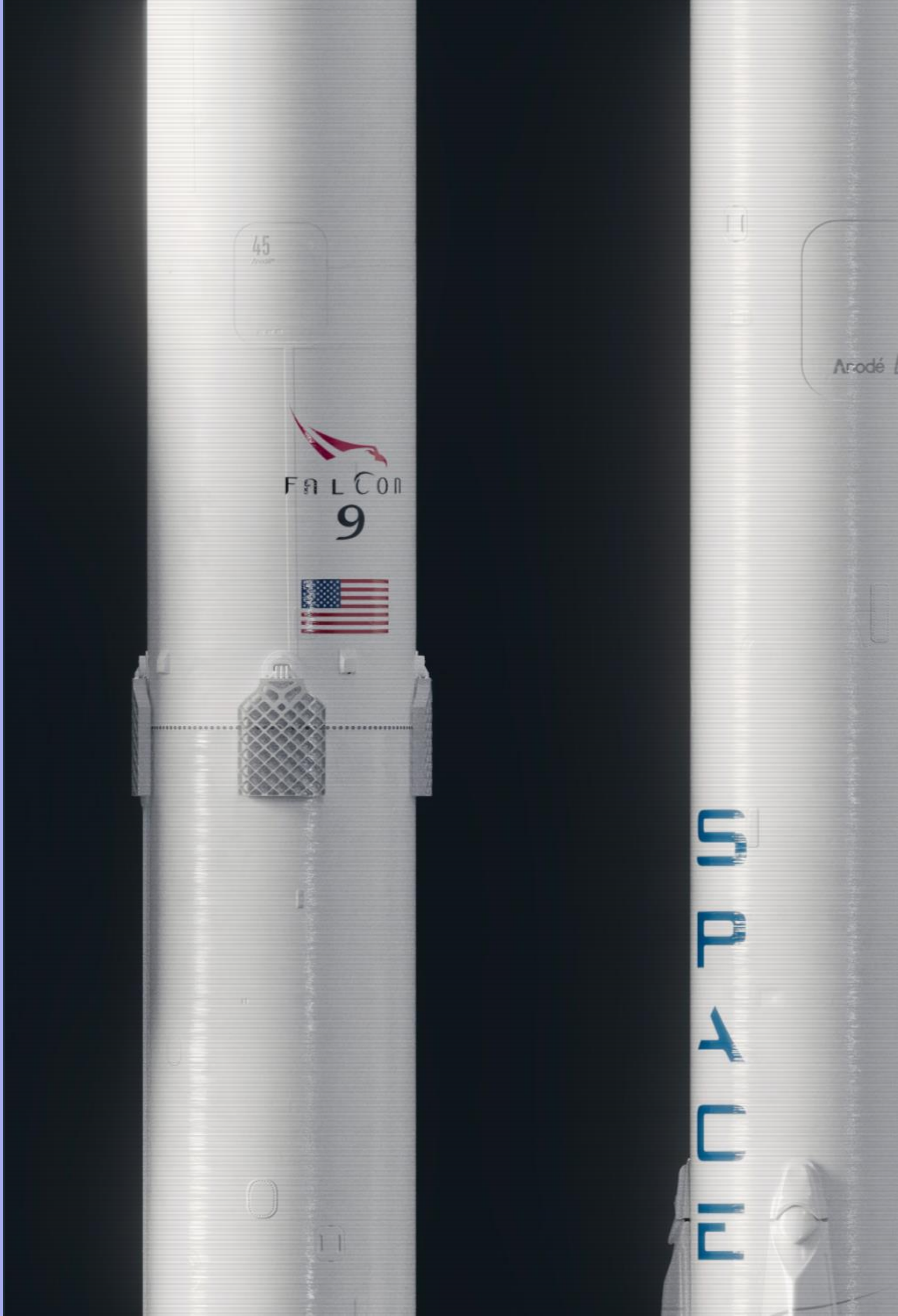
Marcos Soares

SpaceX Data Science Project

Presentation Contents



Executive Summary

Summary of methodologies

This research attempts to identify the factors for a successful rocket landing. To make this determination, the following methodologies were used:

- Data collection through API and web scraping
- Data wrangling
- Exploratory data analysis with data visualization
- Exploratory data analysis with data SQL
- Interactive visual analytics with Folium
- Machine Learning Prediction

Summary of all results

- Exploratory data analytics with interactive visualization
- Best models given through predictive analytics

Introduction

Project background

SpaceX, a leader in the space industry, strives to make space travel affordable for everyone. One of its main factors in doing so is the advertisement of their Falcon 9 rocket, which is a reusable, two staged rocket designed to re-fly the most expensive parts of the rocket to drive down the cost of space access ($62 million per launch). Other providers, which do not reuse the first stage, can cost upwards of $165 million each. This project will attempt to determine the cost of a launch with the outcome if the first stage can land safely. To do this, the project will use public data and machine learning models to predict if competitors can reuse the first stage in attempt to bid against SpaceX for a rocket launch.

Problems that need to be answered

- What factors determine if the rocket will land successfully
- The rate of successful landings over time
- What operating conditions need to be in place to ensure a successful landing program

Methodology

Data Collection - API

Steps

- Request data from SpaceX API
- Convert response to .json file
- Replace missing values and cleaned data
- Use data to create dictionary, in turn to create a dataframe
- Filter dataframe to only include Falcon 9 launches
- Export data to .csv file

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNe
```

We should see that the request was successfull with the 200 status response code

```
response.status_code
```

```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

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

FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite
1	2010-06-04	Falcon 9	6104.959411764706	LEO	CCAFS SLC 40
2	2012-05-22	Falcon 9	525.0	LEO	CCAFS SLC 40
3	2013-03-01	Falcon 9	677.0	ISS	CCAFS SLC 40
4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E
5	2013-12-03	Falcon 9	3170.0	GTO	CCAFS SLC 40

Data Collection – Web Scraping

Steps

- Request Falcon 9 launch data from SpaceX Wikipedia
- Create BeautifulSoup object from HTML response
- Find table and extract column names
- Use data to create dictionary, in turn to create dataframe
- Export data to .csv file

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
# use requests.get() method with the provided static_url
html_data = requests.get(static_url)

# assign the response to a object
html_data.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(html_data.text, 'html.parser')
```

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

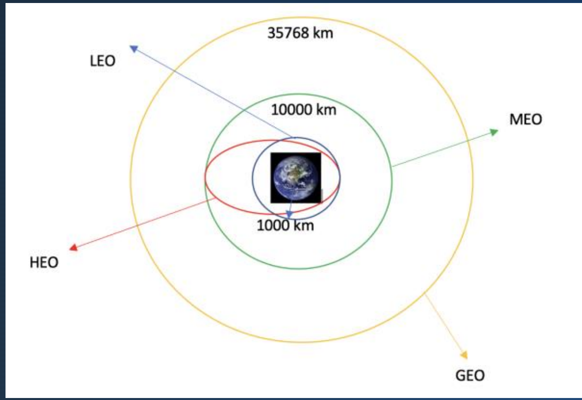
```
# Use soup.title attribute
soup.title
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

Data Wrangling

Steps

- Perform exploratory data analysis to determine data labels
- Calculate the following:
  - Number of launches per site
  - Occurrence of orbits
  - Occurrence of missions and outcome per orbit type
- Create binary landing outcome column
  - Outcomes converted into 1 for a successful landing and 0 for unsuccessful landing



