Winning Space Race with Data Science

Dodema BITENIWE
18th December 2023

Outline

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

Executive Summary

Summary of methodologies

- Data Collection through API
- Data Collection with Web Scraping
- Data Wrangling
- Exploratory Data Analysis with SQL
- Exploratory Data Analysis with Data Visualization
- Interactive Visual Analytics with Folium
- Machine Learning Prediction

Summary of all results

- Exploratory Data Analysis result
- Interactive analytics
- Predictive Analytics result

Introduction

Project background and context

Space X 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 space X for a rocket launch. The goal of this project is to train a machine learning model and use public information to predict if the first stage will land successfully.

Problems you want to find answers

What factors determine if the rocket will land successfully?

The interaction amongst various features that determine the success rate of a successful landing.

What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

Data collection methodology: Data was collected using SpaceX API and web scraping from Wikipedia.

Perform data wrangling
- Clean the Data
- Dealing with Missing Values
- 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: Build, tune, evaluate classification models

Data Collection

- The data was collected using two methods
  - First Data collection was done using get request to the SpaceX API.
  - Then, we decoded the response content as a Json using .json() and turn it into a pandas dataframe using .json_normalize() method.
  - Next, we clean the data and dealing with missing values.
  - 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 – SpaceX API

- We request and parse the SpaceX launch data using the GET request, filter the data to only include Falcon9 launches and dealing with missing values.
- The link to the notebook is https://github.com/dodemabiteniwe/IBM_ds_CapstProject/blob/d42d70ad51ca91ee808d2bd11ada223bbb1ced02/jupyter-labs-spacex-data-collection-api.ipynb

request and parse data	filter the data	Dealing with missing values
response = requests.get(static_json_url) results=json.loads(response.text) data=pd.json_normalize(results)	data_falcon9=data_2.loc[data_2["BoosterVersion"]=='Falcon 9']	meanp=data_falcon9["PayloadMass"].mean() data_falcon9["PayloadMass"].replace(to_replace = np.nan, value =meanp,inplace=True)

Data Collection - Scraping

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- The link to the notebook is https://github.com/dodemabiteniwe/IBM_ds_CapstProject/blob/d42d70ad51ca91ee808d2bd11ada223bbb1ced02/jupyter-labs-webscraping.ipynb

Request HTML page and create BeautifulSoup object

```
response = requests.get(static_url)
soup=BeautifulSoup(response.text," html.parser")
```

Extract launch HTML tables and create dataframe by parsing the tables

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

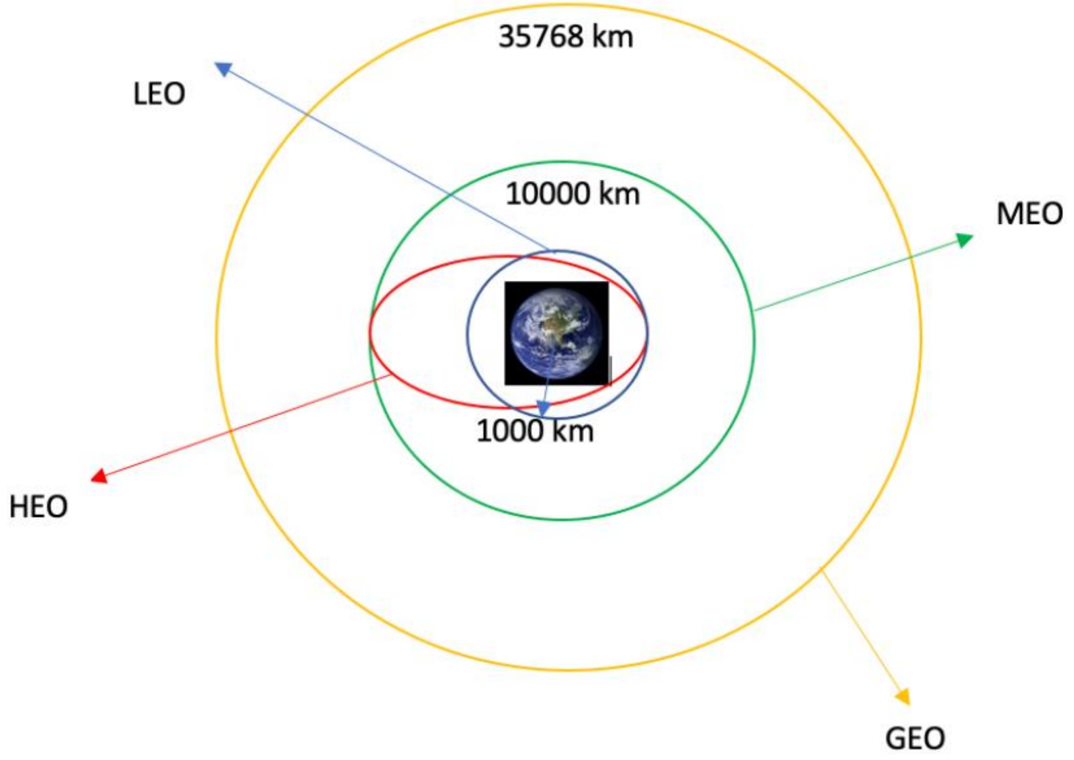
Data Wrangling

We calculated the number of launches at each site

We calculated the number and occurrence of each orbits

We calculated the number and occurrence of mission outcome of the orbits

We created landing outcome label from outcome column and exported the results to csv.



