IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Shubham Gupta
20-09-2021

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

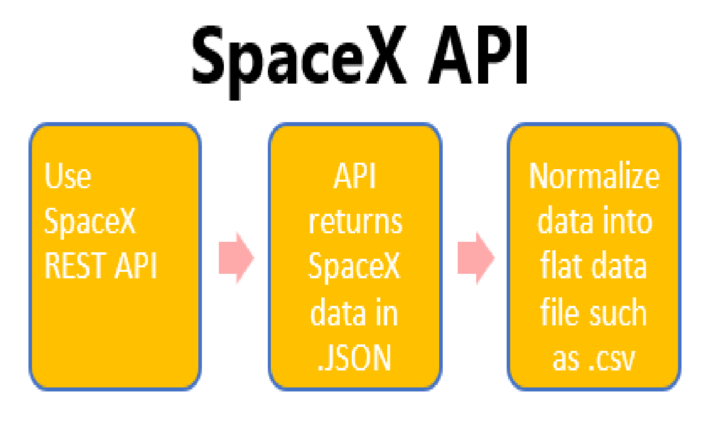
Methodology

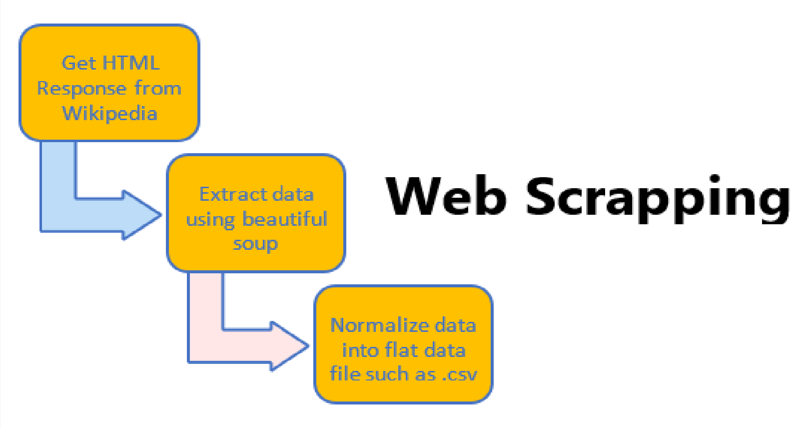
Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Web Scrapping using Beautiful Soup from Wikipedia
- Perform data wrangling
  - One Hot Encoding data fields for Machine Learning and feature selection
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Split data into training and testing set, tune and estimate best model using Gridsearchcv , evaluate classification models to determine best model with highest accuracy.

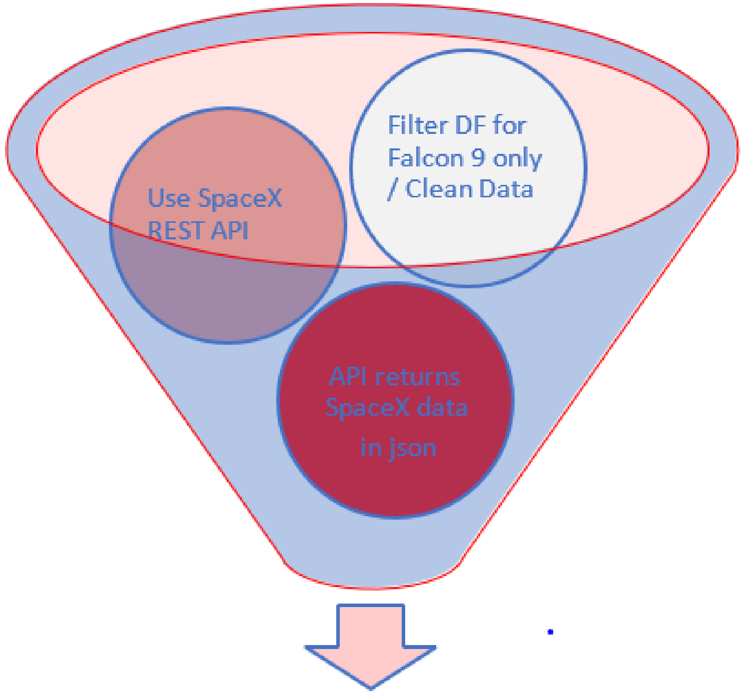
Data Collection

- SpaceX launch data that is gathered from the SpaceX REST API.
- This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
- Our goal is to use this data to predict whether SpaceX will attempt to land a rocket or not.
- The SpaceX REST API endpoints, or URL, starts with https://api.spacexdata.com/v4/..
- Another popular data source for obtaining Falcon 9 Launch data is web scraping Wikipedia using BeautifulSoup.





Data Collection – SpaceX API



Normalize data into flat data file such as .csv

[GitHub Url for Data Collection API]

1 .Getting Response from API

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

2. Converting Response to a .json file

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

3. Apply custom functions to clean data

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

```
getBoosterVersion(data)
```

4. Assign list to dictionary then dataframe

```
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}
```

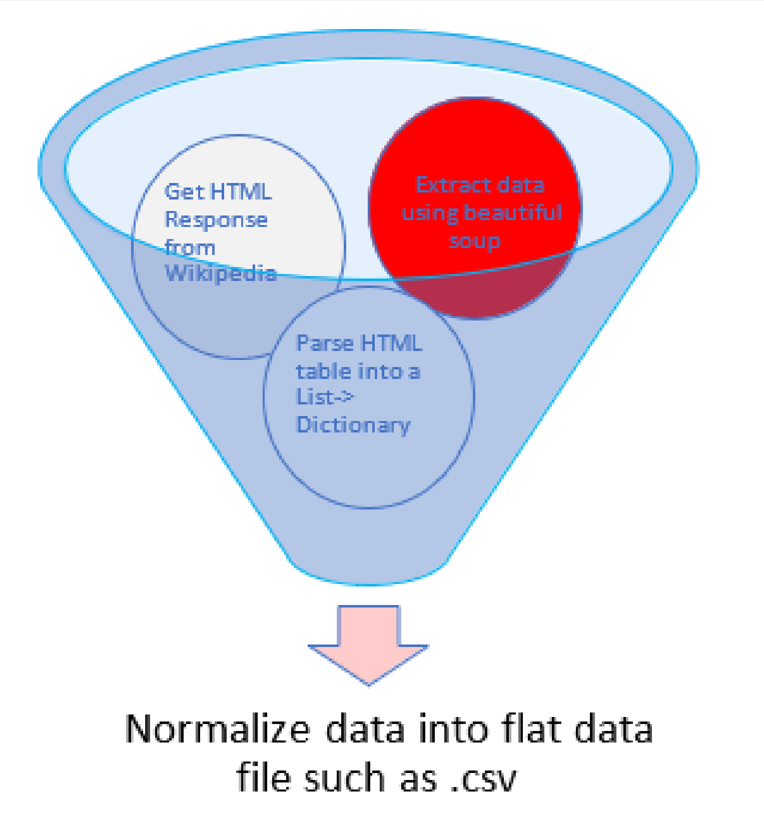
```
df = pd.DataFrame.from_dict(launch_dict)
```

