IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Tomas Iriarte
June 16, 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection API
  - Data collection Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis (EDA) with SQL
  - EDA with Visualization
  - Interactive Visual Analytics with Folium
  - Interactive Visual Analytics with Plotly Dash
  - Predictive Analysis with Machine Learning
- Summary of all results

Introduction

- Project background and context

I will take the role of a data scientist working for a new rocket company Space Y that would like to compete with SpaceX. 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. We will determine the price of each launch and if SpaceX will reuse the first stage. Instead of using rocket science to determine if the first stage will land successfully, we will train a machine learning model and use public information to predict if SpaceX will reuse the first stage.

- Problems you want to find answers
  - Which factors determine if the first stage will land?
  - Price of each launch, and the variables it depends on

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia
- Perform data wrangling
- 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

- Describe how data sets were collected.
    - Request to the SpaceX API
    - Clean the requested data
    - Extract launch records HTML table from Wikipedia
    - Parse the table and convert it into a Pandas data frame

Data Collection – SpaceX API

- We request and parse the SpaceX launch data using the GET request

- https://github.com/TomasIriarte/data_science-final_project/blob/main/10w1%20SpaceX%20Falcon%209%20first%20stage%20Landing%20Prediction%20Lab1%20Collecting%20the%20data.ipynb

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/AP
```

We should see that the request was successfull with the 200 status response code

```
response.status_code
```

```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

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

Data Collection - Scraping

- Request the Falcon9 Launch Wiki page from its URL
- Extract all column/variable names from the HTML table header
- Create a data frame by parsing the launch HTML tables
- https://github.com/TomasIriarte/data_science-final_project/blob/main/10w1%20Space%20X%20Falcon%209%20First%20Stage%20Landing%20Prediction%20Web%20scraping%20Falcon%209%20and%20Falcon%20Heavy%20Laun