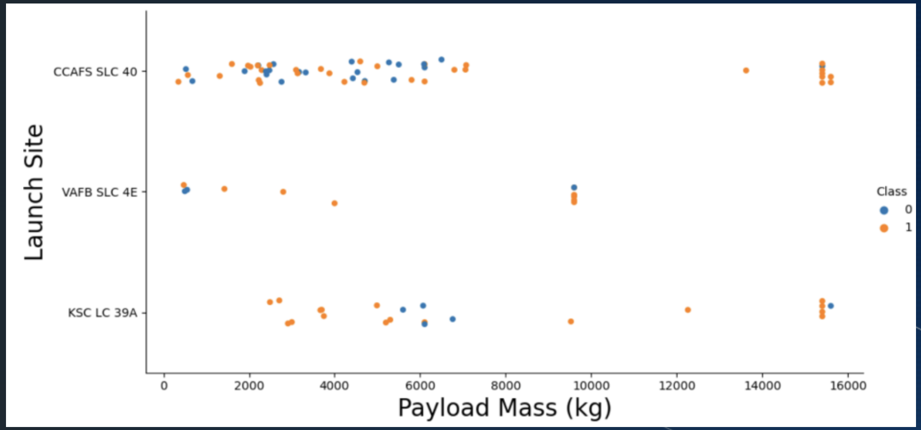
Use the method `.value_counts()` on the column `Outcome` to determine the number of `landing_outcomes` .Then assign it to a variable landing_outcomes.