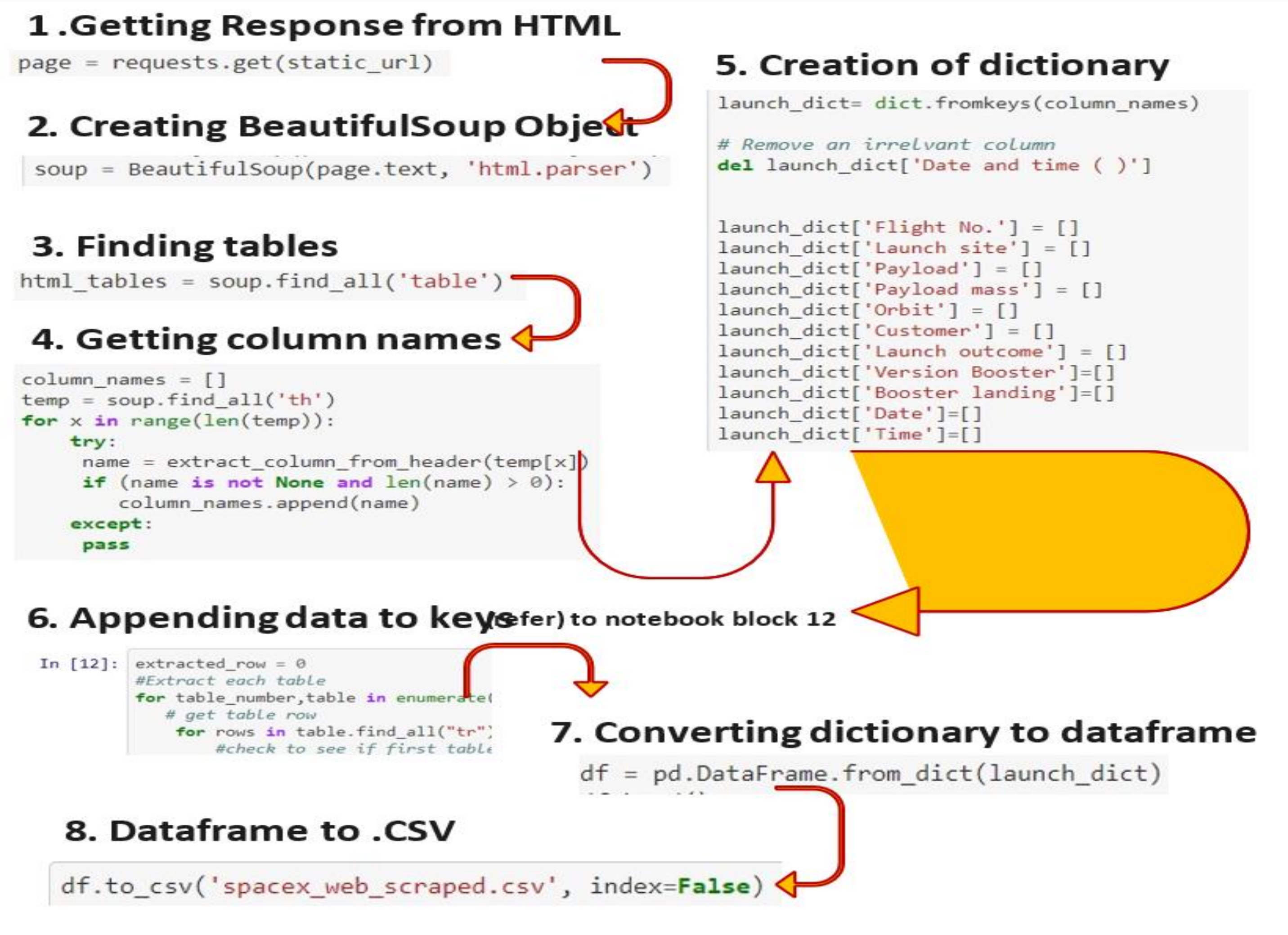
5. Filter dataframe and export to flat file (.csv)

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

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

Data Collection - Scraping

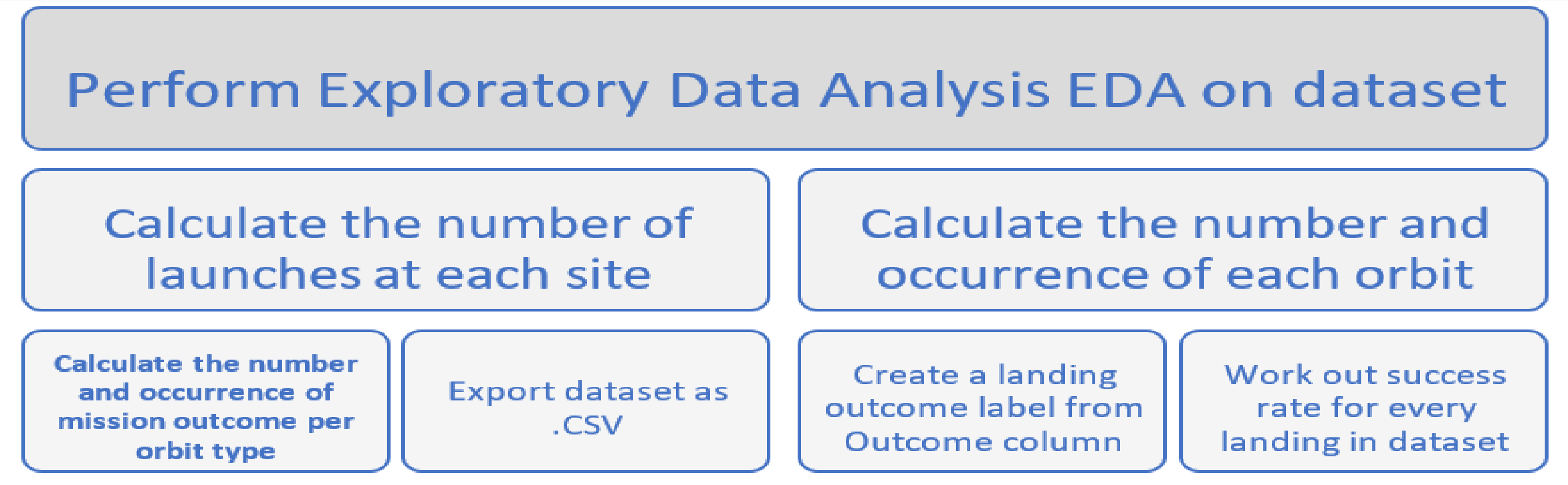




- [GitHub link Data Collection with Web Scraping]()

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.
- We mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.



[GitHub Link Data wrangling]

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
  - Orbit VS. Payload Mass
  - Payload VS. OrbitType

Scatter plots' primary uses are to observe and show relationships between two numeric variables. The dots in a scatter plot not only report the values of individual data points, but also patterns when the data are taken as a whole.

Bar Graph being drawn:

- Mean VS. Orbit

Bar graphs are used to compare things between different groups or to track changes over time. However, when trying to measure change over time, bar graphs are best when the changes are larger.

Line Graph being drawn:

- Success Rate VS. Year

Line graphs are used to track changes over short and long periods of time. When smaller changes exist, line graphs are better to use than bar graphs. Line graphs can also be used to compare changes over the same period of time for more than one group.

GitHub Link Exploring and Preparing Data

EDA with SQL

- Performed SQL queries to gather information about the dataset.
  - ➢ Displaying the names of the unique launch sites in the space mission
  - ➢ Displaying 5 records where launch sites begin with the string 'KSC'
  - ➢ Displaying the total payload mass carried by boosters launched by NASA (CRS)
  - ➢ Displaying average payload mass carried by booster version F9 v1.1
  - ➢ Listing the date where the successful landing outcome in drone ship was achieved.
  - ➢ Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
  - ➢ Listing the total number of successful and failure mission outcomes
  - ➢ Listing the names of the booster_versions which have carried the maximum payload mass.
  - ➢ Listing 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_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

[GitHub Link EDA with SQL]

Build an Interactive Map with Folium

- To visualize the Launch Data into an interactive map. We 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.
- We assigned the dataframe launch_outcomes(failures, successes) to classes 0 and 1 with Green and Red markers on the map in a MarkerCluster()
- Using Haversine's formula we calculated the distance from the Launch Site to various landmarks to find various trends about what is around the Launch Site to measure patterns. Lines are drawn on the map to measure distance to landmarks
- Example of some trends in which the Launch Site is situated in.
  - Are launch sites in close proximity to railways? No
  - Are launch sites in close proximity to highways? No
  - Are launch sites in close proximity to coastline? Yes
  - Do launch sites keep certain distance away from cities? Yes
- GitHub Link Visual Analytics with Folium

Build a Dashboard with Plotly Dash

- Dropdown is created to select All launch site or any particular launch site.
- Based on dropdown selection pie chart represents success rate of that particular site or all sites.
- Dynamic scatter plot between class and payload mass which also changes with dropdown. Payload mass range selector to narrow down the range. Color label is present to distinguish booster version.
- Dashboard can be hosted as website, which is useful in sharing the interactive report with anyone with the internet access.

Pie Chart showing the total launches by a certain site/all sites.

Scatter Graph showing the relationship with Outcome and Payload Mass (Kg) for the different Booster Versions

- It shows the relationship between two variables.
- It is the best method to show you a non-linear pattern.
- The range of data flow, i.e. maximum and minimum value, can be determined.
- Observation and reading are straightforward.

