IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

MARK MOGERE MOSOTI
23RD JUNE 2023

Outline

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

Executive Summary

Summary of methodologies

- SpaceX Data Collection using SpaceX API
- SpaceX Data Collection with Web Scraping
- SpaceX Data Wrangling
- SpaceX Exploratory Data Analysis using SQL
- Space-X EDA DataViz Using Python Pandas and Matplotlib
- Space-X Launch Sites Analysis with Folium-Interactive Visual Analytics and Ploty 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**

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.

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.

- **Problems you want to find answers**

In this capstone, we will predict if the Falcon 9 first stage will land successfully using data from Falcon 9 rocket launches advertised on its website.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Web scrapping from Wikipedia
- Perform data wrangling
  - One hot encoding data fields for machine learning and data cleaning of null values and columns.
- 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

- Data was first collected using SpaceX API (a RESTful API) by making a get request to the SpaceX API. This was done by first defining a series helper functions that would help in the use of the API to extract information using identification numbers in the launch data and then requesting rocket launch data from the SpaceX API url.

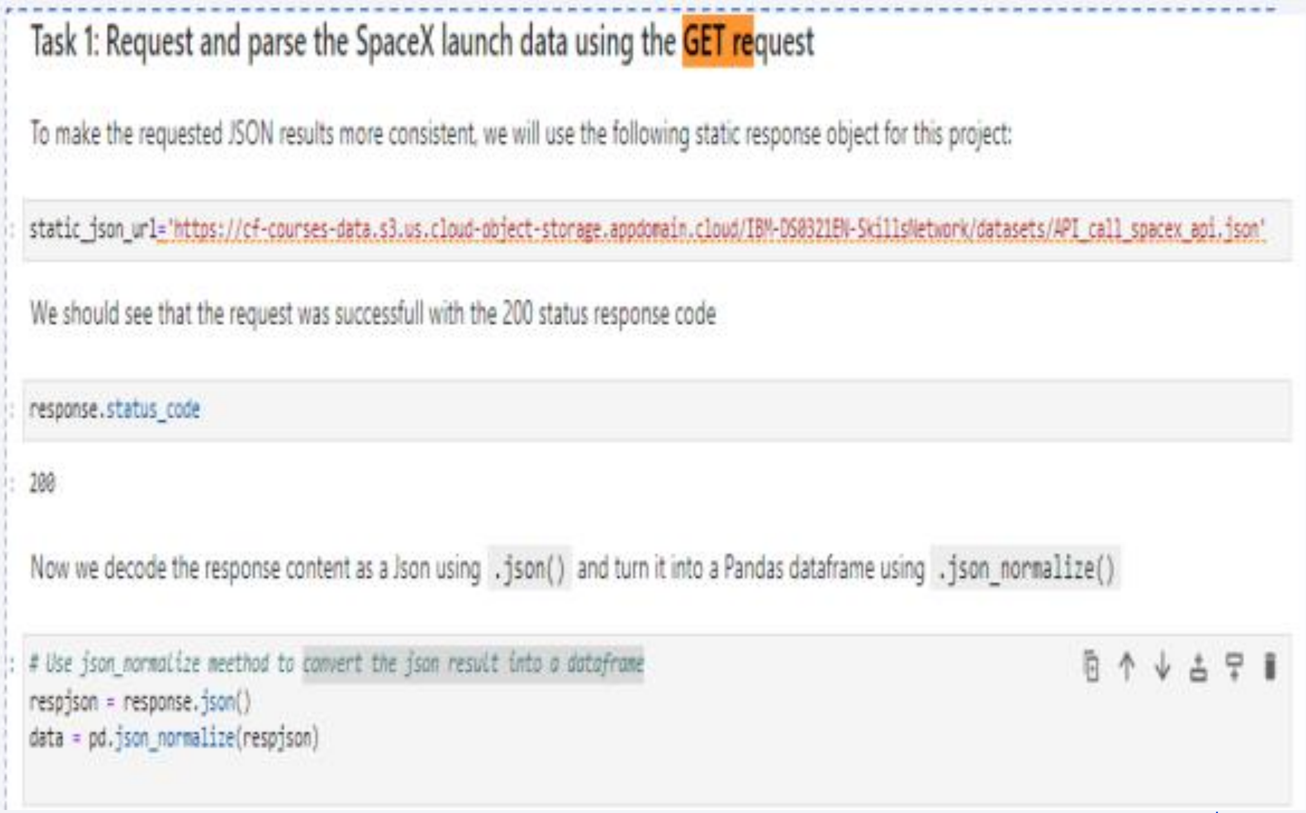
• To make the requested JSON results more consistent, the SpaceX launch data was requested and parsed using the GET request and then decoded the response content as a Json result which was then converted into a Pandas data frame.

• Also performed web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled List of Falcon 9 and Falcon Heavy launches of the launch records are stored in a HTML. Using Beautiful Soup and request Libraries, I extract the Falcon 9 launch HTML table records from the Wikipedia page, Parsed the table and converted it into a Pandas data frame.

Data Collection – SpaceX API

- Data collected using SpaceX API (a RESTful API) by making a get request to the SpaceX API then requested and parsed the SpaceX launch data using the GET request and decoded the response content as a Json result which was then converted into a Pandas data frame.
- Here is the Github Link:

https://github.com/MMOSOTI/Data-science-Capstone/blob/main/Space%20X%20API%20Data%20Collection%20Files.pdf

