IBM Developer
SKILLS NETWORK

How SpaceX is Winning the Space Race?

Simon Nadar

14th Sept 2023

Outline

Executive Summary

Summary of methodologies

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

- **Collect** data using SpaceX REST API and web scraping techniques.
- **Wrangle** data to create success/fail outcome variable.
- **Explore** data with data visualization techniques, considering the following factors: payload, launch site, flight number and yearly trend
- **Analyze** the data with SQL, calculating the following statistics: total payload, payload range for successful launches, and total # of successful and failed outcomes
- **Explore** launch site success rates and proximity to geographical markers.
- **Visualize** the launch sites with the most success and successful payload ranges.
- **Build Models** to predict landing outcomes using logistic regression, support vector machine (SVM), decision tree and K-nearest neighbor (KNN)

Summary of results

Exploratory Data Analysis:

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

Visualization / Analytics:

- Most launch sites are near the equator, and all are close to the coast

Predictive Analytics

- All models performed similarly on the test set.
- The decision tree model slightly outperformed others when looking at .best_score_

Introduction

Project Background

SpaceX, a leader in the space industry, strives to make space travel affordable for everyone. Its accomplishments include sending spacecraft to the international space station, launching a satellite constellation that provides internet access and sending manned missions to space. A big part in making all this possible is because the rocket launches are relatively inexpensive ($62 million per launch) due to its novel reuse of the first stage of its Falcon 9 rocket. Other providers, which are not able to reuse the first stage, cost upwards of $165 million each. By determining if the first stage will land, we can determine the price of the launch. To do this, we can use public data and machine learning models to predict whether SpaceX – or a competing company – can reuse the first stage.

Problems to Explore

-The interaction amongst various features that determine the success rate of a successful landing.

-Rate of successful landings over time

-Best predictive model for successful landing (binary classification)

Section 1

Methodology

Methodology

Executive Summary

- **Collect data** using SpaceX REST API and web scraping techniques
- **Wrangle data** – by filtering the data, handling missing values and applying one hot encoding – to prepare the data for analysis and modeling
- **Explore data** via EDA with SQL and data visualization techniques
- **Visualize** the data using Folium and Plotly Dash
- **Build Models** to predict landing outcomes using classification models. Tune and evaluate models to find best model and parameters

Data Collection API

Steps

- **Request data** from SpaceX API (rocket launch data)
- **Decode response** using .json() and convert to a dataframe using .json_normalize()
- **Request** information about the **launches** from SpaceX API using custom functions
- **Create dictionary** from the data
- **Create dataframe** from the dictionary
- **Filter dataframe** to contain only Falcon 9 launches
- **Replace missing values** of Payload Mass with calculated .mean()
- **Export data** to csv file

Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.
- The link to the notebook is https://github.com/Simon033/SpaceX-IBM-Data-Science-Capstone/blob/5e6553c8d64be4fd60282a7bb3a09c3e3288c98f/01-Data-Collection-SpaceX-API.ipynb

1. Get request for rocket launch data using API

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

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

2. Use json_normalize method to convert json result to dataframe

In [12]:
```
# Use json_normalize method to convert the json result into a dataframe

# decode response content as json
static_json_df = res.json()
```

In [13]:
```
# apply json_normalize
data = pd.json_normalize(static_json_df)
```

3. We then performed data cleaning and filling in the missing values

In [30]:
```
rows = data_falcon9['PayloadMass'].values.tolist()[0]

df_rows = pd.DataFrame(rows)
df_rows = df_rows.replace(np.nan, PayloadMass)

data_falcon9['PayloadMass'][0] = df_rows.values
data_falcon9
```

Data Collection Web-Scrapping

Steps

- **Request data** (Falcon 9 launch data) from Wikipedia
- **Create** *BeautifulSoup* object from HTML response
- **Extract** column names from HTML table header
- Collect data from **parsing HTML** tables
- Create dictionary from the data
- **Create dataframe** from the dictionary
- **Export** data to csv file

Data Collection - Scraping

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- The link to the notebook is https://github.com/Simon033/SpaceX-IBM-Data-Science-Capstone/blob/744b4d5e7393731c792ec4635ae14f371b9e9d15/02-Data-Collection-Web-Scrapping.ipynb

1. Apply HTTP Get method to request the Falcon 9 rocket launch page

```
In [4]: static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

```
In [5]: # use requests.get() method with the provided static_url
        # assign the response to a object
        html_data = requests.get(static_url)
        html_data.status_code

Out[5]: 200
```

2. Create a BeautifulSoup object from the HTML response

```
In [6]: # 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

```
In [7]: # Use soup.title attribute
        soup.title

Out[7]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

3. Extract all column names from the HTML table header

```
In [10]: column_names = []

         # Apply find_all() function with `th` element on first_launch_table
         # Iterate each th element and apply the provided extract_column_from_header() to get a column name
         # Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