GitHub Link Dashboard with polydash

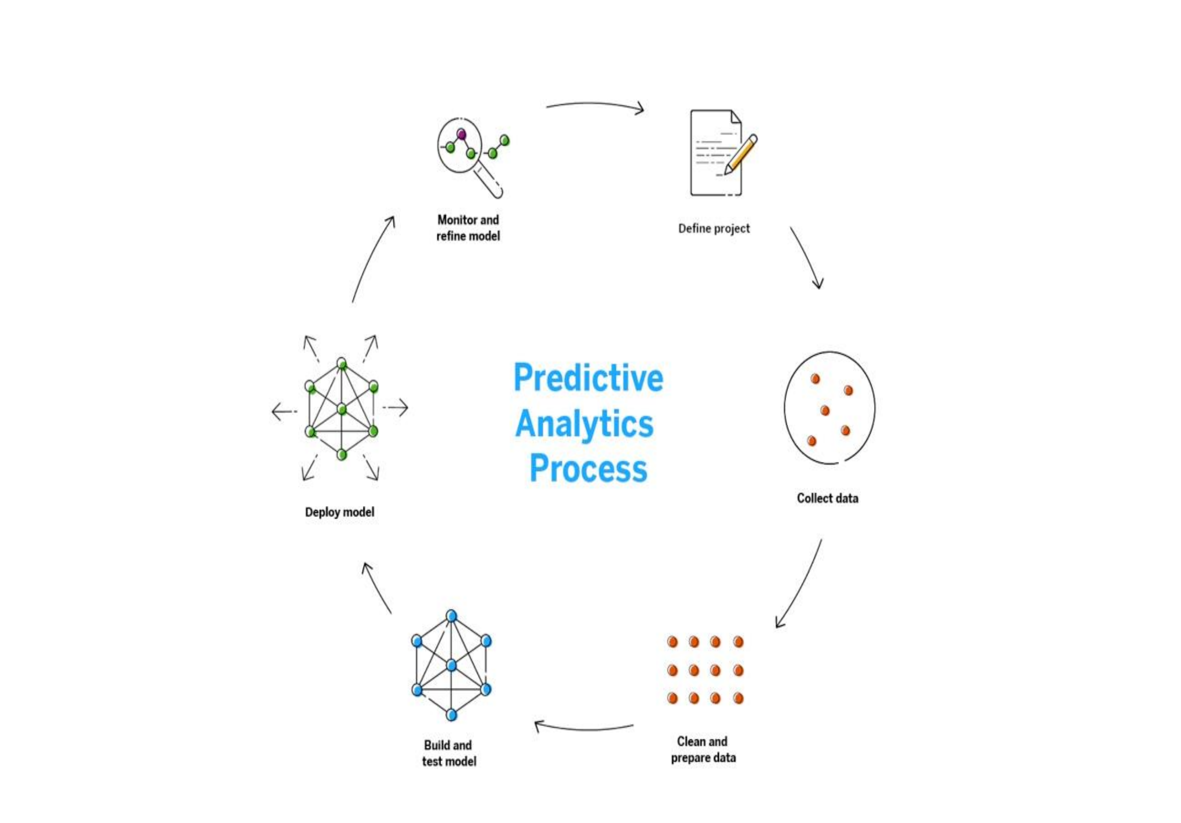
Predictive Analysis (Classification)

1. BUILDING MODEL
   - Load 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 GridSearchCV objects and train our dataset.

2. EVALUATING MODEL
   - Check accuracy for each model
   - Get tuned hyperparameters for each type of algorithms
   - Plot Confusion Matrix

3. IMPROVING MODEL
   - Feature Engineering
   - Algorithm Tuning

4. 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.



GitHub Link Machine Learning Prediction

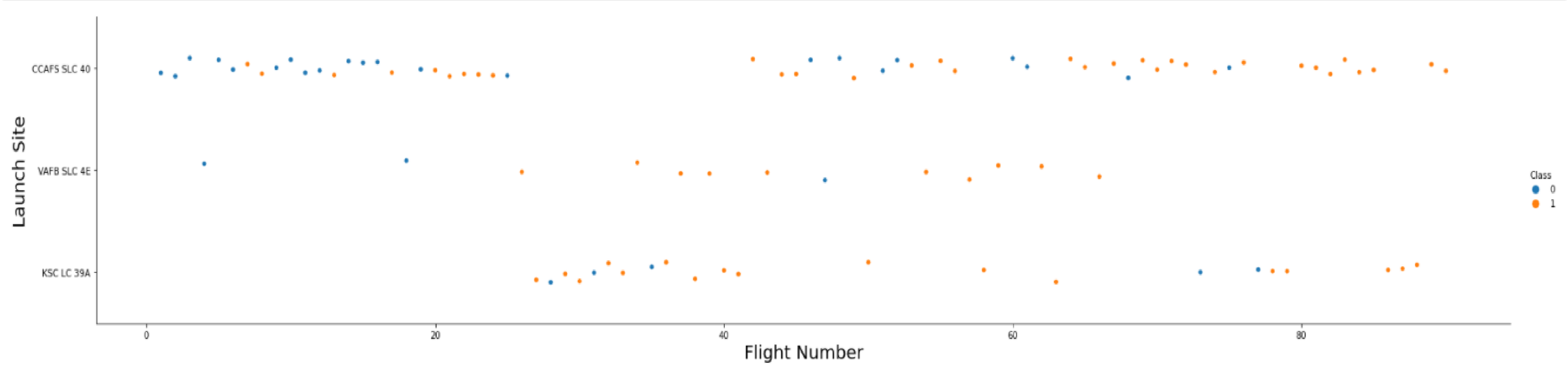
Results

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

Section 2

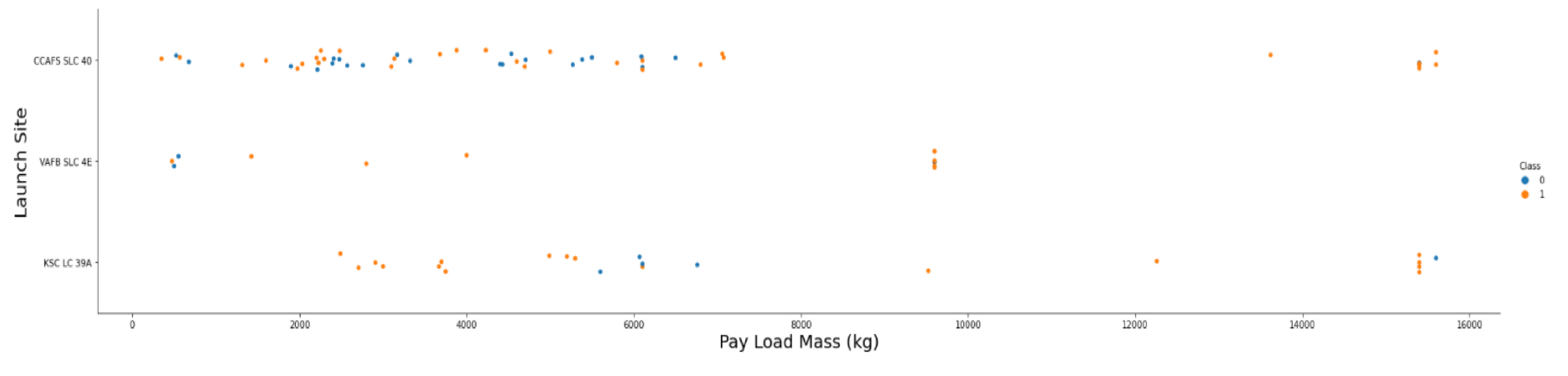
Insights drawn from EDA

Flight Number vs. Launch Site



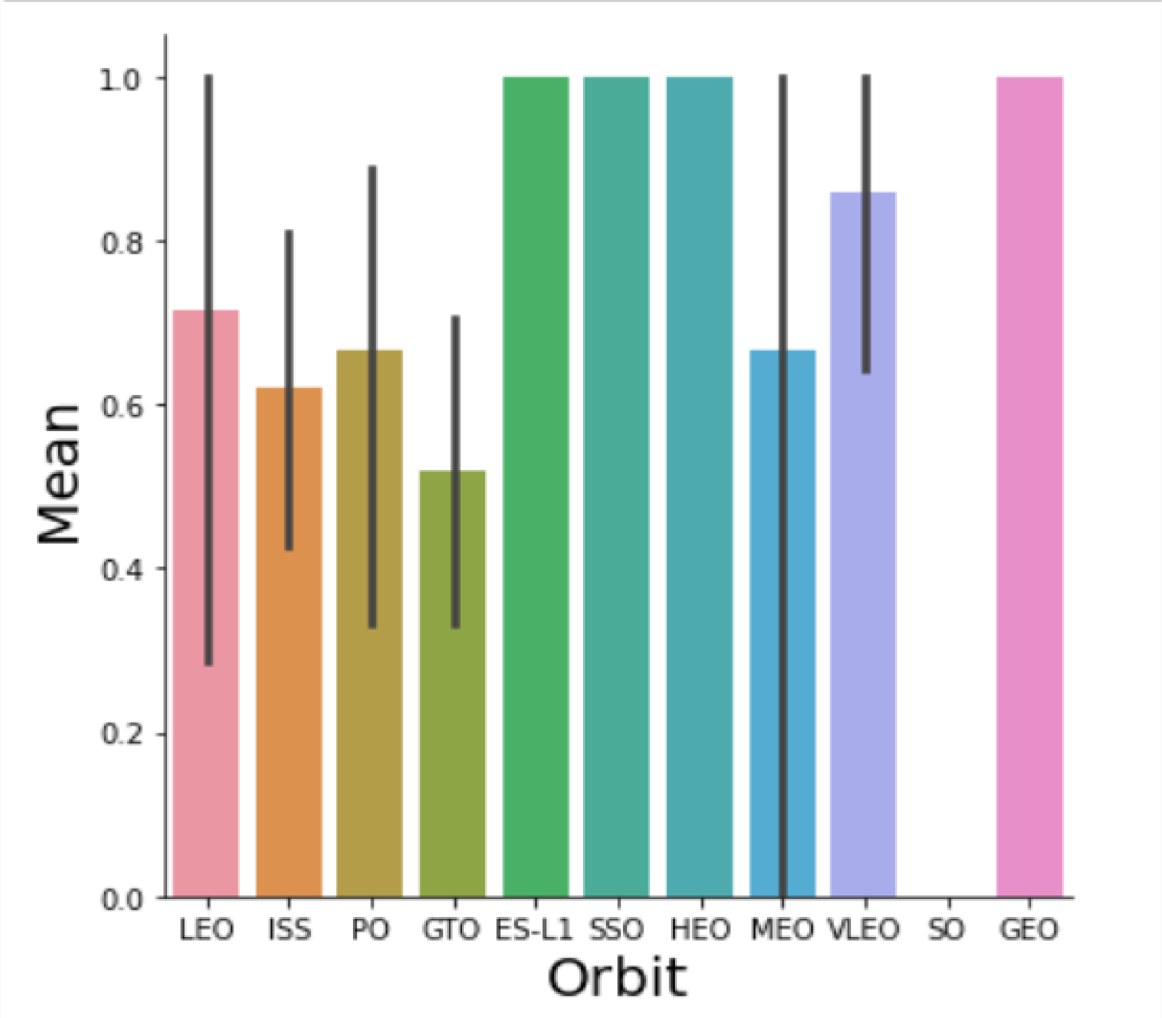
The more amount of flights at a launch site the greater the success rate at a launch site.