The link to the notebook is https://github.com/dodemabiteniwe/IBM_ds_CapstProject/blob/d42d70ad51ca91ee808d2bd11ada223bbb1ced02/labs_jupyter_spacex_Data%20wrangling2.ipynb

EDA with Data Visualization

- We explored the data by visualizing the relationship between:
    - flight number and payload
    - flight number and launch Site,
    - payload and launch site,
    - success rate of each orbit type and flight number and orbit type.
  - The goal was to see how these variables evolve together and how they affect the launch outcome
- We explored the launch success yearly trend.
- The link to the notebook is https://github.com/dodemabiteniwe/IBM_ds_CapstProject/blob/d42d70ad51ca91ee808d2bd11ada223bbb1ced02/labs%20EDA%20dataviz%20Dodema2.ipynb

EDA with SQL

- We loaded the SpaceX dataset into a PostgreSQL database without leaving the jupyter notebook.
- We applied EDA with SQL 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.
- The link to the notebook is https://github.com/dodemabiteniwe/IBM_ds_CapstProject/blob/d42d70ad51ca91ee808d2bd11ada223bbb1ced02/jupyter-labs-eda-sql-edx_sqllite(1).ipynb

Build an 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.

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.

The link to the notebook is https://github.com/dodemabiteniwe/IBM_ds_CapstProject/blob/d42d70ad51ca91ee808d2bd11ada223bbb1ced02/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a 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.

The link to the notebook is https://github.com/dodemabiteniwe/IBM_ds_CapstProject/blob/d42d70ad51ca91ee808d2bd11ada223bbb1ced02/spacex_dash_app.py

Predictive Analysis (Classification)

- We Perform exploratory Data Analysis and determine Training Labels:
  - create a column for the class
  - Standardize the data
  - Split into training data and test data
- We built different machine learning models and tune different hyperparameters using GridSearchCV:
  - Logistic Regression
  - SVM
  - Classification Trees
  - k nearest neighbors
- We found the best performing classification model.
- The link to the notebook is https://github.com/dodemabiteniwe/IBM_ds_CapstProject/blob/d42d70ad51ca91ee808d2bd11ada223bbb1ced02/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite(2).ipynb

Results

EXPLORATORY DATA ANALYSIS RESULTS

INTERACTIVE ANALYTICS DEMO IN SCREENSHOTS

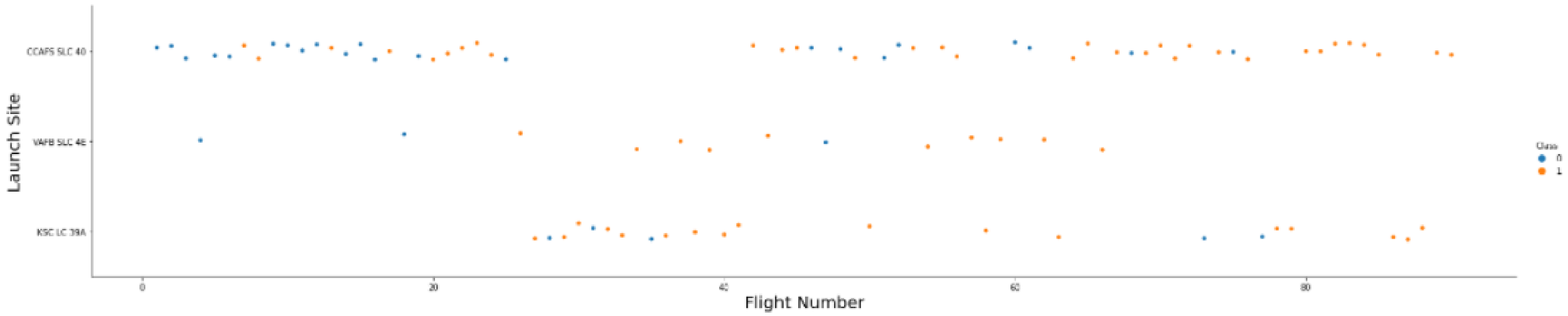
PREDICTIVE ANALYSIS RESULTS

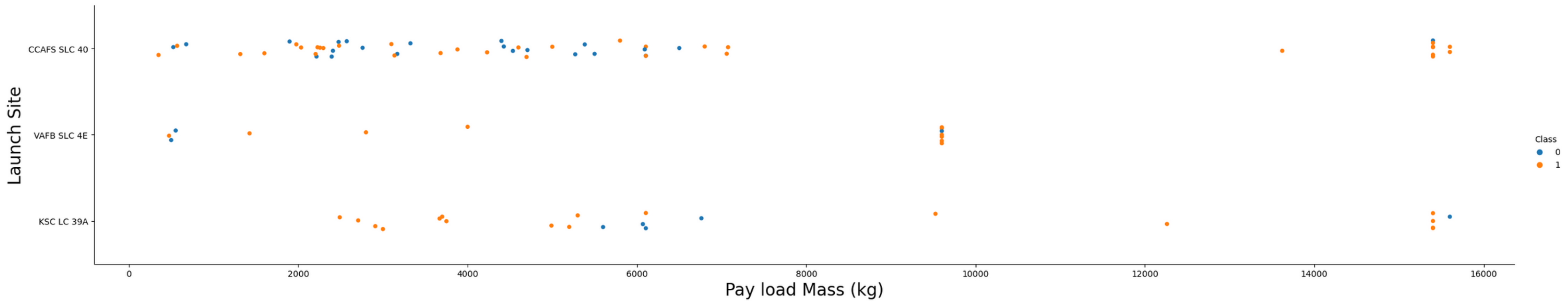
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