         element = soup.find_all('th')
         for row in range(len(element)):
             try:
                 name = extract_column_from_header(element[row])
                 if (name is not None and len(name) > 0):
                     column_names.append(name)
             except:
                 pass
```

4. Create a dataframe by parsing the launch HTML tables
5. Export data to csv

Data Wrangling

Steps

- **Perform EDA** and determine data labels
- **Calculate**:

  # of launches for each site

  # and occurrence of orbit

  # and occurrence of mission outcome per orbit type]
- **Create binary** landing outcome column (dependent variable)
- **Export data** to csv file

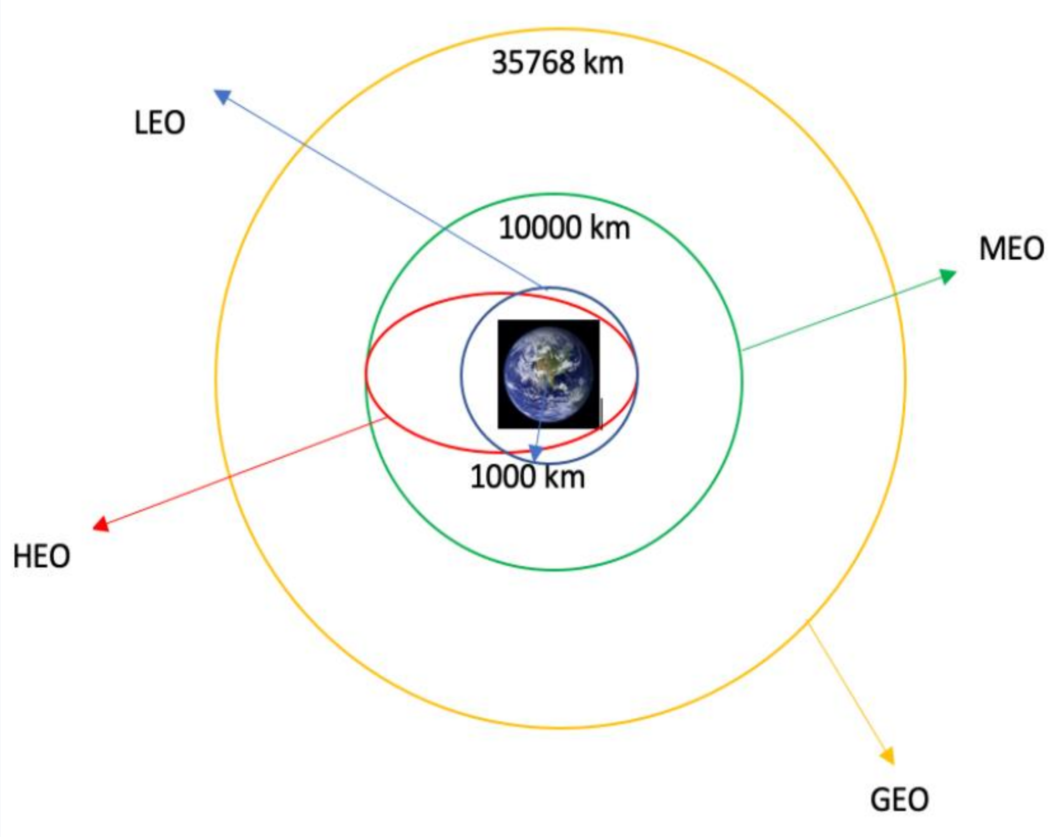
In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example,

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.

Data Wrangling

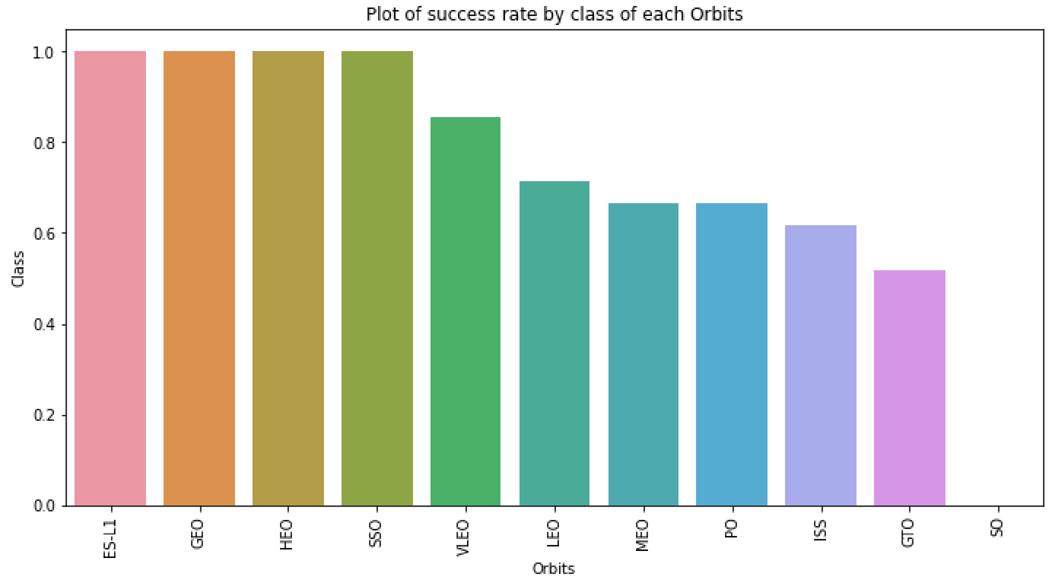


- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- The link to the notebook is https://github.com/Simon033/SpaceX-IBM-Data-Science-Capstone/blob/ff65b7d453fd7bb5e41bc48df46d183e53268714/03-Data-Wrangling.ipynb

EDA with Data Visualization

Charts Used -

- Flight Number vs Payload
- Flight Number vs. Launch Site
- Payload Mass (kg) vs. Launch Site
- Payload Mass (kg) vs. Orbit type



Analysis –

- View relationship by using scatter plots. The variables could be useful for machine learning if a relationship exists
- Show comparisons among discrete categories with bar charts. Bar charts show the relationships among the categories and a measured value.

The link to the notebook is https://github.com/Simon033/SpaceX-IBM-Data-Science-Capstone/blob/b6a54565bcfeb318e844566b99e97958b4a58901/05-EDA-with-Visualization.ipynb

EDA with SQL

Applied EDA with SQL to get insight from the data.

For instance, queries like:

- Names of unique launch sites
- Total number of successful and failed missions
- Total payload mass carried by boosters launched by NASA (CRS)
- Average payload mass carried by booster version F9 v1.1.
- Names of booster versions which have carried the max payload
- Count of landing outcomes between 2010-06-04 and 2017-03-20 ranked (desc)

Link to Notebook:

https://github.com/Simon033/SpaceX-IBM-Data-Science-Capstone/blob/b6a54565bcfeb318e844566b99e97958b4a58901/04-EDA-with-SQL.ipynb

Interactive Map with Folium