Payload vs. Launch Site



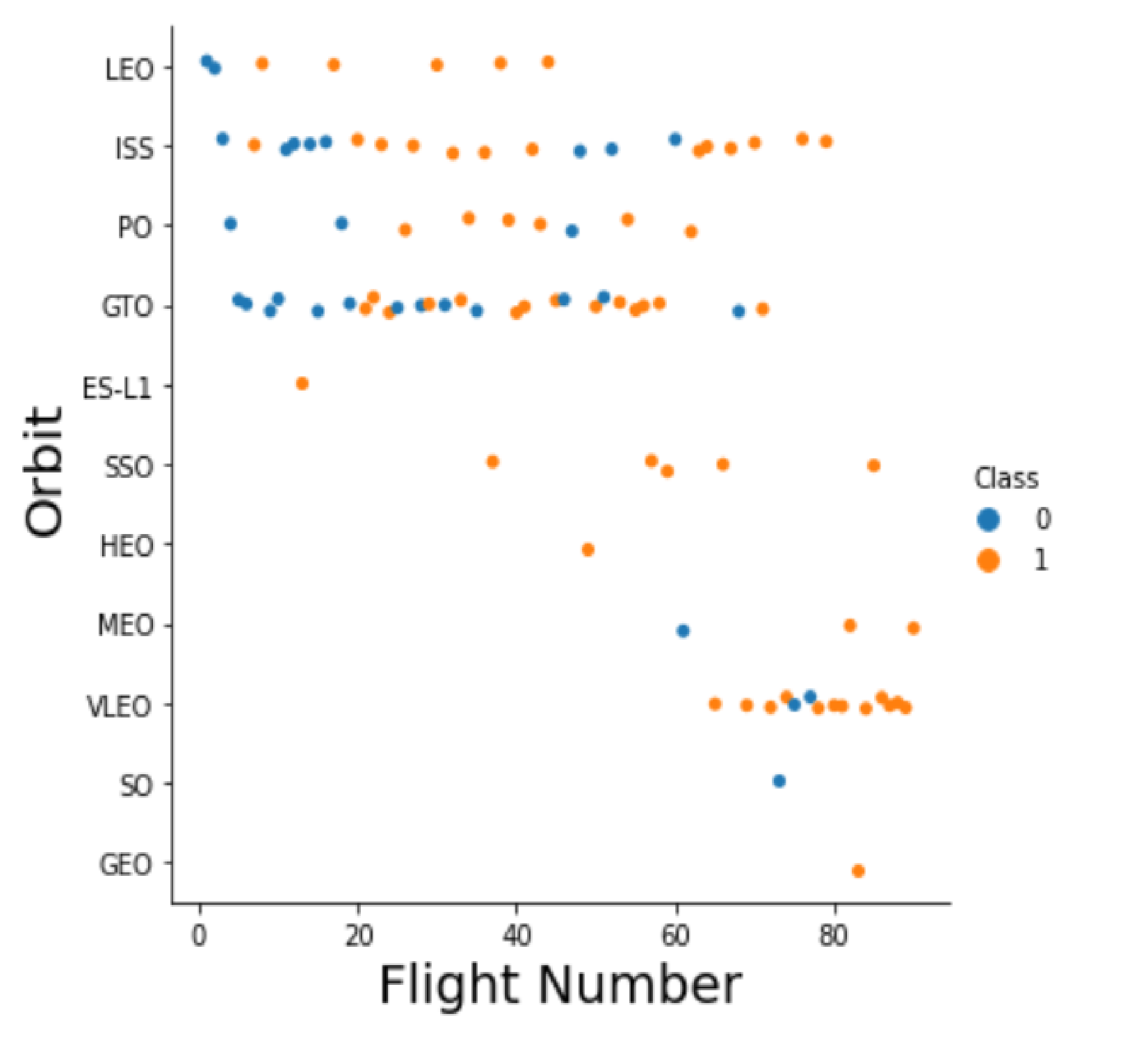
The greater the payload mass higher the success rate for the Rocket.

Success Rate vs. Orbit Type



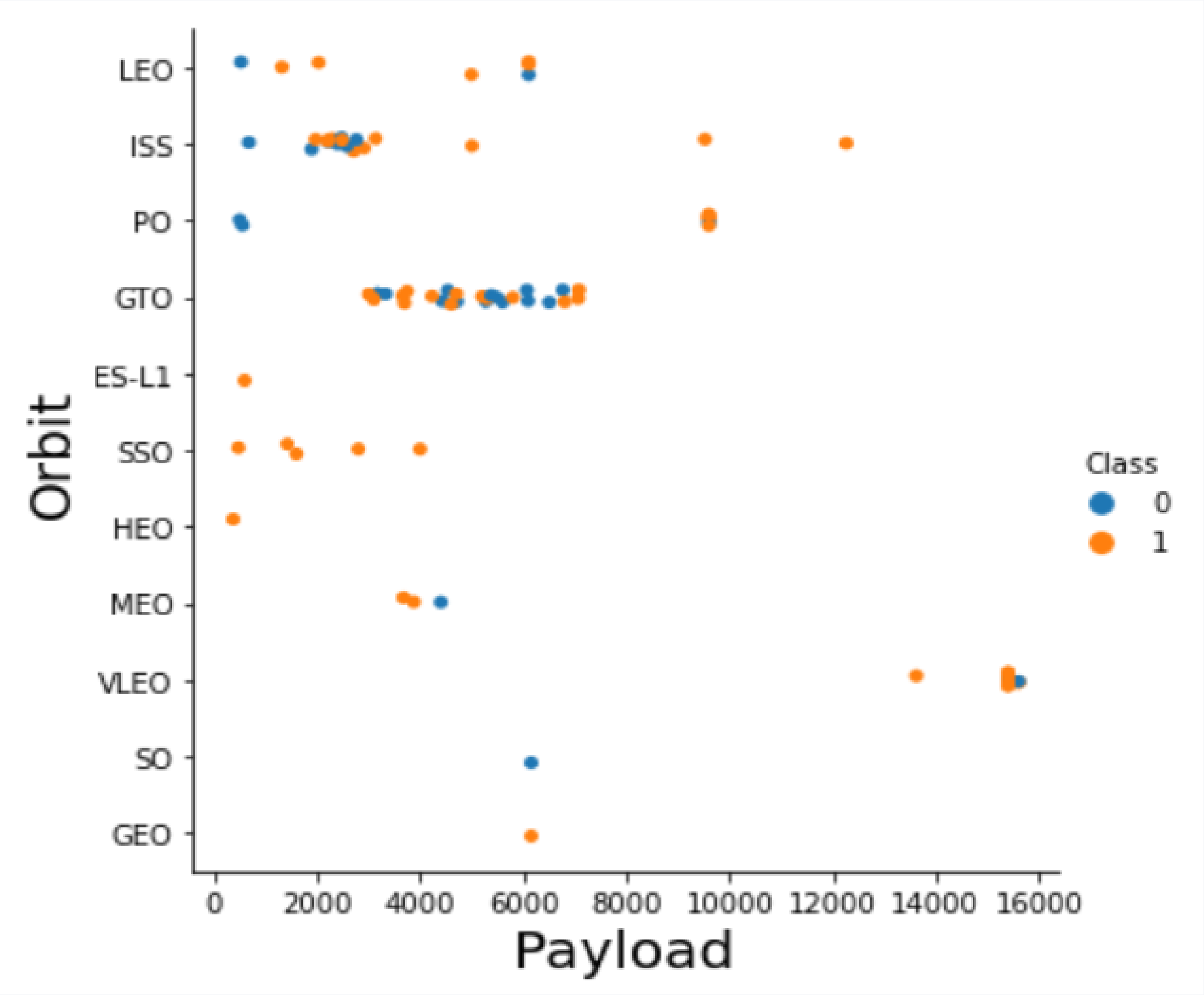
Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate.

