IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Anubhav Sharma
21-11-2024

Outline

Executive Summary

- Summary of methodologies
  - SpaceX Data Collection using API
  - SpaceX Data Wrangling
  - SpaceX EDS using SQL
  - SpaceX EDA using Pandas and Matplotlib
  - SpaceX Launch Analysis with Folium
  - SpaceX Landing Prediction
- Summary of all results
  - EDA Results
  - Visual Dashboard
  - Predictive Analysis

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.
  - 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.
- Problems you want to find answers
  - In this capstone, we will predict if the Falcon9 first stage will land successfully using the previous data of the Falcon9 launch available on the open internet.

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

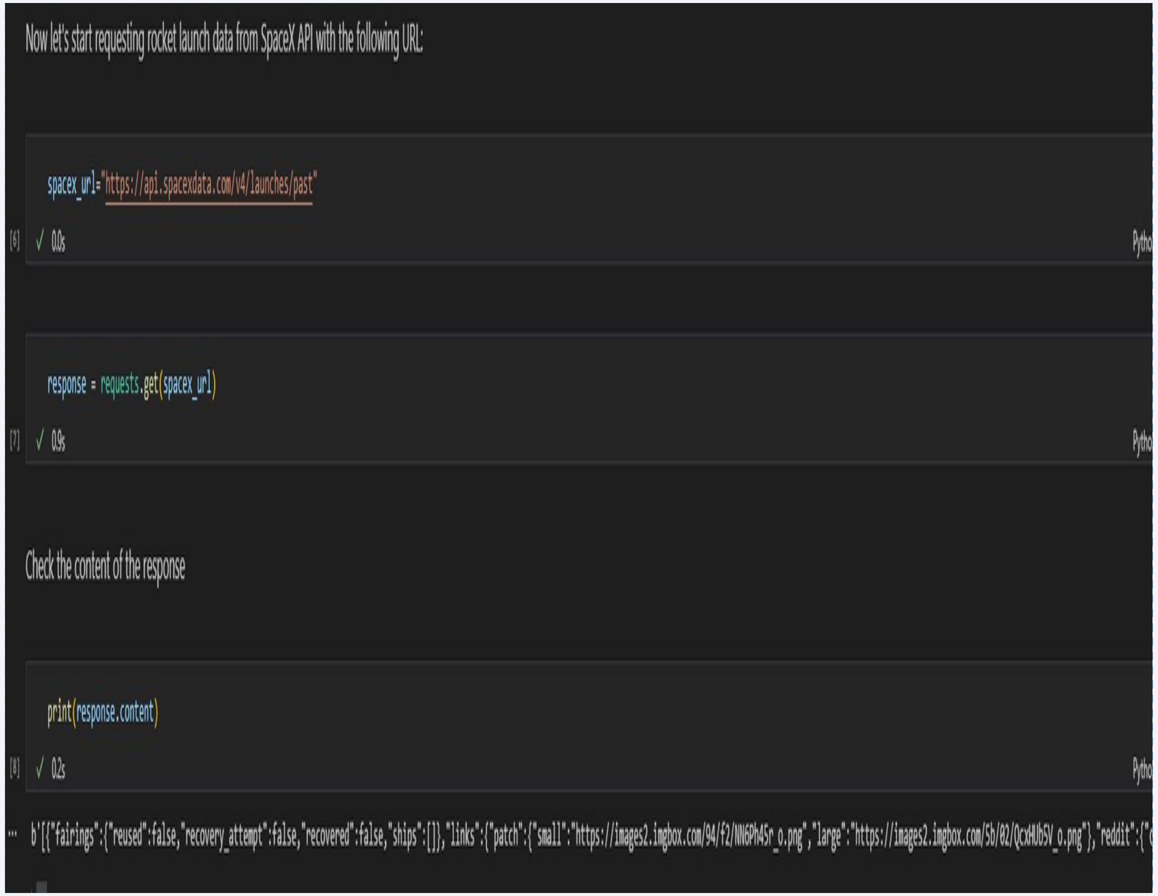
Description as to how the data was collected

The initial data collection was carried out using the SpaceX API, a RESTful API, by sending a GET request. This process involved creating a set of helper functions to facilitate the extraction of information from the API using identification numbers found in the launch data.

