IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Henrikson, Edgard
15/06/2022

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 results
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- Project background and context

  Space X 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 Space X 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 space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Problems you want to find answers

  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API
  - Web Scraping from Wikipedia
- Data wrangling
  - Identify null values and replaced null them with the mean of the column.
  - One-hot encoding was applied to categorical features
- Exploratory data analysis (EDA) using visualization and SQL
- Interactive visual analytics using Folium and Plotly Dash
- Predictive analysis using classification models
  - Using the sklearn library and GridSearchCV to build, tune, evaluate classification models.

Data Collection

- The data was collected using various methods
  - Data collection was done using get request to the SpaceX API.
  - The response content was parsed to Json using the .json() function and turned it into a pandas dataframe with.json_normalize().
  - Missing values were identified and replaced with the mean of the column.
  - In addition, web scraping with BeautifulSoup was performed using Wikipedia for Falcon 9 launch records as a source.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection – SpaceX API

- A GET request was used to get the response object from the SpaceX API and the data was filtered to include only Falcon 9 launches.
- Furthermore, the missing values in the Payload column were replaced with the mean of the column.
- GitHub URL of the completed notebook:

  [Data Collection API]

```python
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdom
# Using the GET request
response = requests.get(static_json_url)
```

```python
# Converting the response into a pandas dataframe
data = pd.json_normalize(response.json())
```

```python
# Filtering dataframe to exclude Falcon 1 launches
falcon = df['BoosterVersion']!='Falcon 1'
data_falcon9 = df[falcon]
```

```python
# Calculate the mean value of PayloadMass column
pay_mean = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].fillna(value=pay_mean, inplace=True)
```

Data Collection - Scraping

- BeautifulSoup was used to extract the Falcon 9 Launch records from Wikipedia.
- This table was parsed and then converted to a pandas dataframe
- GitHub URL of the completed notebook:

  Data Collection with Web scraping

```
# Use requests.get() method with the provided static_url and assign the response to a object
response = requests.get(static_url)
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'html.parser')
```