Flight Number vs. Orbit Type



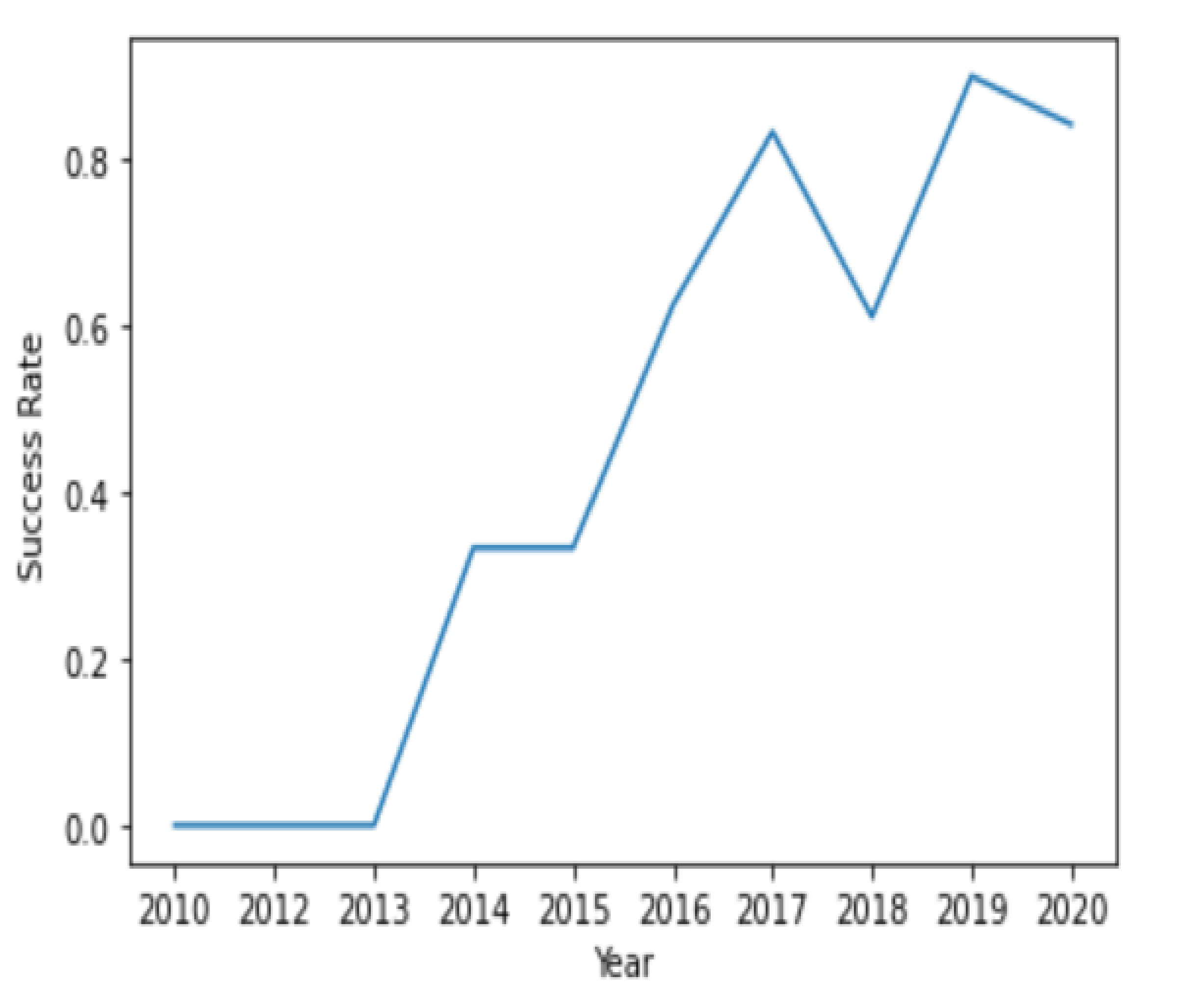
You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



You should observe that Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.

Launch Success Yearly Trend



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

All Launch Site Names

```
%sql SELECT DISTINCT(launch_site) FROM SPACEXTBL
```

```
 * ibm_db_sa://lmc42821:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

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

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

Launch Site Names Begin with 'CCA'

```
%sql SELECT * FROM SPACEXTBL WHERE launch_site like 'CCA%' limit 5
```

```
 * ibm_db_sa://lmc42821:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
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

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

Total Payload Mass

```
%sql SELECT SUM(payload_mass__kg_) AS total_payload_mass FROM SPACEXTBL WHERE customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://lmc42821:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

total_payload_mass
45596

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

Average Payload Mass by F9 v1.1

```
%sql SELECT avg(payload_mass__kg_) AS average_payload_mass FROM SPACEXTBL WHERE booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://lmc42821:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

average_payload_mass
2928

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

First Successful Ground Landing Date

```
%sql SELECT min(date) date FROM SPACEXTBL WHERE "Landing _Outcome" like '%Success (ground pad)%'
```

```
 * ibm_db_sa://lmc42821:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

DATE
2015-12-22

Find the dates of the first successful landing outcome on ground pad

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT booster_version FROM SPACEXTBL WHERE "Landing _Outcome" like '%Success (drone ship)%' and payload_mass__kg_ >4000 and payload_mass__kg_ <6000
```

```
 * ibm_db_sa://lmc42821:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

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

List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT mission_outcome,count(mission_outcome) AS mission_total FROM SPACEXTBL group by mission_outcome
```

```
 * ibm_db_sa://lmc42821:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

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

Calculate the total number of successful and failure mission outcomes

Boosters Carried Maximum Payload

```
%sql SELECT distinct(booster_version) FROM SPACEXTBL where payload_mass__kg_ in (select max(payload_mass__kg_) FROM SPACEXTBL)
```

```
 * ibm_db_sa://lmc42821:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

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

List the names of the booster which have carried the maximum payload mass

2015 Launch Records

```
%sql SELECT "Landing _Outcome",booster_version,launch_site FROM SPACEXTBL WHERE "Landing _Outcome" like '%Failure (drone ship)%' and "DATE" like '2015%'
```

```
 * ibm_db_sa://lmc42821:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

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

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

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

```
%%sql SELECT "Landing _Outcome",count("Landing _Outcome") AS outcome FROM SPACEXTBL
where "DATE" between '2010-06-04' and '2017-03-20'  group by "Landing _Outcome" order by outcome desc
```

```
 * ibm_db_sa://lmc42821:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

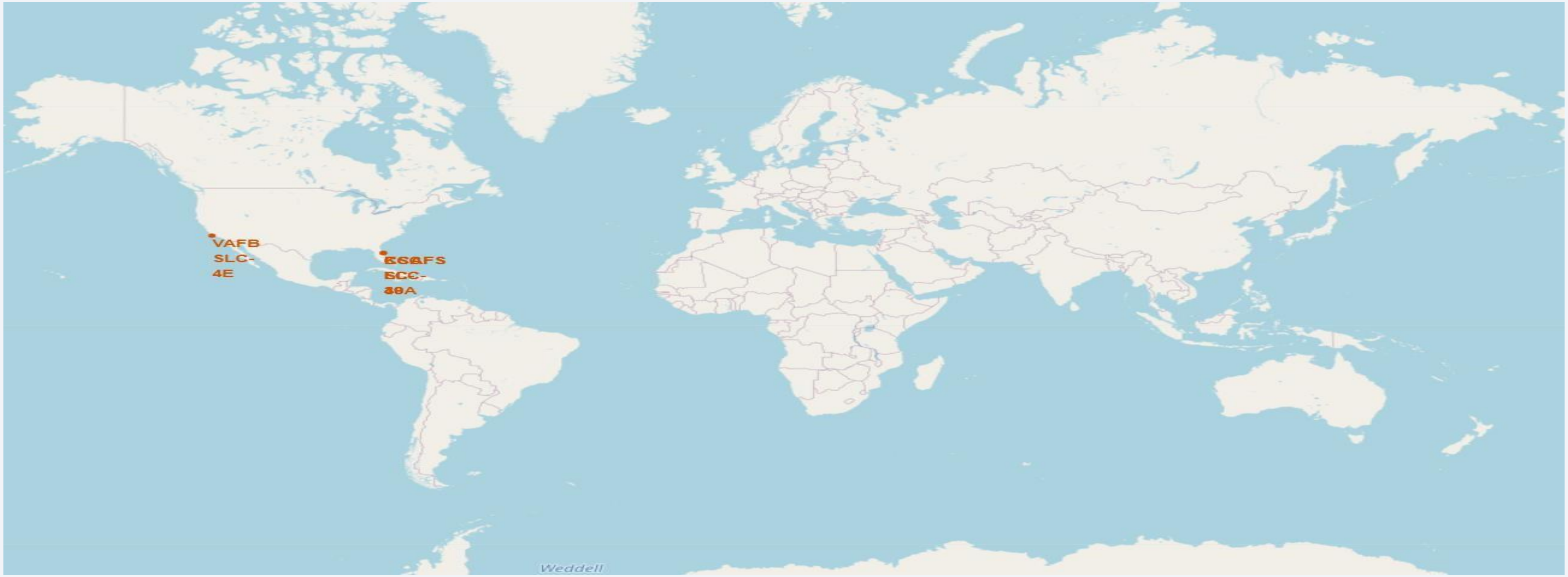
Landing _Outcome	outcome
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

