IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Salami Lukman Bayonle
12th December, 2022

Outline

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

Executive Summary

- **Summary of methodologies**
  - Jupyter Notebooks running on python 3 was used along with IBM Watson studio and DB console.
  - 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 in screenshots
  - Predictive Analytics result

Introduction

- Project background and context

Falcon 9 is a reusable, two-stage rocket designed and manufactured by SpaceX for the reliable and safe transport of people and payloads into Earth orbit and beyond. Falcon 9 is the world's first orbital class reusable rocket. Reusability allows SpaceX to refly the most expensive parts of the rocket, which in turn drives down the cost of space access. 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. This goal of the project is to create a machine learning pipeline 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

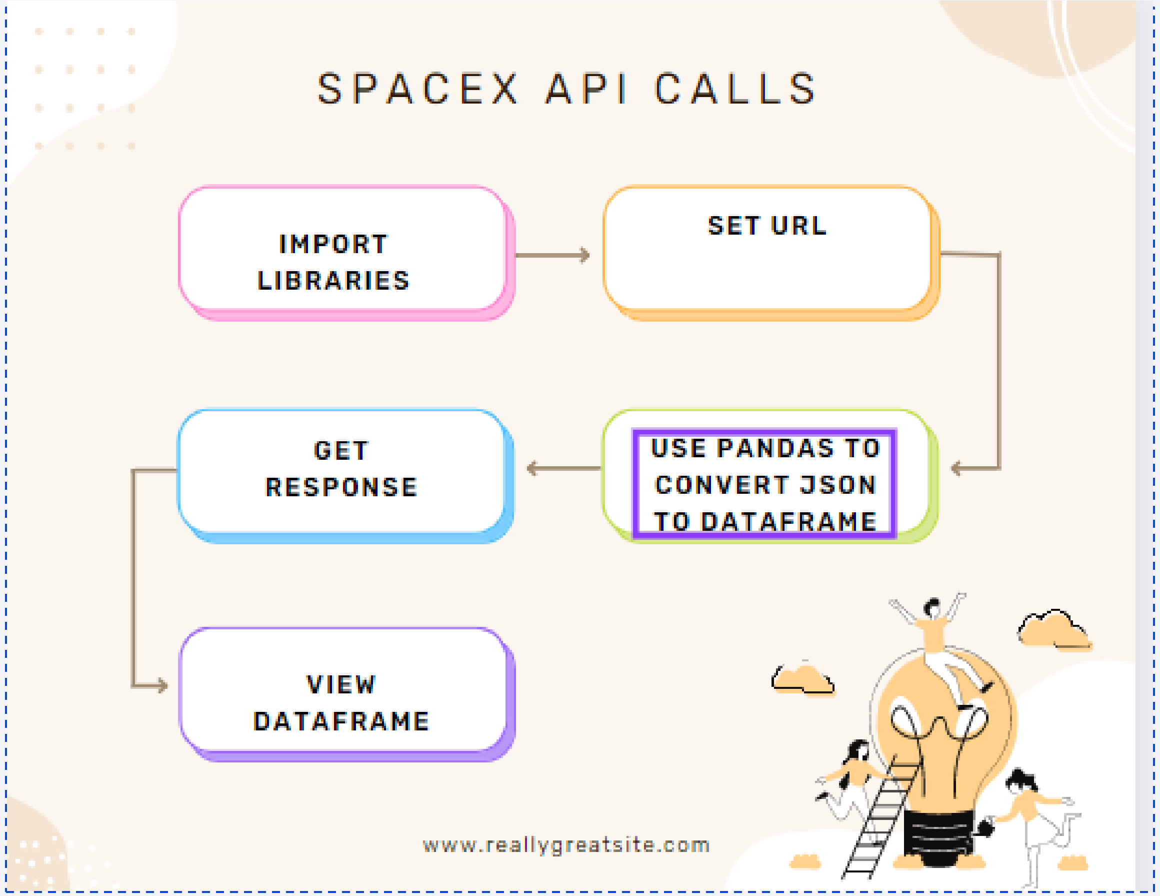
- Data collection methodology:
    - Data was collected using SpaceX API and web scraping from Wikipedia.
- Data wrangling
    - One-hot encoding was applied to categorical features
    - Standard Scaling was applied to the features.
- Exploratory data analysis (EDA) using visualization and SQL
- Interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - Data was split to train-test, Logistic Regression, KNN, SVC and decision tree were use for classification
    - GridSearch was used to select the best parameters for the models

Data Collection

- Describe how data sets were collected.
  - Data collection was done using get request to the SpaceX API.
  - Next, we decoded the response content as a Json using .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.
  - 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.

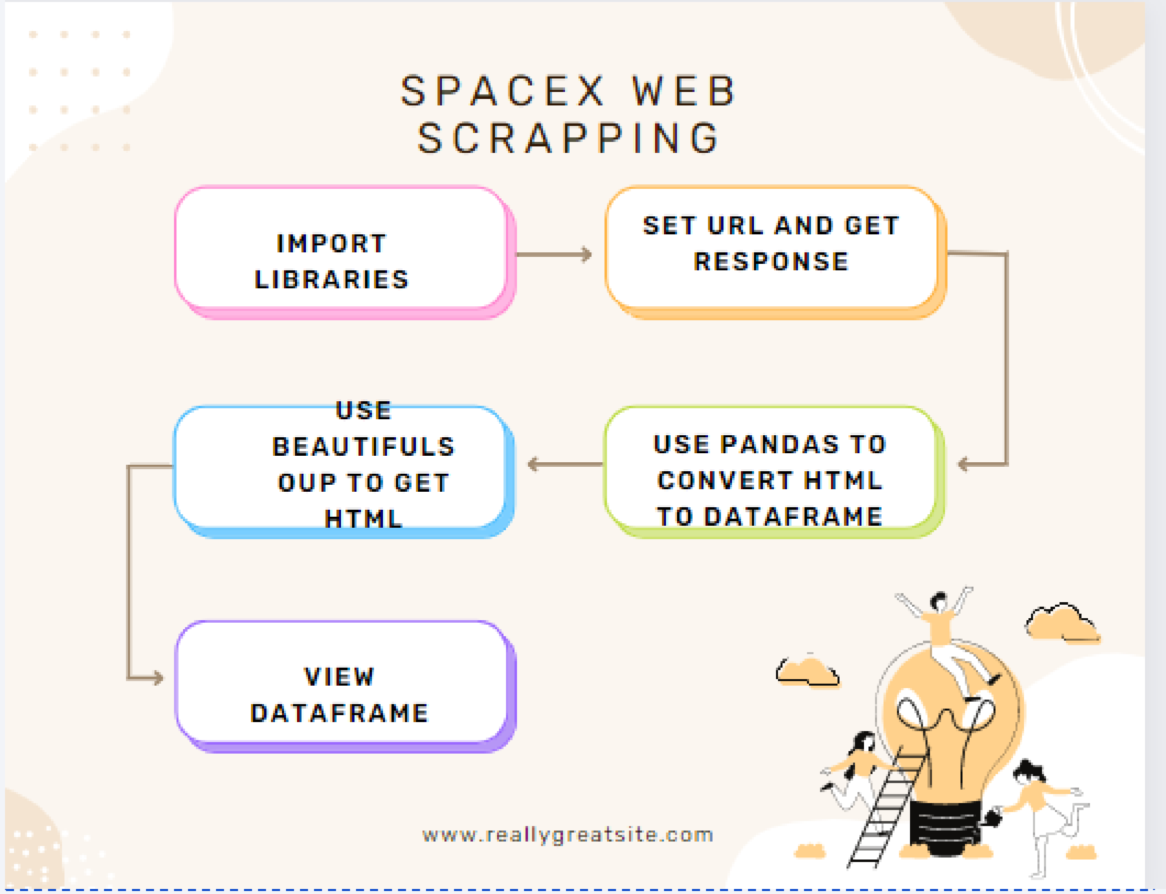
Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.
- https://github.com/bayonlelukmansalami/spacex/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