Extract the column names from the table

```
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

for i in first_launch_table.findAll("th"):
    name = extract_column_from_header(i)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

Created a dataframe from this columns with the data from the HTML tables

Data Wrangling

- Exploratory Data Analysis (EDA) was carried out to find some patterns in the data and determine what would be the label for training supervised models.
- The data contains several Space X launch facilities, each one aims to a dedicated orbit.
- A landing outcome label was created based on the different landing outcomes.
- The success or failure of each landing outcome was stored in the data frame under the new “Class” column that was calculated.

- GitHub URL of the completed notebook:

[Data Wrangling]

EDA with Data Visualization

- Different charts were built to analyze the relationship between:
  - Flight Number and Launch Site
  - Payload and Launch Site
  - Success rate and each orbit type
  - Flight Number and Orbit type
  - Payload and Orbit type

These charts are shown in later sections.

- GitHub URL of the completed notebook:

EDA with Data visualization

EDA with SQL

- SQL queries were performed to get more insights from the data, some of these were:
  - **Names of the unique launch sites in the space mission**
  - **Total payload mass carried by boosters launched by NASA (CRS)**
  - **Average payload mass carried by booster version F9 v1.1**
  - **Date when the first successful landing outcome in ground pad was achieved.**
  - **Total number of successful and failure mission outcomes**
  - **Names of the booster versions which have carried the maximum payload mass.**
  - **Successful landing outcomes between the date 04-06-2010 and 20-03-2017.**
- GitHub URL of the completed notebook:

[SQL Lab]

Build an Interactive Map with Folium

- Al the launch sites have been added to the map along with map objects such as markers, circles, lines to point the success or failure of launches for each site.
