Winning Space Race with Data Science

Krunal Patel
12-08-2023

Executive Summary

Summary of Methodologies :

This Data Science project attempts to bracket the factors for a successful rocket landing and how it affects the cost. To make this determination, the following methodologies where used:

- **Collecting** data using SpaceX REST API and web scraping.
- **Data Wrangling** to create success/fail outcome variable.
- **Data Exploration** with data visualization techniques, considering the following factors: payload, launch site, flight number and yearly trend.
- **Analyze** the data with SQL, exploring the following statistics: total payload, payload range for successful launches and failed outcomes.
- **Explore** accessibility to geographical markers and launch site hit ratio.
- **Visualize** the launch sites with the most success and successful payload ranges.
- **Machine Learning Models** such as logistic regression, support vector machine. (SVM), decision tree and K-nearest neighbor (KNN) to estimate landing outcomes.

Summary of all Results :

Exploratory Data Analysis:

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

Visualization/Analytics:

- Most launch sites are all are near to the equatorial coasts.

Predictive Analytics:

- All Machine Learning algorithms estimated same accuracy.

Introduction

Context and Background :

SpaceX 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 SpaceX 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 Addressed :

- How the factors like payload mass, launch site, number of flights, and orbits contribute the first-stage landing success?
- Rate of successful landings over time.
- What is the best predictive model for successful landing?

Section 1

Methodology

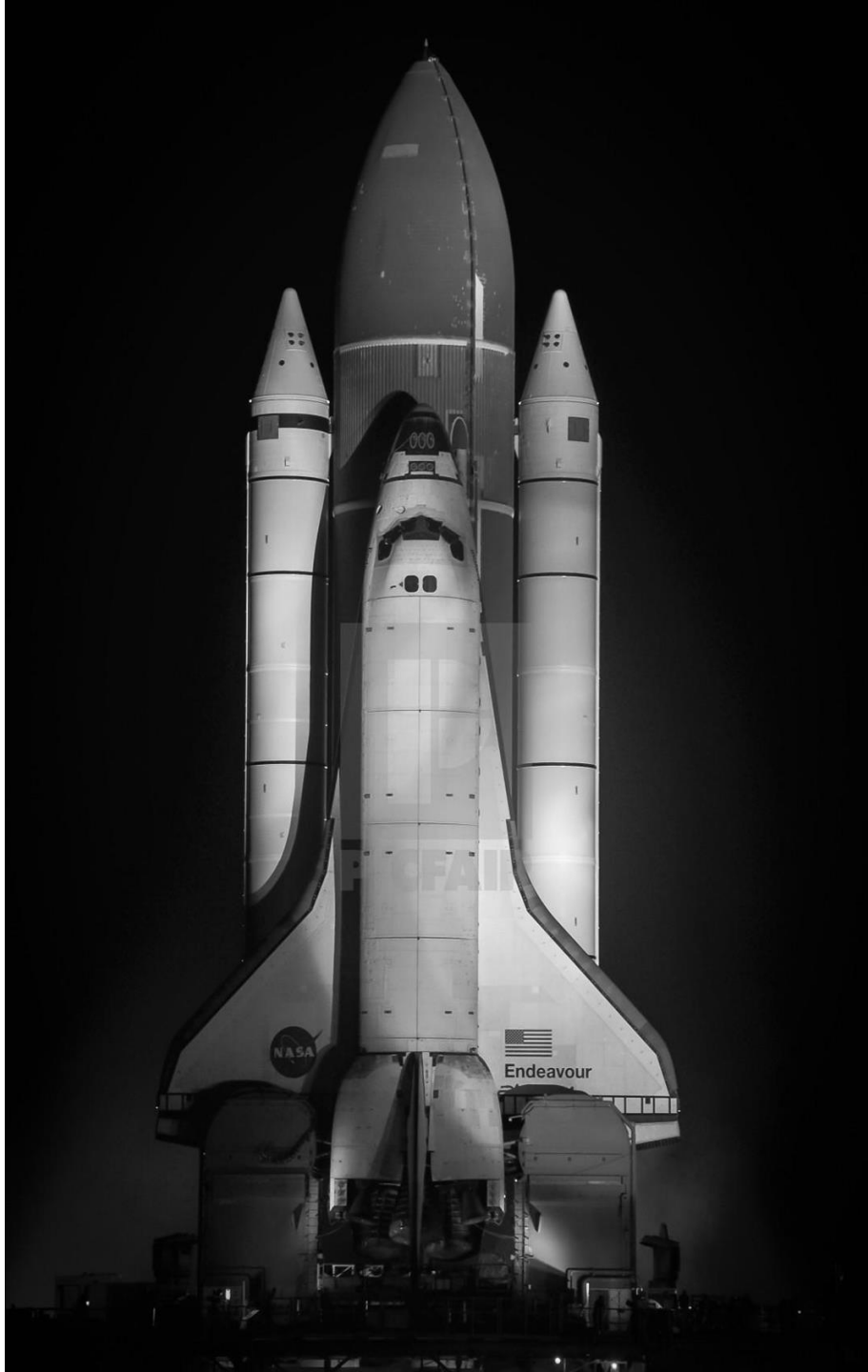
Methodology

Executive Summary :

- **Data Collection Methodology:** Using SpaceX REST API and web scraping techniques.
- **Data Wrangling:** Filtering the data, handling missing values and applying one hot encoding – to prepare the data for analysis and modeling.
- **Exploratory Data Analysis:** Through SQL and Data visualization techniques.
- **Interactive Visual Analytics:** Dashboard and Visualization of the data using Plotly Dash and Folium.
- **Predictive Analysis using Classification Models:** Such as logistic regression, support vector machine (SVM), decision tree and K-nearest neighbor (KNN) to estimate landing outcomes.

Data Collection

- **The data was collected using various methods :**
  - Data collection was done using get request to the **SpaceX API**.
  - Next, we decoded the response content as a Json using **.json() function** call and turn it into a pandas dataframe using **.json_normalize()**.
  - We then cleaned the data, checked for **missing values and fill in missing values** where necessary.
  - In addition, we performed **web scraping from Wikipedia** for Falcon 9 launch records with BeautifulSoup.
  - 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 through 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.
- **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
- **Link to the GitHub Notebook :** https://github.com/Krunalscorp/IBM-SpaceX_Capstone_Project/blob/main/Spacex-data-collection-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 through Web Scraping

- **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 and create dataframe from the dictionar
- **Link to the GitHub Notebook:** https://github.com/Krunalscorp/IBM-SpaceX_Capstone_Project/blob/main/Webscraping.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

Data Transformation

- **Perform EDA** and determine data labels
- **Calculate** the number of launches for each site and occurrence of orbit and occurrence of mission
- **Create binary** landing outcome
- column (dependent variable)
- **Export data** to csv file

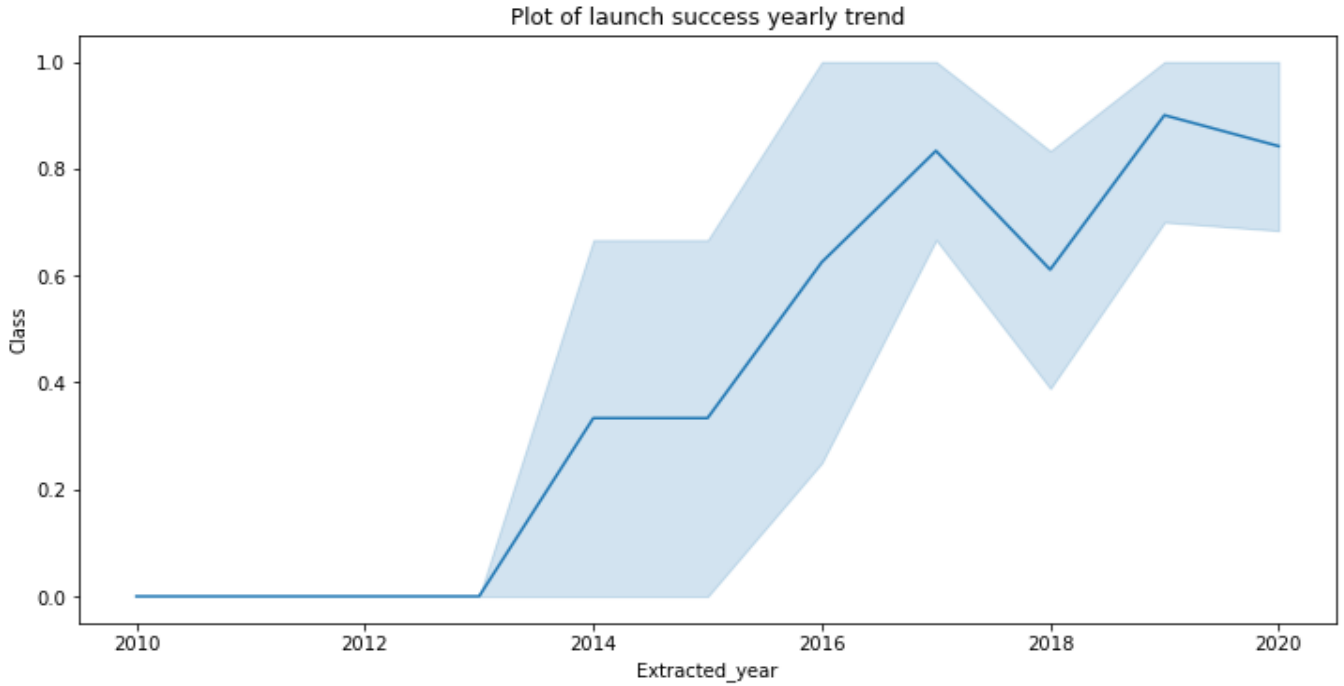
- **Link to the GitHub Notebook:** https://github.com/Krunalscorp/IBM-SpaceX_Capstone_Project/blob/main/Spacex-Data_wrangling_jupyterlite.jupyterlite.ipynb

