IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Leon Schmitt>
<13.03.2022>

Outline

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

Executive Summary

- Data Collection through SpaceX-API and Web Scraping
- Data Wrangling
- Explored data using SQL,
- Visualization with folium maps, and dashboards
- Feature creation using one hot encoding
- predict successful Landings using different Machine Learning Models
- Find the best parameters for the modes by using GridSearchSV
- 4 models were used: Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K Nearest Neighbors.
- Similar results with accuracy rate of about 83.33%.
- With more data and/or other models like DNN even better results are possible.

Introduction

Background:

- Commercial Space Age is Here
- Our Company Space Y wants to compete with Space X
- Compared to other competitors, Space X has best pricing ($62 million vs. $165 million USD)
- Largely due to ability to recover the Stage 1 part of the rocket system

Problem:

- Space Y want to compete for this purpose they want to predict which factors determine if the rocket will land successfully by using different machine learning models.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - using SpaceX API and SpaceX Wikipedia page
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Tuned models using GridSearchCV

Data Collection

- The data was collected using various methods
  - Data collection was done using get request to the SpaceX API.
  - we transform the response with the .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.
  - Furthermore, we used BeatuifulSoup to perform web scraping for Falcon 9 launch records from Wikipedia
  - The objective was to extract the launch data and convert it to a pandas dataframe

Data Collection – SpaceX API

1. GET Request to SpaceX API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url).json()
```

2. Converting Response to json and normalize the data

```
response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```

3. Assign list to dictionary and apply function to clean the dataset

```
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

```
df = pd.DataFrame.from_dict(launch_dict)
```

4. Filter dataframe an export it to a csv file

```
data_falcon9 = df.loc[df['BoosterVersion']!="Falcon 1"]
```

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

https://github.com/Lo0815/IBM-Data-Science-Capstone/blob/master/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

1. GET Request to SpaceX API
Create BeautifulSoup Object and find all tables within it

2. Getting colums name

3. Append data to keys

4. Converting to dataframe an store it as a csv file

```
page = requests.get(static_url)
```

```
soup = BeautifulSoup(page.text, 'html.parser')
html_tables = soup.find_all('table')
```

```
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
```

```
In [12]: extracted_row = 0
         #Extract each table
         for table_number,table in enumera
            # get table row
             for rows in table.find_all("t
                #check to see if first ta
```