- Adding the launch outcomes for each site using the color-labeled marker, allowed to identify which sites have higher success rates.
- The distance between a launch site and its proximities was marked with a line and calculated in order to explore whether it was located near railways, highways, coastlines and cities.
- GitHub URL of the completed notebook:

[Analysis with Folium]

Build a Dashboard with Plotly Dash

- In the Plotly dashboard there are:
  - Pie charts showing the total launches on a site
  - Scatter plot showing the relationship with the Landing Outcome and Payload Mass (Kg) for the different booster versions.

- GitHub URL of the completed .py file:

Plotly App

Predictive Analysis (Classification)

- After loading the dataset and standardizing it Standardize, the data was split into training data and test data
- The best Hyperparameter for SVM, Classification Trees and Logistic Regression were found using GridSearchCV.
- The best performing model was selected.
- GitHub URL of the completed notebook:

Machine learning SpaceX

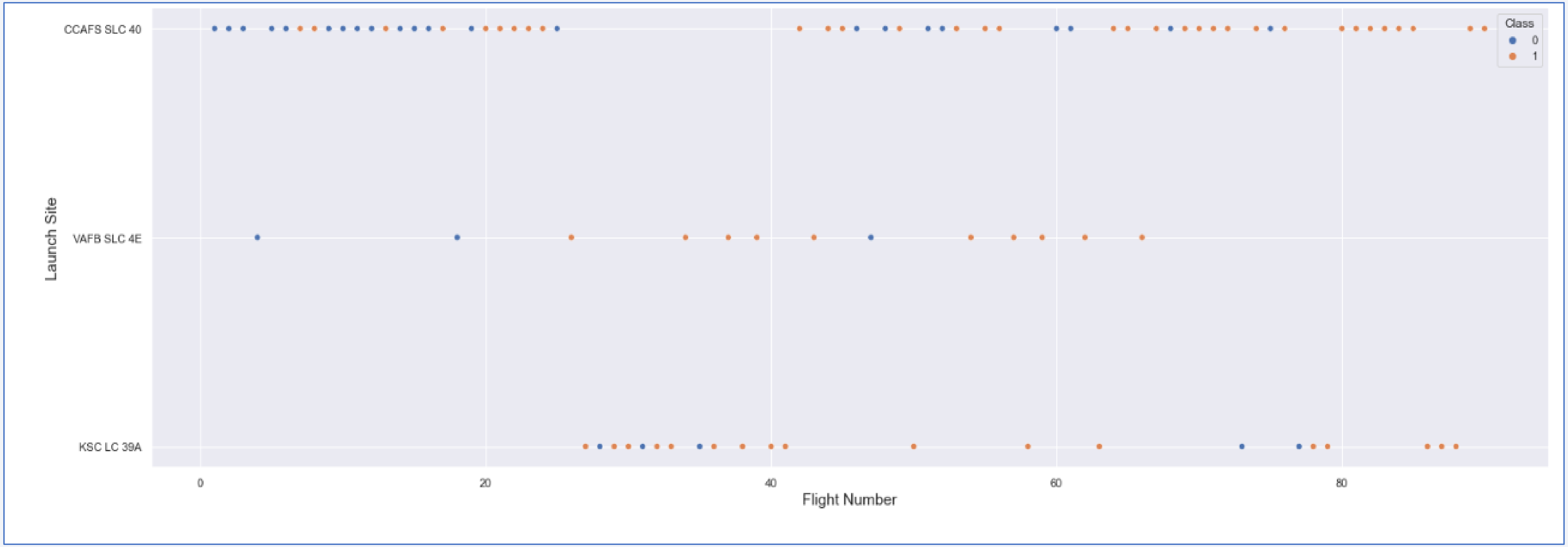
Section 2

Insights drawn from EDA

Results

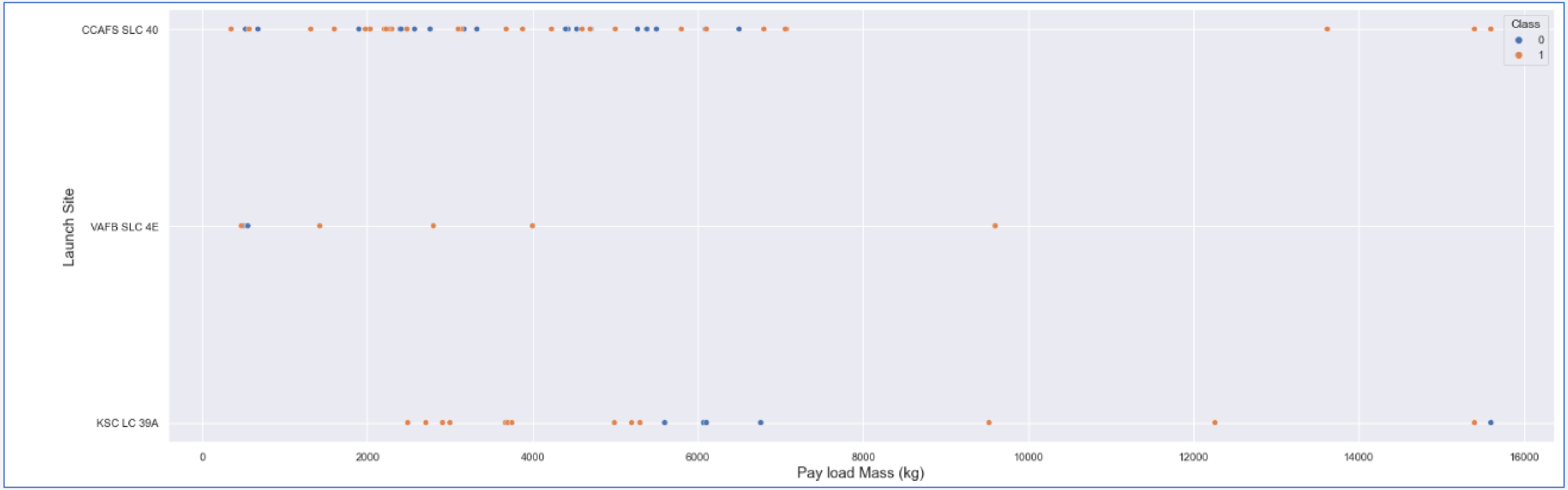
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Flight Number vs. Launch Site

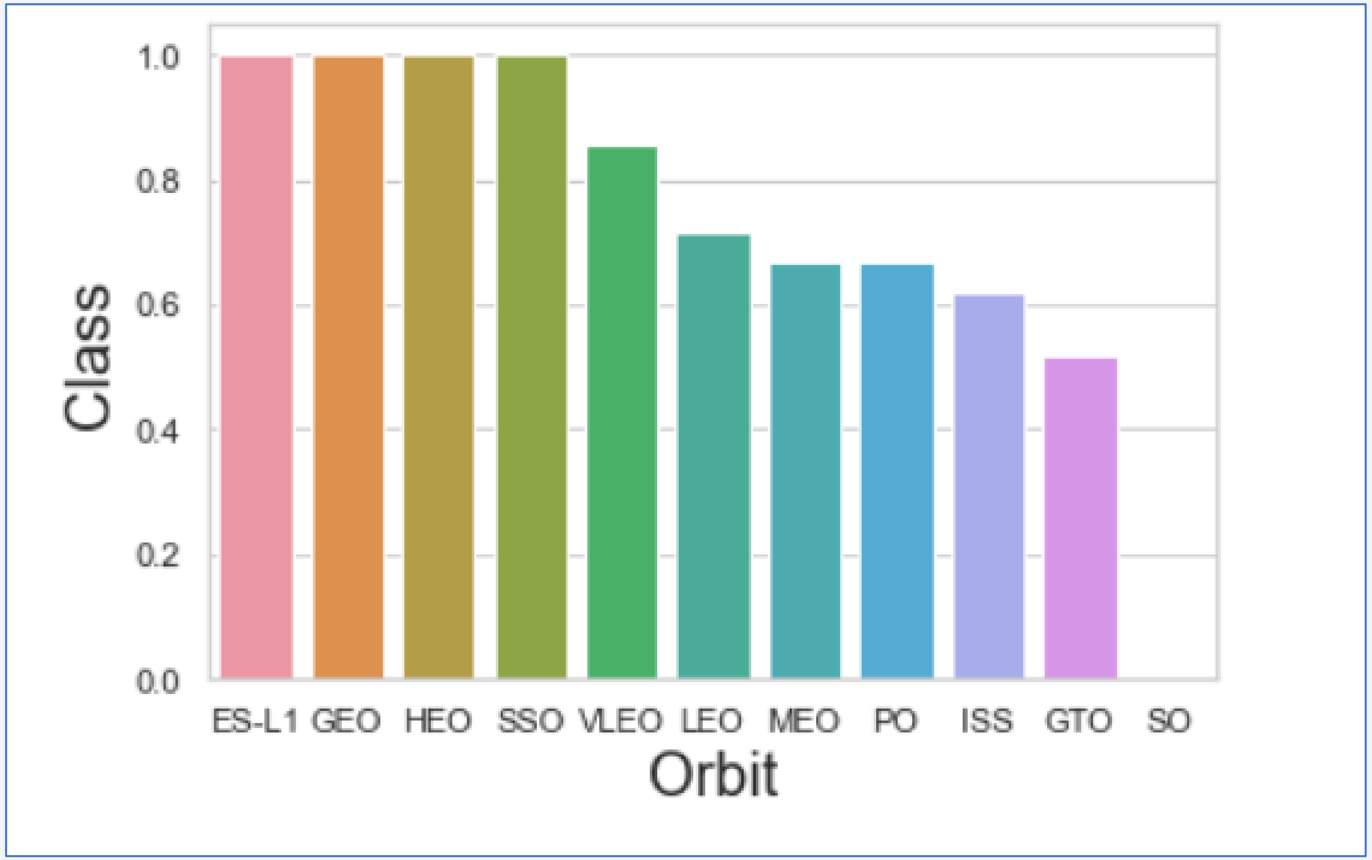


The plot seems to indicate that the success rate at a launch site increases with the number of flights performed at the sites.

Payload vs. Launch Site



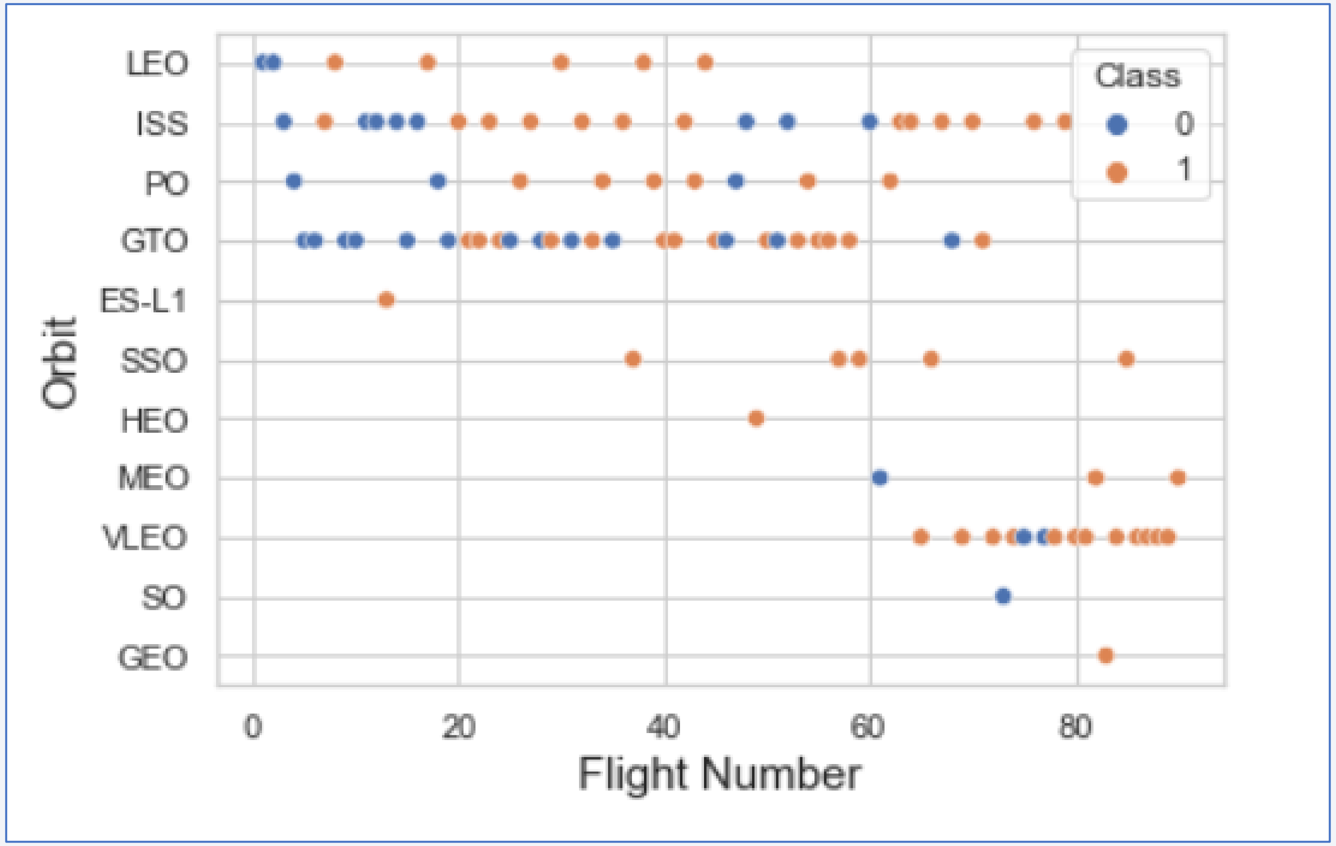
Success Rate vs. Orbit Type



- ES-L1, GEO, HEO, SSO orbits had a success rate of 100%.

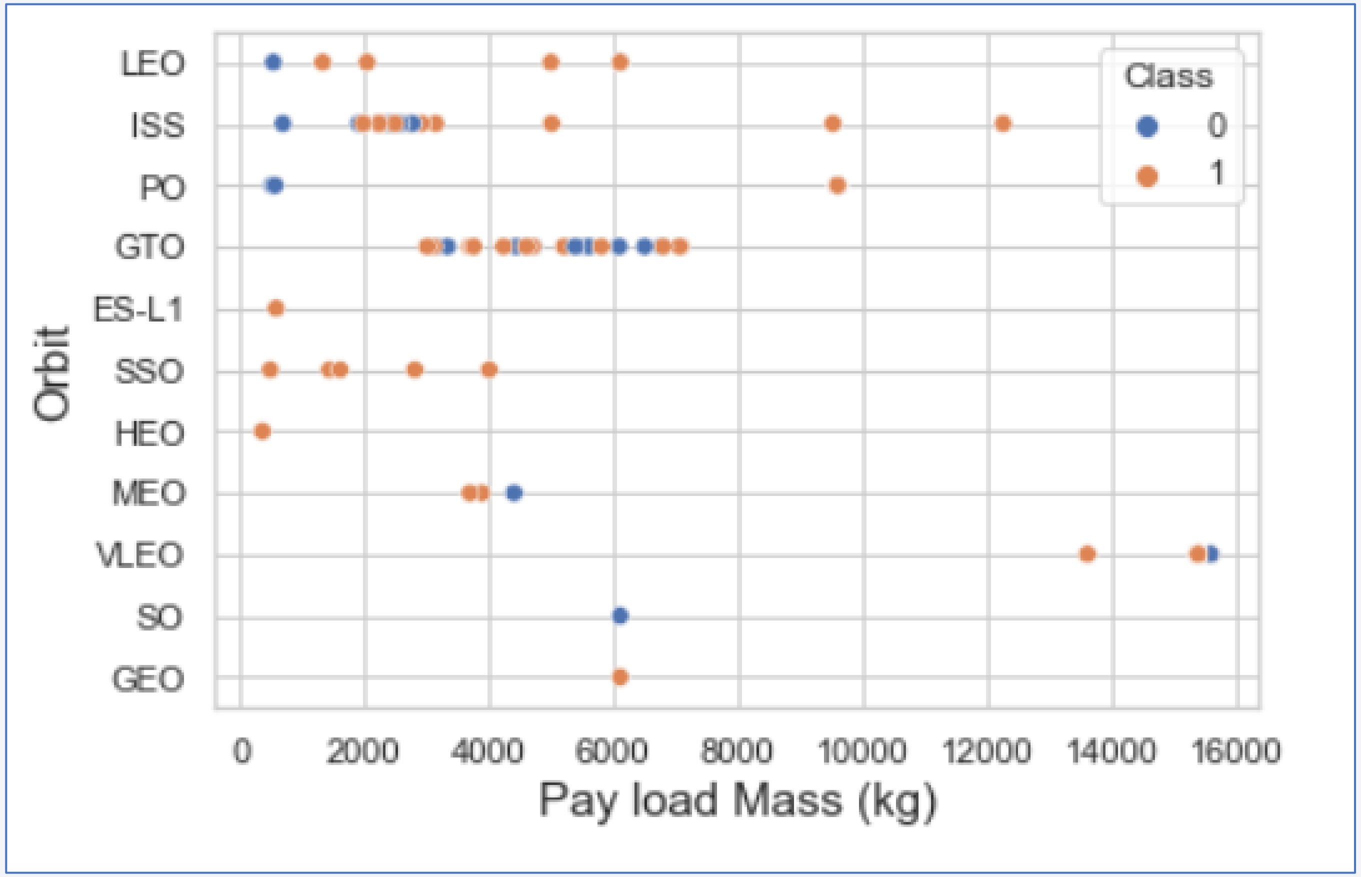
Flight Number vs. Orbit Type

- In the LEO orbit, success appears to be related to the number of flights whereas no relation can be observed in the GTO orbit.



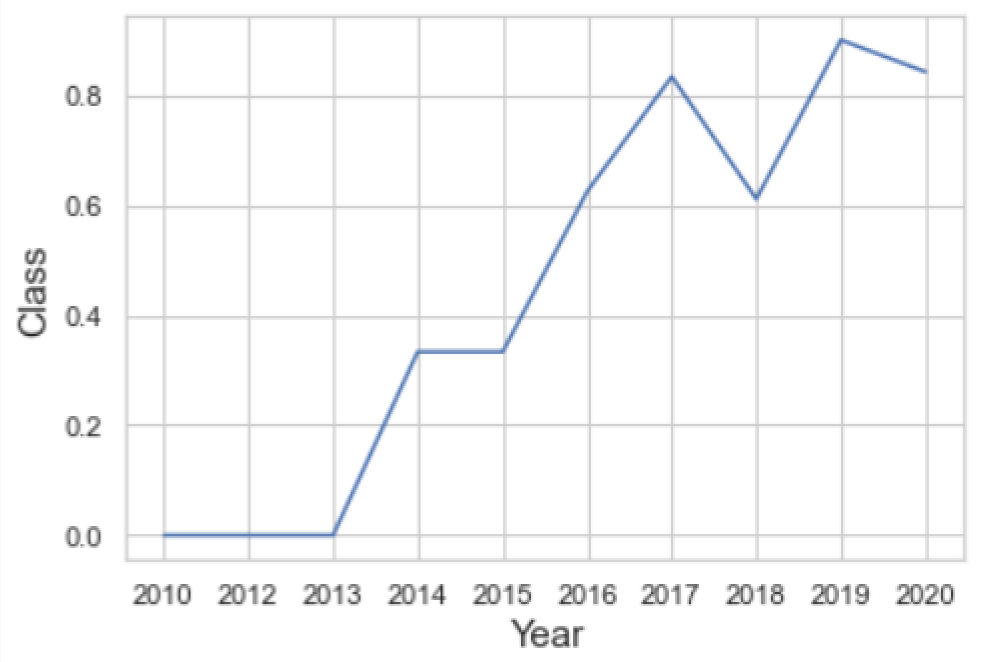
Payload vs. Orbit Type

- Successful landings appear to be higher for the heavier payloads in the orbits LEO, ISS and PO.



Launch Success Yearly Trend

- The success rate has increased since 2013 until 2020.



All Launch Site Names

- GROUP BY or DISTINCT could have been used to achieve the same result.