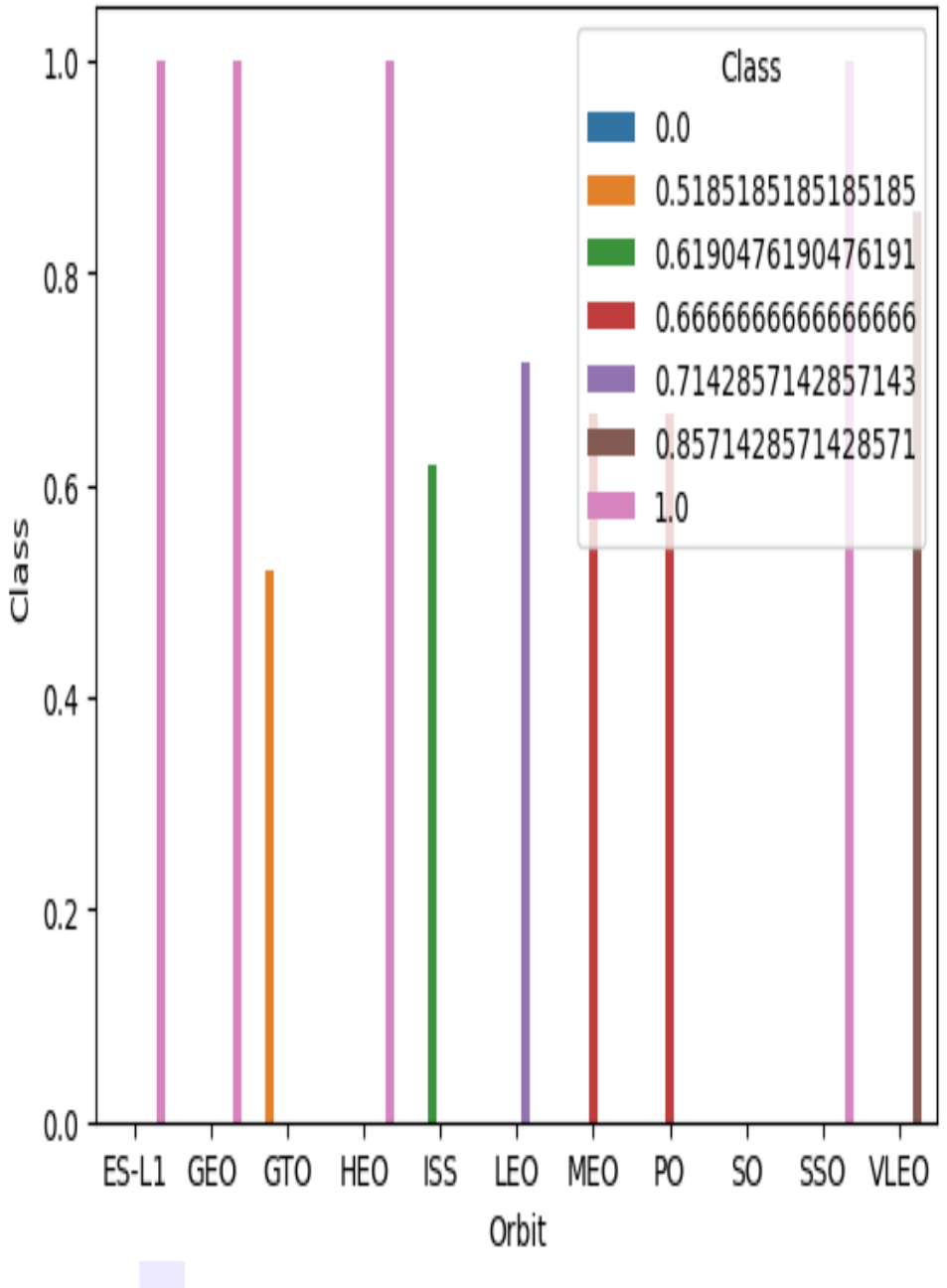
- **True Ocean:** successful landing at a particular part of ocean
- **False Ocean:** it is an unsuccessful landing in a particular part of the ocean
- **True RTLS:** successful landing at ground pad
- **False RTLS:** unsuccessful landing on a ground pad
- **True ASDS:** successful landing on a drone ship
- **False ASDS:** unsuccessful landing on a drone ship
- **Outcomes transformed** into dummy variable: 1 for a successful landing and 0 for an unsuccessful landing

EDA with Visualization

Analysis

- **Find Relationships** between variables to establish a relationship
- **Bar charts show comparisons** among discrete categories, relationships among the categories and a measured value.
- **Link to the GitHub Notebook:** https://github.com/Krunalscorp/IBM-SpaceX_Capstone_Project/blob/main/EDA-dataviz.ipynb.jupyterlite.ipynb





EDA with SQL

We loaded the SpaceX dataset into a PostgreSQL database without leaving the jupyter notebook and wrote queries to find out following instances.

- 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 v1.1.
- Date of first successful landing on ground pad
- Names of boosters which had success landing on drone ship and have
- payload mass greater than 4,000 but less than 6,000
- Total number of successful and failed missions
- Names of booster versions which have 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 2010-06-04 and 2017-03-20 (desc)

Link to the GitHub Notebook: https://github.com/Krunalscorp/IBM-SpaceX_Capstone_Project/blob/main/EDA-sql-coursera_sqllite.ipynb

Interactive Map with Folium

- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.

The link to the notebook is :
https://github.com/Krunalscorp/IBM-SpaceX_Capstone_Project/blob/main/Launch_site_location_Analysis_with_Folium.jupyterlite.ipynb
Launching Outcomes color coding:

To show which launch sites have high success rates Added **colored markers** of **successful** (**green**) and **unsuccessful** (**red**) **launches**

We calculated the distances between a launch site to its proximities. We answered some question for instance:

- Are launch sites near railways, highways and coastlines.
- Do launch sites keep certain distance away from cities.

Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- Dropdown List with Launch Sites
- Pie Chart Showing Successful Launches
- Slider of Payload Mass Range

Link to the GitHub Notebook:
https://github.com/Krunalscorp/IBM-SpaceX_Capstone_Project/blob/main/spacex_dash_app.py.py

Predictive Analytics

- 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 the **best performing classification model. Identify** the best model using Jaccard_Score, F1_Score and Accuracy

The link to the GitHub Notebook:
https://github.com/Krunalscorp/IBM-SpaceX_Capstone_Project/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.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

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

Predictive Analytics: Decision Tree model is the best predictive model for the dataset

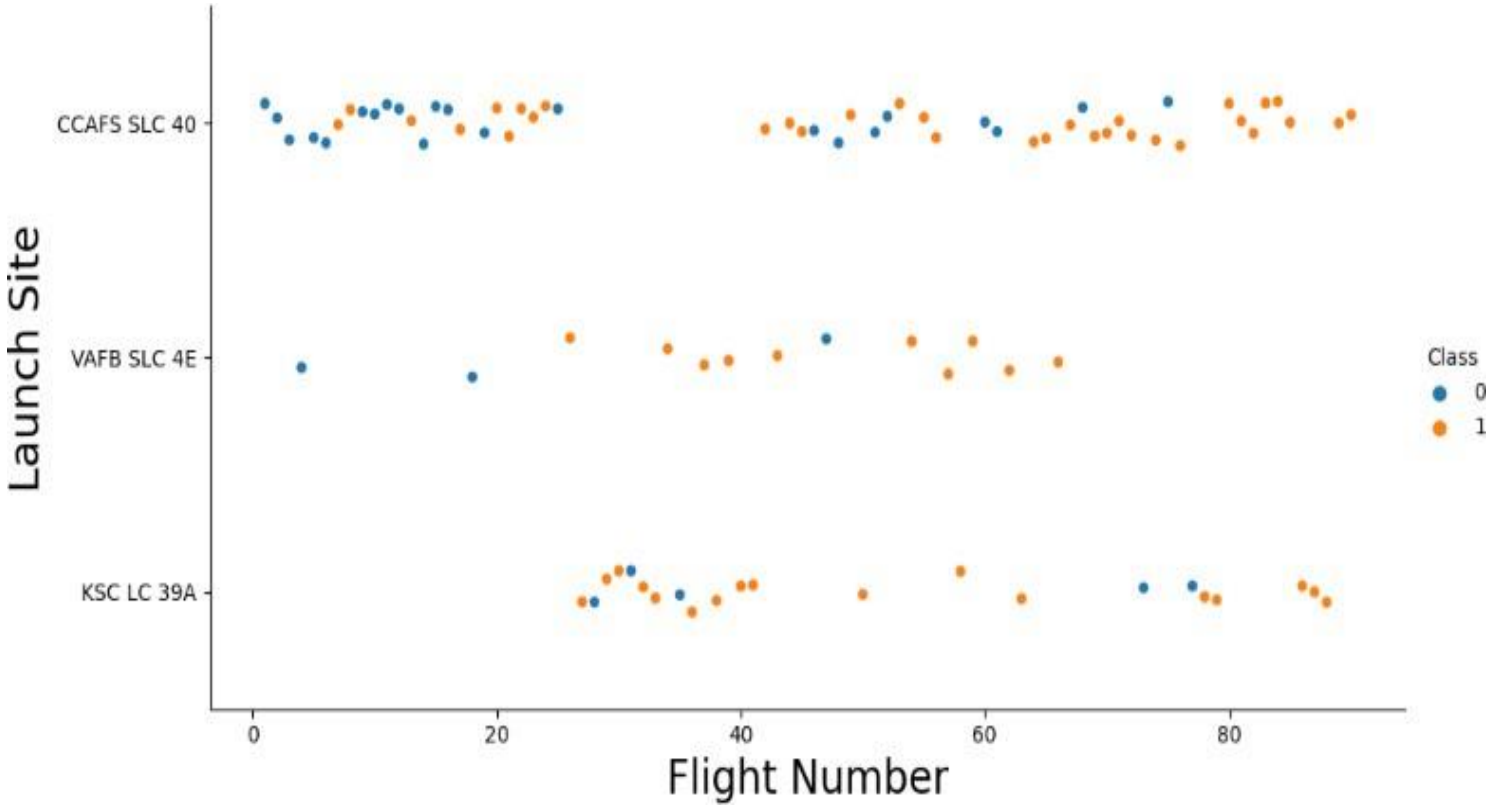
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