Markers Indicating Launch Sites

- Added **blue** circle at NASA Johnson Space Center's coordinate with a popup label showing its name using its latitude and longitude coordinates
- Added **red** circles at all launch sites coordinates with a popup label showing its name using its name using its latitude and longitude coordinates

Colored Markers of Launch Outcomes

- Added colored markers of successful (**green**) and unsuccessful (**red**) launches at each launch site to show which launch sites have high success rates

Distances Between a Launch Site to Proximities

- Added colored lines to show distance between launch site CCAFS SLC40 and its proximity to the nearest ***coastline, railway, highway, and city***

Notebook Link: https://github.com/Simon033/SpaceX-IBM-Data-Science-Capstone/blob/33fd1311ff1d51b6b12a61b832e2dc93bafd64e3/06-Interactive-Visual-Maps-with-Folium.ipynb

Dashboard with Plotly Dash

Dropdown List with Launch Sites

- Allow user to select all launch sites or a certain launch site

Pie Chart Showing Successful Launches

- Allow user to see successful and unsuccessful launches as a percent of the total

Slider of Payload Mass Range

- Allow user to select payload mass range Pie

Scatter Chart Showing Payload Mass vs. Success Rate by Booster Version

- Allow user to see the correlation between Payload and Launch Success

Notebook Link: https://github.com/Simon033/SpaceX-IBM-Data-Science-Capstone/blob/33fd1311ff1d51b6b12a61b832e2dc93bafd64e3/07-Interactive-Visual-dashboard-with-Plotly-Dash-app.py

Predictive Analysis (Classification)

- Create NumPy array from the **Class** column
- **Standardize** the data with StandardScaler. Fit and transform the data.
- **Split** the data using train_test_split
- Create a **GridSearchCV** object with cv=10 for parameter optimization
- Apply GridSearchCV on **different algorithms**: logistic regression (LogisticRegression()), support vector machine (SVC()), decision tree (DecisionTreeClassifier()), K-Nearest Neighbor (KNeighborsClassifier())
- **Calculate accuracy** on the test data using .score() for all models
- Assess the **confusion matrix** for all models
- Identify the best model using ***.best_score_***

Notebook Link: https://github.com/Simon033/SpaceX-IBM-Data-Science-Capstone/blob/45b19a12e1b44ca74cb12788f681e227eac17dc4/08-SpaceX-Machine-Learning-Prediction.ipynb

