IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Camilo Torrón
13/01/2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - EDA with data visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Building a Dashboard with Plotly Dash
  - Predictive analysis (Classification)
- Summary of all results
  - Exploratory data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

Introduction

- Project background and context

We predicted if the Falcon 9 first stage will land successfully. 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 SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems you want to find answers

- What influences if the rocket will land successfully?
- The effect each relationship with certain rocket variables will impact in determining the success rate of a successful landing.
- What conditions does SpaceX have to achieve to get the best results and ensure the best rocket success landing rate.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The data was collected from SpaceX Rest API and Web scrapping from Wikipedia
- Perform data wrangling
  - The data was transformed to One Hot Encoding in order to feed a Machine Learning model
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

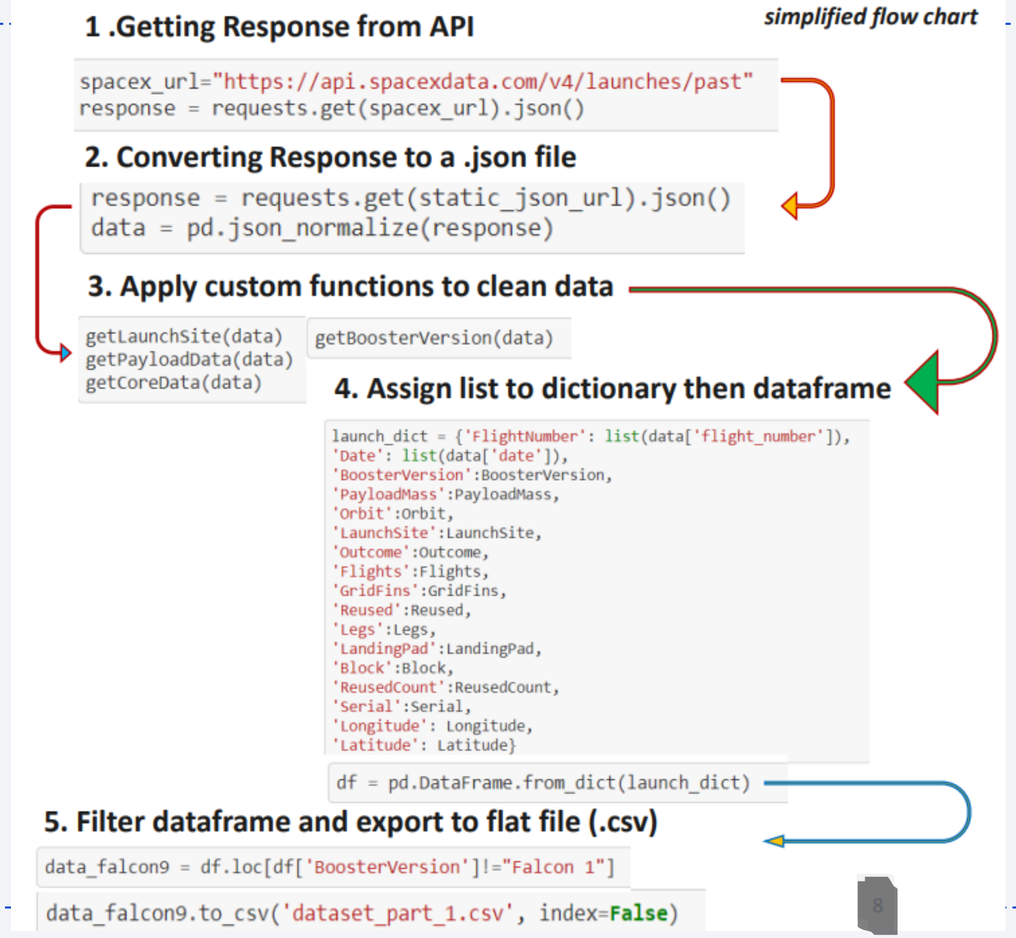
- SpaceX Rest API: the API returns a .json file with all the information we need

- Web Scrapping: Beautiful soup was used in order to obtain the info. After it was transformed into a .csv file.

Data Collection – SpaceX API

- This was the process

- https://github.com/camilotorron/Curso-Applied-Data-Science-Capstone/blob/master/01_Collecting%20the%20data.ipynb



Data Collection - Scraping

- This was the process using Beautiful Soup

- https://github.com/camilotorron/Curso-Applied-Data-Science-Capstone/blob/master/02_Data%20collection%20with%20Web%20Scraping.ipynb

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

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

```python
# Use the find_all function in the BeautifulSoup object, with element type `table`
html_tables = soup.find_all('table')
```

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

```python
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
```

Data Wrangling

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.

- https://github.com/camilotorron/Curso-Applied-Data-Science-Capstone/blob/master/03_Data%20wrangling.ipynb

EDA with Data Visualization

Plots to visualize the relationship between:

- Flight Number and Launch Site
- Payload and Launch Site
- Success rate of each orbit type
- FlightNumber and Orbit type
- Payload and Orbit type
- Success yearly trend

- https://github.com/camilotorron/Curso-Applied-Data-Science-Capstone/blob/master/05_EDA%20with%20Data%20Visualization.ipynb

EDA with SQL

In order to obtain meaningful data I performed the following queries:

- Displayed the names of the unique launch sites in the space mission
- Displayed 5 records where launch sites begin with the string 'KSC'
- Displayed the total payload mass carried by boosters launched by NASA (CRS)
- Displayed average payload mass carried by booster version F9 v1.1
- Listed the date where the successful landing outcome in drone ship was achieved.
- Listed the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
- Listed the total number of successful and failure mission outcomes
- Listed the names of the booster versions which have carried the maximum payload mass.
- Listed the records which will display the month names, successful landing outcomes in ground pad ,booster versions, launch_site for the months in year 2017
- Ranking the count of successful landing_out comes between the date 2010-06-04 and 2017-03-20 in descending order.