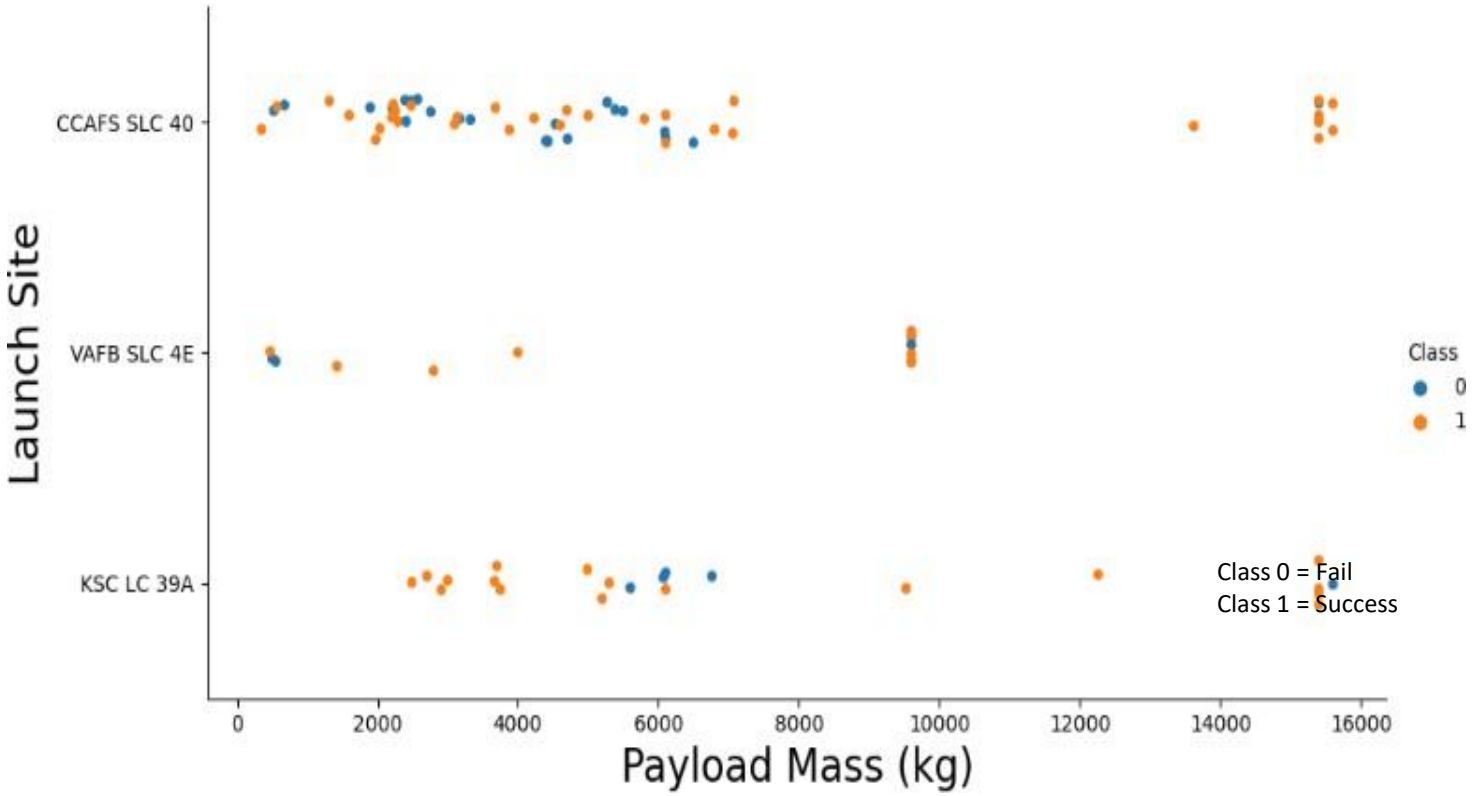
Earlier flights had a lower success rate (blue = fail) Later flights had a higher success rate (orange = success)

Around half of launches were from CCAFS SLC 40. VAFB SLC 4E and KSC LC 39A sites have higher success rates.



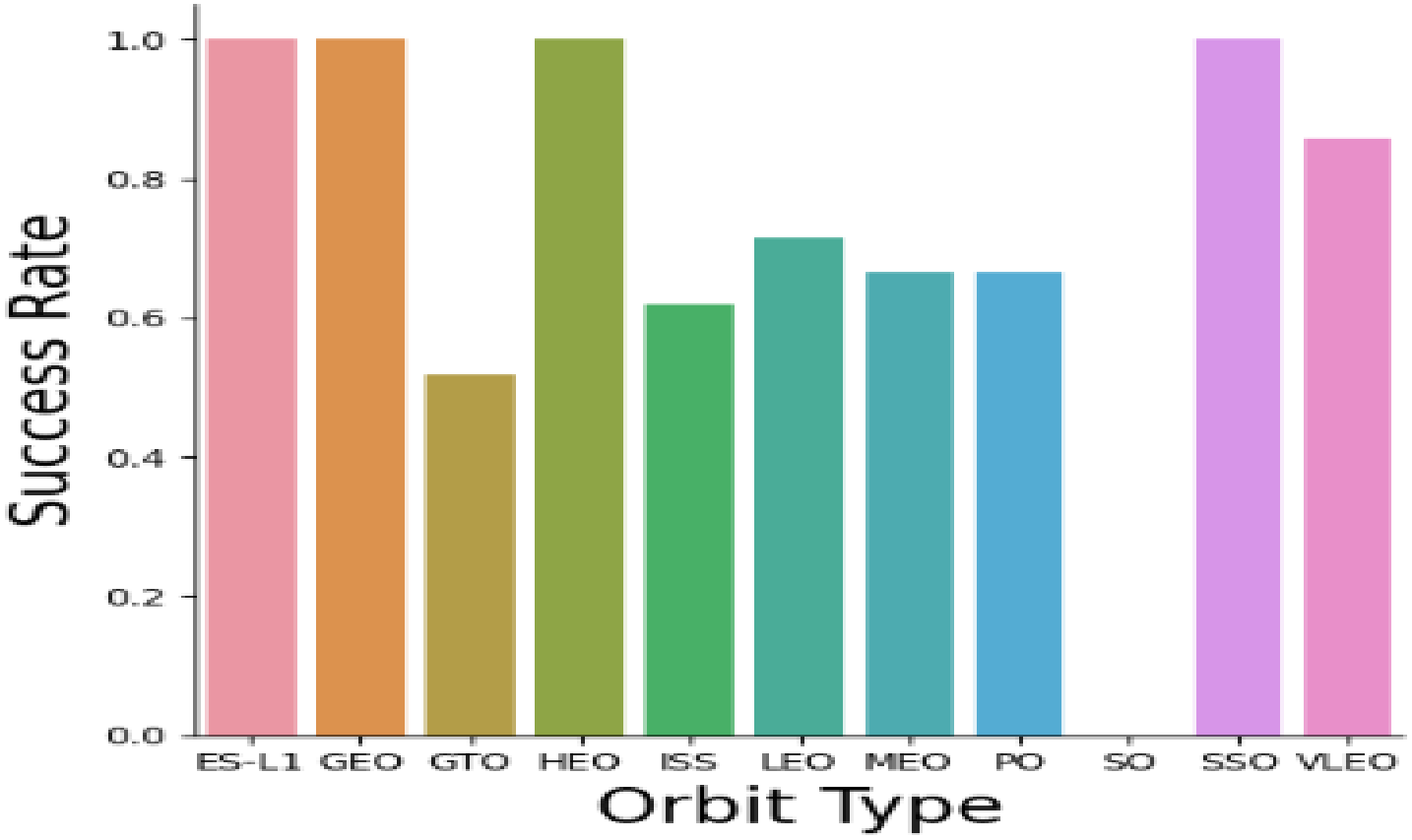
Payload vs. Launch Site

Typically, the **higher** the **payload mass** (kg), the **higher** the **success rate**
Most launches 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



