IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ibrahim Kobeissy
12-Feb-2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection and Scraping
  - Data wrangling
  - EDA using visualization and 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 will predict 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.
- Problems you want to find answers
  - The key factor that determines if the rocket will land successfully
  - 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

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Data encoding and removing irrelevant data
- 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 launch data,
  - This would give us information about the launches, rockets, payload, specifications and landing outcome

- Falcon 9 Launch data from Wikipedia is web scrapped using BeautifulSoup.

Data Collection – SpaceX API

- Getting Response from API

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

- Converting Response to a .json

```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

- Apply custom functions to clean data

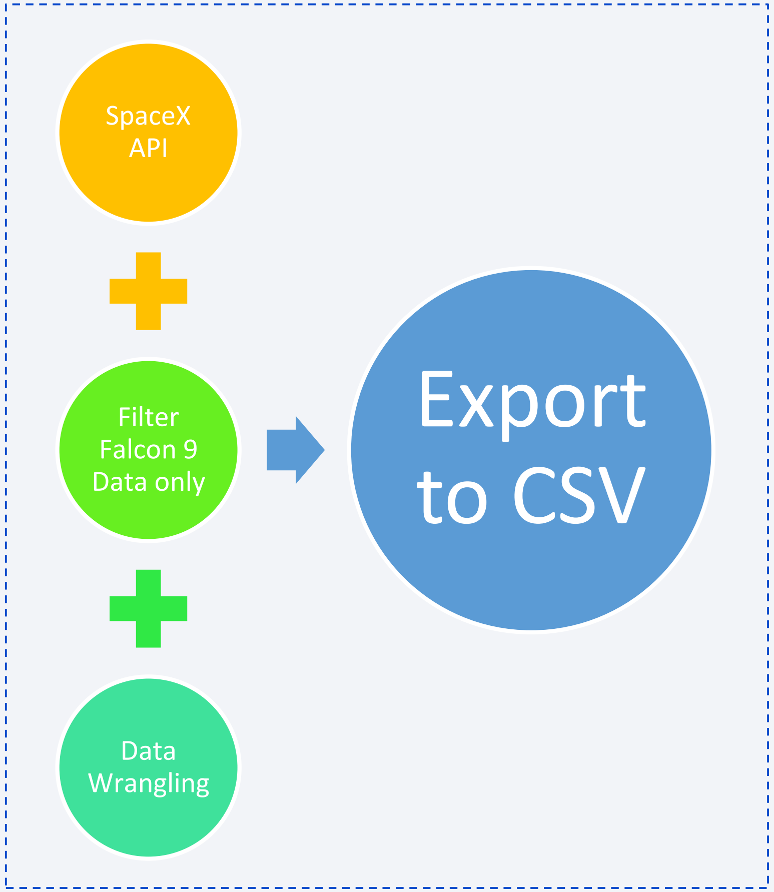
```
getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

- Data cleanup

```
PayloadMassMean = data_falcon9['PayloadMass'].mean()
data_falcon9['PayloadMass'].replace(np.nan,PayloadMassMean,inplace=True)
data_falcon9.isnull().sum()
```

- Filter and export to .csv

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



Github Notebook URL

Data Collection - Scraping

- Getting and parsing HTML

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
response = requests.get(static_url)
soup = BeautifulSoup(response.text,'html.parser')
```

- Filtering tables and columns

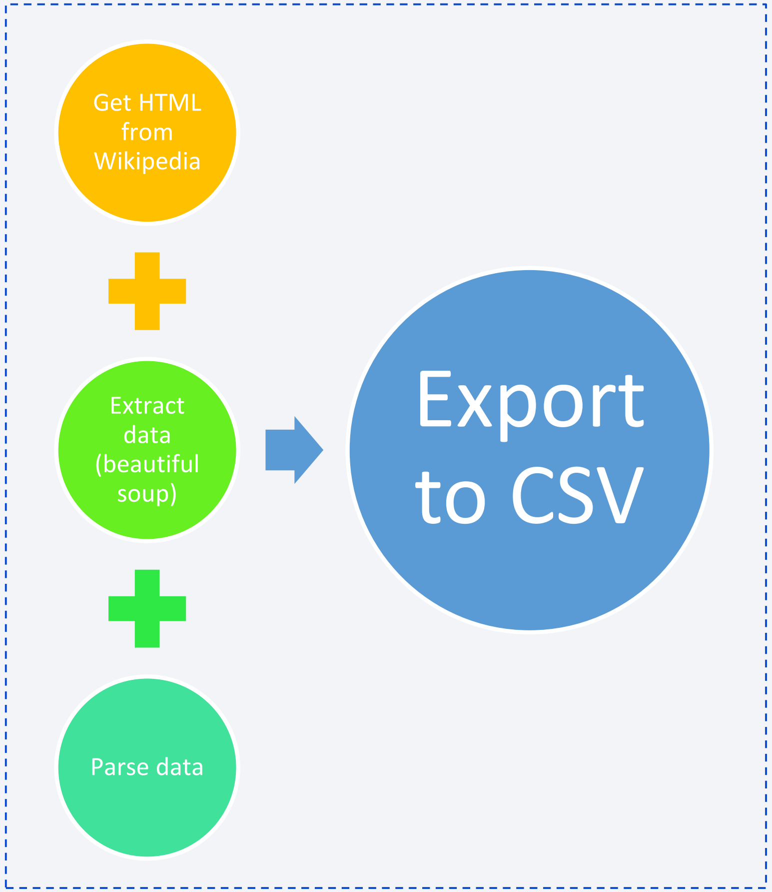
```
html_tables = soup.find_all(name='table')
first_launch_table = html_tables[2]
column_names = []

