IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Hala Haneya
February 6, 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics 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

The commercial space age is here, companies are making space travel affordable for everyone. One of the most prominent companies nowadays is. SpaceX. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards 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. The goal of this project is to train a machine learning model and use public information to predict if SpaceX will reuse the first stage.

- Problems to solve
    - Determine the price of each launch
    - Determine if SpaceX will reuse the first stage.
    - Determine the factors that will land the rocket successfully.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected through the Space X API and web scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was performed on categorical features and cleaning the data.
- 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

- Data was collected using 2 methods:

SpaceX API

Web Scraping

Data Collection – SpaceX API

- We first use the GET request using the API, then normalize using json_normalize then performed some cleaning and filling in the missing values.
- The GitHub URL of the notebook: https://github.com/Hala-H/AppliedDataScienceCapstone/blob/master/Hands-on%20Lab:%20Complete%20the%20Data%20Collection%20API%20Lab.ipynb

1. Request and parse data using GET

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

```
response = requests.get(spacex_url)
```

```
# Use json_normalize method to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

2. Filter only Falcon 9 launches

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = data.loc[data['BoosterVersion'] != 'Falcon 1']
data_falcon9.head()
```

3. Dealing with missing values

```
# Calculate the mean value of PayloadMass column
PayloadMass_mean = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] =
data_falcon9['PayloadMass'].replace(np.nan, PayloadMass_mean)
```

Data Collection - Scraping

- We extract Falcon 9 records HTML table from Wikipedia and the parse it into a Pandas data frame.

- The GitHub URL of the notebook: https://github.com/Hala-H/AppliedDataScienceCapstone/blob/master/Data%20Collection%20with%20Web%20Scraping%20lab.ipynb

1. HTTP GET method to request the page

```
response = requests.get(static_url)
```

```
# Use BeautifulSoup() to create a BeautifulSoup object
soup = BeautifulSoup(response.content, 'html.parser')
```

2. Extract the column names

```
html_tables = soup.find_all('table')
```

```
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
for element in first_launch_table.find_all('th'):
    name = extract_column_from_header(element)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

3. Create a data frame
    1. Create empty dictionary from column names
    2. Fill up the dictionary with table data
    3. Create a data frame from the filled dictionary

