IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Suma
14-Dec-2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection using API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis using SQL
  - EDA Data Visualization
  - Launch Sites Analysis with Folium
  - Dashboard with PlotyDash
  - Predictive Analysis (Classification)

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

Introduction

Project background and context

The commercial space age is here, companies are making space travel affordable for everyone. Virgin Galactic,Rocket Lab,Blue Origin Starlink, Space X are Isome of them. Space X is the most successful one. One reason SpaceX can do this is the rocket launches are relatively inexpensive. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards 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.Spaces X's Falcon if the first stage will land successfully, you will train a machine learning model and use public information to predict if SpaceX will reuse the first stag 9 launch like regular rockets.

Problems you want to find answers

- Determine if SpaceX will reuse the first stage
- Determine if the first stage will land successfully,
- Using public information and machine learning model predict if SpaceX will reuse the first stage.
- Effect of variables payload mass, launch site, number of flights and orbits on first stage landing

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

  SpaceX Rest API

  Web Scrapping from Wikipedia

- Perform data wrangling

  Exploratory data analysis to find patterns and determine the label

- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - LR,KNN,SVM,DT models have been built and evaluated to find best classifier

Data Collection

SpaceX Rest API

The SpaceX REST API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.

We will perform a get request using the requests library to obtain the launch data, which we will use to get the data from the API. This result can be viewed by calling the .json() method. Our response will be in the form of a JSON, specifically a list of JSON objects. Since we are using an API, you will notice in the lab that when we get a response it is in the form of a JSON. Specifically, we have a list of JSON objects which each represent a launch. To convert this JSON to a dataframe, we can use the json_normalize function. This function will allow us to “normalize” the structured json data into a flat table. This is what your JSON will look like in a table form.

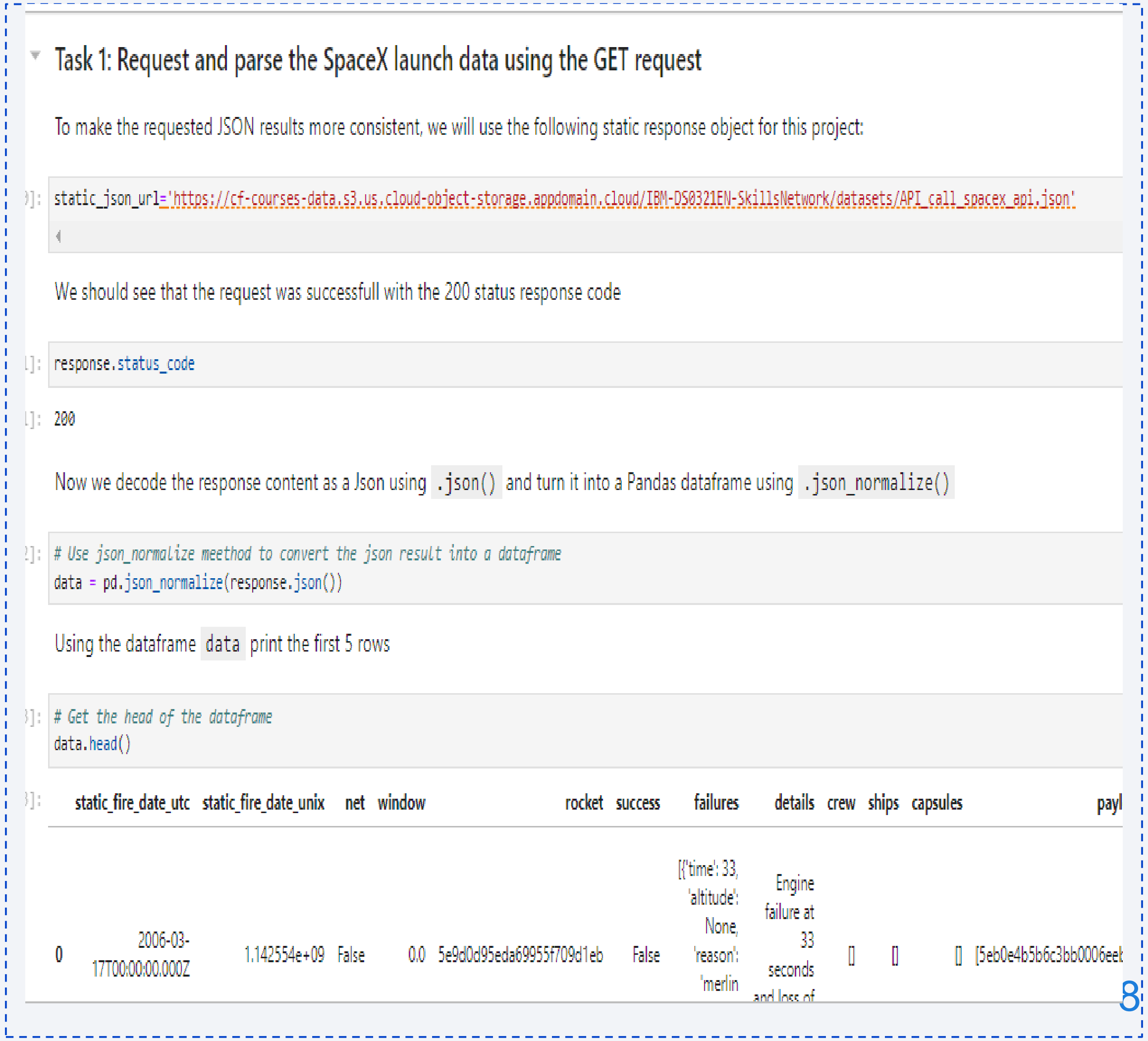
Web Scraping

Web scraping related Wiki pages. using the Python BeautifulSoup package to web scrape some HTML tables that contain valuable Falcon 9 launch records. Then you need to parse the data from those tables and convert them into a Pandas data frame for further visualization and analysis. We want to transform this raw data into a clean dataset which provides meaningful data on the situation we are trying to address: Wrangling Data using an API, Sampling Data, and Dealing with Nulls.

Data Collection – SpaceX API

- Request and parse the SpaceX launch data using the GET request.Decode the response content as a Json using .json()
- Convert it into a Pandas dataframe using .json_normalize()
- Filter the dataframe to only include Falcon 9 launches
- Missing Values. Replace missing values with mean for the Payload column

- **GitHub URL:** https://github.com/Sumaxx/TestRepo1/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Request the Falcon9 Launch Wiki page from its URL. Perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.
- Create a BeautifulSoup object from the HTML response
- Extract all column/variable names from the HTML table header
- Create a data frame by parsing the launch HTML tables
- **GitHub URL:** https://github.com/Sumaxx/TestRepo1/blob/main/jupyter-labs-webscraping%20(1).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.