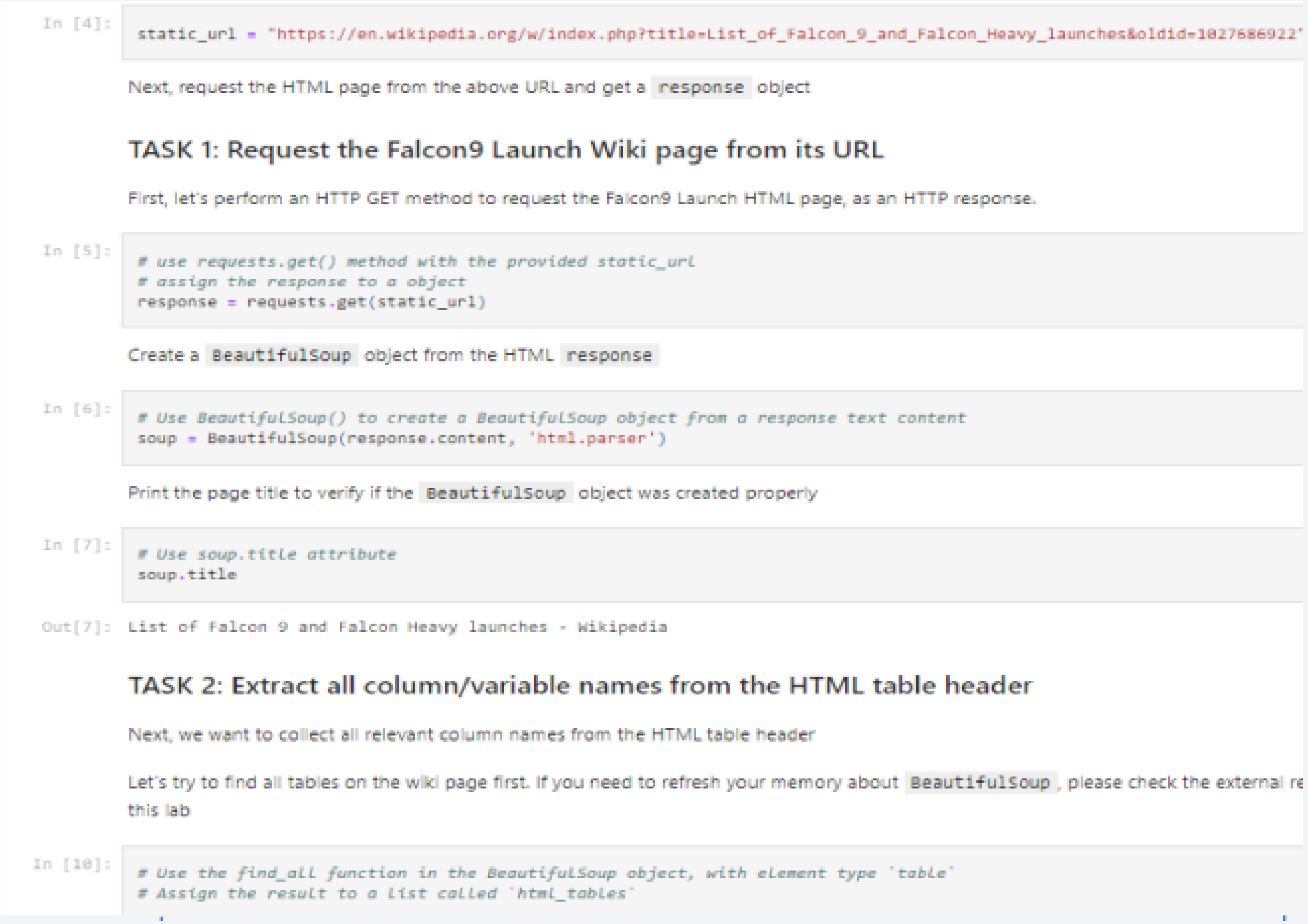


Data Collection - Scraping

- Performed web scraping to collect Falcon 9 historical launch records from a Wikipedia using Beautiful Soup and request, to extract the Falcon 9 launch records from HTML table of the Wikipedia page, then created a data frame by parsing the launch HTML.

- Here is the Github Link:

https://github.com/MMOSOTI/Data-science-Capstone/blob/main/Complete%20the%20Data%20Collection%20with%20Web%20Scraping%20(1).pdf



In [4]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

Next, request the HTML page from the above URL and get a `response` object

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.

In [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

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

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.content, 'html.parser')
```

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

In [7]:
```
# Use soup.title attribute
soup.title
```

Out[7]: List of Falcon 9 and Falcon Heavy launches - Wikipedia

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 re

this lab

In [10]:
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
```

Data Wrangling

- After obtaining and creating a Pandas DF from the collected data, data was filtered using the BoosterVersion column to only keep the Falcon 9 launches, then dealt with the missing data values in the LandingPad and PayloadMass columns. For the PayloadMass , missing data values were replaced using mean value of column.

• Also conducted Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.

Here is the Github Link :

https://github.com/MMOSOTI/Data-science-Capstone/blob/main/SpaceX%20Data%20Wrangling.pdf

EDA with Data Visualization

- Performed data Analysis and Feature Engineering using Pandas and Matplotlib.i.e.Exploratory Data Analysis ,Preparing Data Feature Engineering .
- I utilized scatter plots to Visualize the relationship between Flight Number and Launch Site, Payload and Launch Site, Flight Number and Orbit type, Payload and Orbit type.
- Bar chart was used to Visualize the relationship between success rate of each orbit type, Line plot to Visualize the launch success yearly trend.

Here is the Github Link:

https://github.com/MMOSOTI/Data-science-Capstone/commit/2d2bab95ad62c772f3e49503d9047bf1291a6108

%sql SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;

EDA with SQL

These are the SQL queries I performed :

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

%sql select Unique(LAUNCH_SITE) from SPACEXTBL;

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

%sql SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;

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

%sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;

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

%sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;

%sql SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;

EDA with SQL(Continued…..)

- **List the date when the first successful landing outcome in ground pad was achieved.**

%sql select min(DATE) from SPACEXTBL;

- **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 LANDING_OUTCOME='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000;

- **List the total number of successful and failure mission outcomes**

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

Here is the Github Link: https://github.com/MMOSOTI/Data-science-Capstone/commit/2d2bab95ad62c772f3e49503d9047bf1291a6108

%sql SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;

EDA with SQL(Continued…..)

- **List the names of the booster_versions which have carried the maximum payload mass.**

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

- **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**

%sql select MONTH(DATE),Mission_Outcome,Booster_Version,Launch_Site from SPACEXTBL where EXTRACT(YEAR FROM DATE)='2';

- **Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order**

%sql SELECT LANDING_OUTCOME FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC;

Here is the Github Link: https://github.com/MMOSOTI/Data-science-Capstone/commit/2d2bab95ad62c772f3e49503d9047bf1291a6108

Build an Interactive Map with Folium

- Created folium map to mark all the launch sites, and created map objects such as markers, circles, lines to mark the success or failure of launches for each launch site.
- Created a launch set outcomes (failure=0 or success=1).

Build a Dashboard with Plotly Dash

Built an interactive dashboard application with Plotly dash by:

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

Predictive Analysis (Classification)

After loading the data as a Pandas Data frame, I set out to perform exploratory Data Analysis and determine Training Labels by;

• creating a NumPy array from the column Class in data, by applying the method to numpy() then assigned it to the variable Y as the outcome variable.

• Then standardized the feature dataset (x) by transforming it using preprocessing. StandardScaler() function from Sklearn.

- After which the data was split into training and testing sets using the function train_test_split from sklearn.model_selection with the test_size parameter set to 0.2 and random_state to 2.

Predictive Analysis (Classification)

In order to find the best ML model/ method that would performs best using the test data between SVM, Classification Trees, k nearest neighbors and Logistic Regression;

• First created an object for each of the algorithms then created a GridSearchCV object and assigned them a set of parameters for each model.

• For each of the models under evaluation, the GridsearchCV object was created with cv=10, then fit the training data into the GridSearch object for each to Find best Hyperparameter.

• After fitting the training set, we output GridSearchCV object for each of the models, then displayed the best parameters using the data attribute best_params_ and the accuracy on the validation data using the data attribute best score_.

- Finally using the method score to calculate the accuracy on the test data for each model and plotted a confussion matrix for each using the test and predicted outcomes.
- Here is the Github Link:

https://github.com/MMOSOTI/Data-science-Capstone/blob/main/Machine%20learning%20prediction.pdf

Predictive Analysis (Classification)

The table below shows the test data accuracy score for each of the methods comparing them to show which performed best using the test data between SVM, Classification Trees, k nearest neighbors and Logistic Regression;

Out[68]:

	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

https://github.com/MMOSOTI/Data-science-Capstone/blob/main/Machine%20learning%20prediction.pdf