Results

Exploratory Data Analysis

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

Interactive Visual Analytics

- Most launch sites are near the equator, and all are close to the coast
- Launch sites are isolated far enough such that a failed launch won’t cause harm(city, highway, railway), while still close enough to bring people and material to support launch activities

Predictive Analysis

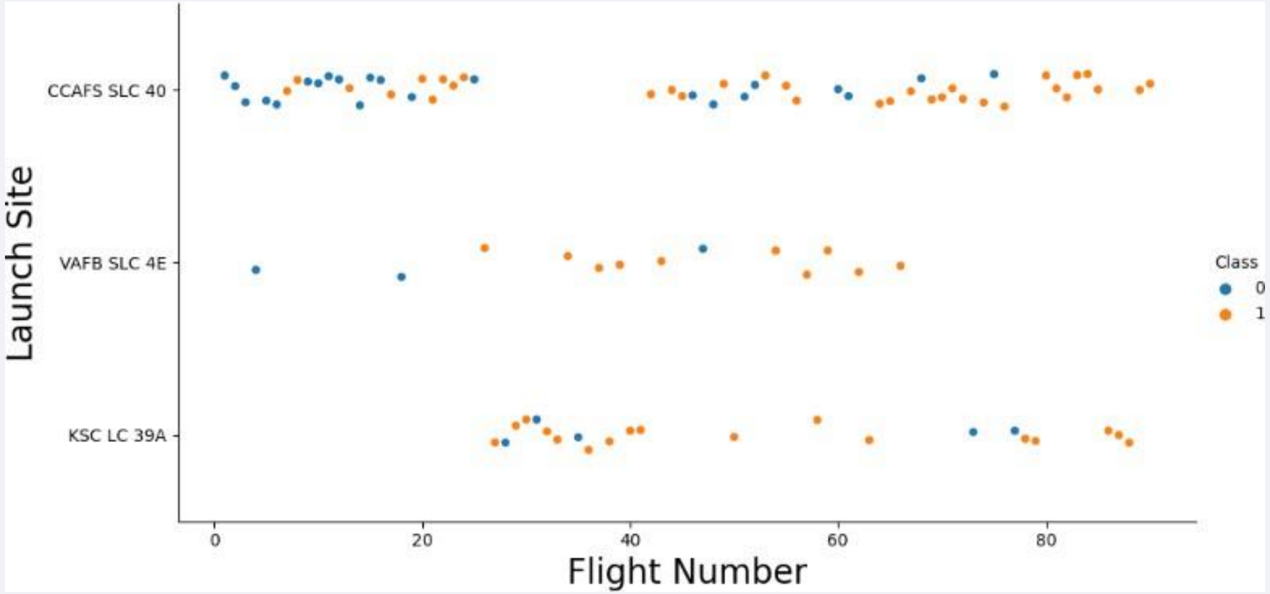
- Decision Tree model is the best predictive model for the dataset

Section 2

Insights drawn from EDA

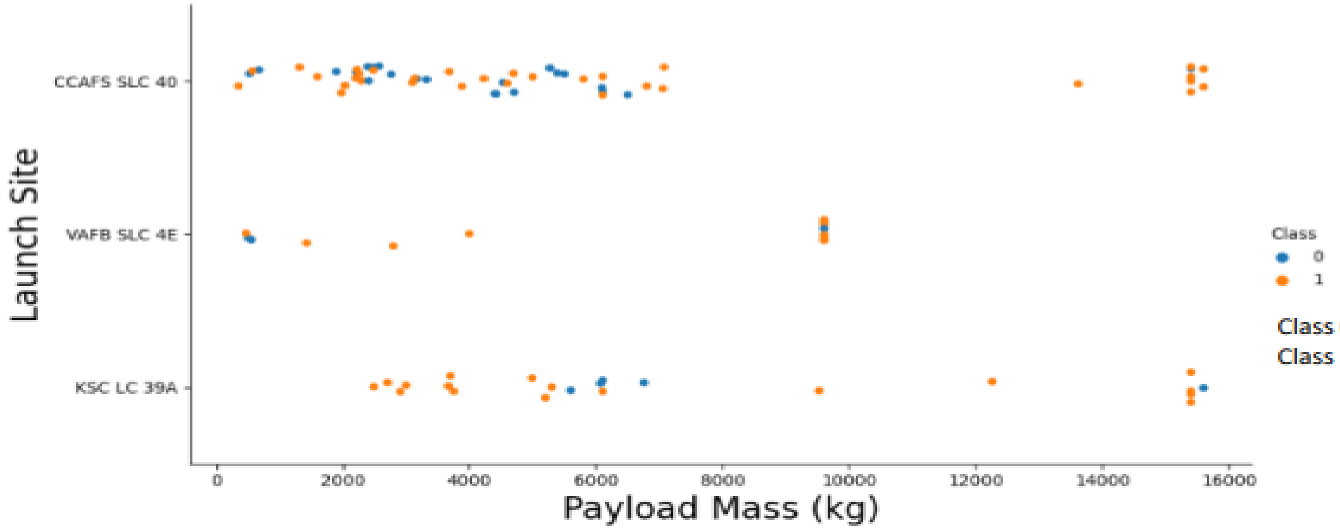
Flight Number vs Launch Site

- **Earlier flights** had a lower success rate (blue = fail)
- **Later flights** had a higher success rate (orange = success)
- Around half of launches were from CCAFS SLC 40 launch site
- VAFB SLC 4E and KSC LC 39A have higher success rates
- We can infer that new launches have a higher success rate



Payload vs. Launch Site

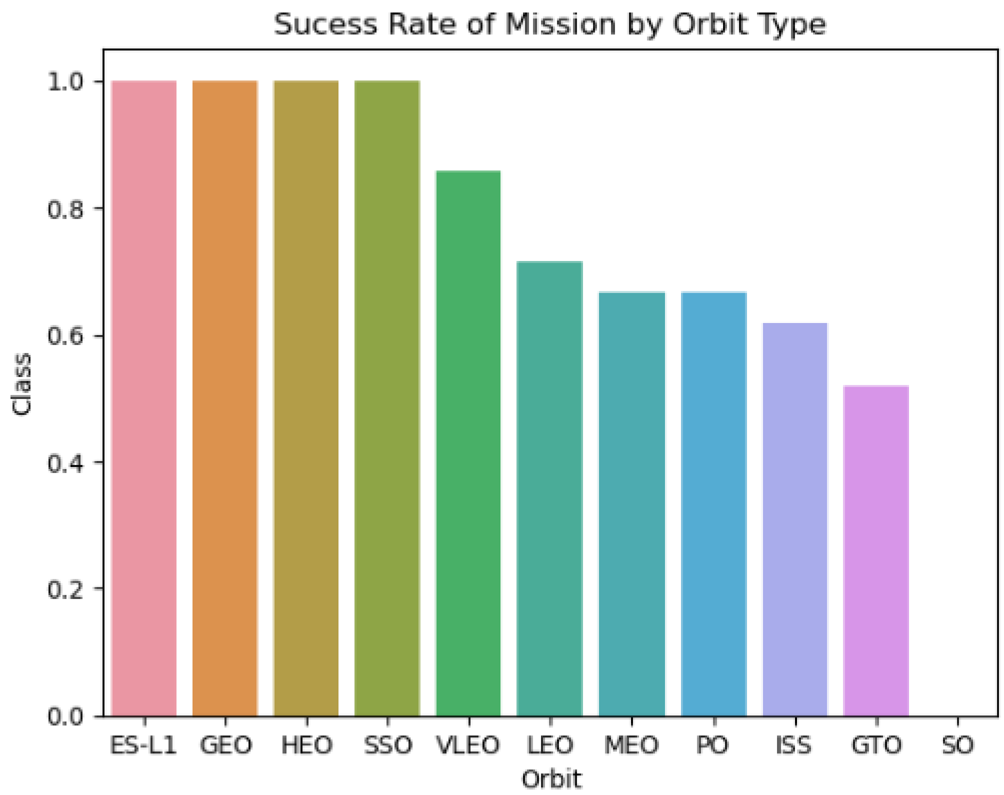
- Typically, the **higher** the **payload mass (kg)**, the higher the **success rate**
- Most launces with a payload greater than 7,000 kg were successful
- **KSC LC 39A** has a 100% success rate for launches less than 5,500 kg
- **VAFB SKC 4E** has not launched anything greater than ~10,000 kg



Class 0 = Fail
Class 1 = Success

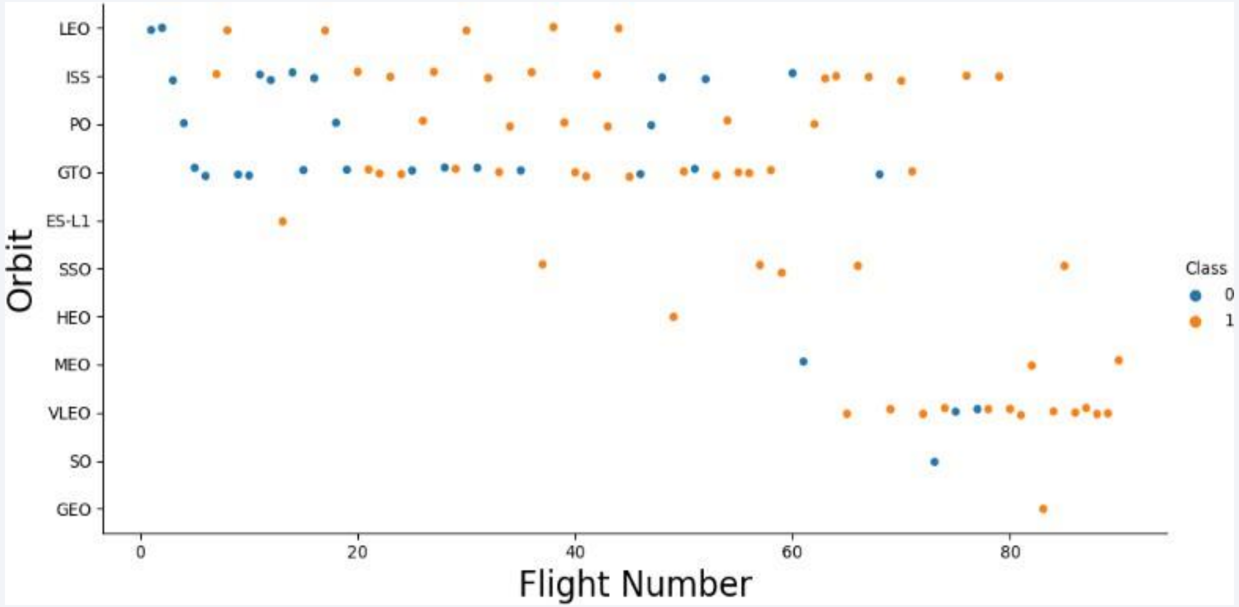
Success Rate vs. Orbit Type

- 100% Success Rate: ES-L1, GEO, HEO and SSO
- 50%-80% Success Rate: GTO, ISS, LEO, MEO, PO
- 0% Success Rate: SO



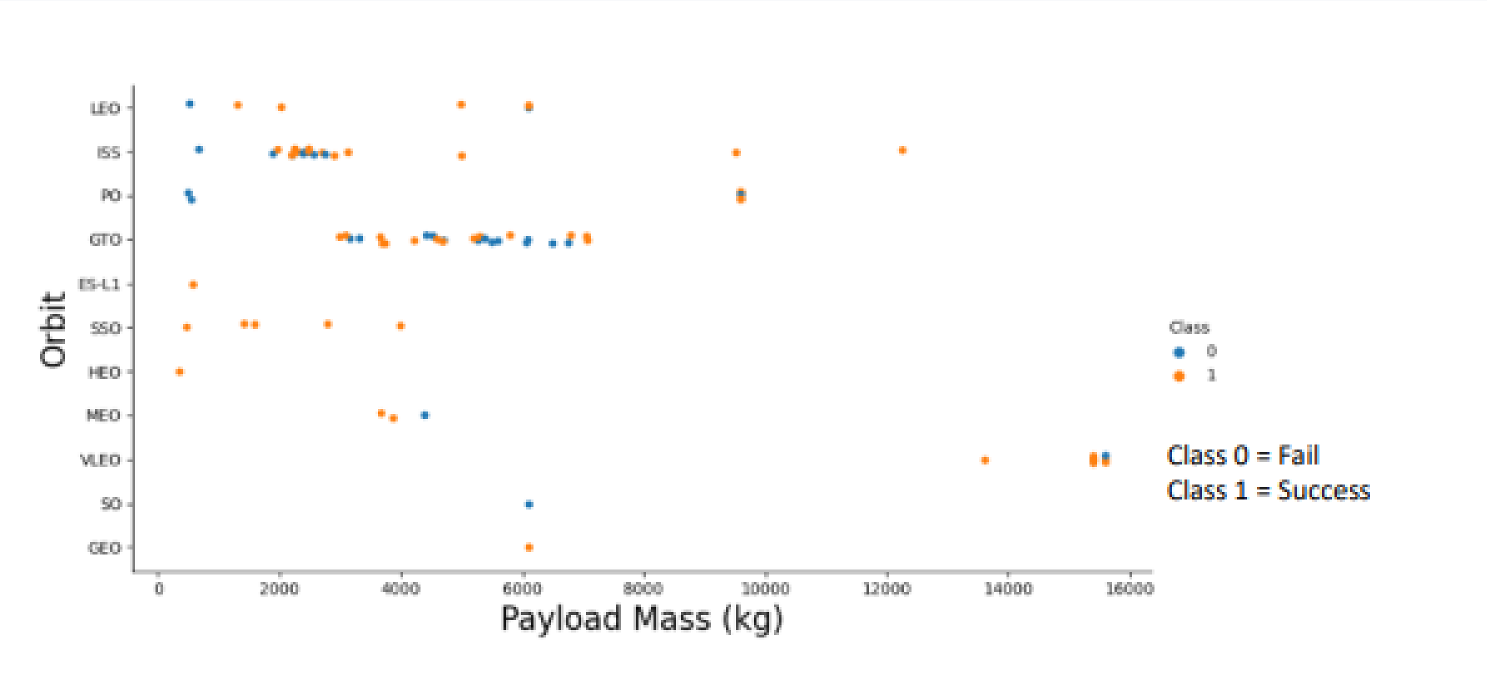
Flight Number vs. Orbit Type

- The success rate typically increases with the number of flights for each orbit
- This relationship is highly apparent for the LEO orbit
- The GTO orbit, however, does not follow this trend



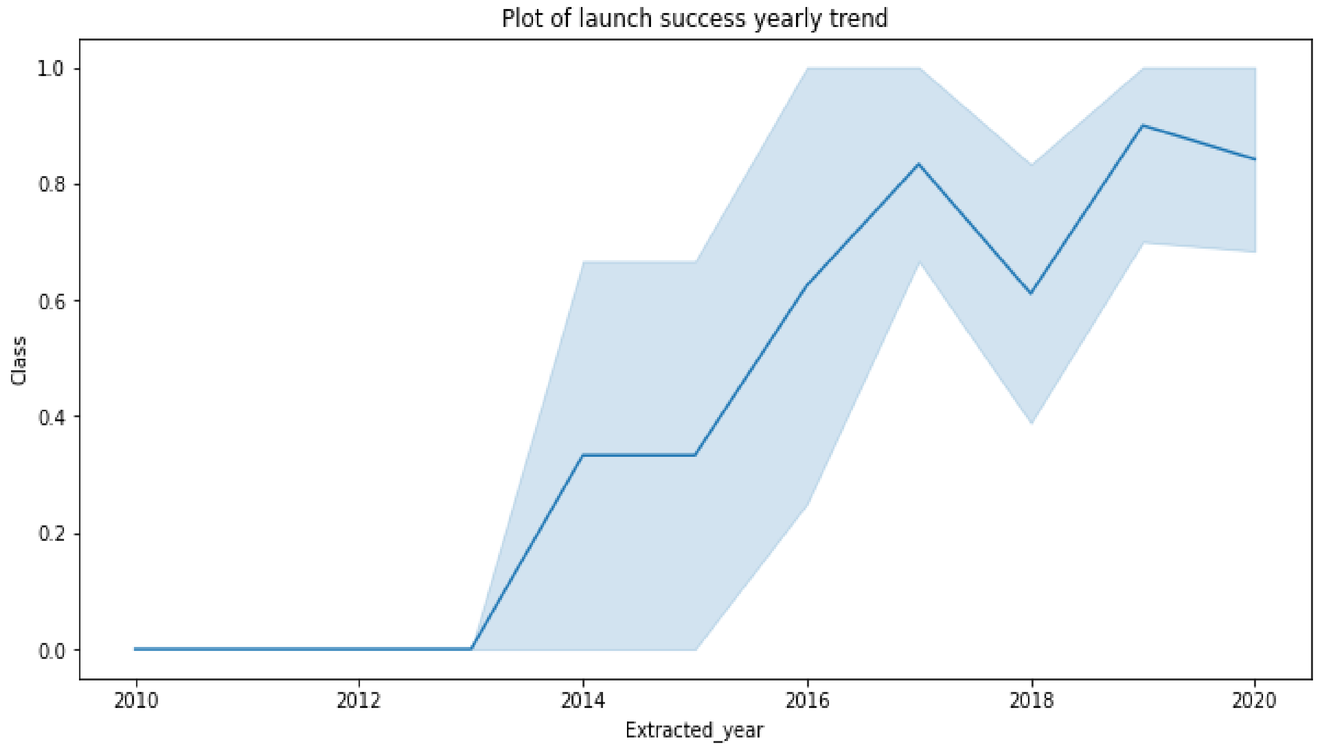
Payload vs. Orbit Type

- Heavy payloads are better and usually done with LEO, ISS and PO orbits
- The GTO orbit has mixed success with heavier payloads



Launch Success Yearly Trend

- The success rate improved from 2013-2017 and 2018-2019
- The success rate decreased from 2017-2018 and from 2019-2020
- Overall, the success rate has improved since 2013



All Launch Site Names

Launch Site Names

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

We used the **UNIQUE** key word to show only unique launch sites from the SpaceX data.