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

    data  = requests.get(static_url).text
    #print(data)
```

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

```
6]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
    #soup = BeautifulSoup(data, 'html5lib')
    soup=BeautifulSoup(data, 'html.parser')
```

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

```
7]: # Use soup.title attribute
    print(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 th

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

Data Wrangling

- Exploratory Data Analysis (EDA) to find some patterns in the data and determine Training Labels
- Calculated the number of launches on each site
- Calculated the number and occurrence of each orbit
- Calculated the number and occurrence of mission outcome of the orbits
- Create a landing outcome label from Outcome column

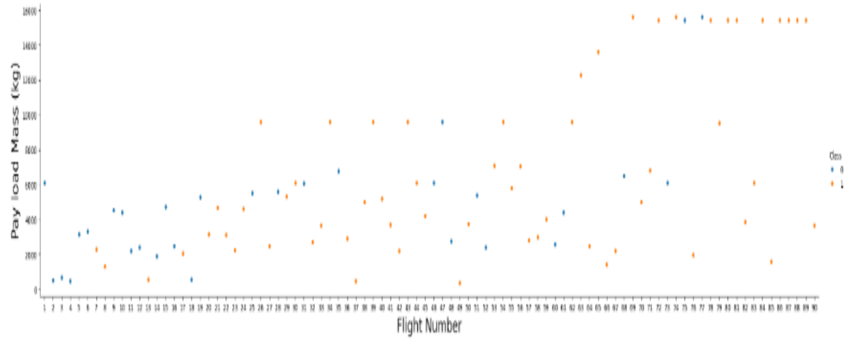
- **GitHub URL** : https://github.com/Sumaxx/TestRepo1/blob/main/labs-jupyter-spacex-Data%20wrangling%20(1).ipynb

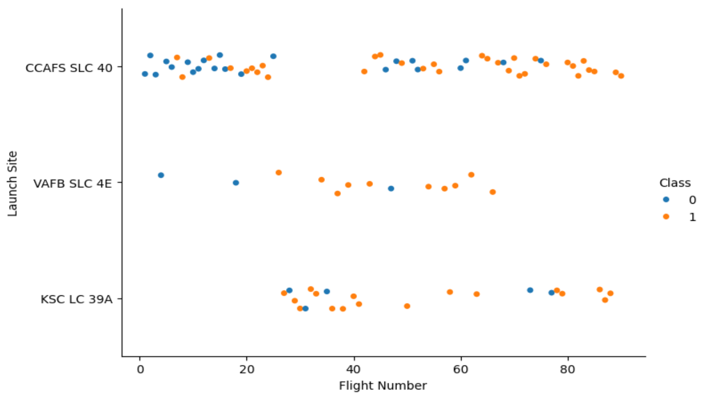
EDA with Data Visualization

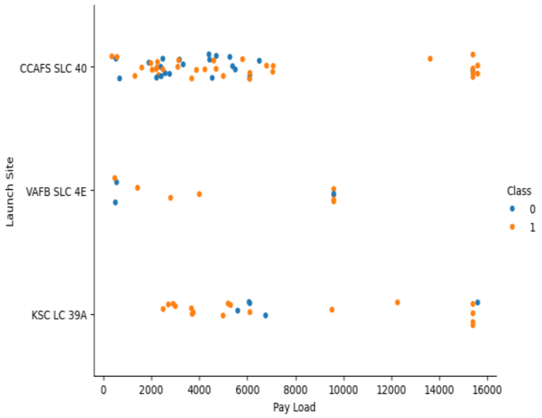
- Scatter plot showing FlightNumber vs. PayloadMass
- Scatter plot showing Flight Number and Launch Site
- Scatter plot showing the relationship between Payload and Launch Site
- Bar Chart showing the relationship between success rate of each orbit type
- Scatter plot showing the relationship between FlightNumber and Orbit type
- Scatter plot showing the relationship between Payload and Orbit type
- Line Chart with launch success yearly trend

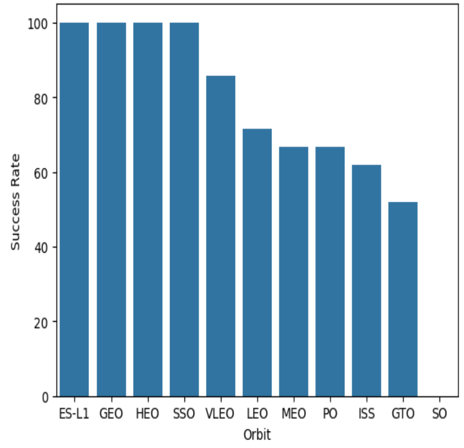
GitHub URL:https://github.com/Sumaxx/TestRepo1/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

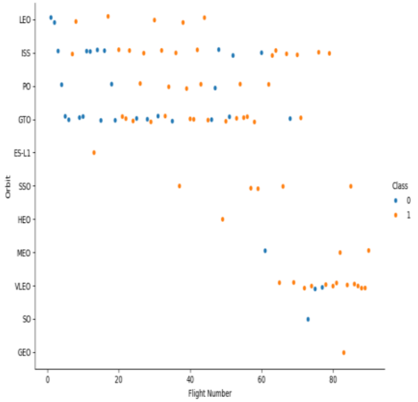
EDA with Data Visualization

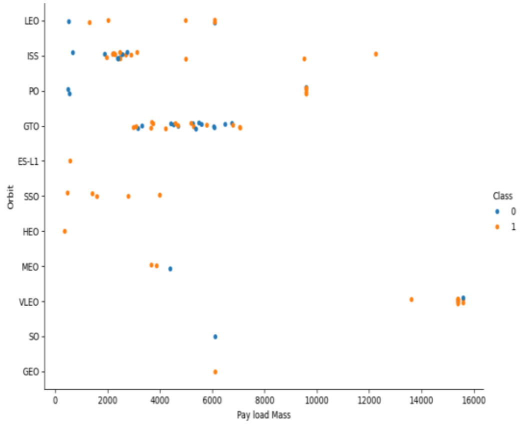


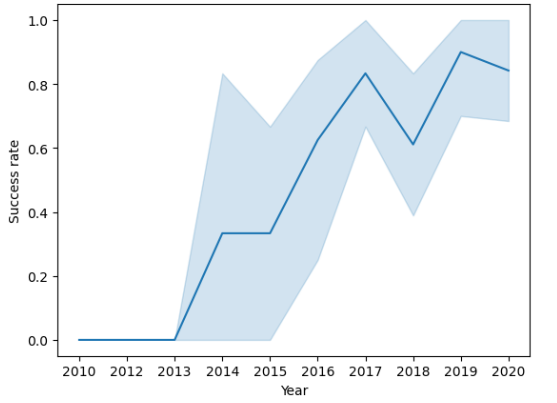












EDA with SQL

- Following EDA are performed using SQL
- 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 successful landing outcome in ground pad was achieved.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster versions which have carried the maximum payload mass. Use a subquery
- 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.
- 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.

