IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Derin Balogun
2/12/2024

Outline

Executive Summary

- Summary of methodologies

    -SpaceX Data Collection using SpaceX API
    -SpaceX Data Collection with Web Scraping
    -SpaceX Data Wrangling
    -SpaceX EDA using SQL
    -SpaceX EDA DataViz using Pandas and Matplotlib
    -SpaceX Launch Sites Analysis using Folium and Plotly Dash
    -SpaceX Machine Learning Landing Prediction

- Summary of all results
  - EDA results
  - Interactive Visual Analytics and Dashboards
  - Predictive Analysis (Classification)

Introduction

- Project background and context

  It costs SpaceX approximately 62 million dollars compared to other competitors which cost more than 165 million dollars. SpaceX reuses the first launch stage hence their reduced cost.

- Problems you want to find answers

  We will be predicting if Space X’s Falcon-9 will land successfully using the data from Falcon 9 rocket launches.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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.
    - The data was collected from the SpaceX API by making a get request. The relevant information is extracted from the API URL to get the rocket launch data.
    - The response received from the API was then decoded into a JSON file format. In order to analyse the data in a tabular format, a Panda dataframe object was created.
    - Using the BeautifulSoup and requests libraries for web scraping to collect historical Launch Data from a Wikipedia page titled - List of Falcon 9 and Falcon Heavy launches. This data was also stored into a pandas dataframe.

Data Collection – SpaceX API

A Pandas dataframe was created using the launch data requested from the SpaceX API using a GET request. The response is the converted to a JSON format to be made into the dataframe.

- URL : https://github.com/balogunderin/SpaceX-Falcon-9-1st-stage-Success-Landing-Prediction/blob/main/jupyter-labs-spacex-data-collection-api.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:

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

[10] Python

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

```python
response.status_code
```

[11] Python

```
200
```

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

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

[12] Python

Data Collection - Scraping

- Historical Launch Dates of SpaceX were collected using Beautiful Soup and the Requests libraries. Data was extracted from the HTML table on the Wikipedia page.
- https://github.com/balogunderin/SpaceX-Falcon-9-1st-stage-Success-Landing-Prediction/blob/main/jupyter-labs-webscraping.ipynb

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.

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

[10] Python

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

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

[11] Python

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

```python
# Use soup.title attribute
print(soup.title)
```

[12] Python

Data Wrangling

- Post Dataframe creation, a BoosterVERSION column was created to filter the data and keep Falcon9 Launches only. Missing data were then altered and replaced using the average value of the data in the specific column.
-
    - https://github.com/balogunderin/SpaceX-Falcon-9-1st-stage-Success-Landing-Prediction/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

TASK 4: Create a landing outcome label from Outcome column

Using the `Outcome`, create a list where the element is zero if the corresponding row in `Outcom` variable `landing_class`:

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
df['Class'] = df['Outcome'].apply(lambda x: 0 if x in bad_outcomes else 1)
df['Class'].value_counts()
```

```
1    60
0    30
Name: Class, dtype: int64
```

This variable will represent the classification variable that represents the outcome of each launch first stage landed Successfully

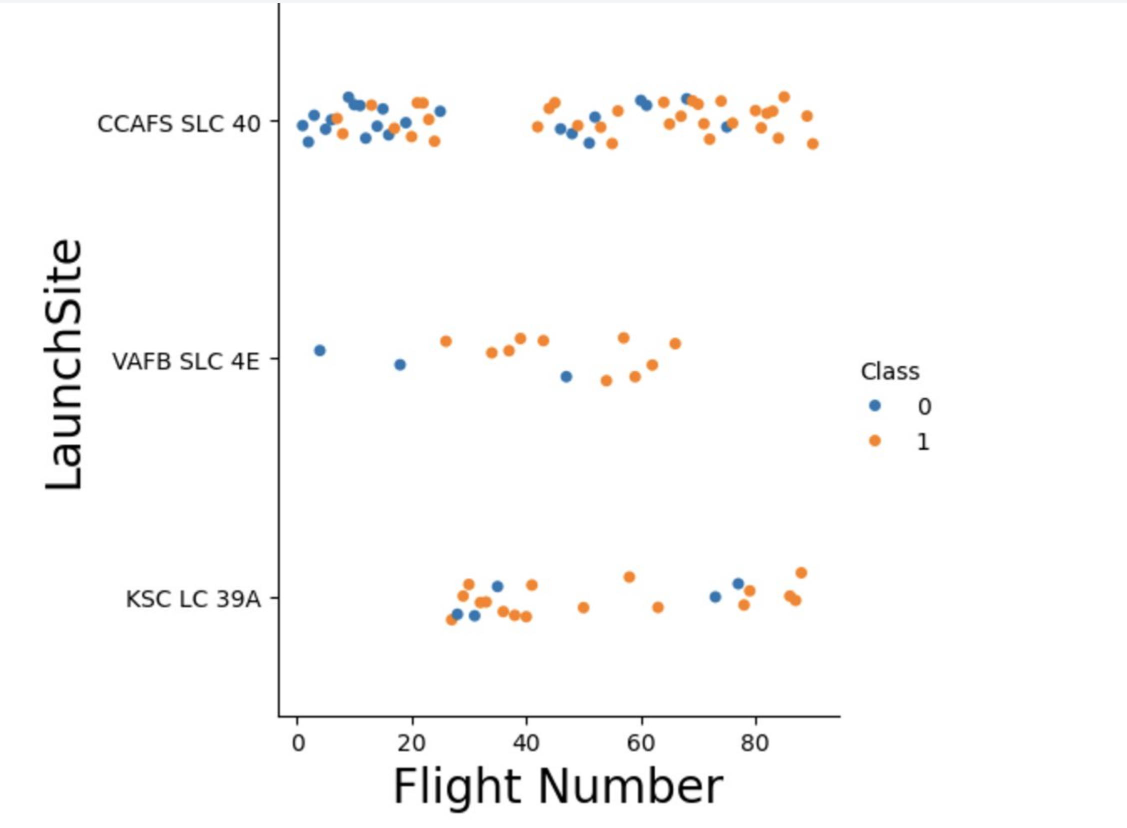
```
landing_class=df['Class']
df[['Class']].head(8)
```

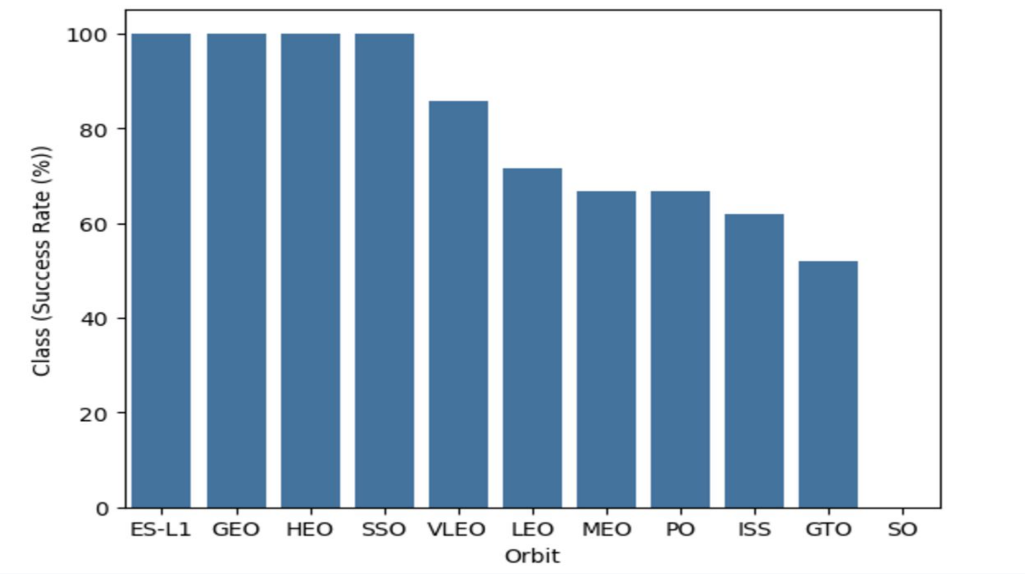
	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

EDA with Data Visualization

- Pandas and Matplotlib were used to create the graphs and charts.
- A scatter plot was created to view the relationship between **Flight Number** and **Launch Site**, **Payload** and **Launch Site**, **FlightNumber** and **OrbitType**, **Payload** and **Orbit type**.
- A line plot was to visualize the **yearly trend of launch success.**
- A bar chart was used to visualize the **success rate** vs **the orbit type.**
- **URL: https://github.com/balogunderin/SpaceX-Falcon-9-1st-stage-Success-Landing-Prediction/blob/main/5.%20Space-X%20EDA%20DataViz%20Using%20Pandas%20and%20Matplotlib%20-%20SpaceX.ipynb**

EDA with Data Visualization (graphs)

- 



EDA with SQL

- The following SQL queries were performed:
  - Display the names of the unique launch sites