• https://github.com/camilotorron/Curso-Applied-Data-Science-Capstone/blob/master/04_EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- To visualize the Launch Data into an interactive map, I took the Latitude and Longitude Coordinates at each launch site and added a Circle Marker around each launch site with a label of the name of the launch site.
- I assigned the dataframe launch_outcomes(failures, successes) to classes 0 and 1 with Greenand Redmarkers on the map in a MarkerCluster()
- Using Haversine's formula i calculate the distance from the Launch Site to various landmarks to find various trends about what is around the Launch Site to measure patterns. Linesare drawn on the map to measure distance to landmarks

- https://github.com/camilotorron/Curso-Applied-Data-Science-Capstone/blob/master/06_Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- https://github.com/camilotorron/Curso-Applied-Data-Science-Capstone/blob/master/06_Visual%20Analytics%20with%20Folium.ipynb

Predictive Analysis (Classification)

BUILDING MODEL

- **L**oad our dataset into NumPy and Pandas
- Transform Data
- Split our data into training and test data sets
- Check how many test samples we have
- Decide which type of machine learning algorithms we want to use
- Set our parameters and algorithms to GridSearchCV
- Fit our datasets into the GridSearchCVobjects and train our dataset.

EVALUATING MODEL

- Check accuracy for each model
- Get tuned hyperparameters for each type of algorithms
- Plot Confusion Matrix
- **IMPROVING MODEL**
- Feature Engineering
- Algorithm Tuning

FINDING THE BEST PERFORMING CLASSIFICATION MODEL

- The model with the best accuracy score wins the best performing model
- In the notebook there is a dictionary of algorithms with scores at the bottom of the notebook.

- https://github.com/camilotorron/Curso-Applied-Data-Science-Capstone/blob/master/07_Predictive%20Analysis.ipynb

Results

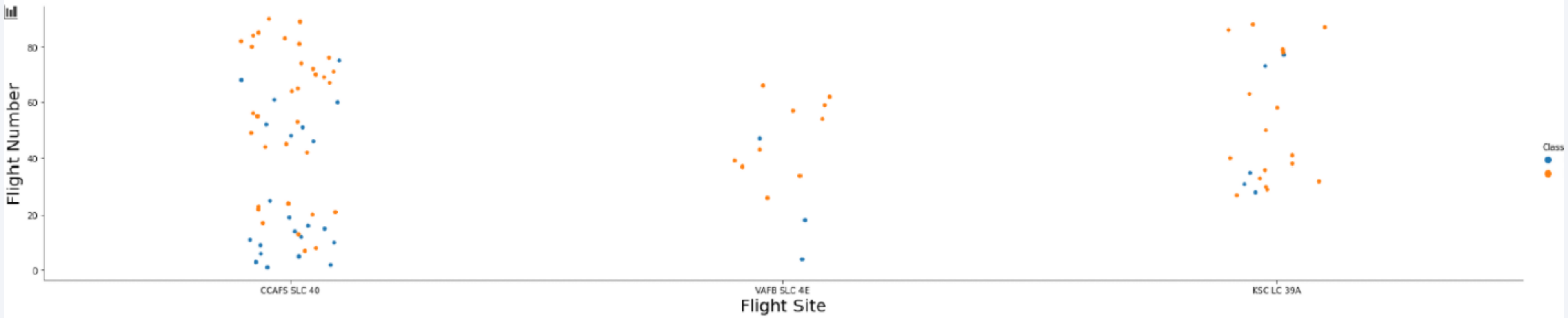
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

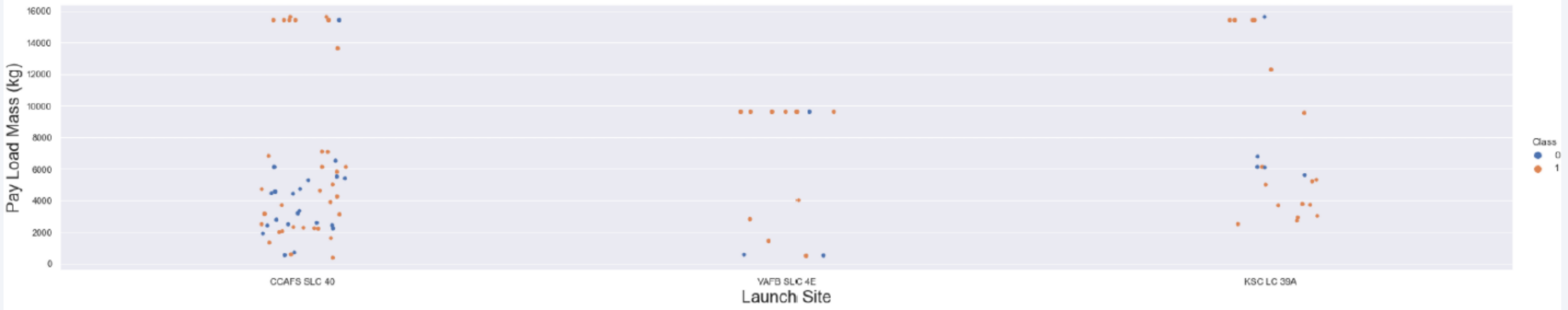
Flight Number vs. Flight Site



The more amount of flights at a launch site the greater the success rate at a launch site.