```
%%sql

SELECT Launch_Site FROM SPACEXTBL
GROUP BY Launch_Site
```

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

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

Launch Site Names Begin with 'CCA'

5 records where launch sites begin with `CCA

```
%%sql

SELECT * FROM SPACEXTBL
WHERE Launch_Site like 'CCA%'
LIMIT 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Calculated the total payload carried by boosters from NASA (CRS) using SUM.

```
%%sql

SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass (Kg)" FROM SPACEXTBL
WHERE Customer like 'NASA (CRS)'
```

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

Total Payload Mass (Kg)
45596

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 v1.1 using AVG

```
%%sql

SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL
WHERE Booster_Version like 'F9 v1.1'
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Date of the first successful landing outcome on ground pad using MIN

```
%%sql

SELECT MIN(Date) FROM SPACEXTBL
WHERE "Landing _Outcome" like 'Success (ground pad)'
```

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

MIN(Date)
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

Using AND condition to determine successful landing with payload mass greater than 4000 but less than 6000

```
%%sql

SELECT Booster_Version FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000 AND "Landing _Outcome"  like 'Success (drone ship)'
```

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

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

Total Number of Successful and Failure Mission Outcomes

- Total number of successful and failure mission outcomes using COUNT

```
%%sql

SELECT COUNT(*) FROM SPACEXTBL
WHERE Mission_Outcome like '%Success%'
```

 * sqlite:///my_data1.db
Done.

COUNT(*)
100

```
%%sql

SELECT COUNT(*) FROM SPACEXTBL
WHERE Mission_Outcome like '%Failure%'
```

 * sqlite:///my_data1.db
Done.

COUNT(*)
1

Boosters Carried Maximum Payload

- Using a Sub query to calculate the maximum payload and then listing the Booster versions.

```
%%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

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

```
%%sql

SELECT substr(Date, 4, 2) AS Month,"Landing _Outcome",Booster_Version,Launch_Site FROM SPACEXTBL
WHERE "Landing _Outcome"  like 'Failure (drone ship)' AND substr(Date,7,4)='2015'
```

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

Month	Landing _Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- 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.

```
%%sql
SELECT "Landing _Outcome", COUNT("Landing _Outcome")
        FROM SPACEXTBL
        WHERE DATE BETWEEN '04-06-2010' AND '20-03-2017'
        GROUP BY "Landing _Outcome"
        ORDER BY COUNT("Landing _Outcome") DESC
```

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

Landing _Outcome	COUNT("Landing _Outcome")
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Failure	3
Controlled (ocean)	3
Failure (parachute)	2
No attempt	1

Section 3

Launch Sites Proximities Analysis

Global map with all launch sites

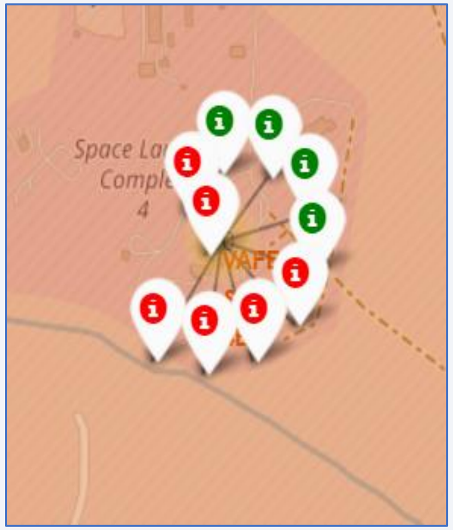
All the SpaceX launch sites are in the USA, the same as the NASA JSC.