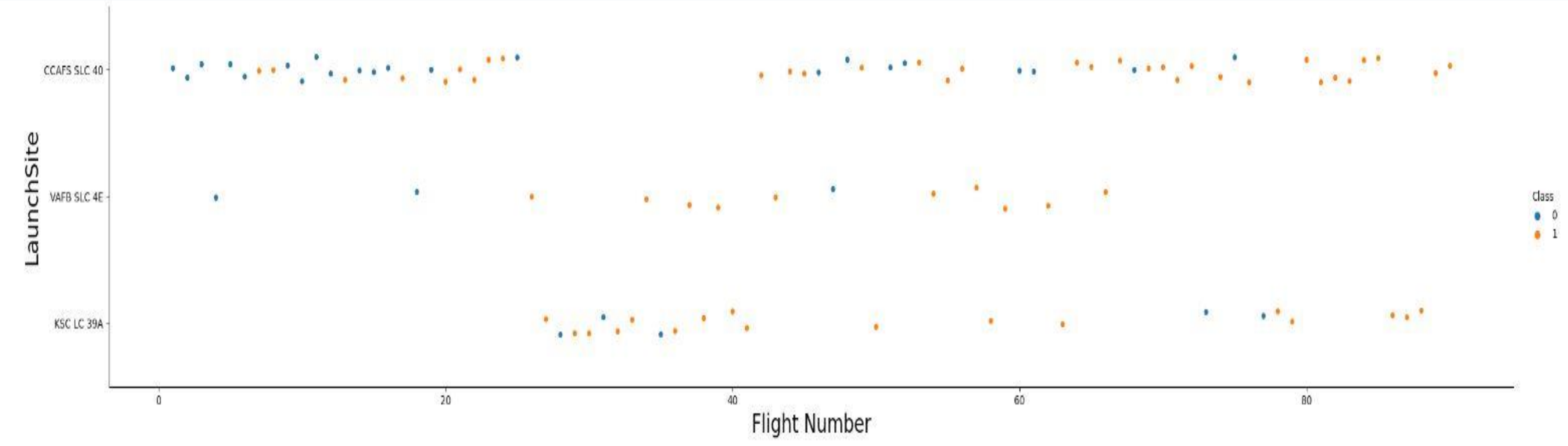
Results

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

Section 2

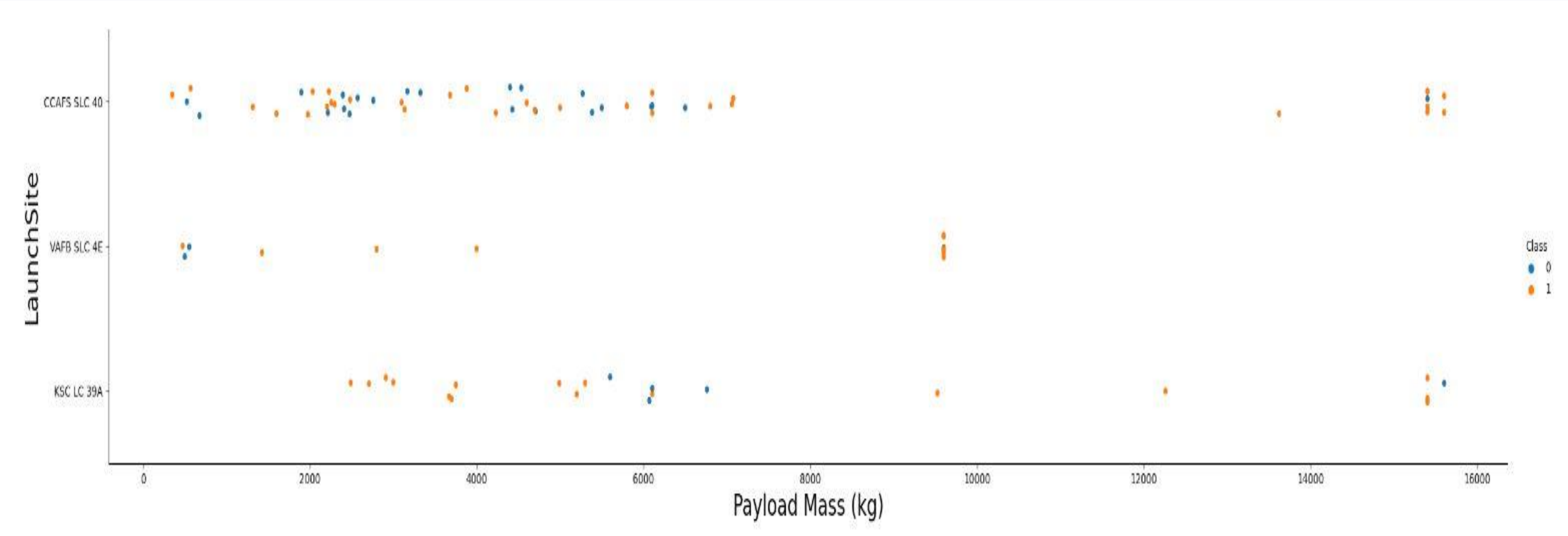
Insights drawn from EDA

Flight Number vs. Launch Site



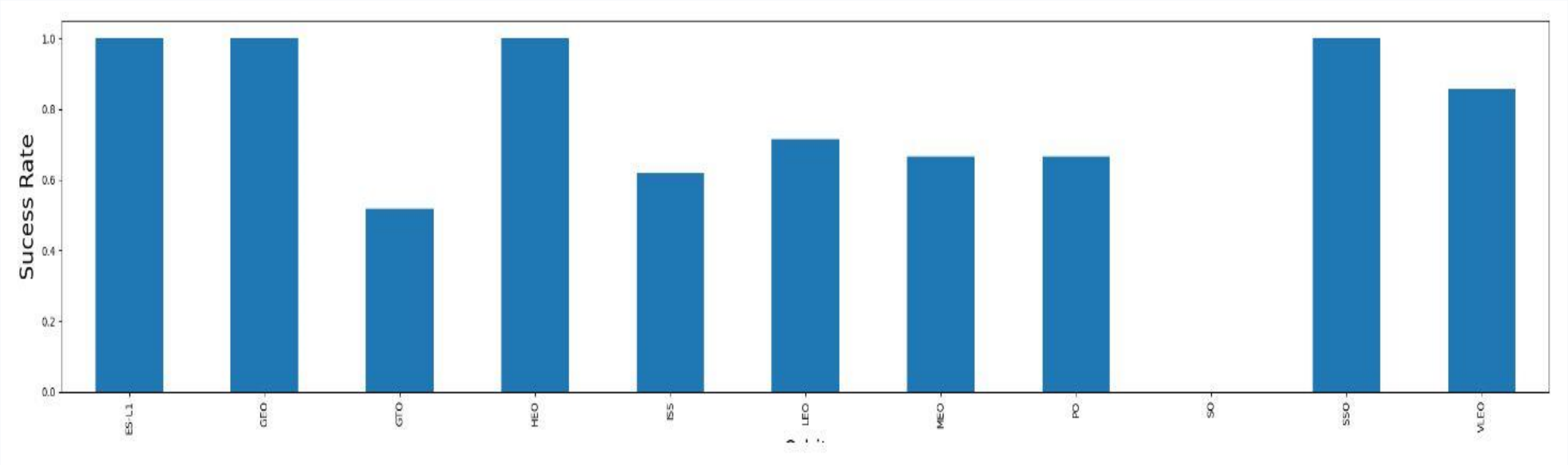
- Launch site CCFAS SLC 40 has a strong relationship with flight number while KSC LC 39A has no relationship before flight 22.