```
df=pd.DataFrame(launch_dict)
```

Data Wrangling

- In this part we perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.

- The flowchart on the right demonstrates the process.

- The GitHub URL of the notebook: https://github.com/Hala-H/AppliedDataScienceCapstone/blob/master/Complete%20the%20EDA%20lab.ipynb

Calculate the number of launches on each site

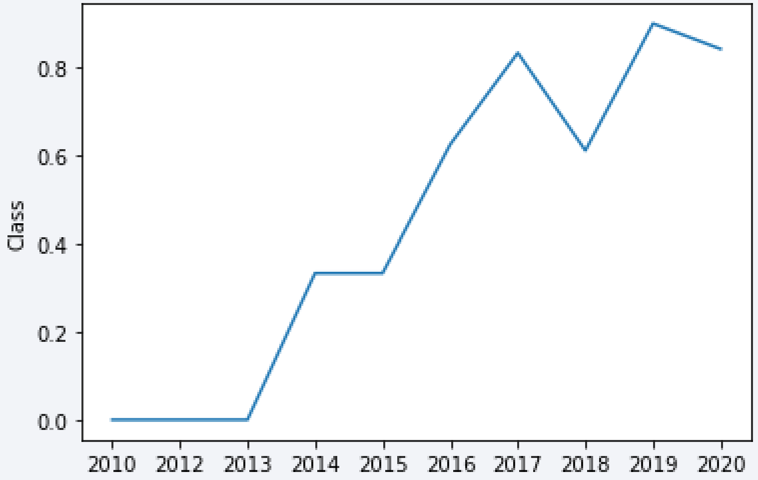
Calculate the number and occurrence of each orbit

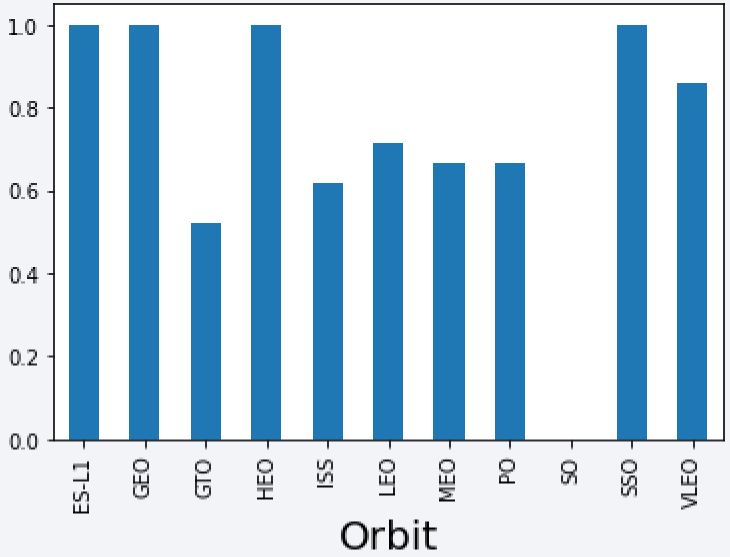
Calculate the number and occurrence of mission outcome per orbit type

Create a landing outcome label from Outcome column

EDA with Data Visualization

- In this part, we visualize relationships between flight number and launch sites, payload and launch site, success rate of each orbit, flight number and orbit type, payload and orbit type and yearly trends of success rate.





- The GitHub URL of the notebook: https://github.com/Hala-H/AppliedDataScienceCapstone/blob/master/Complete%20the%20EDA%20with%20Visualization%20lab.ipynb

EDA with SQL

- We loaded the dataset using DB2 IBM Watson tool. Then, we performed EDA with SQL to gain some insight from the data.
- We wrote some SQL queries to find out:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.