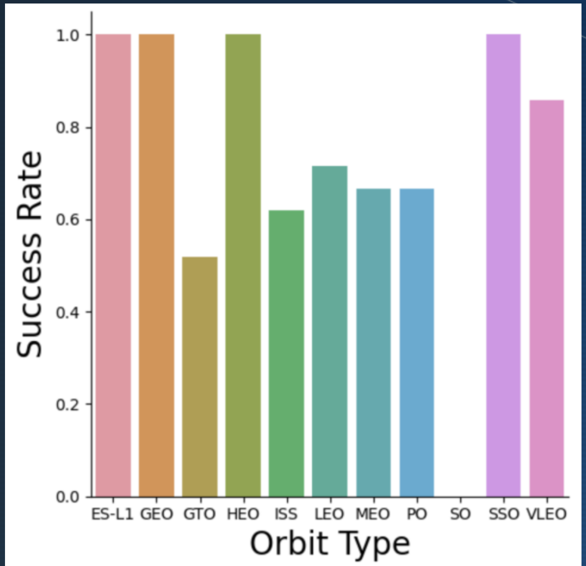
```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

`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 to a drone ship `False ASDS` means the mission outcome was unsuccessfully landed to a drone ship. `None ASDS` and `None None` these represent a failure to land.

EDA with Visualization





Data will be explored visualizing the relationship between:

- Payload Mass vs. Orbit Type
- Payload Mass vs. Launch Site
- Flight Number vs. Launch Site
- Flight Number vs. Payload

Comparisions will be made between Success Rate vs. Orbit type while trends will be analyzed through the Launch Success Yearly Trend

EDA with SQL

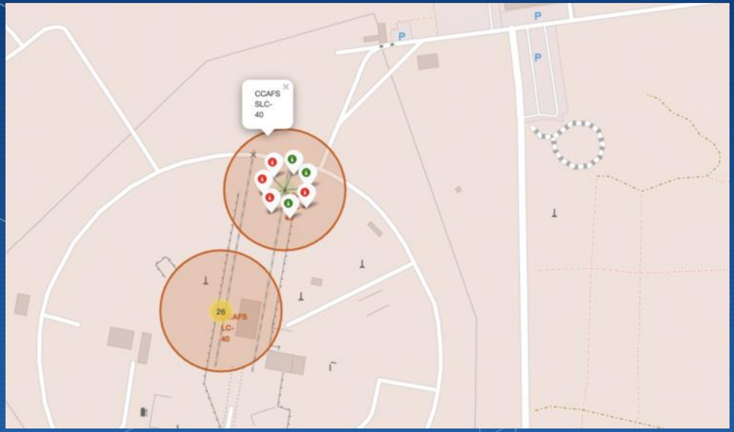
Dataset was loaded and queried from the IBM Db2 for Cloud. The following queries were written to gather information:

- **Display**
  - Names of unique launch sites
  - 5 records where launch site begins with 'CCA'
  - Total payload mass carried by boosters launched by NASA (CRS)
  - Average payload mass carried by booster version F9 v 1.1
- **List**
  - Date of first successful landing on grounded pad
  - Names of boosters which had successful landings on drone ship and have payload mass greater than 4,000 but less than 6,000
  - Total number of successful and failed missions
  - Name of booster versions that carried the max payload
  - Failed landing outcomes on drone ship, their booster version and launch site for the months in the year 2015
  - Count of landing outcomes between 06/04/2010 and 03/20/2017 in descending order

Map with Folium

Folium is used to visualize data that has been manipulated with python and displayed on an interactive map

- Red circles were added to indicated all launch site coordinates
- Colored markers were added to indicate a successful (green) and unsuccessful (red) launch at each launch site, to show which launch sites have the highest success rate
- Colored lines were added to show distance between the launch site CCAFS SLC-40 and its proximity to the nearest city, highway, railway and coastline



Dashboard with Plotly Dash

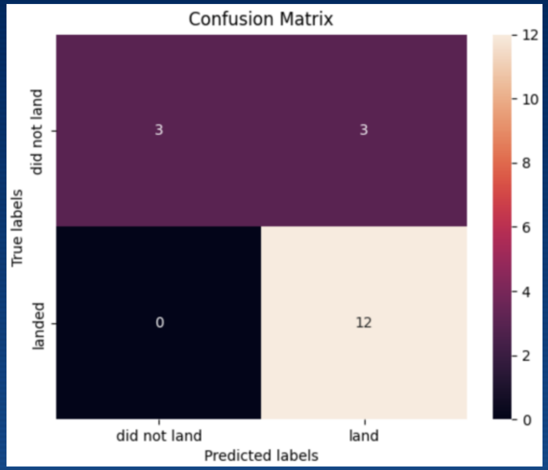
An interactive dashboard bulit through Plotly Dash will provide the following:

- Dropdown list with launch sites
    - Allows the user to select all launch sites or a particular site
- Pie chart showing successful launches
    - Allows the user to see successful and unsuccessful launches as percentage of total
- Scatter chart showing payload mass vs. success rate by booster version
    - Allows the user to see the correlation between payload and launch success
- Slider of the payload mass range
    - Allows the user to select payload mass range

Predictive Analytics

A machine learning model created to determine the best performing model through three main phases:

- Building the model
  - Create NumPy array from the class column
  - Standardize, fit and transform the data
  - Split the data using train_test_split
  - Create a GridSearchCV object with cv=10 for parameter optimization
- Evaluating the model
  - Apply GridSearchCV on different algorithms: logistic regression, support vector machine, decision tree, K-Nearest Neighbor
  - Calculate the accuracy on the test data using .score() for all models
  - Assess the confusion matrix for all models
- Finding best performing model
  - Identify the best model using Jaccard_Score, F1_Score and Accuracy



Results

Results Summary

Exploratory Data Analytics

- KSC LC-39A had the highest success rate among landing sites
- Orbits ES-L1, GEO, HEO and SSO have a 100% success rate
- Launch success improved over time

Visual Analytics

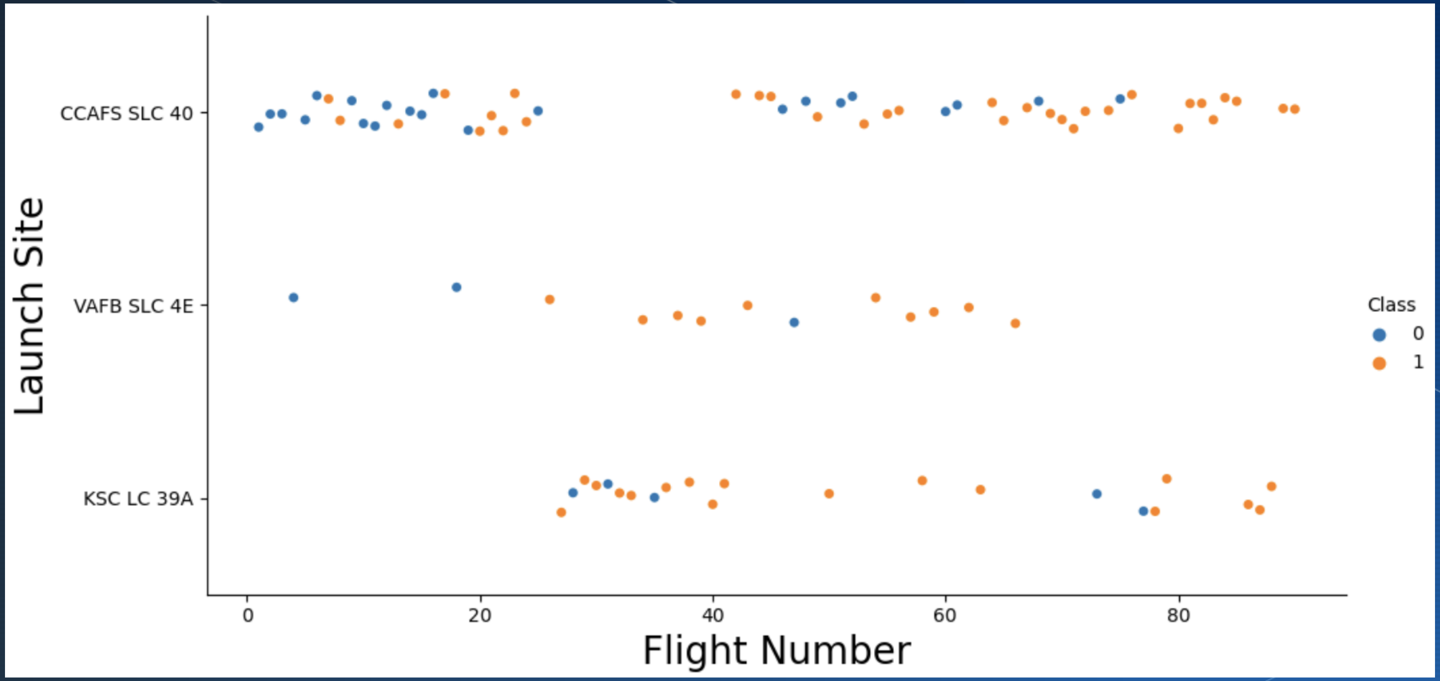
- Most launch sites are near the equator, and all are close to the coast
- Launch sites are far enough away from anything that a failed launch can damage (city, highway, railway) while close enough to bring people and material to support launch activities

Predictive Analytics

- Decision tree model is the best predictive model for the dataset

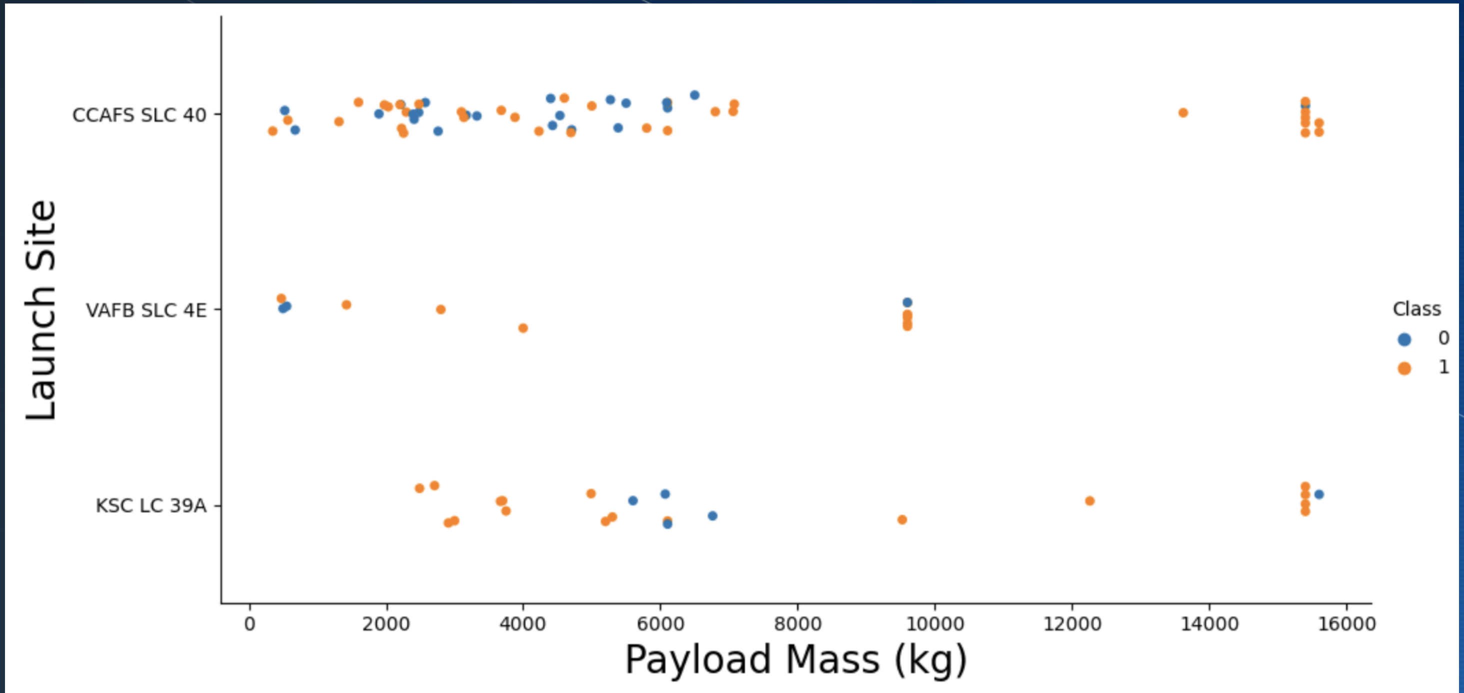
Flight Number vs. Launch Site

- Earlier flights had a lower success rate (blue = failure)
- Later flights had a higher success rate (orange = success)
- Over half of the total launches were at the CCAFS SLC 40 site
- KSC LC 39A and VAFB SLC 4E had a higher success launch rate



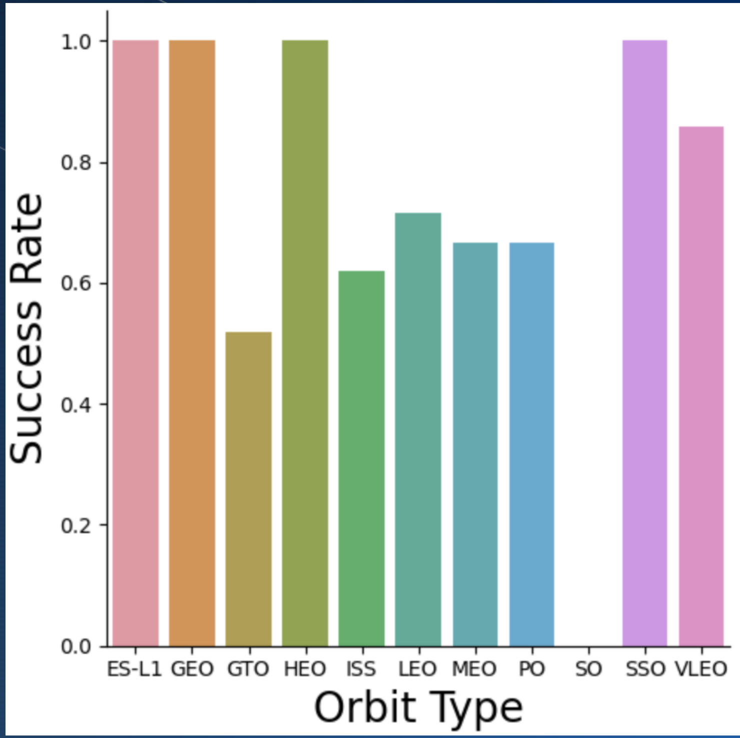
Payload vs. Launch Site

- The higher the payload mass (kg), the higher the success rate
- VAFB SLC 4E did not launch anything with a payload greater than 10,000 kg
- KSC LC 39A had a 100% success rate with launches less than ~5,500 kg



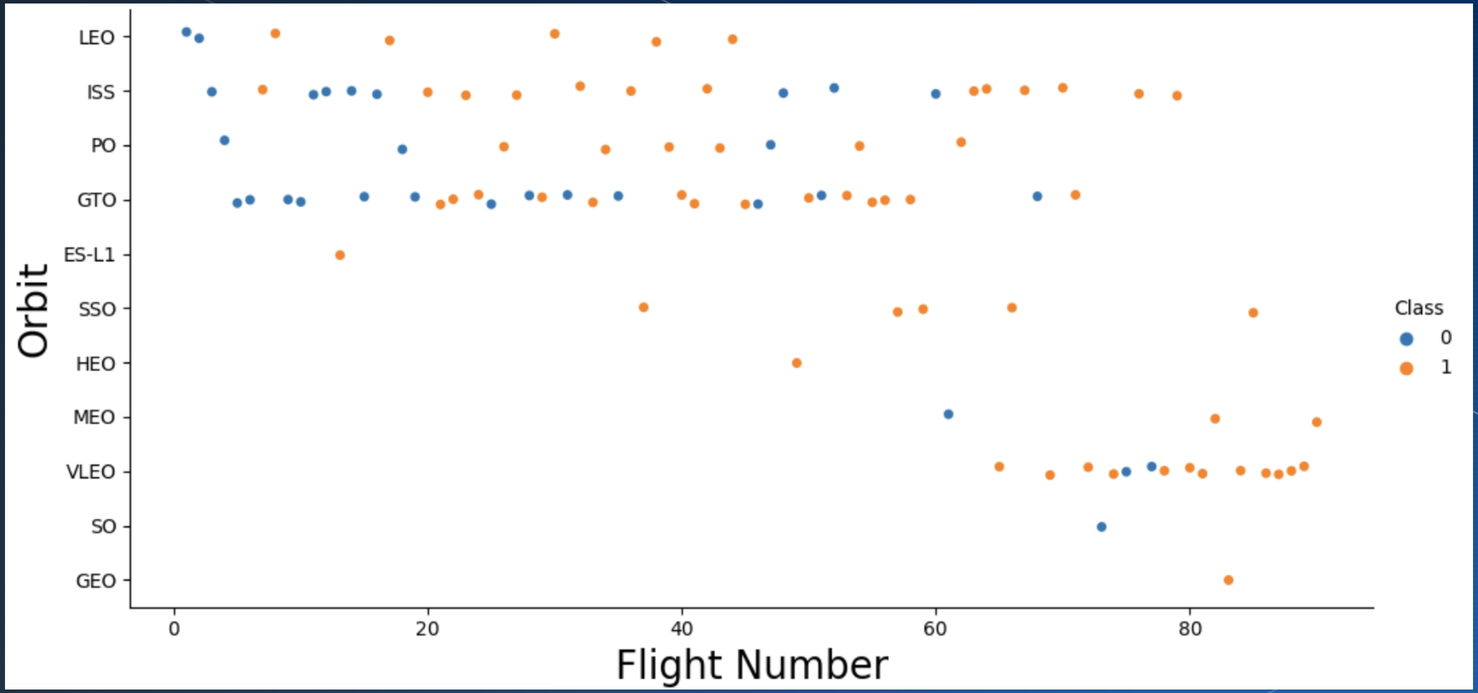
Success Rate vs. Orbit

- 100% success rate: ES-L1, GEO, HEO, SSO
- 50%-80% success rate: GTO, ISS< LEO, MEO, PO, VLEO
- 0% success rate: SO



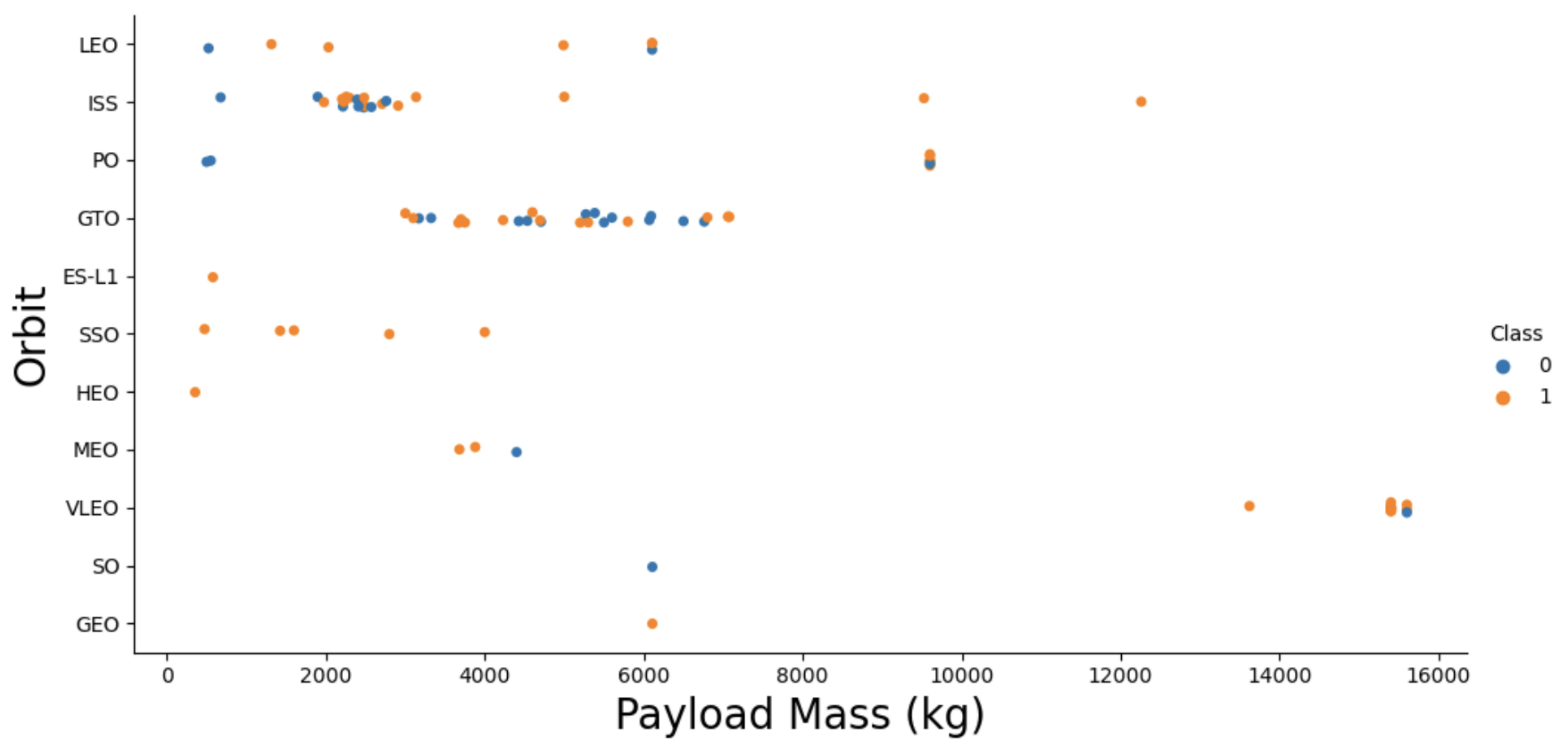
Flight Number vs. Orbit

- The success rate typically increases with the number of flights for each orbit
- LEO orbit has the highest success rate with the increase of flights
- GTO orbit, howev er, does not match this trend and has no relationship between flight number and the orbit



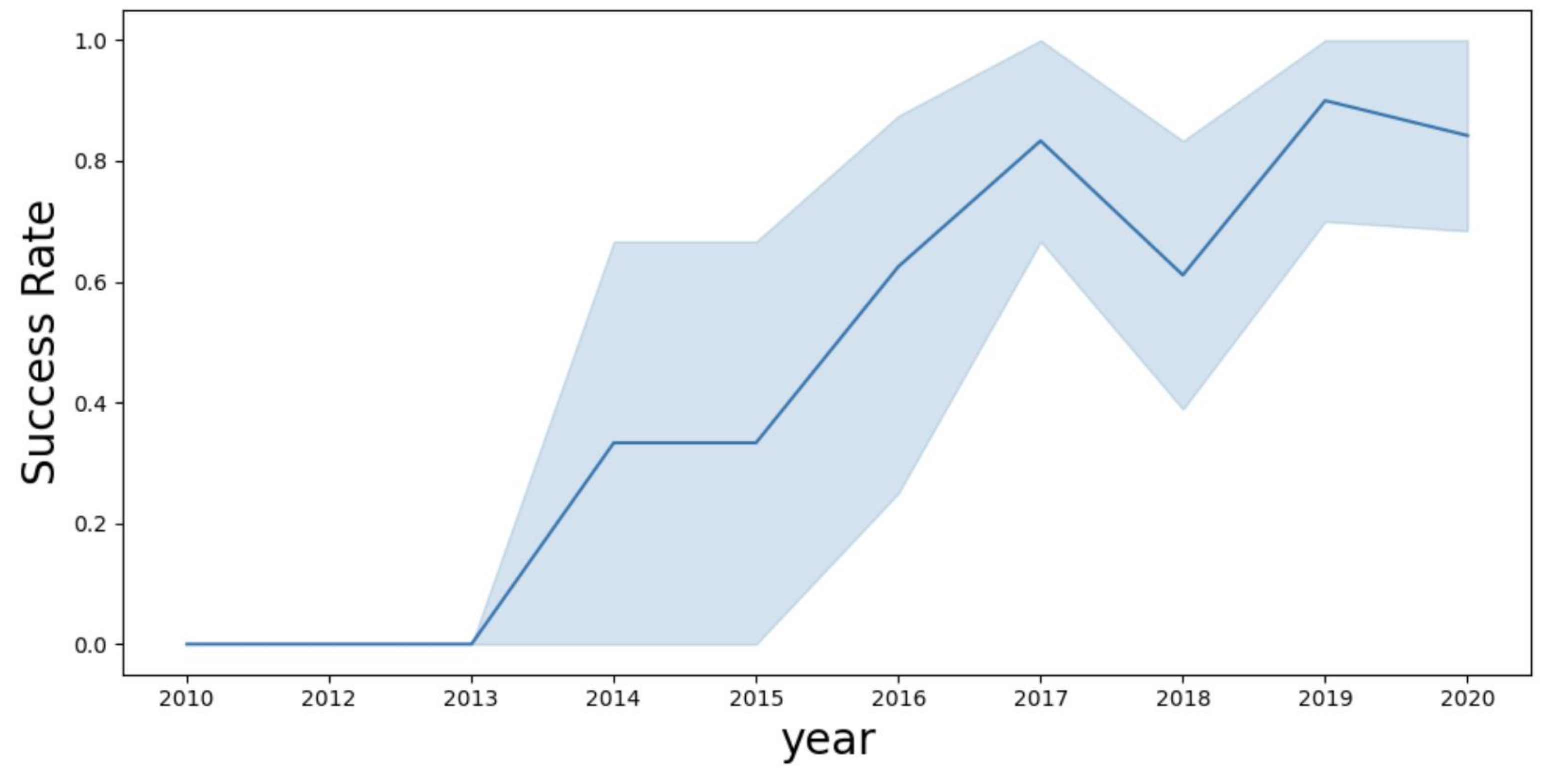
Payload vs. Orbit

- Heav ier payloads worked better with ISS, LEO and PO orbits
- GTO orbits had mixed results with heavier payloads



Launch Success Yearly Trend

- Launch success rate increased from 2013-2017 and 2018-2019
- Despite dips in 2018 and 2020, launch success rate improv ed ov er time



Launch Site Information

Launch Site Names:

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