Payload vs. Launch Site



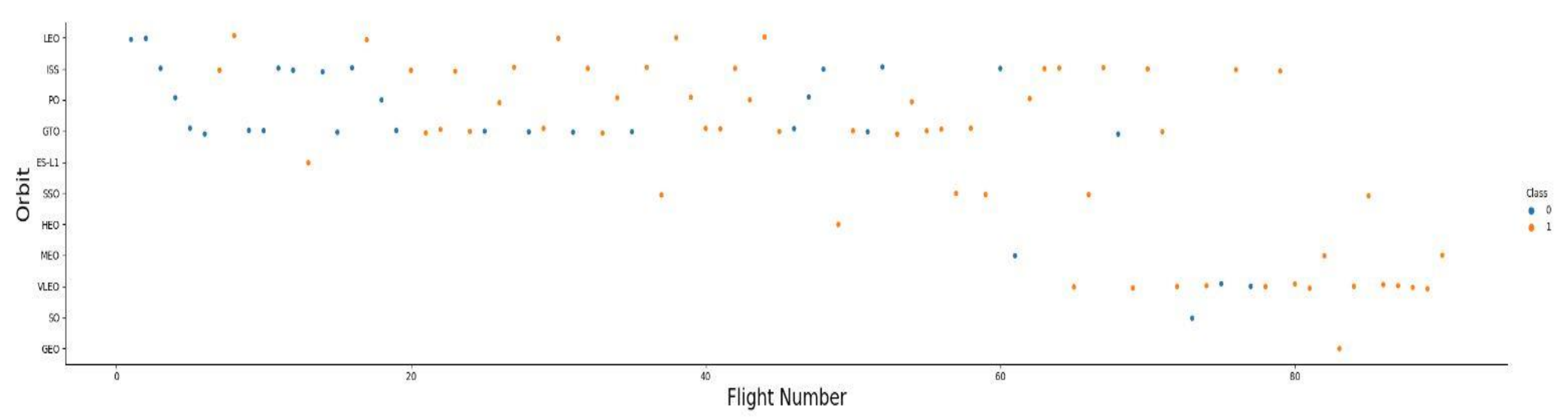
- For Launch Site VAFB SLC 4E there are no rockets launched for heavy payload mass(greater than 10000).

Success Rate vs. Orbit Type



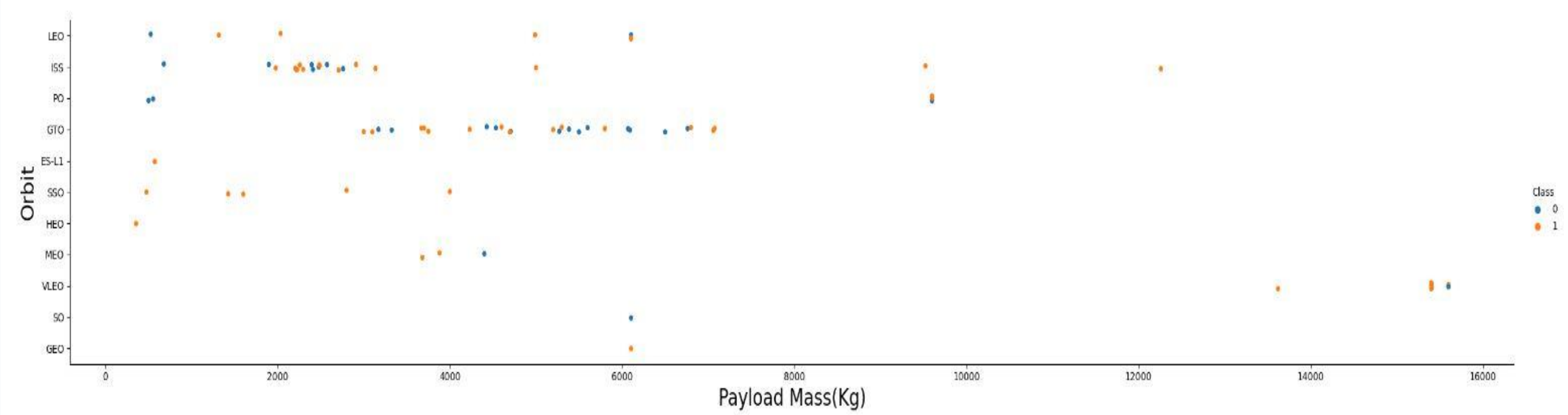
- ES-L1,GEO,HEO and SSO Orbit types have the highest success rates while GTO has the least.

Flight Number vs. Orbit Type



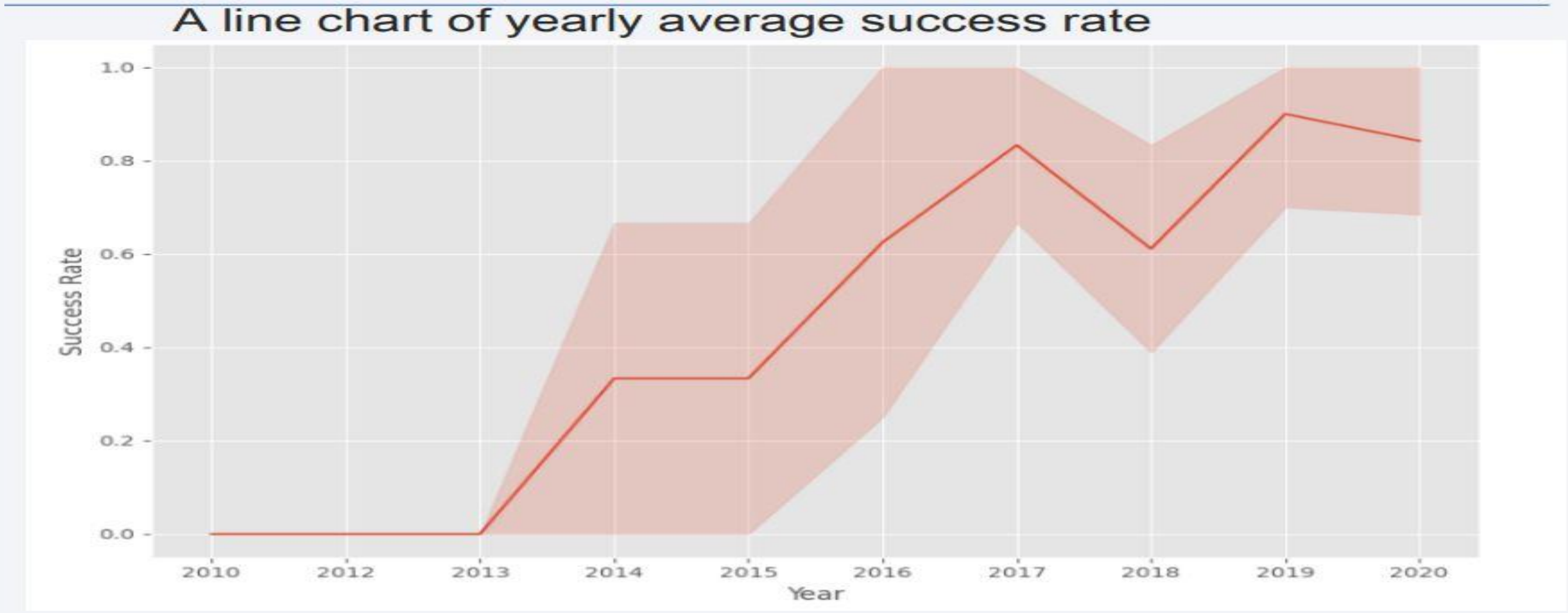
- In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



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

Launch Success Yearly Trend



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

All Launch Site Names

```
In [6]: %sql select Unique(LAUNCH_SITE) from SPACEXTBL;

 * ibm_db_sa://ktf76410:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.

Out[6]:
```

launch_site
CCAFS LC-40
CCAFS SLC-40
CCAFSSLC-40
KSC LC-39A
VAFB SLC-4E

Launch Site Names Begin with 'CCA'

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

```
%sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

```
 * ibm_db_sa://ktf76410:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3132
1/bludb
Done.
```

payloadmass
619967

- Total Payload Mass is **619967.0 Kg**

Average Payload Mass by F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