- The GitHub URL of the notebook: https://github.com/Hala-H/AppliedDataScienceCapstone/blob/master/Complete%20the%20EDA%20with%20SQL%20lab.ipynb

Build an Interactive Map with Folium

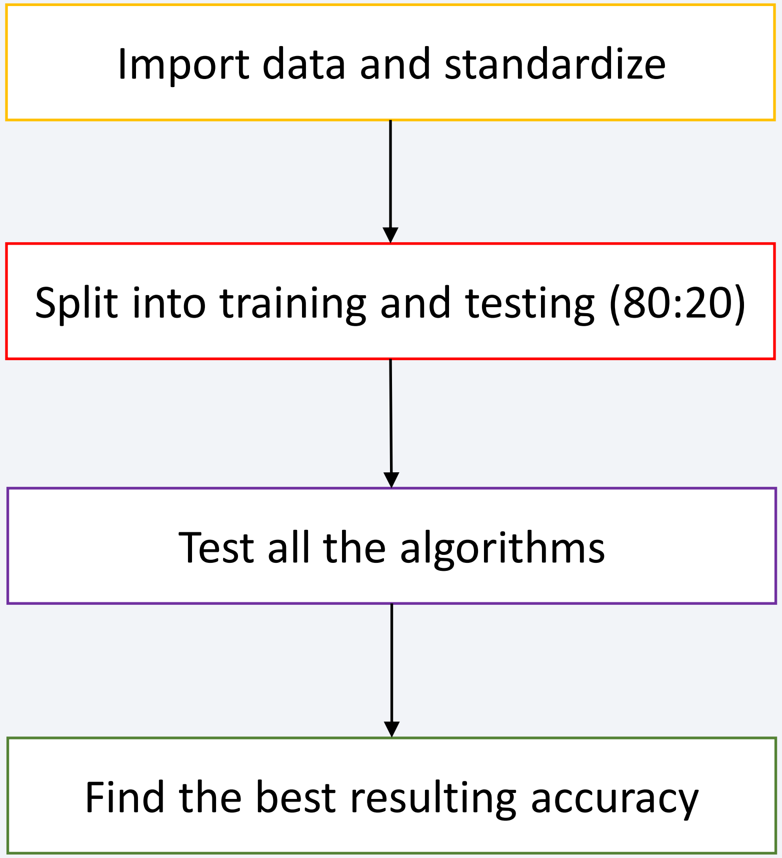
- The goal for this part was to:
  - TASK 1: Mark all launch sites on a map
  - TASK 2: Mark the success/failed launches for each site on the map
  - TASK 3: Calculate the distances between a launch site to its proximities
- We added circles with popups to highlight certain areas, markers to mark the launch sites, color-labeled markers to identify success rates and lines to find places with close proximity to the site.
- We answered some questions:
  - Are all launch sites in proximity to the coast, railway, highways,?
  - Are all launch sites in very close proximity to the coast?
- The GitHub URL of the notebook: https://github.com/Hala-H/AppliedDataScienceCapstone/blob/master/Complete%20the%20Interactive%20Visual%20Analytics%20(Folium).ipynb

Build a Dashboard with Plotly Dash

- In this part, we built an interactive dashboard with Plotly Dash.
- We have a dropdown to choose the launch site.
- Then, a pie chart will be built based on the option which shows the success portion.