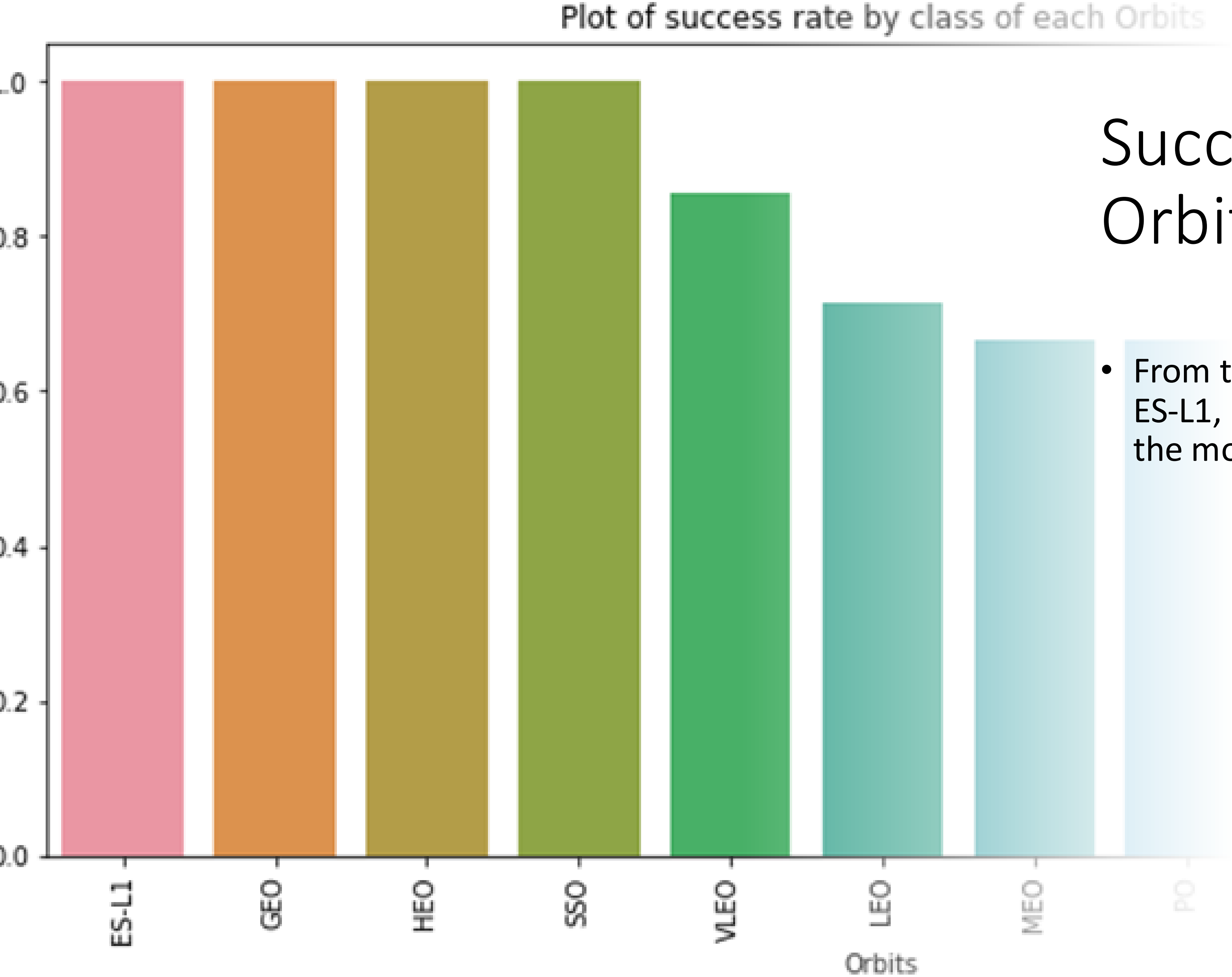
- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.





Payload vs. Launch Site

- The greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket
- We can also see that for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

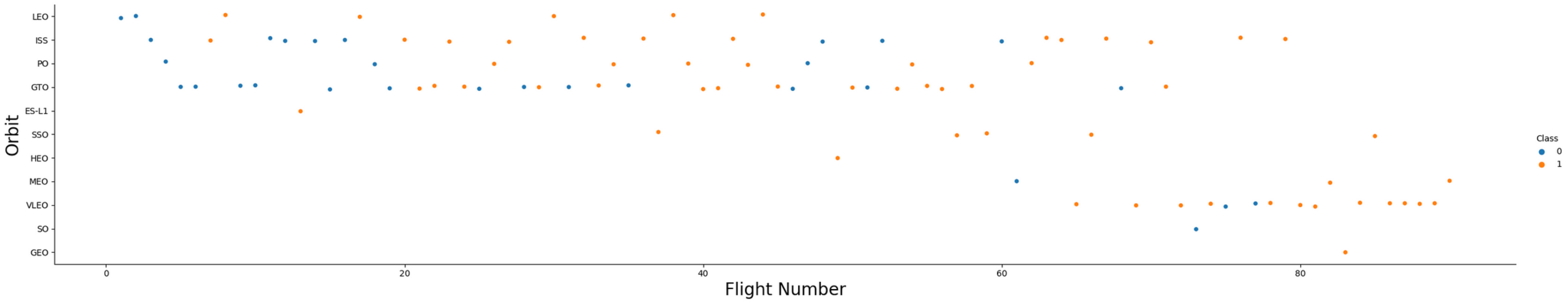


Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.