- **GitHub URL :https://github.com/Sumaxx/TestRepo1/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb**

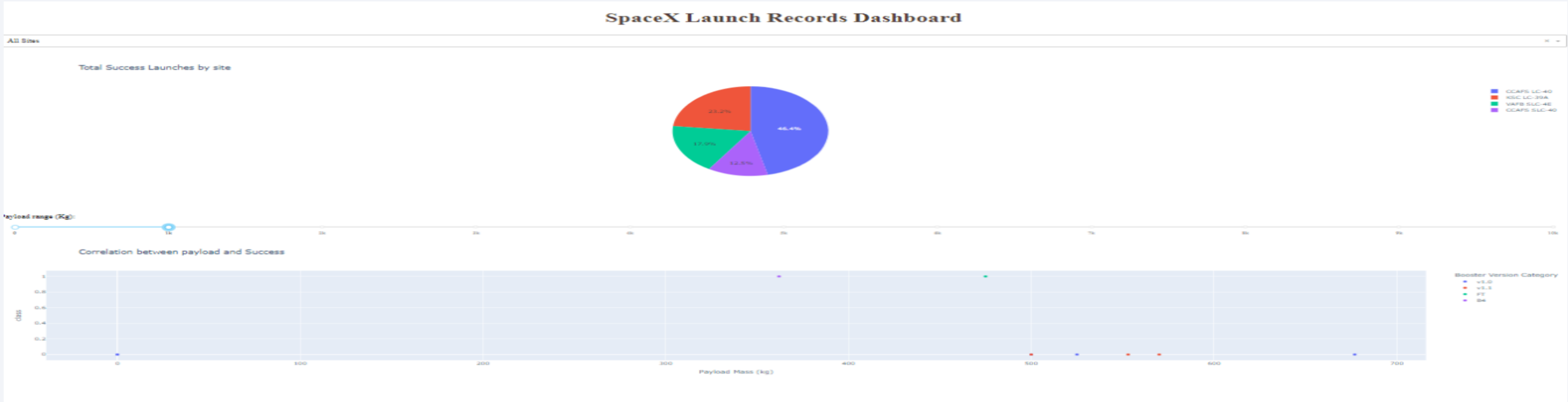
Build an Interactive Map with Folium

- Marked all launch sites on a map using folium.circle and folium.marker
- Marked the success/failed launches for each site on the map in different colurs
- Calculated the distances between a launch site to its proximities and marked

With this analysis was able to find some geographical patterns about launch sites.

- **GitHub URL :** https://github.com/Sumaxx/TestRepo1/blob/main/lab_jupyter_launch_site_location.jupyterlite%20(2).ipynb

Build a Dashboard with Plotly Dash



1.Which site has the largest successful launches?
2.Which site has the highest launch success rate?
3.Which payload range(s) has the highest launch success rate?
4.Which payload range(s) has the lowest launch success rate?
5.Which F9 Booster version (`v1.0`, `v1.1`, `FT`, `B4`, `B5`, etc.) has the highest launch success rate?

GitHub URL: https://github.com/Sumaxx/TestRepo1/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

Classification Model development and evaluation process

- Create a column for the class
- Standardize the data
- Split into training data and test data
- Create a GridSearchCV object with cv = 10 to find the best parameters
- Apply LogReg,SVM.Decision Tree and KNN models
- Train the models with train data
- Calculate the accuracy for test data
- Examine the confusion metrix for all models

GitHub URL
https://github.com/Sumaxx/TestRepo1/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

- Exploratory data analysis results

  Both API and web scrapping are capable to collect Xspace data

- Interactive analytics demo in screenshots

  Details in the following slides

- Predictive analysis results

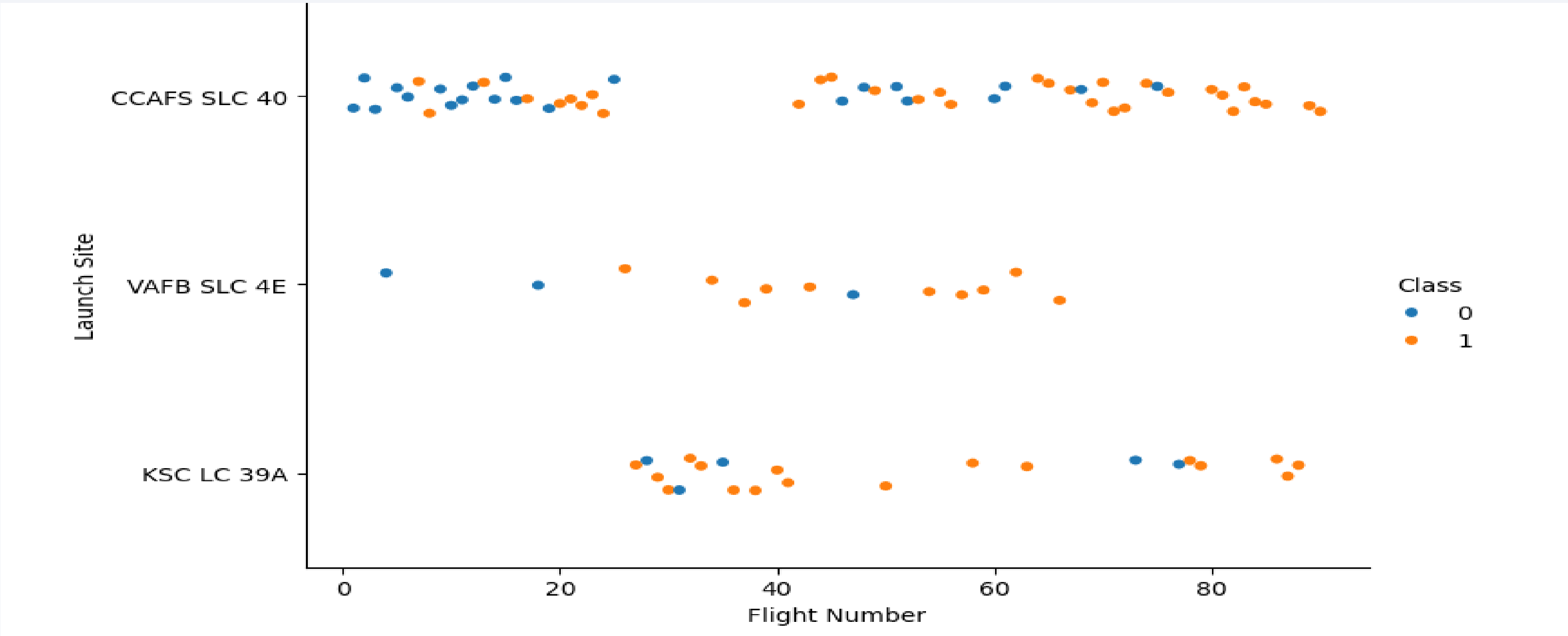
  All the classification models gave the same accuracy