- Next, a payload range will be chosen and be plotted in a scatter plot against the success rate of the chosen site from the dropdown.
- The GitHub URL of the code: https://github.com/Hala-H/AppliedDataScienceCapstone/blob/master/spacex_dash_app.py

Predictive Analysis (Classification)

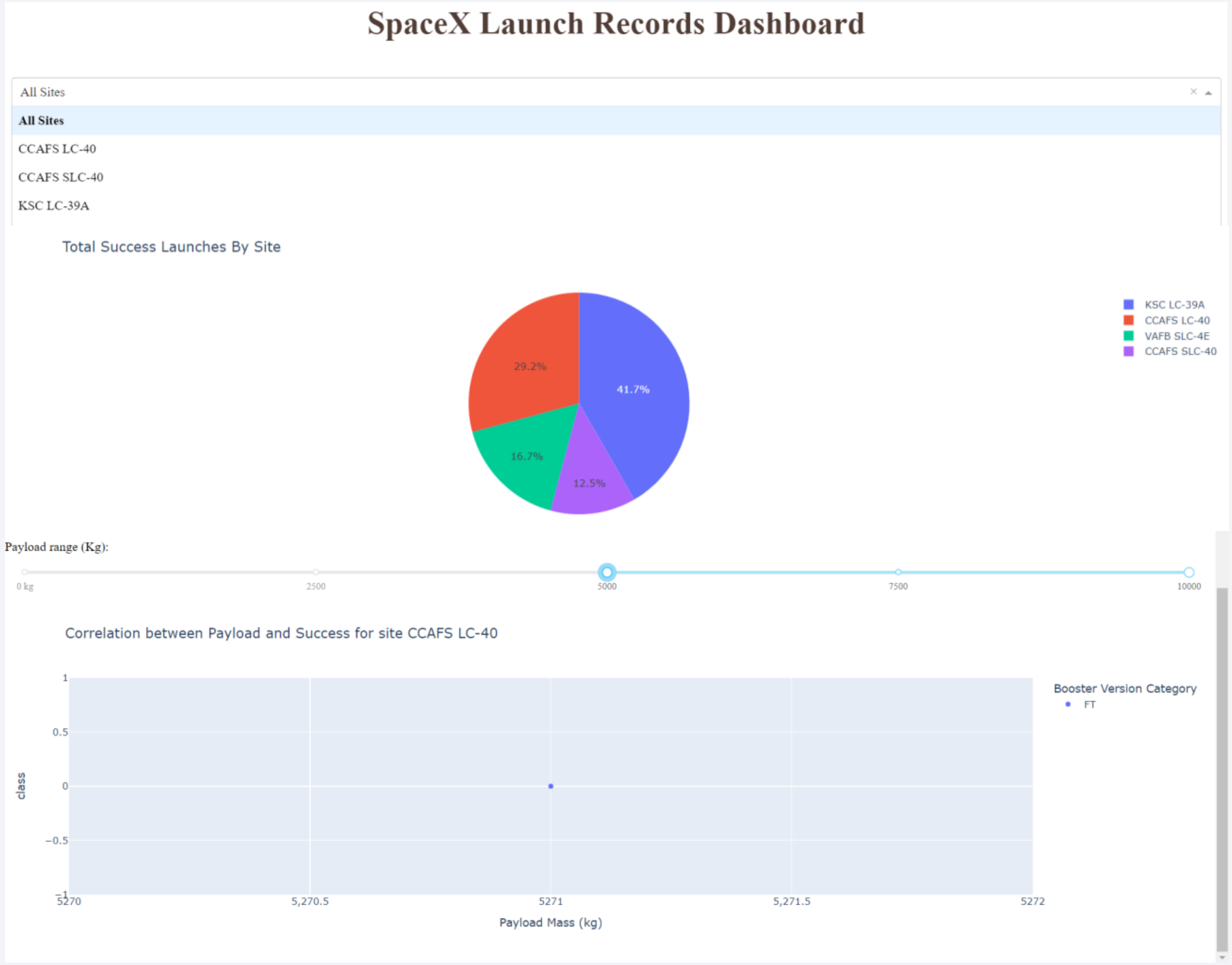
- The goal was to find the best hyperparameters for SVM, DT, LR. The steps followed were:
  - We loaded the data using NumPy and pandas, transformed the data, split our data into training and testing.
  - We built different machine learning models and tune different hyperparameters using GridSearchCV.
  - We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
  - We found out that all the algorithms have the same accuracy of 83.33%
- The GitHub of the notebook: https://github.com/Hala-H/AppliedDataScienceCapstone/blob/master/Complete%20the%20Machine%20Learning%20Prediction%20lab.ipynb



Results

- Exploratory data analysis results:
  - The EDA showed us which features to choose based on their relationships
- Predictive analysis results:
  - All algorithms achieved an accuracy of 83.33%

Interactive analytics demo in screenshots:

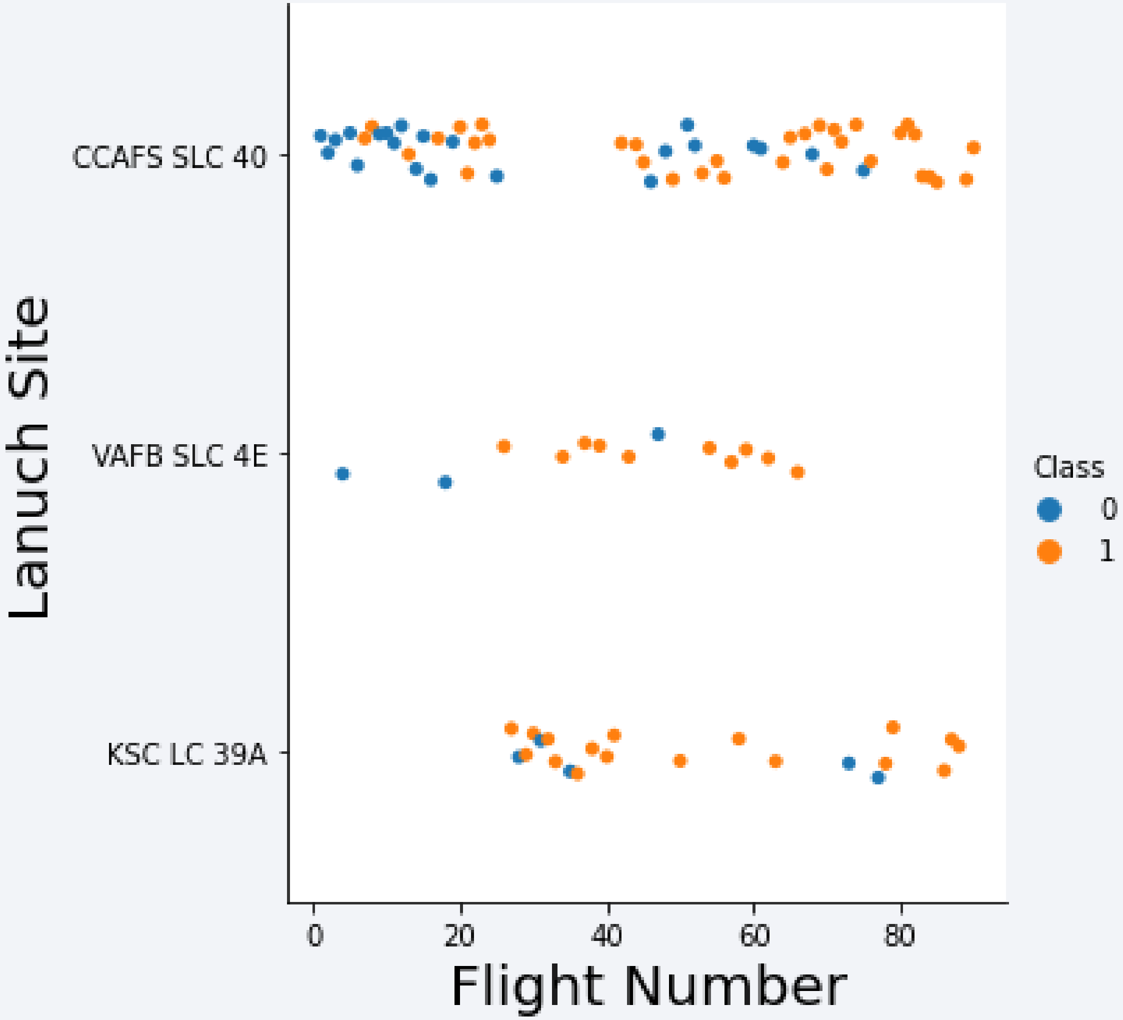


Section 2