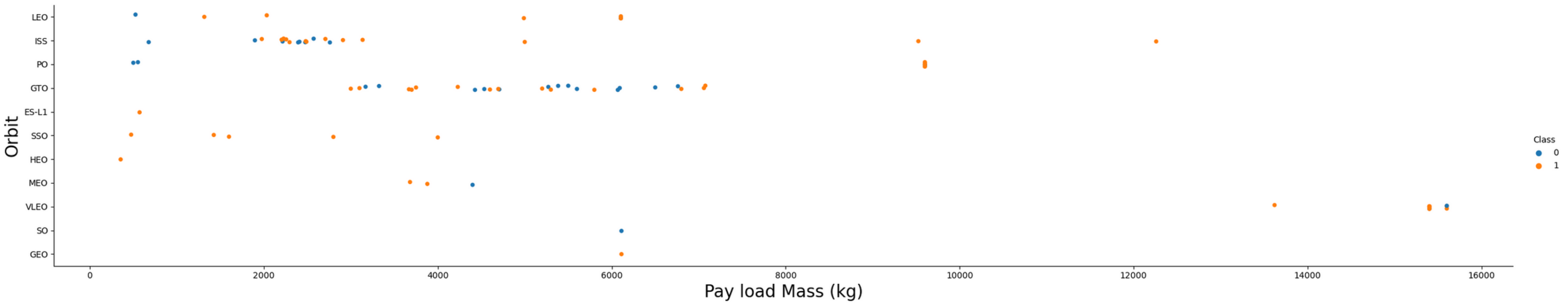
Flight Number vs. Orbit Type

- We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

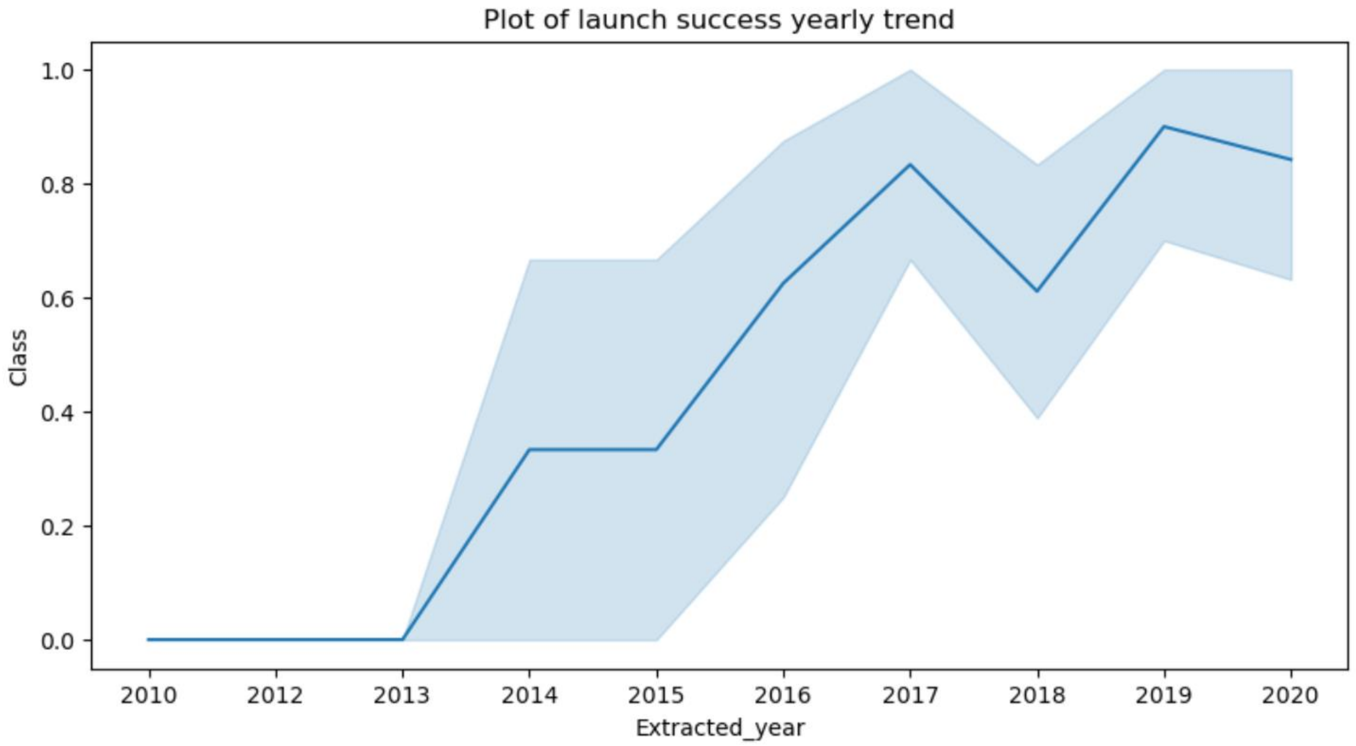


Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS. However for GTO we cannot distinguish this well



Launch Success Yearly Trend



- From the plot, we can observe that success rate since 2013 kept increasing till 2017 (stable in 2014) and after 2017 it started increasing.

All Launch Site Names

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

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

```
Entrée [14]: %sql select distinct "Launch_Site" from SPACEXTABLE

 * sqlite:///my_data1.db
Done.

Out[14]:
```

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

Launch Site Names Begin with 'KSC'

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

Entrée [15]:
```
%sql select * from SPACEXTABLE where "Launch_Site" like 'KSC%' limit 5
```

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

Out[15]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-03-16	6:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
2017-05-15	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

We used the query above to display 5 records where launch sites begin with `KSC`

Total Payload Mass

We calculated the total payload carried by boosters from NASA as 45596 using the query below

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

Entrée [16]:
```
%sql select SUM("PAYLOAD_MASS__KG_") from SPACEXTABLE where "Customer" = 'NASA (CRS)'
```

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

Out[16]:

SUM("PAYLOAD_MASS__KG_")
45596

Average Payload Mass by F9 v1.1

- We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

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

ntrée [10]:
```
%sql select AVG("PAYLOAD_MASS__KG_") from SPACEXTABLE where "Booster_Version" = 'F9 v1.1'
```

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

Out[10]:

AVG("PAYLOAD_MASS__KG_")
2928.4

First Successful Ground Landing Date

- We observed that the dates of the first successful landing outcome on ground pad was 08th April 2016

```
Entrée [11]: %sql select MIN("Date") from SPACEXTABLE where "Landing_Outcome" like '%Success (drone ship)%'

 * sqlite:///my_data1.db
Done.

Out[11]:
```

MIN("Date")
2016-04-08

Successful Drone Ship Landing with Payload between 4000 and 6000

- We used the **WHERE** clause to filter for boosters which have successfully landed on drone ship and applied the **AND** and **BETWEEN** condition to determine successful landing with payload mass greater than 4000 but less than 6000

Entrée [13]:
```
%sql select "Booster_Version" from SPACEXTABLE
where "Landing_Outcome" like 'Success (ground pad)' and "PAYLOAD_MASS__KG_" between 4000 and 6000
```