```
%sql select DISTINCT "Launch_Site" from SPACEXTABLE
```

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

```
%sql select "Launch_Site" from SPACEXTABLE where "Launch_Site" like "CCA%" limit 5
```

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

```
%sql select SUM("PAYLOAD_MASS__KG_") from SPACEXTABLE
```

- Display average payload mass

```
%sql select AVG("PAYLOAD_MASS__KG_") from SPACEXTABLE
```

EDA with SQL (coun…)

- The following SQL queries were performed:
  - List the date the first successful landing occured

```
%sql select min(Date) from SPACEXTABLE
```

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

min(Date)
2010-06-04

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, Payload FROM SPACEXTBL WHERE "Landing _Outcome" = "Success (drone ship)" AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;)
```

List successful and failed mission outcomes

```
%sql select count(MISSION_OUTCOME) as missionoutcomes from SPACEXTBL GROUP BY MISSION_OUTCOME;
```

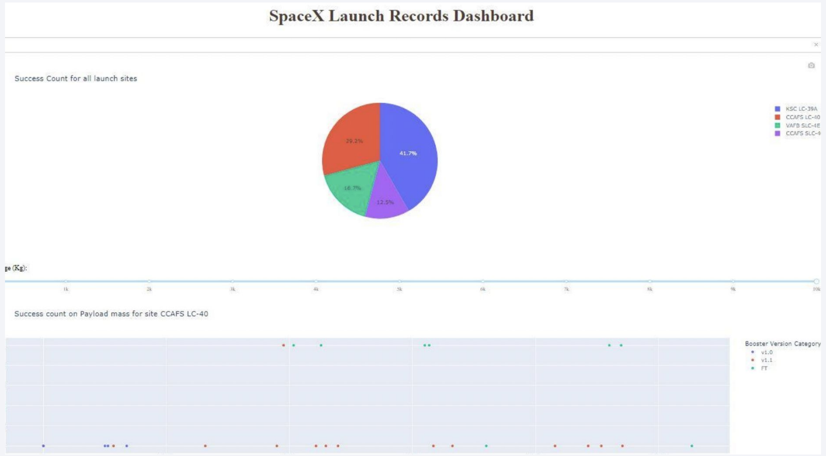
- URL:https://github.com/balogunderin/SpaceX-Falcon-9-1st-stage-Success-Landing-Prediction/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- A folium map was created to mark ther launch sites using markers and circles to mark the success or failure of individual launches at separate sites.
- The Launch outcomes then divided into failure=0 and success=1
- URL:https://github.com/balogunderin/SpaceX-Falcon-9-1st-stage-Success-Landing-Prediction/blob/main/6.Space-X%20Launch%20Sites%20Locations%20Analysis%20with%20Folium-Interactive%20Visual%20Analytics.ipynb

Build a Dashboard with Plotly Dash

- Created an interactive Plotly dashboard:
  - added a range slider to select payload
  - a callback function to render the scatter plot
  - adding a launch site drop-down input component
  - a callback to render selected site dropdown
- URL : https://github.com/balogunderin/SpaceX-Falcon-9-1st-stage-Success-Landing-Prediction/blob/main/Plotly%20Interactive%20Dashboard.py



Predictive Analysis (Classification)

- Post creating the pandas dataframe next was to perform EDA and classify the training sets and labels.
- I created a Numpy array from the data, using the to_numpy() method and assigning it to an a Y variable.
- The dataset was then standarized using the sklearn library. the dataset was the split into training and test data using the train_test_split from sklearn.model_selection.
- URL: https://github.com/balogunderin/SpaceX-Falcon-9-1st-stage-Success-Landing-Prediction/blob/main/8.%20SpaceX%20Machine%20Learning%20Prediction.ipynb

Predictive Analysis (Classification)

- To find the best Machine Learning model between k nearest neighbours, classification trees, logistic regression and svm;
- For each machine learning model, a GridSearchCV object then the training data is fit to find the best Hyperparameter
- Post fitting the training set I displayed the best attributes using the best_params attribute and to determine the accuracy of the trained model I used the best_score data attribute.
- The score method to calculate the accuracy on the test data for each respective model and plot a confusion matrix using the test/predicted outcomes.
- URL , test data score

Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

Results

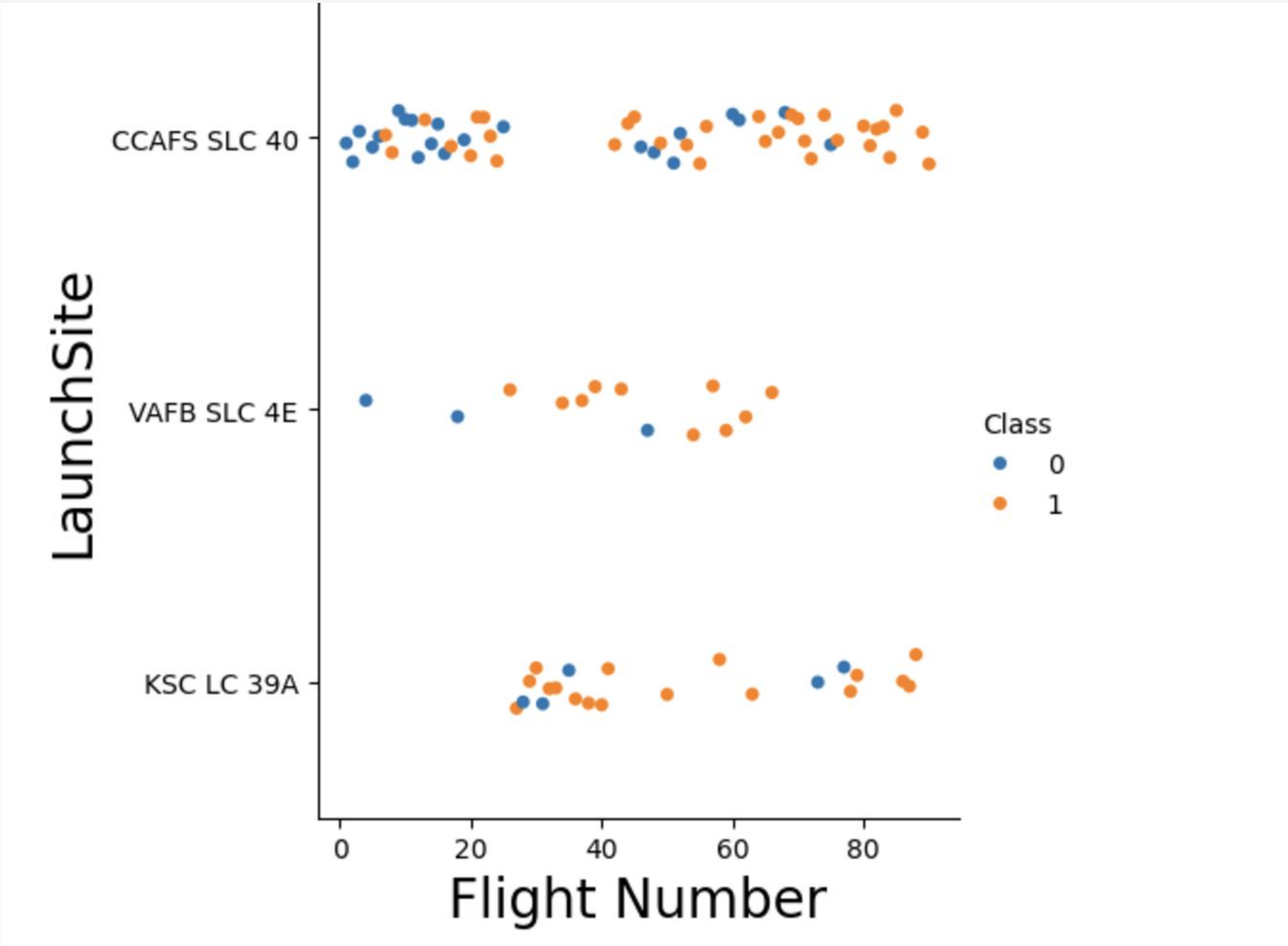
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

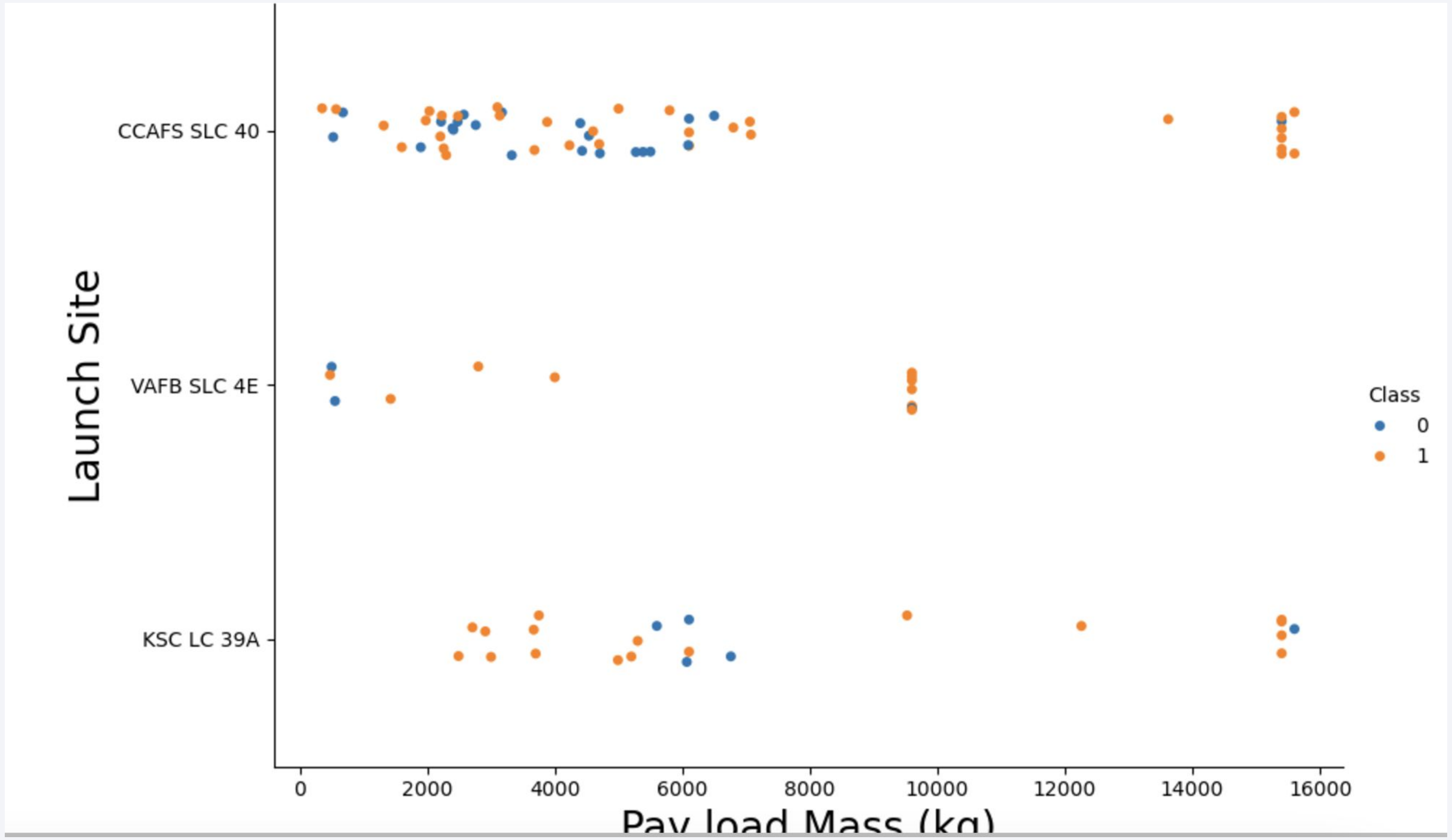
Flight Number vs. Launch Site

- Flight number and Launch site have a linear relationship. As success rate increases with flight number on each launch site.
-