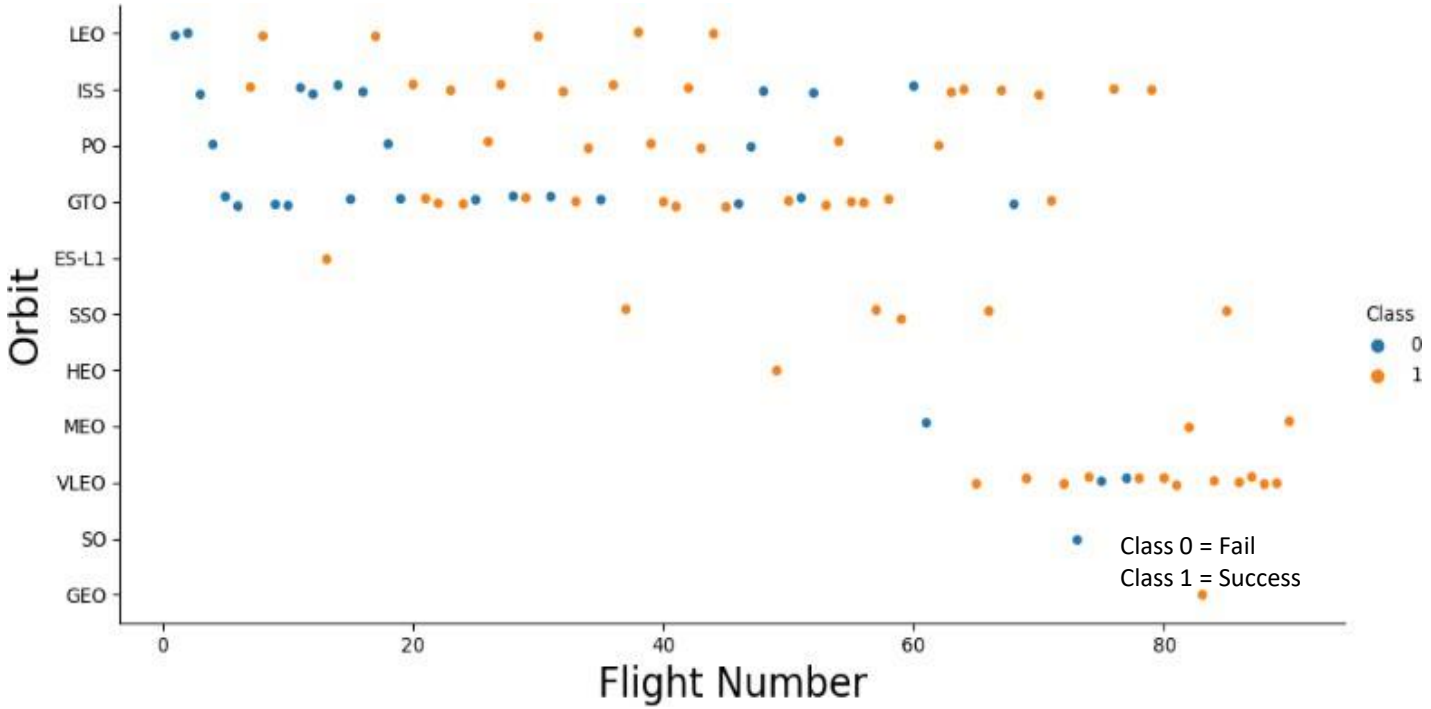
Success Rate by Orbit

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



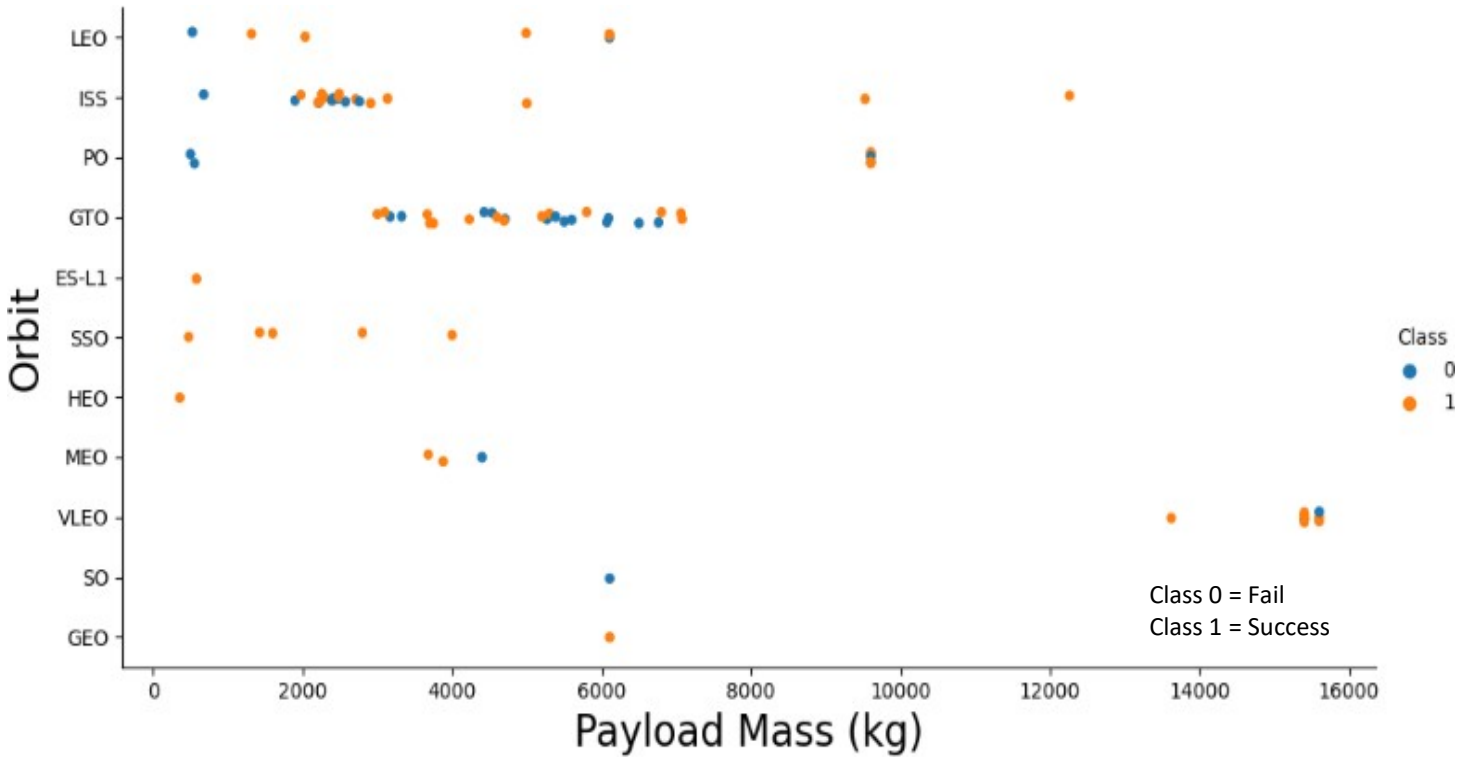
Flight Number vs. Orbit

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



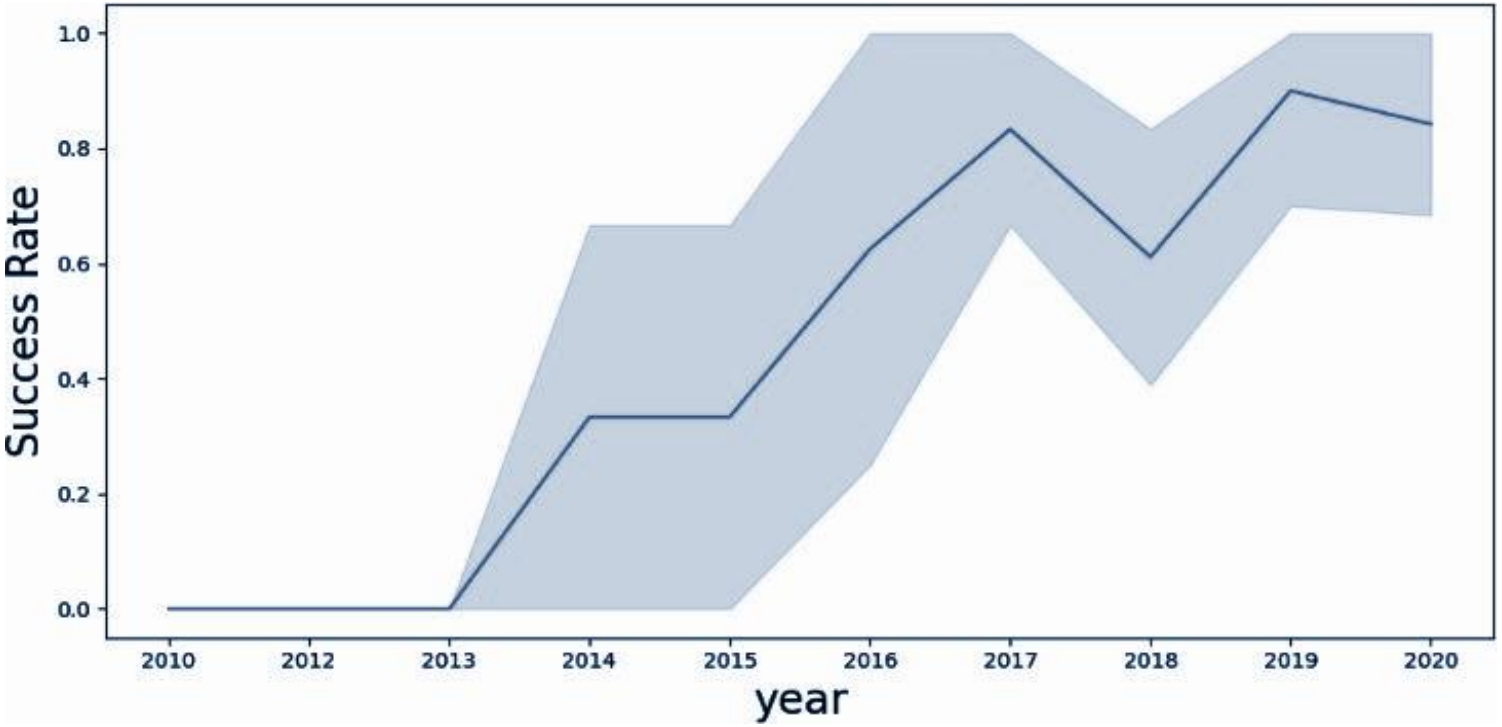
Payload vs. Orbit

- Heavy payloads are better with LEO, ISS and PO orbits
- With heavier payloads the GTO orbit has mixed success



Launch Success Yearly Trend

- From 2013-2017 and 2018-2019 the success rate has improved
- From 2017-2018 and from 2019-2020 the success rate has declined
- Since 2013 the success rate has improved.