for i, row in enumerate(first_launch_table.find_all(name='th')):
    col_name = extract_column_from_header(row)
    if col_name is not None and len(col_name) > 0:
        column_names.append(col_name)
```

- Dictionary creation

```
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
```

- Export to CSV

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



Github Notebook URL

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

Perform Exploratory Data Analysis EDA on dataset

Calculate the number of launches at each site

Calculate the number and occurrence of each orbit

Calculate the number and occurrence of mission outcome per orbit type

Create a landing outcome label from Outcome column

Export dataset as .CSV

Github Notebook URL

EDA with Data Visualization

- Scatter Chart
    - 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 Chart
    - Success Rate of each orbit type
- Line Chart
    - Launch success Yearly trend

Github Notebook URL

EDA with SQL

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA’
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was acheived.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- 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

Github Notebook URL

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- On the Folium map we added the following object
  - Map Marker, to create a marker on the map
  - Icon Marker, to add an icon on the map
  - Circle Marker, to add a highlighted circle area with a text label on a specific coordinate
  - Polyline, to create a line between points
  - Marker Cluster Object, to visualize multiple marker having the same coordinates

Github Notebook URL

Build a Dashboard with Plotly Dash

- Pie Chart showing the total launches by a certain site/all sites
- Scatter Chart Scatter Graph showing the relationship with Outcome and Payload Mass (Kg) for the different Booster Versions

Predictive Analysis (Classification)

- 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.
- 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.

Github Notebook URL

Results

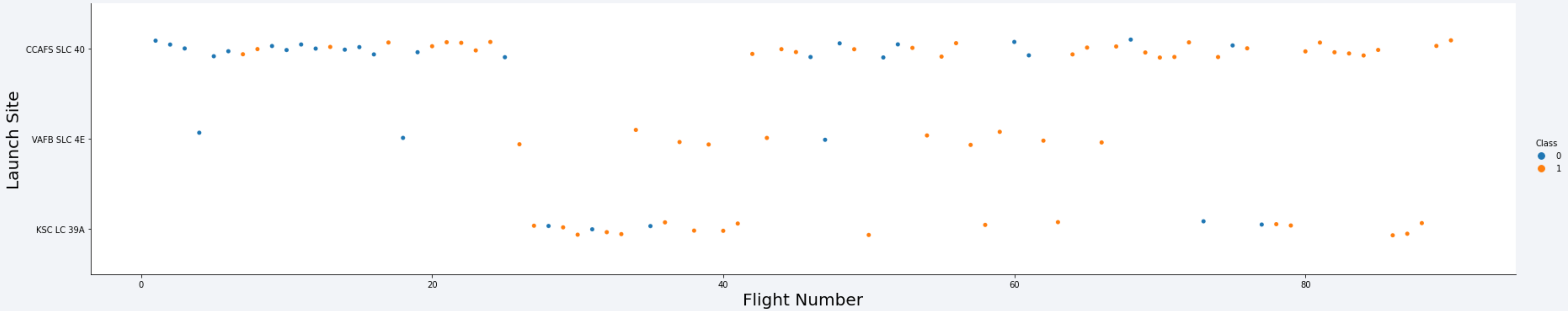
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

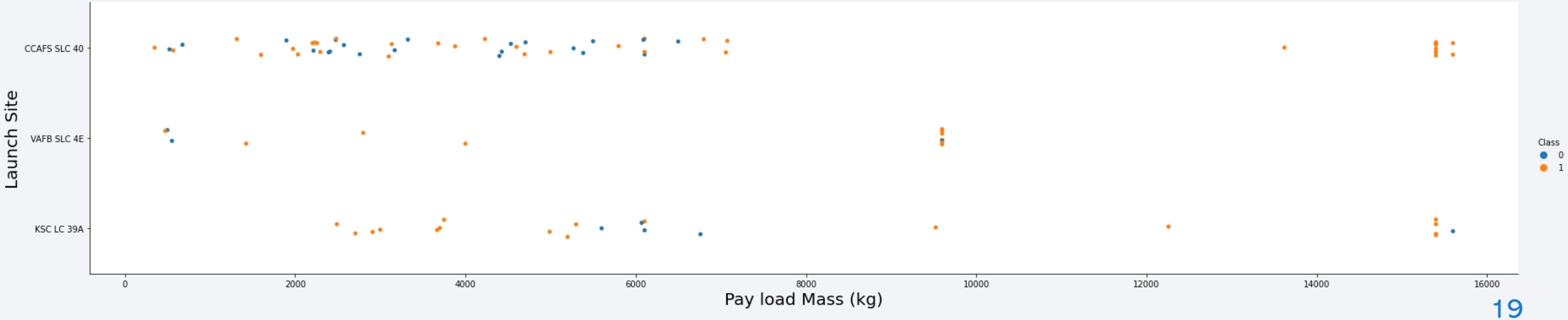
Flight Number vs. Launch Site

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



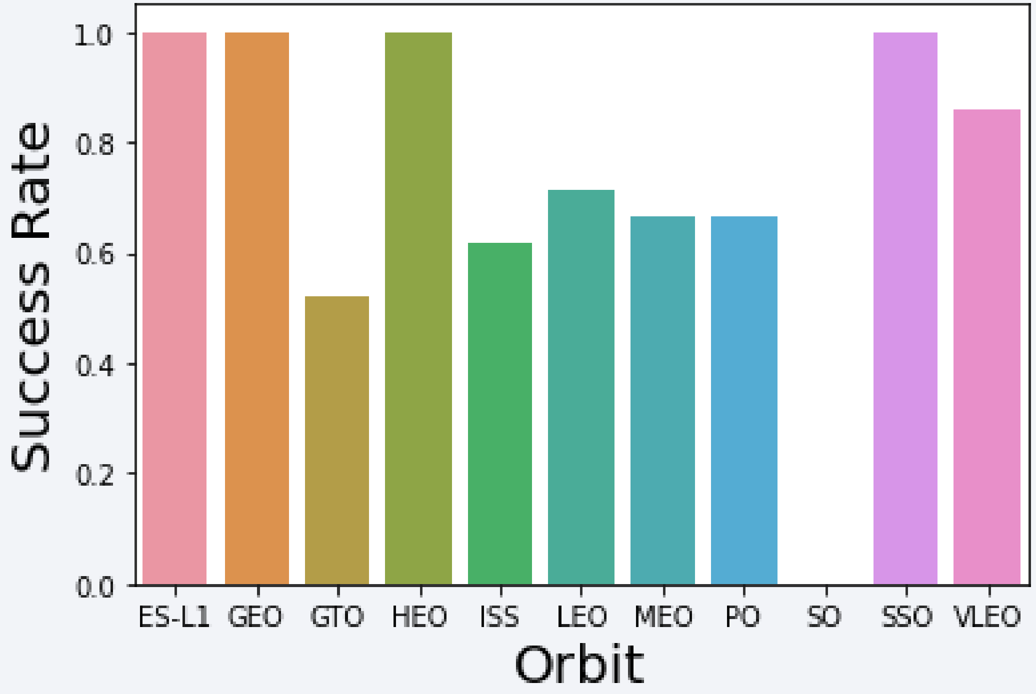
Payload vs. Launch Site

- The greater the payload mass 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.



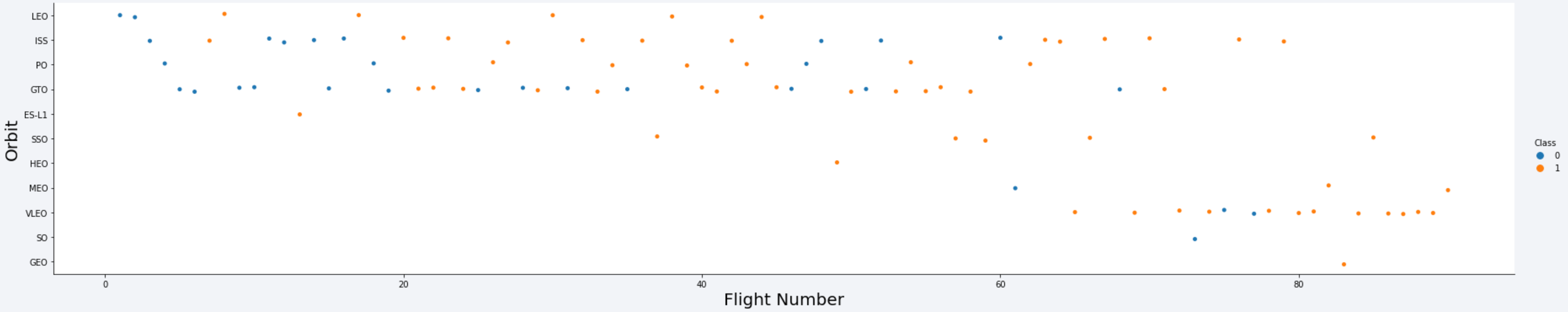
Success Rate vs. Orbit Type

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



Flight Number vs. Orbit Type

- 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.

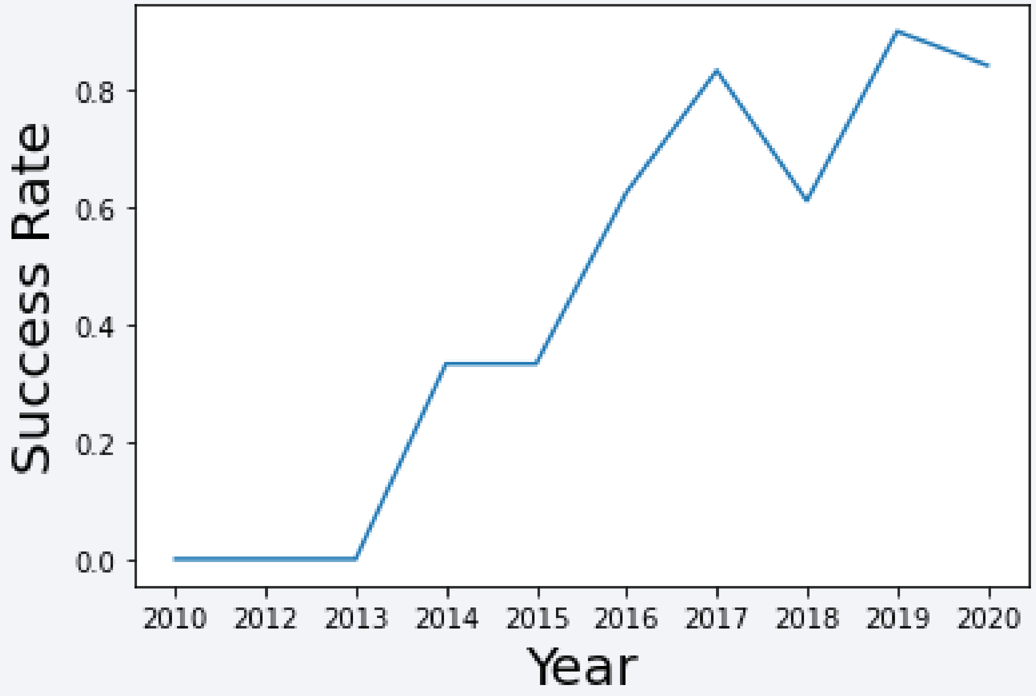


Payload vs. Orbit Type

- 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

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