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

Out[13]:

Booster_Version
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1

Total Number of Successful and Failure Mission Outcomes

- We used wildcard like '%' to filter for **WHERE** MissionOutcome was a success or a failure.

```
Entrée [ ]: %sql select "Mission_Outcome", count(*) as SuccessOutcome from SPACEXTABLE where "Mission_Outcome" like '%Success%'

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

Out[14]:

Mission_Outcome	SuccessOutcome
Success	100

```
Entrée [15]: %sql select "Mission_Outcome", count(*) as FailureOutcome from SPACEXTABLE where "Mission_Outcome" like '%Failure%'

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

Out[15]:

Mission_Outcome	FailureOutcome
Failure (in flight)	1

Boosters Carried Maximum Payload

- We determined the booster that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX()** function.

```
Entrée [21]: %sql select "Booster_Version" from SPACEXTABLE where "PAYLOAD_MASS__KG_"=(select MAX("PAYLOAD_MASS__KG_") from SPACEXTABLE)

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

Out[21]:

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

2017 Launch Records

- We used a combinations of the **WHERE** clause, **LIKE** and **AND** to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2017

```
Entrée [17]: %sql select substr(Date,6,2),"Landing_Outcome","Booster_Version",
"Launch_Site" from SPACEXTABLE where "Landing_Outcome" like '%Success (ground pad)%' and substr(Date,0,5)='2017'

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

Out[17]:

substr(Date,6,2)	Landing_Outcome	Booster_Version	Launch_Site
02	Success (ground pad)	F9 FT B1031.1	KSC LC-39A
05	Success (ground pad)	F9 FT B1032.1	KSC LC-39A
06	Success (ground pad)	F9 FT B1035.1	KSC LC-39A
08	Success (ground pad)	F9 B4 B1039.1	KSC LC-39A
09	Success (ground pad)	F9 B4 B1040.1	KSC LC-39A
12	Success (ground pad)	F9 FT B1035.2	CCAFS SLC-40

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.

Entrée [18]:

```
%sql select "Landing_Outcome", count(*) as "number",
"Date" from SPACEXTABLE where "Date" between '2010-06-04' and '2017-03-20' group by "Landing_Outcome" order by "number" desc
```

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

Out[18]:

Landing_Outcome	number	Date
No attempt	10	2012-05-22
Success (drone ship)	5	2016-04-08
Failure (drone ship)	5	2015-01-10
Success (ground pad)	3	2015-12-22
Controlled (ocean)	3	2014-04-18
Uncontrolled (ocean)	2	2013-09-29
Failure (parachute)	2	2010-06-04
Precluded (drone ship)	1	2015-06-28

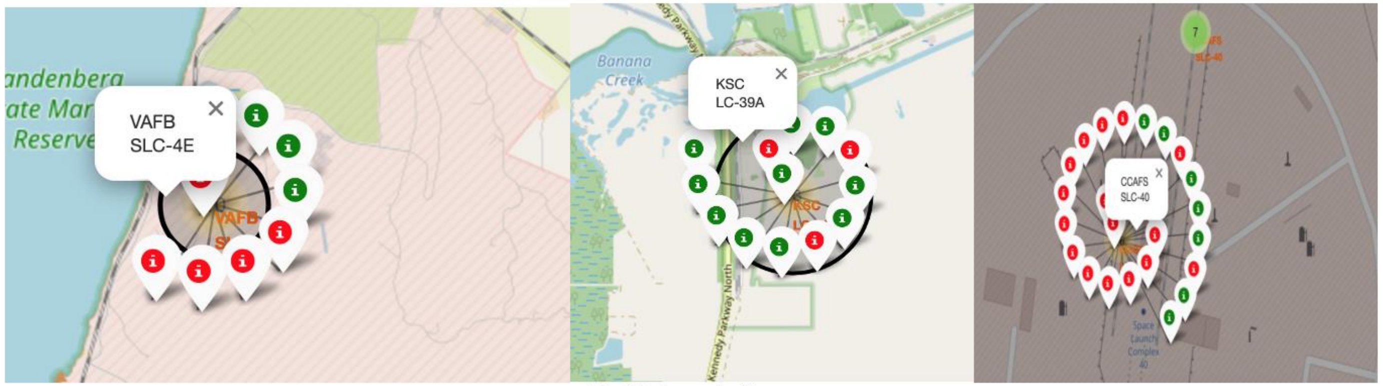
Section 4

Launch Sites Proximities Analysis

All launch sites global map markers

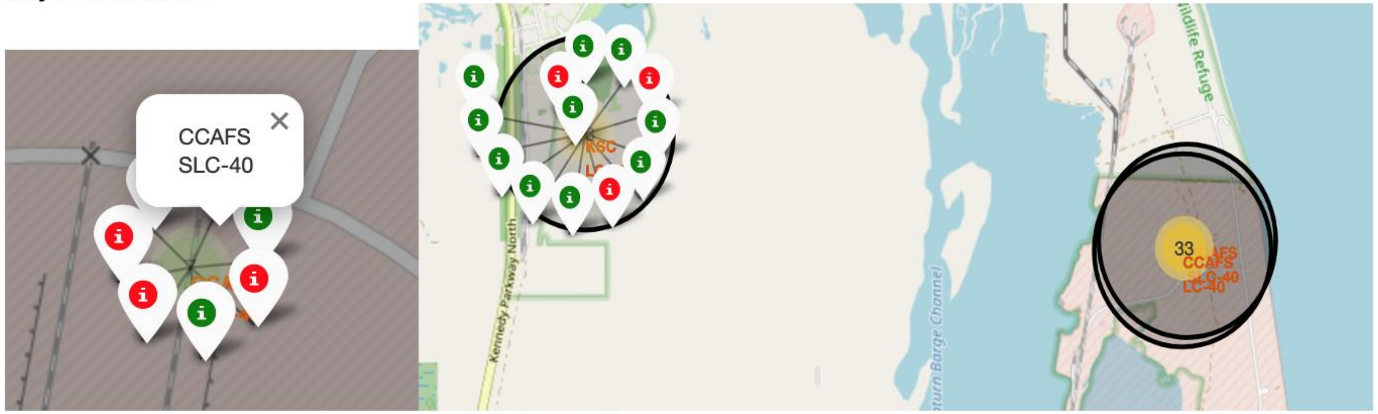
- We can see that the SpaceX launch sites are in the United States of America coasts, Florida and California