```
%sql ibm_db_sa://yyy33800:dwNKg8J3L0IBd6CP@1bbf73c5-d84a-4bb0-
%sql SELECT DISTINCT(launch_site) FROM spacextbl;
```

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Records with launch site starting with 'CCA'

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No attempt
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attempt
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No attempt

```
%sql SELECT * \
    FROM spacextbl \
    WHERE launch_site LIKE'CCA%' LIMIT 5;
```

Payload Mass

- The total payload mass carried by boosters by NASA (CRS)

```
%sql SELECT SUM(payload_mass__kg_) \
FROM spacextbl \
WHERE customer = 'NASA (CRS)';
```

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

SUM(payload_mass__kg_)
45596.0

- The average payload mass carried by booster version F9 v1.1

```
%sql SELECT AVG(payload_mass__kg_) \
    FROM spacextbl \
    WHERE booster_version = 'F9 v1.1';
```

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

AVG(payload_mass__kg_)
2928.4

Mission and Landing Info

- The 1st successful landing in ground pad was on 01/08/2018

```
%sql SELECT MIN(date) \
FROM spacextbl \
WHERE landing_outcome = 'Success (ground pad)'
```

* sqlite:///my_data1.db

Done.

MIN(date)
01/08/2018

- The names of the boosters which had success in drone ships and had a payload greater than 4,000 kg but less than 6,000 kg

```
%sql SELECT payload \
FROM spacextbl \
WHERE landing_outcome = 'Success (drone ship)' \
AND payload_mass__kg_ BETWEEN 4000 AND 6000;
```

* sqlite:///my_data1.db

Done.

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

- The total number of successful and failed mission outcomes

```
%sql SELECT mission_outcome, COUNT(*) AS total_number \
FROM spacextbl \
GROUP BY mission_outcome;
```

* sqlite:///my_data1.db

Done.

Mission_Outcome	total_number
None	898
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters

Listed are the boosters that were carrying the maximum payload mass.