All Launch Site Information

Launch Site Names :

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

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

In [10]:
```
task_1 = '''
        SELECT DISTINCT LaunchSite
        FROM SpaceX
'''
create_pandas_df(task_1, database=conn)
```

Out[10]:

	launchsite
0	KSC LC-39A
1	CCAFS LC-40
2	CCAFS SLC-40
3	VAFB SLC-4E

Launch Site Starting with CCA

```
%sql SELECT * \
    FROM SPACEXTBL \
    WHERE LAUNCH_SITE LIKE'CCA%' LIMIT 5;
```

```
 * ibm_db_sa://yyy33800:***@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:32286/BLUDB
   sqlite:///my_data1.db
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

Total Payload Mass:

45,596 kg (total) carried by boosters launched by NASA (CRS)

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) \
    FROM SPACEXTBL \
    WHERE CUSTOMER = 'NASA (CRS)';
```

```
 * ibm_db_sa://yyy33800:***@1bbf73c5-d84a-4
   sqlite:///my_data1.db
Done.
```

1
45596

Average Payload Mass:

2,928 kg (average) carried by booster version F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) \
    FROM SPACEXTBL \
    WHERE BOOSTER_VERSION = 'F9 v1.1';
```

```
 * ibm_db_sa://yyy33800:***@1bbf73c5-d84a-4
   sqlite:///my_data1.db
Done.
```

1
2928

Landing & Mission Info

1st Successful Landing in Ground Pad Landing Date: 12/22/2015

Successful Drone Ship Landing with Payload between 4000 and 6000:
JSCAT-14, JSCAT-16, SES-10, SES-11 / EchoStar 105

```
%sql SELECT MIN(DATE) \
FROM SPACEXTBL \
WHERE LANDING__OUTCOME = 'Success (ground pad)'
```

```
 * ibm_db_sa://yyy33800:***@1bbf73c5-d84a-4bb0-85b
   sqlite:///my_data1.db
Done.
```

1
2015-12-22

```
%sql SELECT PAYLOAD \
FROM SPACEXTBL \
WHERE LANDING__OUTCOME = 'Success (drone ship)' \
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

```
 * ibm_db_sa://yyy33800:***@1bbf73c5-d84a-4bb0-85b9-
   sqlite:///my_data1.db
Done.
```

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

Total Number of Successful and Failed Mission Outcomes:
1 Failure in Flight

99 Success

1 Success (payload status unclear)

```
%sql SELECT MISSION_OUTCOME, COUNT(*) as total_number \
FROM SPACEXTBL \
GROUP BY MISSION_OUTCOME;
```

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

Mission_Outcome	total_number
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters 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

```
%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

In 2015, Showing month, date, booster version, launch site and landing outcome

```
%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
01	10-01-2015	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	14-04-2015	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

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

We applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order**.**

```
%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.
```

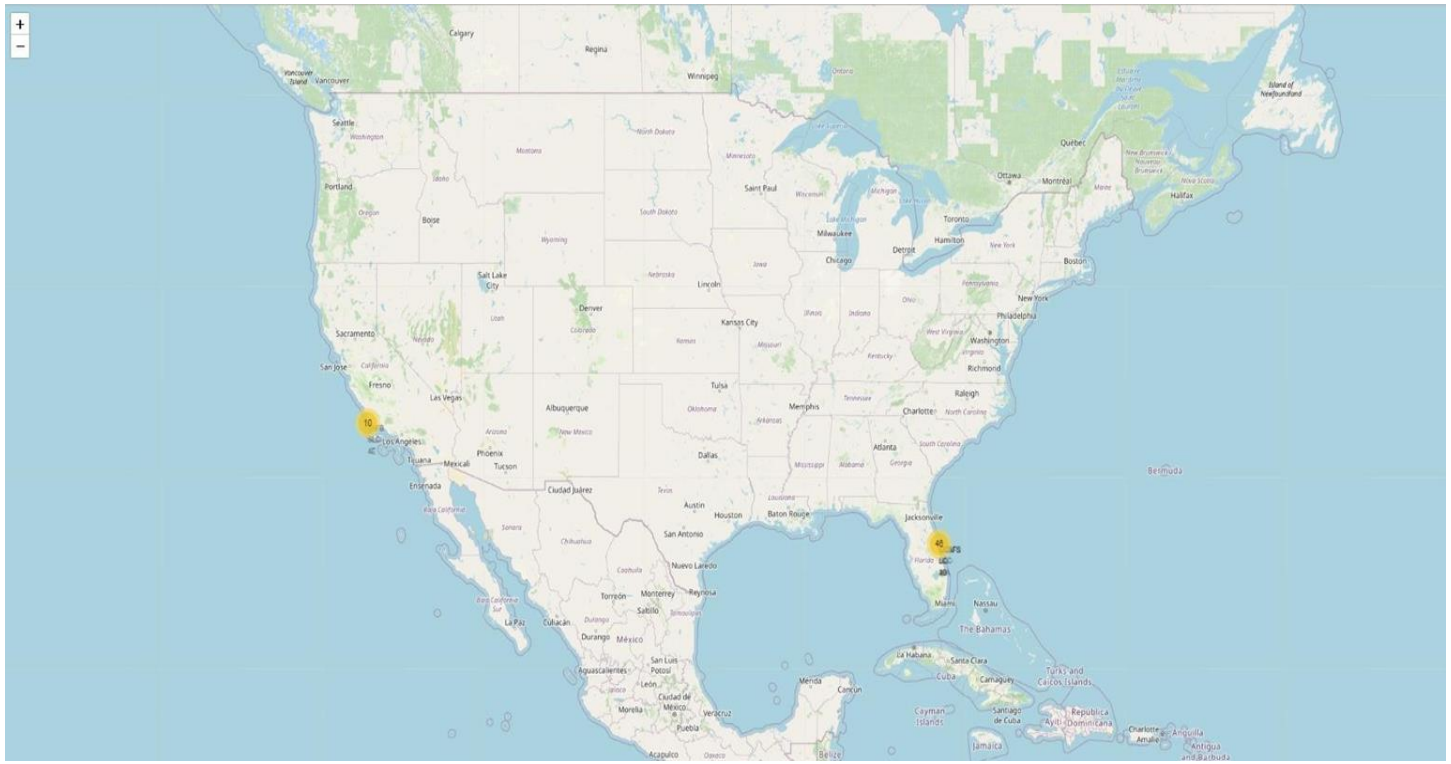
Landing _Outcome	count_outcomes
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Failure	3
Controlled (ocean)	3
Failure (parachute)	2
No attempt	1

Section 3

Launch Sites Proximities Analysis

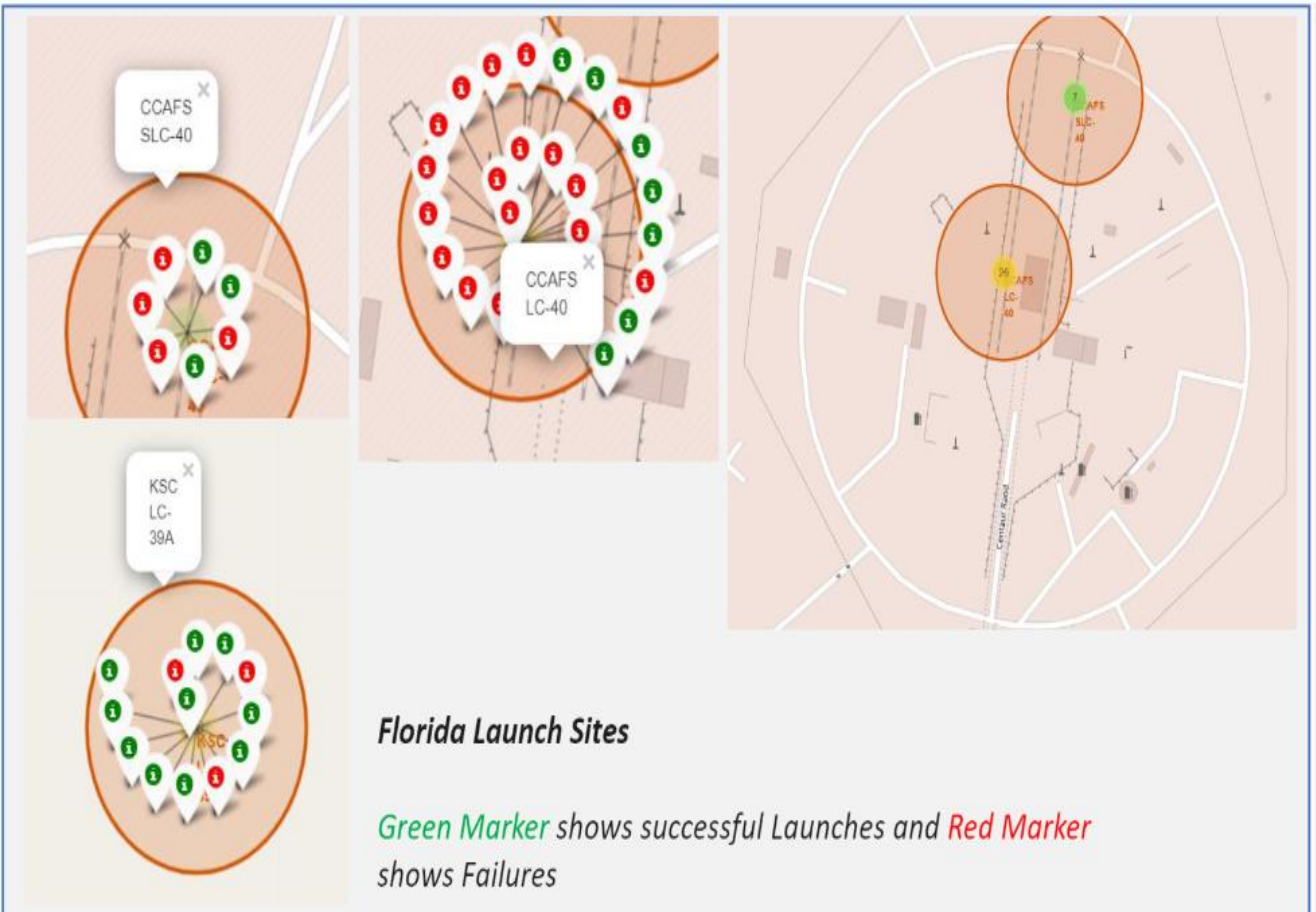
All launch sites global map markers

Near Equator: 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.