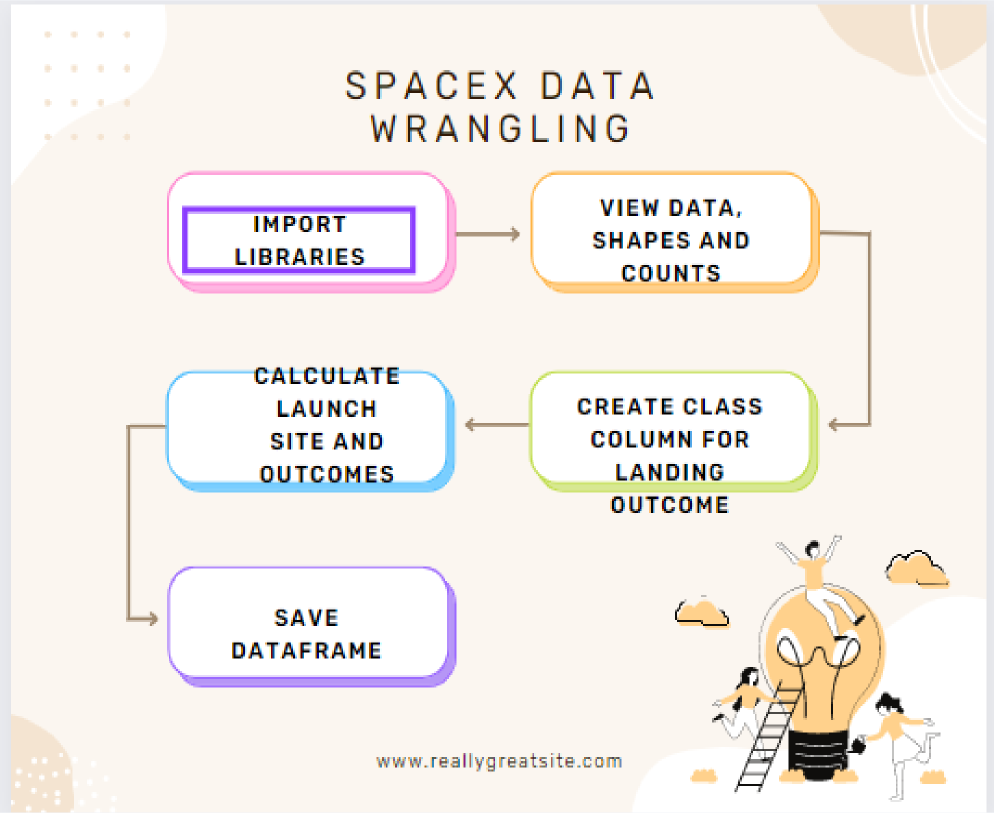
Data Collection - Scraping

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- https://github.com/bayonlelukmansalami/spacex/blob/main/jupyter-labs-webscraping1.ipynb



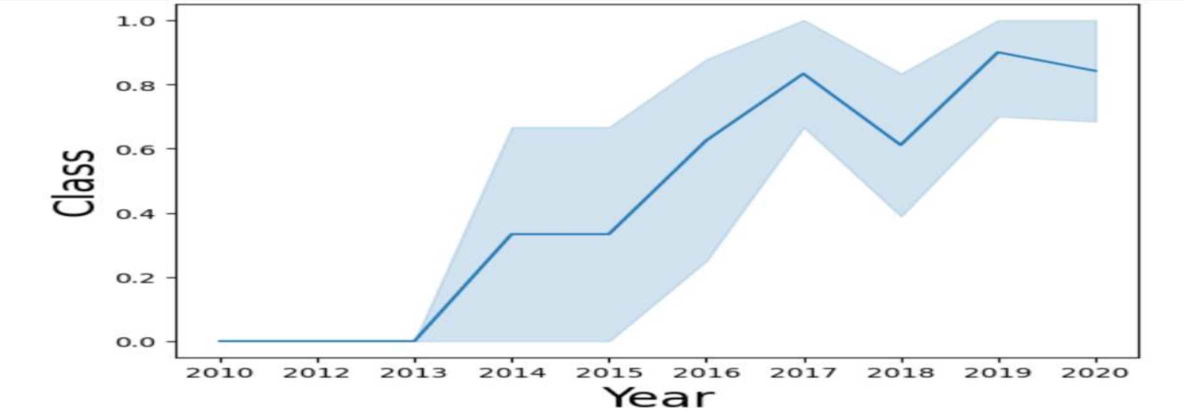
Data Wrangling

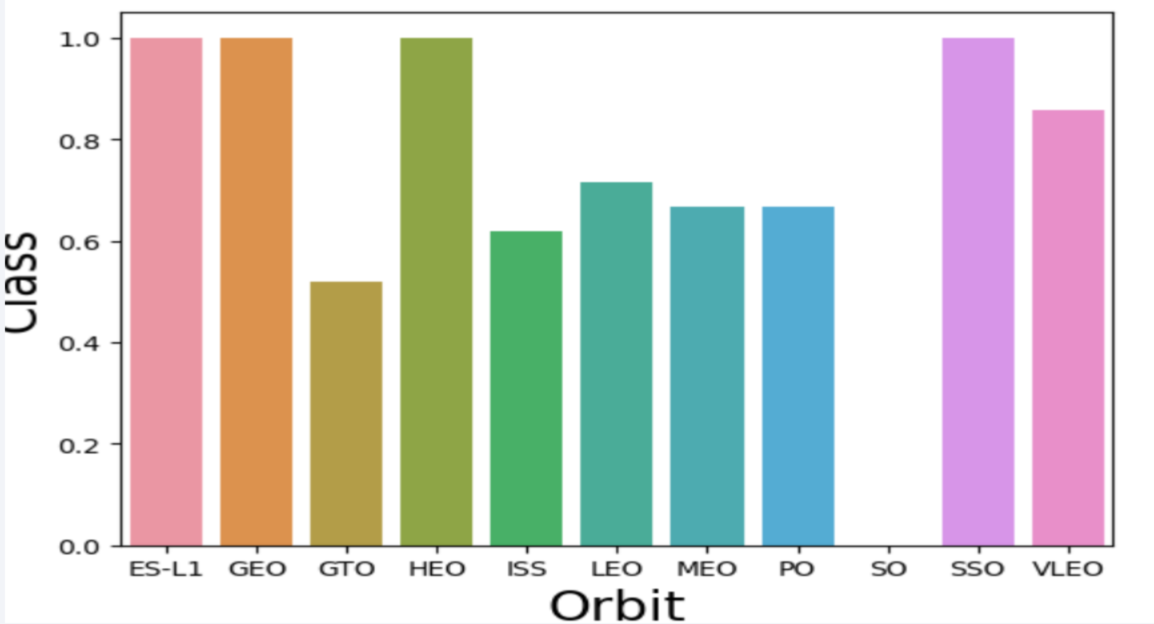
- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created Class column from landing outcome column and exported the results to csv
- https://github.com/bayonlelukmansalami/spacex/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly
- https://github.com/bayonlelukmansalami/spacex/blob/main/jupyter-labs-eda-dataviz.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/bayonlelukmansalami/spacex/blob/main/jupyter-labs-eda-sql-coursera_sqllite.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.
- https://github.com/bayonlelukmansalami/spacex/blob/main/lab_jupyter_launch_site_location.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.
- https://github.com/bayonlelukmansalami/spacex/blob/main/spacex_dash_app2.py

Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- https://github.com/bayonlelukmansalami/spacex/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