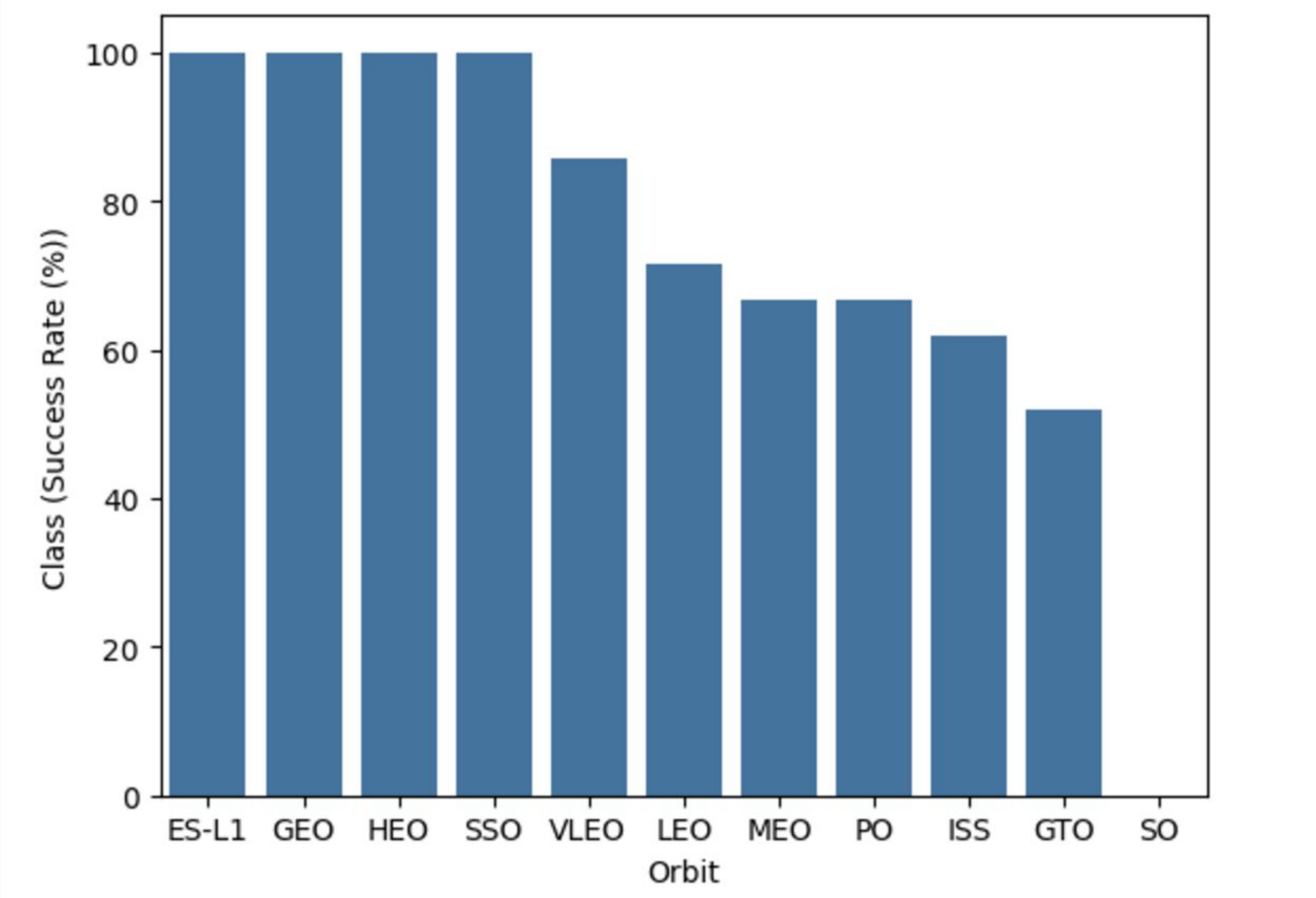
Payload vs. Launch Site

- Now if you observe Payload Mass Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).



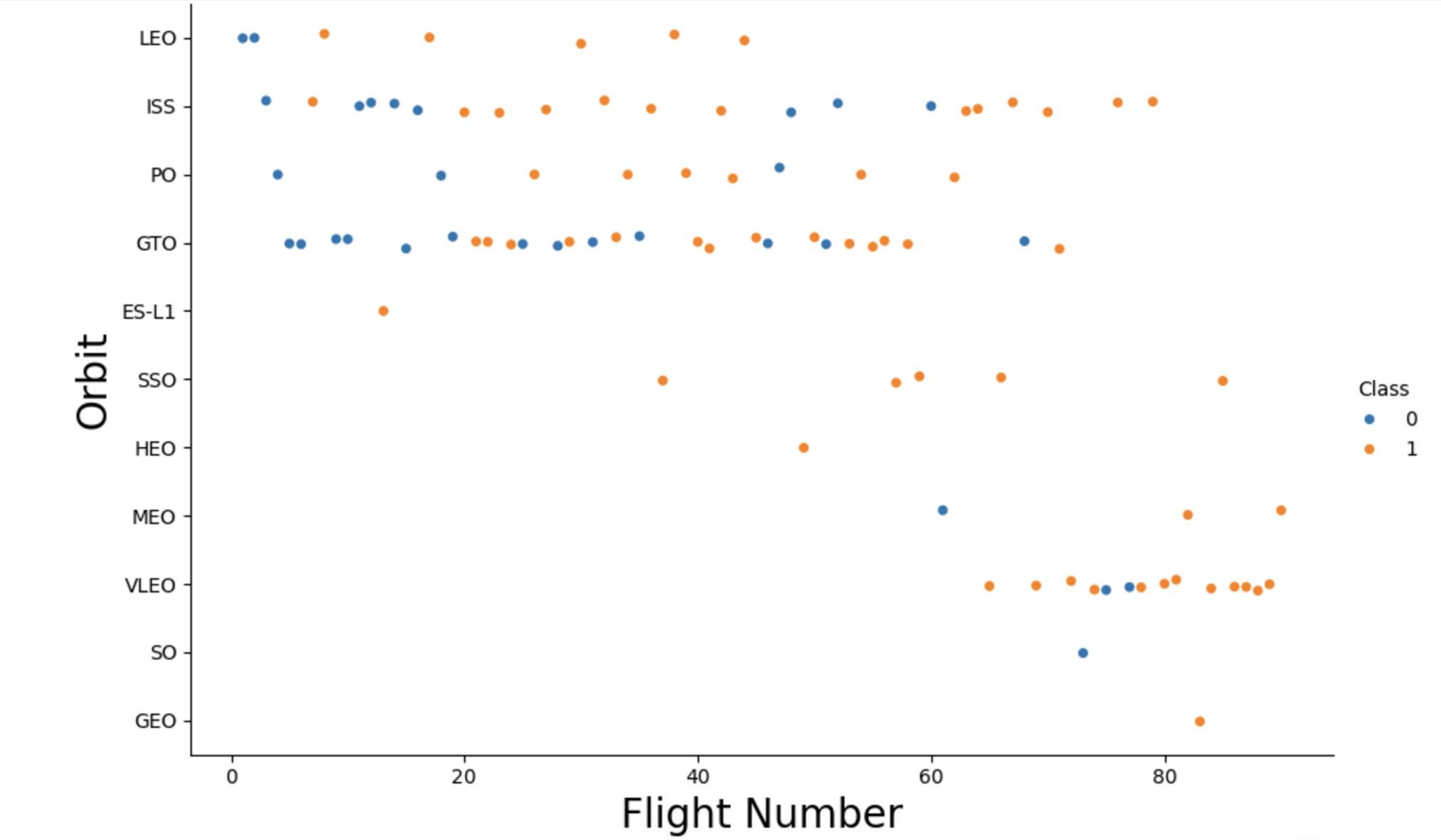
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO and SSO all have a 100% success rate. SO orbit has a 0% success rate. All other orbits are between 50% to about 85%