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

Scatter plot of Flight Number vs. Launch Site

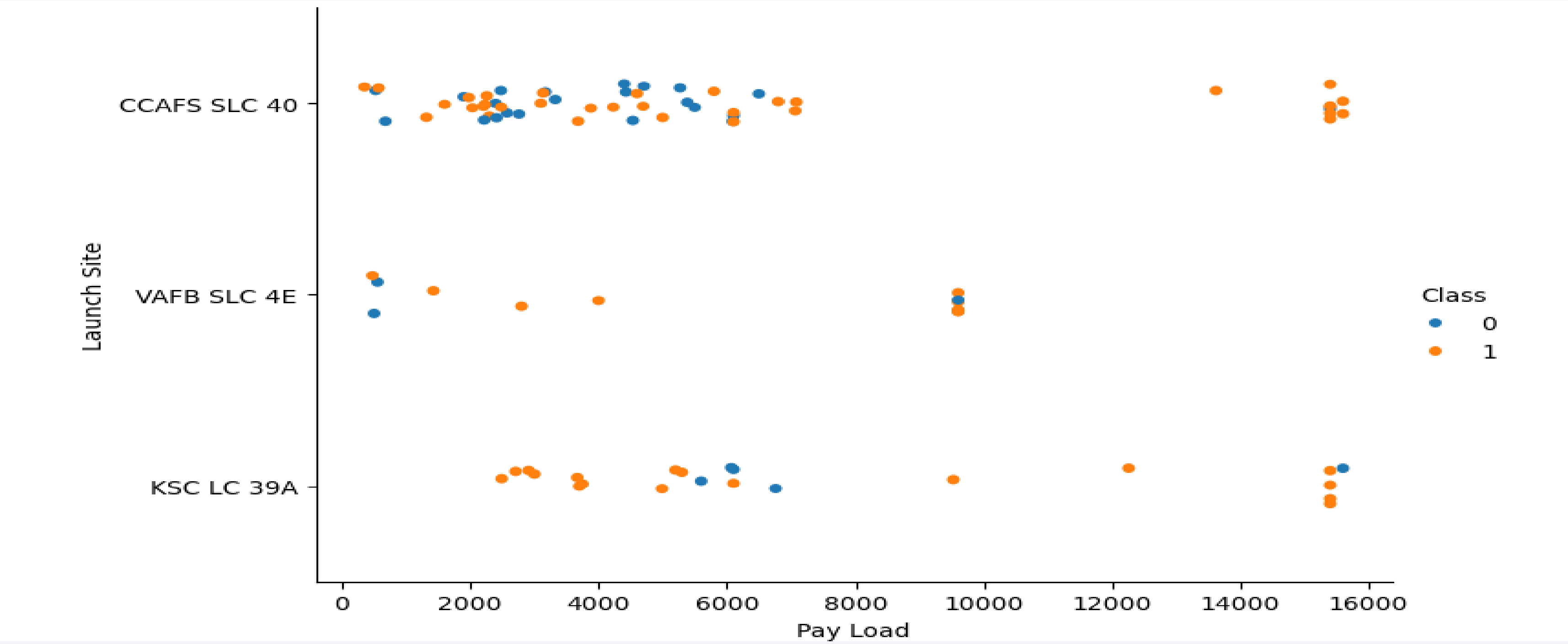


Launch Site CCAFS SLC 40 have more flights.

With increase in flight number success rate increase

Payload vs. Launch Site

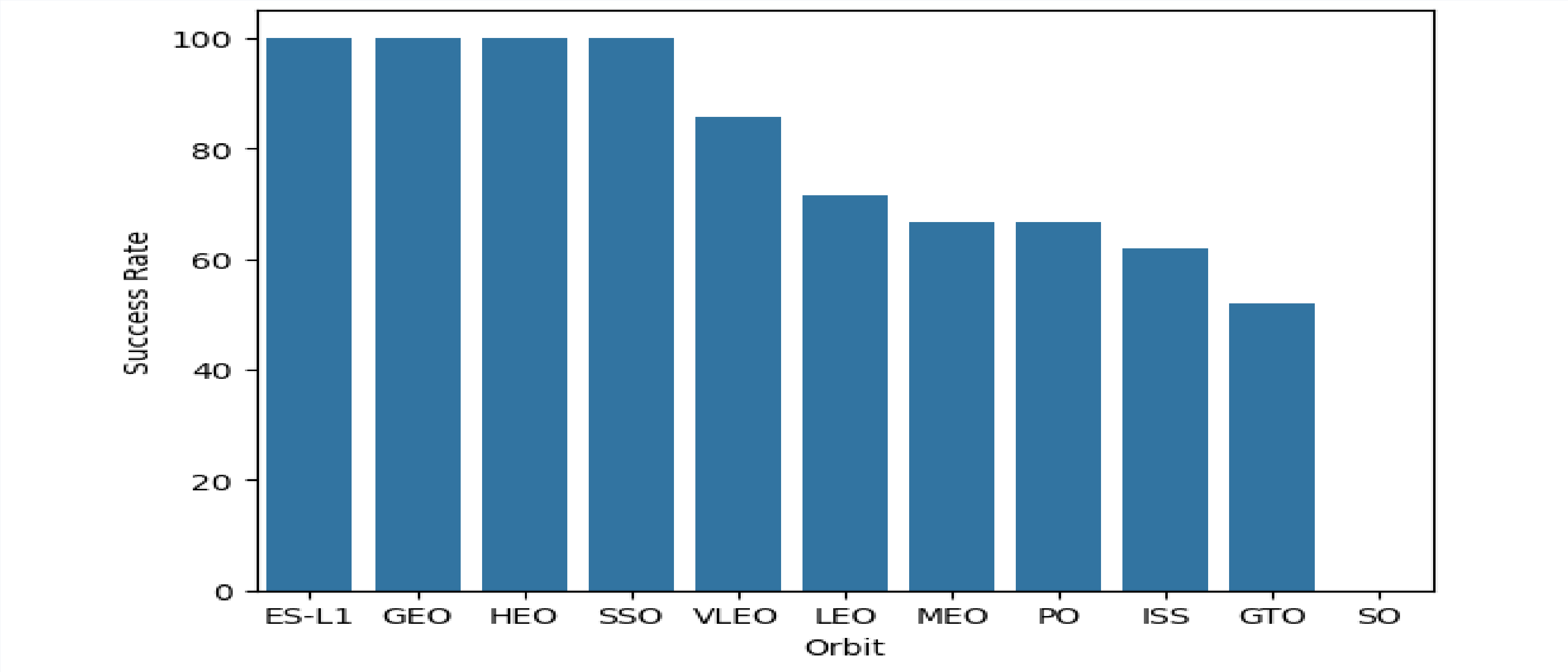
- Scatter plot of Payload vs. Launch Site



VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).
Success rate increases with increase in payload

Success Rate vs. Orbit Type

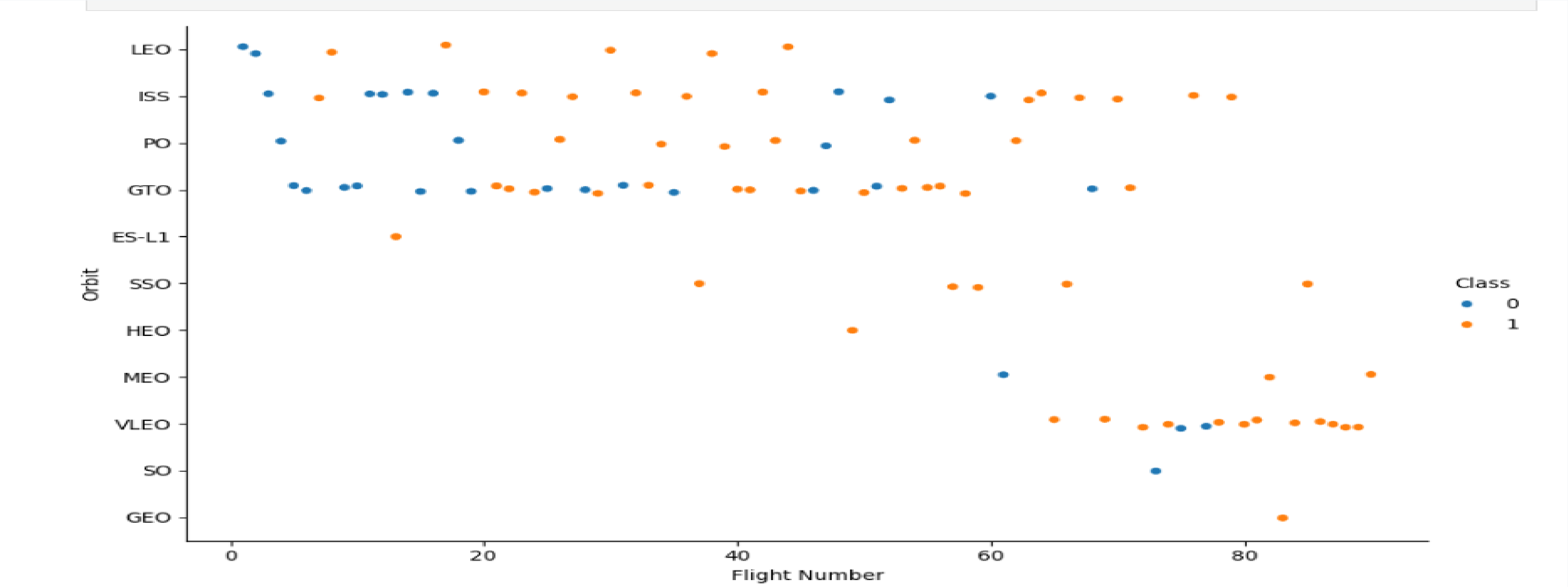
- Bar chart for the success rate of each orbit type