Markers of success and failures in launch sites

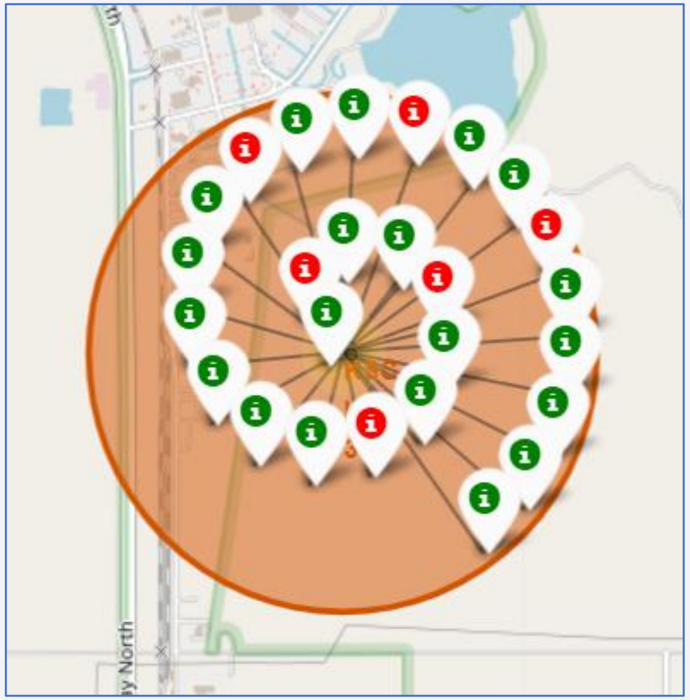
Green markers show show succesful landing outcomes, whereas Red markers indicate failures