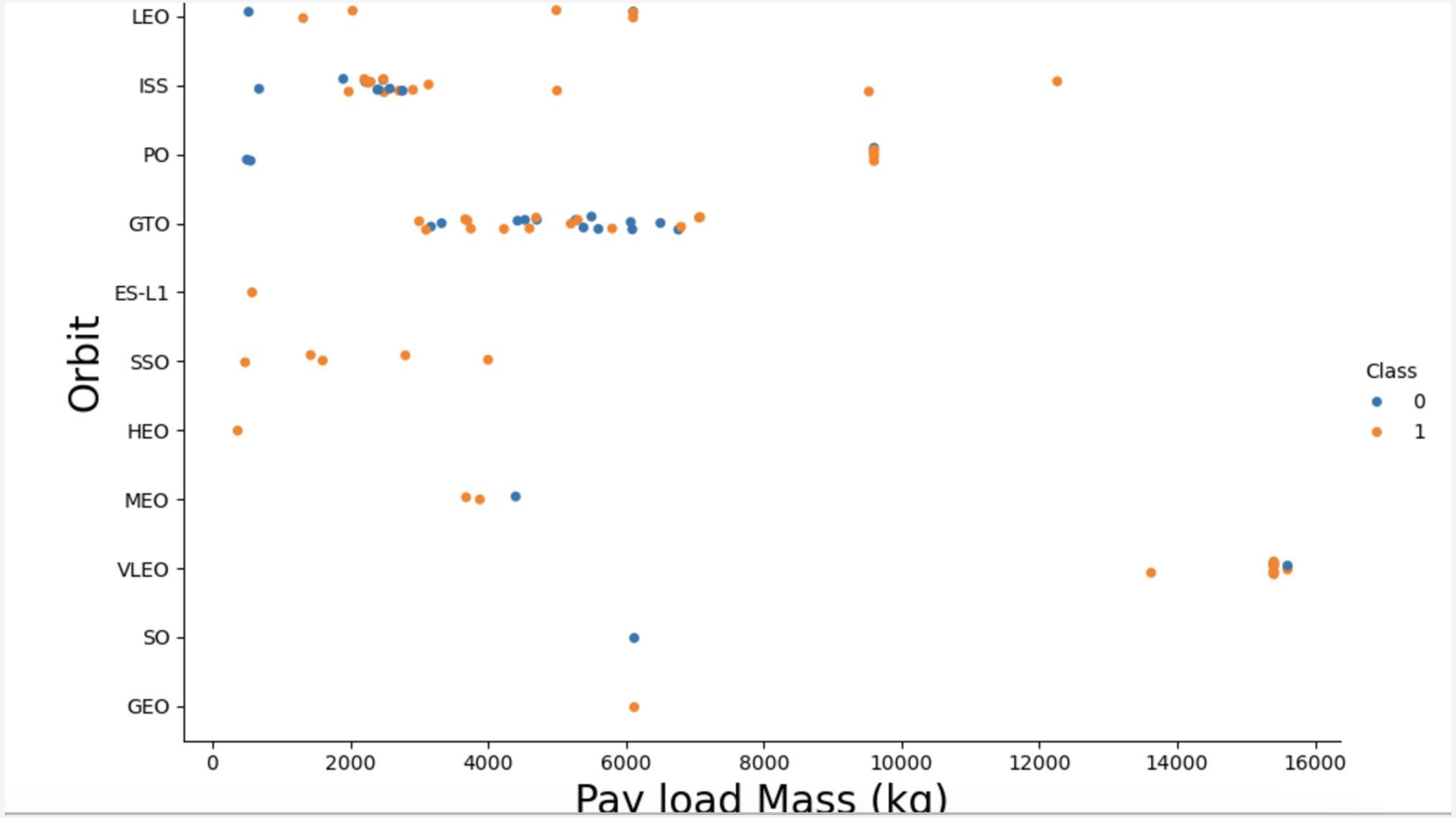
Flight Number vs. Orbit Type

- You can observe that in the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.



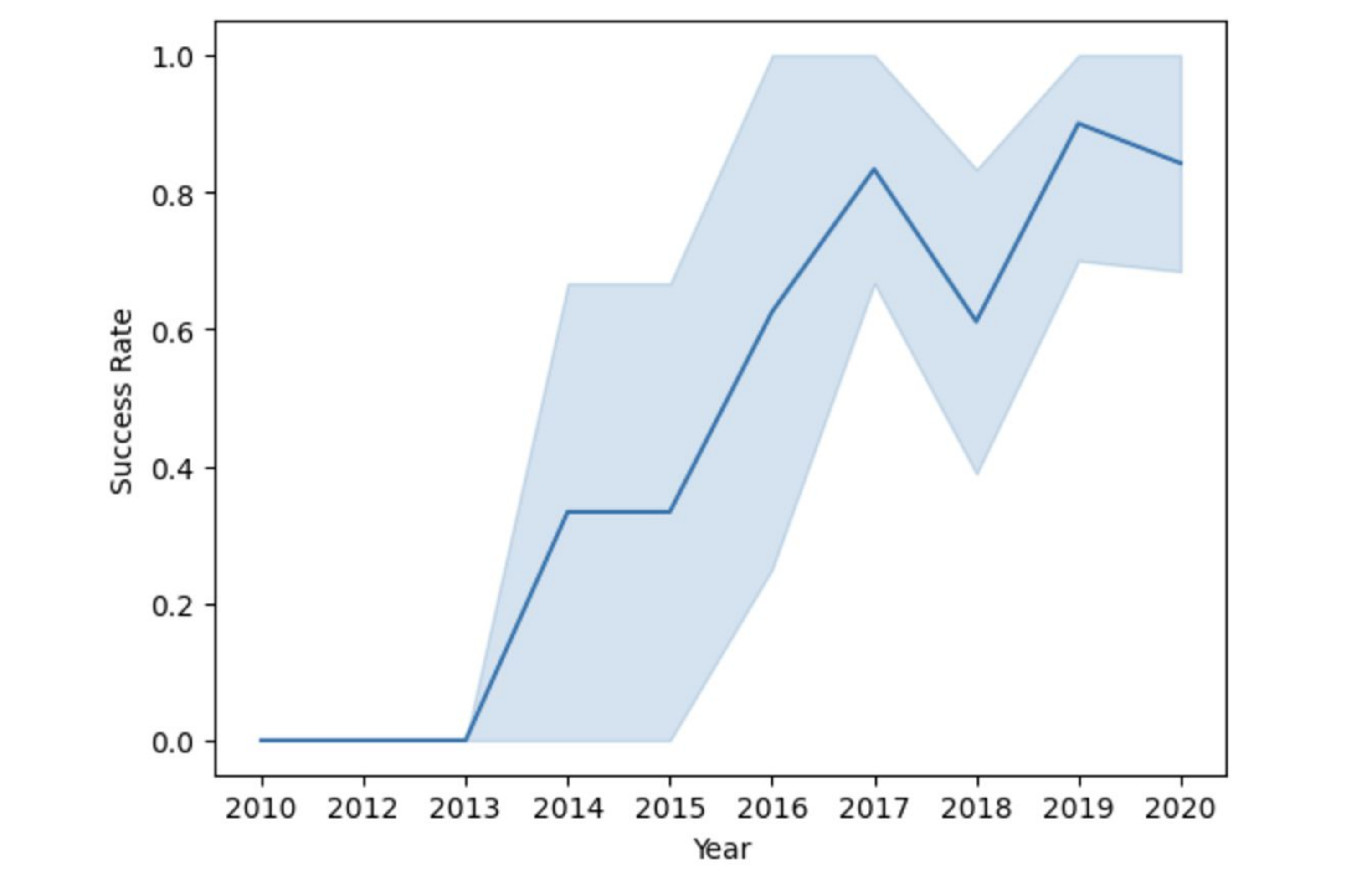
Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.