```
df=pd.DataFrame(launch_dict)
```

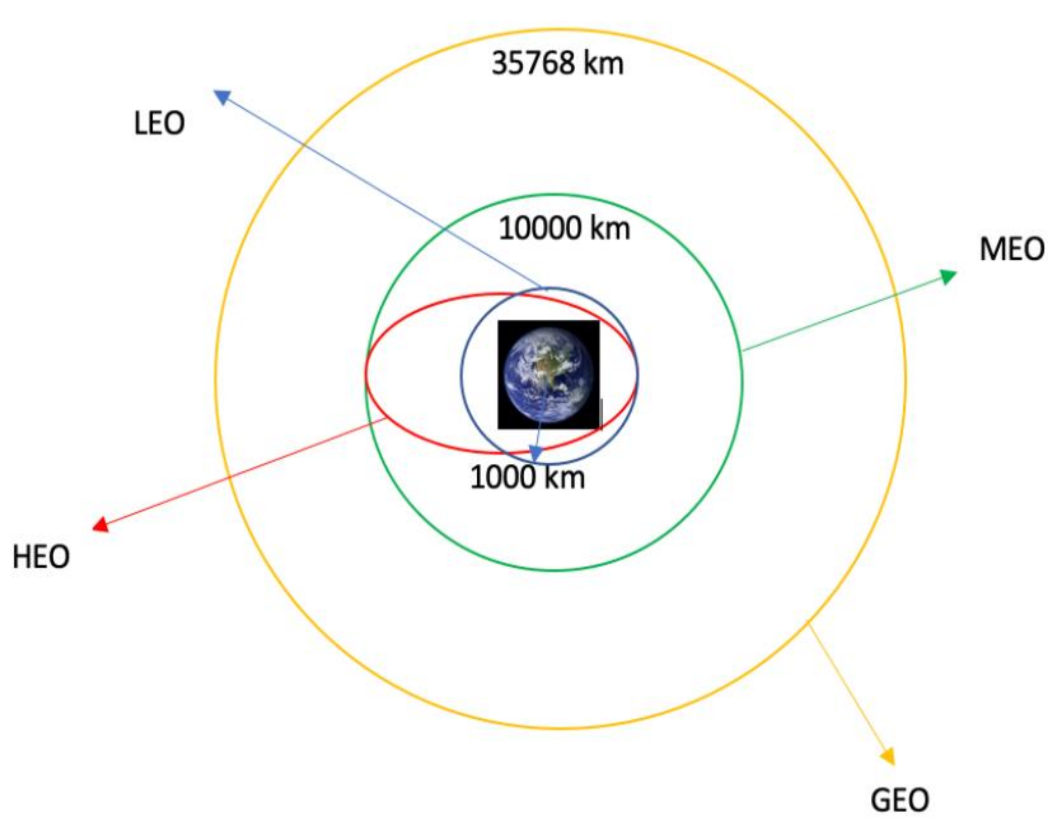
```
df.to_csv('spacex_web_scraped.csv', index=False)
```

https://github.com/Lo0815/IBM-Data-Science-Capstone/blob/master/jupyter-labs-webscraping.ipynb

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 frequency of each orbit.
- The "Landing" label was created from the "Result" column and the results were exported to a csv file.

https://github.com/Lo0815/IBM-Data-Science-Capstone/blob/master/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

Scatter Graphs being drawn:

- **Flight Number VS. Payload Mass**
- **Flight Number VS. Launch Site**
- **Payload VS. Launch Site**
- **Orbit VS. Flight Number**
- **Payload VS. Orbit Type**
- **Orbit VS. Payload Mass**

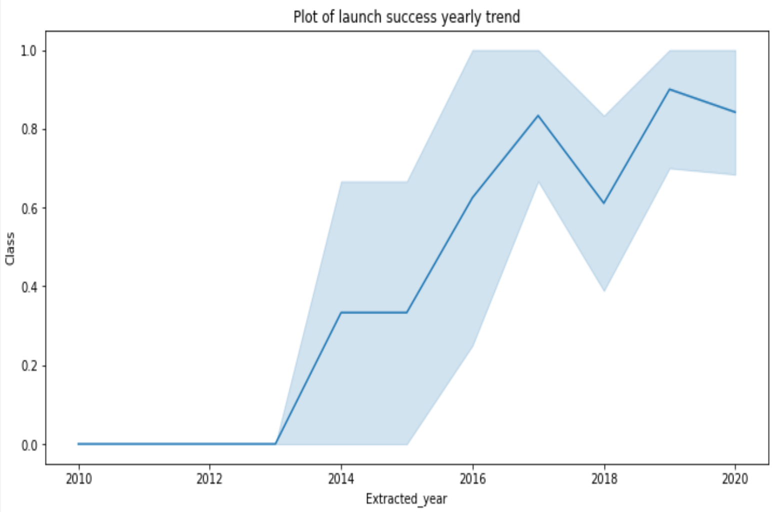
Bar Graph being drawn:

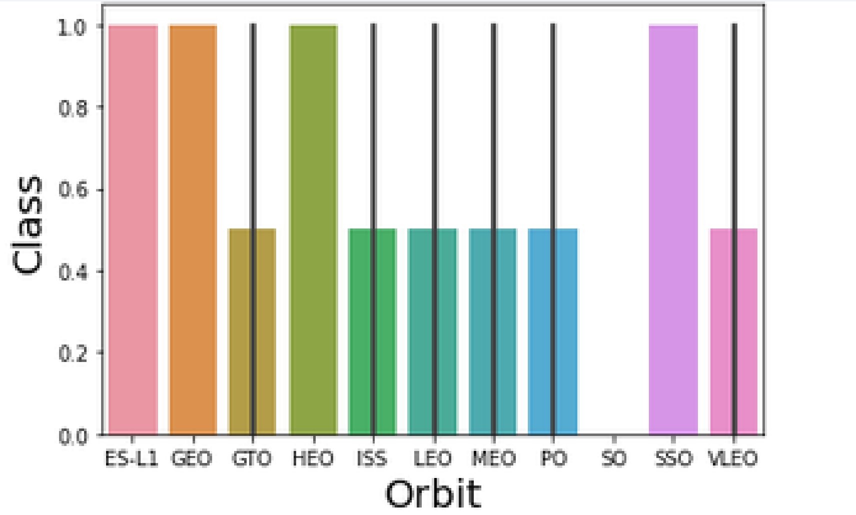
Mean VS. Orbit

Line Graph being drawn:

Success Rate VS. Year

https://github.com/Lo0815/IBM-Data-Science-Capstone/blob/master/jupyter-labs-eda-dataviz.ipynb





EDA with SQL

- We loaded the SpaceX dataset into a into IBM DB2 Database within Jupyter Notebook
- We applied EDA with SQL Python to get insight from the data. We wrote queries to find out for instance:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.

https://github.com/Lo0815/IBM-Data-Science-Capstone/blob/master/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- To visualize the Data we use Folium Maps
- With folium maps we marked launch sites, successful and unsuccessful landings, and important infrastructures such as : railroad, highway, coast and city.
- We also assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success
- With this visualization it is possible to understand which takeoff sites exist and why the takeoff sites were built there. In addition, the successful landings are visualized in relation to the location.
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

https://github.com/Lo0815/IBM-Data-Science-Capstone/blob/master/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- The dashboard includes a pie chart and a scatter plot
- With the pie charts we want showing the total launches by a certain sites
- With the plotted scatter graph we are showing the relationship with Outcome and Payload Mass (Kg) for the different booster version

Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