To ensure consistency in the requested JSON results, the SpaceX launch data was retrieved and parsed using a GET request. The response content was then decoded into a JSON format and subsequently converted into a Pandas data frame.

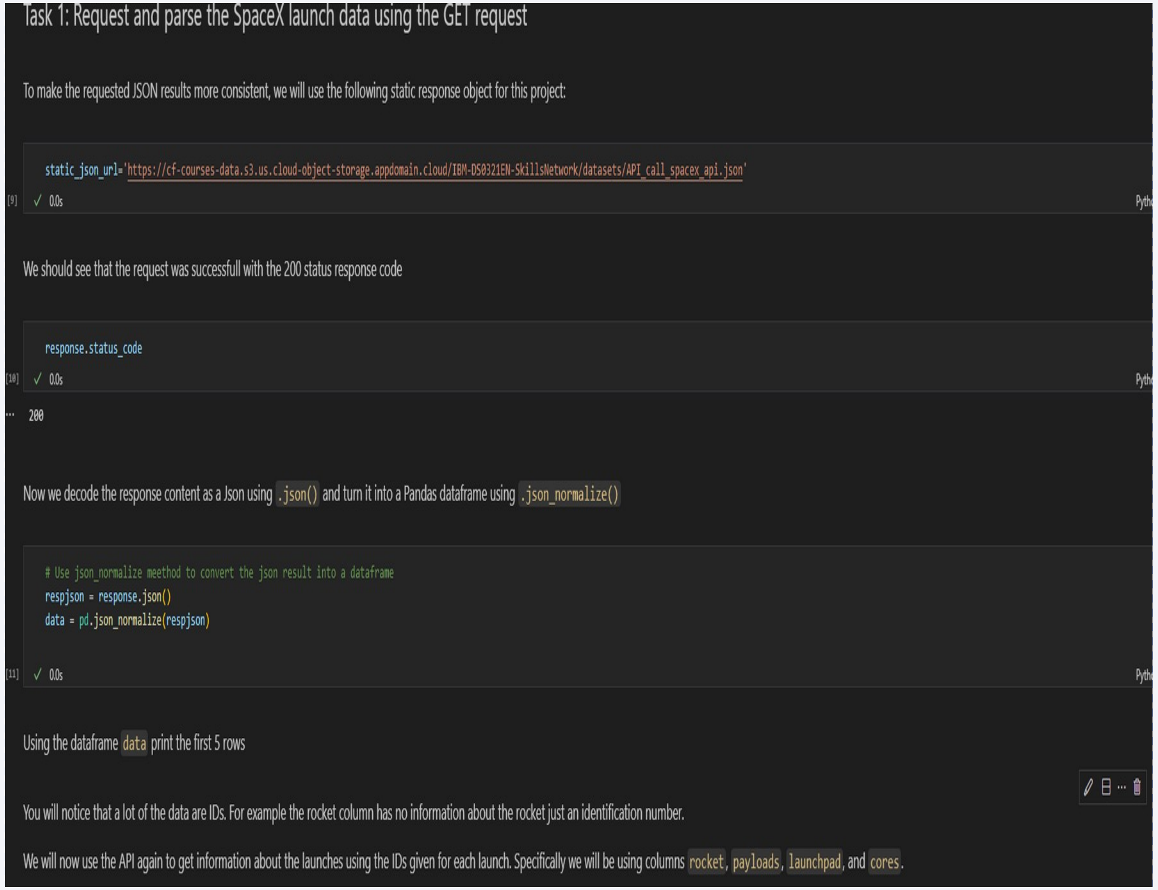
Data Collection – SpaceX API

- The data acquisition process was executed utilizing the SpaceX API, a RESTful API, by initiating a GET request to the SpaceX API endpoint. Subsequently, the SpaceX launch data was solicited and meticulously parsed through the GET request mechanism. The response content, upon retrieval, was decoded into a JSON format. This JSON result was then systematically transformed into a Pandas data frame for further analysis and processing. This method ensured the consistency and reliability of the data collected, adhering to the standards and protocols established for data handling and manipulation.