Launch Success Yearly Trend

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



All Launch Site Names

- Find the names of the unique launch sites
- Used 'SELECT DISTINCT' in order to attain unique launch sites from the 'LAUNCH_SITE' column of the SPACEXTBL table

Task 1

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

```
%sql select DISTINCT "Launch_Site" from SPACEXTABLE
```

```
 * 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'

- Find 5 records where launch sites begin with `CCA`

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

```
%sql select "Launch_Site" from SPACEXTABLE where "Launch_Site" like "CCA%" limit 5
```

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

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

Used ‘LIKE’ command with ‘%’ wildcard in the ‘WHERE’ clause to select records where launch sites begin with the string 'CCA' making the limit 5

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

Task 3

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

```
%sql select SUM("PAYLOAD_MASS__KG_") from SPACEXTABLE
```

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

SUM("PAYLOAD_MASS__KG_")
619967

Used the ‘SUM()’ function to show the total sum of ‘PAYLOAD_MASS_KG’ column for Customer ‘NASA(CRS’

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

Task 4

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

```
%sql select AVG("PAYLOAD_MASS__KG_") from SPACEXTABLE
```

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

AVG("PAYLOAD_MASS__KG_")
6138.287128712871

Used the 'AVG()' function to calculate and display the average payload mass carried by booster version F9 v1.1

First Successful Ground Landing Date

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

Task 5

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

Hint:Use min function

```
%sql select min(Date) from SPACEXTABLE
```

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

min(Date)
2010-06-04

Used the 'MIN()' function to return and dispaly the first (oldest) date when first successful landing outcome on ground pad

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

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 * FROM 'SPACEXTBL'
```

```
%sql SELECT DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE "Landing _Outcome" = "Success (drone ship)" AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOA
```

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

Booster_Version	Payload
F9 FT B1022	JCSAT-14
F9 FT B1026	JCSAT-16
F9 FT B1021.2	SES-10
F9 FT B1031.2	SES-11 / EchoStar 105

Used ‘Select Distinct’ statement to list the names of boosters with operators >4000 and <6000 in payload mass with landing outcome of ‘Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

Task 7

List the total number of successful and failure mission outcomes

```
%sql select count(MISSION_OUTCOME) as missionoutcomes from SPACEXTBL GROUP BY MISSION_OUTCOME;
```

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

missionoutcomes
1
98
1
1

Used the ‘COUNT()’ with the ‘GROUP BY’ statement to return total number of missions outcomes

Boosters Carried Maximum Payload

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

Task 8

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

```
%sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACE
```

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

boosterversion
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

Using a Subquerry to give the Max payload and used it to list all the boosters that have carried the Max payload of 15600kg

2015 Launch Records

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

Used the ‘subsrt()’ in the select statement to get the month and year from the date column and return the records matching the filter.

Task 9

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, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%sql SELECT substr(Date, 6,2),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL where substr(Date,0,5)='2015';
```

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

substr(Date, 6,2)	Mission_Outcome	Booster_Version	Launch_Site
01	Success	F9 v1.1 B1012	CCAFS LC-40
02	Success	F9 v1.1 B1013	CCAFS LC-40
03	Success	F9 v1.1 B1014	CCAFS LC-40
04	Success	F9 v1.1 B1015	CCAFS LC-40
04	Success	F9 v1.1 B1016	CCAFS LC-40
06	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40
12	Success	F9 FT B1019	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 successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

```
%sql SELECT * FROM SPACEXTBL WHERE "Landing _Outcome" LIKE 'Success%' AND (Date BETWEEN '04-06-2010' AND '20-03-2017') ORDER BY Date DESC;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
19-02-2017	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
18-10-2020	12:25:57	F9 B5 B1051.6	KSC LC-39A	Starlink 13 v1.0, Starlink 14 v1.0	15600	LEO	SpaceX	Success	Success
18-08-2020	14:31:00	F9 B5 B1049.6	CCAFS SLC-40	Starlink 10 v1.0, SkySat-19, -20, -21, SAOCOM 1B	15440	LEO	SpaceX, Planet Labs, PlanetIQ	Success	Success
18-07-2016	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
18-04-2018	22:51:00	F9 B4 B1045.1	CCAFS SLC-40	Transiting Exoplanet Survey Satellite (TESS)	362	HEO	NASA (LSP)	Success	Success (drone ship)