https://github.com/Lo0815/IBM-Data-Science-Capstone/blob/master/spacex_dash_app.py

Predictive Analysis (Classification)

Model Building

Load our dataset into NumPy and Pandas
Transform Data using StandardScaler

Model Building

Split our data into training and test data sets
Check how many test samples we have

Model Building

Set the parameters to GridSearchCV . Use GridSearch on SVM, KNN, Dtree, Log.Reg

Evaluation & Improving

Check accuracy for each model Get tuned hyperparameters for each type of algorithms

Find the best model

Plot the confusions Matrix and find the best model

https://github.com/Lo0815/IBM-Data-Science-Capstone/blob/master/SpaceX_Machine%20Learning%20Prediction.ipynb

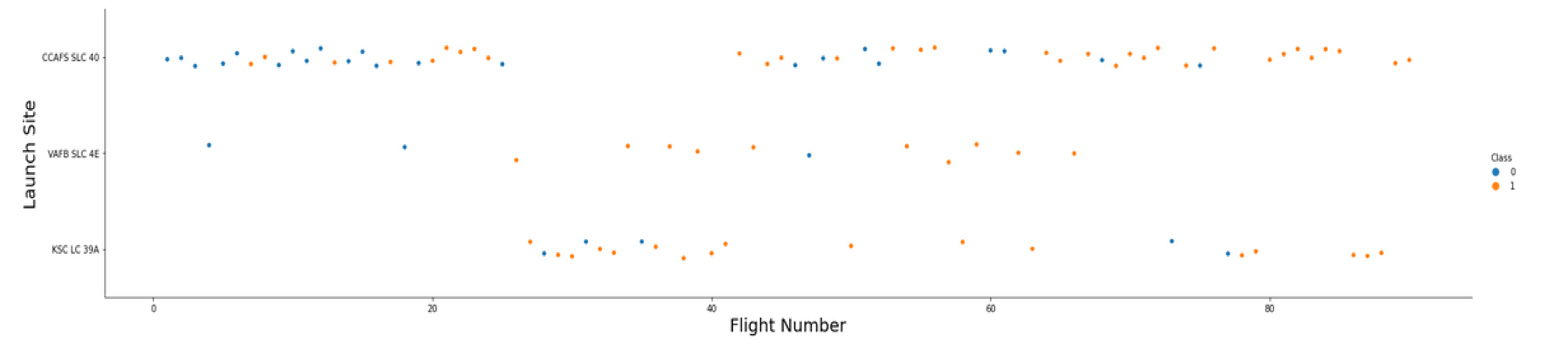
Results

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

Section 2

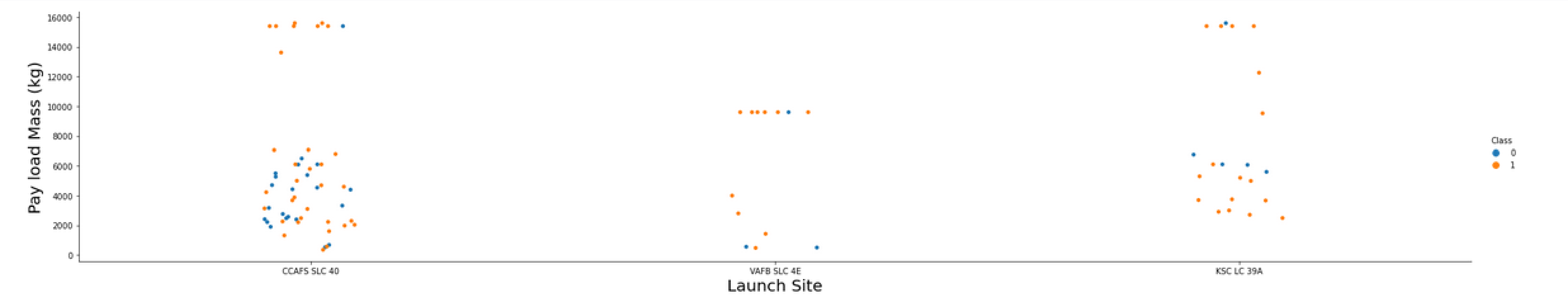
Insights drawn from EDA

Flight Number vs. Launch Site



The graph shows that there was an increase in the success rate over time (indicated in flight number). Probably around flight 20 there was a major breakthrough in technological adaptations that significantly increased the success rate.

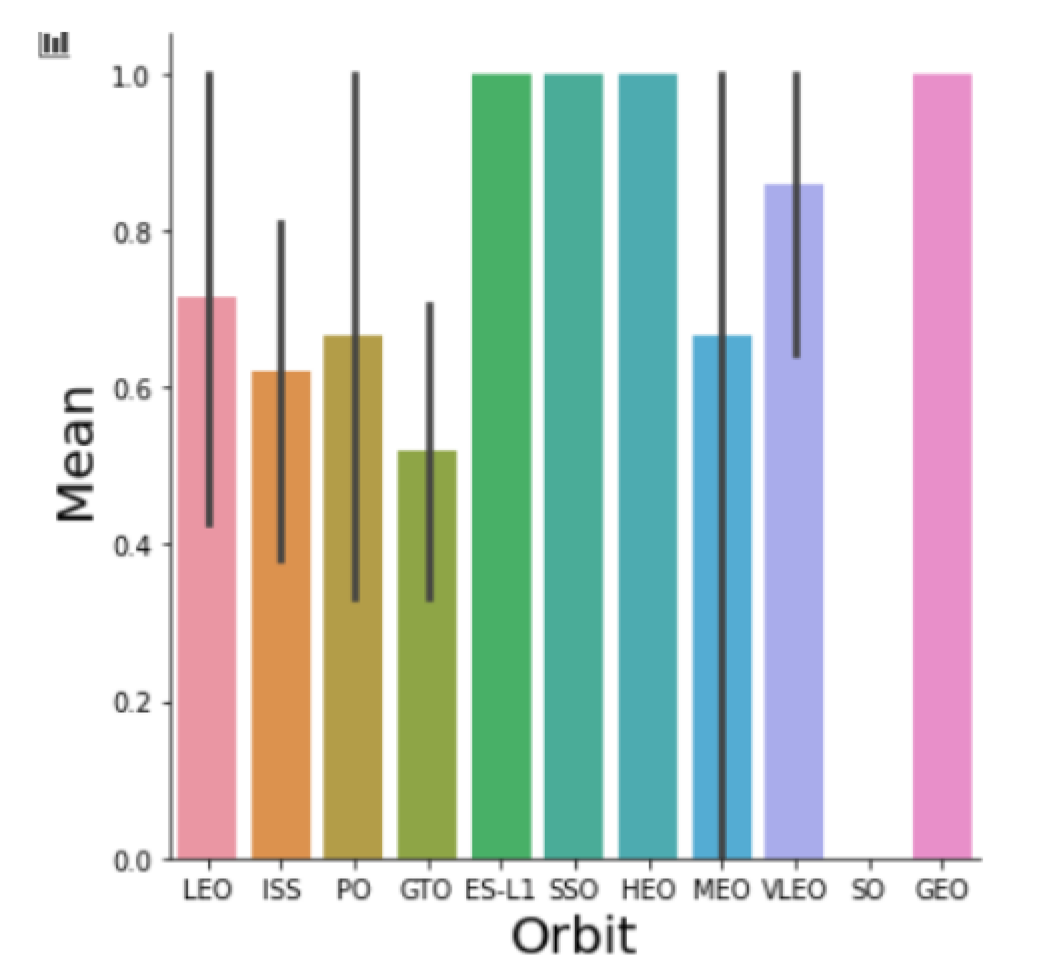
Payload vs. Launch Site



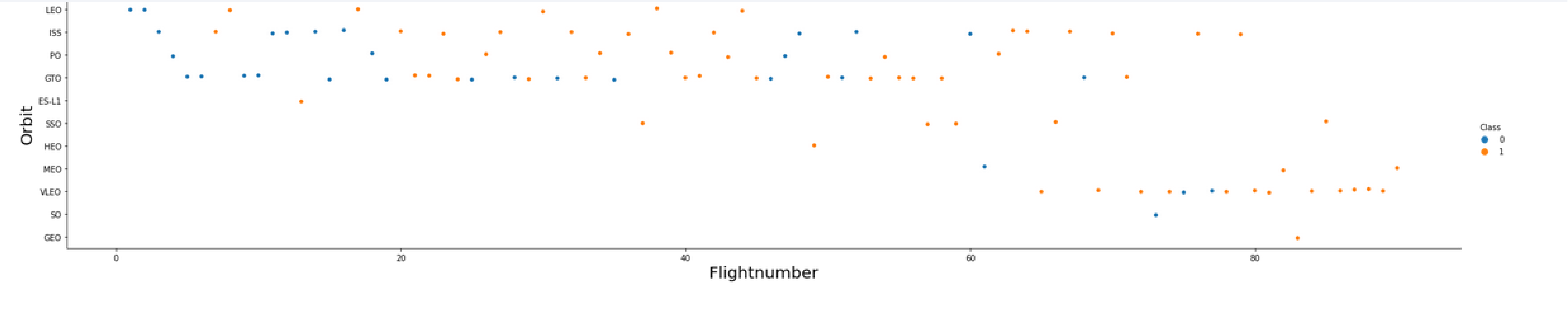
From the graph, it can be seen that the larger the payload mass for the CCAFS SLC 40 launch site, the higher the success rate for the rocket.
Based on this graph, no clear pattern can be seen to make a decision whether the launch site is depends on the payload mass for a successful launch.

Success Rate vs. Orbit Type

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

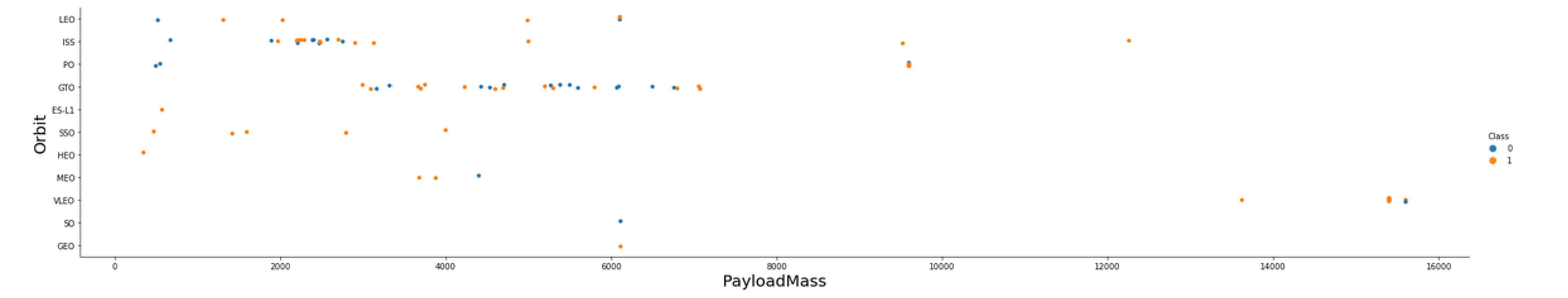


Flight Number vs. Orbit Type



The graph shows the number of flights versus the orbit type. It can be seen that success in LEO orbit is related to the number of flights, while in GTO orbit there is no relationship between the number of flights and the orbit type

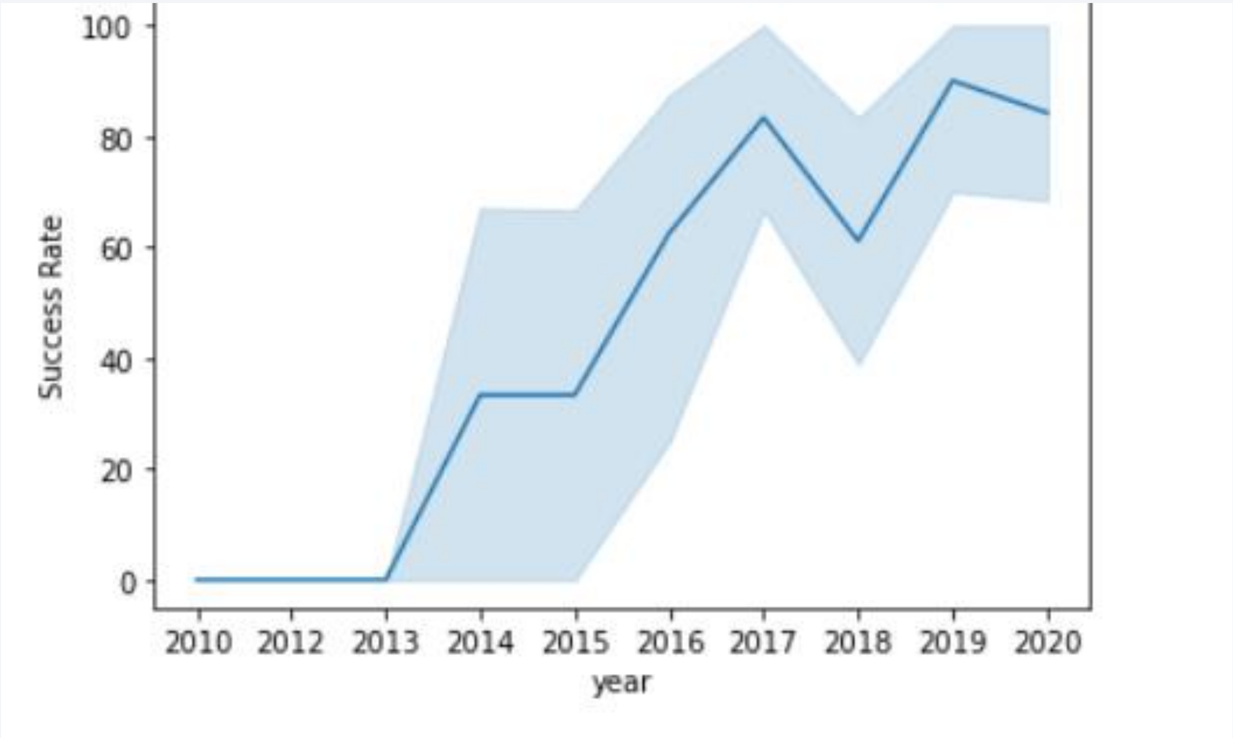
Payload vs. Orbit Type



There seems to be a correlation between the LEO /SSO orbit and the payload. We can also observe that the PO, LEO and ISS orbits with heavy payloads have more successful landings.

Launch Success Yearly Trend

you can observe that the sucess rate since 2013 kept increasing till 2020



All Launch Site Names