Data Collection - Scraping

- The process involved performing web scraping to gather historical launch records of the Falcon 9 rocket from a Wikipedia page. The HTML table containing the launch records was extracted from the Wikipedia page, and the data was then parsed and converted into a Pandas data frame for further analysis.



Data Wrangling

After acquiring and constructing a Pandas DataFrame from the gathered data, the data was filtered by the BoosterVersion column to retain only the Falcon 9 launches. Missing values in the LandingPad and PayloadMass columns were addressed, with the PayloadMass missing values being replaced by the column's mean value. Additionally, Exploratory Data Analysis (EDA) was conducted to identify patterns in the data and to determine the appropriate labels for training supervised models.

Link to Notebook

EDA with Data Visualization

Conducted data analysis and feature engineering utilizing Pandas and Matplotlib.

- Engaged in EDA
- Prepared data and performed data engineering

Employed scatter plot between

- Flight number and launch site
- Payload and Launch Site
- Payload and Orbit type

Link to Notebook

EDA with SQL

The following SQL queries were performed for EDA

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

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

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

```
%sql SELECT * FROM 'SPACEXTBL' WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Version
```

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

Hint:Use min function

```
%sql SELECT MIN(DATE) FROM 'SPACEXTBL' WHERE "Landing _Outcome" = "Success (ground pad)";
```

Build an Interactive Map with Folium

A folium map was created to mark all the launch sites. Various map objects, such as markers, circles, and lines, were added to indicate the success or failure of launches at each site. This visual representation provided a clear and interactive way to analyze the launch outcomes for each location.

Created launch codes i.e 0 and 1

[Link to Notebook]

Build a Dashboard with Plotly Dash

An interactive dashboard application was developed using Plotly Dash, incorporating the following features:

Launch Site Drop-down Input Component: Added to allow users to select a specific launch site.

Range Slider for Payload Selection: Included to enable users to select a range of payload values.

Callback Function for Success-Payload Scatter Chart: Added to render a scatter plot that visualizes the relationship between payload and launch success based on the selected payload range.

Link to Python Code

Predictive Analysis (Classification)

Here's a summary of how I built, evaluated, improved, and identified the best-performing classification model:

- Data Loading and Preparation
  - Loaded the data into Pandas DataFrame
  - Performed EDA to understand the data
  - Standardized the data for easier calculations
- Data Splitting
  - Split the data into training and testing sets using sklearn with test_size = 0.2 and random_state = 2.

Predictive Analysis (Classification) Cont.

To identify the most effective machine learning model or method for the test data among SVM, Classification Trees, k-Nearest Neighbors, and Logistic Regression, the following steps were undertaken:

- Model Initialization
  - Created an object for different algorithms (SVM, Decision Trees, kNN, Logistic Regression).
  - Constructed a GridSearchCV for each of the aforementioned algorithms.
- Hyperparameter Tuning
  - For each model, cv was placed at 10 (10-fold cross-validation).
  - The training data was fitted into the GridSearchCV
- Evaluation
  - After fitting the model, the GridSearchCV object was tested.
  - Best parameters were displayed using <object_name>.best_params_ attribute.

Predictive Analysis (Classification) Cont.

Testing and Visualization

- The score method was used to calculate the accuracy of each model
- Confusion Matrix for each was plotted