Payload vs. Launch Site

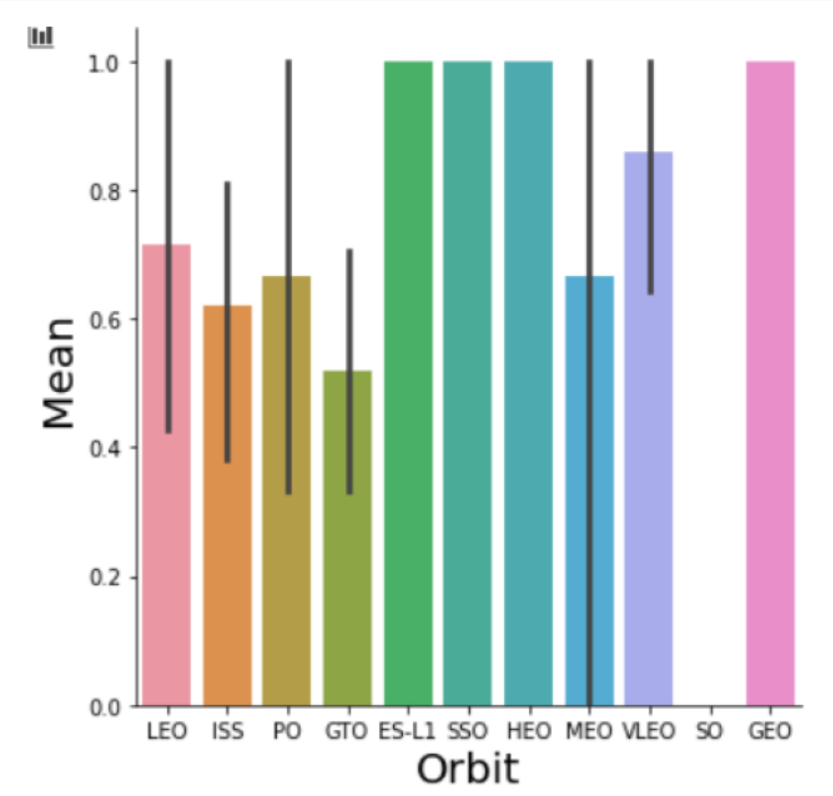
Payload Mass vs. Launch Site



The greater the payload mass for Launch Site CCAFS SLC 40 the higher the success rate for the Rocket. There is not quite a clear pattern to be found using this visualization to make a decision if the Launch Site is dependant on Pay Load Mass for a success launch.