```
In [9]: %sql SELECT UNIQUE launch_site AS Launch_Sites FROM SPACEX;

 * ibm_db_sa://dqf44339:***@0c77d6f2-5da9-48a9-81f8-86b520b8
Done.
```

Out[9]:

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

Launch Site Names Begin with 'CCA'

LIKE clause is used to find 5 records where launch sites begin with `CCA`

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

```
 * ibm_db_sa://dqf44339:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90108kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing_outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass by Nasa (CRS)

A total of **45,596 kg** was carried by boosters launched by NASA (CRS)

```
%%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEX
WHERE CUSTOMER='NASA (CRS)'

 * ibm_db_sa://dqf44339:***@0c77d6f2-5da9-48a9-81
Done.
```

1
45596

WHERE clause is used to match customer record with 'Nasa (CRS)'

Average Payload Mass by F9 v1.1

An average of **2,928 kg** was carried by booster version F9 v1.1

```
%%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEX
WHERE booster_version ='F9 v1.1' ;

 * ibm_db_sa://dqf44339:***@0c77d6f2-5da9-48a9-81f
Done.
```

1
2928

WHERE clause is used to match booster version with 'F9 v1.1'

Landing & Mission Info

First Successful Landing in Ground Pad

The first successful landing outcome on ground pad was observed on 22nd December 2015

```
%%sql SELECT MIN(DATE) FROM SPACEX
WHERE landing_outcome='Success (ground pad)';
```

```
 * ibm_db_sa://dqf44339:***@0c77d6f2-5da9-48a9-8
Done.
```

1
2015-12-22

WHERE clause is used to find landing outcome entries matching *'Success(ground pad)'* then **MIN** function is used to find the earliest date

Boosters with payload between 4000kg and 6000kg that successfully landed in drone ship

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

```
%%sql SELECT booster_version FROM SPACEX WHERE landing_outcome='Success (drone ship)'
AND payload_mass__kg_ BETWEEN 4000 and 6000;
```

```
 * ibm_db_sa://dqf44339:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.
Done.
```

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

WHERE clause is used on landing outcome and payload mass in which a **BETWEEN** clause is used to filter entries between 4000-6000kg

Total Number of Successful and Failure Mission Outcomes

```
%%sql SELECT  mission_outcome,COUNT(mission_outcome)
FROM SPACEX GROUP BY mission_outcome;
```

```
 * ibm_db_sa://dqf44339:***@0c77d6f2-5da9-48a9-81f8-86
Done.
```

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

The entries are **GROUP BY** mission outcome then the mission outcome along with the **COUNT** of outcome are selected.