Markers showing launch sites with color labels

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 distance to landmarks

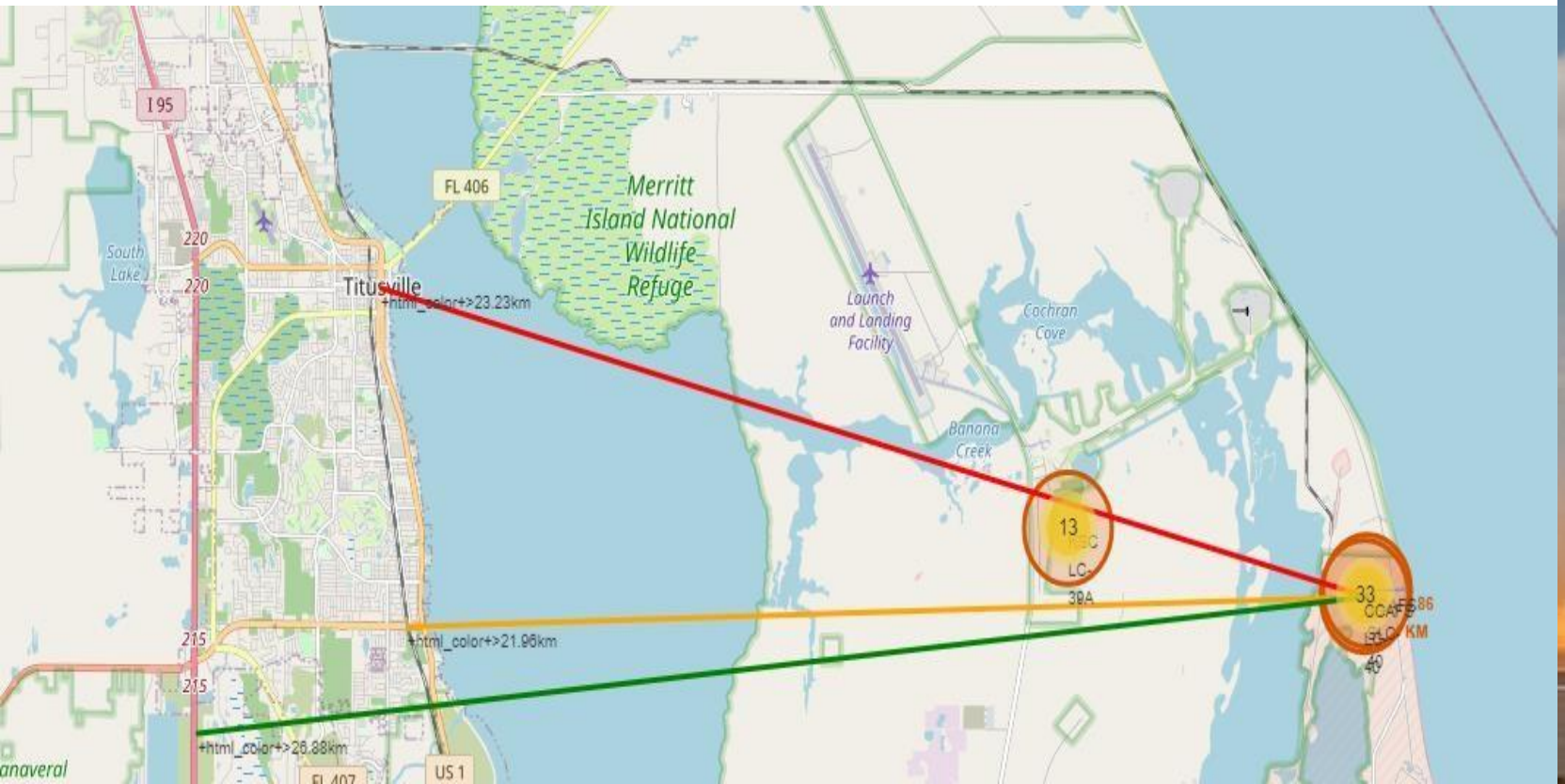
CCAFS SLC-40

.86 km from nearest coastline

21.96 km from nearest railway

23.23 km from nearest city

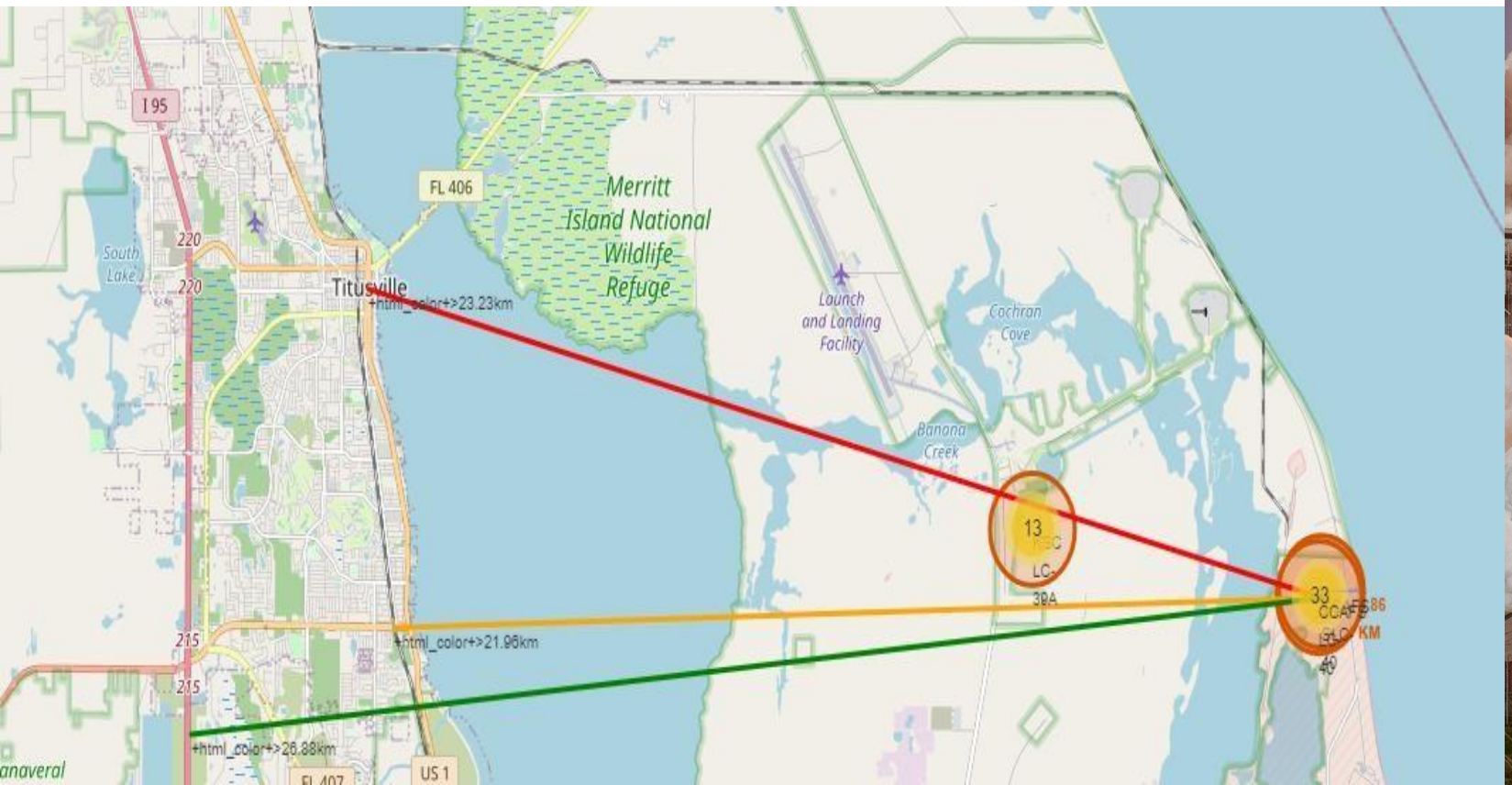
26.88 km from nearest highway



Distance to Proximities

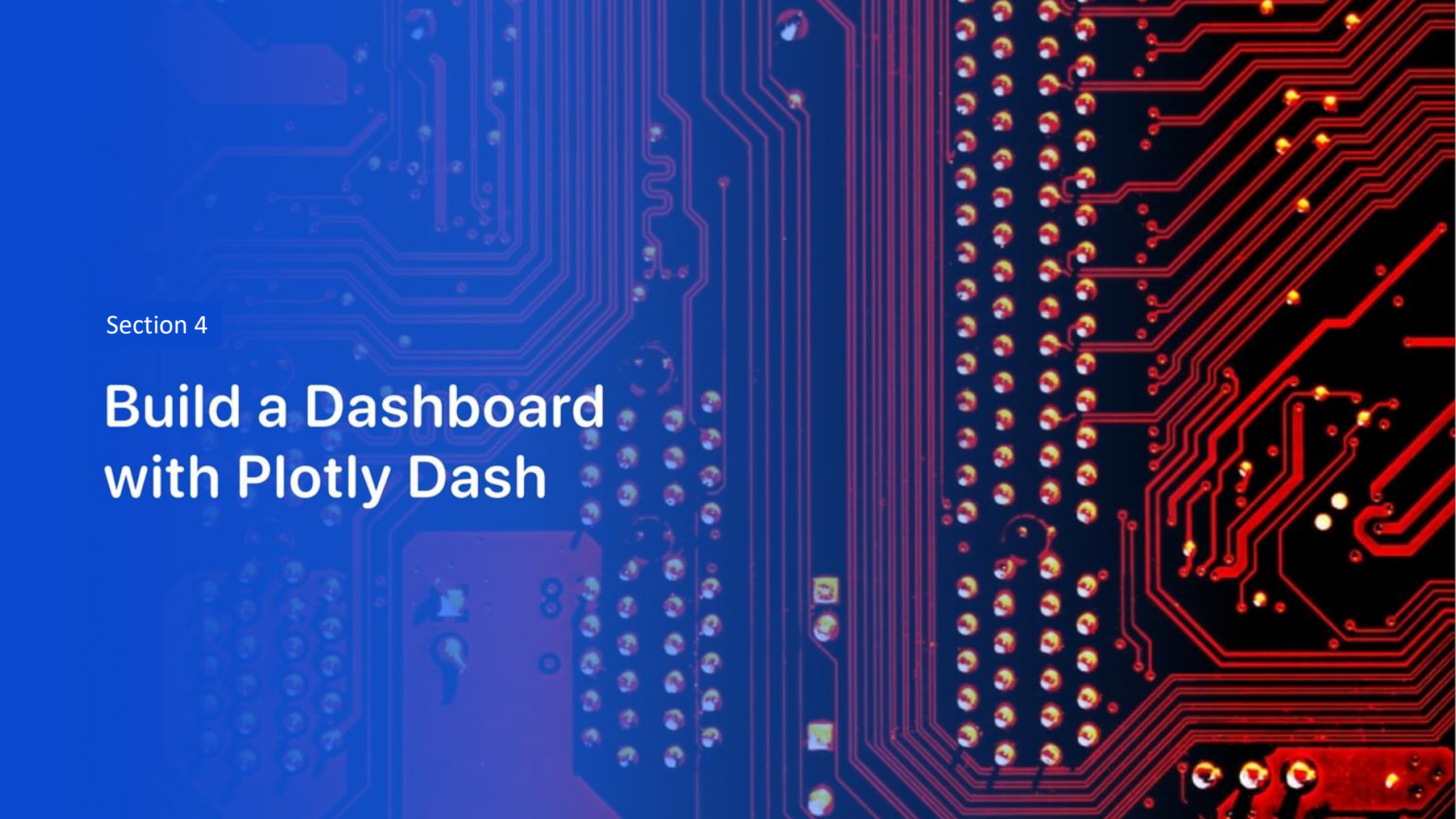
Coasts: help ensure that spent stages dropped along the launch path or failed launches don't fall on people or property.

Safety / Security: needs to be an exclusion zone around the launch site to keep unauthorized people away and keep people safe.



Section 4

Build a Dashboard with Plotly Dash