```
%sql select distinct launch_site from SPACEXDATASET
```

✓ 0.7s Python

```
* ibm_db_sa://qsd36383:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

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'

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

✓ 0.7s Python

```
* ibm_db_sa://qsd36383:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/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

Total Payload Mass

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXDATASET WHERE CUSTOMER = 'NASA (CRS)'
```

✓ 0.7s Python

```
 * ibm_db_sa://qsd36383:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

1
45596

Average Payload Mass by F9 v1.1

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

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

✓ 0.6s

```
 * ibm_db_sa://qsd36383:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

1
2928

First Successful Ground Landing Date

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

Hint:Use min function

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

✓ 0.6s

```
 * ibm_db_sa://qsd36383:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

1
2015-12-22

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

```
%sql SELECT DISTINCT booster_version FROM SPACEXDATASET WHERE LANDING__OUTCOME = 'Success (drone ship)' AND payload_mass__kg_ BETWEEN
```

✓ 0.6s Python

```
* ibm_db_sa://qsd36383:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql SELECT COUNT(MISSION_OUTCOME), MISSION_OUTCOME FROM SPACEXDATASET GROUP BY MISSION_OUTCOME
```

✓ 0.6s

```
* ibm_db_sa://qsd36383:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

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

Boosters Carried Maximum Payload

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

```
%sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXDATASET T WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXDATASET
```

✓ 0.6s Python

```
 * ibm_db_sa://qsd36383:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/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

2015 Launch Records

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

```
%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXDATASET WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND TO_CHAR(DATE, 'YYYY')
```

✓ 0.6s Python

```
 * ibm_db_sa://qsd36383:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	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

```
'E BETWEEN TO_DATE('2010-06-04', 'YYYY-MM-DD') AND TO_DATE('2017-03-20', 'YYYY-MM-DD') GROUP BY LANDING__OUTCOME ORDER BY COUNT(1)
```