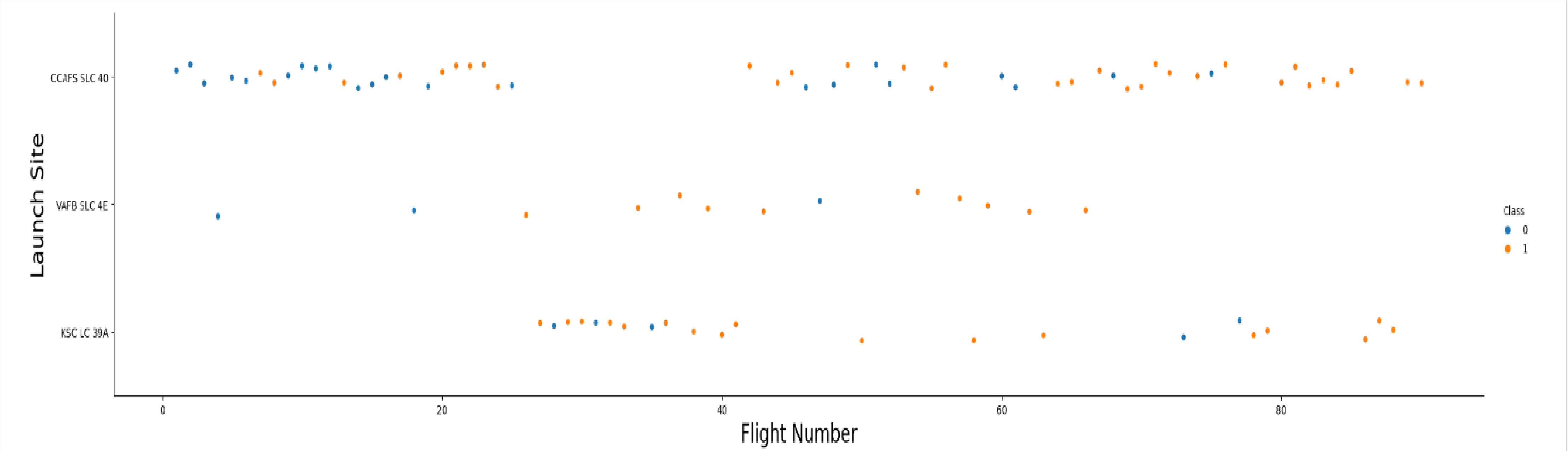
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

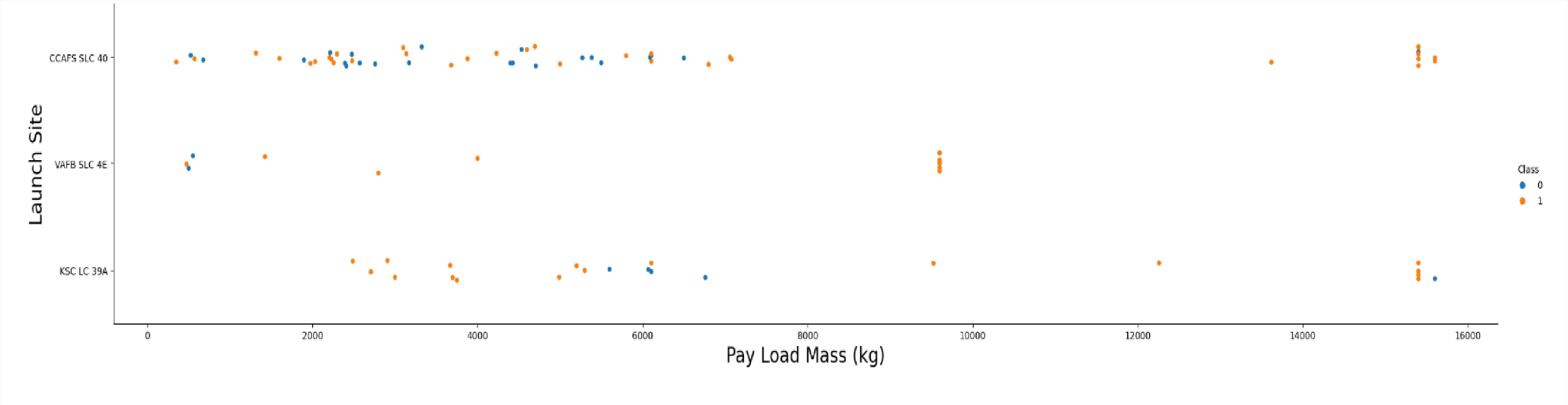
- Scatter plot of Flight Number vs. Launch Site



- 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

- Scatter plot of Payload vs. Launch Site

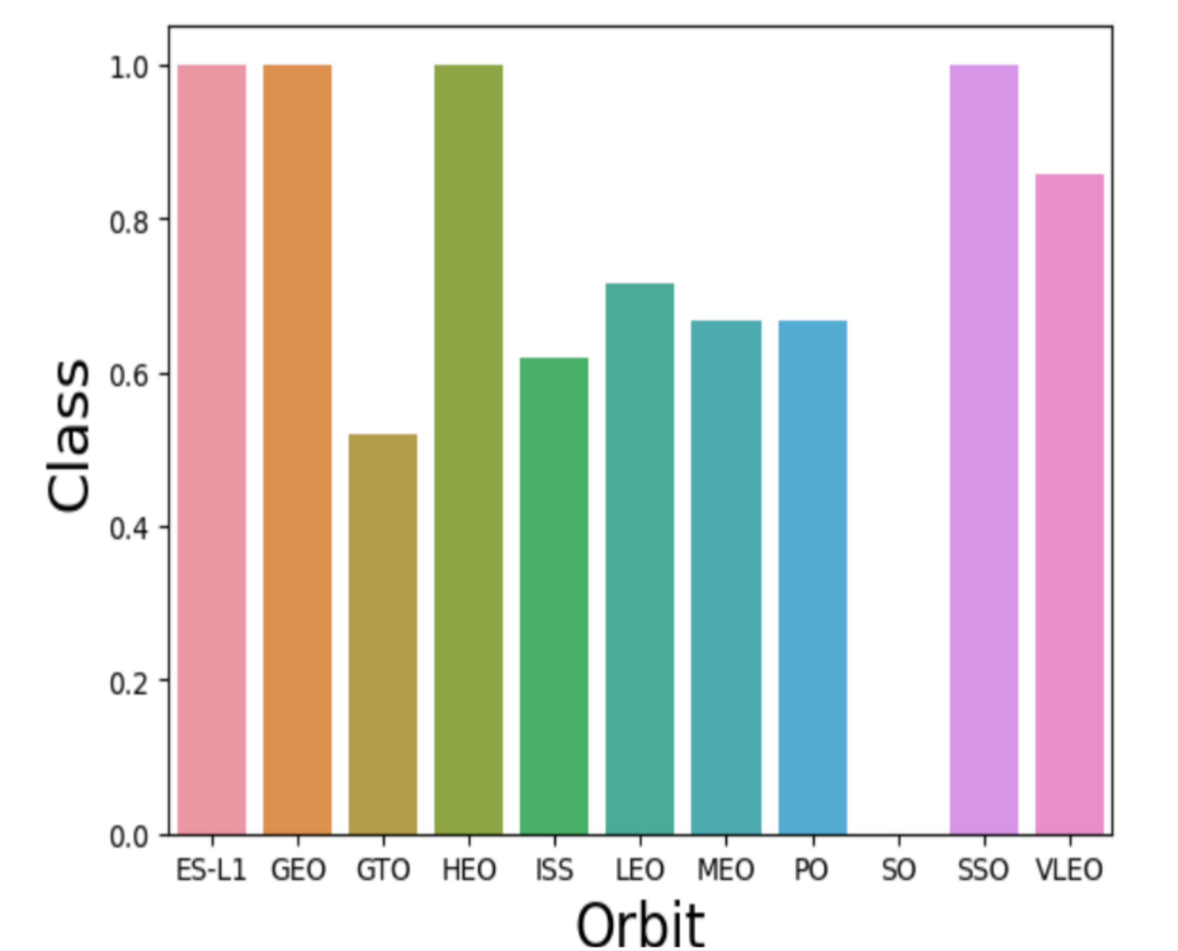


- VAFB-SLC launchsite has no rockets launched for heavypayload mass(greater than 10000) while the higher the payload mass for others the higher the success rate.