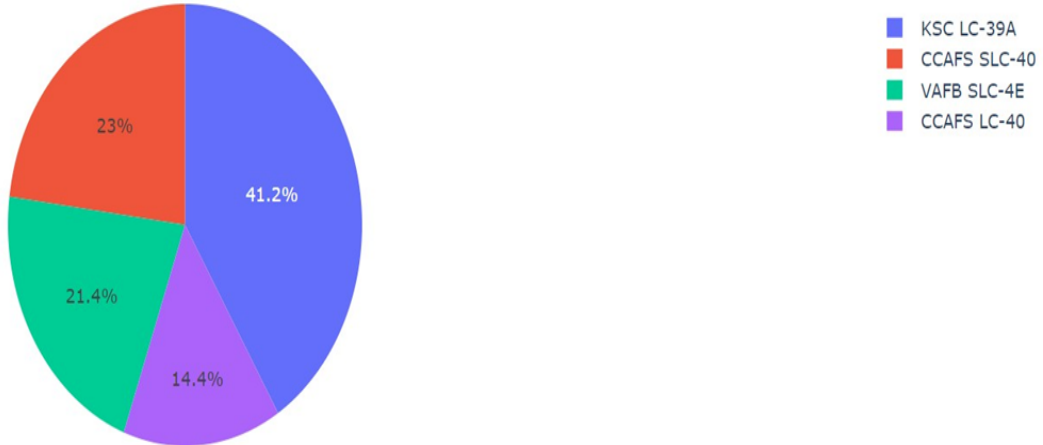
Pie chart showing the success percentage achieved by each launch site

The most successful launches amongst launch sites is KSC LC-39A (41.2%)

SpaceX Launch Records Dashboard

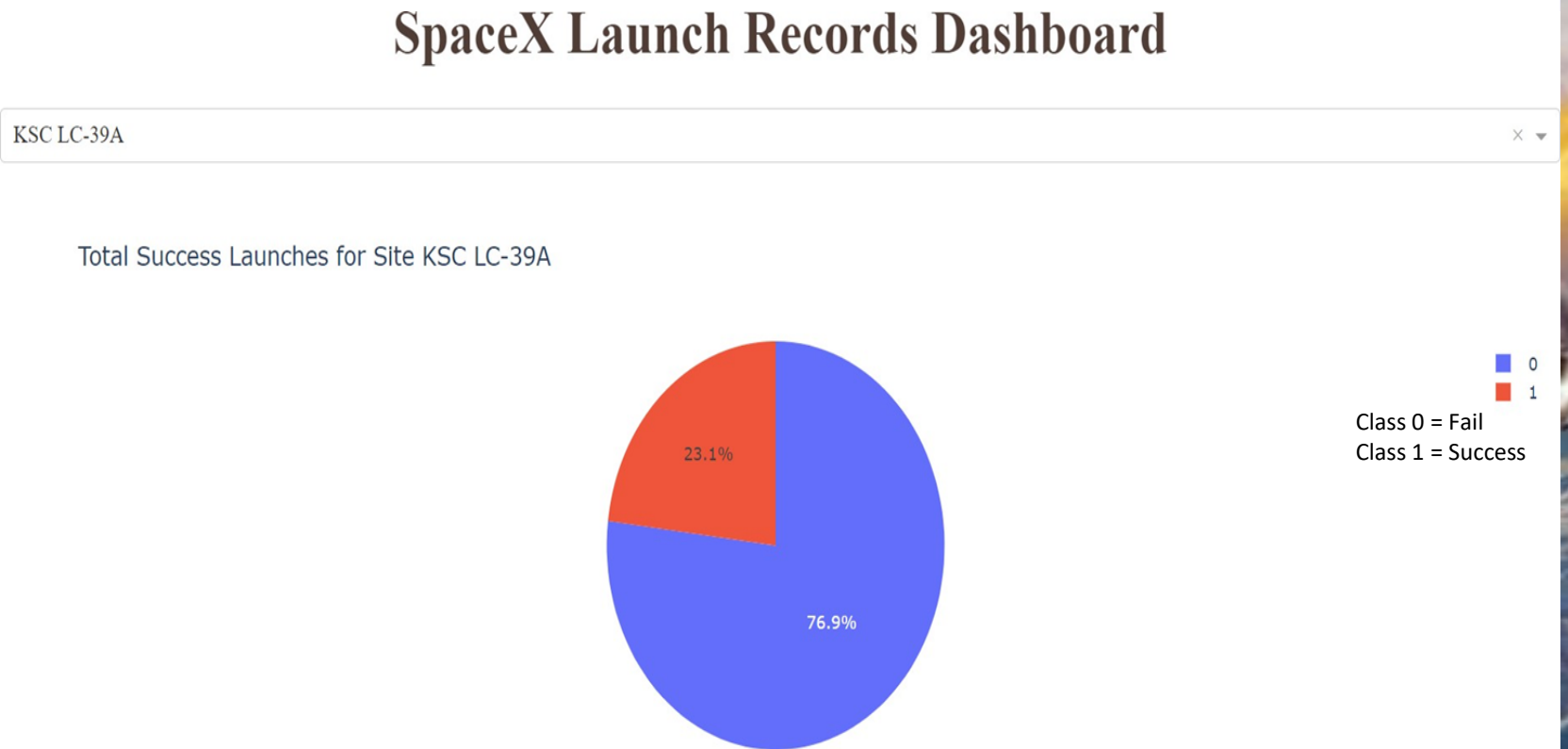
All Sites

Total Success Launches by Site



Highest Launch Success

The highest success rate amongst launch sites is KSC LC-39A(76.9%). 10 successful launches and only 3 failed launches



Scatter plot of Payload Mass Vs. Success for all sites

- Payloads between 2,000 kg and 5,000 kg have the highest success rate
- 1 indicating successful outcome and 0 indicating an unsuccessful outcome

Section 5

Predictive Analysis (Classification)