✓ 0.6s Python

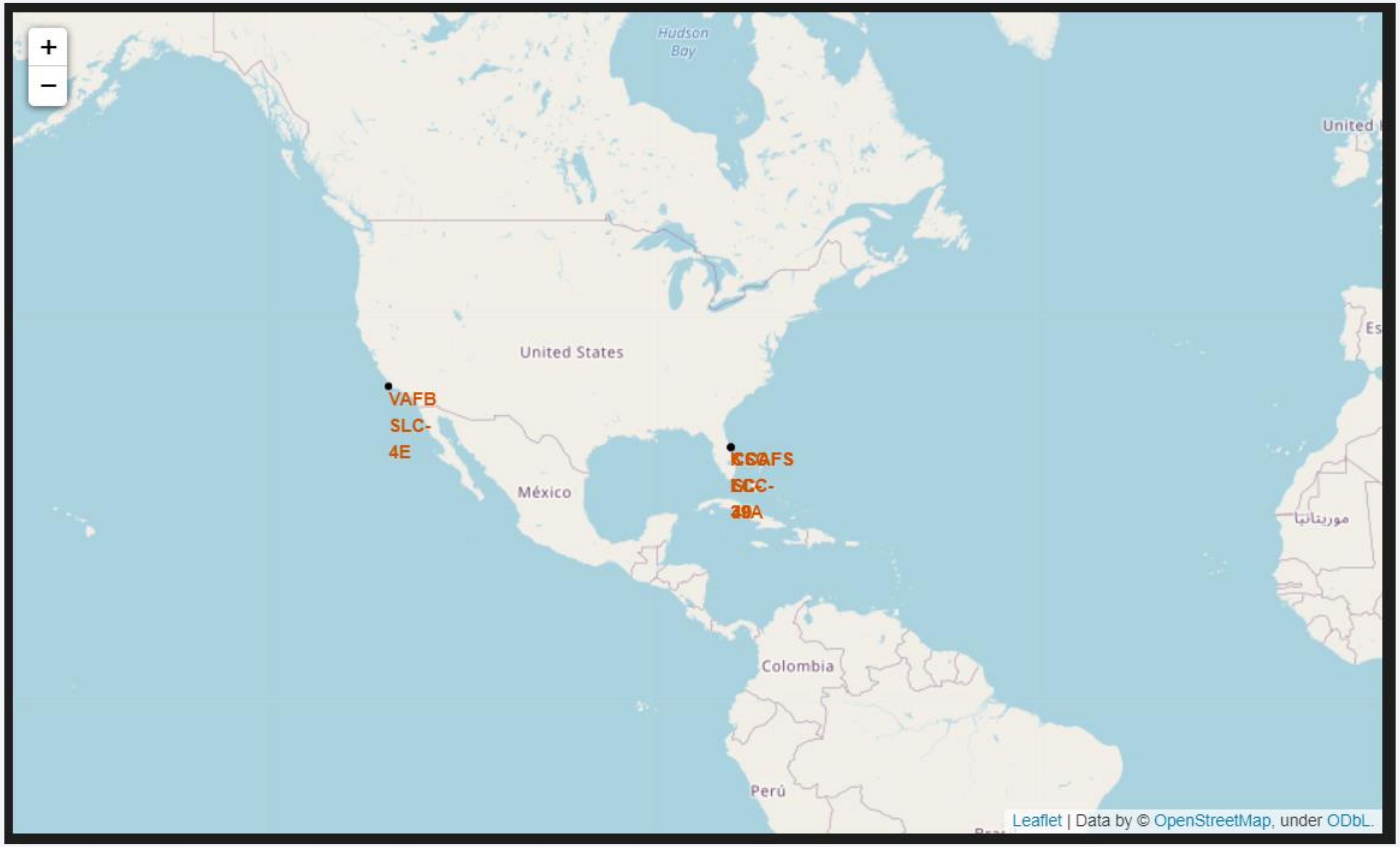
```
 * ibm_db_sa://qsd36383:***@2f3279a5-73d1-4859-88f0-a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30756/bludb
Done.
```