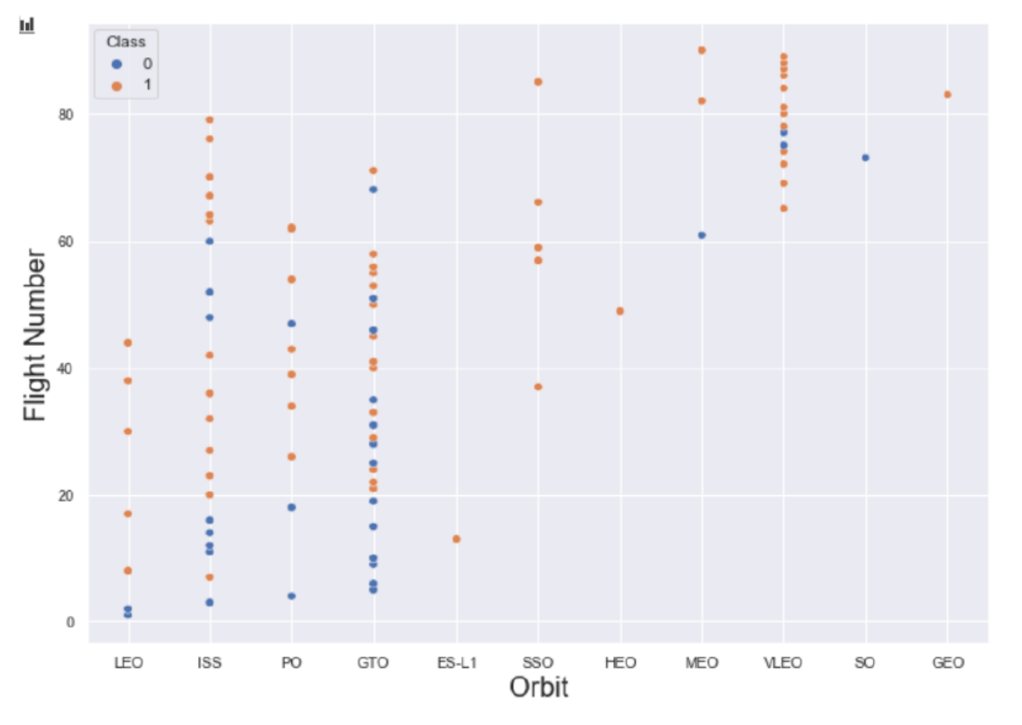
Success Rate vs. Orbit Type

- Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate



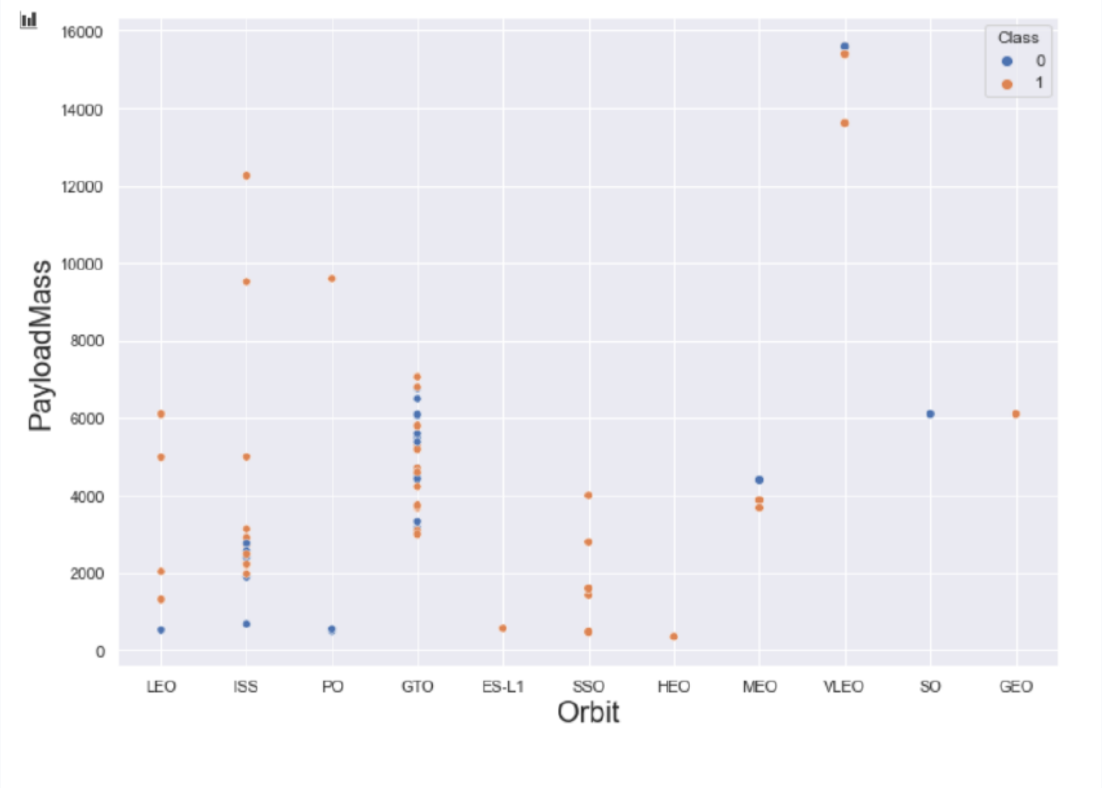
Flight Number vs. Orbit Type

In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



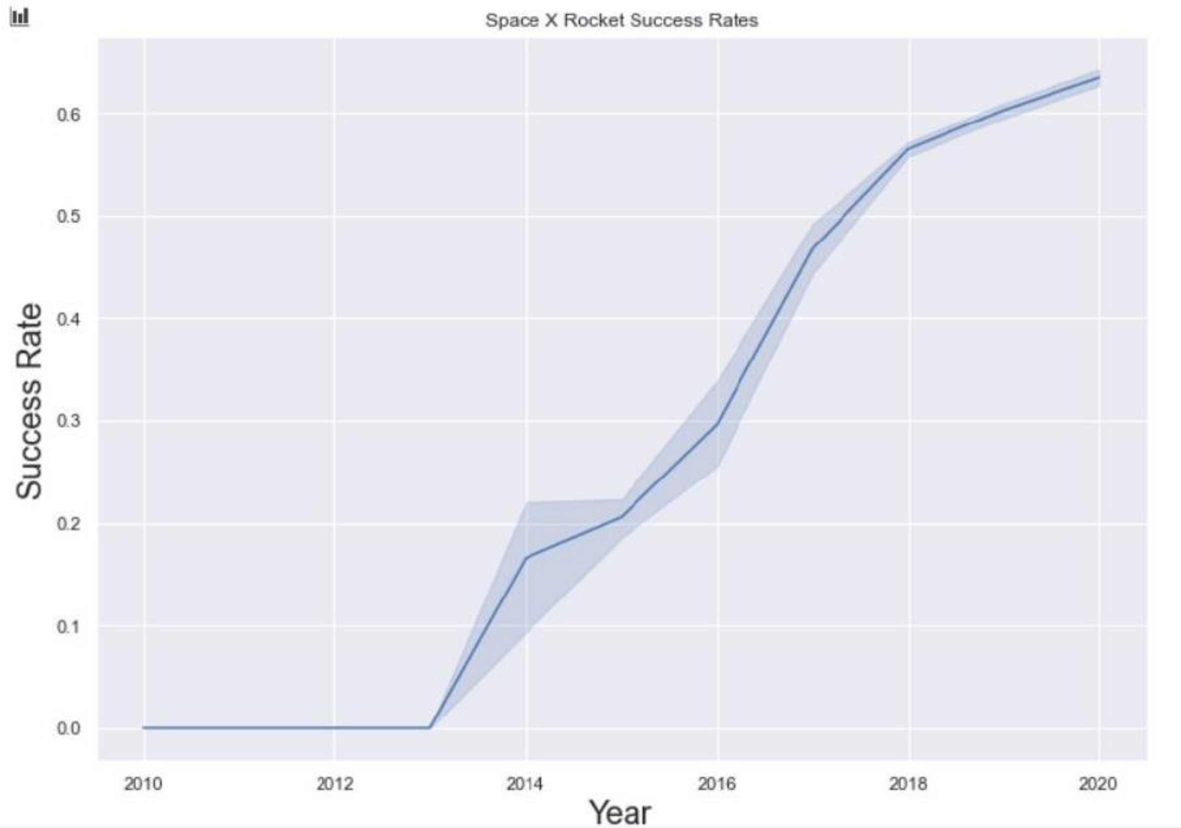
Payload vs. Orbit Type

- Havy payloads have a negative impact on GTO orbits and positive on Gtço ad Polar LEA (ISS) orbits.



Launch Success Yearly Trend

- Success rate is improving since 2013



All Launch Site Names

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

In [12]:
```
%sql SELECT Distinct LAUNCH_SITE FROM XRF37902.SPACEXTBL
```

```
 * ibm_db_sa://xrf37902:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[12]:

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

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'

In [13]:
```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://xrf37902:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[13]:

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)
2012-08-10	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
2013-03-12	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES	Success	No attempt

Total Payload Mass

Display the total payload mass carried by boosters launched by NASA (CRS)

```
In [14]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER='NASA (CRS)'

 * ibm_db_sa://xrf37902:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[14]:

1
22007

Average Payload Mass by F9 v1.1

Display average payload mass carried by booster version F9 v1.1

In [15]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION='F9 v1.1'
```

```
 * ibm_db_sa://xrf37902:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[15]:

1
3676

First Successful Ground Landing Date

List the date when the first successful landing outcome in ground pad was acheived.

Hint:Use min function

In [16]:
```
%sql SELECT min(DATE) FROM SPACEXTBL WHERE LANDING__OUTCOME='Success (ground pad)'
```

```
 * ibm_db_sa://xrf37902:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[16]:

1
2017-01-05

Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [17]:
```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 AND LANDING__OUTCOME='Success (drone sh
ip)'
```

```
 * ibm_db_sa://xrf37902:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[17]:

booster_version
F9 FT B1022
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

In [18]:
```
%sql SELECT COUNT(*) FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE '%Success%' OR MISSION_OUTCOME LIKE '%Failure%'
```

```
 * ibm_db_sa://xrf37902:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[18]:

1
45

Boosters Carried Maximum Payload

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [19]:
```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

```
 * ibm_db_sa://xrf37902:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[19]:

booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3

2015 Launch Records

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

In [20]:
```
%sql SELECT TO_CHAR(TO_DATE(MONTH("DATE"), 'MM'), 'MONTH') AS MONTH_NAME, \
    LANDING__OUTCOME AS LANDING__OUTCOME, \
    BOOSTER_VERSION AS BOOSTER_VERSION, \
    LAUNCH_SITE AS LAUNCH_SITE \
    FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND "DATE" LIKE '%2015%'
```

```
 * ibm_db_sa://xrf37902:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[20]:

month_name	landing__outcome	booster_version	launch_site
OCTOBER	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40

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

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

In [21]:
```
%sql SELECT "DATE", COUNT(LANDING__OUTCOME) as COUNT FROM SPACEXTBL \
    WHERE "DATE" BETWEEN '2010-06-04' and '2017-03-20' AND LANDING__OUTCOME LIKE '%Success%' \
    GROUP BY "DATE" \
    ORDER BY COUNT(LANDING__OUTCOME) DESC
```