```
 * ibm_db_sa://ktf76410:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3132
1/bludb
Done.
```

payloadmass
6138

First Successful Ground Landing Date

```
%sql select min(DATE) from SPACEXTBL;
```

```
 * ibm_db_sa://ktf76410:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:31321/bludb
Done.
```

1
2010-06-04

- The first successful landing outcome on ground pad was 01/06/2014.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
In [12]: %sql select BOOSTER_VERSION from SPACEXTBL where LANDING__OUTCOME='Success (drone ship)' and PAYLOAD_MASS__KG_ BETW

 * ibm_db_sa://ktf76410:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3132
1/bludb
Done.
```

Out[12]:

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

Total Number of Successful and Failure Mission Outcomes

In [16]:

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

```
 * ibm_db_sa://ktf76410:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:3132
1/bludb
Done.
```

Out[16]:

missionoutcomes
1
99
1

Boosters Carried Maximum Payload

```
%sql SELECT "Booster_Version",Payload, "PAYLOAD_MASS__KG_" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

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

Booster_Version	Payload	PAYLOAD_MASS__KG_
F9 B5 B1048.4	Starlink 1 v1.0, SpaceX CRS-19	15600
F9 B5 B1049.4	Starlink 2 v1.0, Crew Dragon in-flight abort test	15600
F9 B5 B1051.3	Starlink 3 v1.0, Starlink 4 v1.0	15600
F9 B5 B1056.4	Starlink 4 v1.0, SpaceX CRS-20	15600
F9 B5 B1048.5	Starlink 5 v1.0, Starlink 6 v1.0	15600
F9 B5 B1051.4	Starlink 6 v1.0, Crew Dragon Demo-2	15600
F9 B5 B1049.5	Starlink 7 v1.0, Starlink 8 v1.0	15600
F9 B5 B1060.2	Starlink 11 v1.0, Starlink 12 v1.0	15600
F9 B5 B1058.3	Starlink 12 v1.0, Starlink 13 v1.0	15600
F9 B5 B1051.6	Starlink 13 v1.0, Starlink 14 v1.0	15600
F9 B5 B1060.3	Starlink 14 v1.0, GPS III-04	15600
F9 B5 B1049.7	Starlink 15 v1.0, SpaceX CRS-21	15600

2015 Launch Records

1	mission_outcome	booster_version	launch_site
1	Success	F9 v1.1 B1012	CCAFS LC-40
2	Success	F9 v1.1 B1013	CCAFS LC-40
3	Success	F9 v1.1 B1014	CCAFS LC-40
4	Success	F9 v1.1 B1015	CCAFS LC-40
4	Success	F9 v1.1 B1016	CCAFS LC-40
6	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