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
# use requests.get() method with the provided static_url
# assign the response to a object
data  = requests.get(static_url).text
```

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data, "html.parser")
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
# Use soup.title attribute
soup.title
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

Data Wrangling

- We performed some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models
- We calculated the number of launches on each site
- We calculated the number and occurrence of each orbit
- We calculated the number and occurrence of mission outcome per orbit type
- We created a landing outcome label from Outcome column
- https://github.com/TomasIriarte/data_science-final_project/blob/main/10w1%20Space%20X%20Falcon%209%20First%20Stage%20Landing%20Prediction%20Lab%202%20Data%20Wrangling.ipynb

EDA with Data Visualization

- The EDA was performed by visualizing the data, plotting the relationship between Flight number and Payload mass, Flight number and Launch site, Payload and Launch site, Flight number and Orbit type, Payload and Orbit type using scatter point chart, and the sucess rate of each orbit using a bar chart
- https://github.com/TomasIriarte/data_science-final_project/blob/main/10w2%20SpaceX%20%20Falcon%209%20First%20Stage%20Landing%20Prediction%20Assignment%20Exploring%20and%20Preparing%20Data.ipynb

EDA with SQL

- We performed EDA with SQL to get insights from the data:
  - 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 succesful 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
  - And many more in the link below:
- https://github.com/TomasIriarte/data_science-final_project/blob/main/10w2%20Assignment%20SQL%20Notebook%20for%20Peer%20Assignment.ipynb

Build an Interactive Map with Folium

With Folium we

- Mark all launch sites on a map
- Mark the success/failed launches for each site on the map
- Calculate the distances between a launch site to its proximities
- Using Circles, Markers, Clusters, Polyline

https://github.com/TomasIriarte/data_science-final_project/blob/main/10w3%20Launch%20Sites%20Locations%20Analysis%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

We added to the dashboard:

- A Launch Site Drop-down Input Component
- A callback function to render success pie chart based on selected site dropdown
- A Range Slider to Select Payload
- A callback function to render the success payload scatter chart scatter plot

https://github.com/TomasIriarte/data_science-final_project/blob/main/10w3%20Build%20a%20Dashboard%20Application%20with%20Plotly%20Dash.py

Predictive Analysis (Classification)

We performed exploratory Data Analysis and determined Training Labels

- Create a column for the class
- Standardize the data
- Split into training data and test data
- Find best Hyperparameter for SVM, Classification Trees and Logistic Regression
- Find the method performs best using test data

https://github.com/TomasIriarte/data_science-final_project/blob/main/10w4%20SpaceX%20%20Falcon%209%20First%20Stage%20Landing%20Prediction%20Assignment%20Machine%20Learning%20Prediction.ipynb

Results

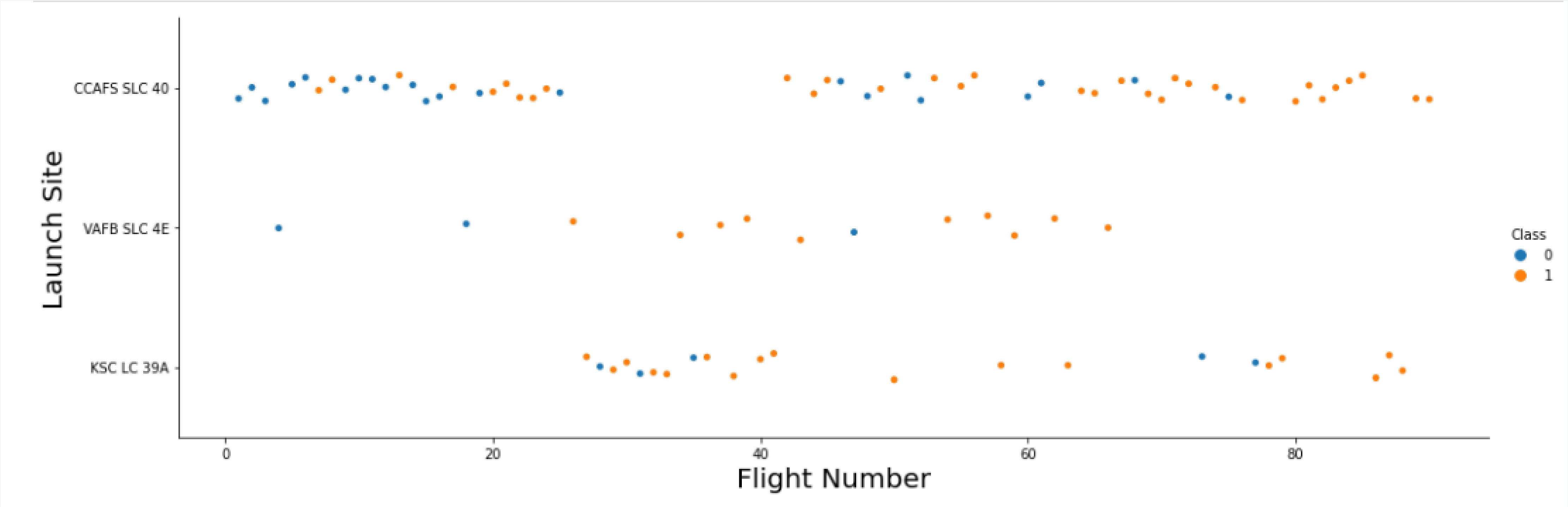
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

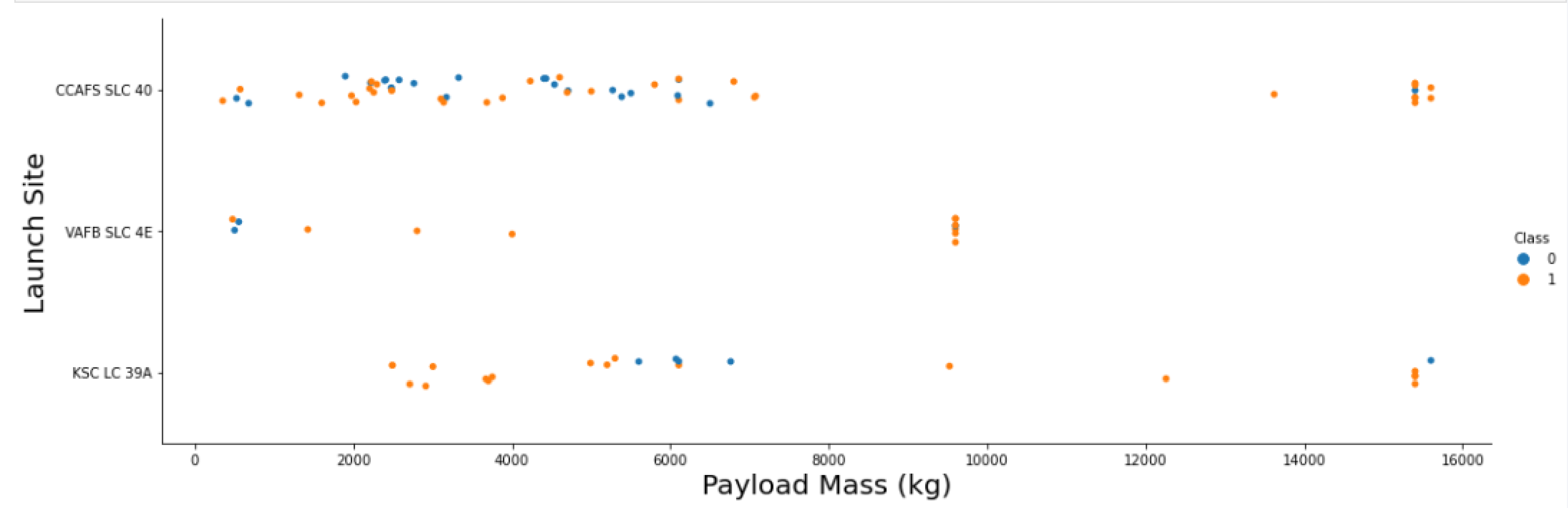
Flight Number vs. Launch Site

- Observing the plot, we can see that the larger the flight number at a launch site, the greater the success rate will be.