Classification Accuracy

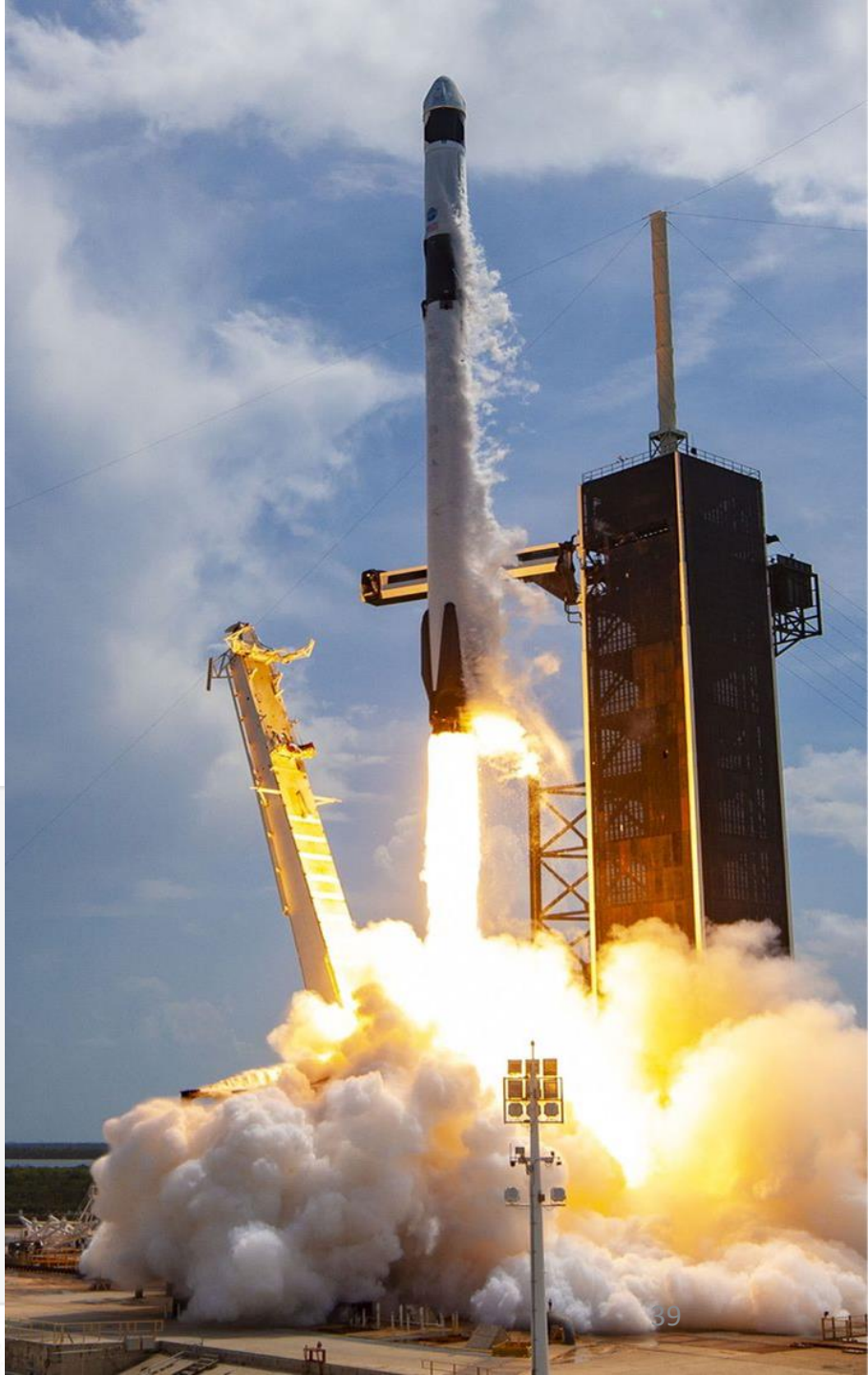
- All the models performed at about the same level and had the same scores and accuracy. This is likely due to the small dataset. The Decision Tree model slightly outperformed the rest when looking at .best_score_
- The decision tree classifier is the model with the highest classification accuracy

	LogReg	SVM	Tree	KNN
Jaccard_Score	0.800000	0.800000	0.800000	0.800000
F1_Score	0.888889	0.888889	0.888889	0.888889
Accuracy	0.833333	0.833333	0.833333	0.833333

```
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_)
```

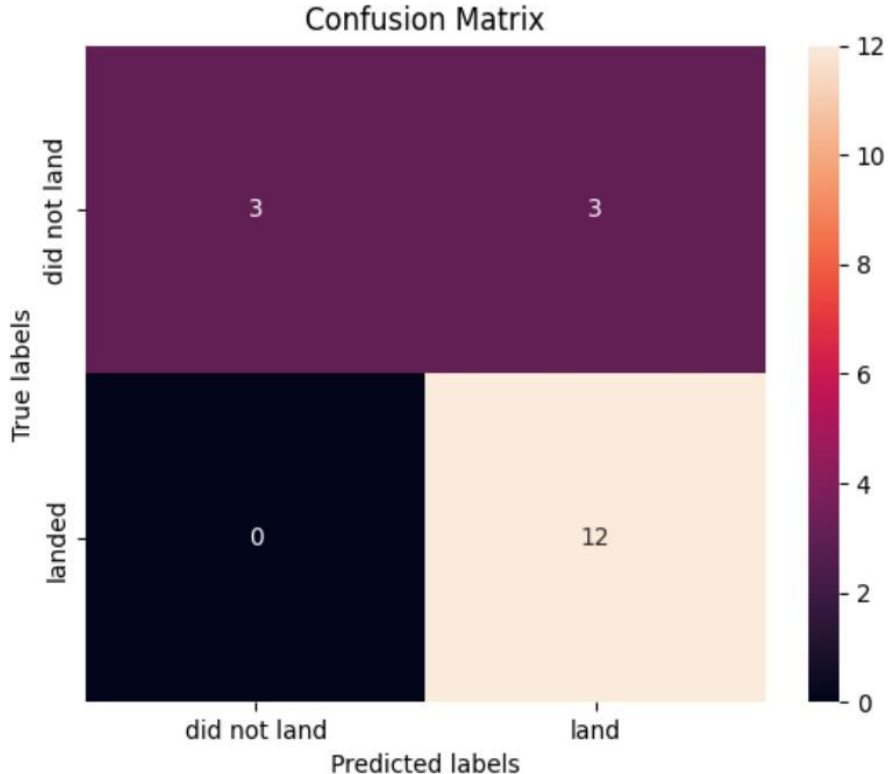
```
Best model is DecisionTree with a score of 0.9017857142857142
Best params is : {'criterion': 'gini', 'max_depth': 16, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_split': 10, 'splitter': 'random'}
```

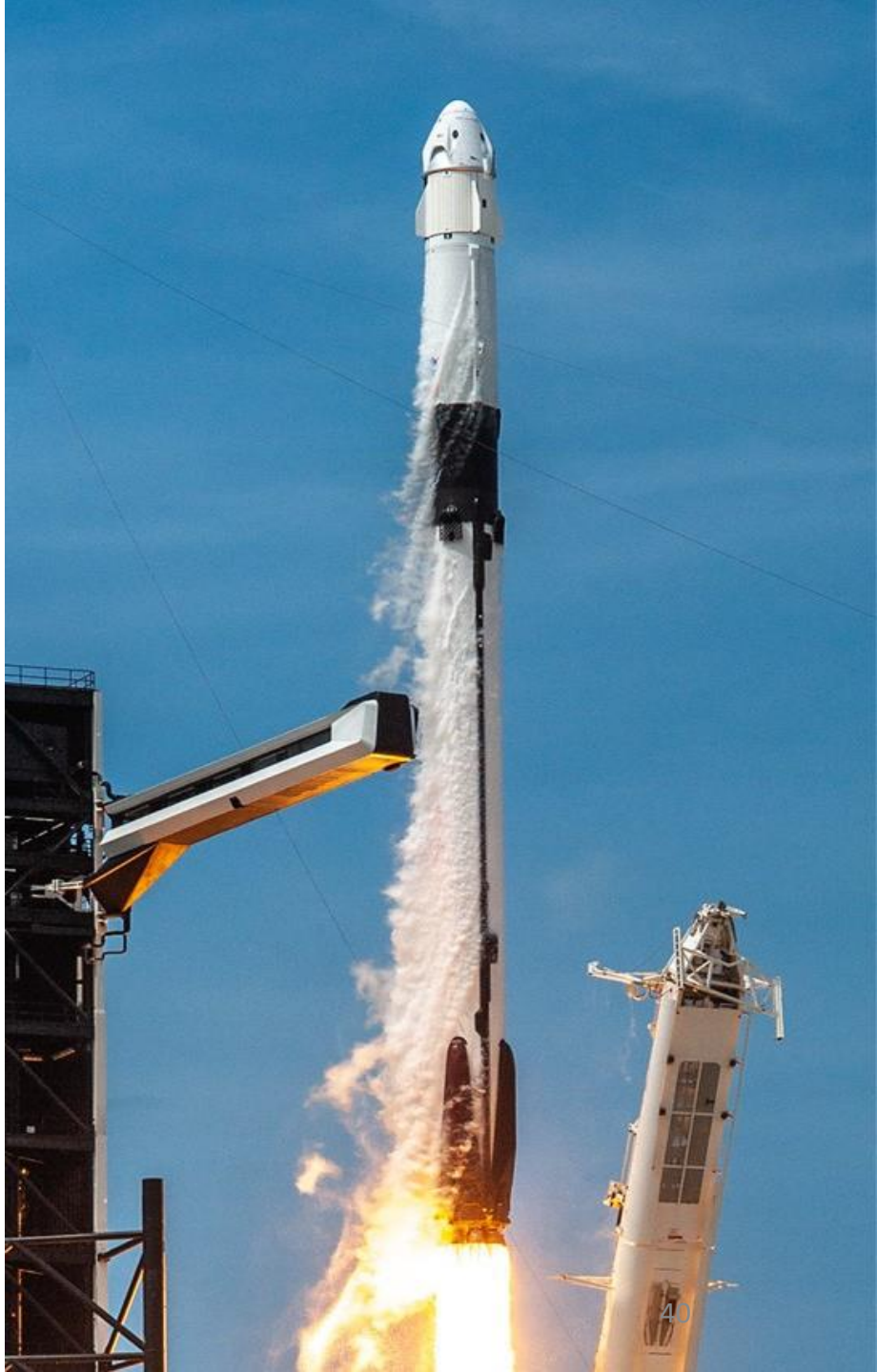


Confusion Matrices

Summary :

- A confusion matrix 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
- Confusion Matrix Outputs:
  - 12 True positive
  - 3 True negative
  - 3 False positive
  - 0 False Negative





Conclusion

- Except the decision tree model slightly outperforming rest of the models performed similarly on the test set with.
- Natural boost due to the rotational speed of earth helps save the cost of putting in extra fuel and boosters hence most of the launch sites are near the equator.
- 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
- Across all launch sites, the higher the payload mass (kg), the higher the success rate

Thank you!