```
%sql SELECT booster_version \
FROM spacextbl \
WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM spacextbl);
```

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

Listed are all the 2015 failed launch landings on drone ships, along with the month, date, booster v ersion, and launch site

```
%sql SELECT substr(Date,4,2) AS month, date, booster_version, launch_site, [Landing_Outcome] \
FROM spacextbl \
WHERE [Landing_Outcome] = 'Failure (drone ship)' AND substr(Date,7,4) = '2015';
```

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

Done.
```

month	Date	Booster_Version	Launch_Site	Landing_Outcome
10	01/10/2015	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	14/04/2015	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

Count of Successful Landings

Count of all the successful landings spanning from 06/04/2010 to 03/20/2017

```
%%sql SELECT [Landing_Outcome], COUNT(*) AS count_outcomes
FROM spacextbl
WHERE date BETWEEN '04-06-2010' AND '20-03-2017' GROUP BY [Landing_Outcome] ORDER BY count_outcomes DESC;
```

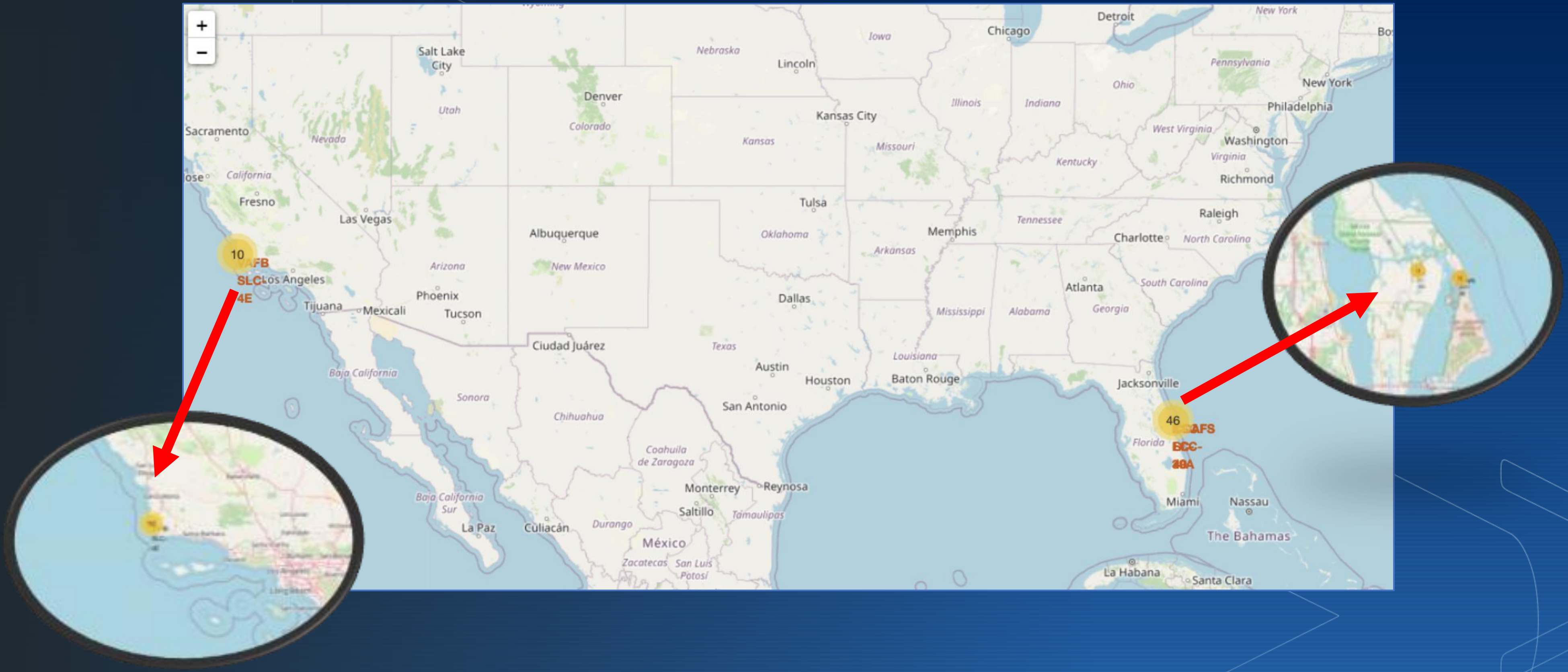
```
* sqlite:///my_data1.db
Done.
```

Landing_Outcome	count_outcomes
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	7
Failure (drone ship)	3
Failure	3
Failure (parachute)	2
Controlled (ocean)	2
No attempt	1

Launch Site Analysis

Launch Sites

Sites are on the coasts of the USA, closest to the equator where it's easier to launch into equatorial orbit. Rockets launched at this site get an additional natural boost –due to the rotational speed on Earth - that helps save on cost of putting in extra fuel and boosters.

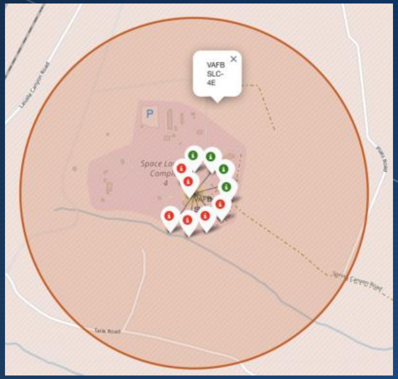


Launch Outcomes

- At each launch site, green markers indicate a successful launch while red markers indicate an unsuccessful launch
- KSC LC-39 had a 77% success rate, VAFB SLC-4E had 60%, CCAFS SLC-40 had 43% and CCAFS LC-40 had 26%