There are 100 successful mission outcome and 1 failure and out of the 100 success one has an unclear payload status

- **1** Failure in Flight
- **99** Success
- **1** Success (payload status unclear)

Boosters that Carried Maximum Payload

- **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**

A **Sub-Query** was used following the **WHERE** clause which compared the payload of each entry to the max payload and returned those that matched the **Max**

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

```
 * ibm_db_sa://dqf44339:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2i
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

Failed Landings on Drone Ship in 2015

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

```
%%sql SELECT booster_version, launch_site, landing_outcome  FROM SPACEX
WHERE DATE LIKE '%2015%' AND landing_outcome LIKE 'Failure (drone ship)'
```

```
 * ibm_db_sa://dqf44339:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90108
Done.
```


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

- A combinations of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions was used to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for the year 2015

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

Ranked (desc)

```
%%sql SELECT landing_outcome, COUNT(landing_outcome) AS COUNT
FROM SPACEX WHERE DATE BETWEEN '2010-06-04' and '2017-03-20'
GROUP BY landing_outcome ORDER BY COUNT DESC;
```

```
 * ibm_db_sa://dqf44339:***@0c77d6f2-5da9-48a9-81f8-86b520b8751
Done.
```

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

- Landing outcomes and the **COUNT** of landing outcomes are selected from the data and the **WHERE** clause is used to filter for landing outcomes **BETWEEN** 2010-06-04 to 2010-03-20.

- Then **GROUP BY** clause is applied to the landing outcomes column and the **ORDER BY** clause to order the grouped landing outcome in descending order.