```
%sql select DISTINCT LAUNCH_SITE from SPACEXDATASET
```

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

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

Launch Site Names Begin with 'CCA'

```
%sql select * from SPACEXDATASET where launch_site like 'CCA%' limit 5
```

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

```
%sql select sum(payload_mass__kg_) as sum from SPACEXDATASET where customer like 'NASA (CRS)'
```

SUM
45596

This query sums the total payload mass in kg where NASA was the customer.

Average Payload Mass by F9 v1.1

```
%sql select avg(payload_mass__kg_) as Average from SPACEXDATASET where booster_version like 'F9 v1.1%'
```

average
2534

This query calculates the average payload mass or launches which used booster version F9 v1.1

First Successful Ground Landing Date

```
%sql select min(date) as Date from SPACEXDATASET where mission_outcome like 'Success'
```

DATE
2010-06-04

This query returns the first successful ground pad landing date.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select booster_version from SPACEXDATASET where (mission_outcome like 'Success')
AND (payload_mass__kg_ BETWEEN 4000 AND 6000) AND (landing__outcome like 'Success (drone ship)')
```

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

With this query, we filter for boosters that have successfully landed on a drone ship to also identify a successful landing with a payload mass greater than 4000 but less than 6000.

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT mission_outcome, count(*) as Count FROM SPACEXDATASET GROUP by mission_outcome ORDER BY mission_outcome
```

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

This query provides the total number of the
individual
mission outcome.
SpaceX appears to achieve its mission
objective nearly 99% of the time.
From this, it can be deduced that most
landing
Failures are intentional.

Boosters Carried Maximum Payload

```
maxm = %sql select max(payload_mass__kg_) from SPACEXDATASET
maxv = maxm[0][0]
%sql select booster_version from SPACEXDATASET where
payload_mass__kg_=(select max(payload_mass__kg_) from SPACEXDATASET)
```

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

The query indicates that the payload mass correlates with the booster version used.

2015 Launch Records

```
%sql select MONTHNAME(DATE) as Month, landing__outcome, booster_version, launch_site
from SPACEXDATASET where DATE like '2015%' AND landing__outcome like 'Failure (drone ship)'
```

MONTH	landing__outcome	booster_version	launch_site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

This Query displays the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015

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