Markers showing launch sites with color labels



Florida launch site

California launch site



Florida launch site

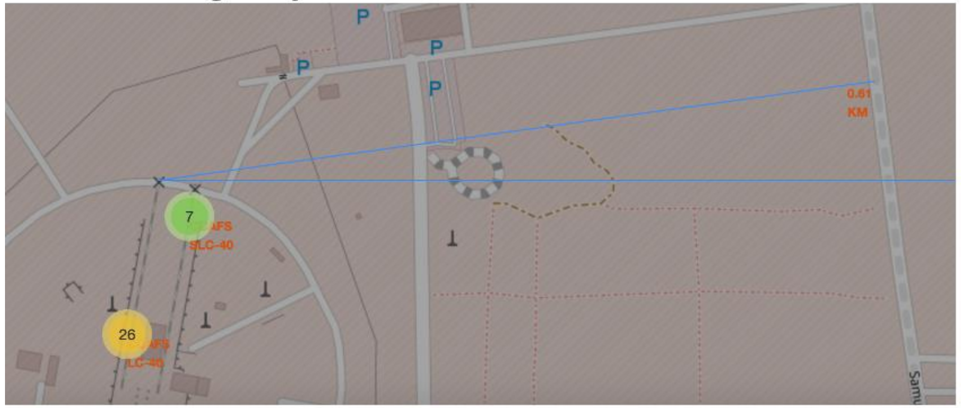
Green Marker shows Successful Launches and Red Marker shows Failures

Launch Site distance to landmarks

Distance for the launch site to the coastline

railway, highway, coastline

Distance to highway

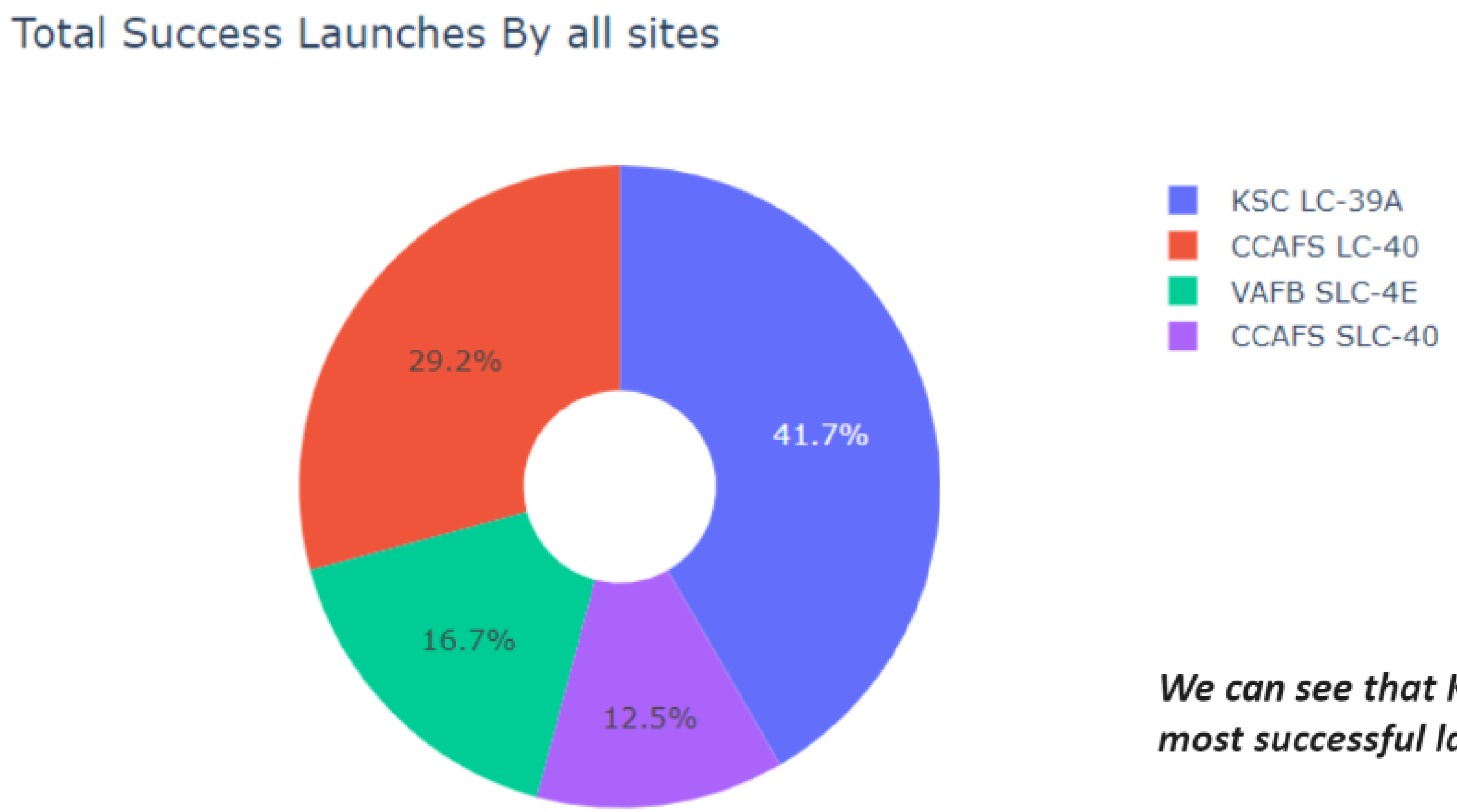


- Are launch sites in close proximity t
  YES
- Are launch sites in close proximity t
  YES
- Are launch sites in close proximity t
- YES
- Do launch sites keep certain distanc
  cities? YES