- Orbits ES-L1, GEO ,HEO and SSO have high success rates(100%)
- SO success rate is 0%

Flight Number vs. Orbit Type

- Scatter plot of Flight number vs. Orbit type

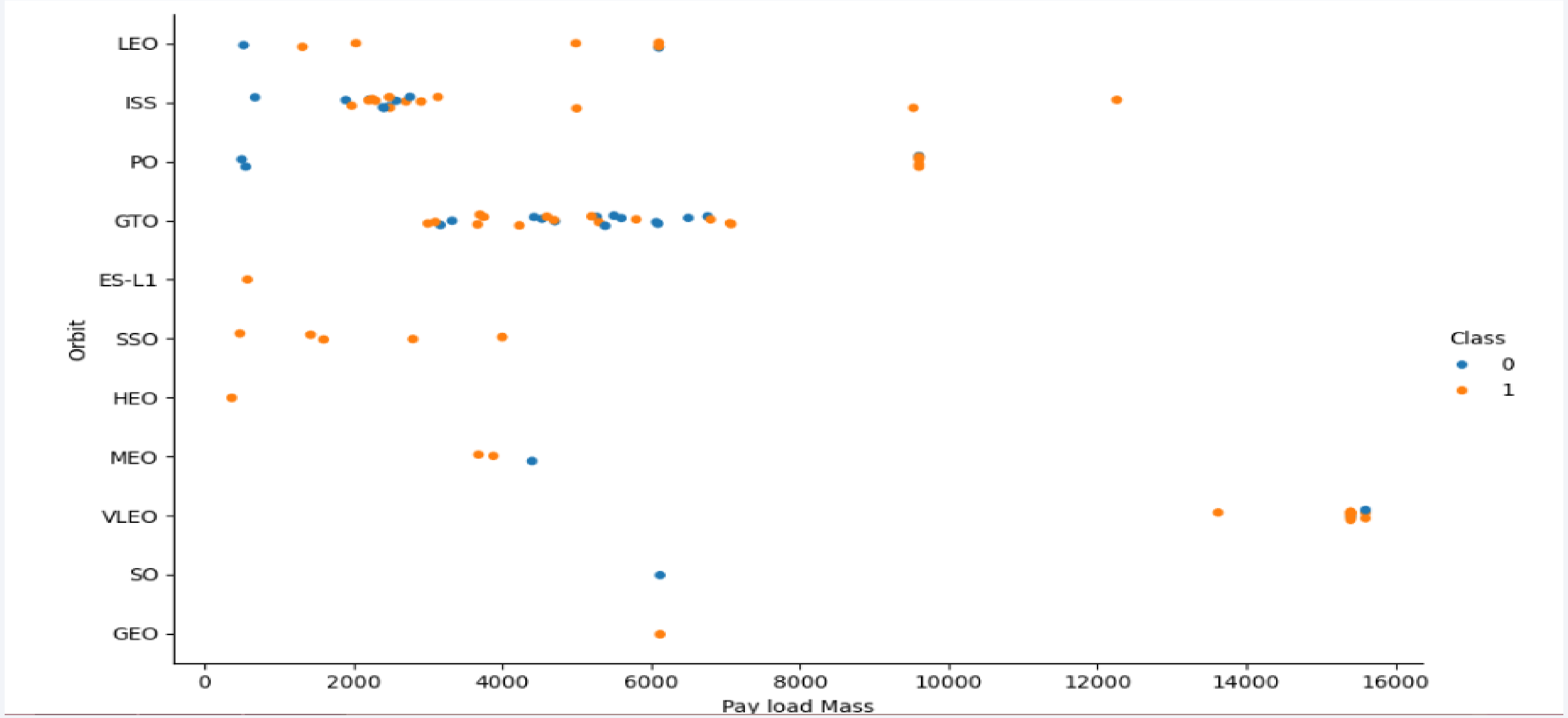


LEO orbit Success appears related to the number of flights;

There seems to be no relationship between flight number when in GEO orbit.

Payload vs. Orbit Type

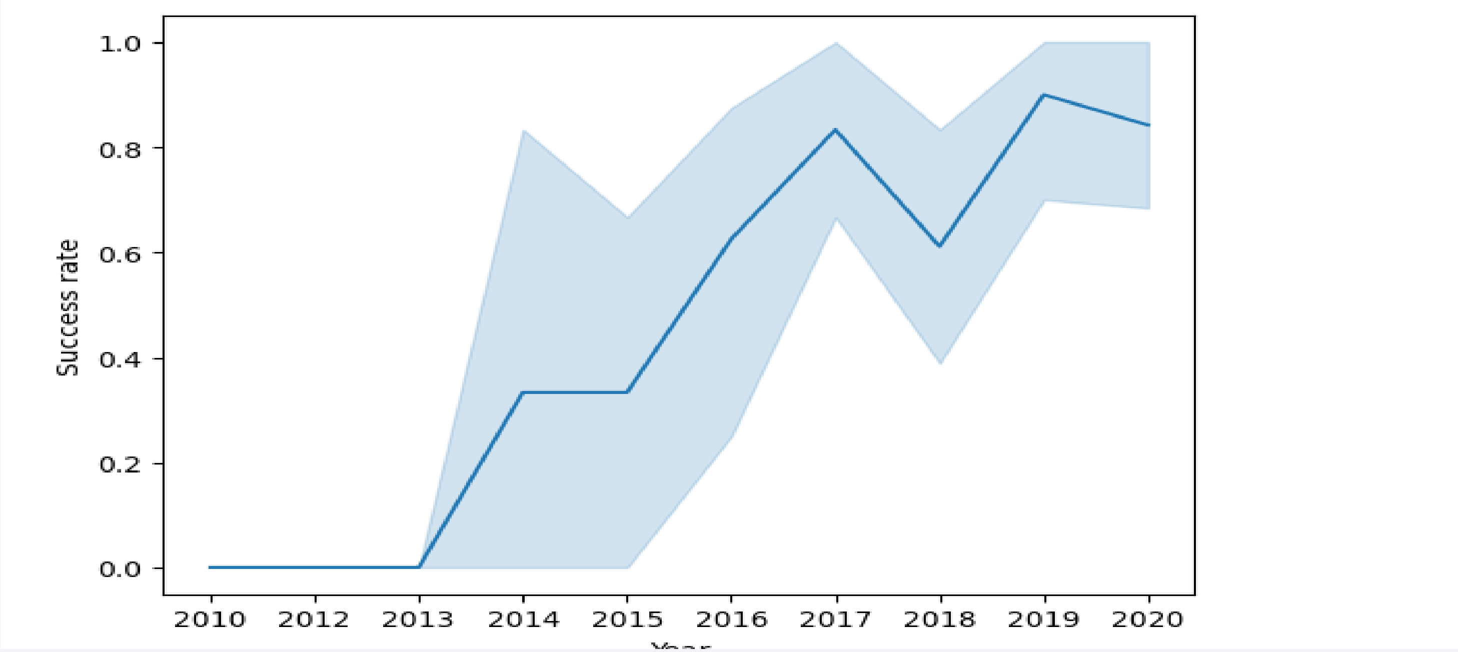
- Scatter point of payload vs. orbit type



With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.