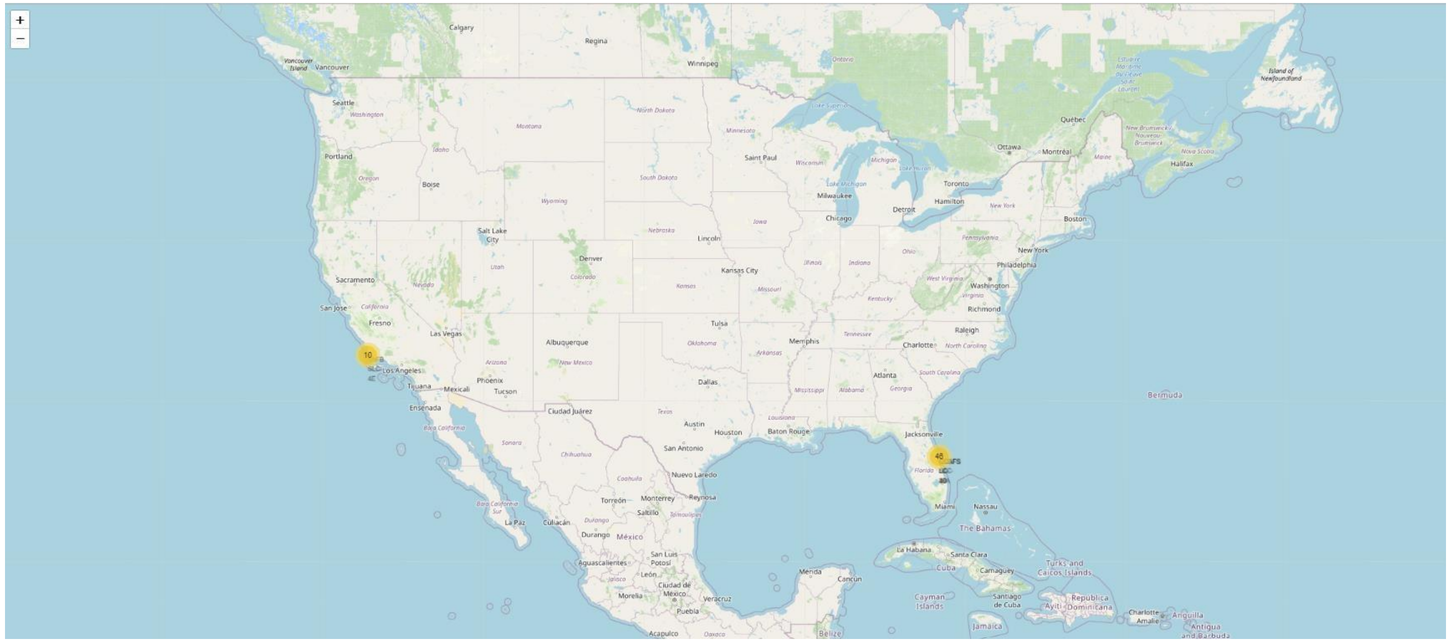
Section 4

Launch Sites Analysis

Launch Sites

Near Equator:

We can see all the launch sites are closer to equator as the closer the launch site to the equator, the **easier** it is **to launch** to equatorial orbit, and the more help you get from Earth's rotation for a prograde orbit. Rockets launched from sites near the equator get an **additional natural boost** - due to the rotational speed of earth - that **helps save the cost** of putting in extra fuel and boosters.



Launch Site Outcomes

At Each Launch Site

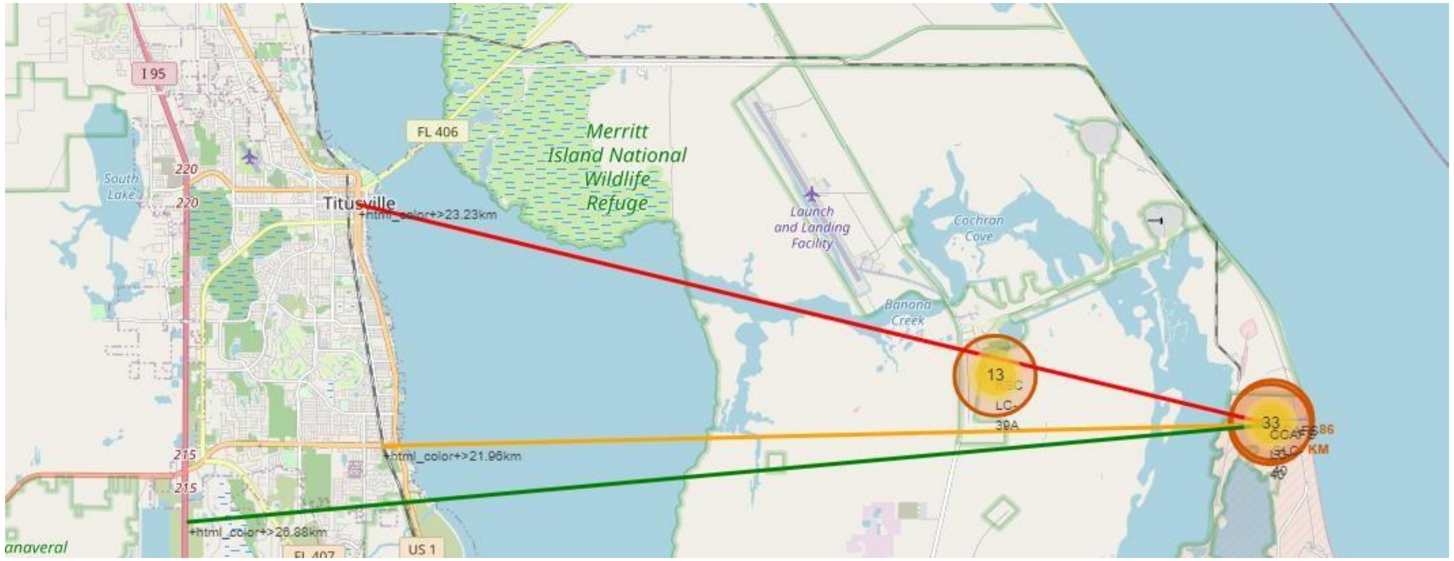
- Green markers for successful launches
- Red markers for unsuccessful launches
- Launch site **CCAFS SLC-40** has a **3/7 success rate** **(42.9%)**

Launch Site Proximities

CCAFS SLC-40

- **0.86 km** from nearest coastline
- **21.96 km** from nearest railway
- **23.23 km** from nearest city
- **26.88 km** from nearest highway

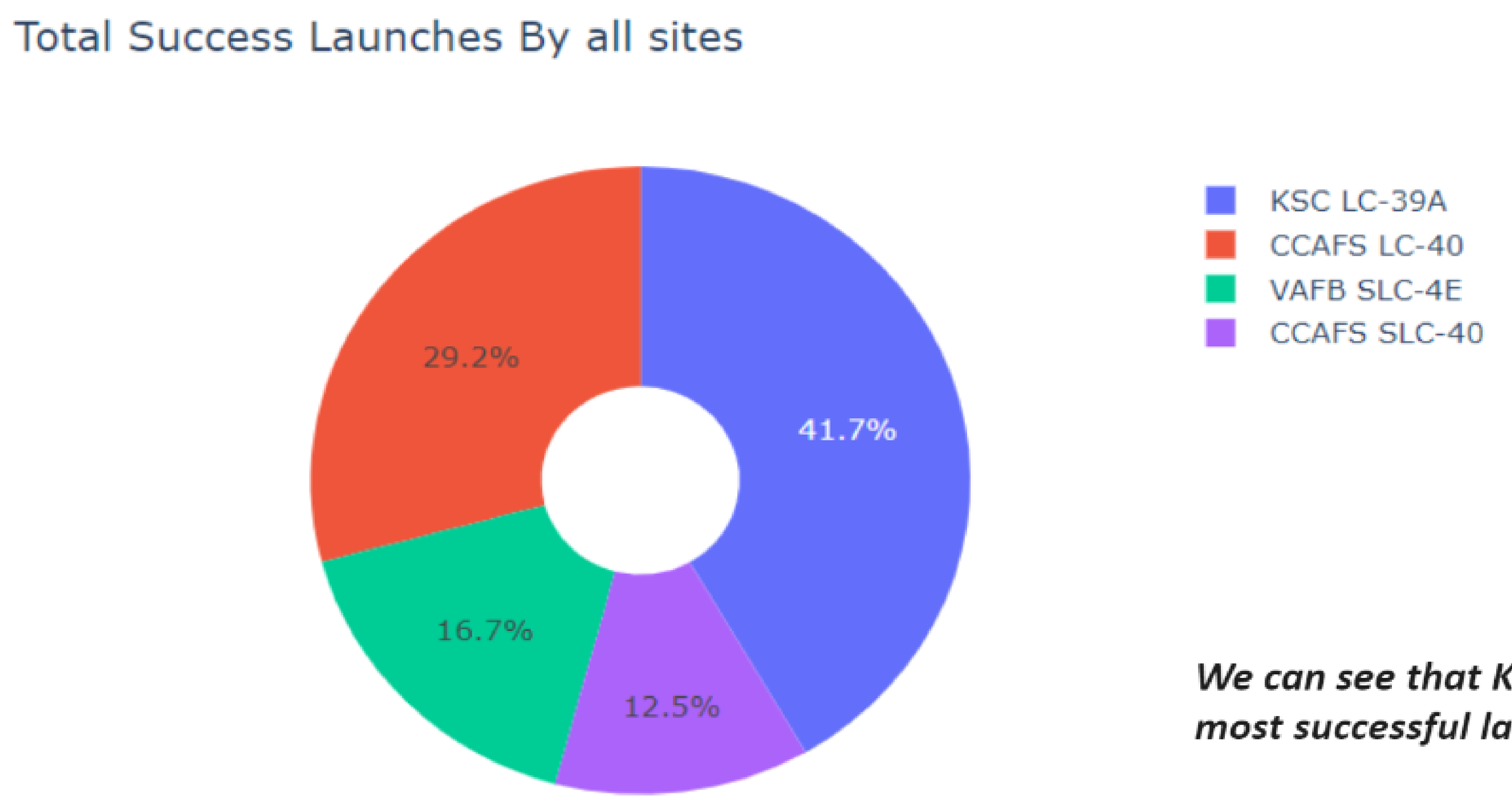
- We notice that the launch sites are far away enough from Public so as to not cause any accidents yet close enough to facilitate commute and resources.
- It is also very close to coastline to facilitate working with the drone ships, etc



Section 5

Dashboard with Plotly

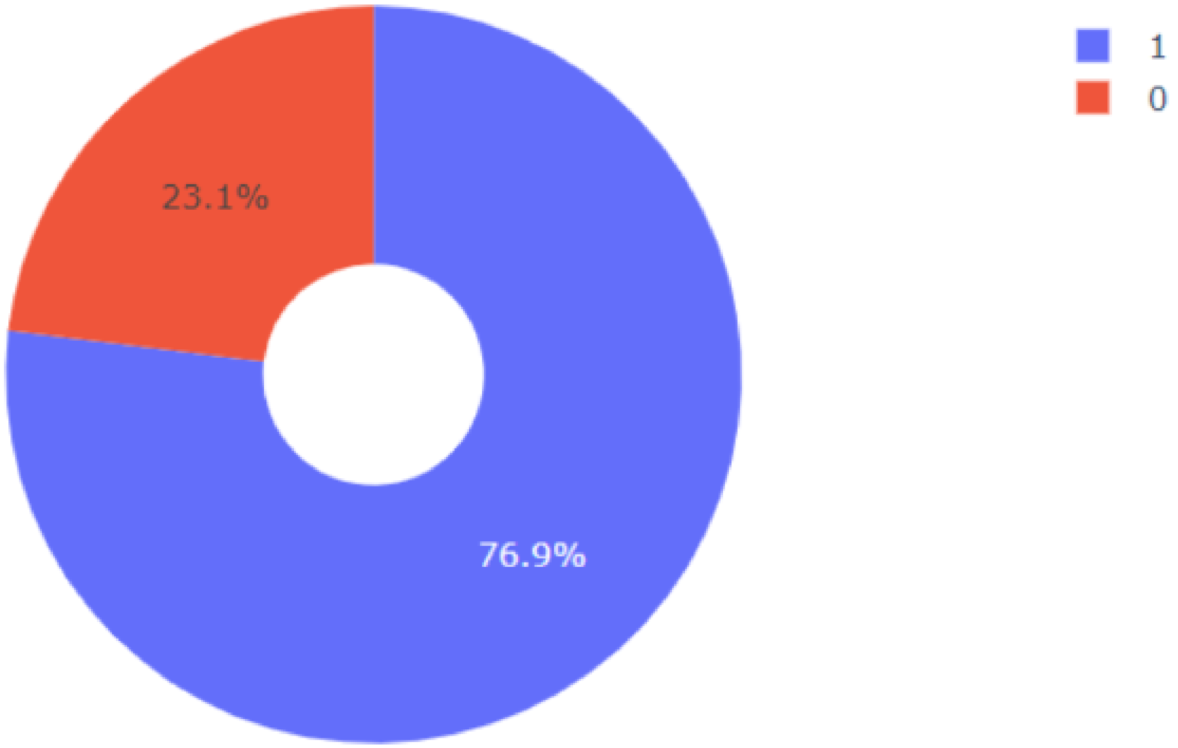
Launch Success by Site



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

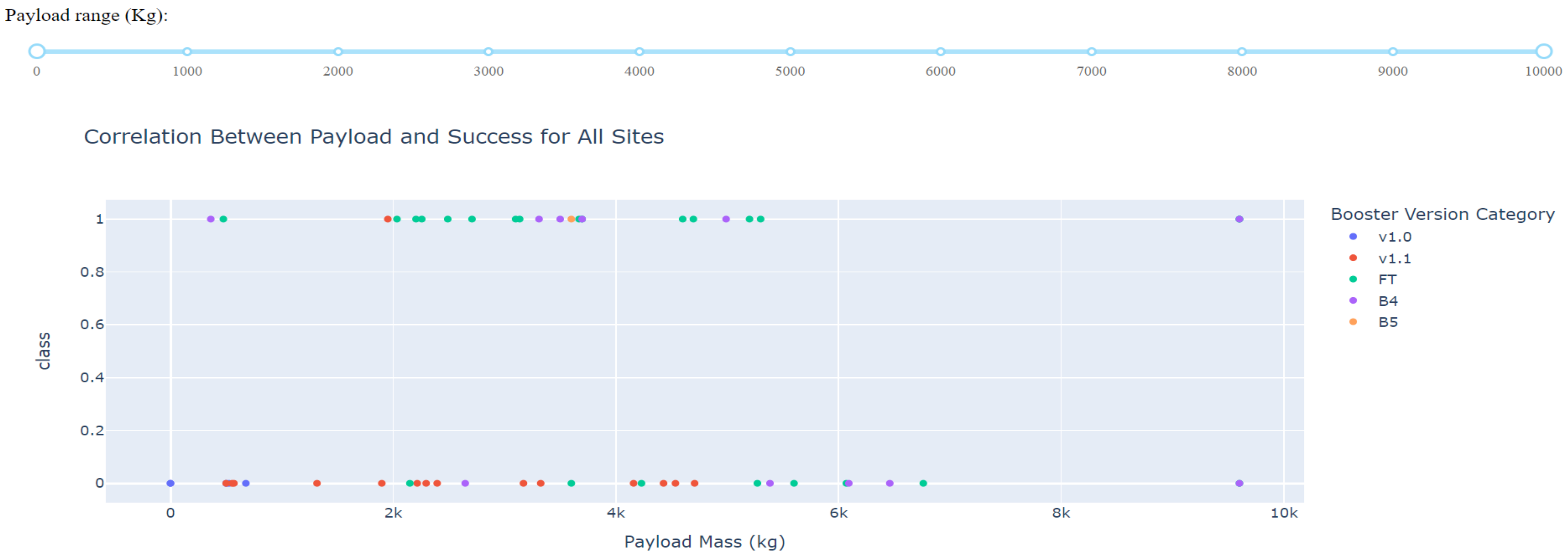
Most Successful Launch Site(KSC LC-29A)

- **KSC LC-39A** has the **highest success rate** amongst launch sites with a **76.9%** success
- It has 10 successful launches and 3 failed launches



Payload Mass and Booster Version

- Payloads between **2,000 kg** and **5,000 kg** have the **highest success rate**
- Booster version FT has the most successes



Section 6

Predictive Analysis (Classification)

Classification Accuracy