West coast launch site

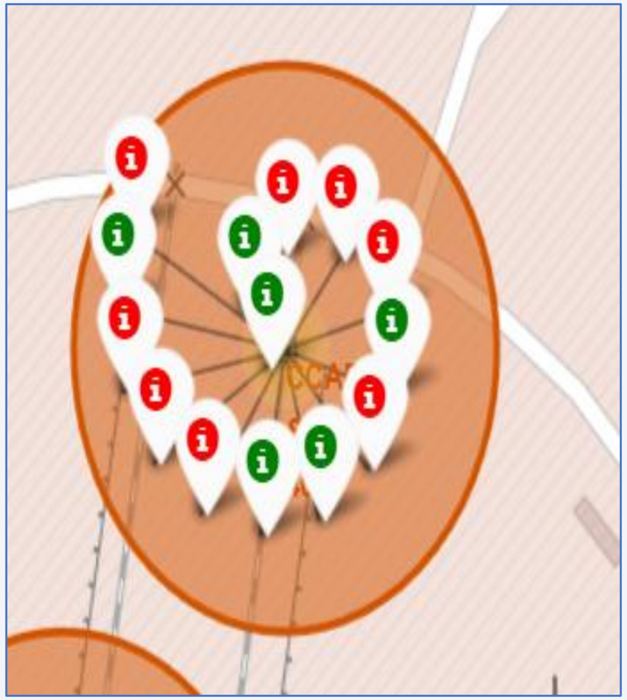
East coast launch sites



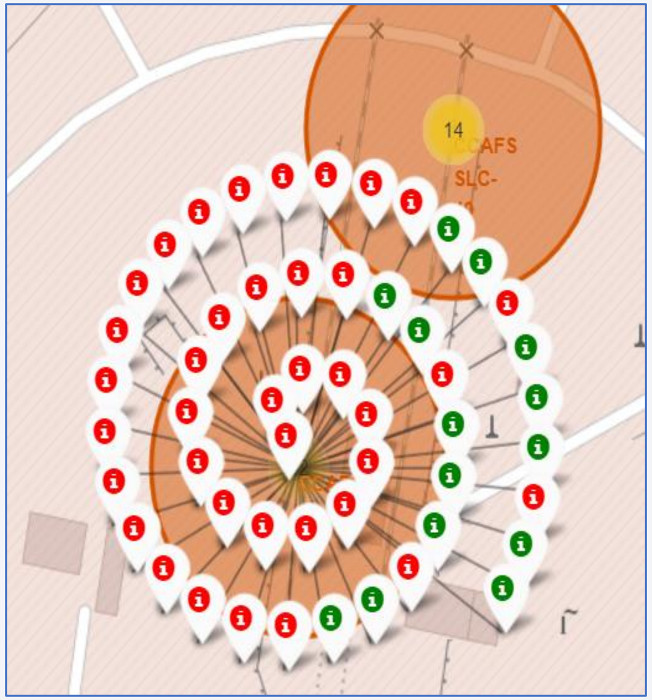
VAFB SLC-4E



KSC LC-39A



CCAFS SLC-40

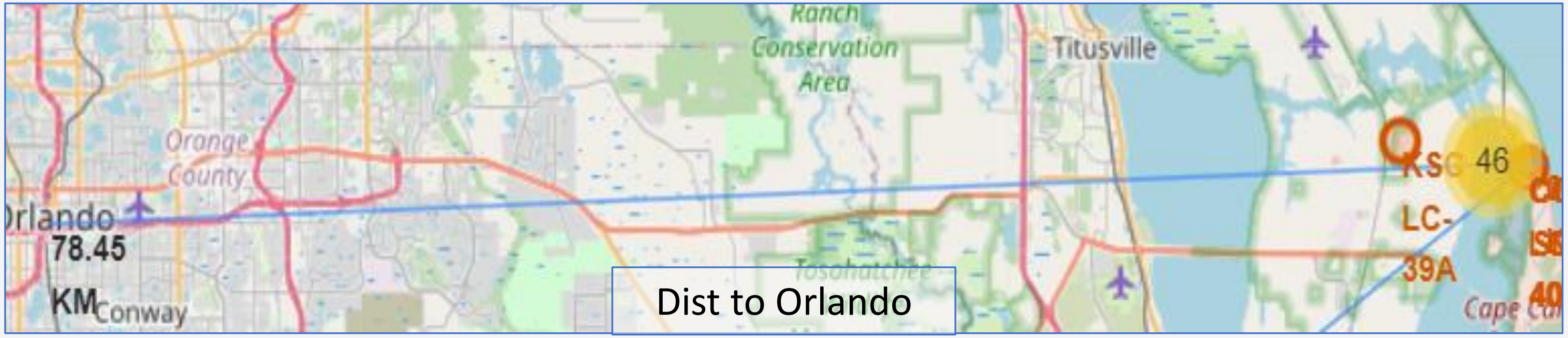


CCAFS LC-40

Distances between a west coast launch sites to its proximities

Dist to highway

Dist to coastline



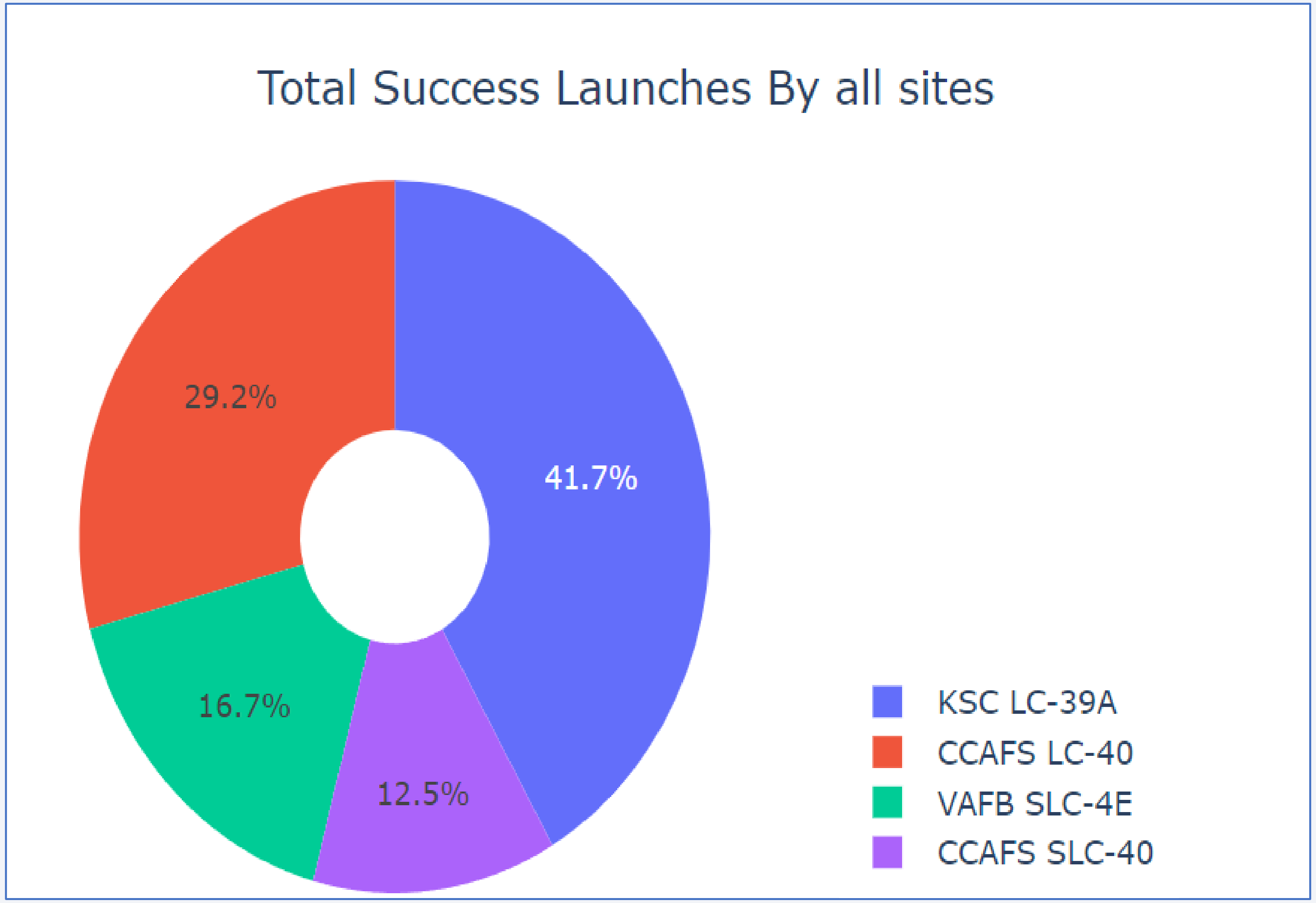
Dist to Orlando

- The launch sites can be found in proximity to coastlines.
- However, they are located within a larger distance from highways or cities.