Success Rate vs. Orbit Type

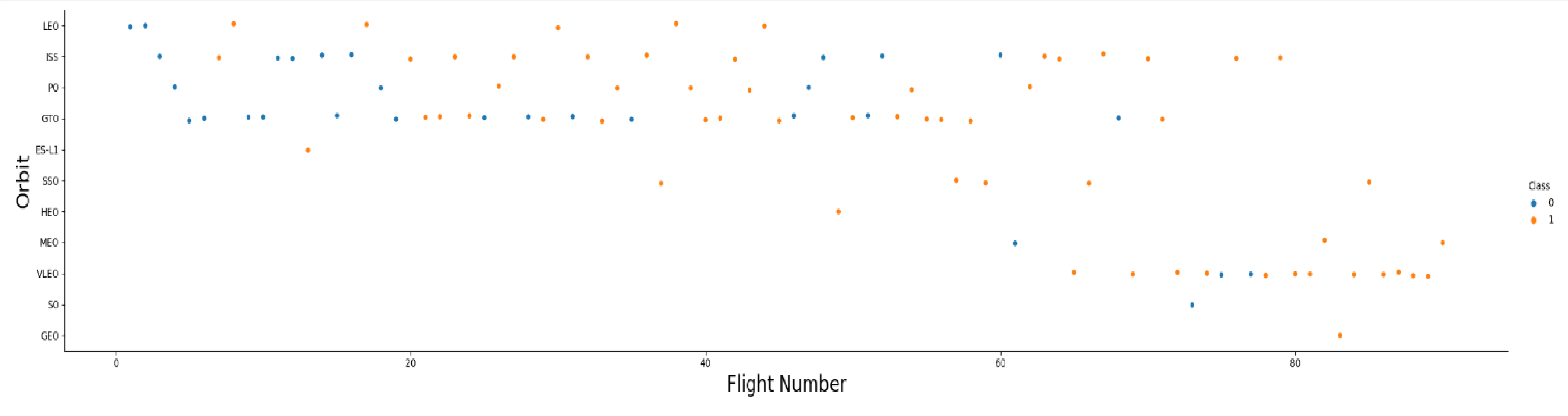
- Bar chart for the success rate of each orbit type

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



Flight Number vs. Orbit Type

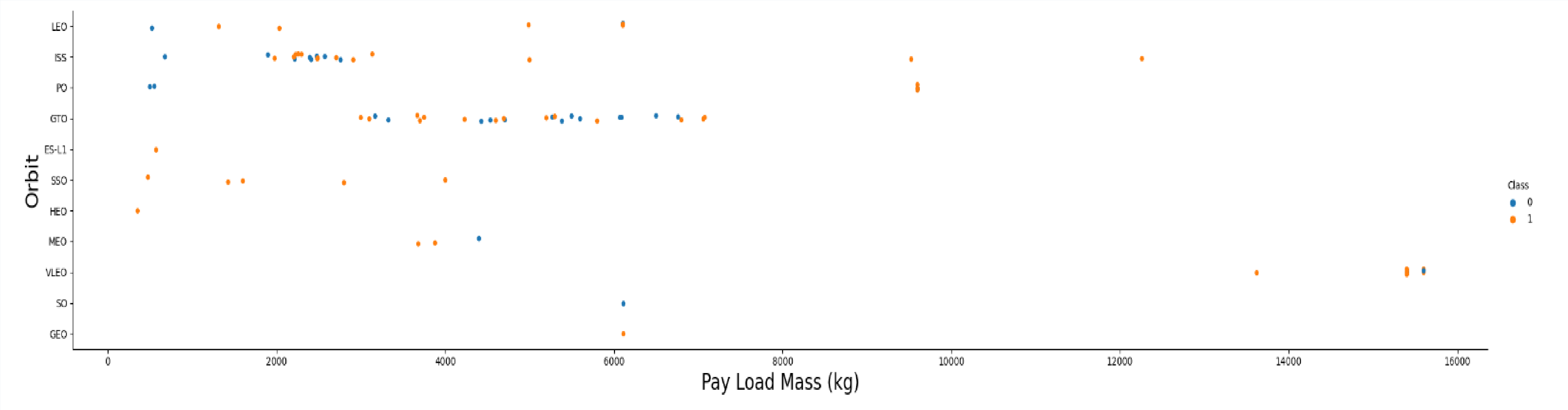
- Scatter plot of Flight number vs. Orbit type



- 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

- scatter plot of payload vs. orbit type

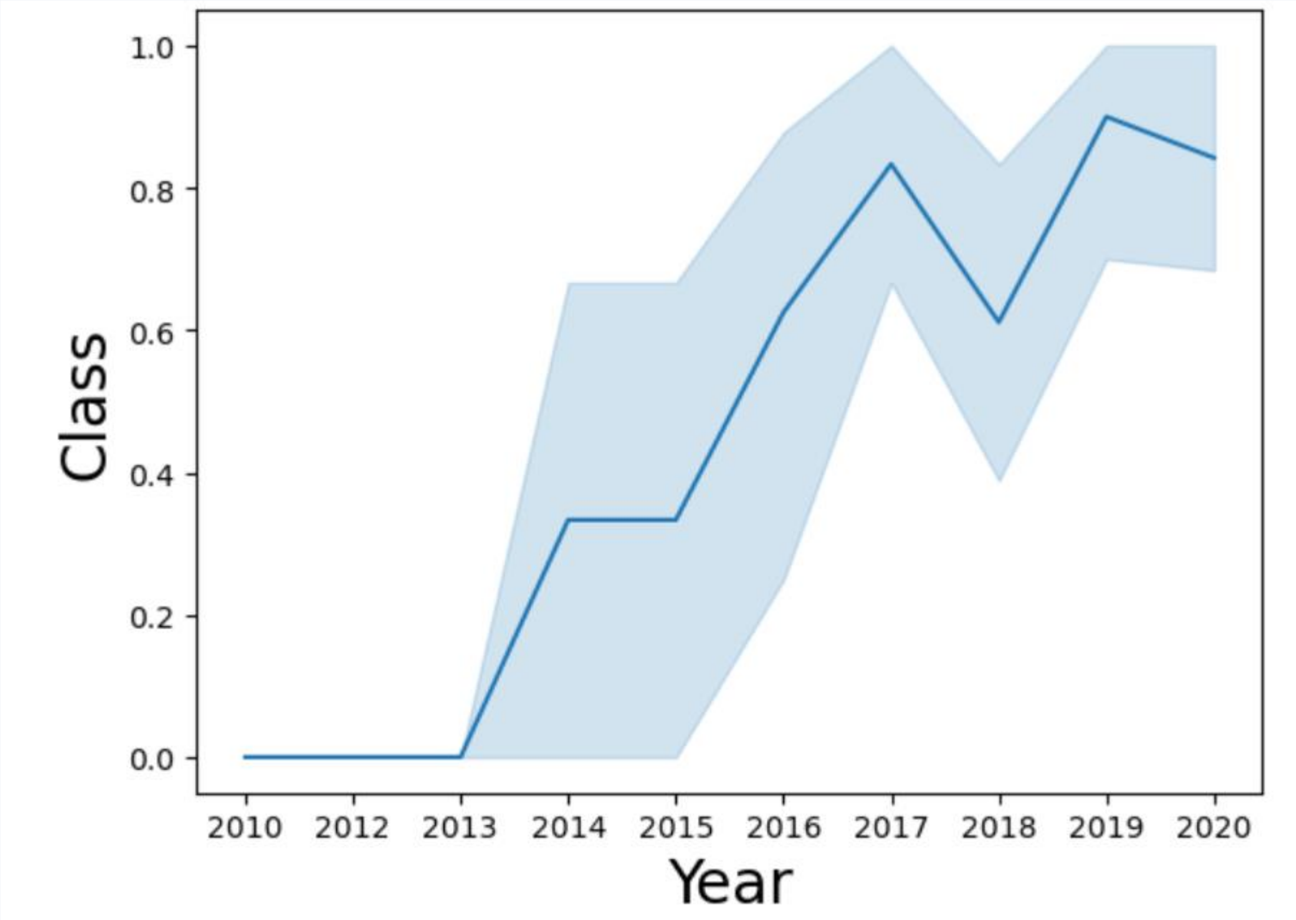


- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here

Launch Success Yearly Trend

- Line chart of yearly average success rate

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



All Launch Site Names

- We used the key word DISTINCT on EDA sqlite on python to show only unique launch sites from the SpaceX data.

```
In [16]: %sql select DISTINCT("Launch_site") from SPACEX

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

Out[16]:

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

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`

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

In [24]:
```
%sql select "Launch_site" from SPACEX WHERE "Launch_site" LIKE '%CCA%' limit 5
```

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

Out[24]:

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

- Using the SQL syntax Launch_site LIKE %CCA%, we get launch sites which begins with CCA

Total Payload Mass

- We calculated the total payload carried by boosters from NASA as 45596kg using the sum syntax on the payload mass column.

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

```
In [40]: %sql select sum(PAYLOAD_MASS__KG_) from SPACEX where "Customer" = 'NASA (CRS)'
```

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

Out[40]:

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 2928.4kg

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

In [68]:
```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEX where "Booster_Version" = 'F9 v1.1'
```

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

Out[68]:

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 22nd December 2015 using the where clause on landing outcome equals to success on ground pad

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

Hint:Use min function

In [110...