KSC LC-39

VAFB SLC-4E



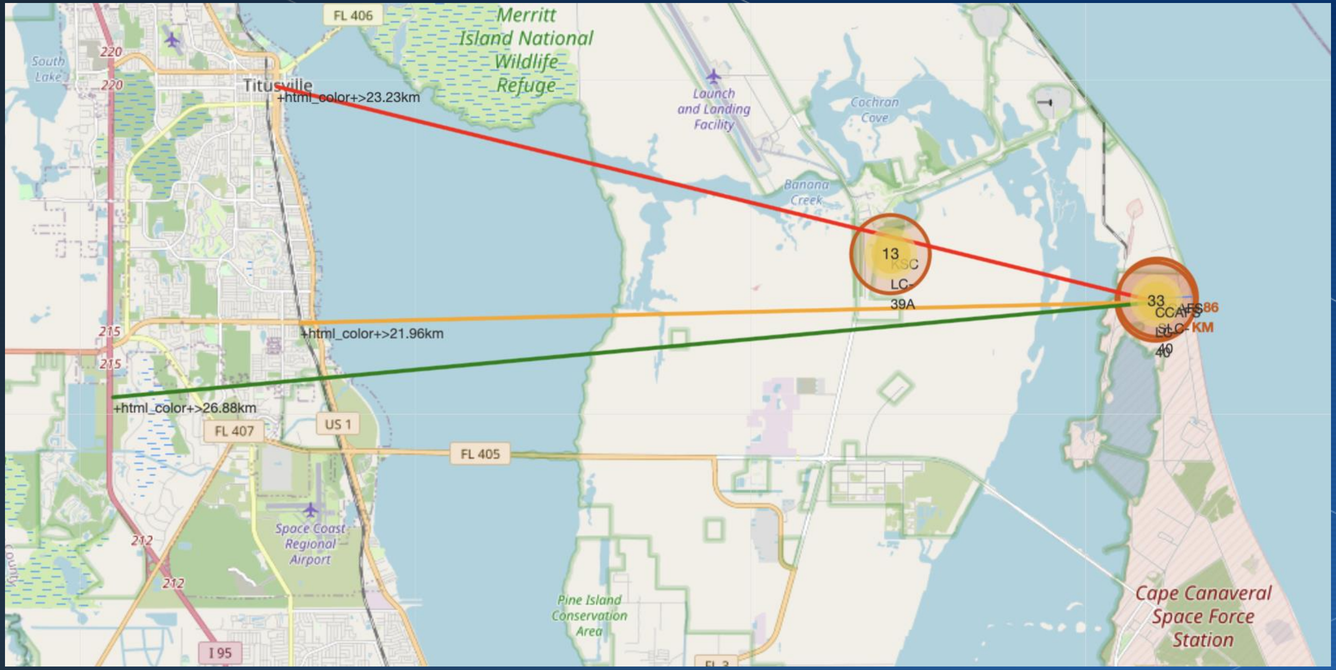
CCAFS SLC-40

CCAFS LC-40

Distance to Proximities

CCAFS SLC-40

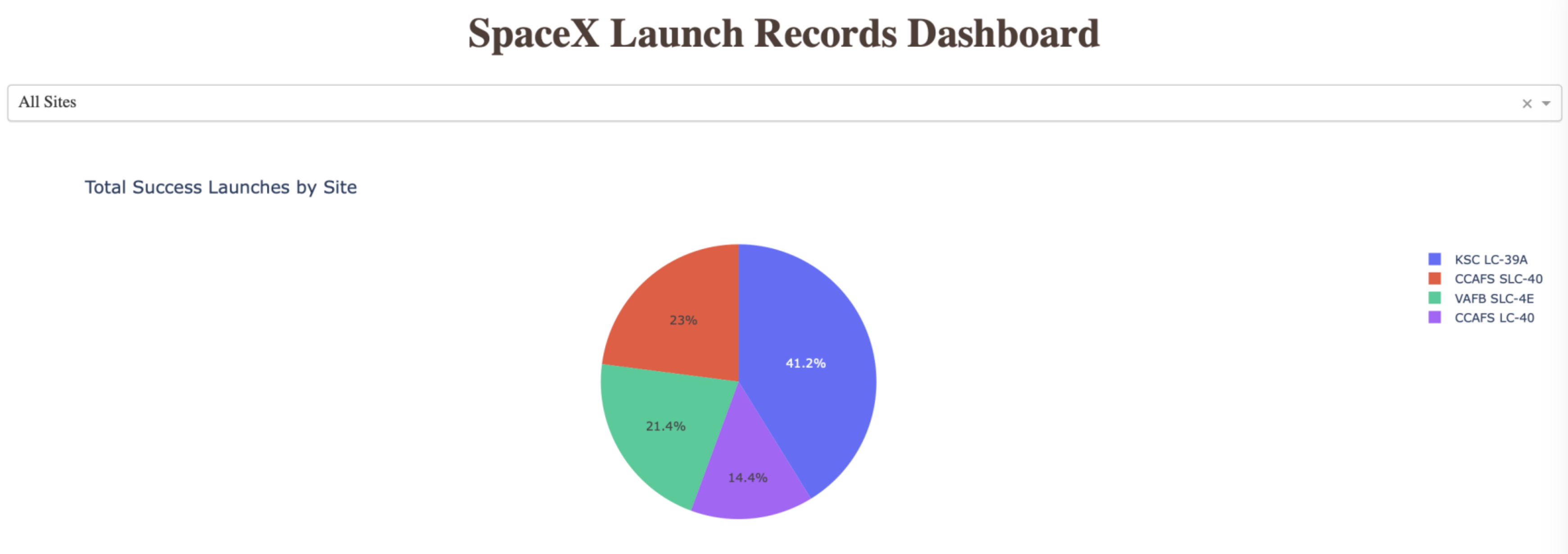
- 0.86km to nearest coastline (blue)
- 21.96km to nearest railway (yellow)
- 23.23km to nearest city (red)
- 26.88km to nearest highway (green)



Plotly Dashboard

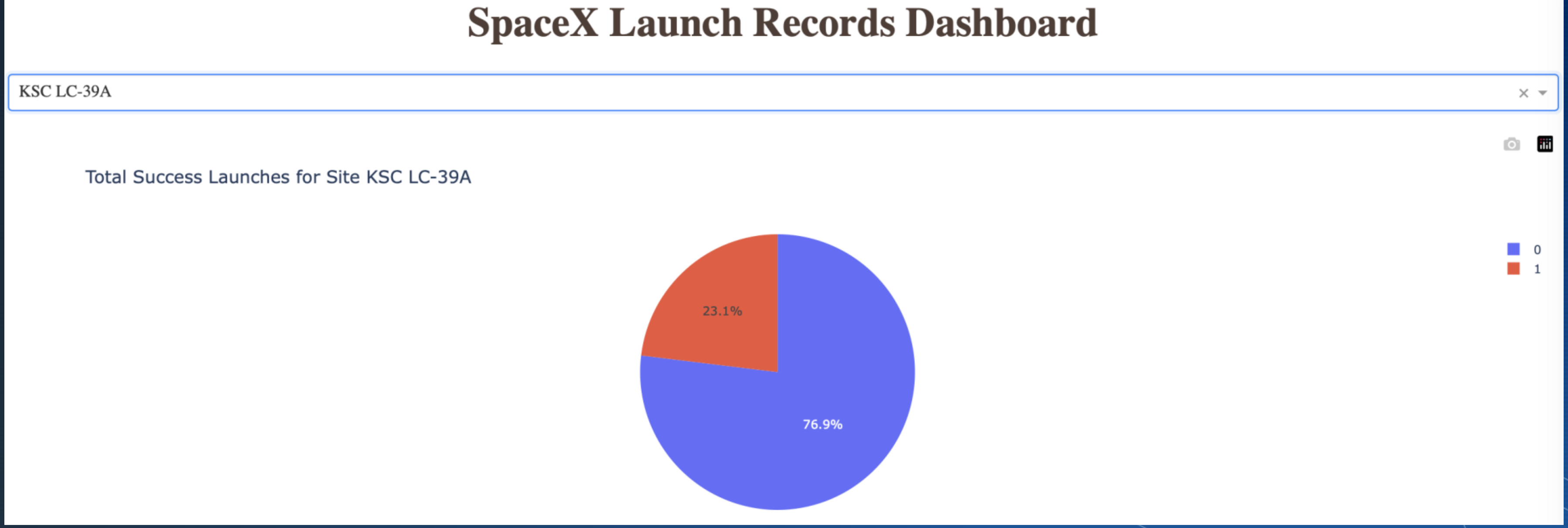
Launch Success by Site

Shown below is the success rate shown as percentages of the total launches. KSC LC-39A had the most successful launches amongst launch sites.



