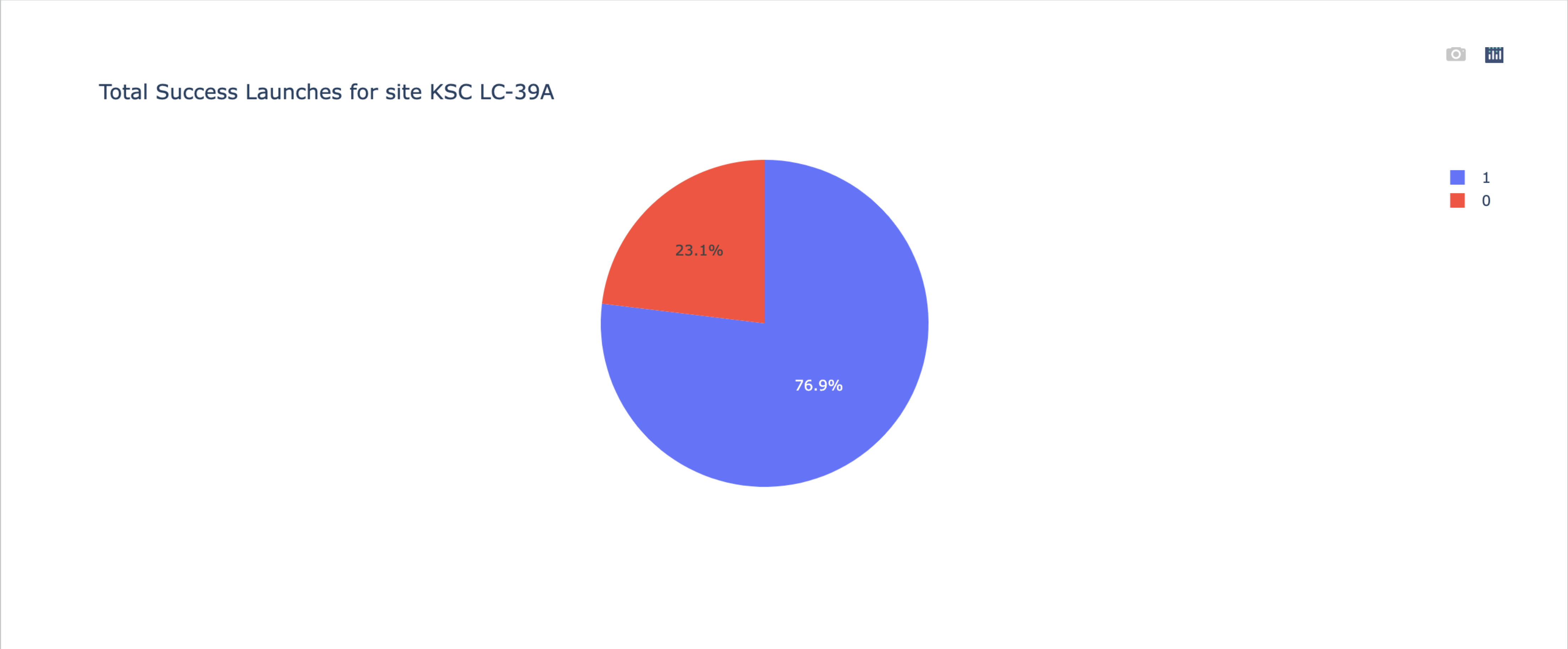
Section 5

Build a Dashboard with Plotly Dash

Launch Success All Sites



Highest Launch Success



Payload vs. Launch Outcome scatter plot for all sites

Section 6

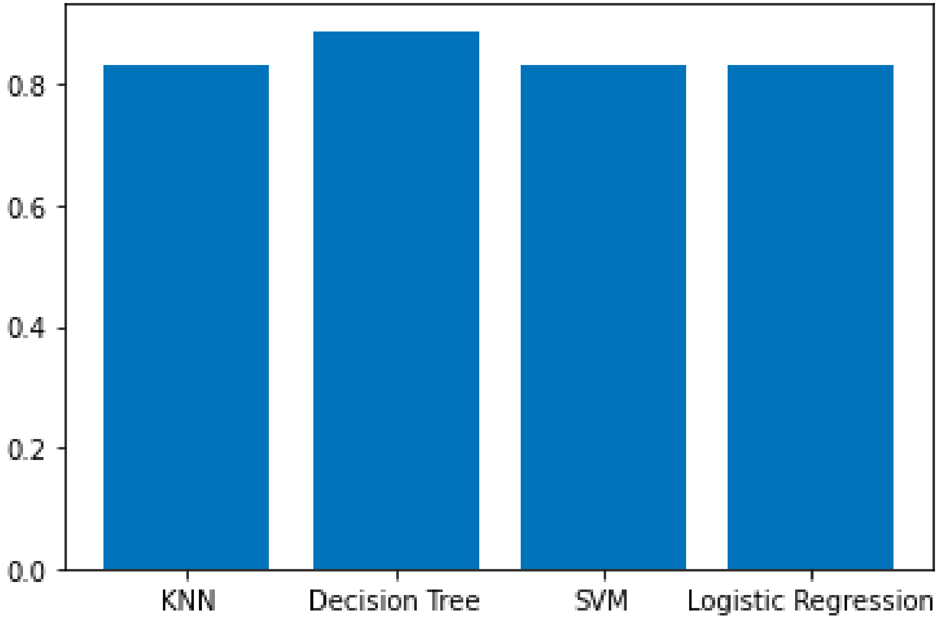
Predictive Analysis (Classification)

Classification Accuracy

All Relatively have same accuracy.

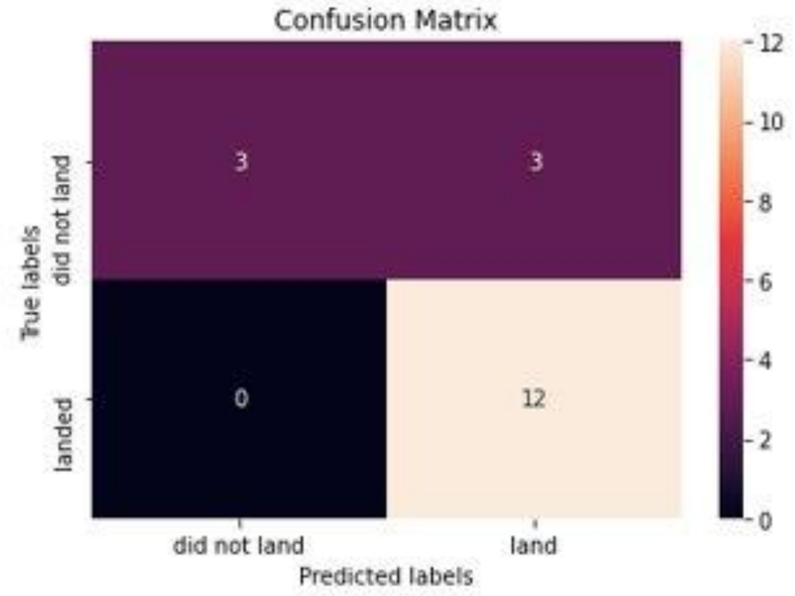
```
In [28]: # They all pretty much preformed the same on the test data
         x = ['KNN','Decision Tree','SVM','Logistic Regression']
         y = [knn_cv.score(X_test,Y_test),tree_cv.score(X_test,Y_test)
         plt.bar(x,y)

Out[28]: <BarContainer object of 4 artists>
```



Confusion Matrix

- This confusion matrix was the same for all different machine learning models we used and all had the same accuracy on the test set.



Conclusions

- Data came from API and Wikipedia
- The accuracy of our model is relatively high but our dataset is lacking in size
- We can use this model to help predict whether the retrieval of the first stage will be successful or not
- Further analysis can still be done

APPENDIX

Github Repository: https://github.com/MalamaPono/IBM-Data-Science-Specialization

Thanks to all instructors for this oppurtunity.

Thank you!