```
%sql select landing__outcome, count(*) as count from SPACEXDATASET
where Date >= '2010-06-04' AND Date <= '2017-03-20'
GROUP by landing__outcome ORDER BY count Desc
```

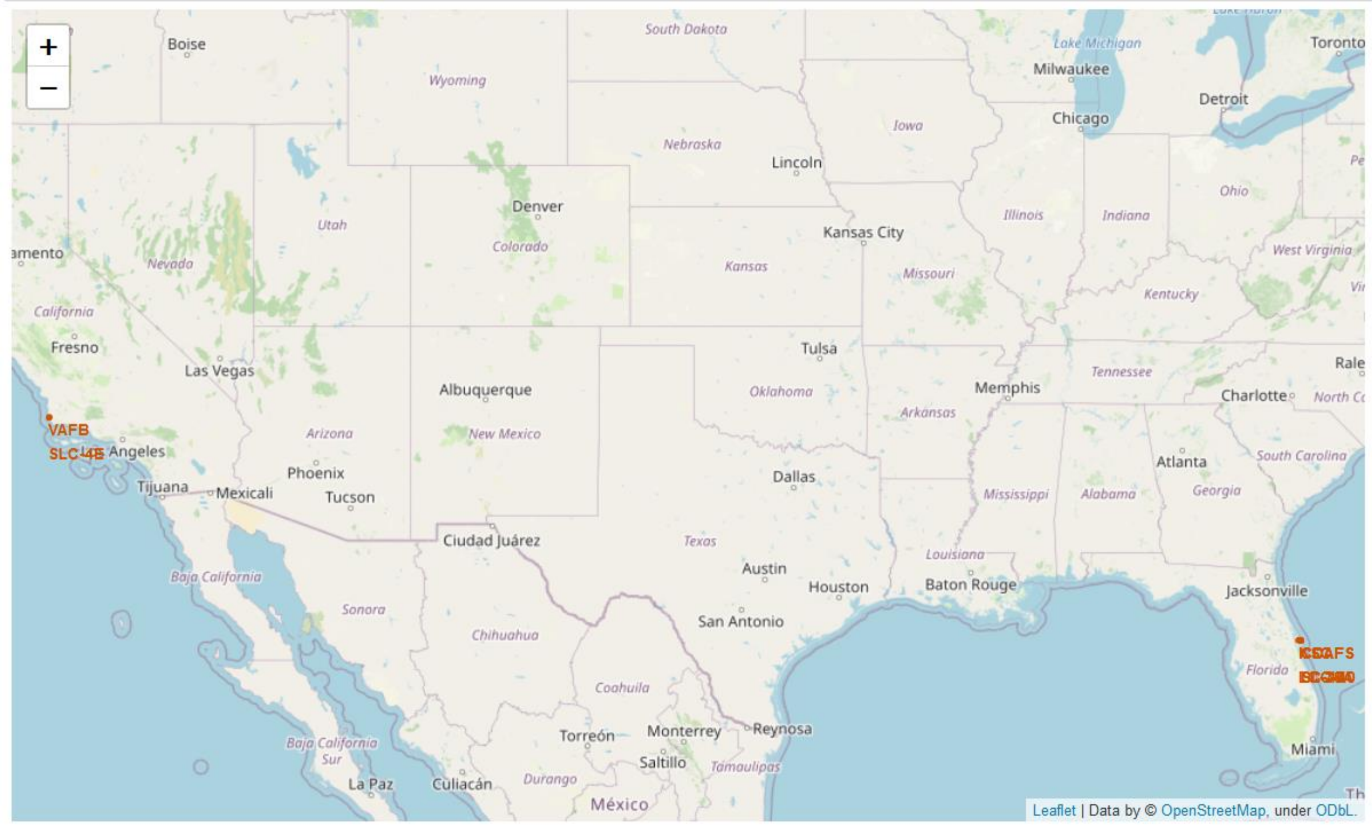
landing__outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

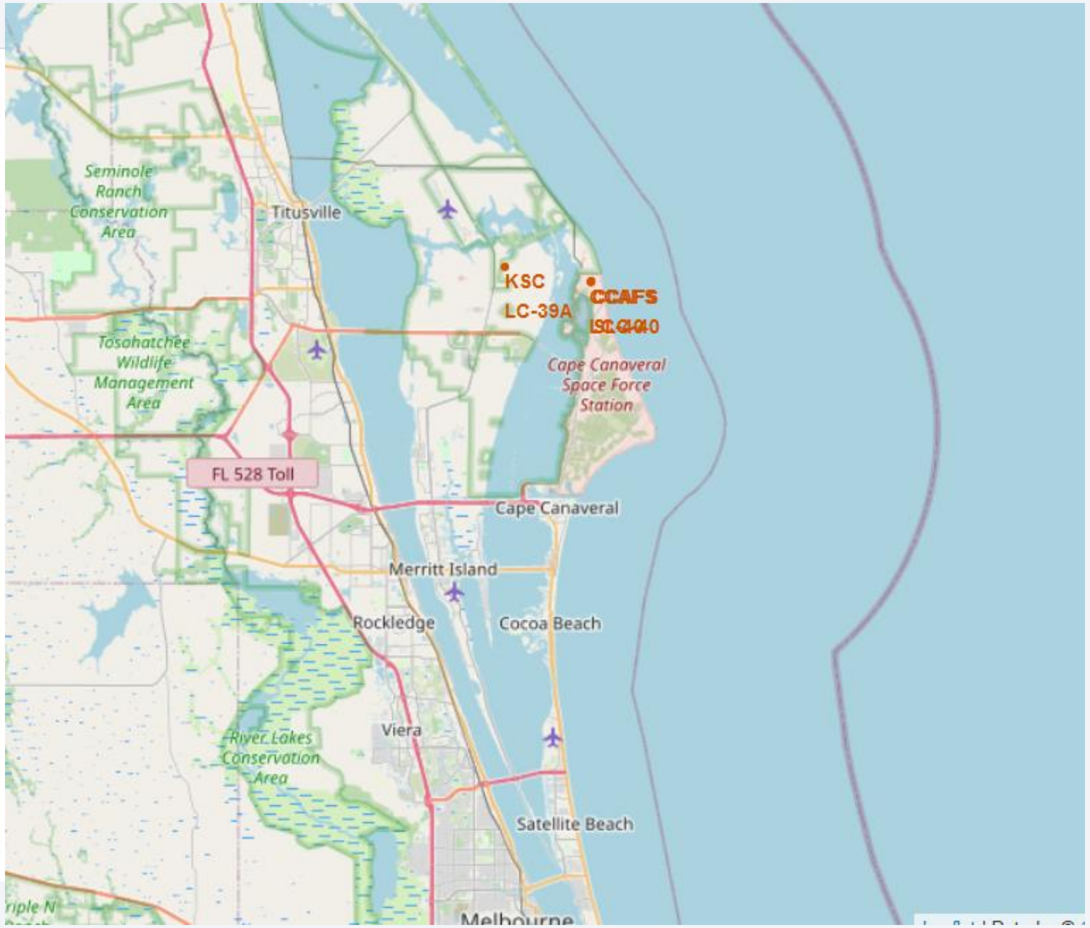
This Query ranks the count of successful landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

Section 3

Launch Sites Proximities Analysis

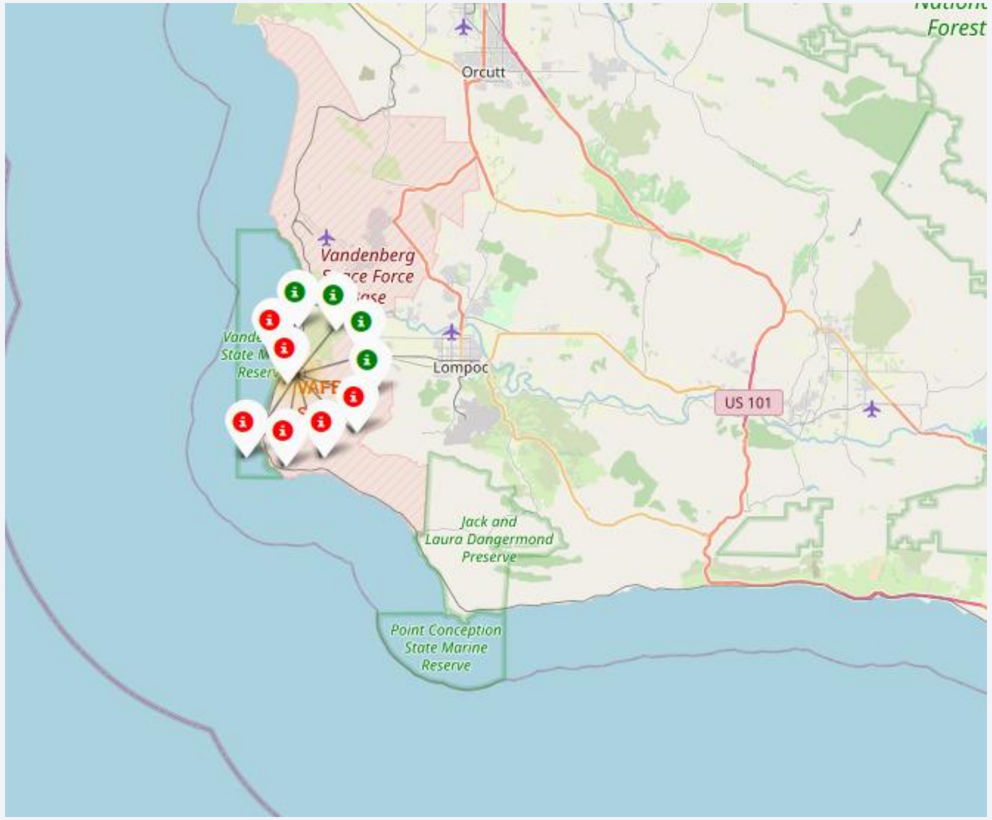
All launch sites global map markers

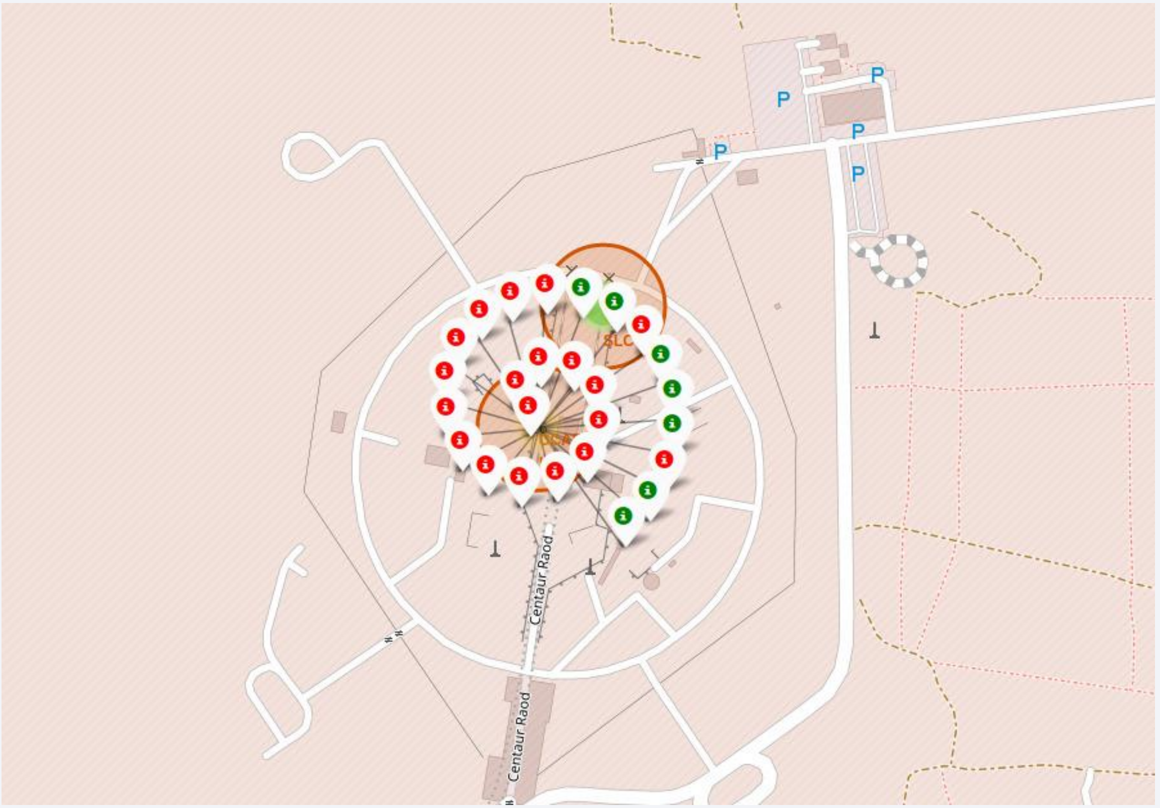




The left map shows all launch sites within USA. The right map shows the two Florida launch sites since they are very close to each other. All launch sites are near the ocean for the case of a crash