```
 * ibm_db_sa://xrf37902:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

Out[21]:

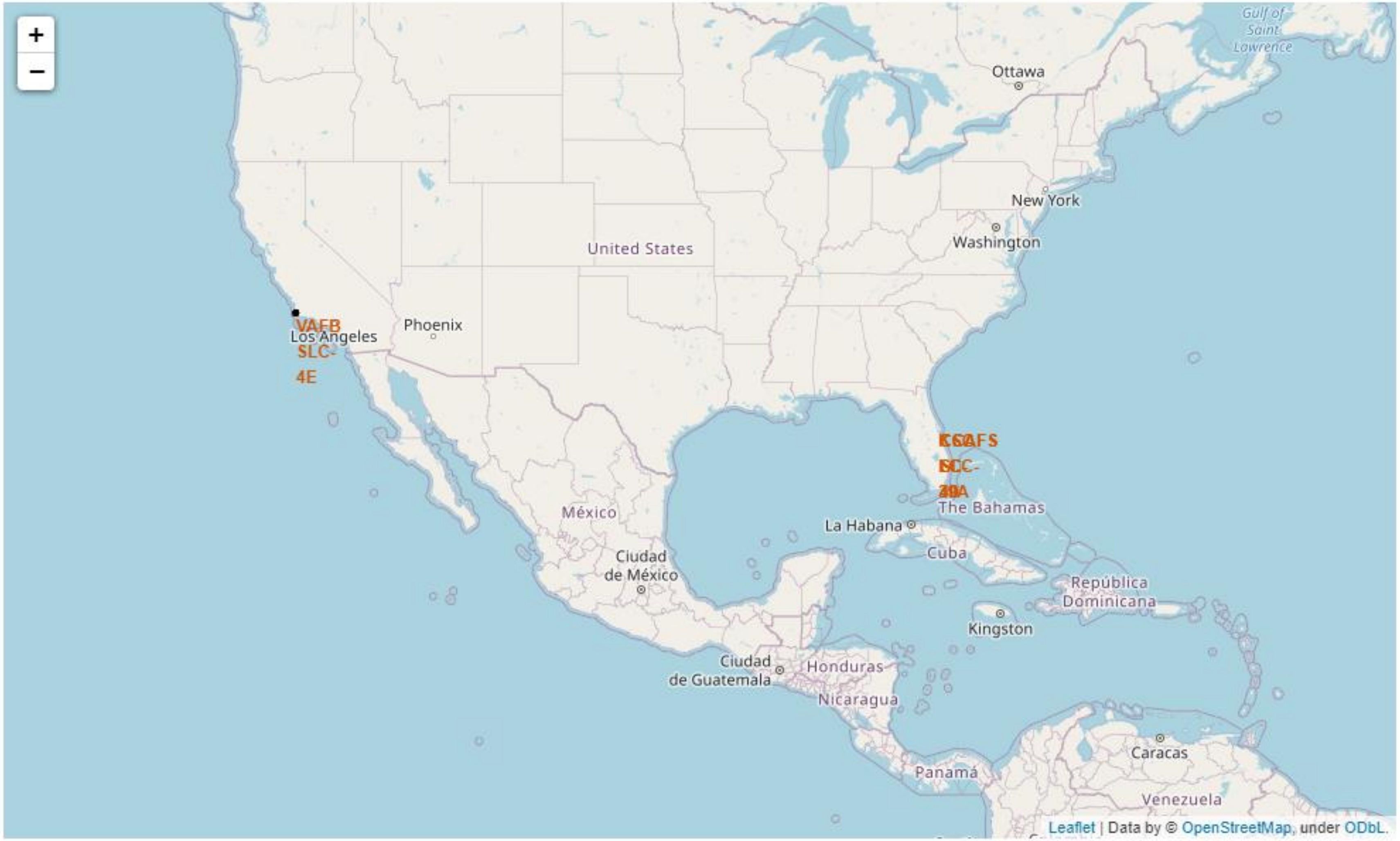
DATE	COUNT
2016-06-05	1
2016-08-04	1
2017-01-05	1
2017-03-06	1

Section 4

Launch Sites Proximities Analysis

Launch sites

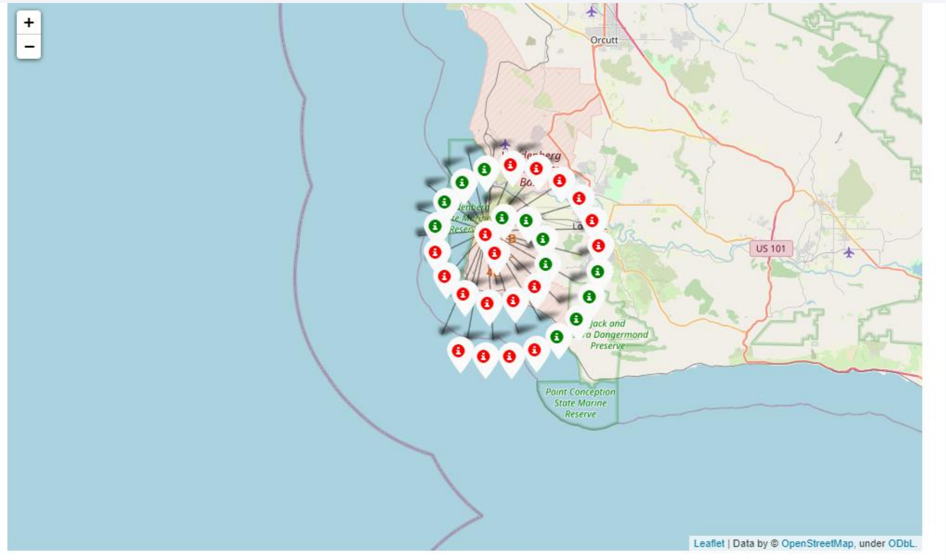
Out[10]:

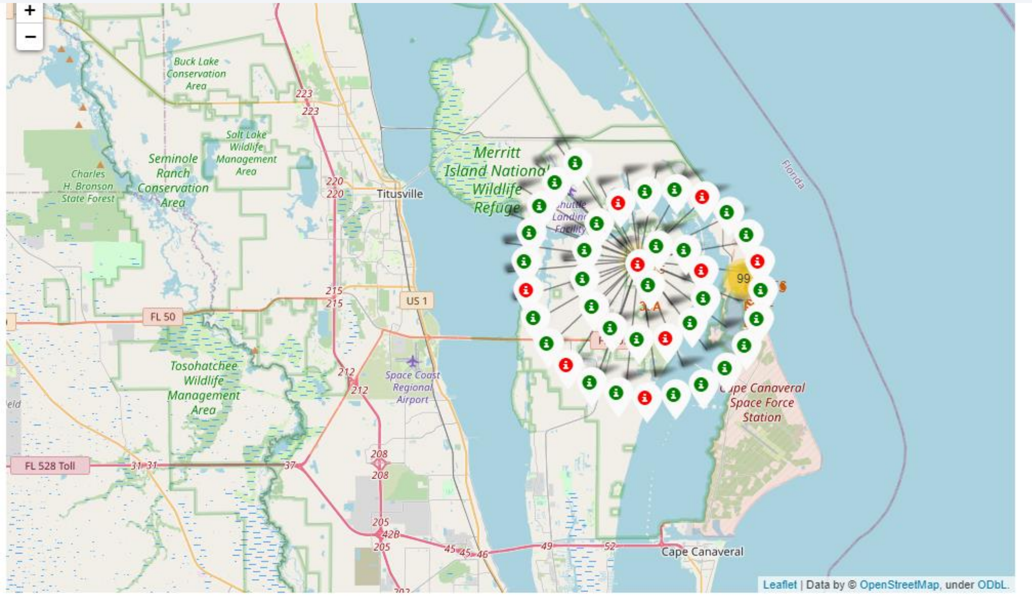


Colored Markers

- California launch sites

Orlando launch sites





Distance to coastline launch site

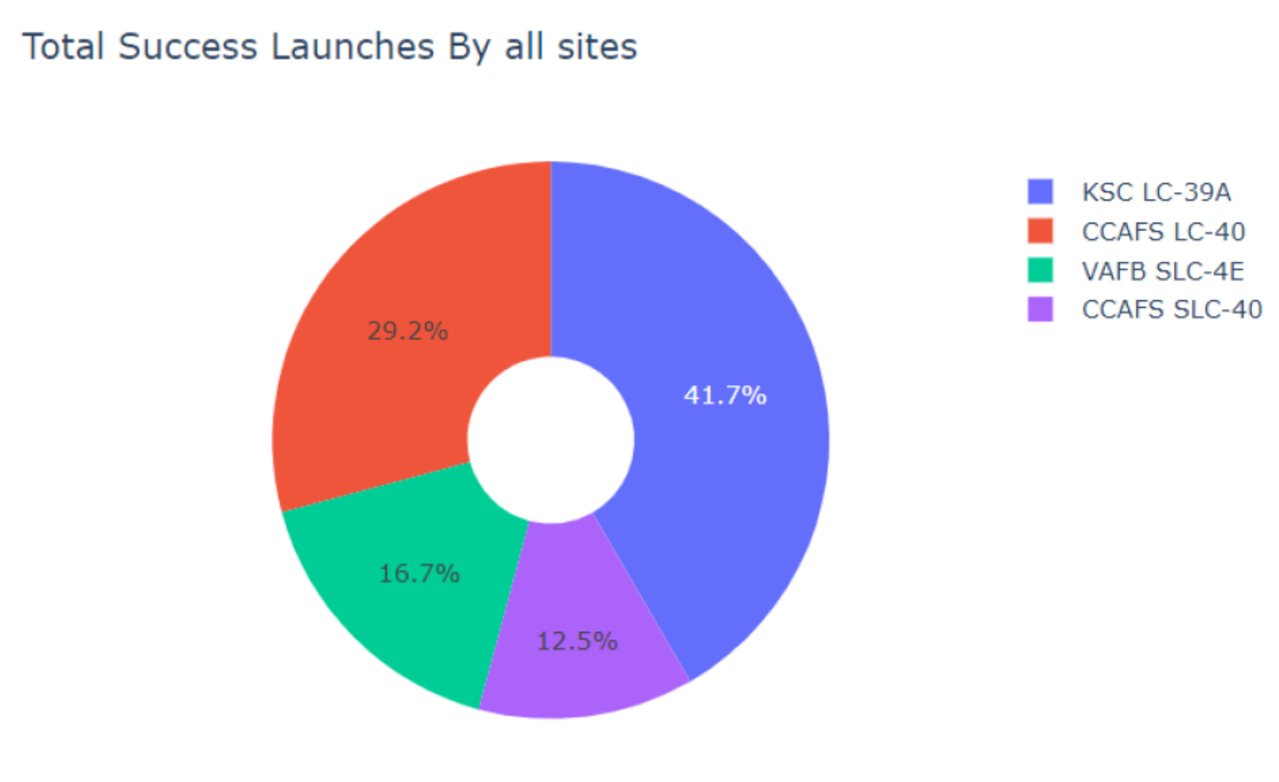
- Distance to coastline

Section 5

Build a Dashboard with Plotly Dash

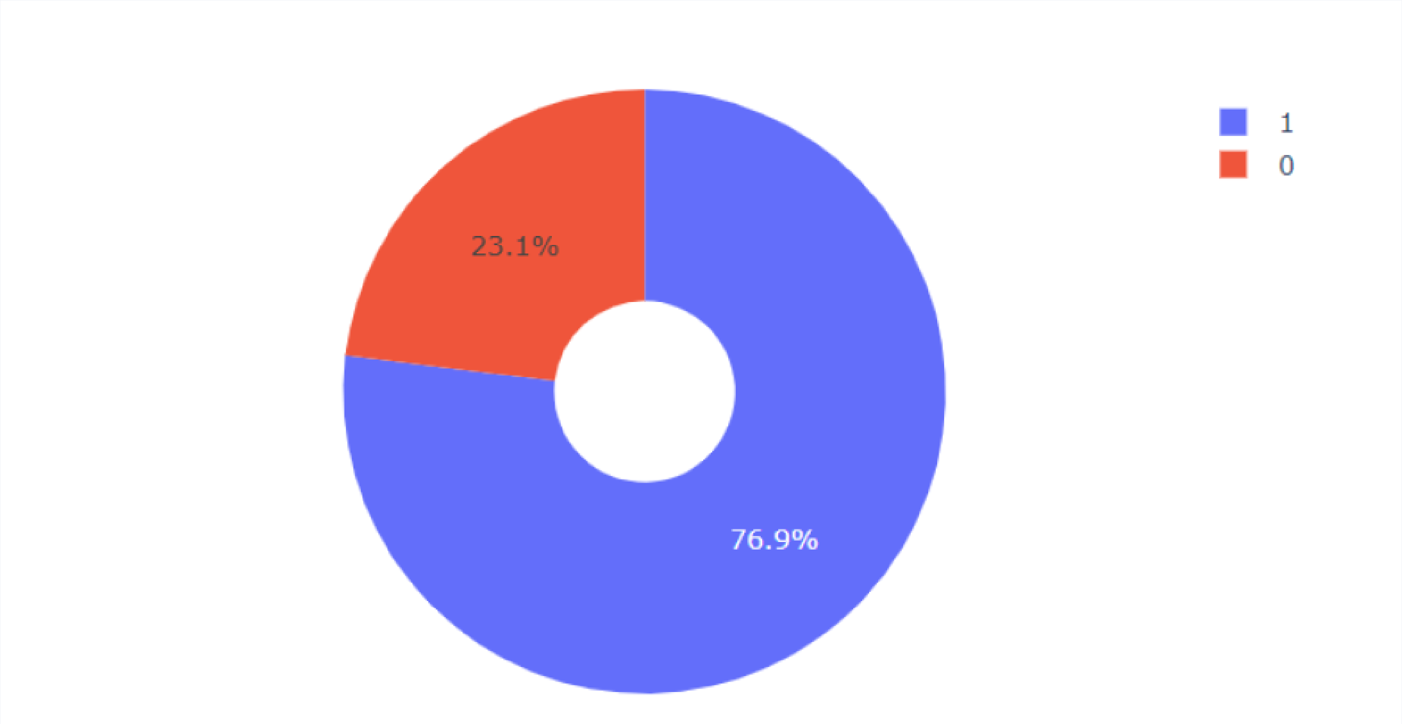
Success per launch site

- KSC LC-39A is the most successful launch site



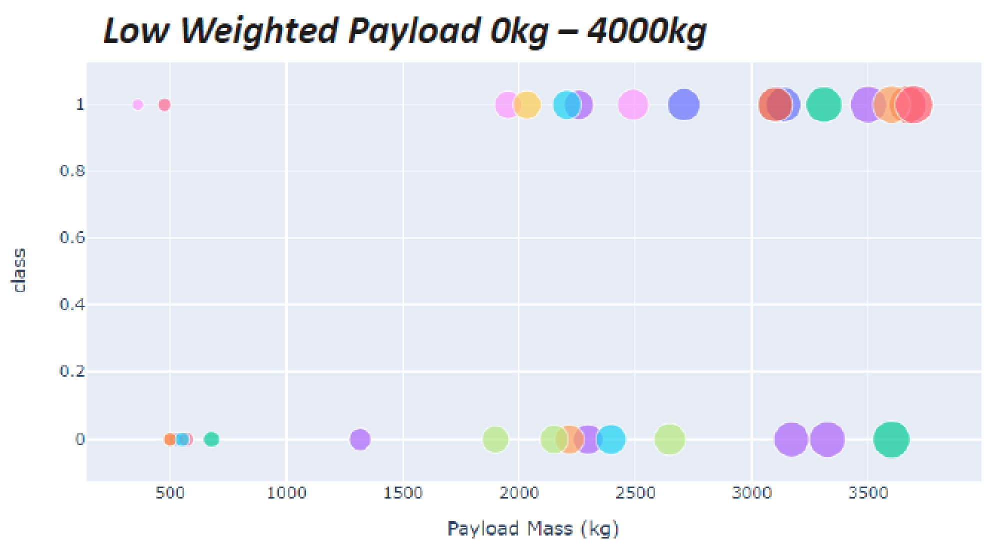