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

Section 4

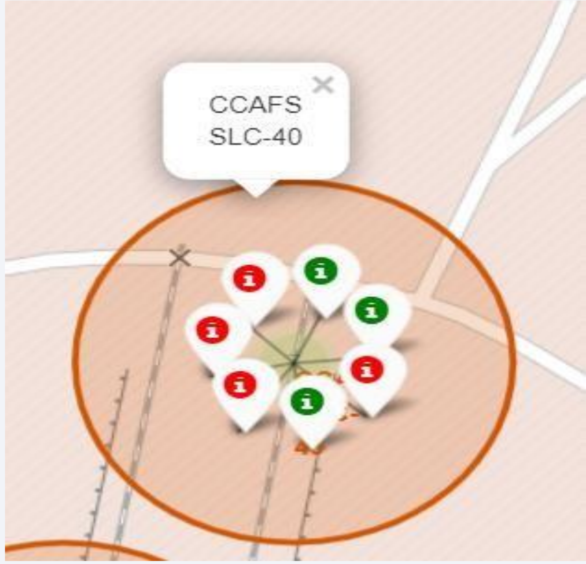
Launch Sites Proximities Analysis

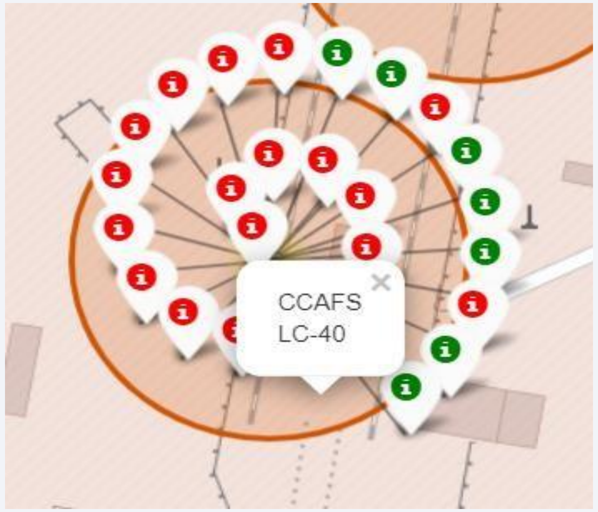
All launch sites global map markers



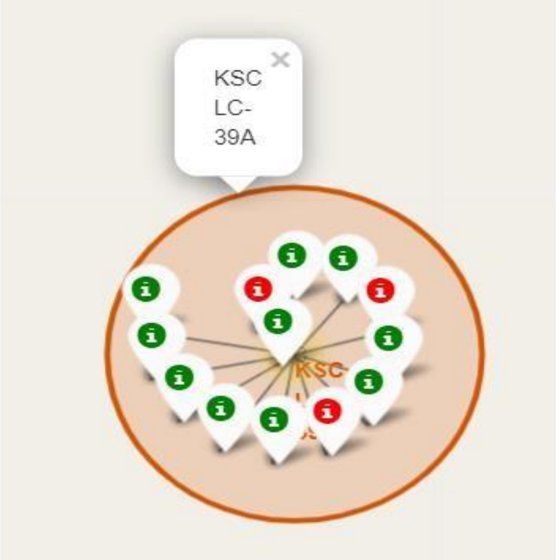
We can see that the SpaceX launch sites are in the United States of America coasts.