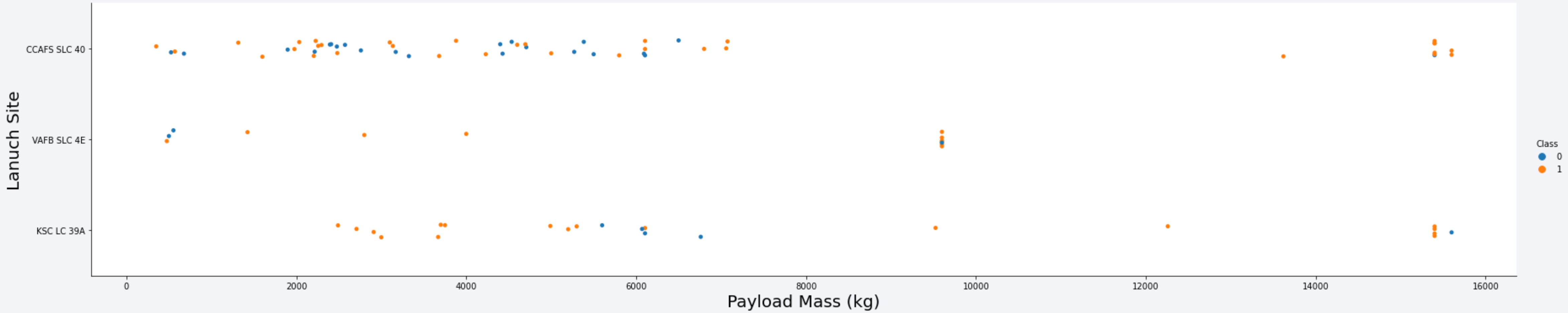
Insights drawn from EDA

Flight Number vs. Launch Site

As shown in the plot, the larger the flight amount at a launch site, the greater the success rate at the launch site.



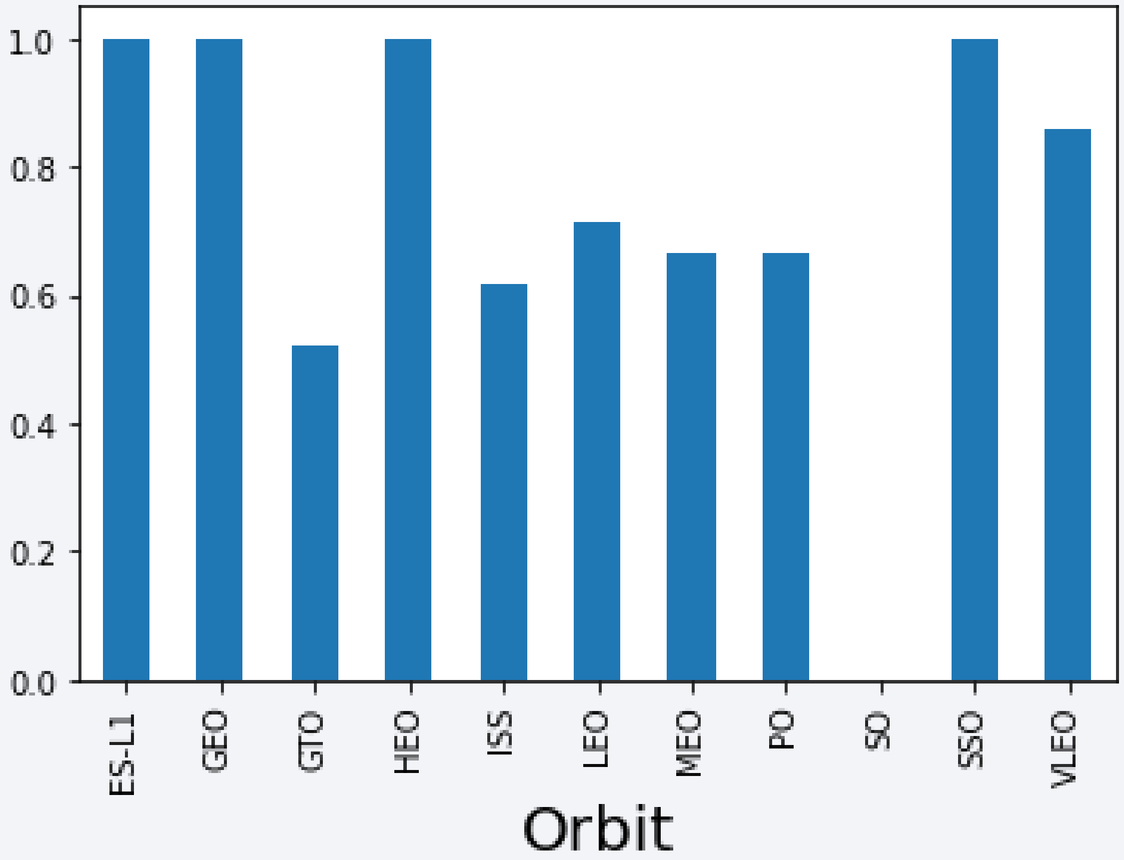
Payload vs. Launch Site



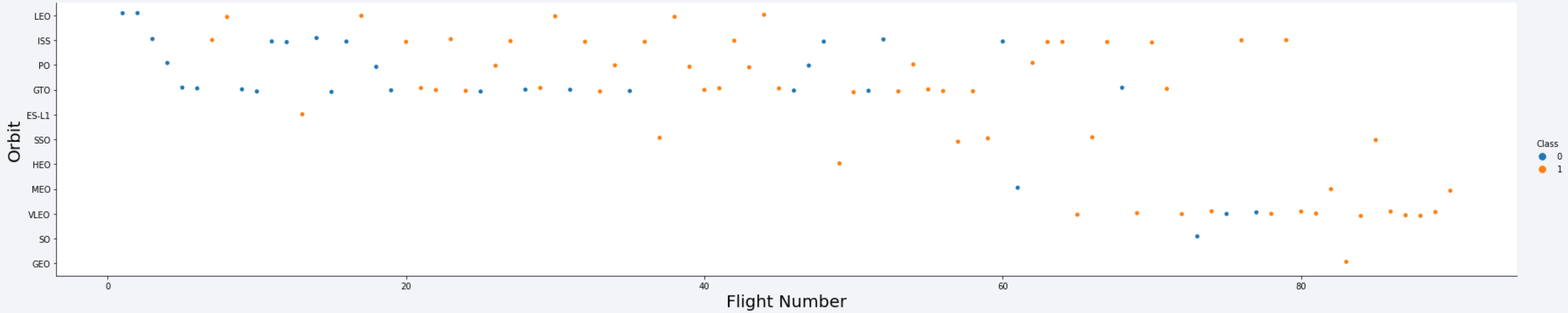
As shown below, lower payload mass has more success rate especially for CCAFS SLC 40. Also, for the VAFB-SLC launch site there are no rockets launched for heavy payload mass (greater than 10000).

Success Rate vs. Orbit Type

As the plot shows, ESOL1, GEO, HEO, and SSO has the highest success rates.

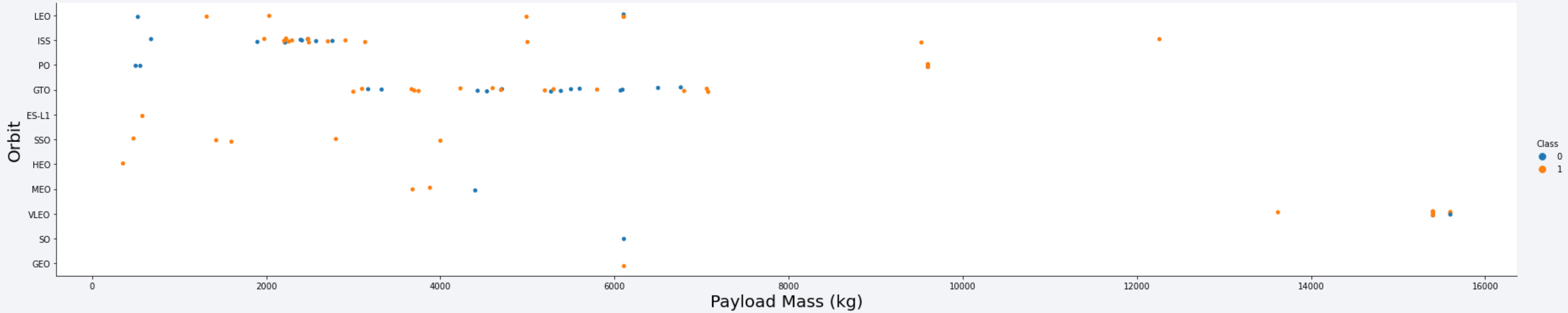


Flight Number vs. Orbit Type



As shown below, for 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.

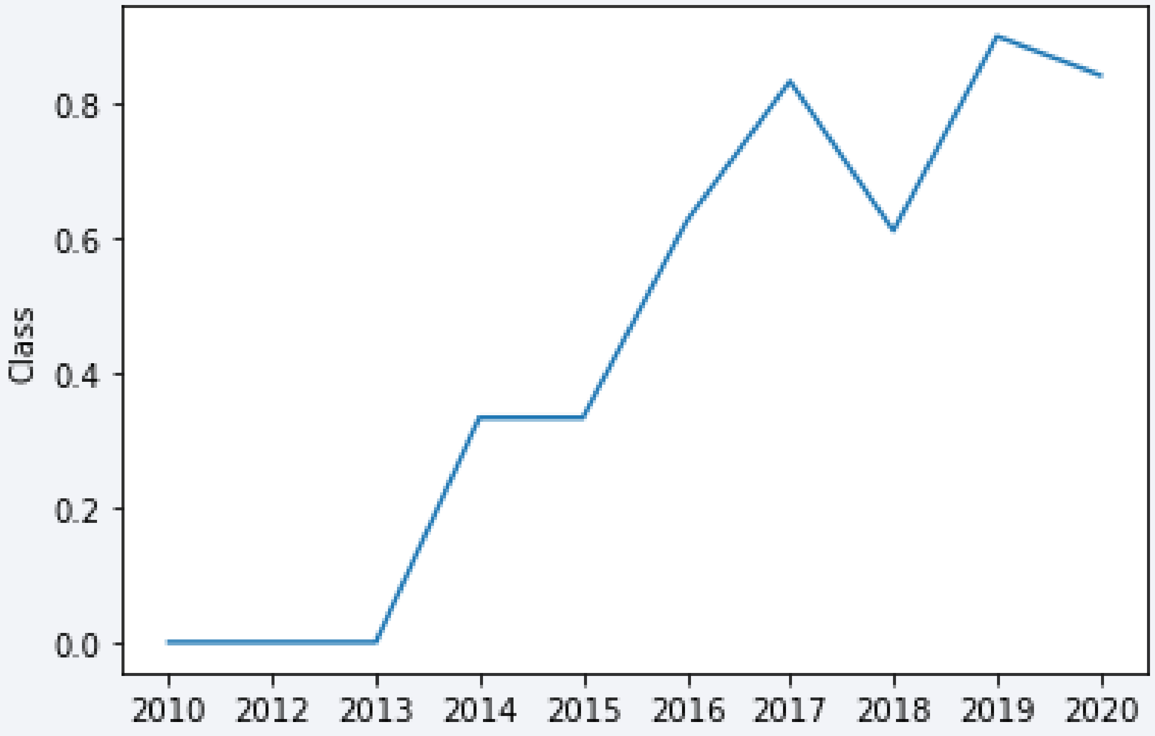
Payload vs. Orbit Type



As shown below, the 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

As shown on the plot, the success rate started increasing in 2013 and had a steady rise until 2017 where it dropped to 0.6. Then, it rose up again until mid 2019 where it started to decrease a little.



All Launch Site Names

- We used the key word **DISTINCT** to get the unique launch sites.
- The query used is:

```
%%sql
SELECT DISTINCT(LAUNCH_SITE)
from SPACEXTBL
```

- The launch sites:

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

Launch Site Names Begin with 'CCA'

- We use the query using the condition in WHERE to show 5 records with launch sites starting with CCA.

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

```
 * ibm_db_sa://gxm00617:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/BLUDB
Done.
```

]:

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

- The total payload carried by boosters launched by NASA (CRS) is **45596 kg**.

```
%%sql
SELECT sum(payload_mass__kg_)
FROM SPACEXTBL
WHERE customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://gxm00617:***@76
Done.
```

]:

1
45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1 is **2928 kg**.

```
%%sql
SELECT avg(payload_mass__kg_)
FROM SPACEXTBL
WHERE booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://gxm00617:***@76426
Done.
```

]:

1
2928

First Successful Ground Landing Date

- The date of the first successful landing outcome on ground pad is

December 22, 2015.

```
%%sql
SELECT min(date)
FROM SPACEXTBL
WHERE landing__outcome = 'Success (groun
```

```
 * ibm_db_sa://gxm00617:***@764264db-
Done.
```

]:

1
2015-12-22

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 are shown below.
- We used WHERE to specify the condition of cusses and the keyword BETWEEN and AND to specify the range.

```
%%sql
SELECT payload
FROM SPACEXTBL
WHERE landing__outcome = 'Success (drone ship)' and
payload_mass__kg_ between 4000 and 6000
```

```
 * ibm_db_sa://gxm00617:***@764264db-9824-4b7c-82c
Done.
```

]:

payload
JCSAT-14
JCSAT-16
SES-10
SES-11 / EchoStar 105

Total Number of Successful and Failure Mission Outcomes

- The total number of successful mission outcomes is 100 and the failures is 1 and the distribution is shown below.
- We use the function **COUNT** and then **GROUP BY**.

```
%%sql
SELECT mission_outcome, count(*) total
FROM SPACEXTBL
GROUP BY mission_outcome
```

```
 * ibm_db_sa://gxm00617:***@764264db-9824-4
Done.
```

7]:

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

Boosters Carried Maximum Payload

- The names of the booster which have carried the maximum payload mass.
- We used a nested query to select the maximum payload first and then find the all boosters with that payload.

```
%%sql
SELECT booster_version
FROM SPACEXTBL
WHERE payload_mass__kg_ = (SELECT max(payload_mass__kg_) FROM SPACEXTBL)
```

```
 * ibm_db_sa://gxm00617:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l
Done.
```

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

- The failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015 are shown below.
- We used WHERE to specify the landing outcome and we extracted the year from the date using the YEAR() function.

```
%%sql
SELECT landing__outcome, booster_version, launch_site
FROM SPACEXTBL
WHERE landing__outcome = 'Failure (drone ship)' and year(date) = 2015
```

```
 * ibm_db_sa://gxm00617:***@764264db-9824-4b7c-82df-40d1b13897c2.bs
Done.
```

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

- 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.
- We first filtered the dates by WHERE.
- Then, we GROUP BY the landing outcome.
- Finally, we ORDER BY the count in a descending order.