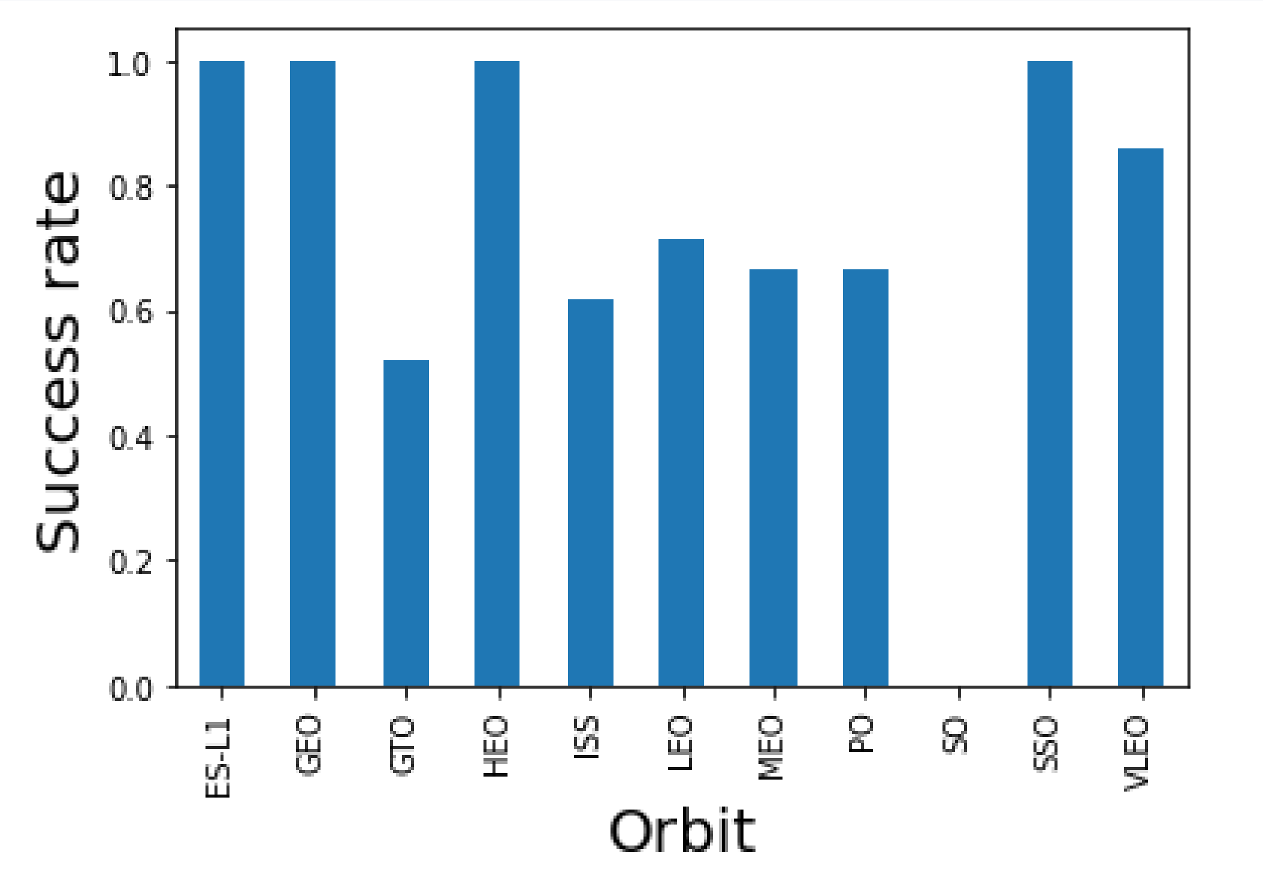
Payload vs. Launch Site

- We can see that for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass



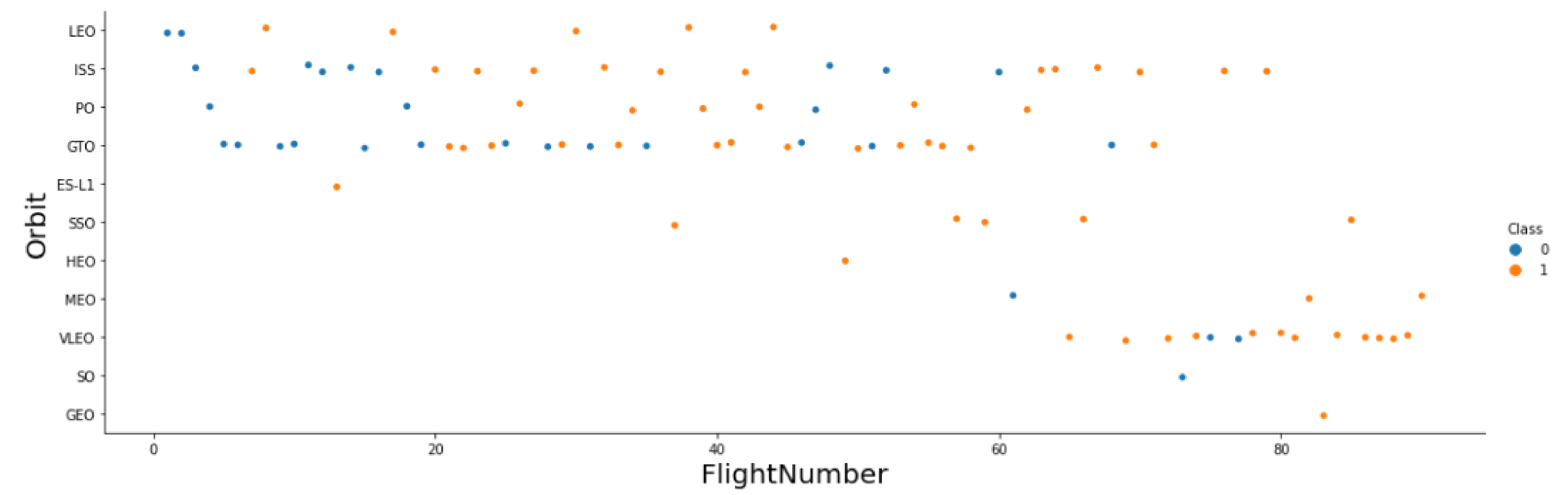
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO, VLEO had the highest success rates



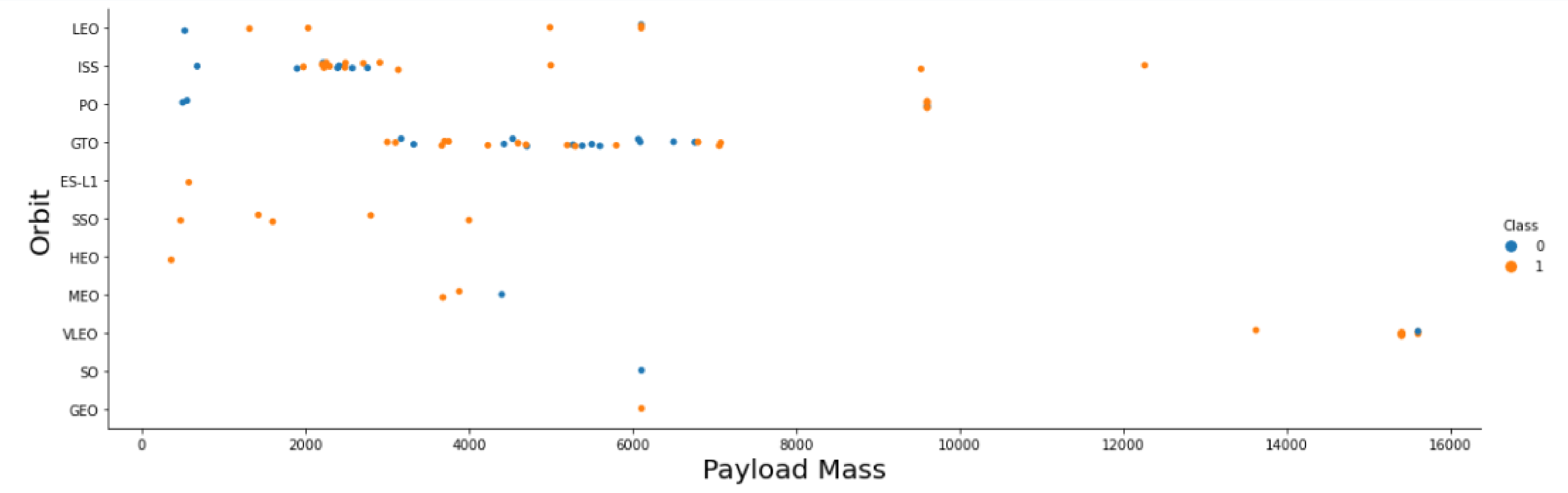
Flight Number vs. Orbit Type

- We can see that in the LEO orbit, success improves with the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit. For the ISS and PO orbits the success tends to be higher with the number of flights



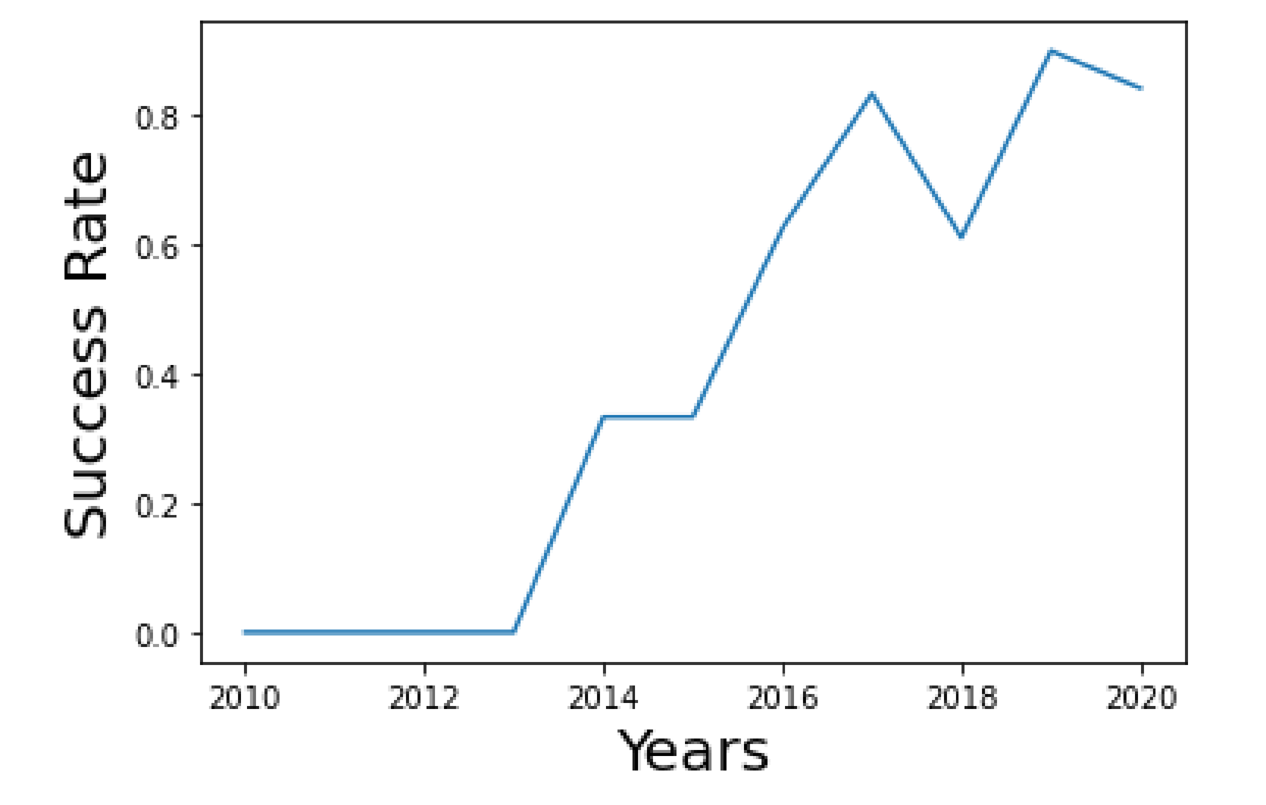
Payload vs. Orbit Type

- We observe a slight tendency of the flights to become more successful the bigger the payload mass for Polar, LEO and ISS. For GTO we cannot distinguish this well as both positive landing rate and negative landing are both there



Launch Success Yearly Trend

- The sucess rate since 2013 kept increasing till 2020



All Launch Site Names

- We used **DISTINCT** to select each launch site only once

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