Launch Success (KSC LC-39A)

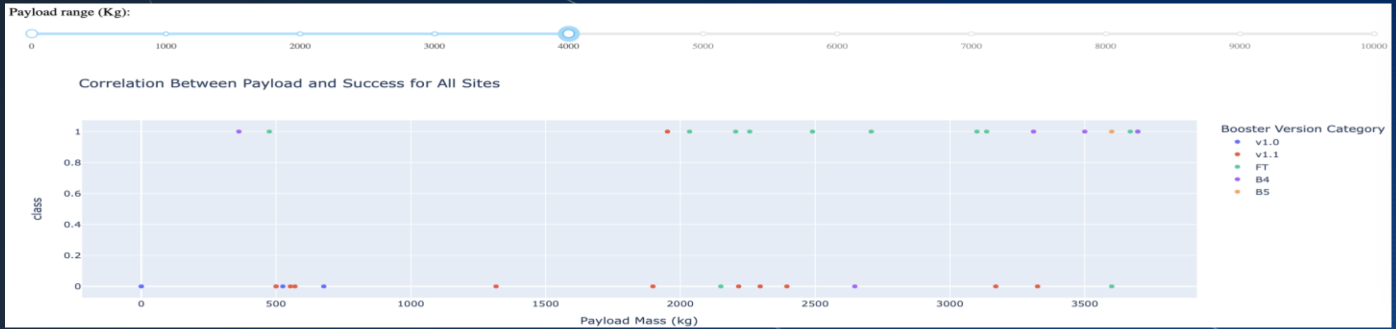
KSC LC-39A had the highest success rate amongst launch sites at 76.9%. There were 10 successful launches and 3 failed launches.



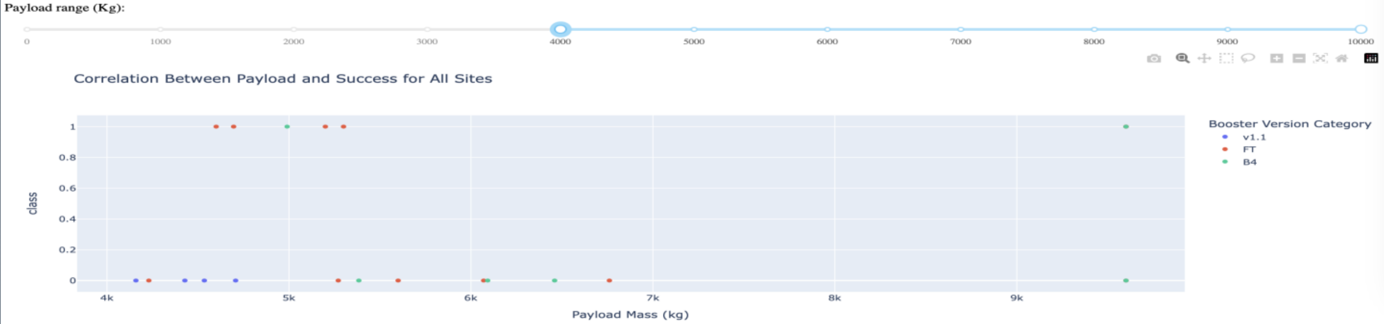
Payload Mass vs. Launch Outcome

By booster version

- The success rate for low weighted payloads (0-4,000kg) was higher than that of the heavy weighted payloads (4,000-10,000kg).



Low weighted payload



Heavy weighted payload

Predictive Analytics

Confusion Matrices

A confusion matrix summarizes the performance of a classification algorithm. All four matrices (for the four algorithms used) were identical, and the fact that there were three false positives is less than ideal (unsuccessful landings marked as successful by the classifier).

The confusion matrix outputs are as followed:

- 12 true positives
- 3 true negatives
- 3 false positives
- 0 false negatives

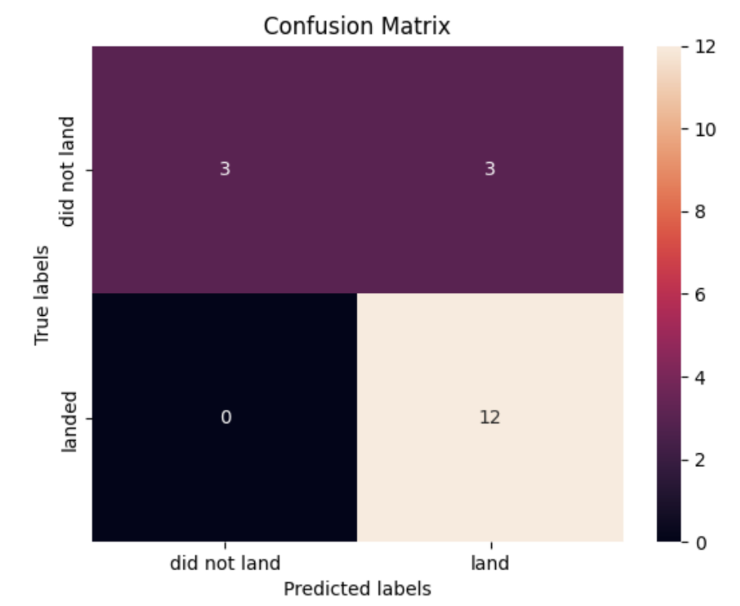
Precision = TP / (TP + FP) = 12 / 15 = **0.8**

Recall = TP / (TP + FN) = 12 / 12 = **1**

F1 Score = 2 * (Precision * Recall) / (Precision + Recall

- 2 * (0.8)/(1.8) = **0.89**

Accuracy = (TP + TN) / (TP +TN + FP + FN) = **0.833**



Classifications

All the models performed at about the same level and had the same scores and accuracy, but this is likely due to having a smaller dataset. The decision tree model outperformed the rest when looking at the best score.

```
accuracy = [svm_cv_score, logreg_score, knn_cv_score, tree_cv_score]
accuracy = [i * 100 for i in accuracy]

method = ['Support Vector Machine', 'Logistic Regression', 'K Nearest Neighbour', 'Decision Tree']
models = {'ML Method':method, 'Accuracy Score (%)':accuracy}

ML_df = pd.DataFrame(models)
ML_df

from sklearn.metrics import jaccard_score, f1_score

# Examining the scores from Test sets
jaccard_scores = [
                  jaccard_score(Y_test, logreg_yhat, average='binary'),
                  jaccard_score(Y_test, svm_yhat, average='binary'),
                  jaccard_score(Y_test, tree_yhat, average='binary'),
                  jaccard_score(Y_test, knn_yhat, average='binary'),
                 ]

f1_scores = [
             f1_score(Y_test, logreg_yhat, average='binary'),
             f1_score(Y_test, svm_yhat, average='binary'),
             f1_score(Y_test, tree_yhat, average='binary'),
             f1_score(Y_test, knn_yhat, average='binary'),
            ]

accuracy = [logreg_score, svm_cv_score, tree_cv_score, knn_cv_score]

scores_test = pd.DataFrame(np.array([jaccard_scores, f1_scores, accuracy]), index=['Jaccard_Score', 'F1_Score', 'Accuracy'] , columns=['LogReg', 'SVM', 'Tree', 'KN
scores_test

models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8892857142857142
Best params is : {'criterion': 'gini', 'max_depth': 16, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_split': 10, 'splitter': 'random'}
```

Conclusion

Following all the research provided, it can be concluded that:

- Launch success has steadily increased over time
- The orbits ES-L1, GEO, HEO and SSO had 100% success rate
- KSC LC-39A site had the highest success rate among all launch sites
- All launch sites are on the coast and near the equator, which adds an additional natural boost and helps save on costs of fuel and boosters
- The predictive models all performed similarly on the test sets, with the decision tree model slightly outperforming