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

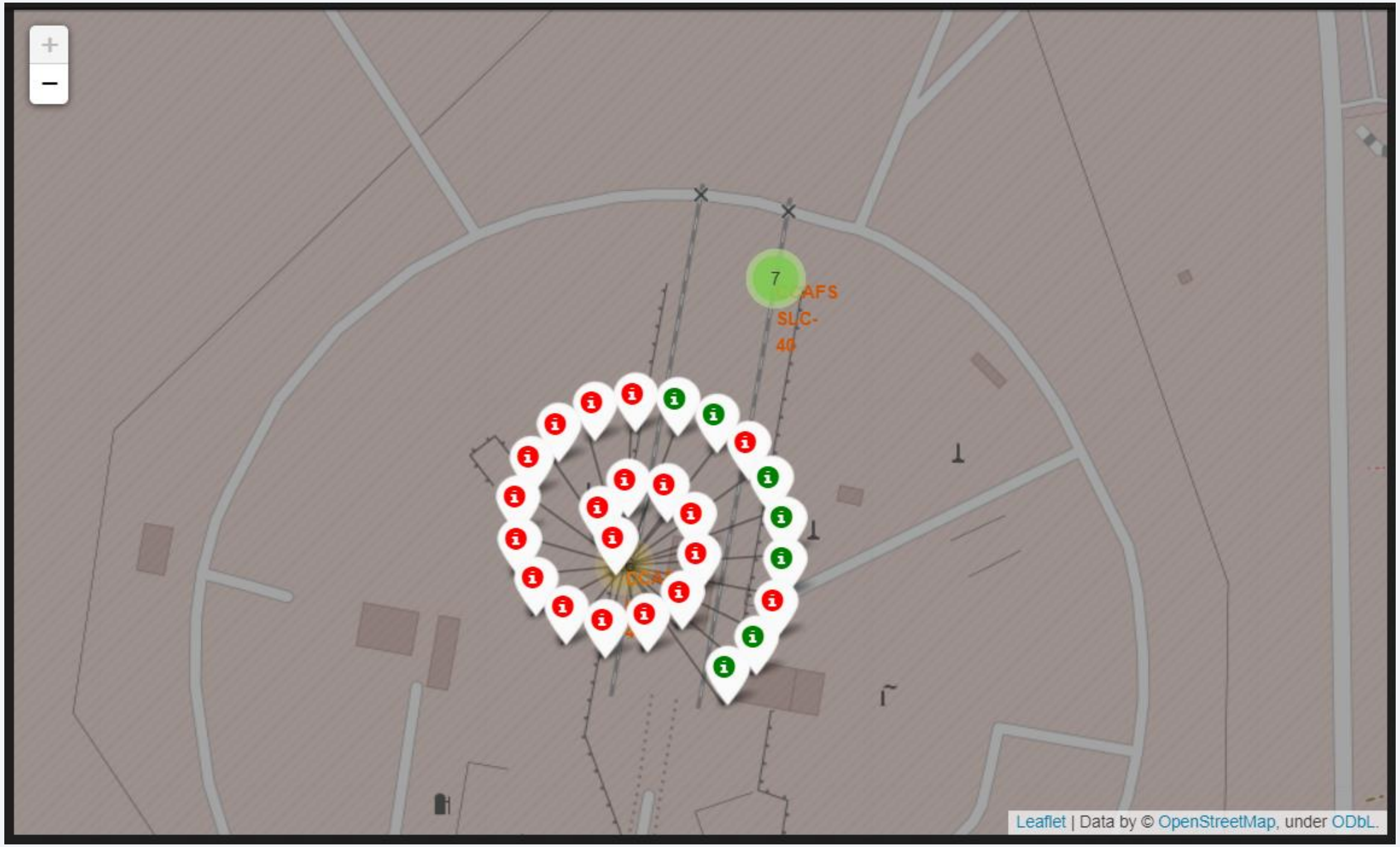
Section 4

Launch Sites Proximities Analysis

All launch sites



Colour Labelled Markers

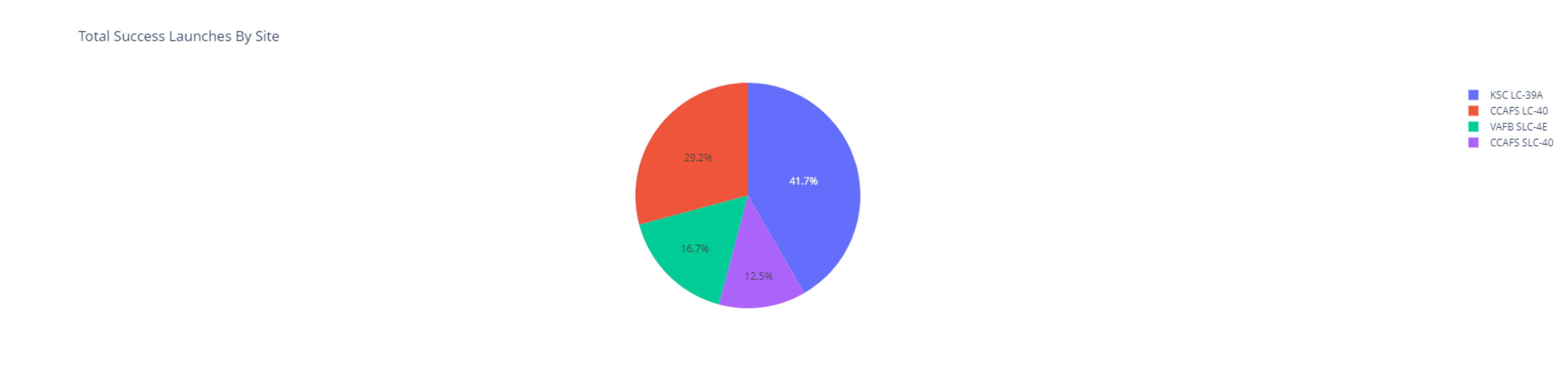


Working out Launch Sites distance to landmarks

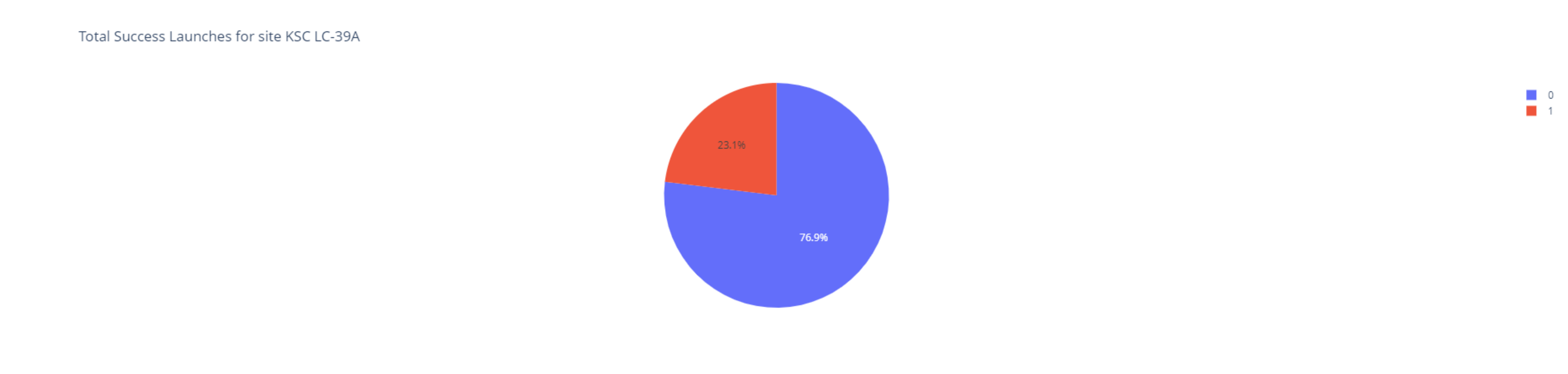