```
%%sql
SELECT landing__outcome, count(*) count
FROM SPACEXTBL
WHERE date between '2010-06-04' and '2017-03-20'
GROUP BY landing__outcome
ORDER BY count desc
```

```
 * ibm_db_sa://gxm00617:***@764264db-9824-4b7c-
Done.
```

]:

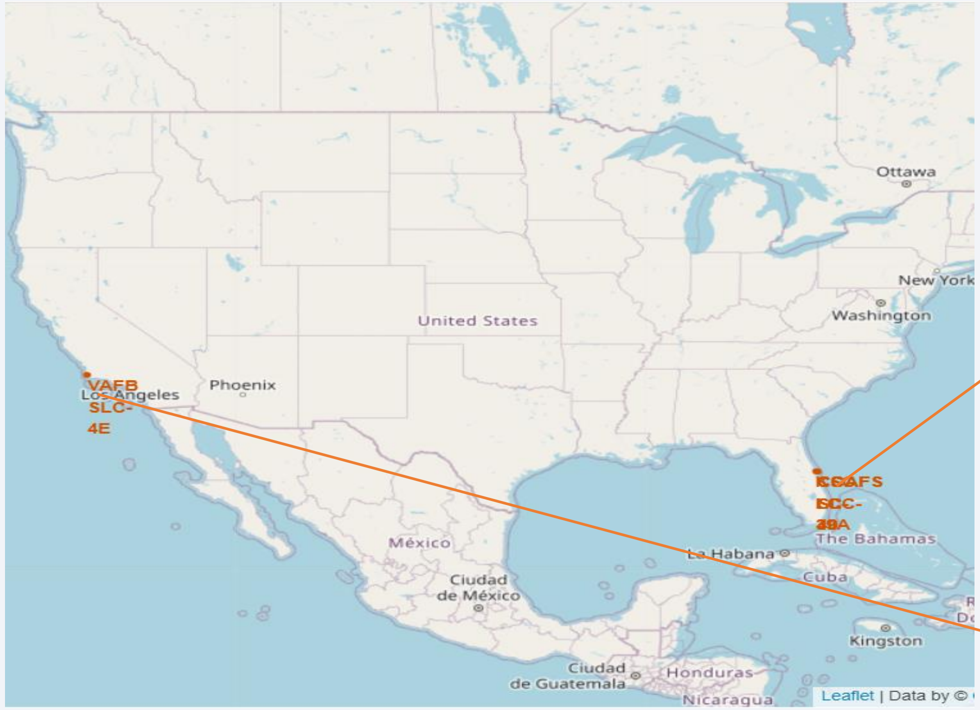
landing__outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

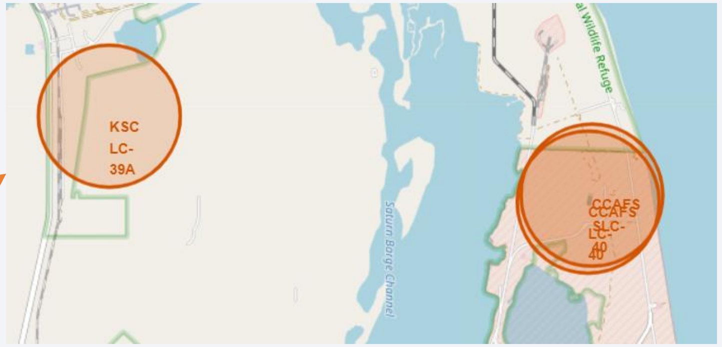
Launch Sites Proximities Analysis

All launch sites

All launch sites on the map.

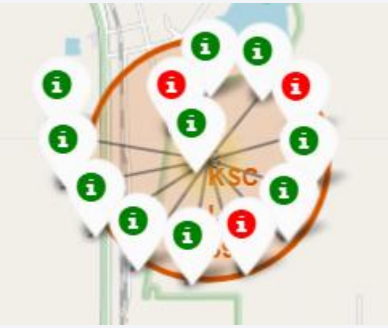


A close up of the three launch sites in Florida.



A close up of the launch site in Santa Maria

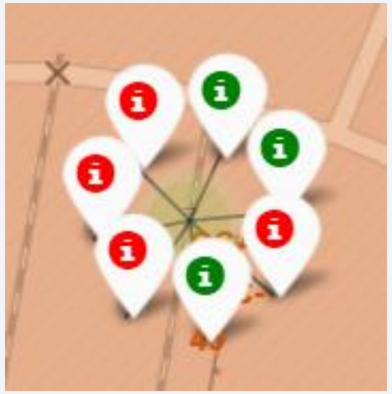
Launch sites with success and failure markers



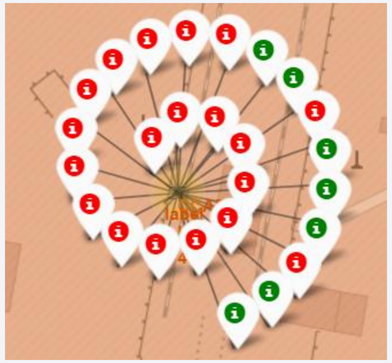
KSC LC-39A

VAFB SLC 4E

As shown, we have highlighted the launch sites.



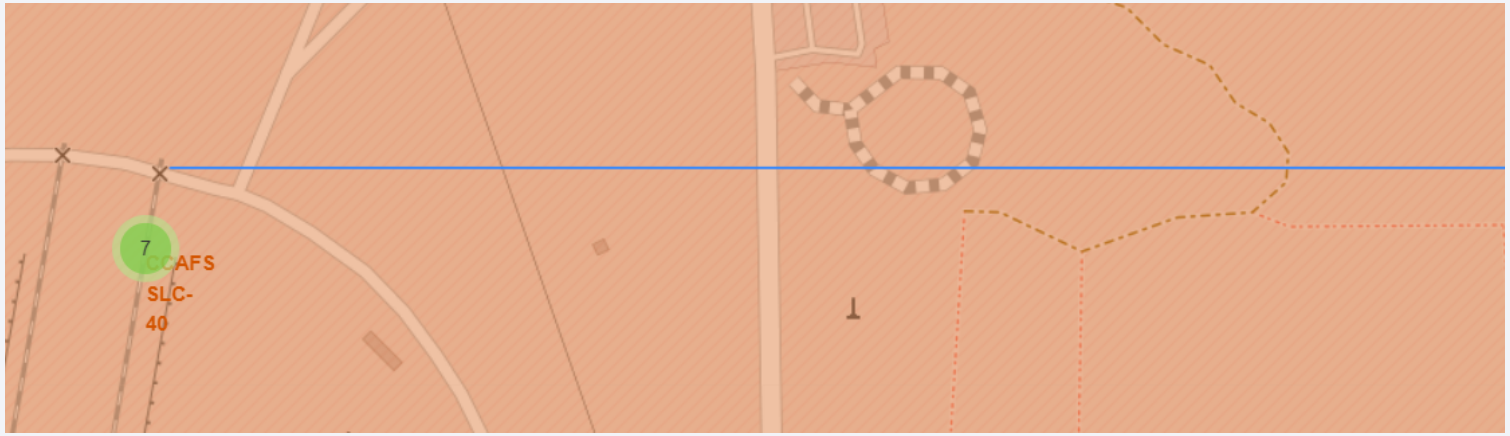
CCAFS SLC-40



CCAFS LC-40

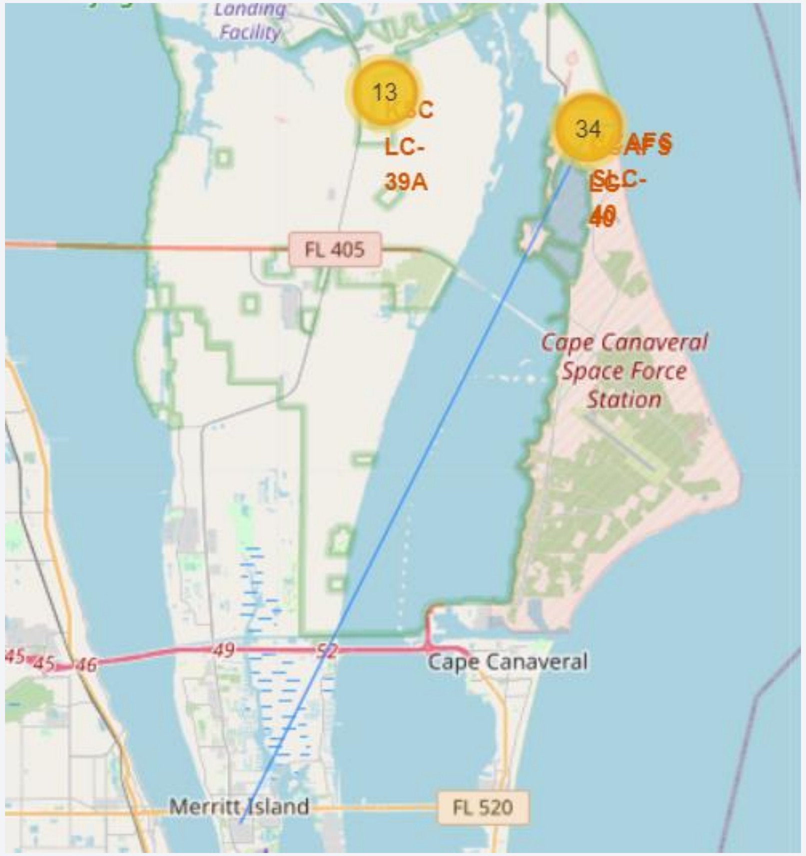
Launch sites proximity to landmarks

Distance to coast



- 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

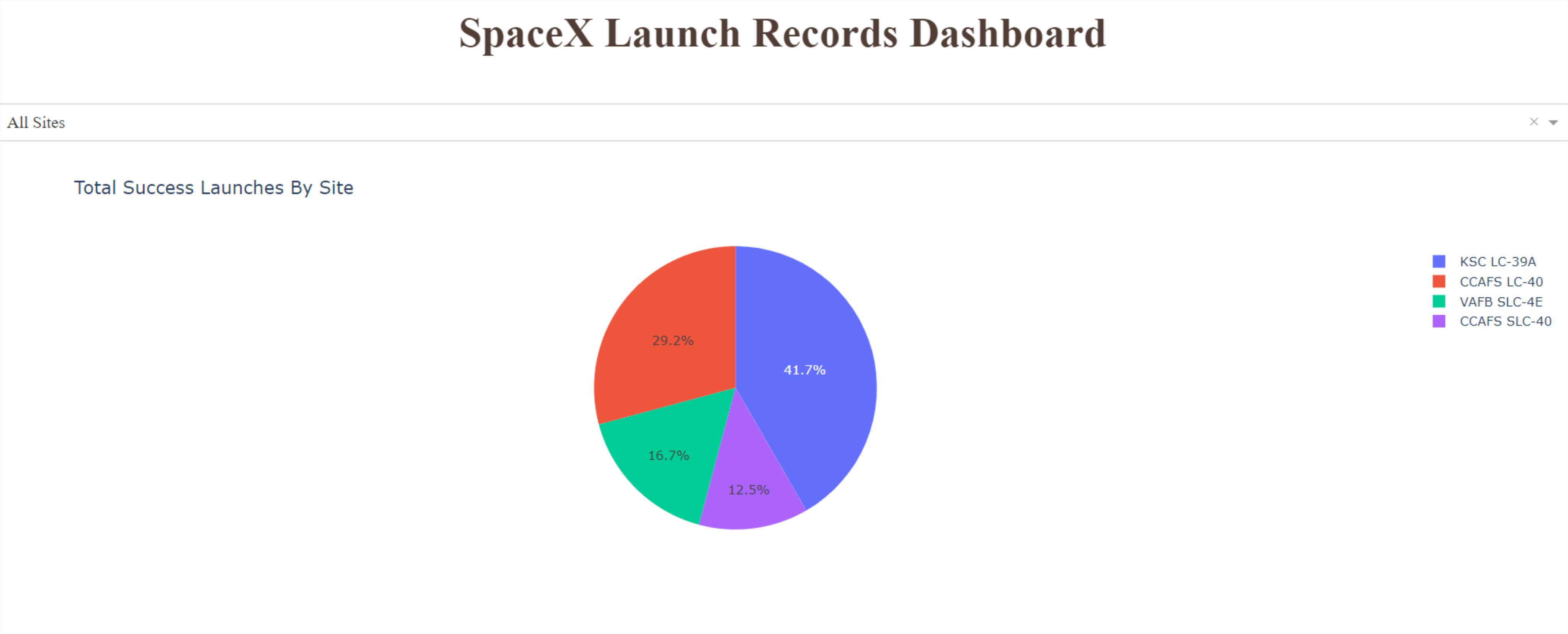
Distance to Merrit Island



Section 4

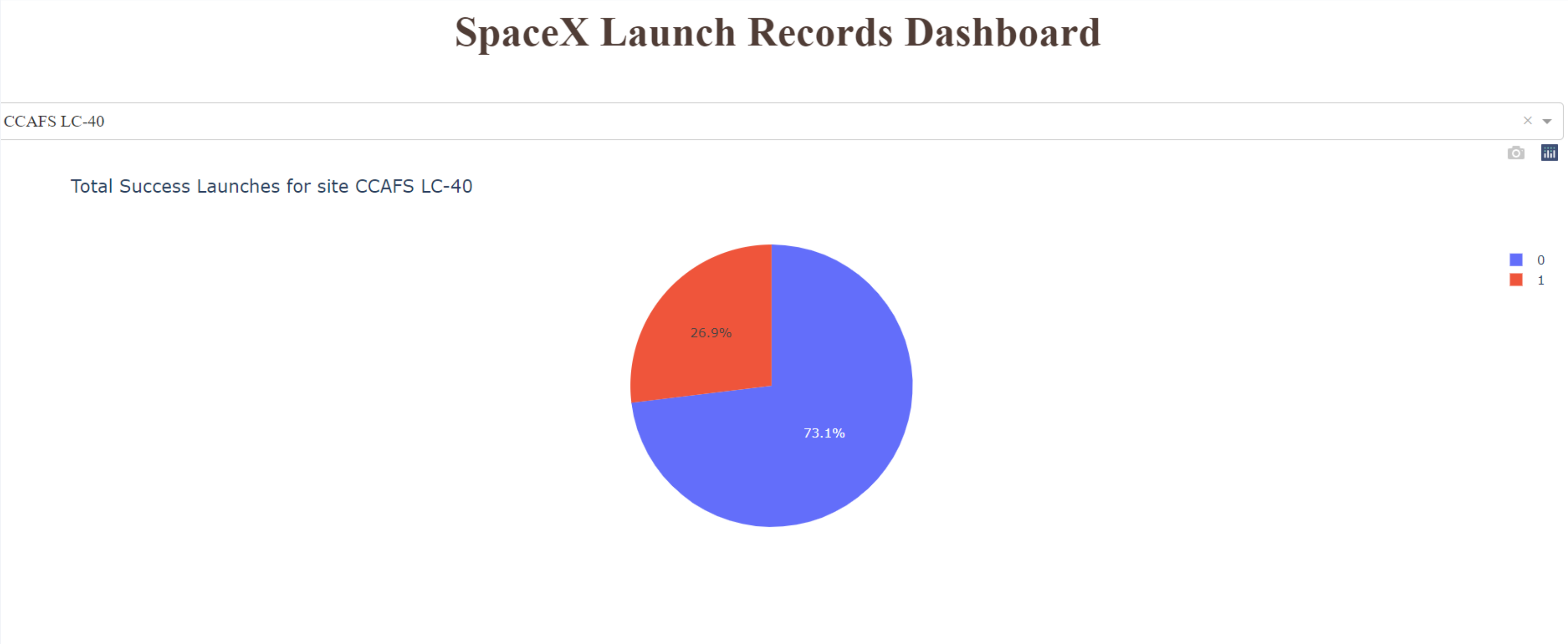
Build a Dashboard with Plotly Dash

Pie of success percentage achieved by each launch site