Section 4

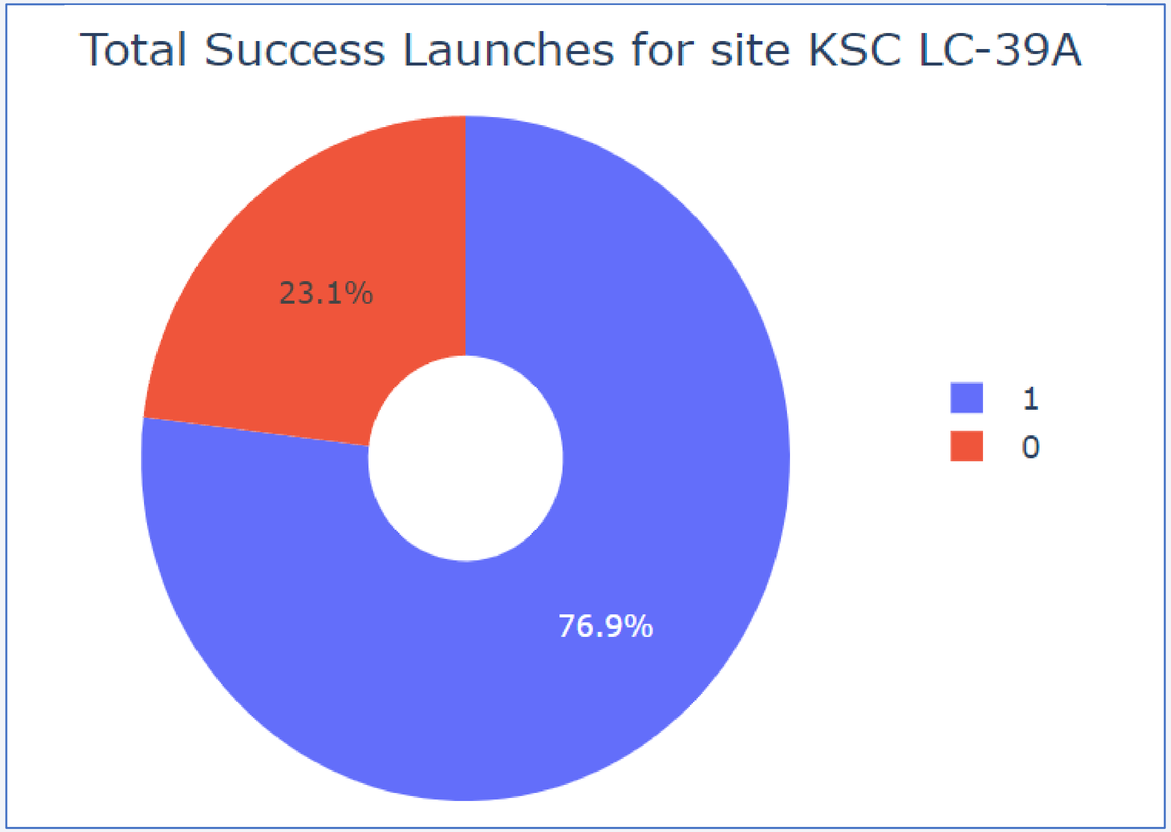
Build a Dashboard with Plotly Dash

Launch success count for all sites



Launch site KSC LC-39A has had the higher success rate

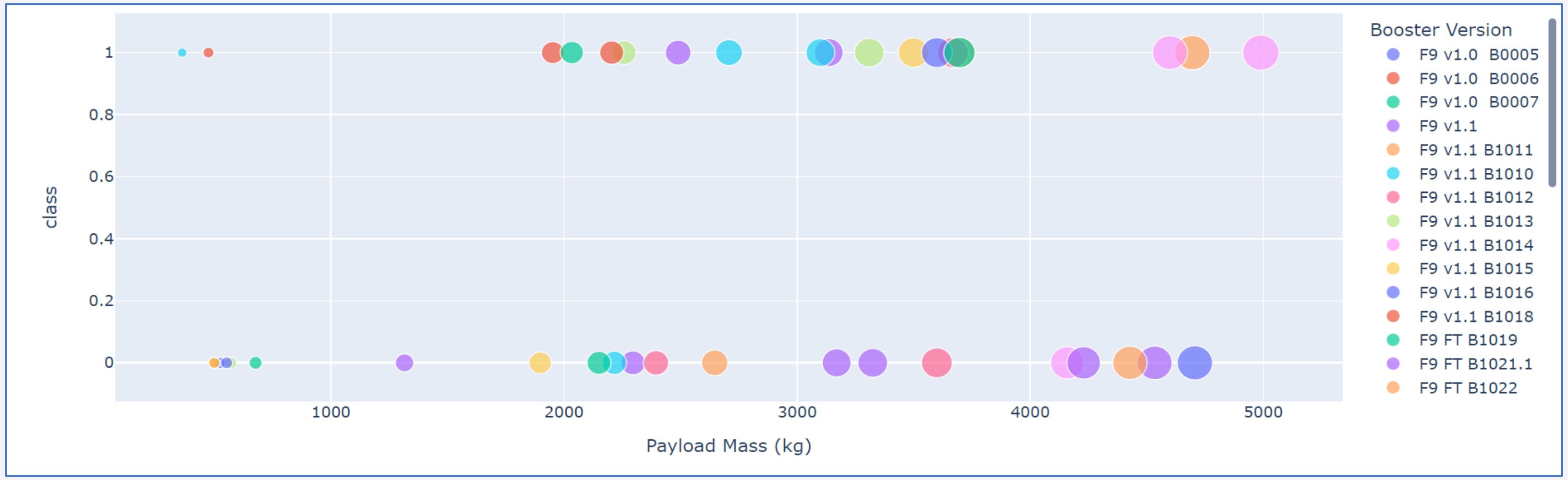
Launch site KSC LC-39A success ratio



Launch site KSC LC-39A has a success rate of 76.9%

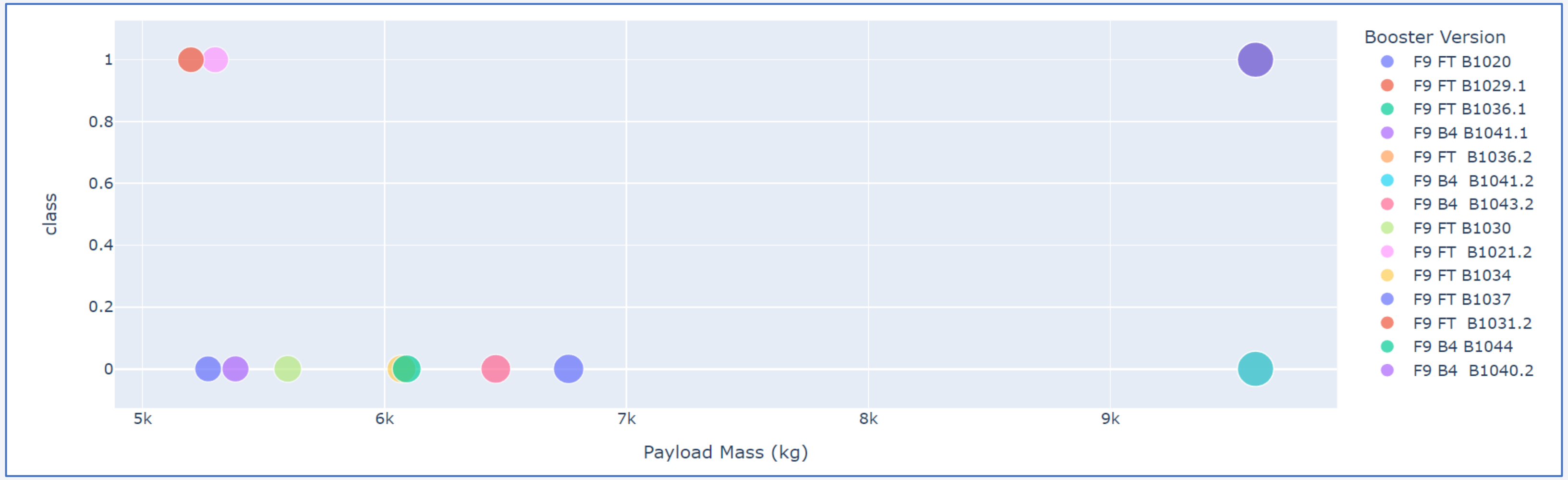
Payload vs. Launch Outcome scatter plot for all sites