```
%sql select Distinct(Launch_Site) from SPACEXTBL;
```

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

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

Launch Site Names Begin with 'CCA'

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

```
%sql select * from SPACEXTBL WHERE Launch_Site LIKE '%CCA%' Limit 5;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

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

```
%sql select SUM(PAYLOAD_MASS__KG_) from SPACEXTBL WHERE Payload LIKE '%CRS%';
```

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

SUM(PAYLOAD_MASS__KG_)
111268

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1. We used AVG to calculate the average

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

```
%sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTBL WHERE Booster_Version LIKE '%F9 v1.1%';
```

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad. This date was 22nd December 2015

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

Hint:Use min function

```
%sql
#%sql select MIN(Date) from SPACEXTBL WHERE "Landing_Outcome" LIKE '%RTLS%' and Mission_Outcome LIKE 'Success';
#%sql SELECT Date from SPACEXTBL WHERE "Landing_Outcome" LIKE '%parachute%' and Mission_Outcome LIKE 'Success';
#%sql select "Date" from SPACEXTBL WHERE "Landing_Outcome" like "%parachute%" and "Mission_Outcome" like "%Success%";
```

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

Date

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000. We used WHERE to establish the conditions.

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 Booster_Version from SPACEXTBL where Mission_Outcome like 'Success' and PAYLOAD_MASS__KG_>4000 and PAYLOAD_MASS__KG_ < 6000;
```

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

Booster_Version
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1014
F9 v1.1 B1016
F9 FT B1020

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes. We used count to show the total number of each possibility

List the total number of successful and failure mission outcomes

```
%sql select Mission_Outcome, count(Mission_Outcome) from SPACEXTBL group by Mission_Outcome ;
```

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

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

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass. To perform this task we used a subquery and selected the maximum payload mass

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

```
%sql select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_ = (select MAX(PAYLOAD_MASS__KG_) from SPACEXTBL) ;
```

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

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015. We used WHERE and LIKE to specify the conditions

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.