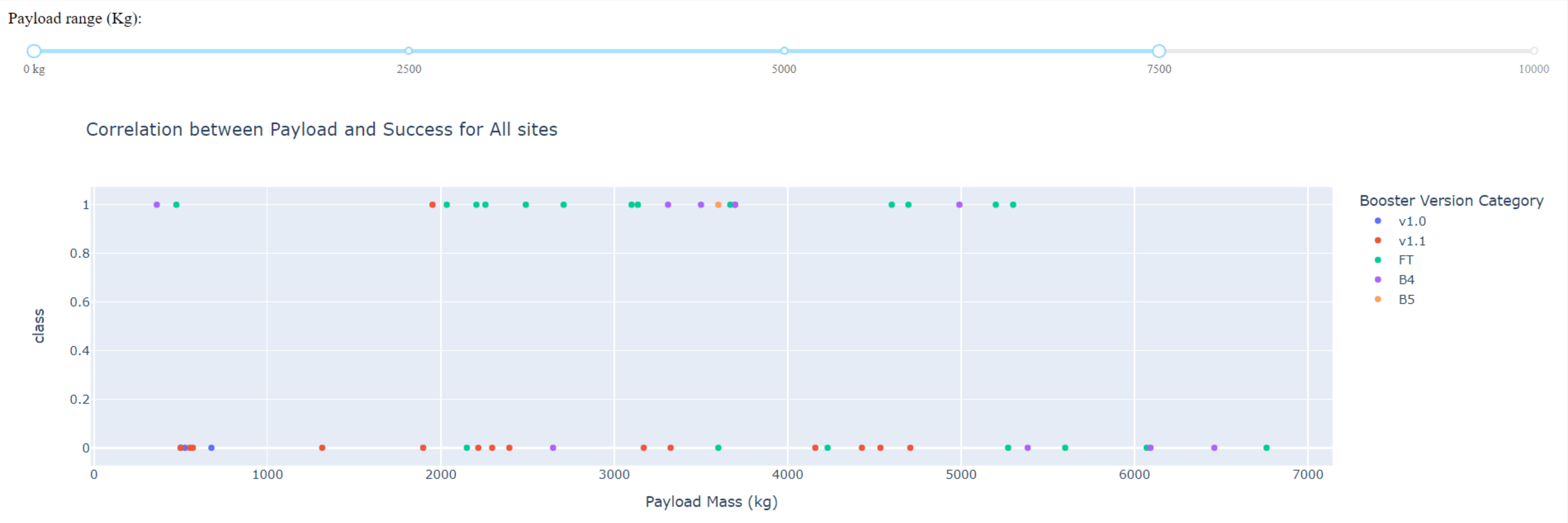
As shown above, KSC KC-39A has the most successful launched at 41.7%.

Pie Chart of CCAFS LC-40



As shown, **CCAFS LC-40** has **73.1% success rate**.

Correlation between Payload and various Launch site



As shown above, the correlation between the payload (0 – 7500) and all launch sites. These values can be changed according to the requirements.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The code used to find the best classifier:

```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

- We found out that the **Decision Tree Classifier has the highest classification accuracy at 88.9 %**.

```
Best Algorithm is Tree with a score of 0.8892857142857145
```

Confusion Matrix

The confusion matrix for the decision tree classifier shows that the classifier **can distinguish between the different classes**.

The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.

Conclusions

- The larger the flight count at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, and SSO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision Tree classifier is the best machine learning algorithm for this task.

Appendix

- This is the link for the repository: https://github.com/Hala-H/AppliedDataScienceCapstone/tree/master

Thank you!