Launch Success Yearly Trend

- Line chart of yearly average success rate



Success rate since 2013 kept increasing till 2020

All Launch Site Names

- Find the names of the unique launch sites.(Using DISTINCT)

Task 1

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

In [10]:

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

Out[10]:

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

Vandenberg AFB Space Launch - VAFB SLC -4E
Kennedy Space Center - KSC LC -39A
CCAFS SLC 40 , CCAFS SLC 40

Launch Site Names Begin with '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_Outc
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parach
2010-12-08	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 (parach
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No atte
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No atte
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No atte

Filter data using LIKE command and LIMIT to 5 rows

Total Payload Mass

Task 3

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

```
%sql SELECT sum(PAYLOAD_MASS__KG_) as Total_Payload_Mass FROM SPACEXTBL where Customer='NASA (CRS)'
```

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

Total_Payload_Mass
45596

Filter NASA data and sum the Payload mass

Average Payload Mass by F9 v1.1

Task 4

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as Avg_Payload_Mass FROM SPACEXTBL where booster_version like 'F9 v1.1%'
```

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

Avg_Payload_Mass
2534.6666666666665

Calculate average using AVG() function for the selected records with booster version F9 v1.1

First Successful Ground Landing Date

Task 5

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

Hint:Use min function

```
0]:  %sql Select min(date) FROM SPACEXTBL where landing_outcome ='Success (ground pad)'
```

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

0]:

min(date)
2015-12-22

Filter Success records and use Min function to find the oldest date

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

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 FROM SPACEXTBL where landing_outcome ='Success (drone ship)' and PAYLOAD_MASS__KG_
```

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

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

Filter Success records with payload between 4000 and 6000 and get the distinct booster versions

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome"
```

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

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

Group by records based on Mission outcome and get the count for each group

Boosters Carried Maximum Payload

Task 8

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

28]:
```
%sql SELECT distinct "Booster_Version" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM
```

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

28]:

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

Used a Subquery to return and pass the Max payload and used it list all the boosters that have carried the Max payload of 15600kgs

2015 Launch Records

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.

[36]:

```
%sql SELECT substr(Date,6,2) as "Month","landing_outcome","Booster_Version", "Launch_Site" FROM SPACEXTBL WHERE substr(
```

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

[36]:

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Filter failure records and get the year from date using substr() function

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

Task 10

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.