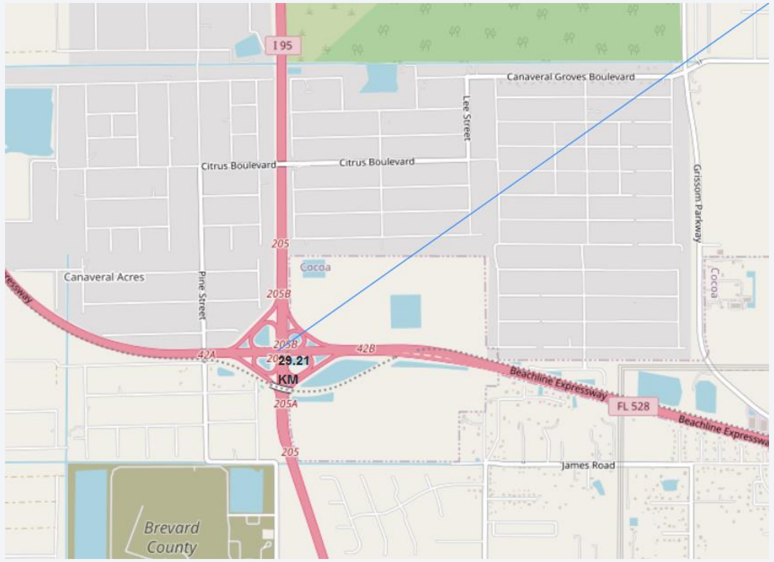
Markers showing launch sites with color labels

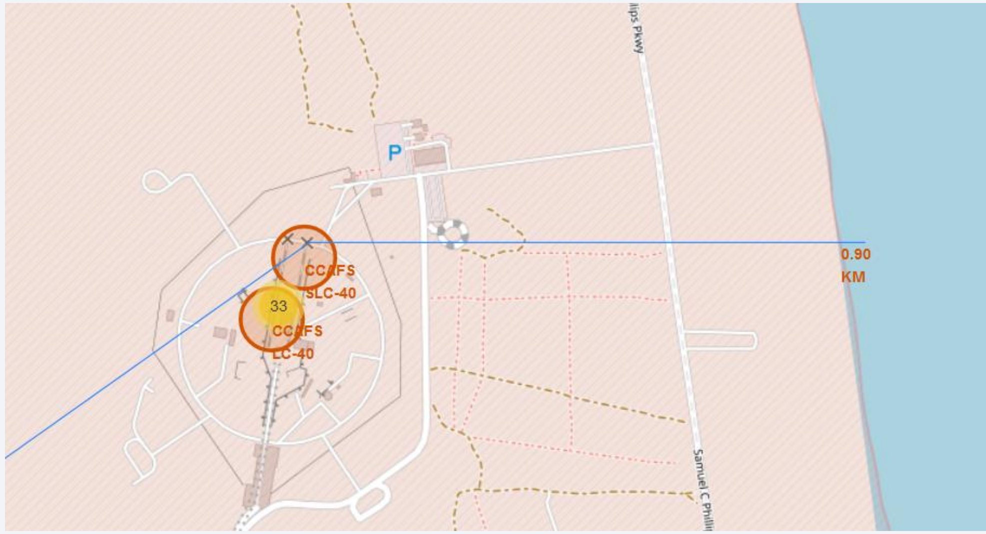




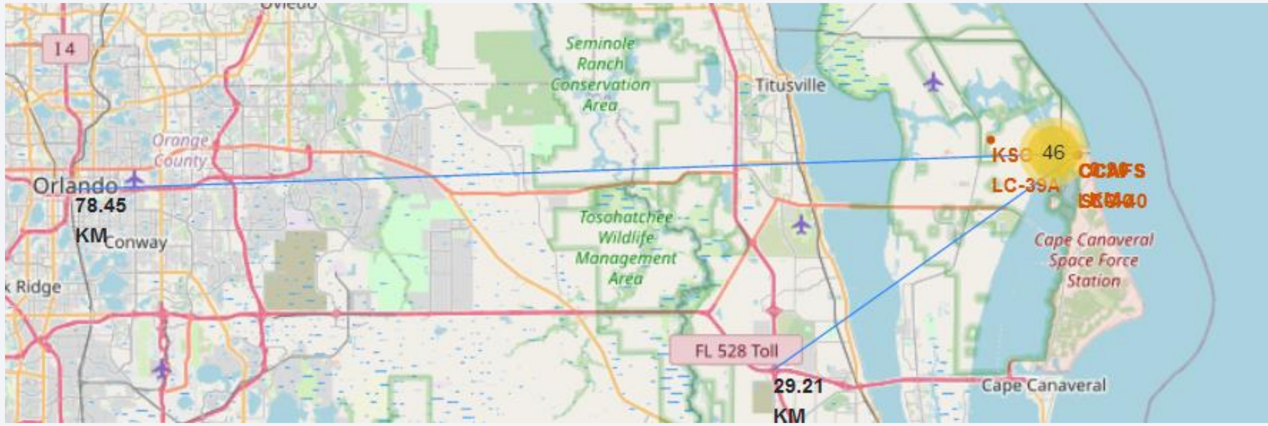
Green Marker shows successful Launches and Red Marker shows Failures

Launch site distance to important Landmarks





The map shows the distance of the launch sites from important infrastructures such as railroad lines, highways, city centers and the sea.

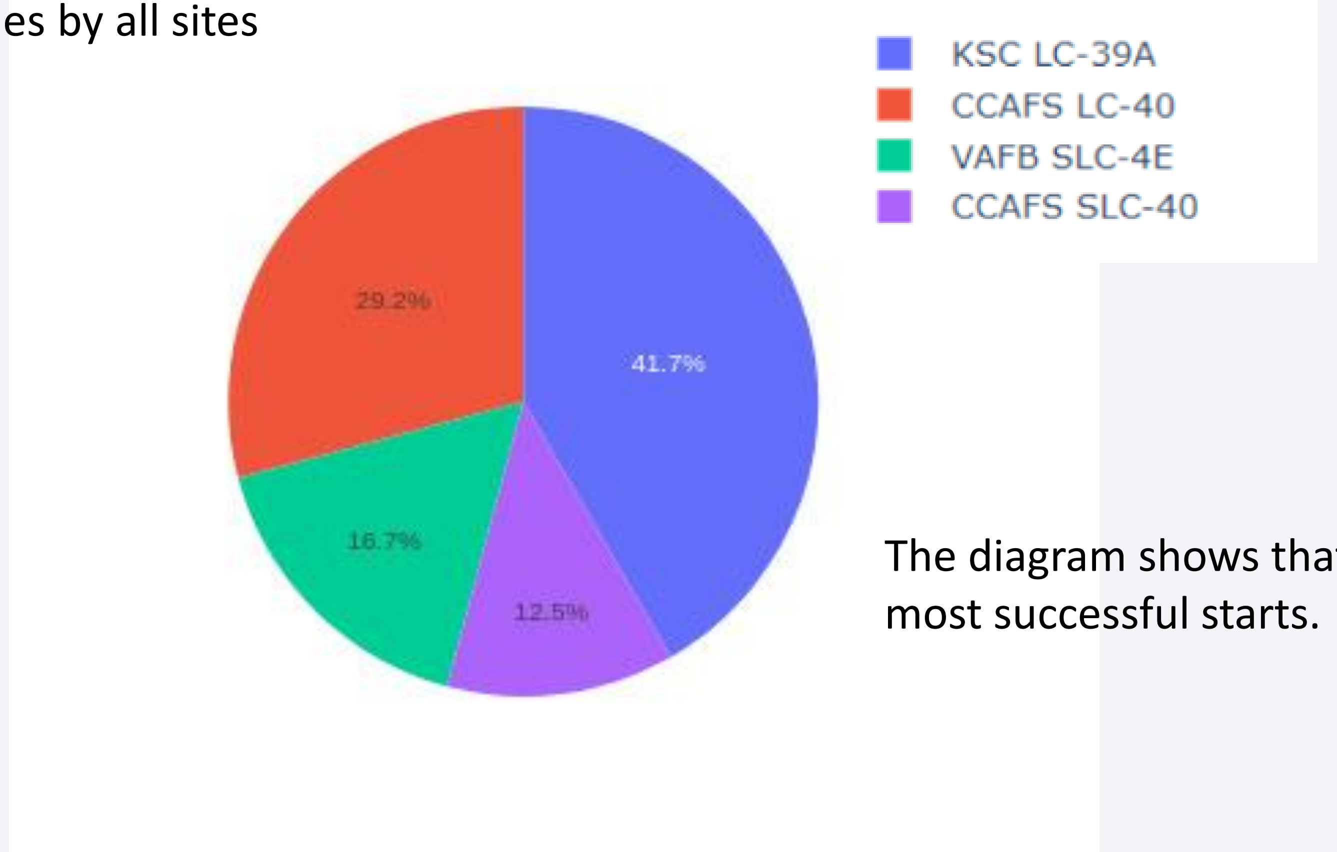


Section 4

Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site

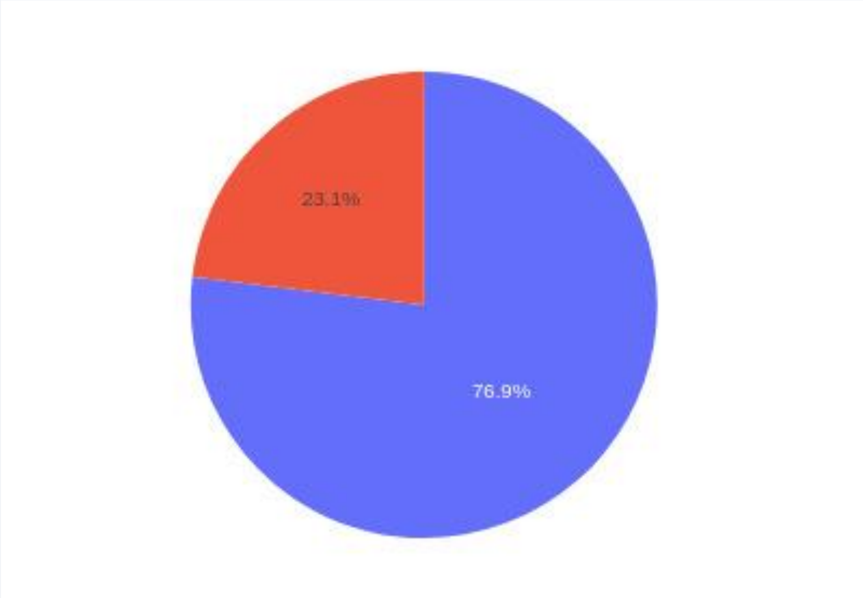
Total success Launches by all sites