Payload range from 0kg -5000kg



The success rate is higher for lower weight payloads

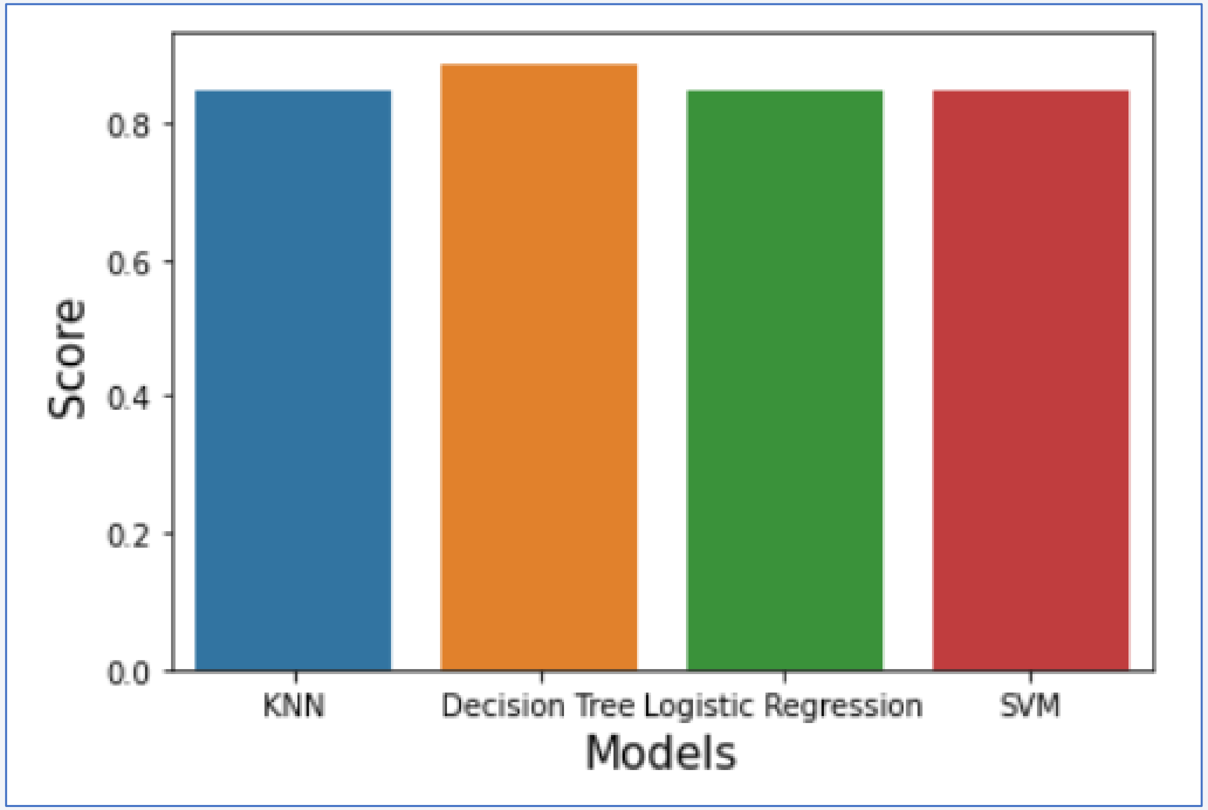
Payload range from 5000kg -10000kg



Section 5

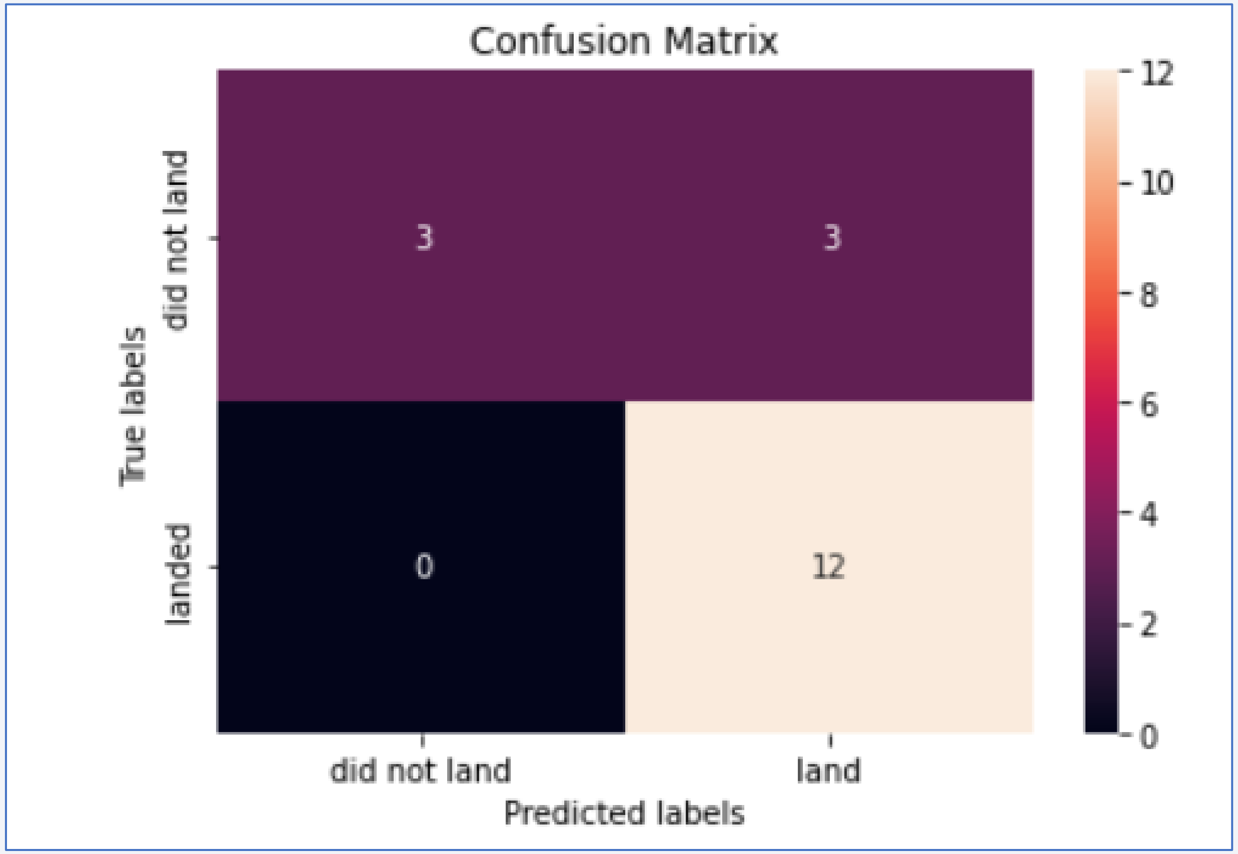
Predictive Analysis (Classification)

Classification Accuracy



The decision tree model is the one with the higher accuracy

Confusion Matrix



The confusion matrix of the decision tree shows that the biggest issue the model has is the false positives in predictions, i.e., landings than would be unsuccessful would be predicted as successful.

Conclusions

From the analysis we can conclude that:

- The success rate of landings from a launch site increase with the number of flights.
- The success rate increased from 2013 until 2020.
- There are orbits that have a higher success rate:
  - ES-L1, GEO, HEO, SSO orbits had a success rate of 100%.
- Launch site KSC LC-39A had the most successful landings.
- The Decision tree is the best model for predicting the landing outcomes.

Thank you!