```
%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.
```

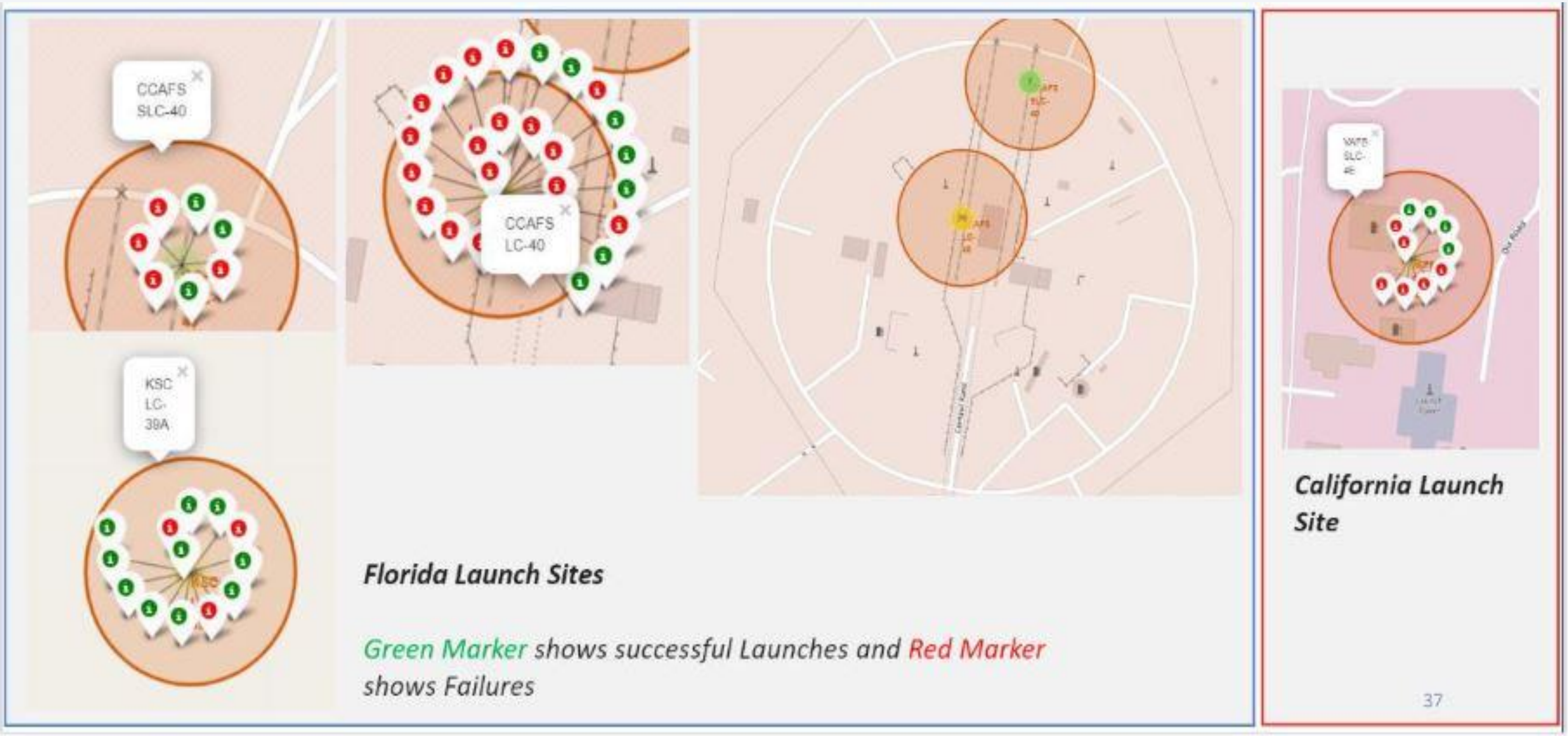
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
19-02-2017	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
18-10-2020	12:25:57	F9 B5 B1051.6	KSC LC-39A	Starlink 13 v1.0, Starlink 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

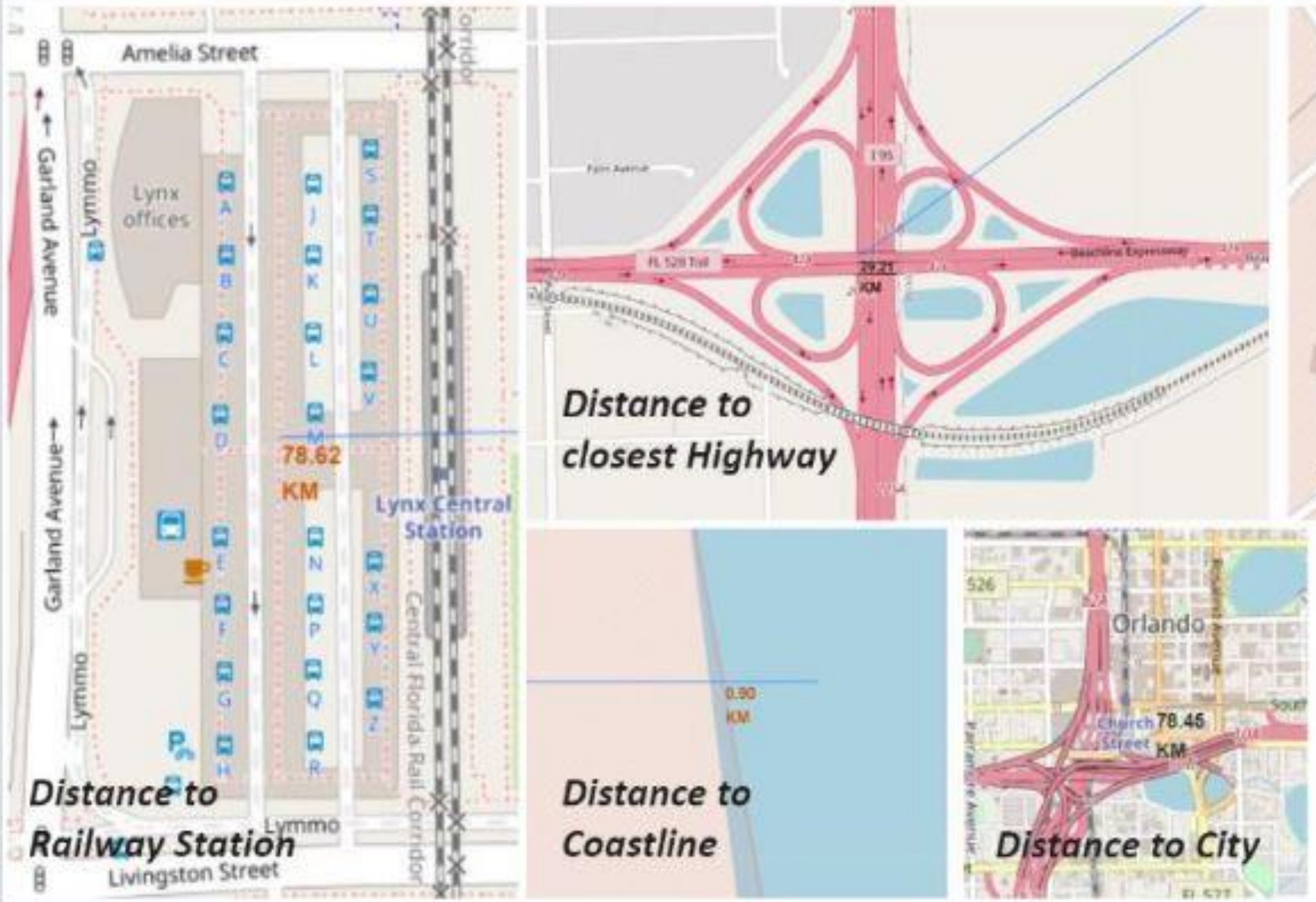
Launch Sites Proximities Analysis

Spacex Launch Sites

SpaceX Launch Outcomes



SpaceX Launch Site Proximity

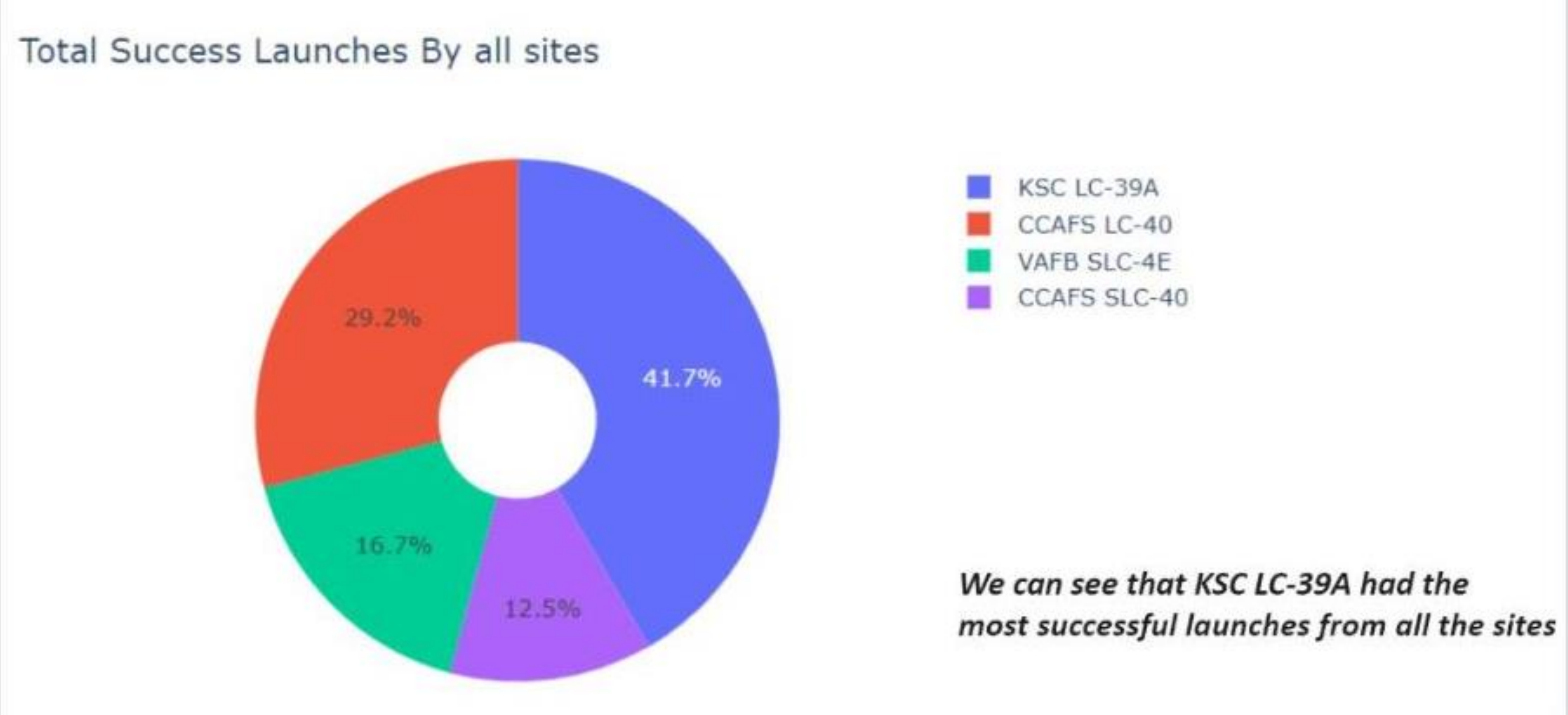


- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

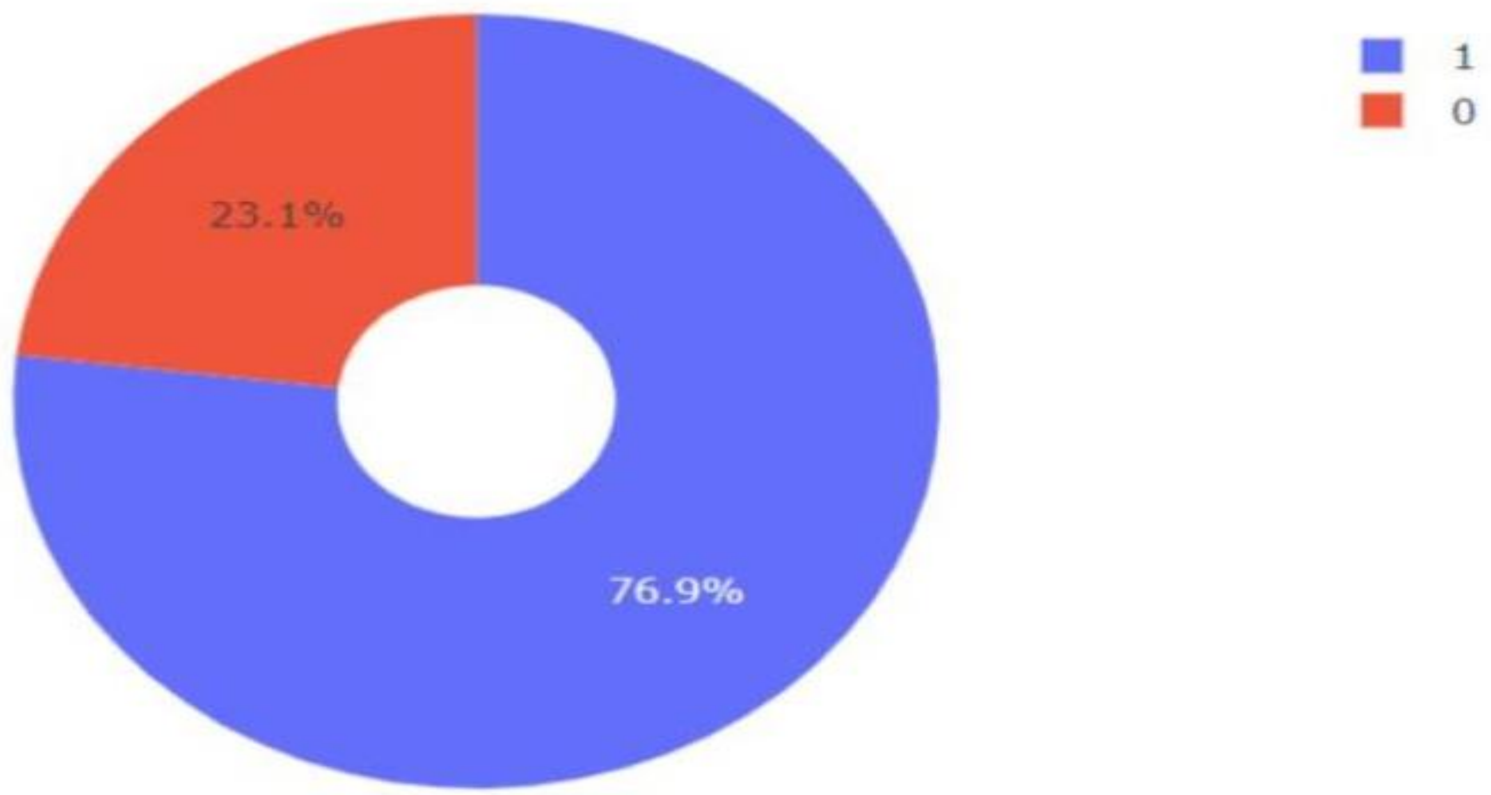
Section 4

Build a Dashboard with Plotly Dash

SPaceX Launch Success By Sites

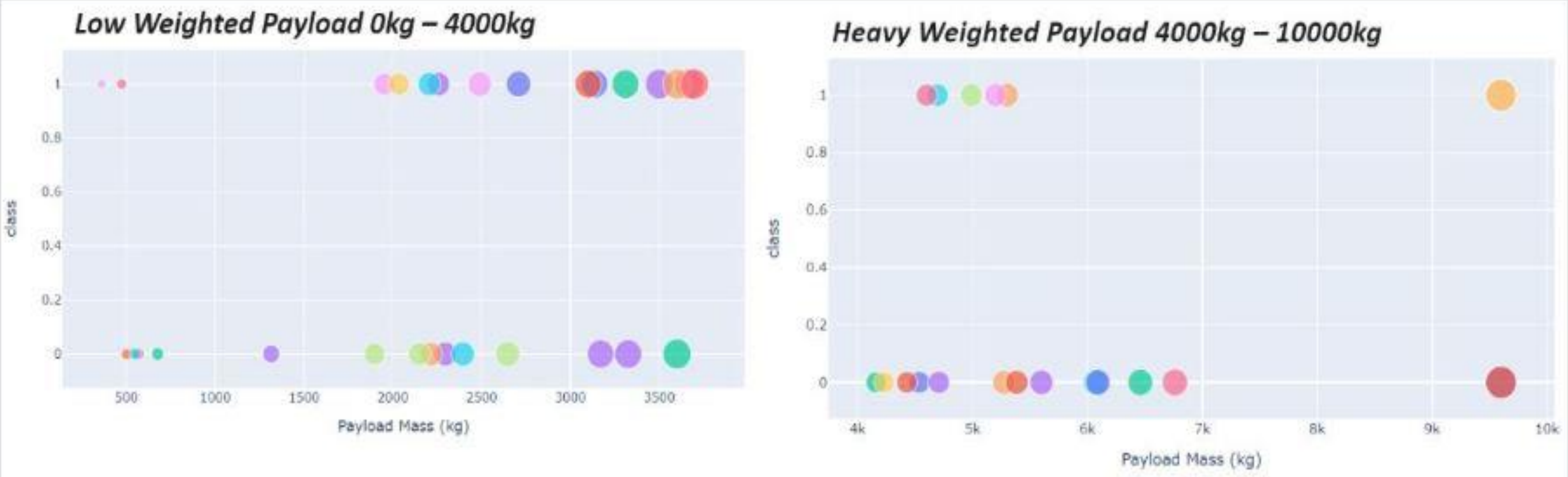


Launch Sites Success Rates



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