Color Labelled Markers

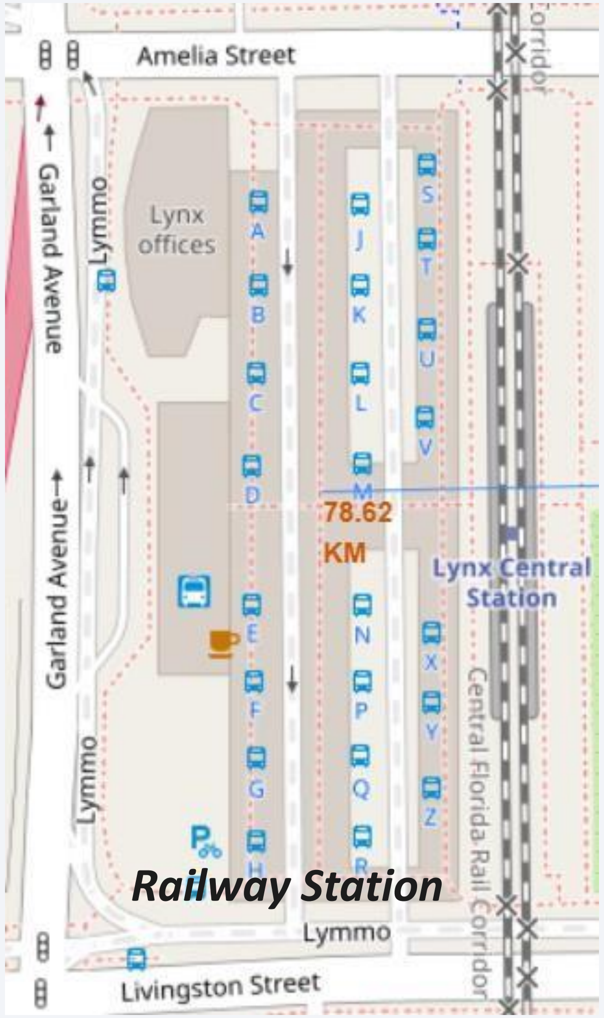


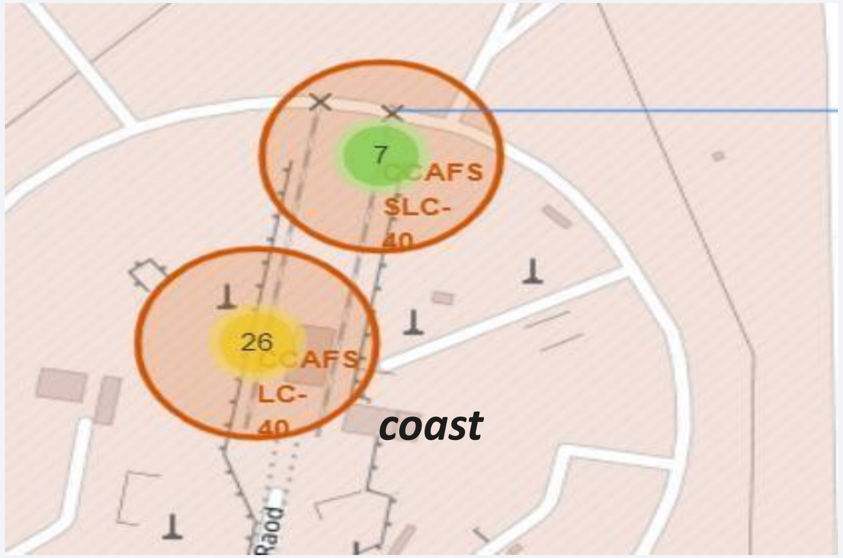


Green Marker shows successful Launches and Red Marker shows Failures



Launch Site Distance from Landmarks



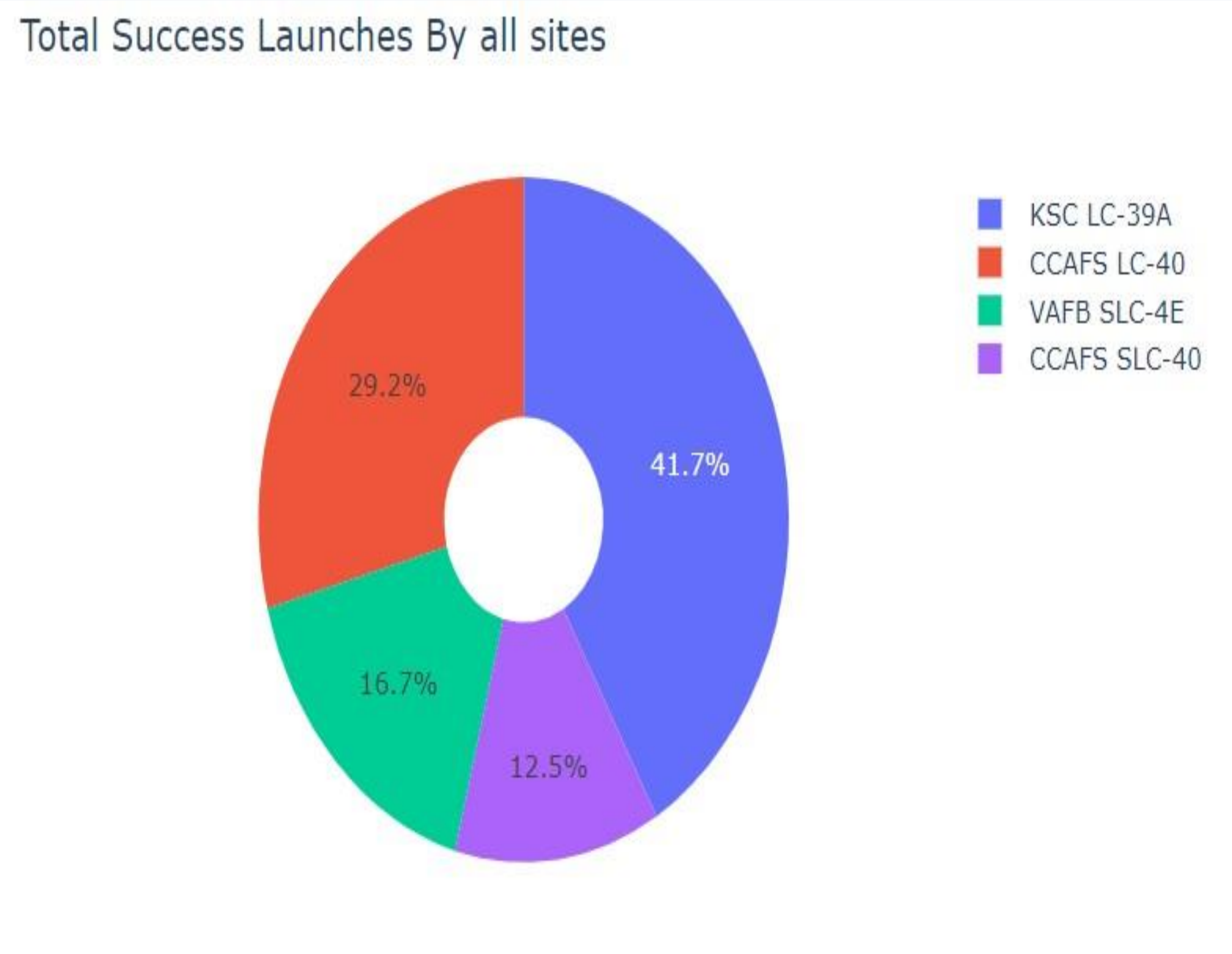


Are launch sites in close proximity to railways? No
Are launch sites in close proximity to highways? No
Are launch sites in close proximity to coastline? Yes
Do launch sites keep certain distance away from cities? Yes