```
%%sql
select Date, "Landing _Outcome"
from SPACEX
where "Landing _Outcome" = "Success (ground pad)"
limit 1
```

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

Out[110...

Date	Landing _Outcome
22-12-2015	Success (ground pad)

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** condition to determine successful landing with payload mass greater than 4000 but less than 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 [118...

```
%%sql
select "Booster_Version", "Landing _Outcome", "PAYLOAD_MASS__KG_"
from SPACEX
where "Landing _Outcome" = "Success (drone ship)" and "PAYLOAD_MASS__KG_" > 4000 and "PAYLOAD_MASS__KG_" <6000
```

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

Out[118...

Booster_Version	Landing _Outcome	PAYLOAD_MASS__KG_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

Total Number of Successful and Failure Mission Outcomes

- We use the group by Mission Outcome to get the total number of successful and failure mission outcomes

List the total number of successful and failure mission outcomes

In [140…

```
%%sql
select "Mission_Outcome", count("Mission_Outcome") as Total
from SPACEX
group by "Mission_Outcome"
```

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

Out[140…

Mission_Outcome	Total
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

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

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

[143...

```
%%sql
select "Booster_Version", PAYLOAD_MASS__KG_
from SPACEX
where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEX)
```

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

t[143...

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

2015 Launch Records

- We extracted the month and year, then used a combinations of the **WHERE** clause and **AND** clause conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

In [154…

```
%%sql
select substr(Date,4,2) as month,substr(Date,7,4) as year, "Landing _Outcome", "Booster_Version", "Launch_Site"
from SPACEX
where "Landing _Outcome" = "Failure (drone ship)" and year = '2015'
```

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

Out[154…

month	year	Landing _Outcome	Booster_Version	Launch_Site
01	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	2015	Failure (drone ship)	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

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad))

In [19]:

```
task_10 = '''
        SELECT LandingOutcome, COUNT(LandingOutcome)
        FROM SpaceX
        WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
        GROUP BY LandingOutcome
        ORDER BY COUNT(LandingOutcome) DESC
        '''
create_pandas_df(task_10, database=conn)
```

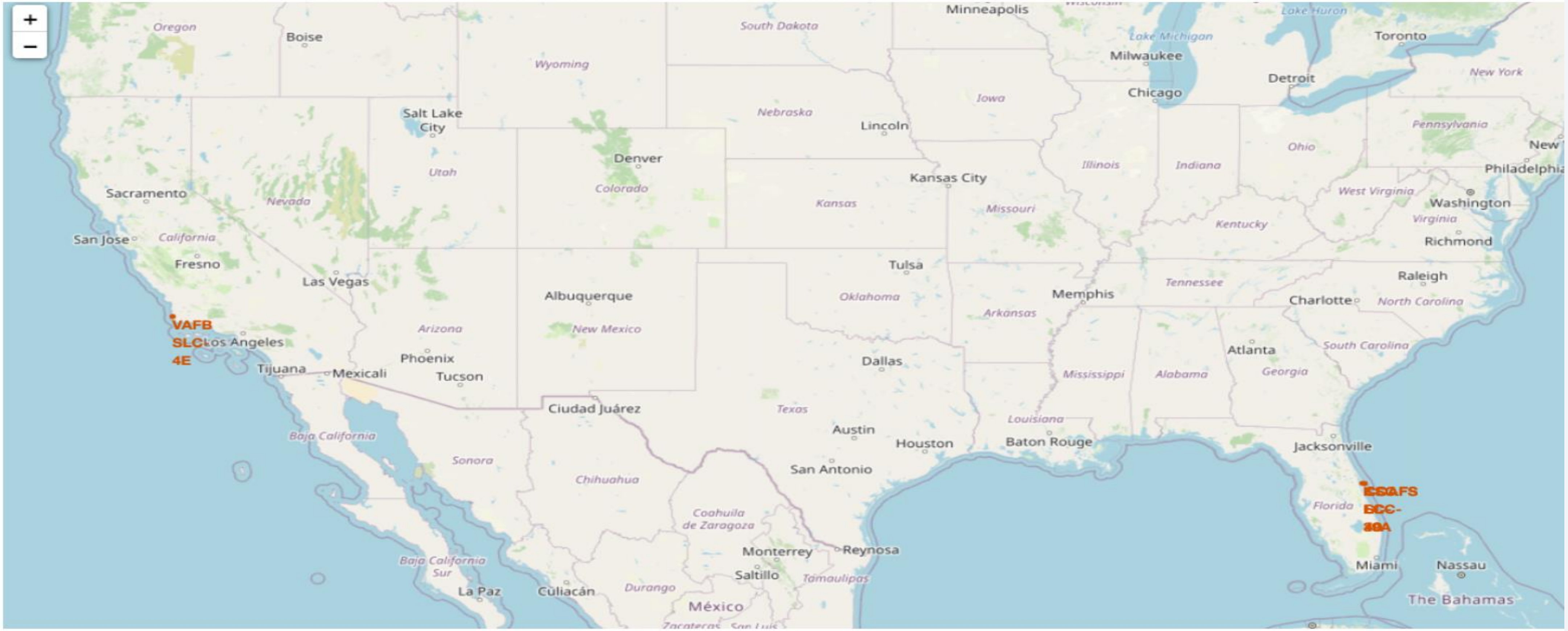
Out[19]:

	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

Section 3

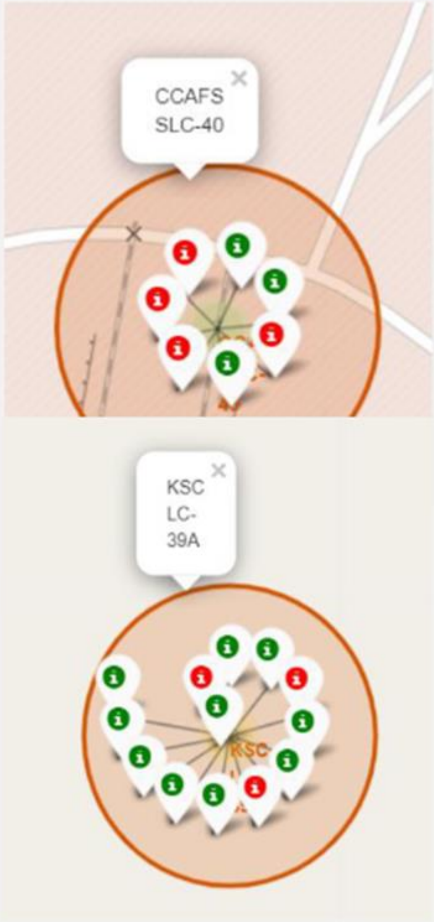
Launch Sites Proximities Analysis

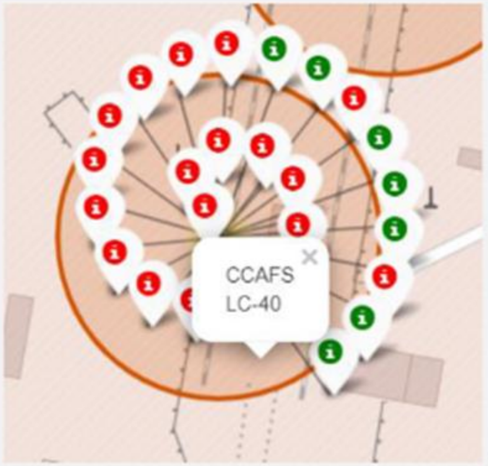
SPACEX LAUNCH SITES



- SpaceX launch sites are located along the USA coastlines.

Markers showing launch sites with color labels





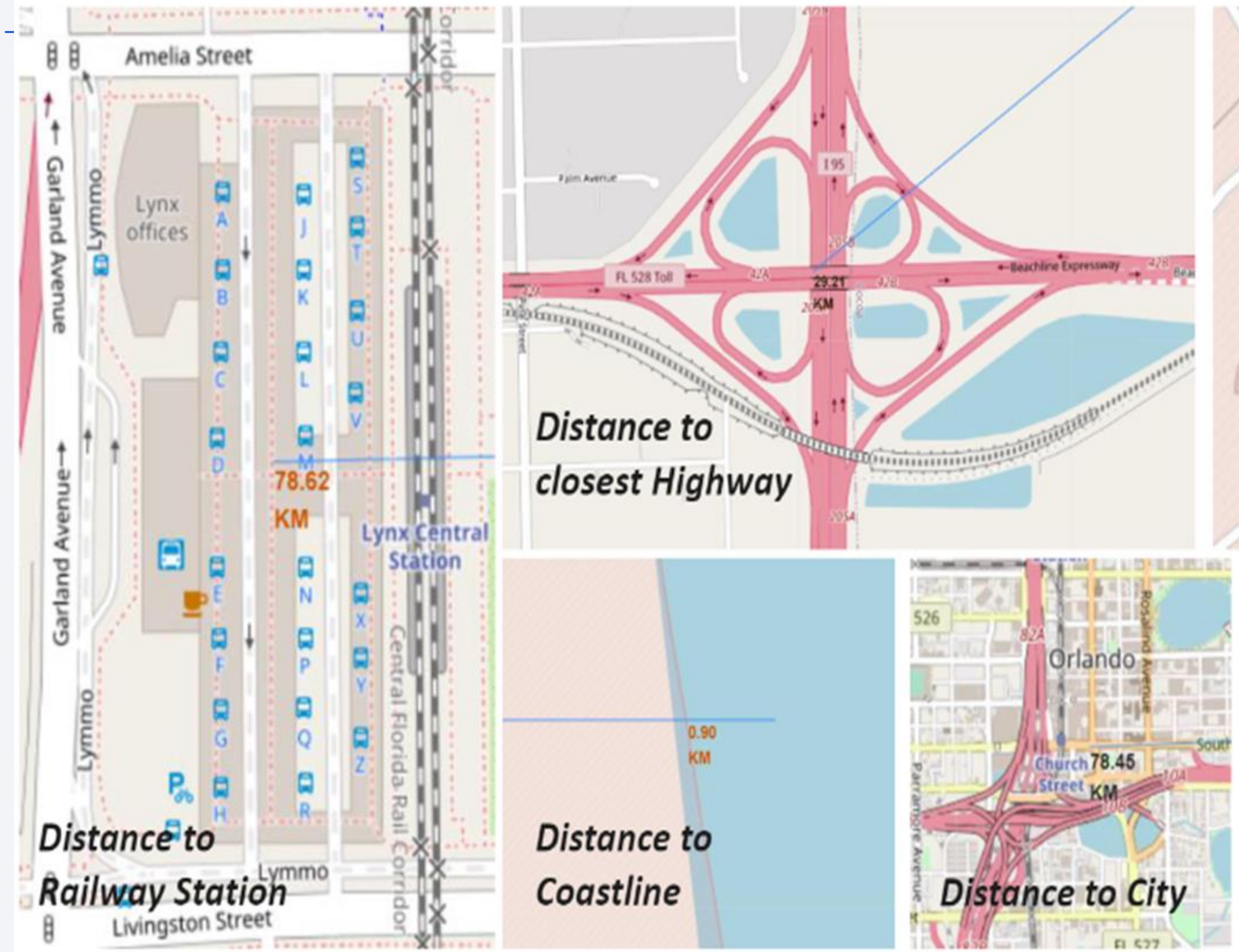
Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

37