```
]: %sql SELECT landing_outcome,count(*) FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' group by landing_o
```

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

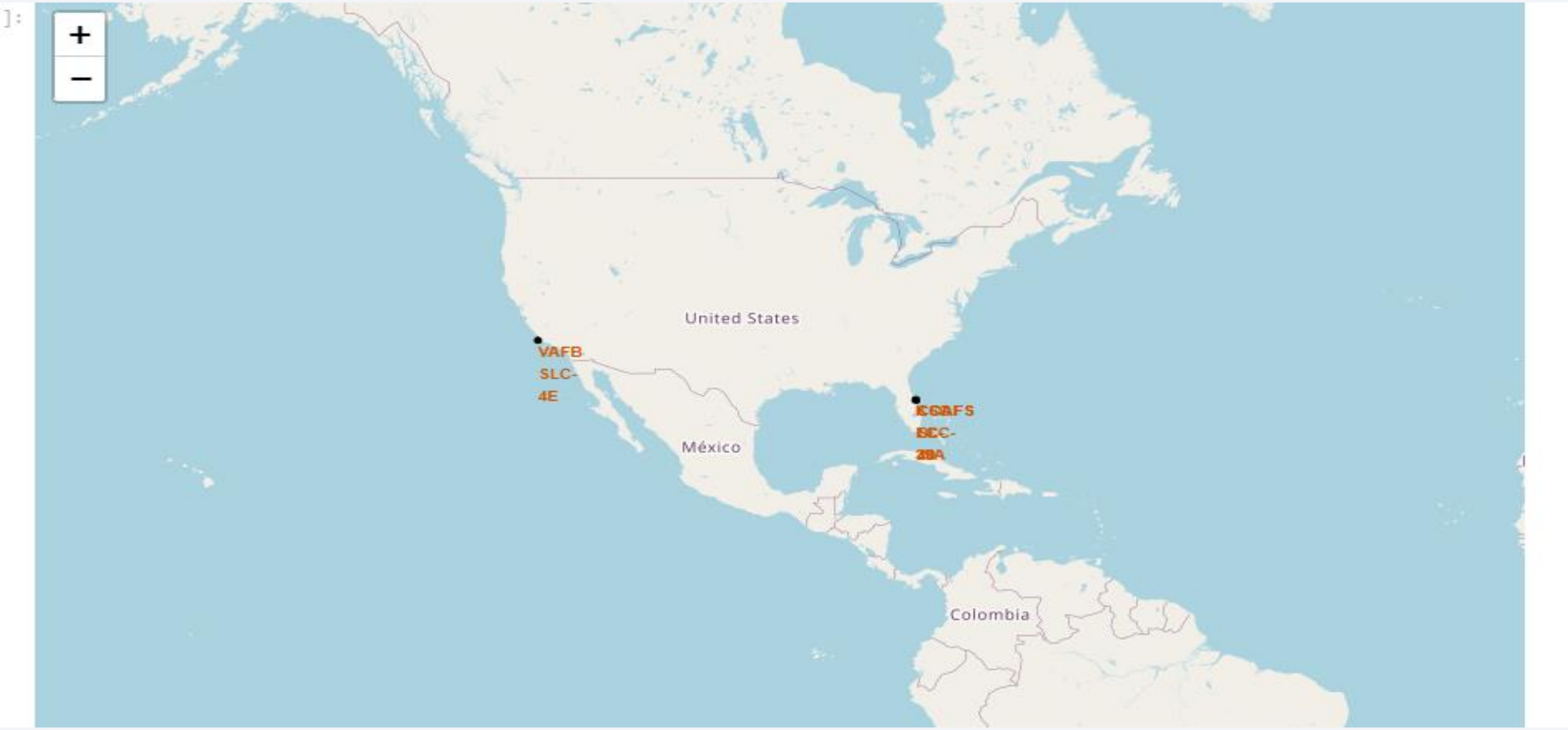
Landing_Outcome	count(*)
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Arrange the data using ORDER BY

Section 3

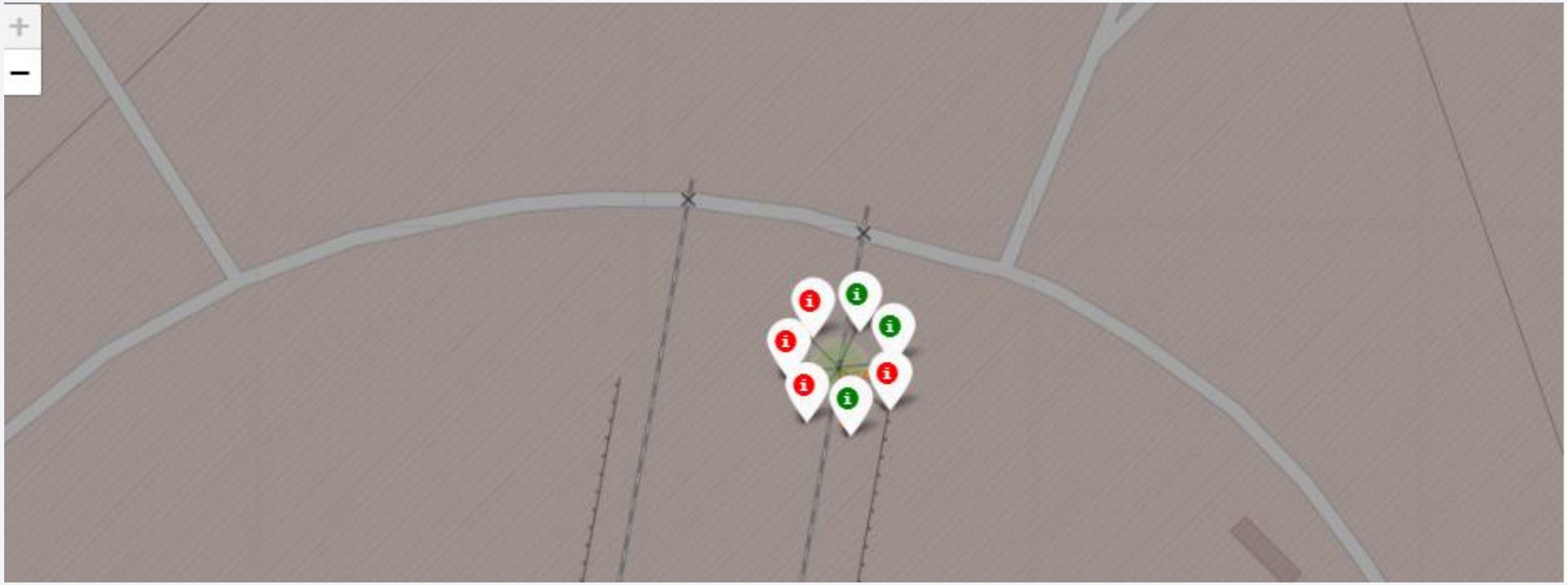
Launch Sites Proximities Analysis

Launch sites on a global map



- All Launch sites are marked on the global map

Launch Outcome(Success/Failure)



- Launch outcomes(Success/Failure) for each site are marked in different colors based on outcome

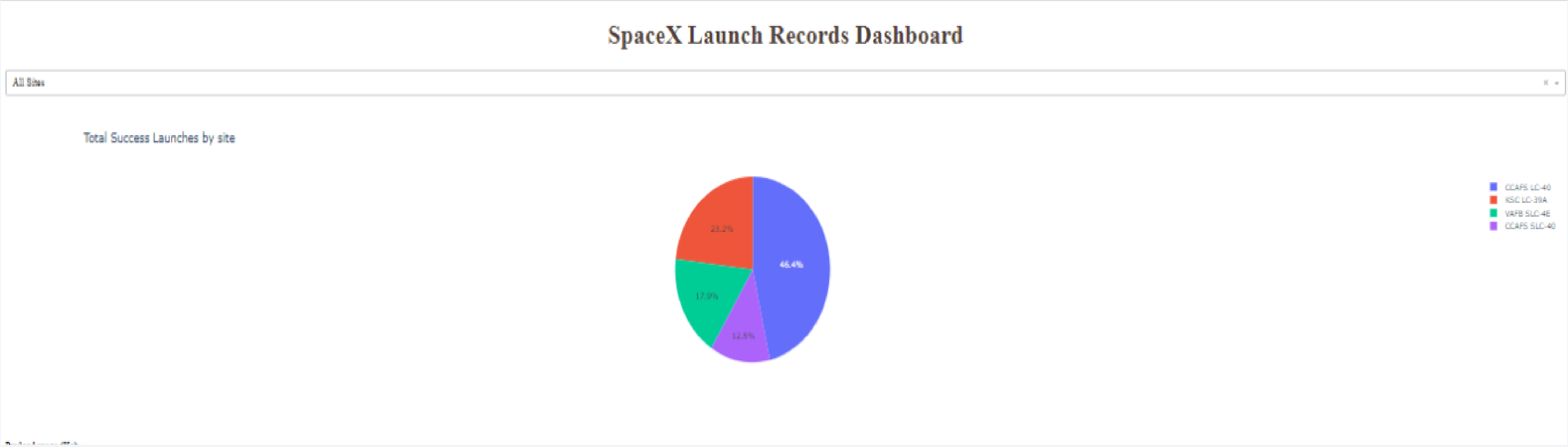
Distance from Launch site to proximities

- Launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

Section 4

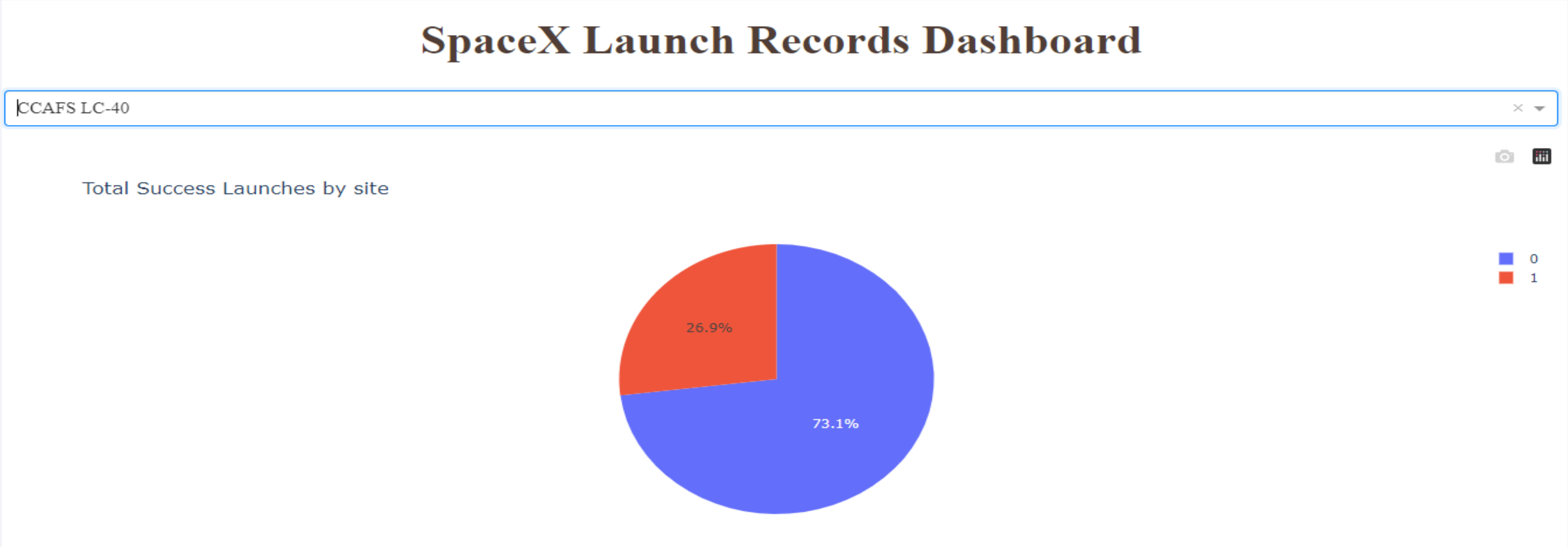
Build a Dashboard with Plotly Dash

Pie Chart for launch success counts



- Launch site CCAFS –LC 40 has the high success rate followed by other locations