Section 5

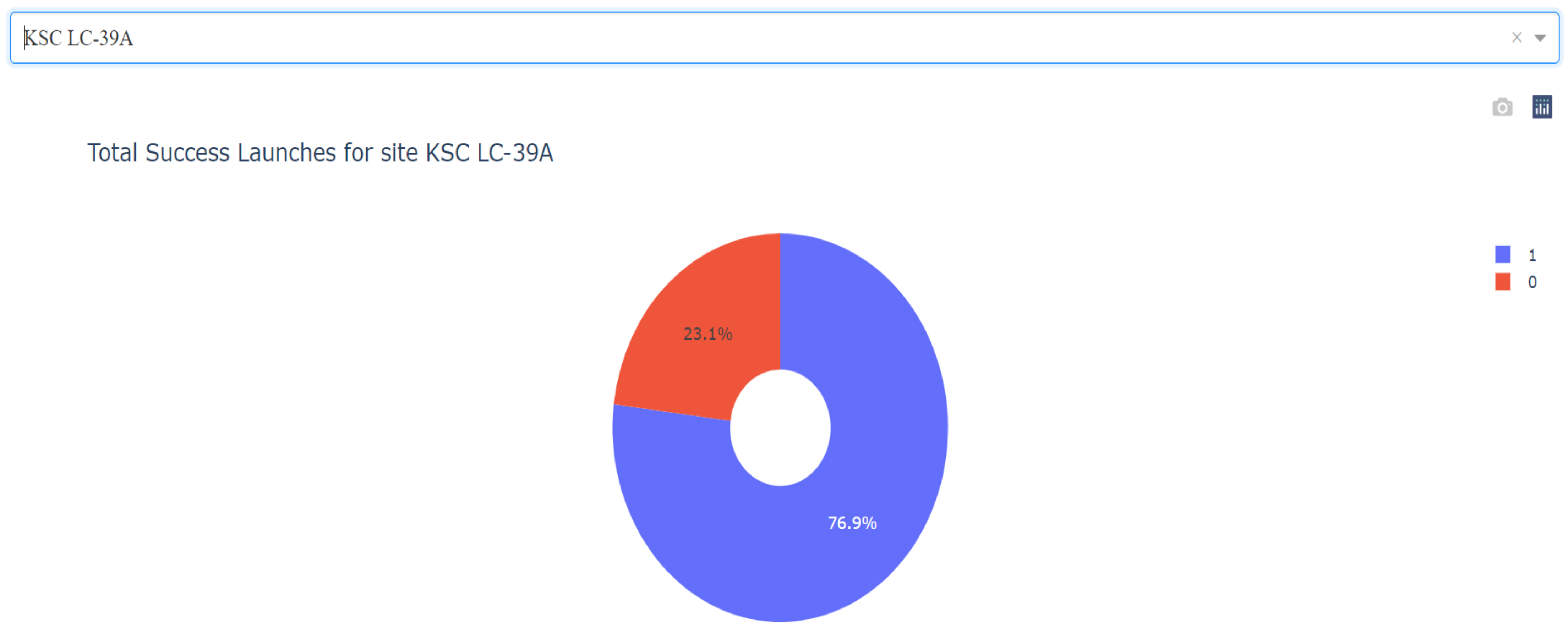
Build a Dashboard with Plotly Dash

Success Count of All Launch Sites



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

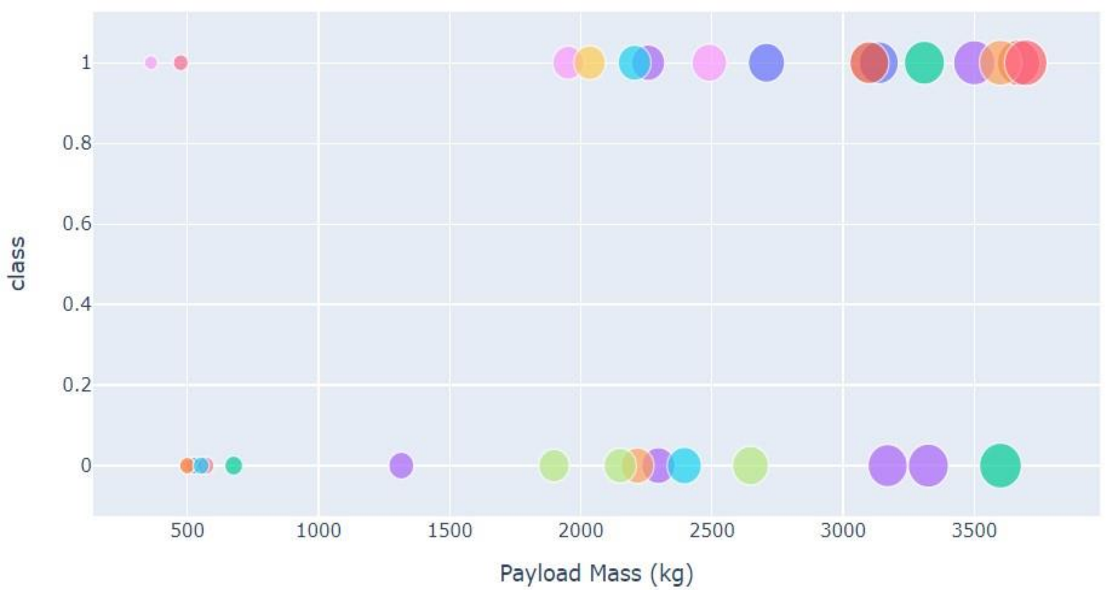
Success Ratio of KSC LC-39A Launch Site



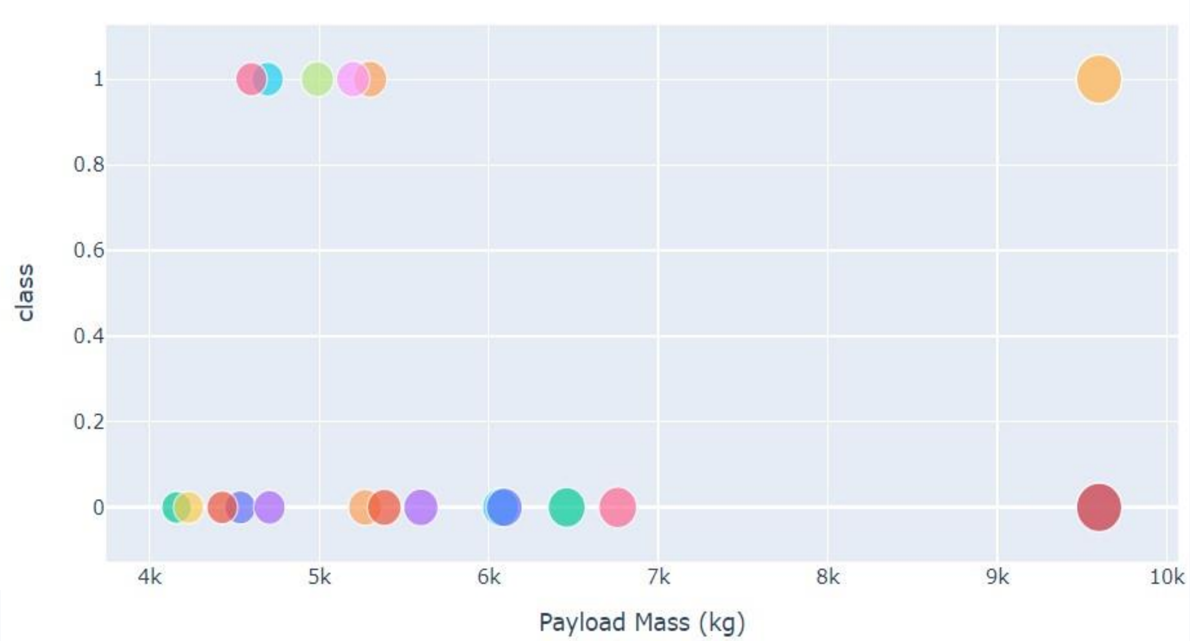
KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range

Low Weighted Payload 0kg – 4000kg



Heavy Weighted Payload 4000kg – 10000kg



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

Section 6

Predictive Analysis (Classification)

Classification Accuracy

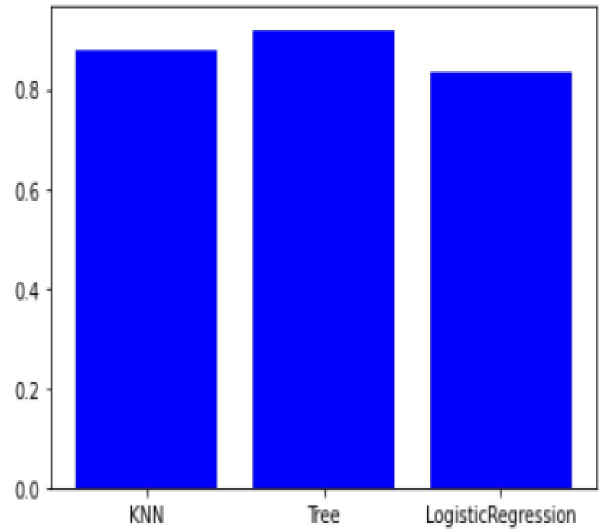
Decision Tree algorithm has the highest train accuracy of 91.96%

```python
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

```
Best Algorithm is Tree with a score of 0.9196428571428573
Best Params is : {'criterion': 'gini', 'max_depth': 12, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 5, 'splitter': 'random'}
```

```python
import matplotlib.pyplot as plt
width = 1.0
plt.bar(algorithms.keys(), algorithms.values(), color='b')
```

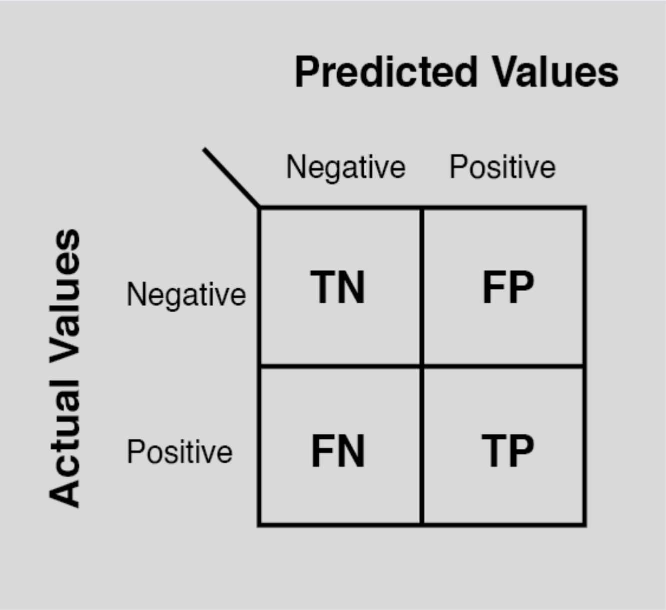
```
33]: <BarContainer object of 3 artists>
```

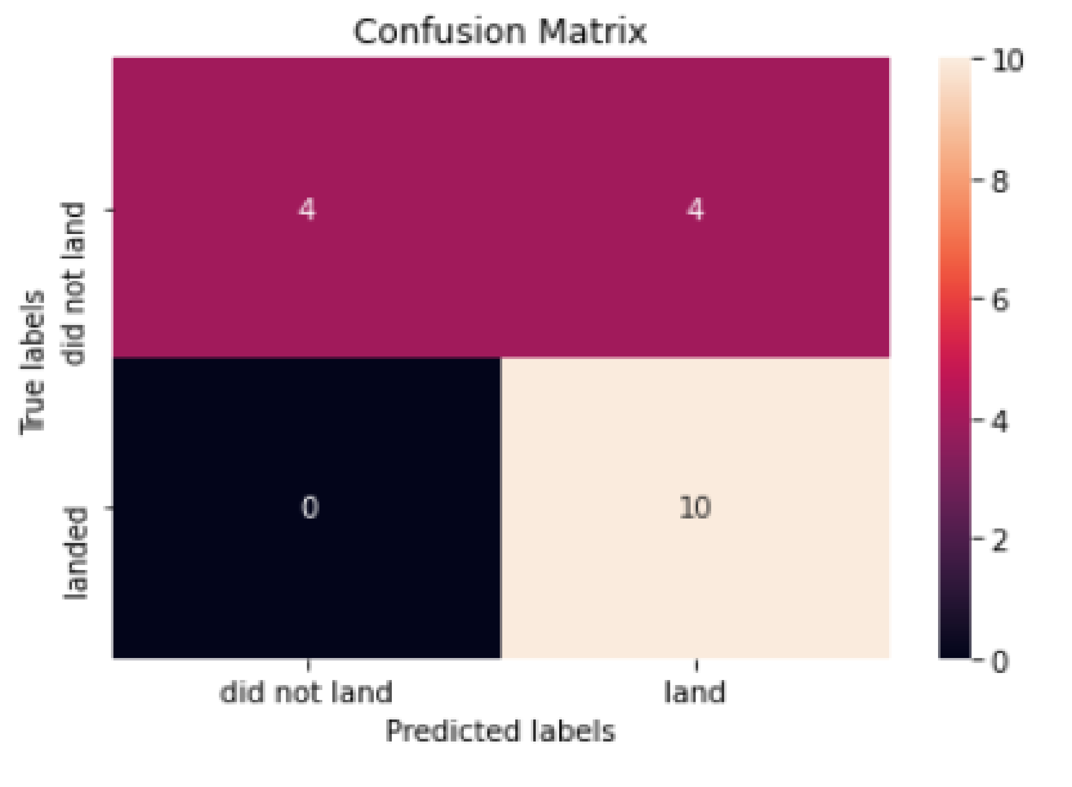


After selecting the best hyperparameters for the decision tree classifier using the validation data, we achieved 66.67% accuracy on the test data.

Confusion Matrix

Examining the confusion matrix, we see that Tree can distinguish between the different classes. We see that the major problem is false positives.





- The Tree Classifier Algorithm is the best for Machine Learning for this dataset
- Low weighted payloads perform better than the heavier payloads

Conclusions

- 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.
- Decision Tree Classifier showed promising result with given dataset predicting the success/failure takeoff based on feature parameters.

Appendix

- GitHub Repository

Thank you!