Launch Site distance to landmarks

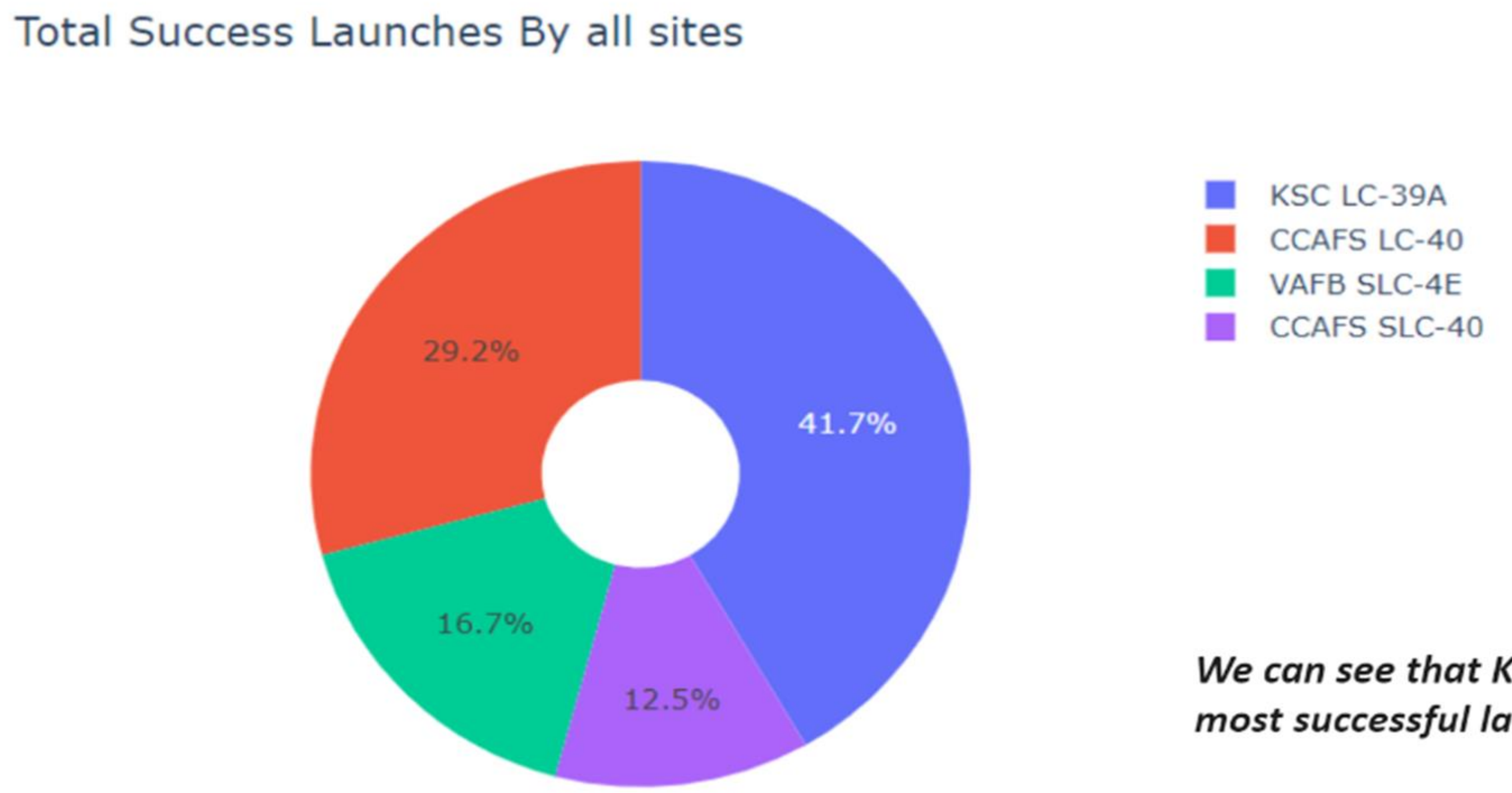


- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

Section 4

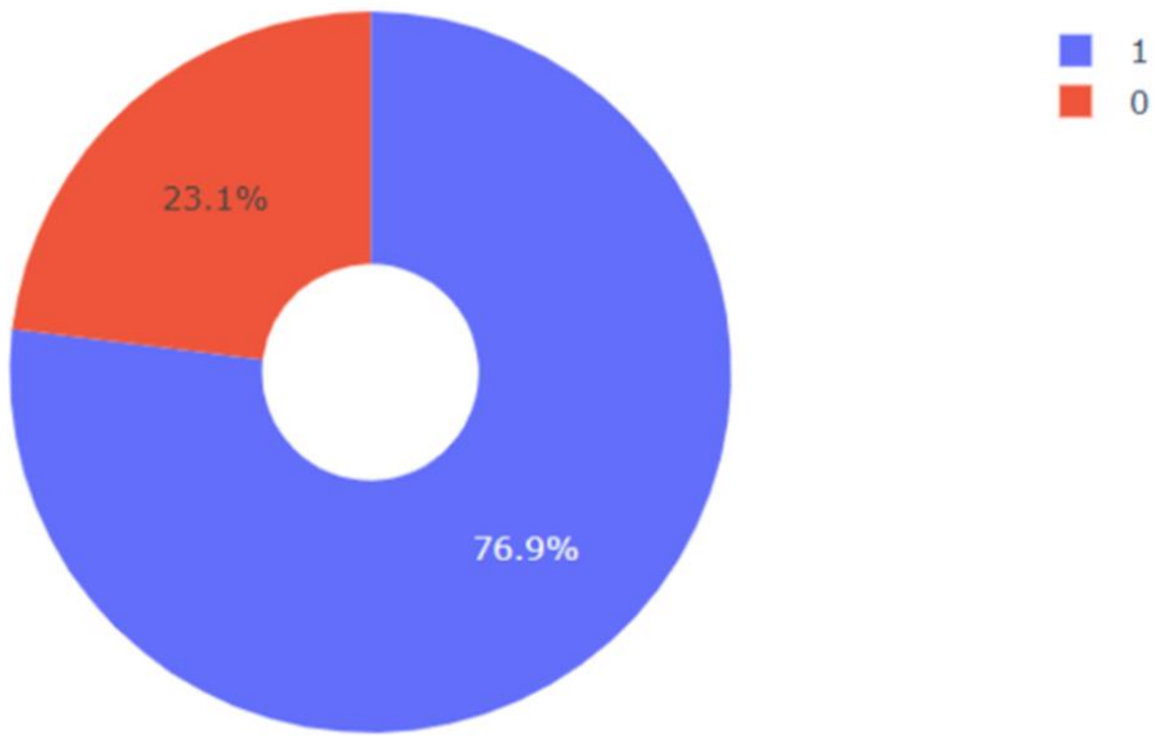
Build a Dashboard with Plotly Dash

SPACEX LAUNCH RECORDS DASHBOARD



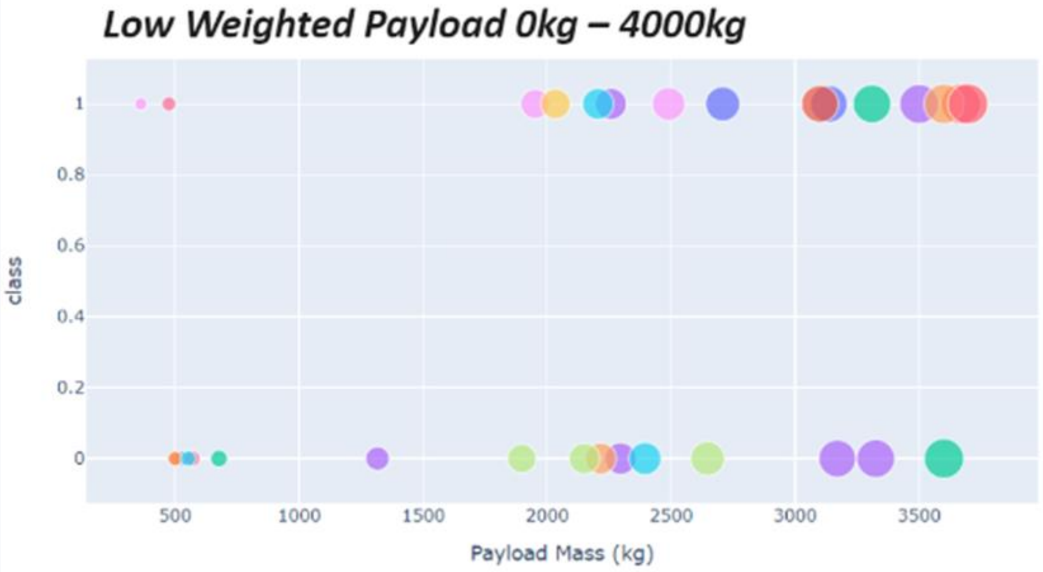
We can see that KSC LC-39A had the most successful launches from all the sites

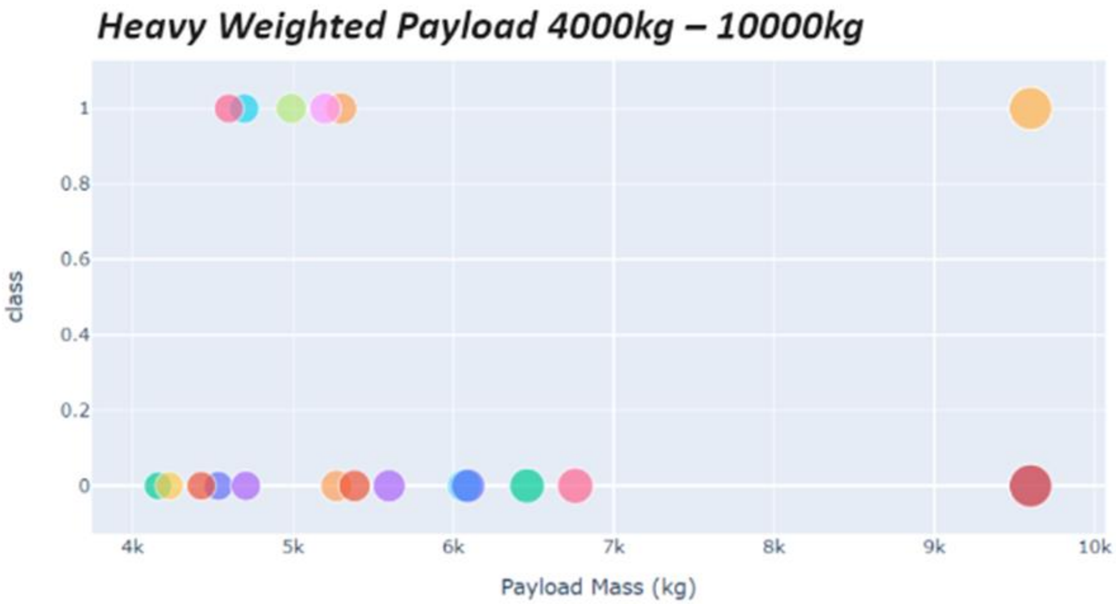
Pie chart showing the Launch site with the highest launch success ratio



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





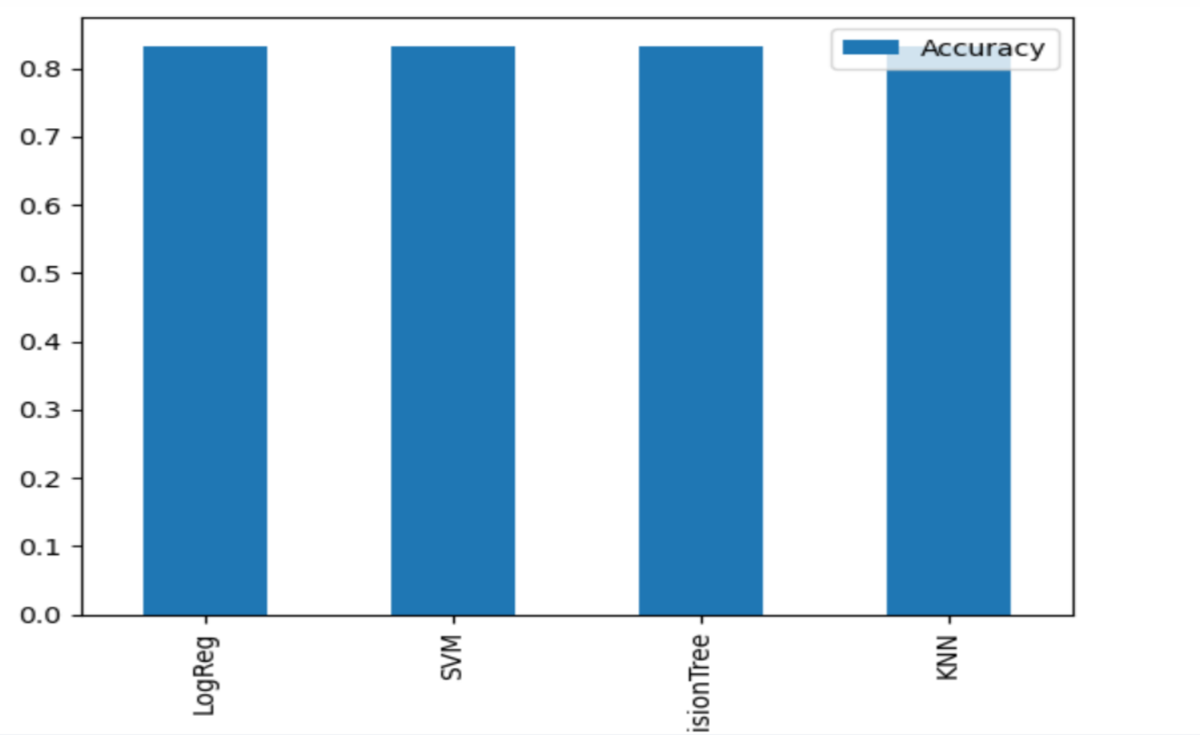
We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

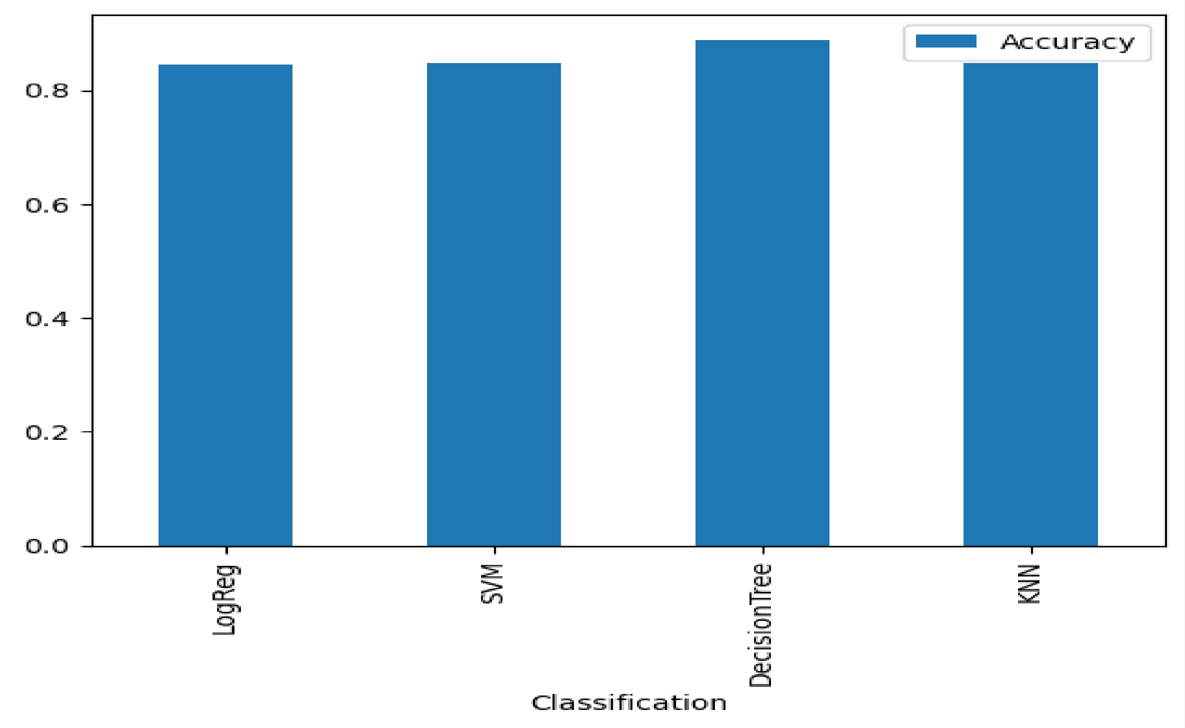
Section 5

Predictive Analysis (Classification)

Classification Accuracy