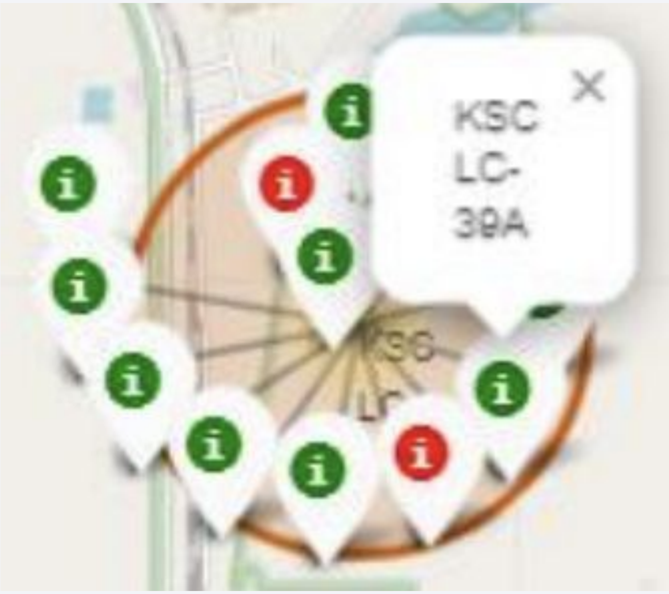
Section 3

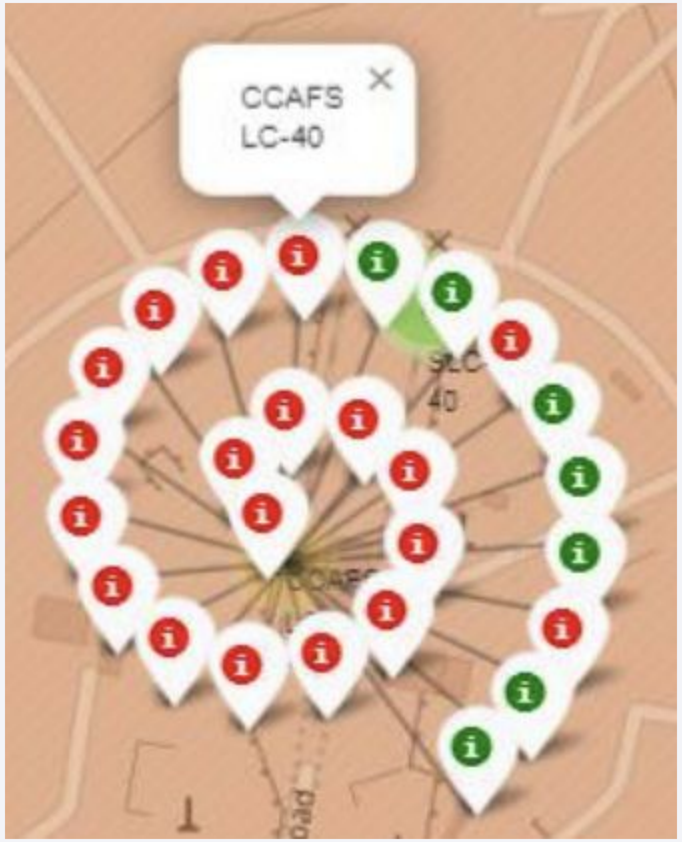
Launch Sites Proximities Analysis

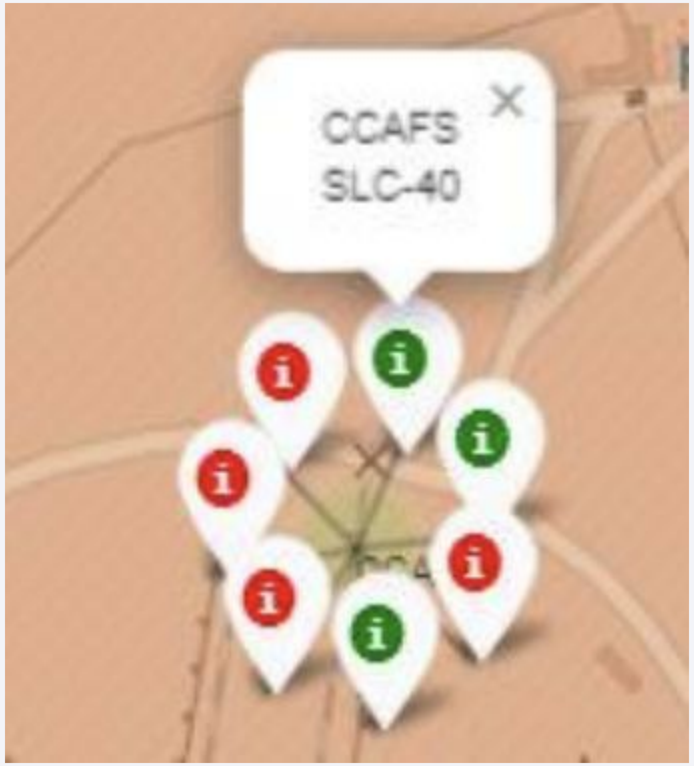
Makers on global map

- All the launch sites are on the western hemisphere. They are also all on the coastline.

Launch outcomes for each site on the map







Launch site KSC LC-39A has a much higher success rate compared to CCAFS SLC-40 & CCAFS LC-40. CCAFS LC-40 also has the most launches in relation to the other sites in florida.

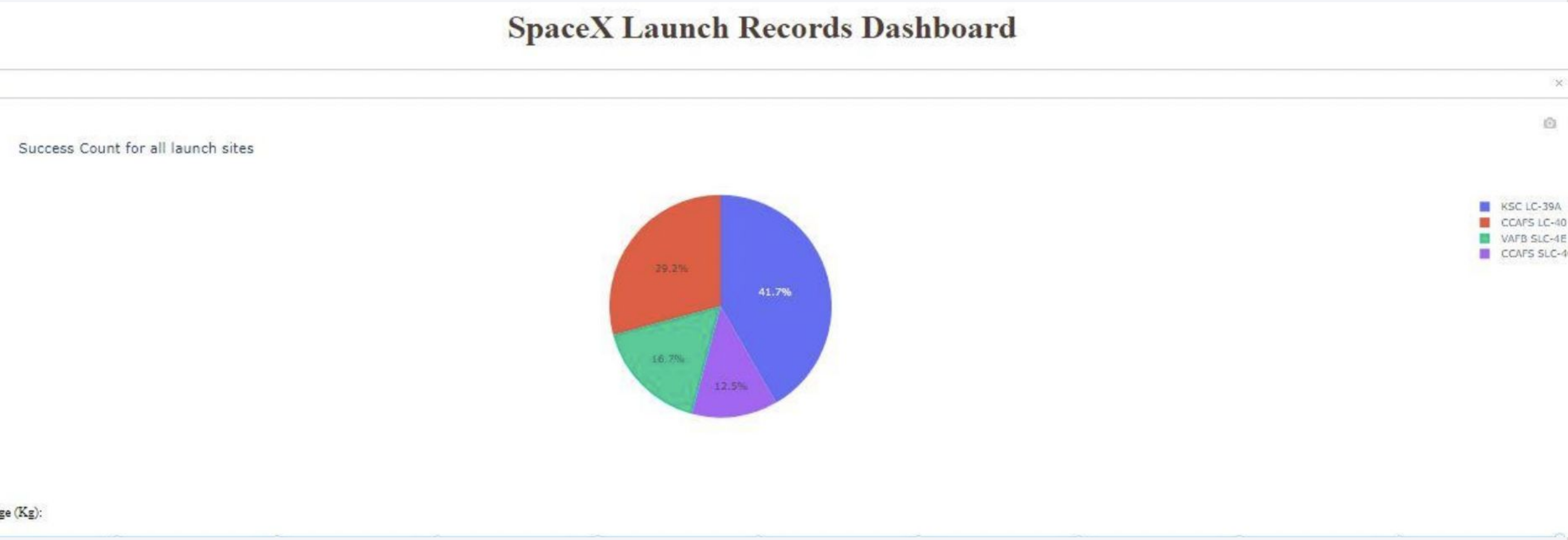
Launch outcomes for each site on the map

In California Launch site VAFB SLC-4E has 40% success rate compared to florida launch sites, this is the 2nd most successful site in terms of launches