Report of every single algorithm used

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

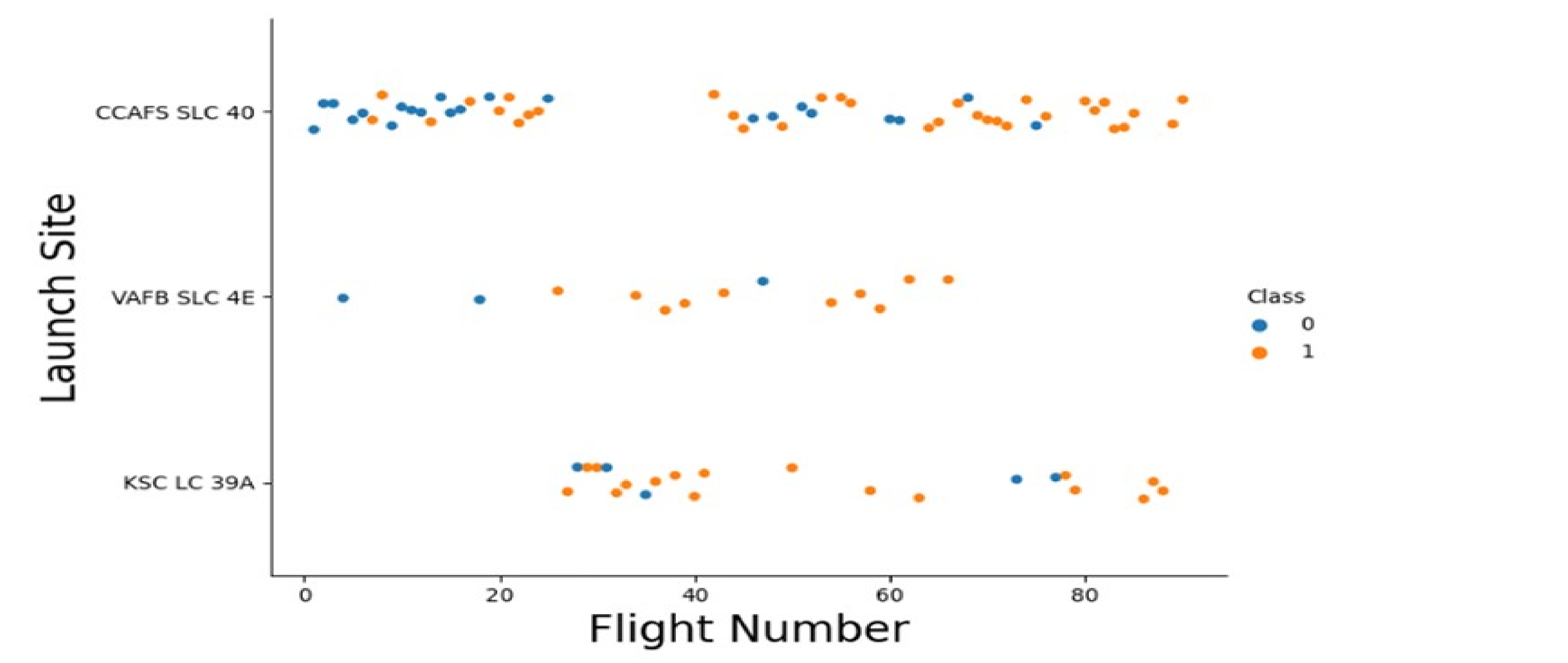
Results

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

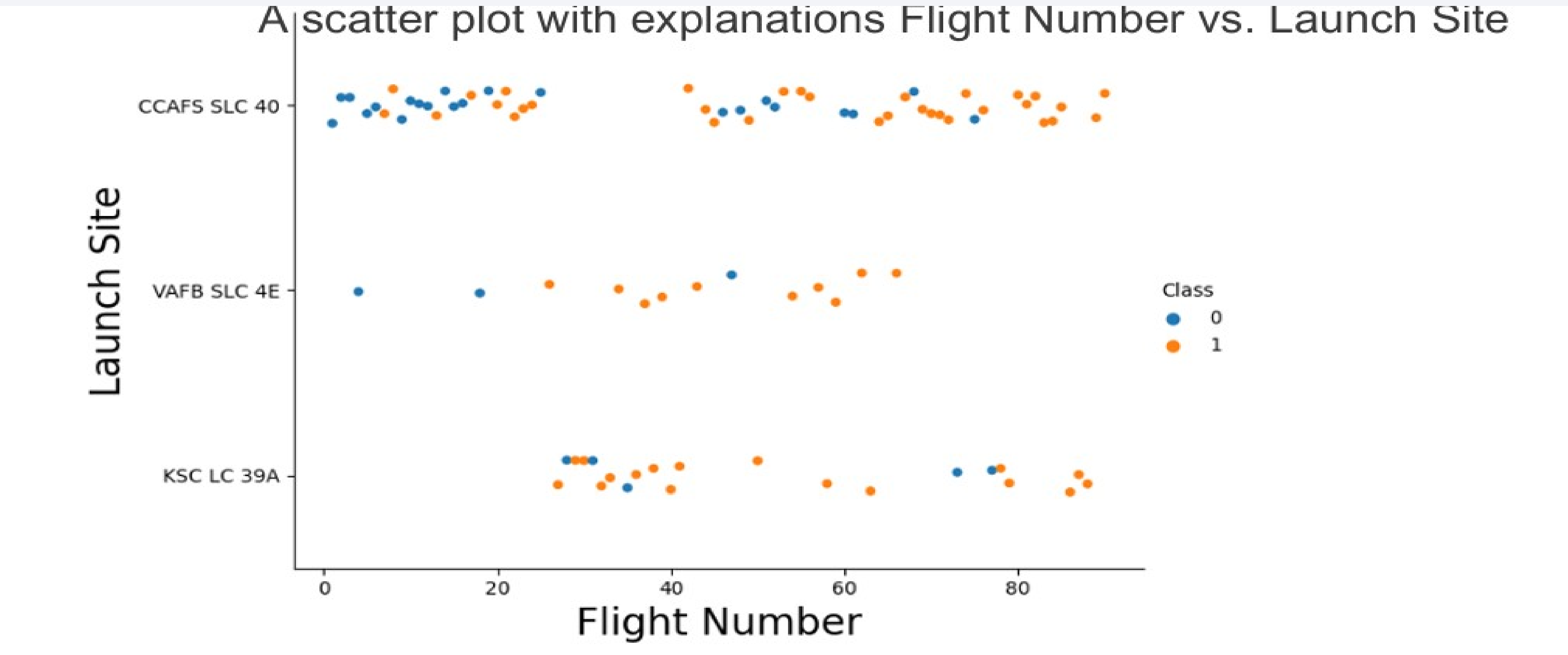
Section 2

Insights drawn from EDA

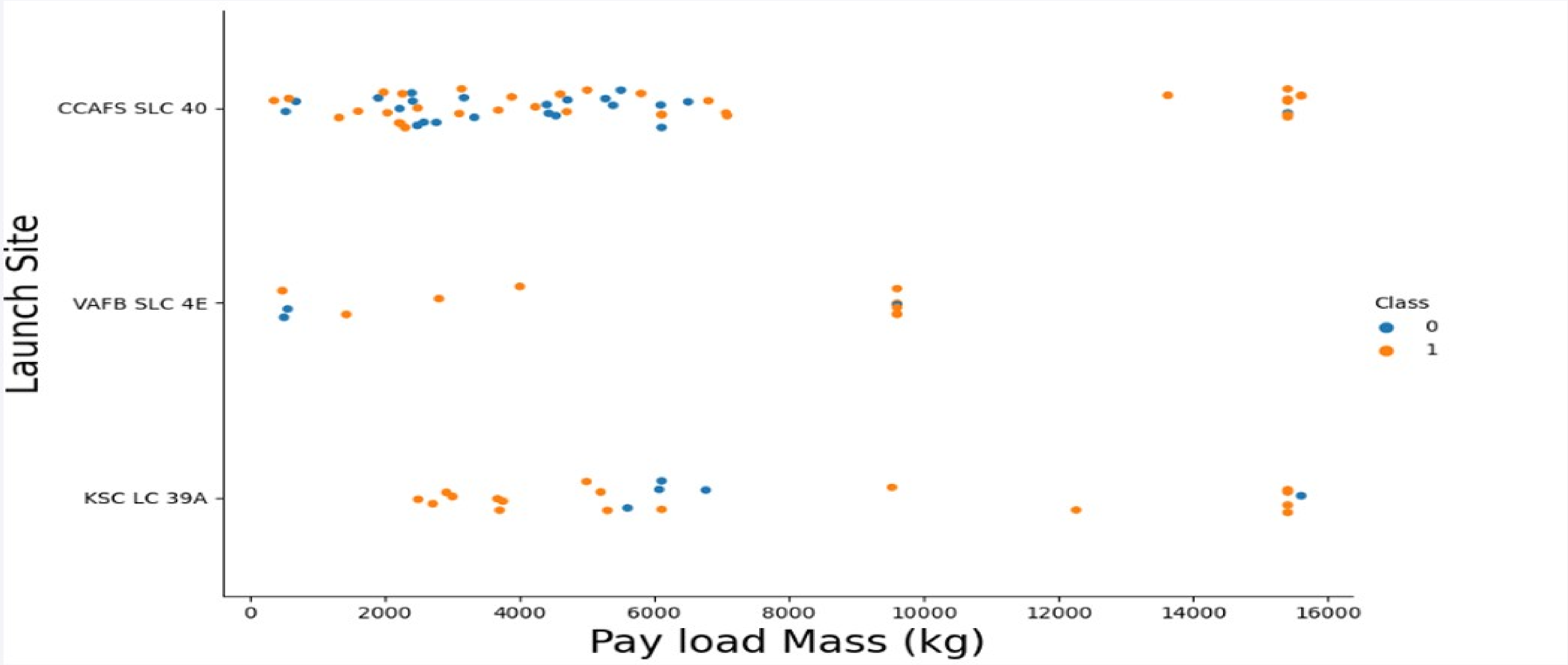
Flight Number vs. Launch Site



Payload vs. Launch Site



Success Rate vs. Orbit Type



All Launch Site Names

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

In [31]:
```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

Out[31]:

Launch_Sites
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'

In [72]:
```
%sql SELECT * FROM 'SPACEXTBL' WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

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

Out[72]:

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

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

In [17]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA (CRS)';
```

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

Out[17]:

Total Payload Mass(Kgs)	Customer
45596	NASA (CRS)

Average Payload Mass by F9 v1.1

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

In [19]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Version 
```

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

Out[19]:

Payload Mass Kgs	Customer	Booster_Version
2534.6666666666665	MDA	F9 v1.1 B1003

First Successful Ground Landing Date

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

Hint:Use min function

```
%sql SELECT MIN(DATE) FROM 'SPACEXTBL' WHERE "Landing _Outcome" = "Success (ground pad)";
```

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

MIN(DATE)
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 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_I
```

```
 * 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

Total Number of Successful and Failure Mission Outcomes

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

Boosters Carried Maximum Payload

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

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

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

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

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 substr(Date,7,4), substr(Date, 4, 2),"Booster_Version", "Launch_Site", Payload, "PAYLOAD_MASS__KG_", "Mission_Ou
```

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

substr(Date,7,4)	substr(Date, 4, 2)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Mission_Outcome	Landing _Outcome
2015	01	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	Success	Failure (drone ship)
2015	04	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	Success	Failure (drone ship)

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

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') ORDE
```