KSC LC-39A success rate

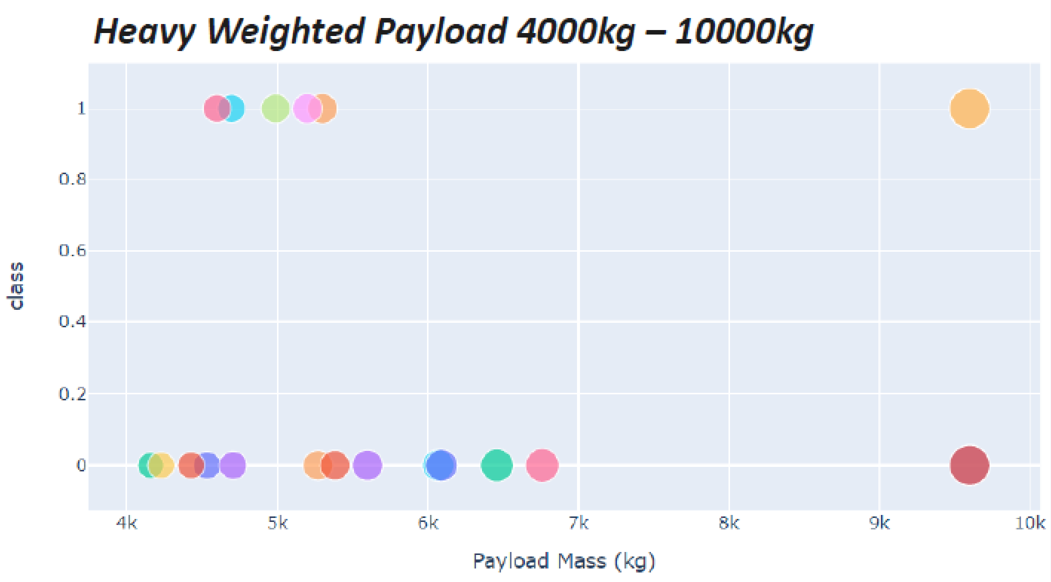
- This site has a 76.9% of success



Payload vs Lauch outcome

- Low weighted payload success rate is higher than heavy payloads



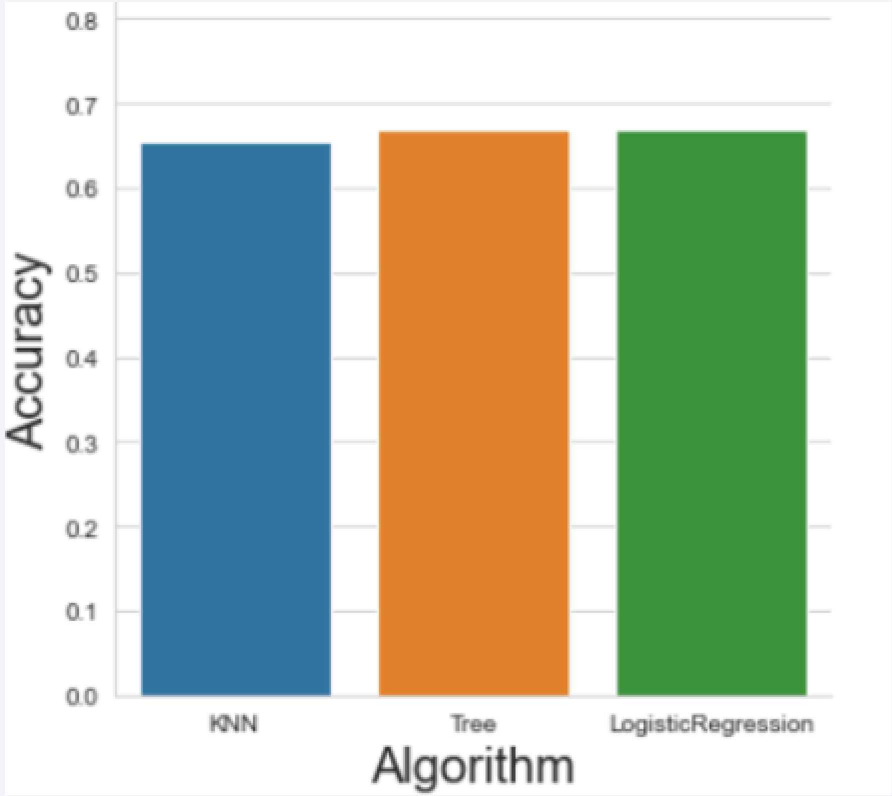


Section 6

Predictive Analysis (Classification)

Classification Accuracy

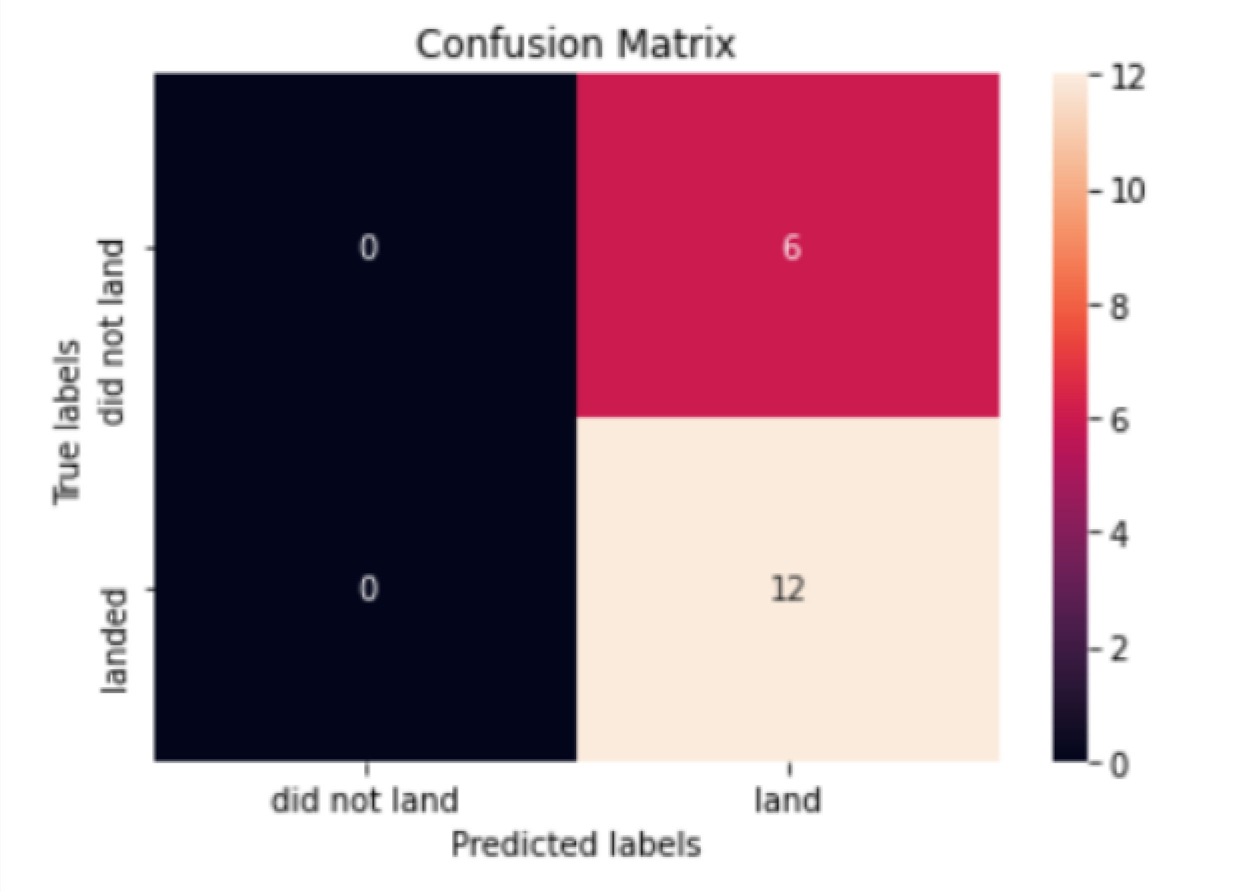
- The best accuracy is obtained with Tree algorithm



Confusion Matrix

- The biggest problem here is the false

Positives (6)



Conclusions

- The Tree Classifier Algorithm is the best for Machine Learning for this dataset

• Low weighted payloads perform better than the heavier payloads

• The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches

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

• Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!