Section 4

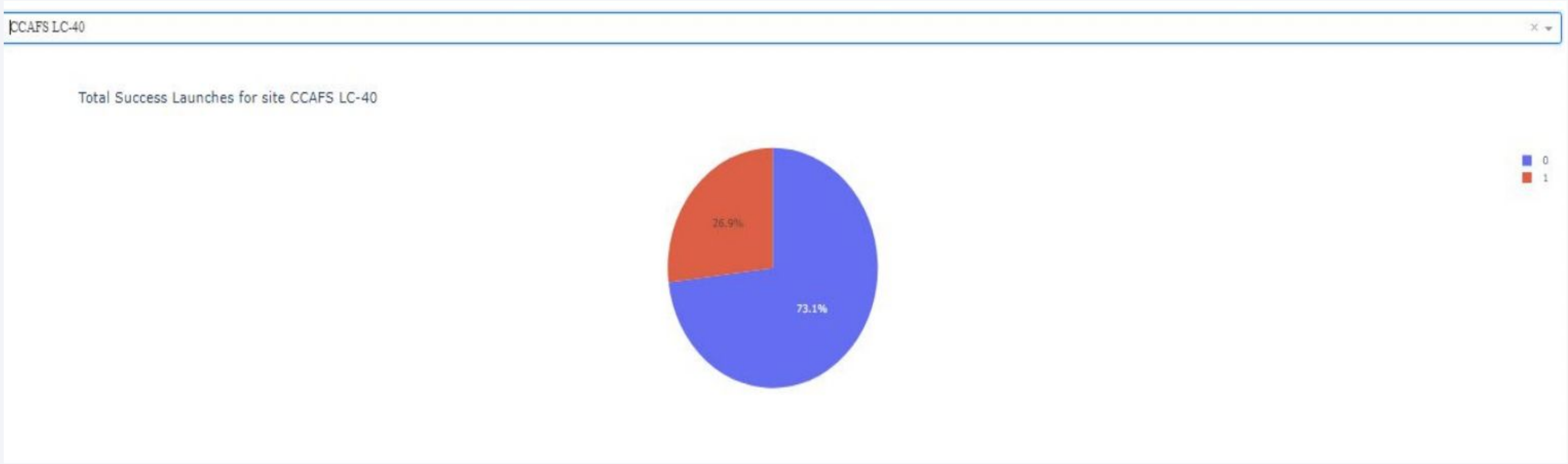
Build a Dashboard with Plotly Dash

Pie chart showing success% of all sites



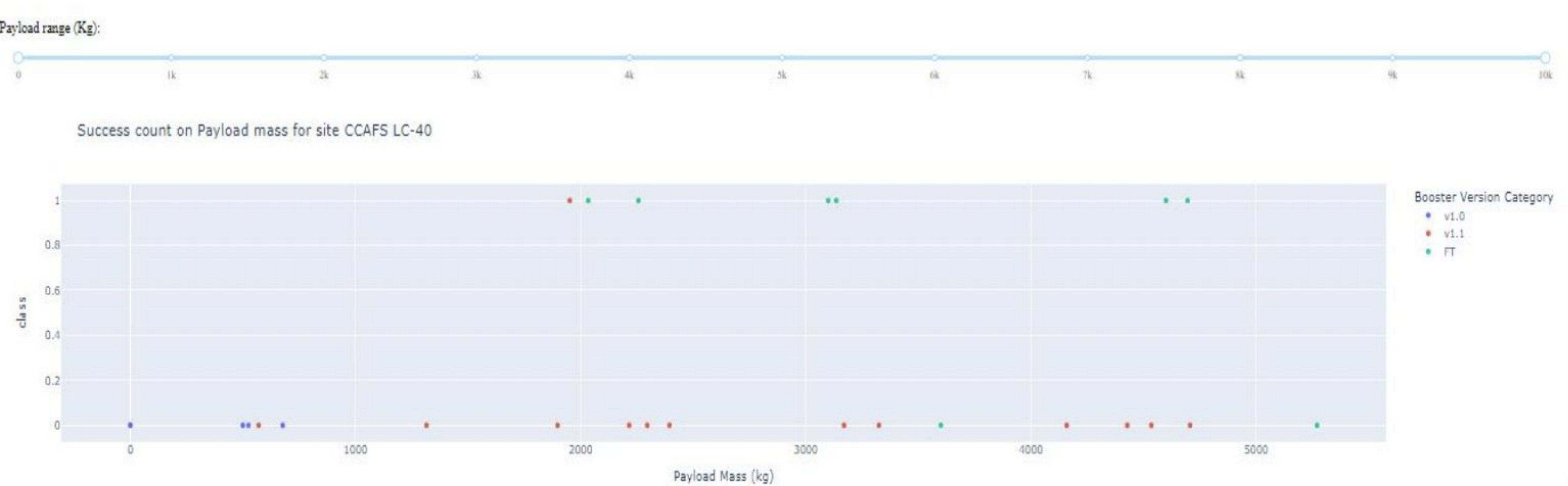
Launch site KSC LC-39A has the highest launch success rate at 42% followed by CCAFS LC-40 at 29%, VAFB SLC-4E at 17% and lastly launch site CCAFS SLC-40 with a success rate of 13%

Pie chart showing the launch site with 2nd highest launch success ratio



Launch site CCAFS LC-40 had the 2nd highest success ratio of 73% success

Payload vs. Launch Outcome scatter plot for all sites



For this Launch site (CCAFS LC-40) the largest success rate tends to show when the payload mass is >2000kg

Section 5

Predictive Analysis (Classification)

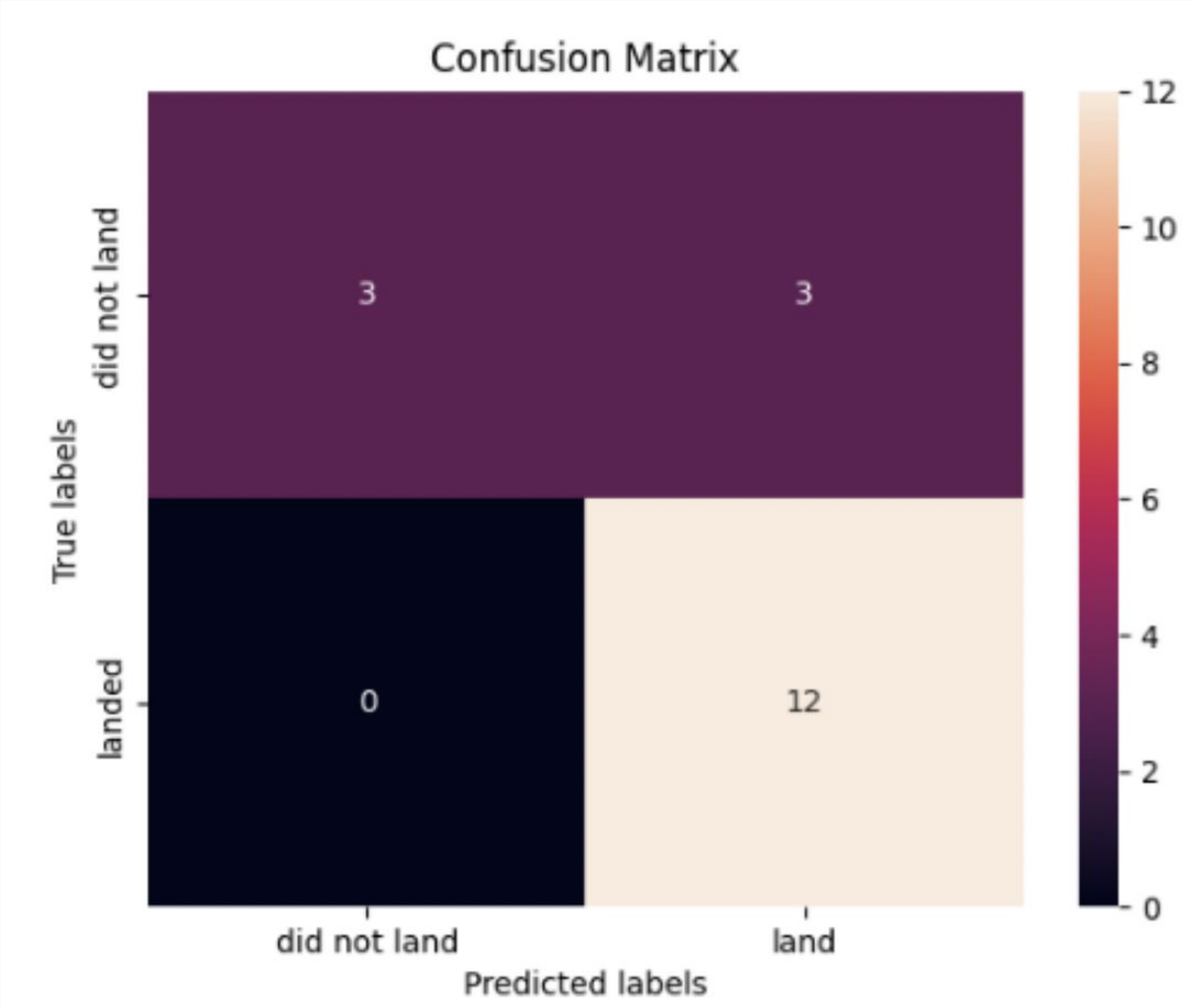
Classification Accuracy

- According to the results all the classifications have the same accuracy for the test data.

Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

Confusion Matrix

All the 4 classification models had the same confusion matrixes and were able equally distinguish between the different classes.



Conclusions

- The success rates of all Florida launch sites except for CCAFS LC-40 are more than 50% probability of landing successfully.
- We can also conclude from the Flight Number vs Launch site plot that as the number of flights increase also does the probability of a successful launch. Launchs are also a lot more successful when the payload mass is >2000kg. As the success rate for the VAFB SLC 4E launch site is 100% after the Flight number 50. Both KSC LC 39A and CCAFS SLC 40 have a 100% success rates after 80th flight
- ES-L1, GEO, HEO and SSO all have a 100% success rate. SO orbit has a 0% success rate. All other orbits are between 50% to about 85%
- According to the yearly trend line plot, the success rate of landings was consistent for about three years then continued to increase from 2013 until 2020.

Thank you!