Payload Vs Launch Outcomes For Sites



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

Section 5

Predictive Analysis (Classification)

Classification Accuracy

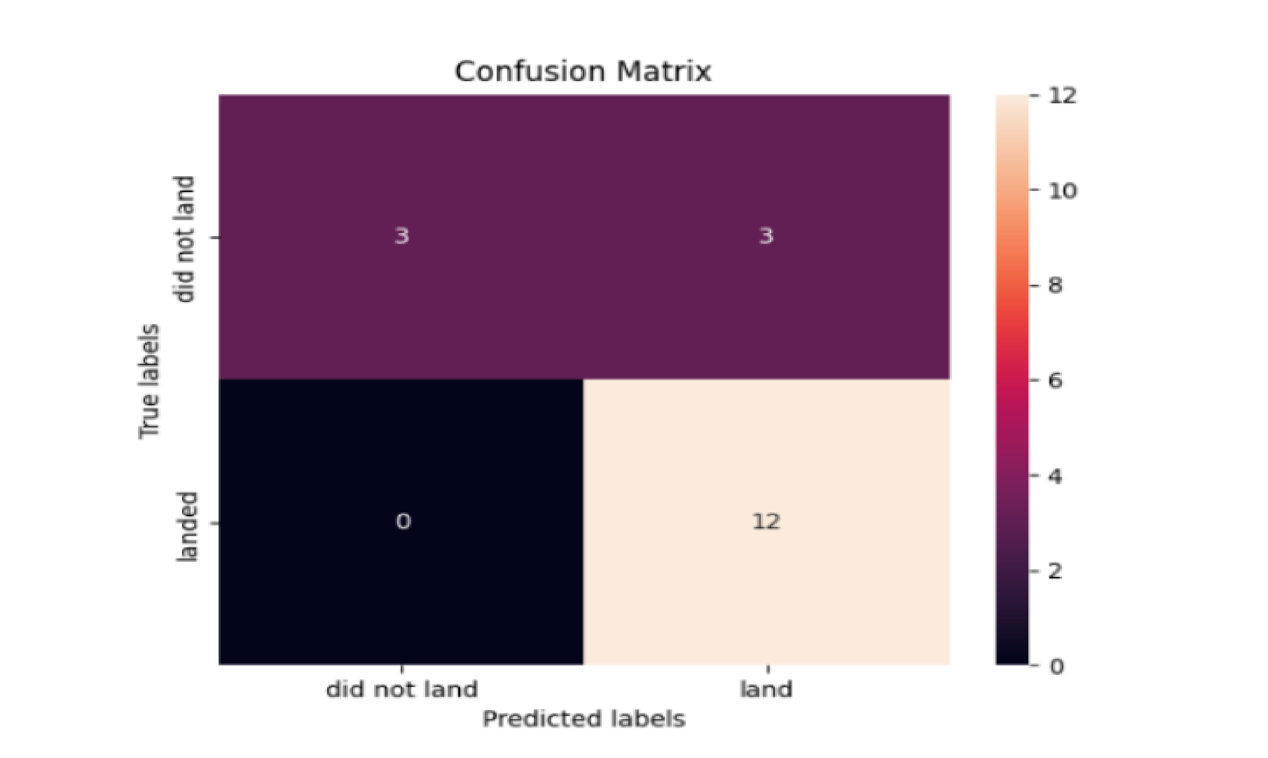
Out[68]:

	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

All the methods perform equally on the test data: i.e. They all have the same accuracy of 0.833333 on the test Data

Confusion Matrix

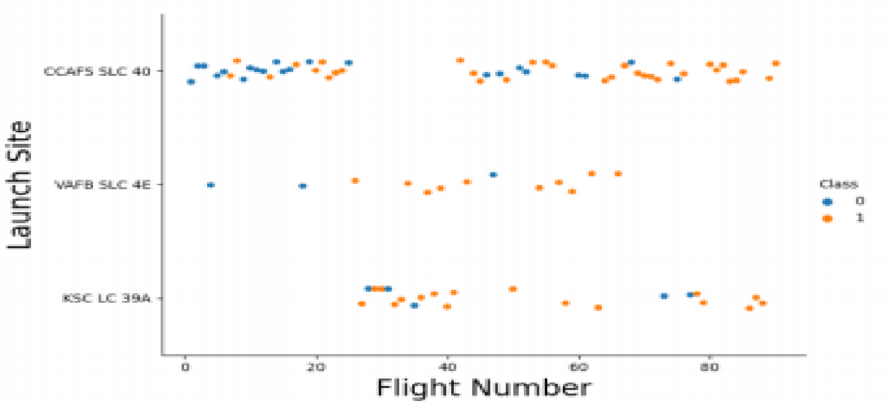
- All the 4 classification models had the same confusion matrixes and were able equally distinguish between the different classes. The major problem is false positives for all the models.

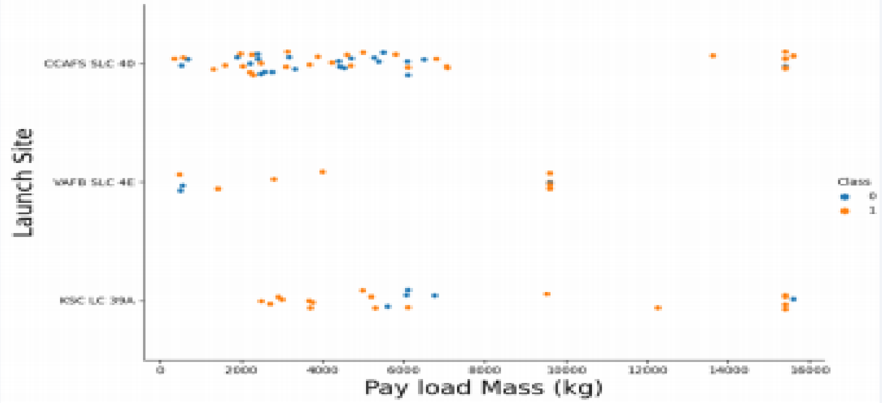


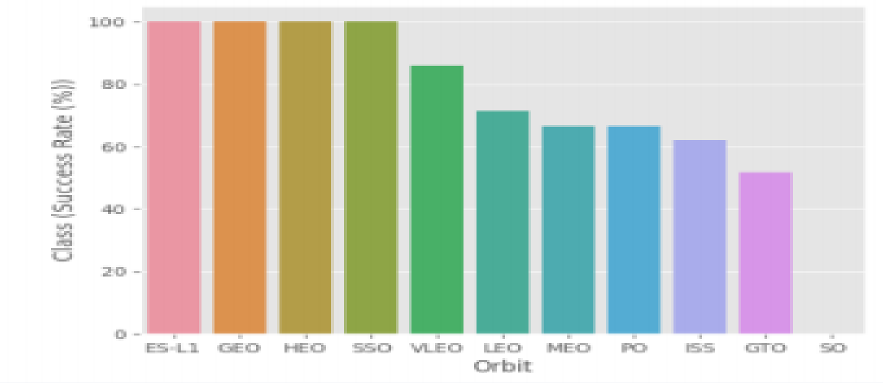
Conclusions

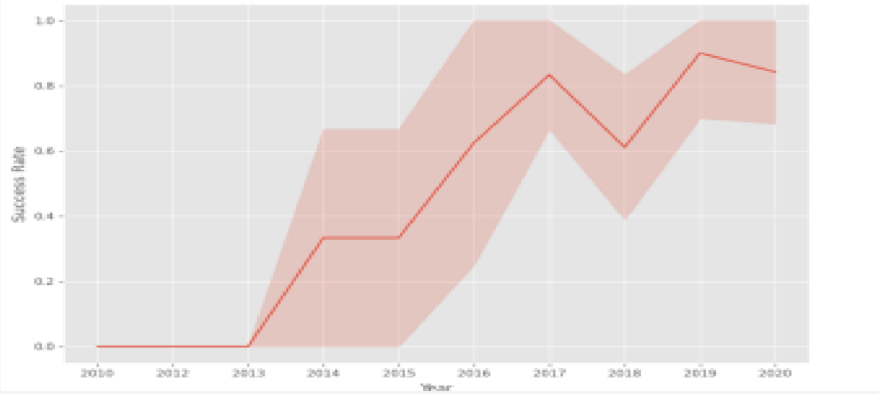
- Different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.
- It can be deduced that, as the flight number increases in each of the 3 launch sites, so does the success rate. For instance, the success rate 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 .
- Orbits ES-L1, GEO, HEO & SSO have the highest success rates at 100%, with SO orbit having the lowest success rate at 50%. Orbit SO has 0% success rate.
- 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 present.
- Finally the success rate since 2013 kept increasing till 2020.

Appendix


Launch Site
CCAFS SLC 40
VAFB SLC 4E
KSC LC 39A
Class
0
1
0
20
40
60
80
Flight Number


Launch Site
CCAFS SLC 40
VAFB SLC 4E
KSC LC 39A
Class
0
1
Pay load Mass (kg)


Class (Success Rate (%))
100
80
60
40
20
0
ES-L1
GEO
HEO
SSO
VLEO
LEO
MEO
PO
ISS
GTO
SO
Orbit


Success Rate
1.0
0.8
0.6
0.4
0.2
0.0
2010
2012
2013
2014
2015
2016
2017
2018
2019
2020

Thank you!