Pie chart for the highest success rate launch site



- Launch site CCAFS –LC 40 has the high success rate with 73% success and 26 % failure

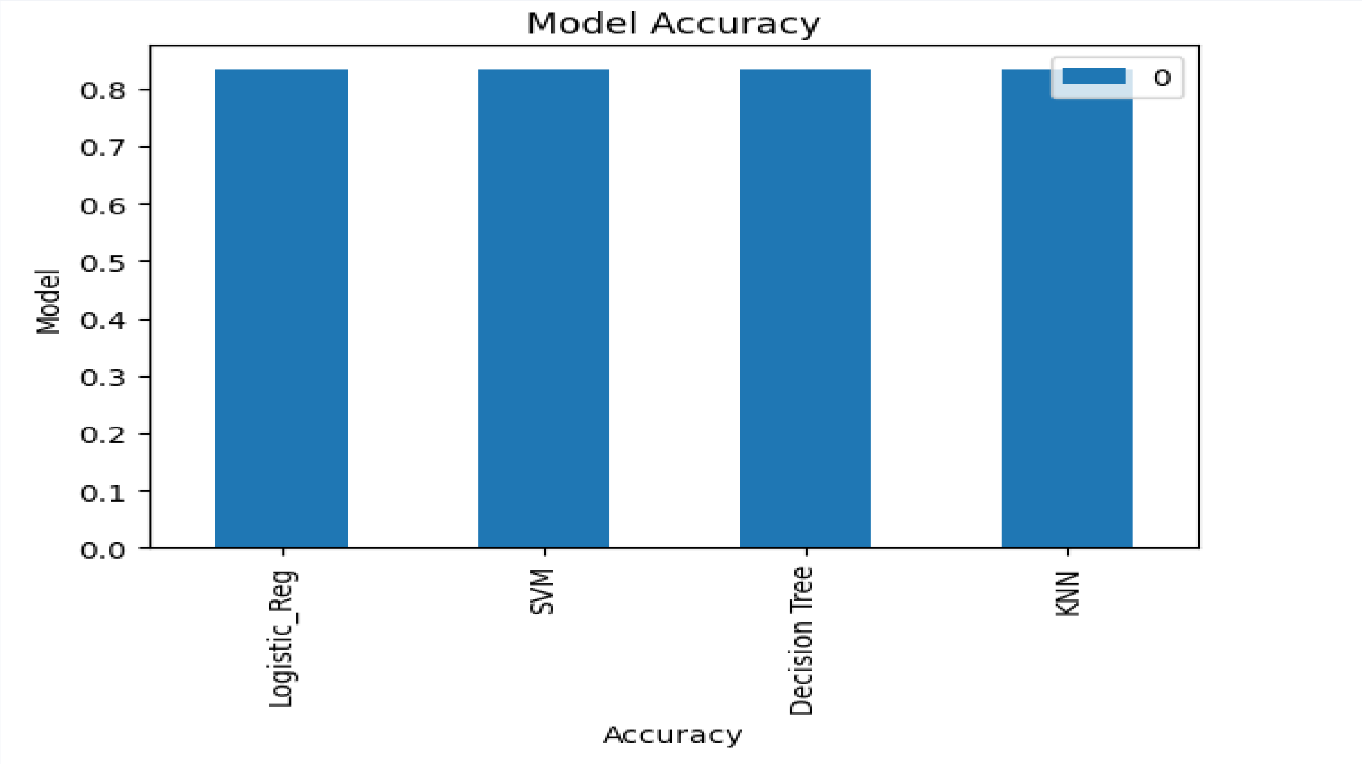
Payload vs. Launch Outcome scatter plot for all sites

Section 5

Predictive Analysis (Classification)

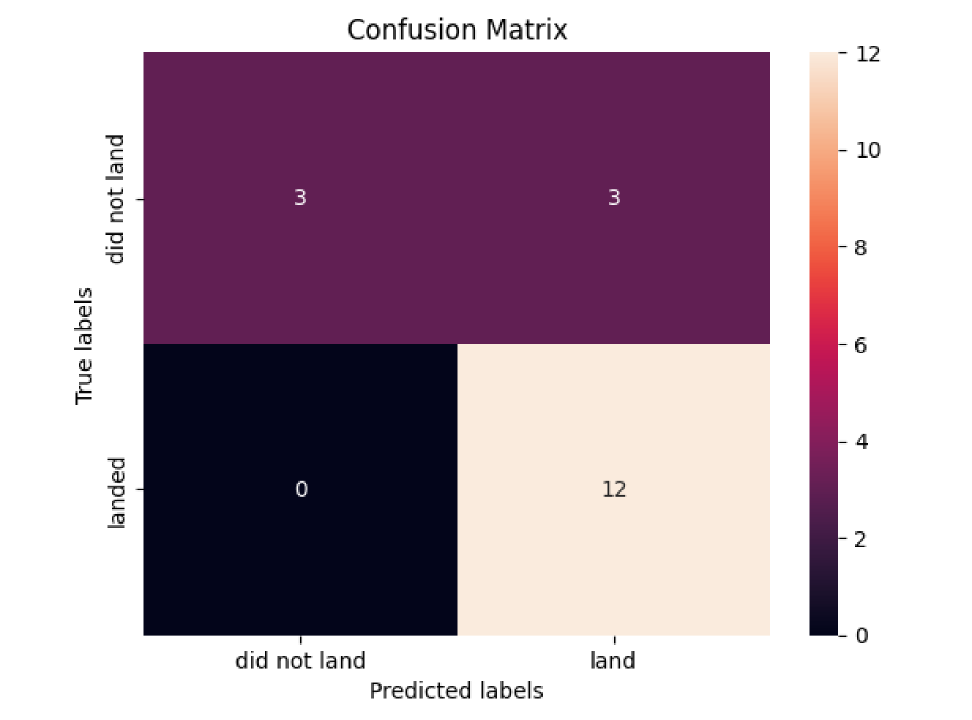
Classification Accuracy

All the models(KNN, SVM, Decision Tree and Logistic Regression) have the same accuracy : 0.833333



Confusion Matrix

- All the models(KNN, SVM, Decision Tree and Logistic Regression) have the same accuracy : 0.833333



Conclusions

- Launch site CCAFS –LC 40 has the high success rate followed by other locations
- Point 2
- Orbits ES-L1, GEO ,HEO and SSO have high success rates(100%)
- Launch Success rate kept increasing since 2013 till 2020
- All the machine learning models gave the same accuracy for classification

Appendix

- Jupiter notebooks are uploaded in the GITHUB

Thank you!