The diagram shows that the KSC-LC39A has the most successful starts.

KSC LC-39A Success Rate Launch Site

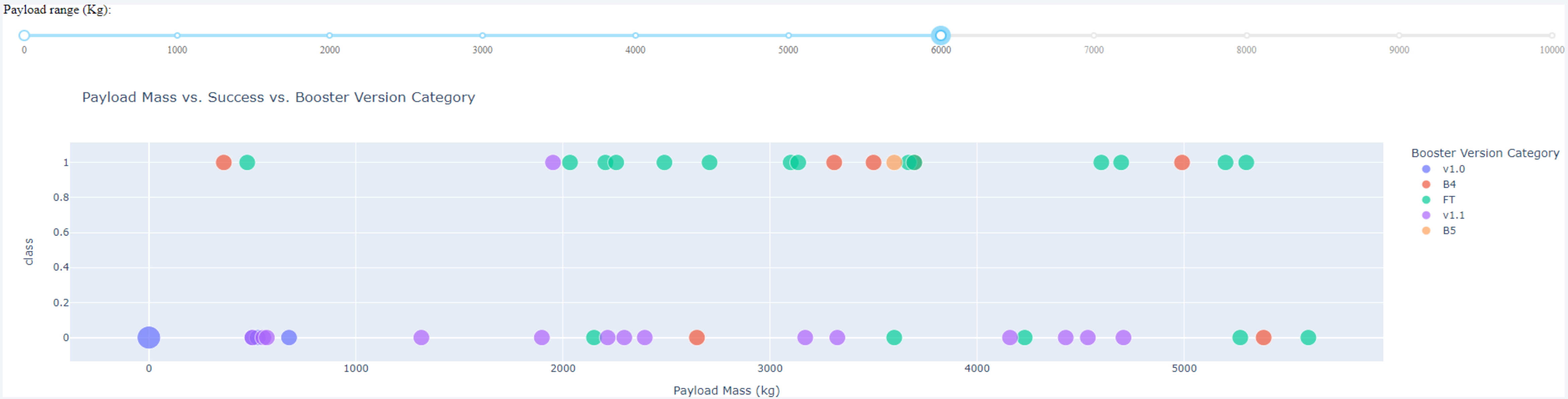
blue = success



KSC LC-39A has 10 successful landings and 3 failed landings

Scatter plot of Payload vs Launch Outcome for all sites

Class indicates 1 for successful landing and 0 for failure.

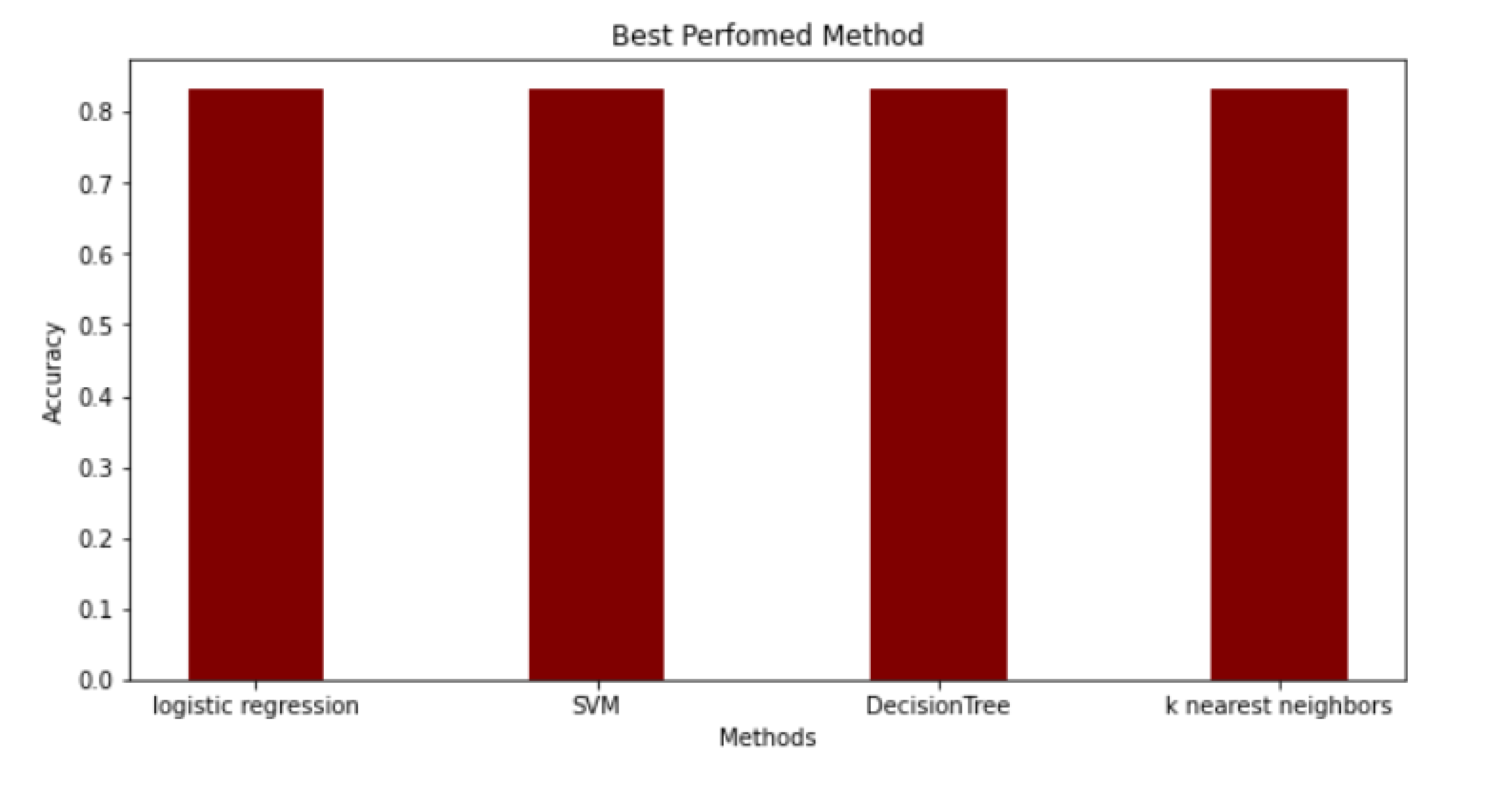


We can see the success rates for low weighted payloads is higher than the heavy weighted payload

Section 5

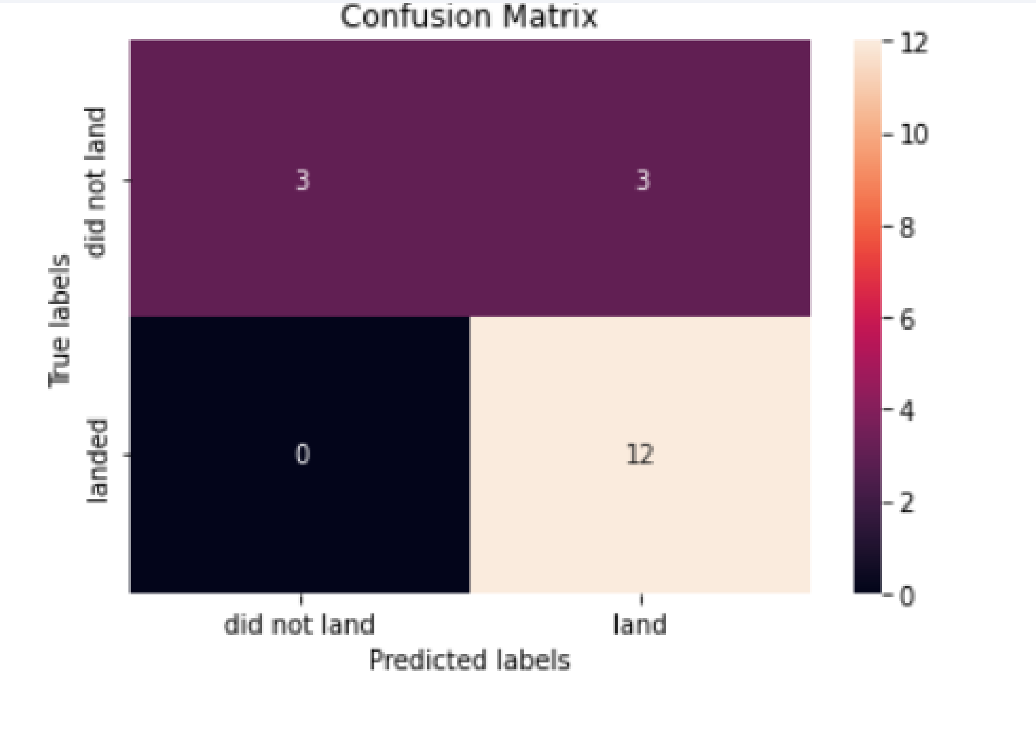
Predictive Analysis (Classification)

Classification Accuracy



- As can be seen all models had virtually the same accuracy on the test set at 83.33% accuracy.(test size = 18 examples)
- More data is needed to get better results.
- Also, other models like DNN can be tested

Confusion Matrix



- Since all models performed the same for the test set, the confusion matrix is the same across all models
- We see that the major problem is false positives.

Conclusions

- The goal was to predict when Stage 1 will successfully land to save ~$100 million USD
- 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
- With more data and the use of other models such as DNN, better results could be obtained.
- Based on the data, it can be seen that spacex has undergone a technological process over the years that has improved the results. From this it can be concluded that SpaceY must also go through this process.

Thank you!