Section 5

Build a Dashboard with Plotly Dash

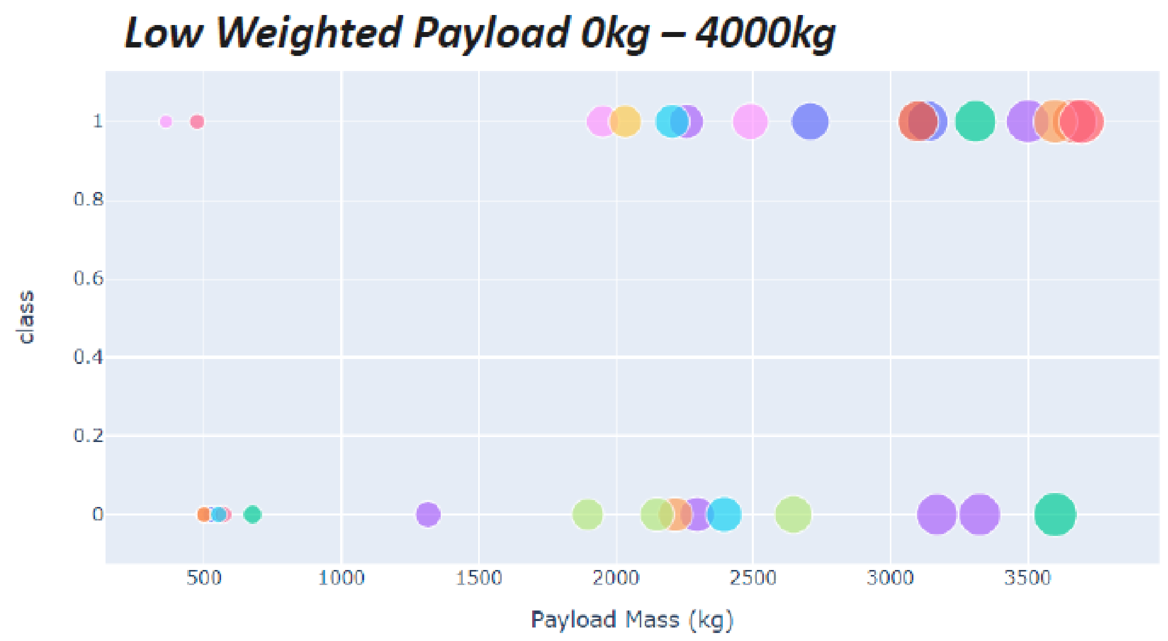
Total Success Launch By Site

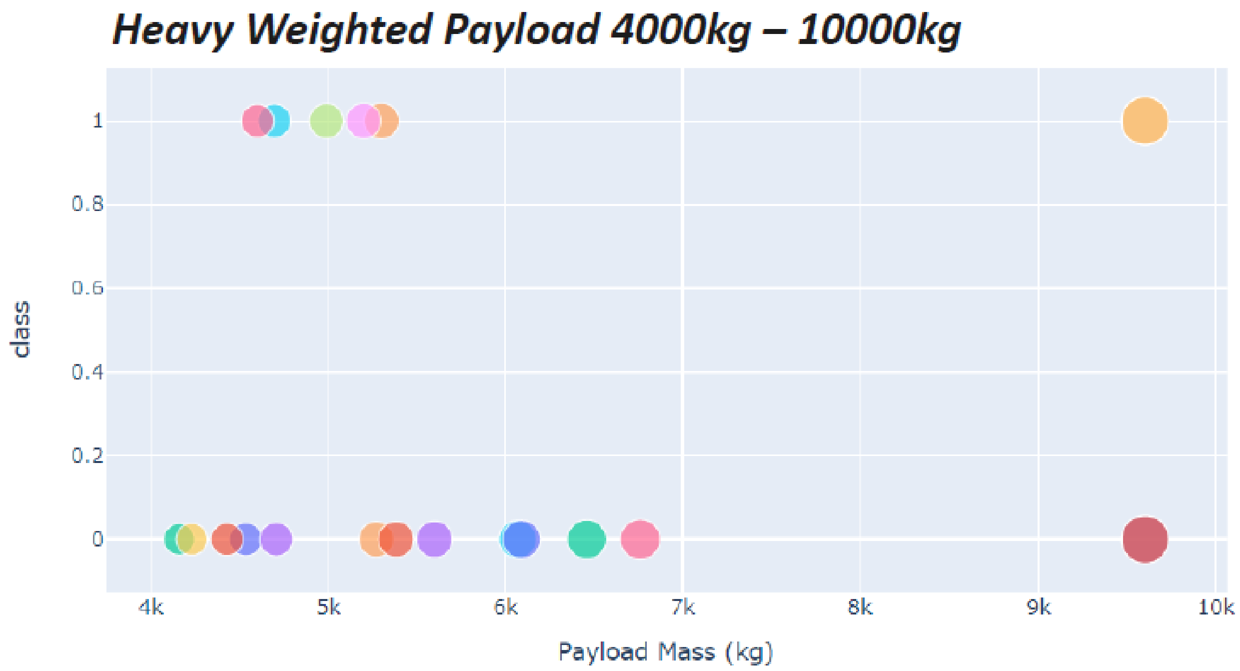


Highest launch success ratio



Payload vs. Launch Outcome





Section 6

Predictive Analysis (Classification)

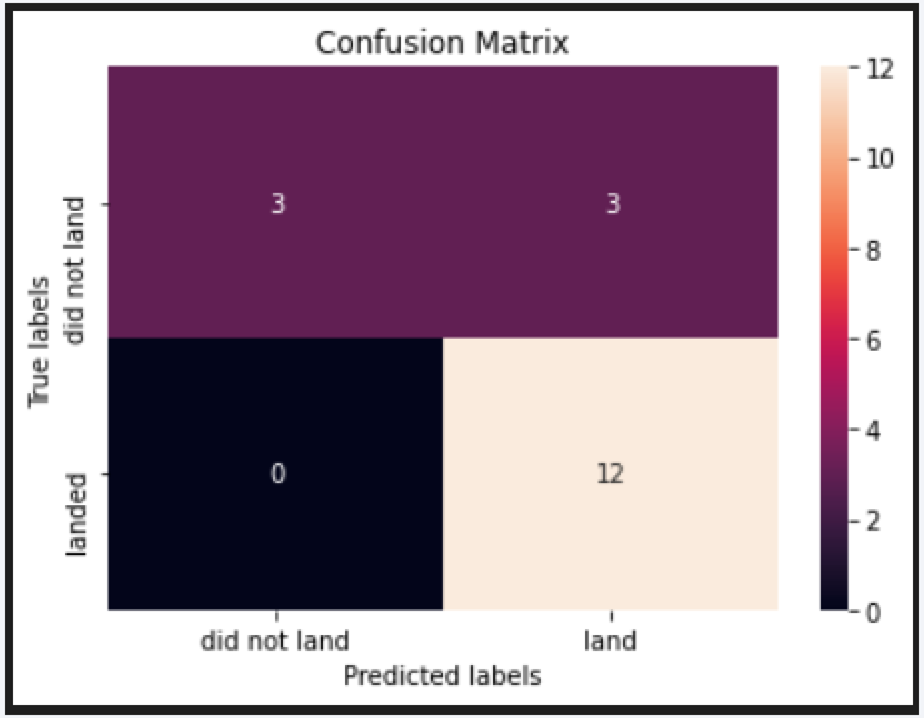
Classification Accuracy

```
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 0.8625
Best Params is : {'criterion': 'entropy', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 10,
'splitter': 'best'}
```

Confusion Matrix

- Confusion matrix for the Tree



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

Thank you!