```
%sql select date, "Landing _Outcome", Booster_Version, Launch_Site from SPACEXTBL where date like '%2015%' and "Landing _Outcome" like "%failure%";
```

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

Date	Landing _Outcome	Booster_Version	Launch_Site
10-01-2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
14-04-2015	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

- Present your query result with a short explanation here

Section 3

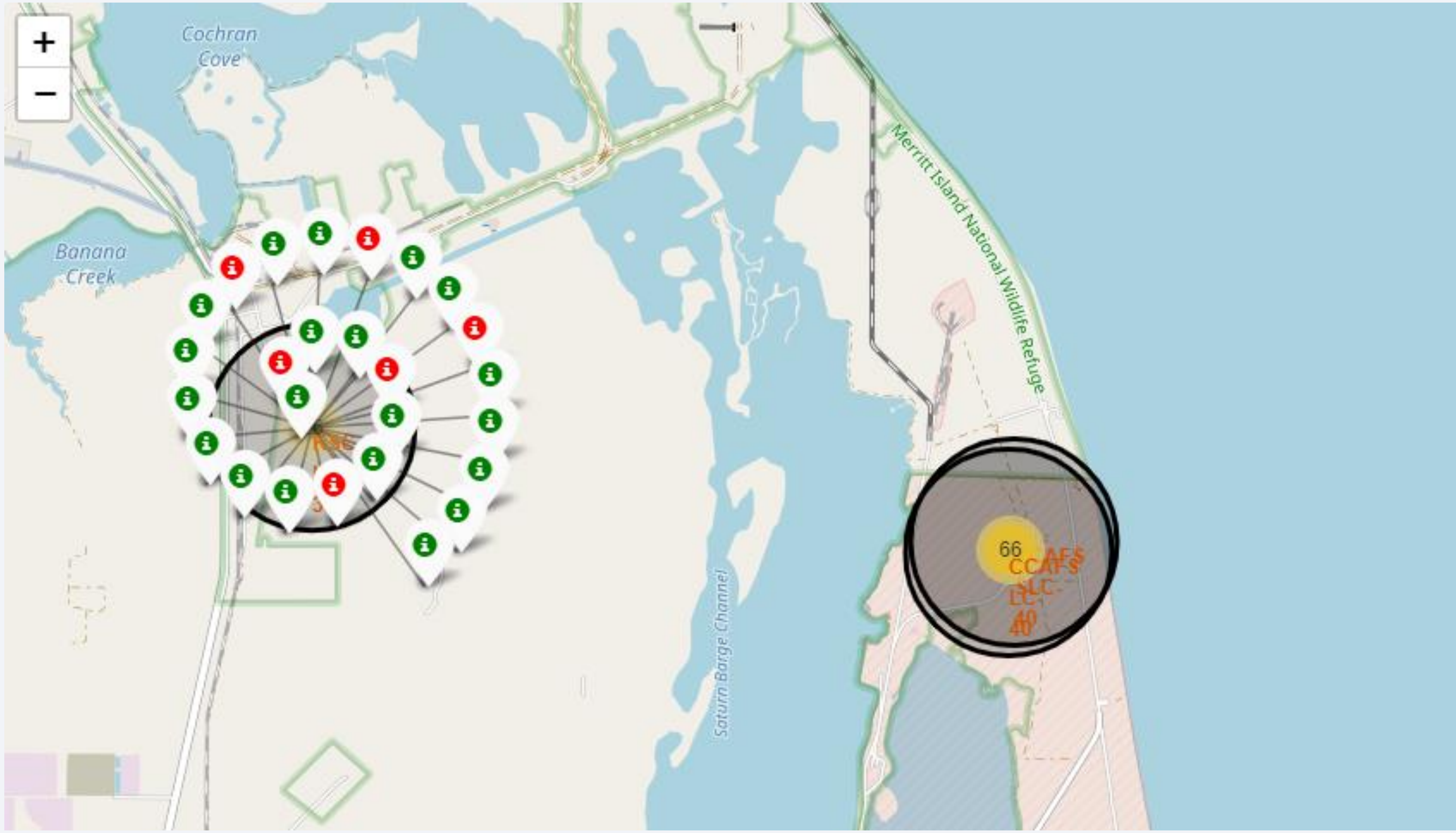
Launch Sites Proximities Analysis

Launch sites locations

- We observe that the launch sites are in the coasts of USA

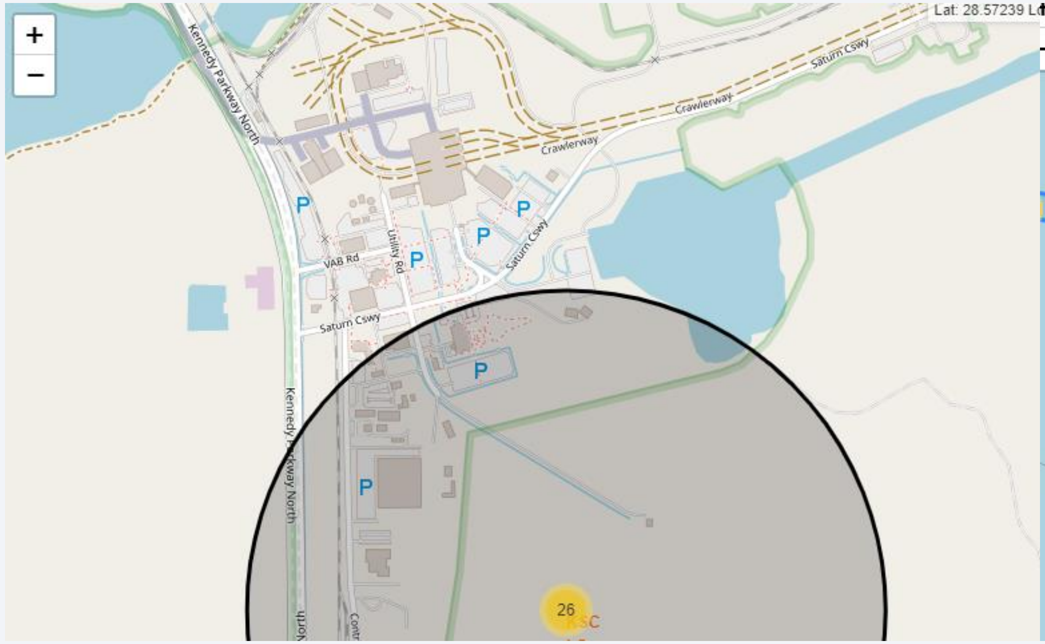
Color-labeled launch outcomes on the map

- Green marker shows successful launches and red ones are failures



Launch site to its proximities (railway, highway, coastline)

- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

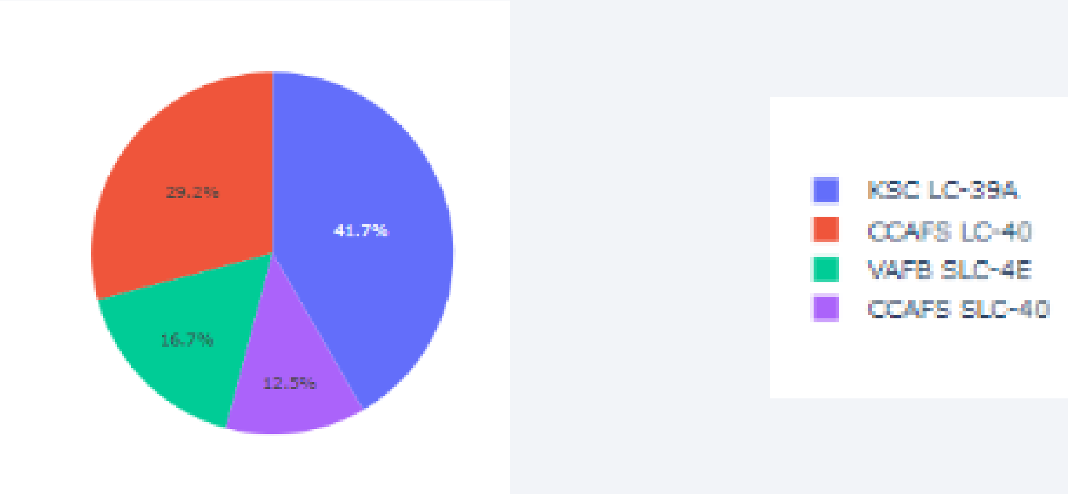


Section 4

Build a Dashboard with Plotly Dash

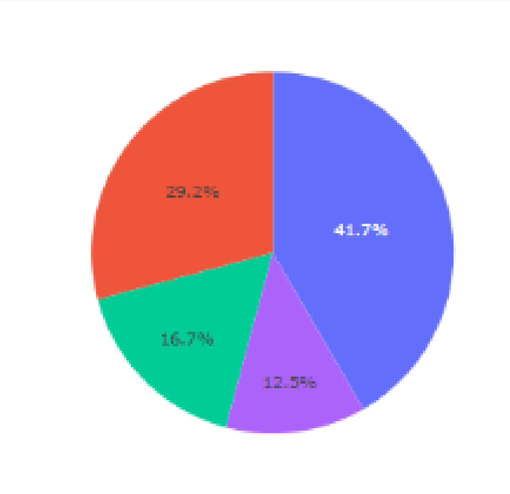
Pie Chart - Success achieved by each launch site

- In this pie chart we can observe the success rate of each launch site. The most successful site was KSC LC-39A

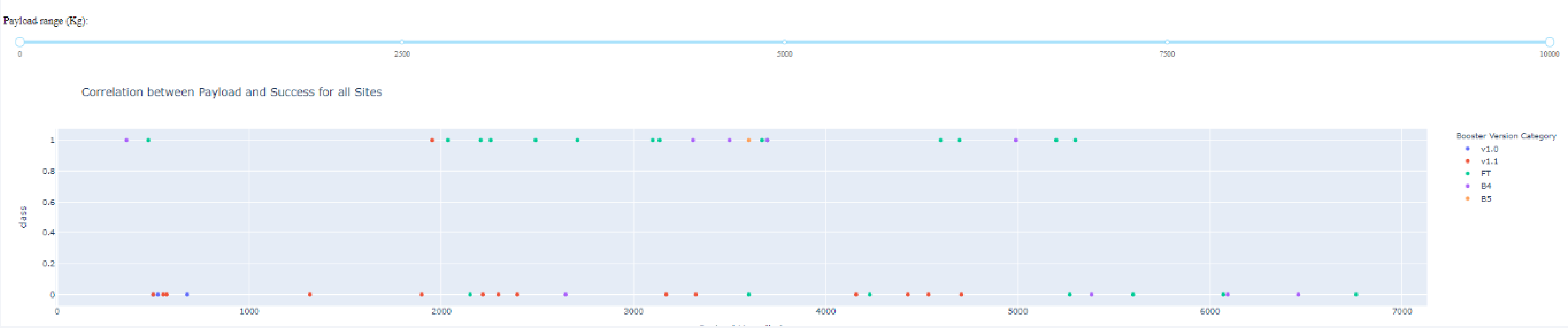


Pie Chart - Success achieved by each launch site

- In this pie chart we can observe the success rate of each launch site. The most successful site was KSC LC-39A