Section 5

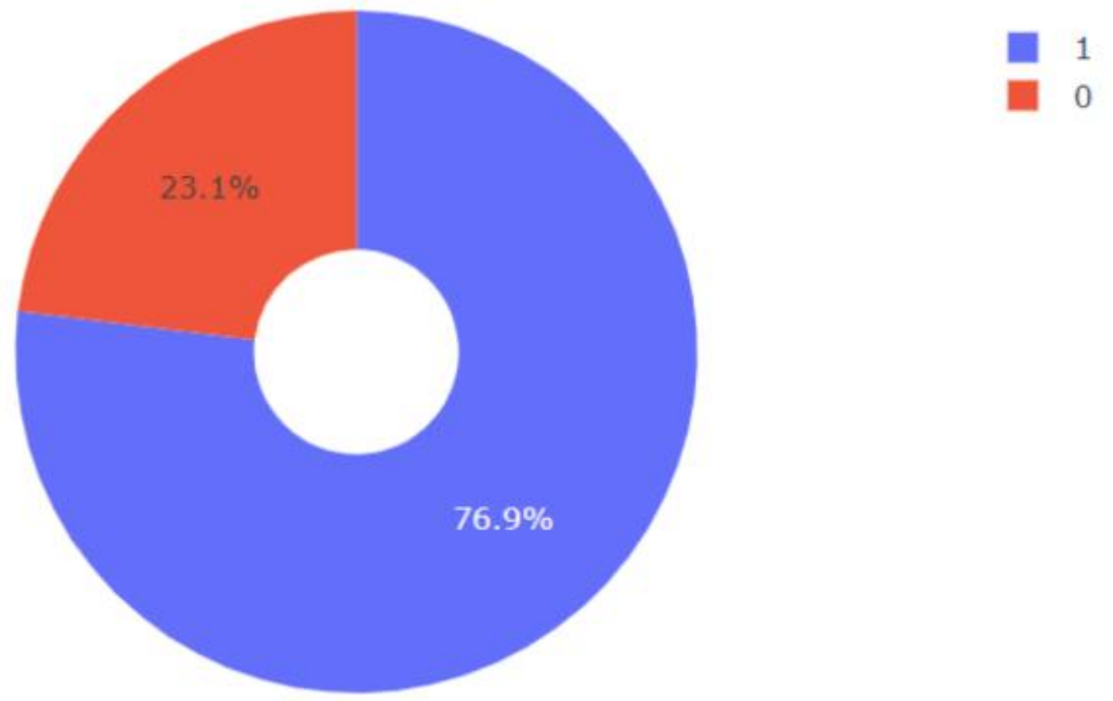
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



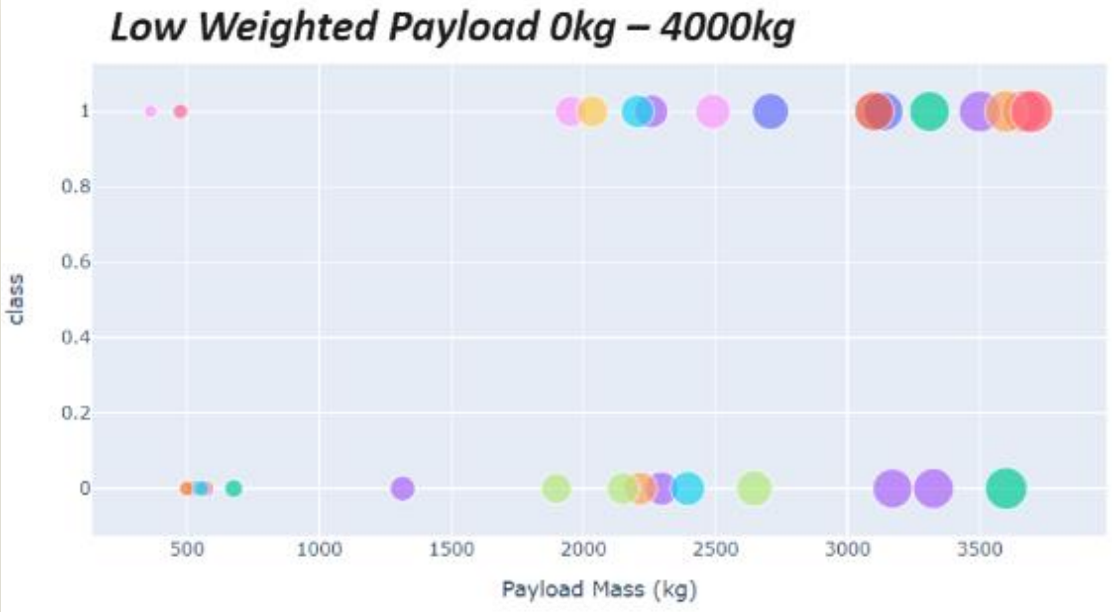
We can see that KSC LC-39A had the most successful launches from all the sites