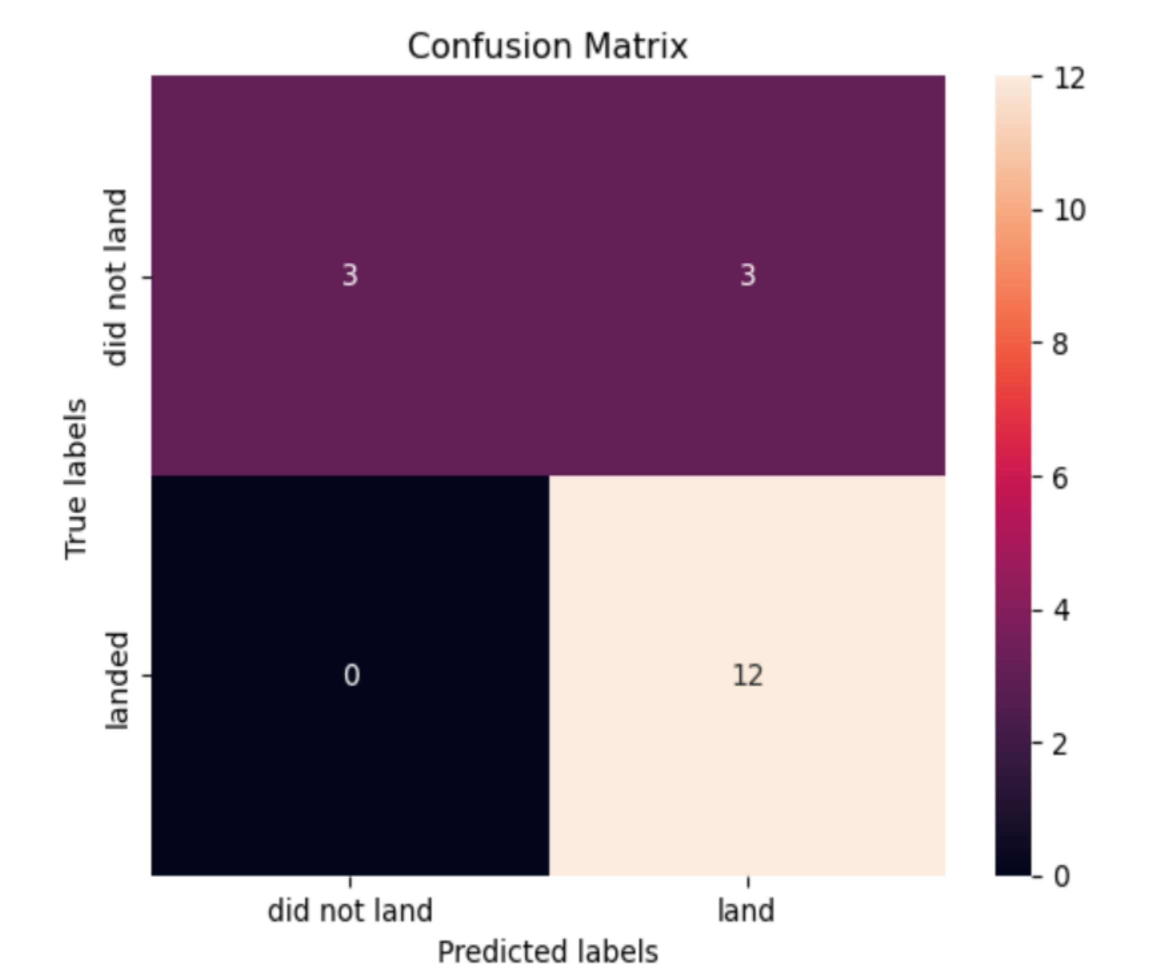
- Decision Tree performs better on training set.
- They all have the same accuracy on the validation dataset.





Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can classify successful landing(True Positives) 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 higher the number of flights at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013.
- 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.

Appendix

- dataset_part_1,2,3
- mySpaceX.csv
- lab_jupyter_launch_site_location
- labs-jupyter-spacex-Data wrangling
- jupyter-labs-eda-dataviz
- jupyter-labs-eda-sql-coursera_sqllite
- jupyter-labs-spacex-data-collection-api
- jupyter-labs-spacex-data-collection-api
- spacex_dash_app2

Thank you!