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

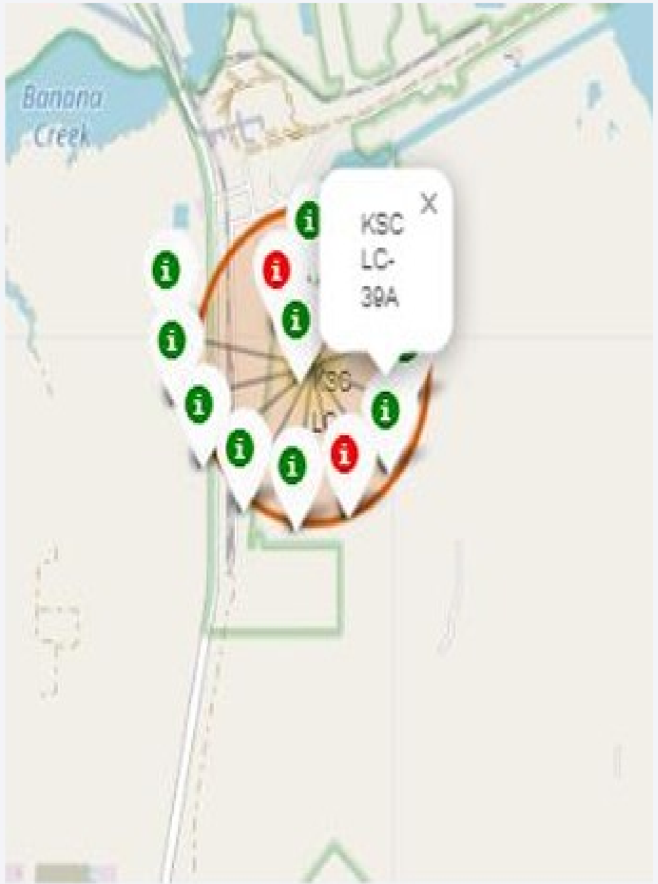
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
19-02-2017	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
18-10-2020	12:25:57	F9 B5 B1051.6	KSC LC-39A	Starlink 13 v1.0, Starlink 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)
17-12-2019	00:10:00	F9 B5 B1056.3	CCAFS SLC-40	JCSat-18 / Kacific 1, Starlink 2 v1.0	6956	GTO	Sky Perfect JSAT, Kacific 1	Success	Success
16-11-2020	00:27:00	F9 B5B1061.1	KSC LC-39A	Crew-1, Sentinel-6 Michael Freilich	12500	LEO (ISS)	NASA (CCP)	Success	Success
15-12-2017	15:36:00	F9 FT B1035.2	CCAFS SLC-40	SpaceX CRS-13	2205	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
15-11-2018	20:46:00	F9 B5 B1047.2	KSC LC-39A	Es hail 2	5300	GTO	Es hailSat	Success	Success
14-[illegible]	[illegible]	[illegible]	[illegible]	SpaceX CRS-[illegible]	[illegible]	LEO [illegible]	[illegible]	[illegible]	Success [illegible]