Payload mass vs Class



Section 5

Predictive Analysis (Classification)

Classification Accuracy

- We tried with different models and tested their performance. The model with the highest classification accuracy was the decision tree

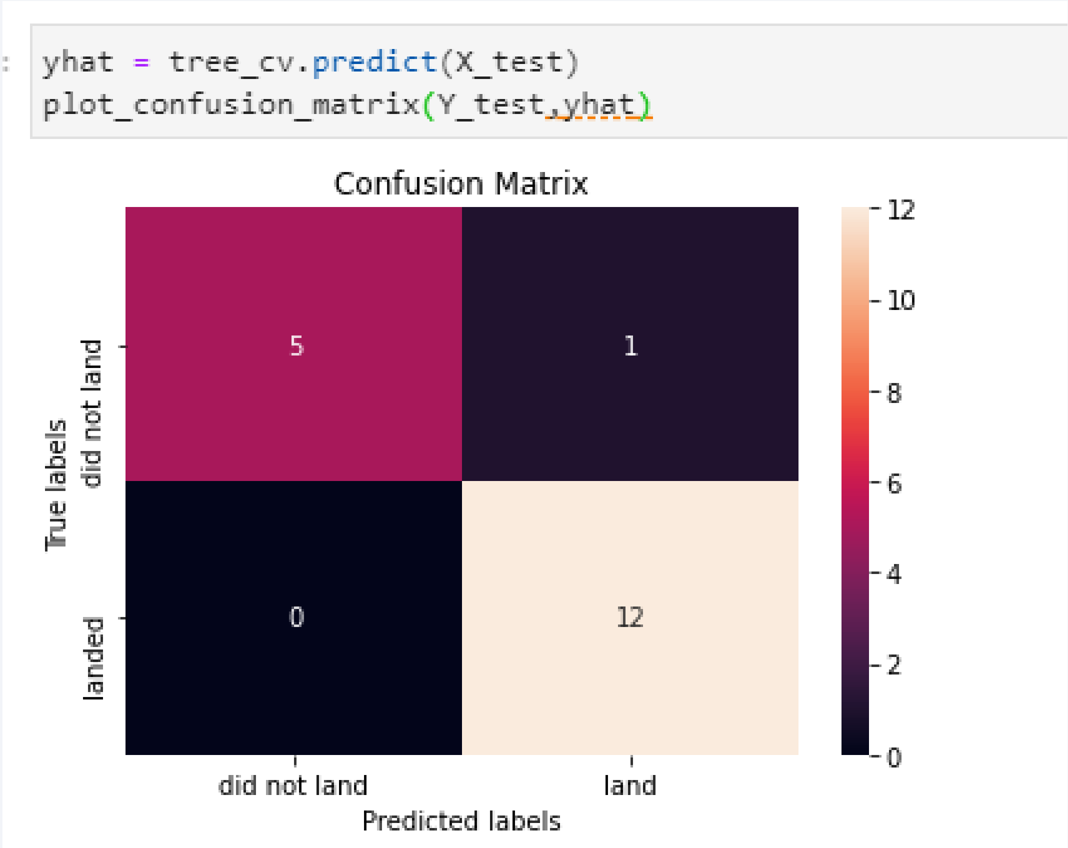
Find the method performs best:

```
print('Accuracy for Logistics Regression:', logreg_cv.score(X_test, Y_test))
print( 'Accuracy for Support Vector Machine:', svm_cv.score(X_test, Y_test))
print('Accuracy for Decision Tree:', tree_cv.score(X_test, Y_test))
print('Accuracy for K Nearest Neighbors:', knn_cv.score(X_test, Y_test))
```

```
Accuracy for Logistics Regression: 0.8333333333333334
Accuracy for Support Vector Machine: 0.8333333333333334
Accuracy for Decision Tree: 0.9444444444444444
Accuracy for K Nearest Neighbors: 0.8333333333333334
```

Confusion Matrix

- This is the confusion matrix for the decision tree. Here we see the true positives, true negatives, false positives and false negatives of the prediction on the test set



Conclusions

- The success rate got higher with the number of flights
- ES-L1, GEO, HEO, SSO, VLEO had the highest success rates
- The most successful launch site was KSC LC-39A
- The best algorithm for this task is the decision tree

Appendix

Thank you!