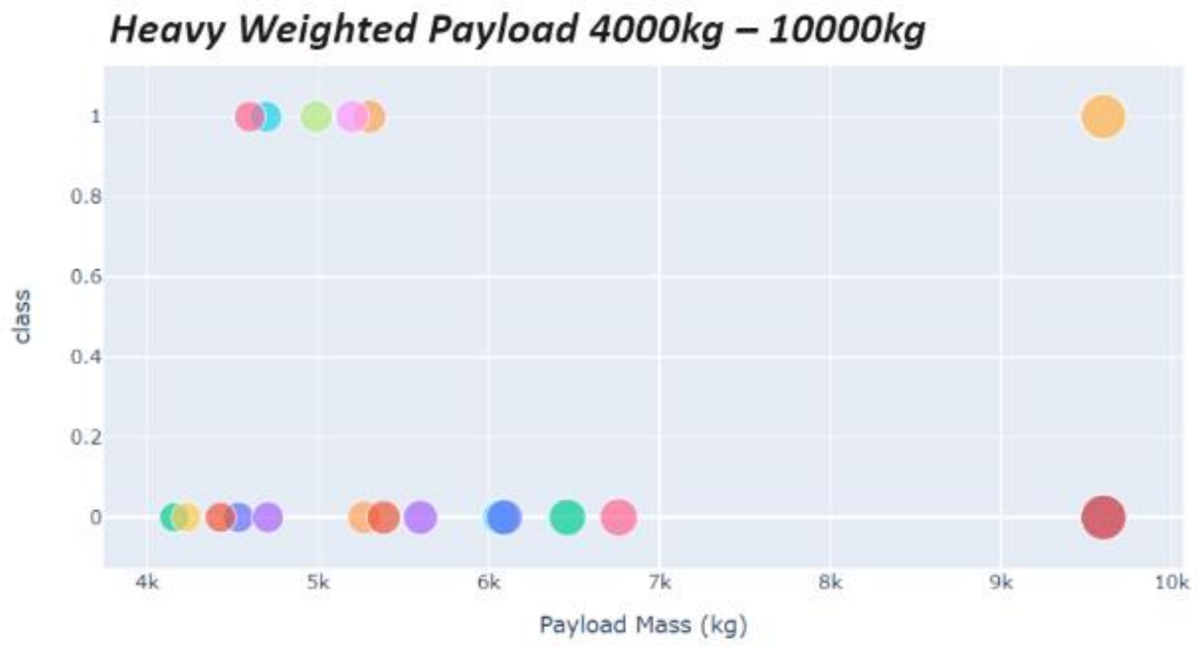
Pie chart showing the Launch site with the highest launch success ratio



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

Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider





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

Section 6

Predictive Analysis (Classification)

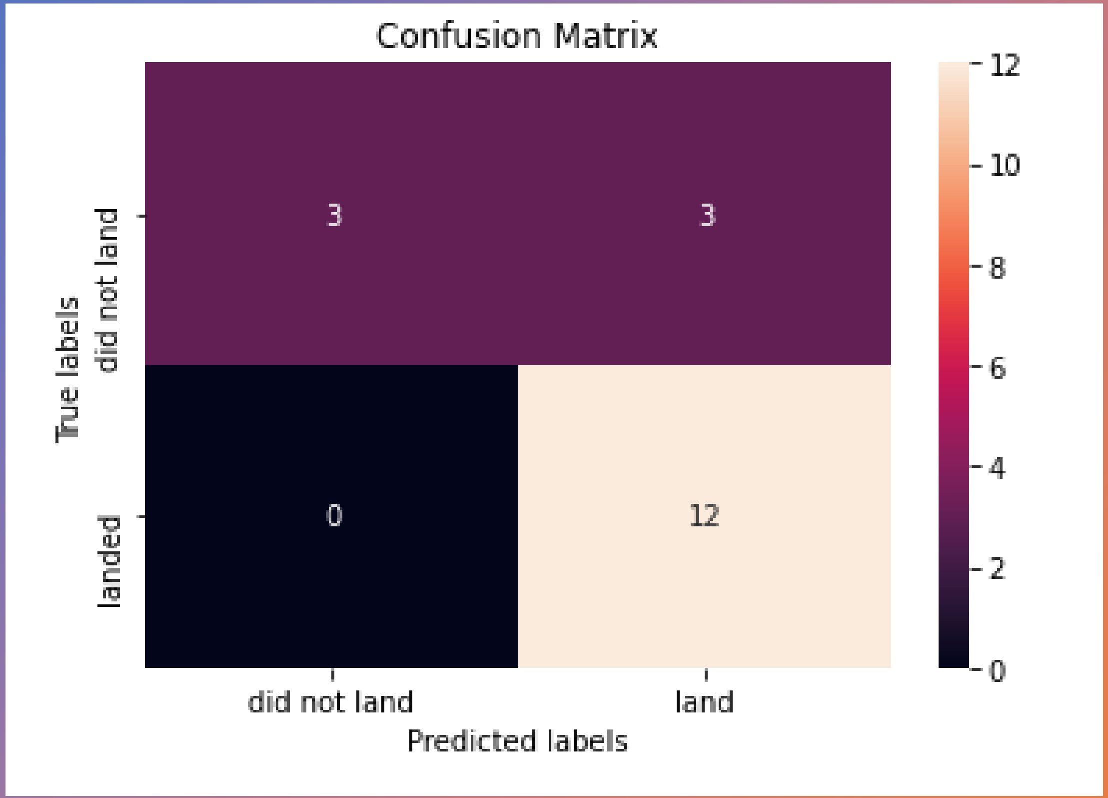
Classification Accuracy

- The decision tree classifier is the model with the highest classification accuracy

Entrée [37]:
```
models_scor = {'KNeighbors':knn_cv.best_score_,
               'DecisionTree':tree_cv.best_score_,
               'LogisticRegression':logreg_cv.best_score_,
               'SupportVector': svm_cv.best_score_}

best_method = max(models_scor, key=models_scor.get)
print('method performs best:', best_method,'with a score of', models_scor[best_method])
if best_method == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if best_method == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
if best_method == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if best_method == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
```

```
method performs best: DecisionTree with a score of 0.8767857142857143
Best params is : {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 10,
'splitter': 'random'}
```



Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.

Conclusions

We can conclude that:

The larger the flight amount at a launch site, the greater the success rate at a launch site.

Launch success rate started to increase in 2013 till 2020.

Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.

KSC LC-39A had the most successful launches of any sites.

The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!