- **All** the **models** performed at about the same level and had the **same scores** and **accuracy**. This is likely due to the **small dataset**. The **Decision Tree model slightly outperformed** the rest when looking at .best_score_
- .best_score_ is the average of all cv folds for a single combination of the parameters

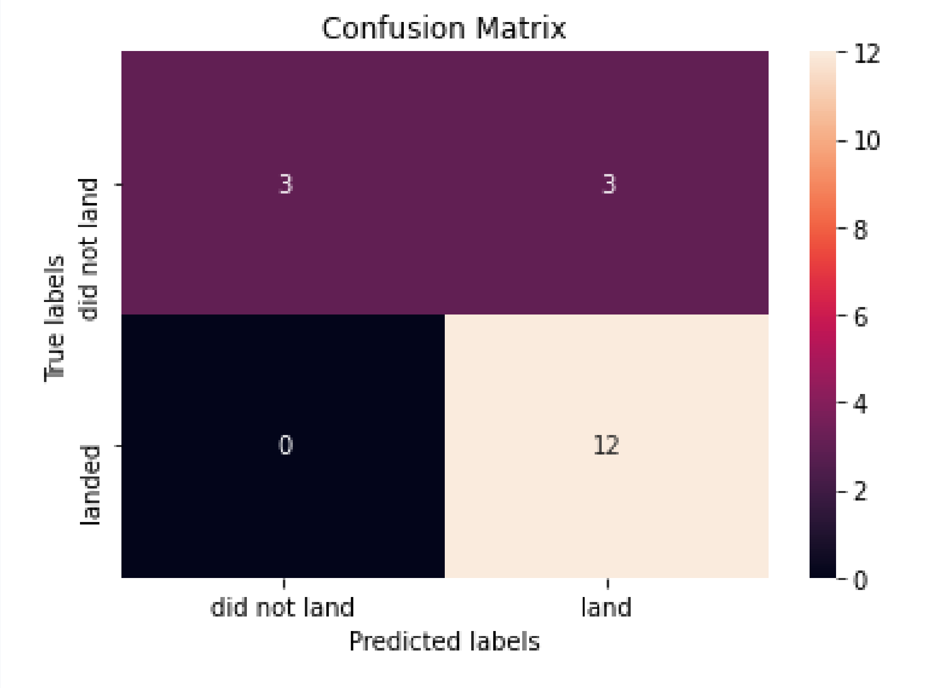
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

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 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

Confusion Matrix

- A confusion matrix of decision tree summarizes the performance of a classification algorithm
- All the confusion matrices were identical
- The fact that there are false positives (Type 1 error) is not good, but it is probably because of the small dataset size.
- Confusion Matrix Outputs:
  - 12 True positive
  - 3 True negative
  - **3 False positive**
  - 0 False Negative
- **Precision** = TP / (TP + FP)
  - 12 / 15 = .80
- **Recall** = TP / (TP + FN)
  - 12 / 12 = 1
- **F1 Score** = 2 * (Precision * Recall) / (Precision + Recall)
  - 2 * (.8 * 1) / (.8 + 1) = .89
- **Accuracy** = (TP + TN) / (TP + TN + FP + FN) = .833



Conclusions

Research

- **Model Performance**: The models performed similarly on the test set with the decision tree model slightly outperforming
- **Equator**: Most of the launch sites are near the equator for an additional natural boost - due to the rotational speed of earth - which helps save the cost of putting in extra fuel and boosters
- **Coast**: All the launch sites are close to the coast
- **Launch Success**: Increases over time
- **KSC LC-39A**: Has the highest success rate among launch sites. Has a 100% success rate for launches less than 5,500 kg
- **Orbits**: ES-L1, GEO, HEO, and SSO have a 100% success rate
- **Payload Mass**: Across all launch sites, the higher the payload mass (kg), the higher the success rate

Thank you!

– Simon