Section 3

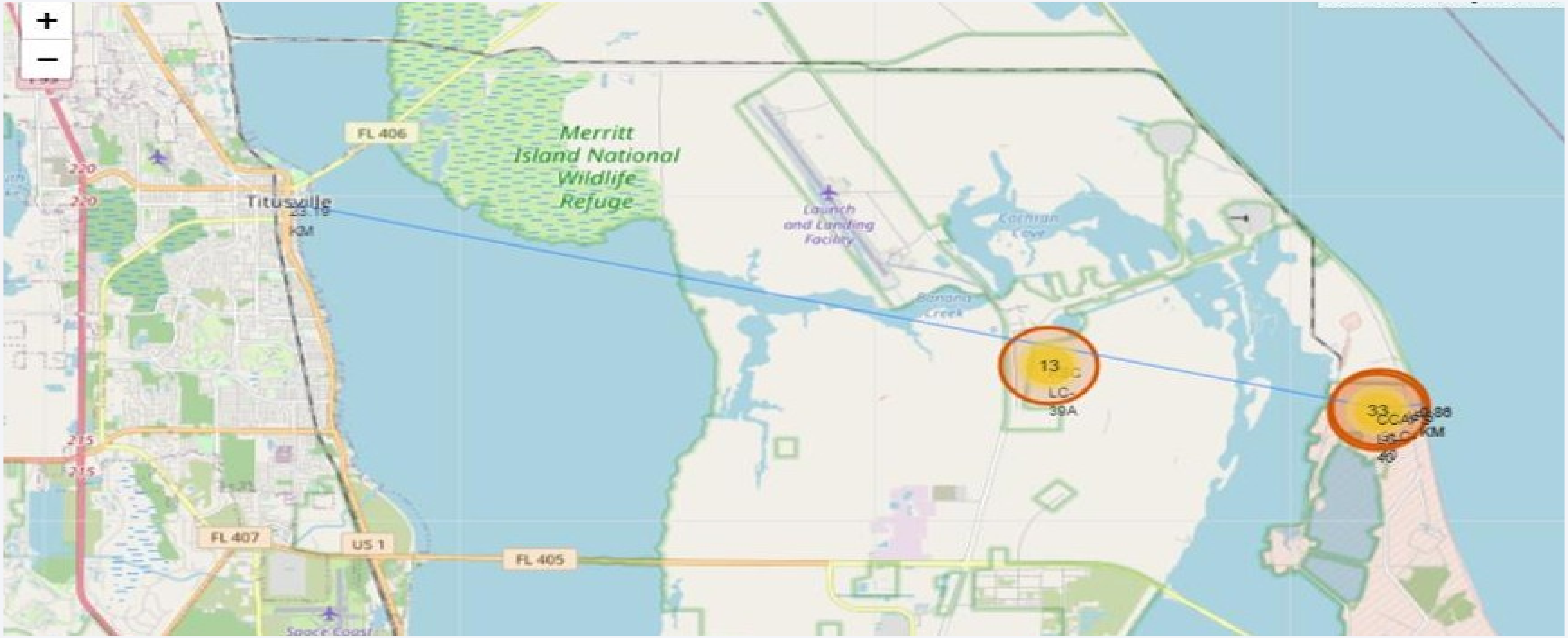
Launch Sites Proximities Analysis

Markers of all launch sites on global map

Launch outcome for each site



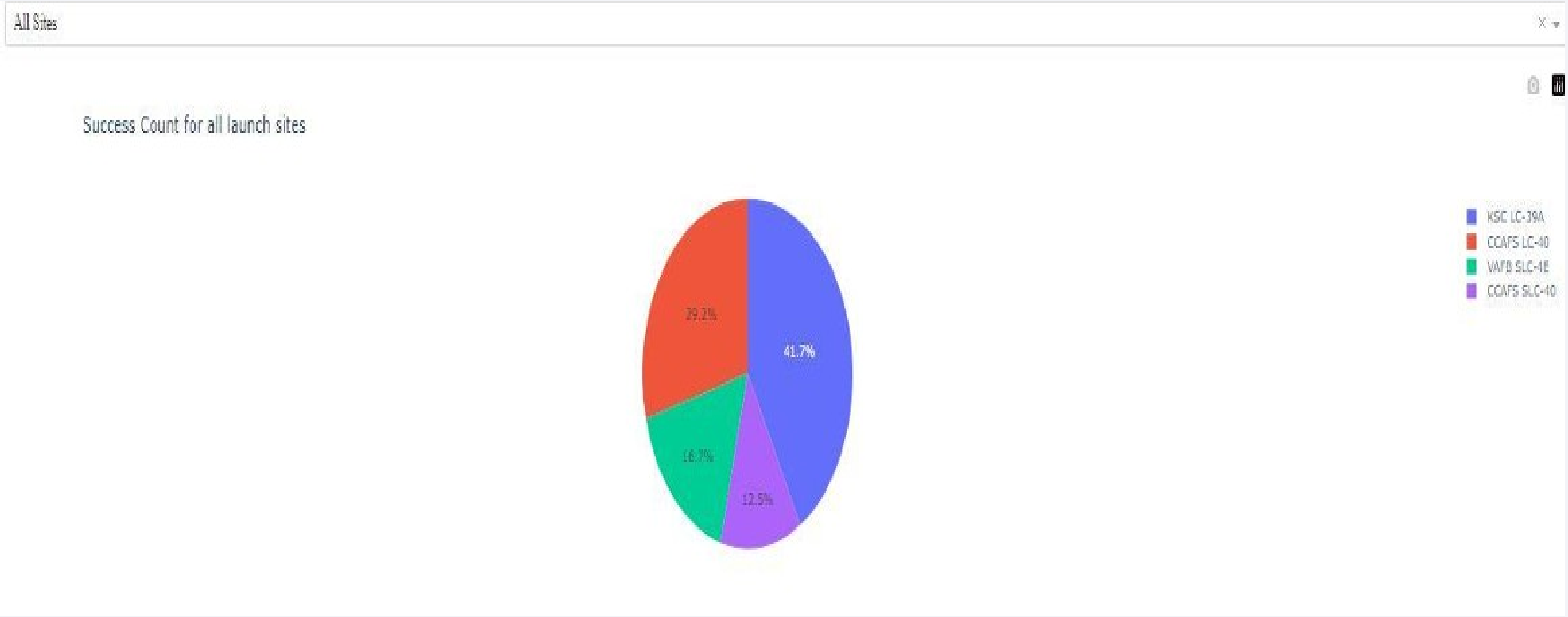
Distances from a Launch Site to Nearby Locations



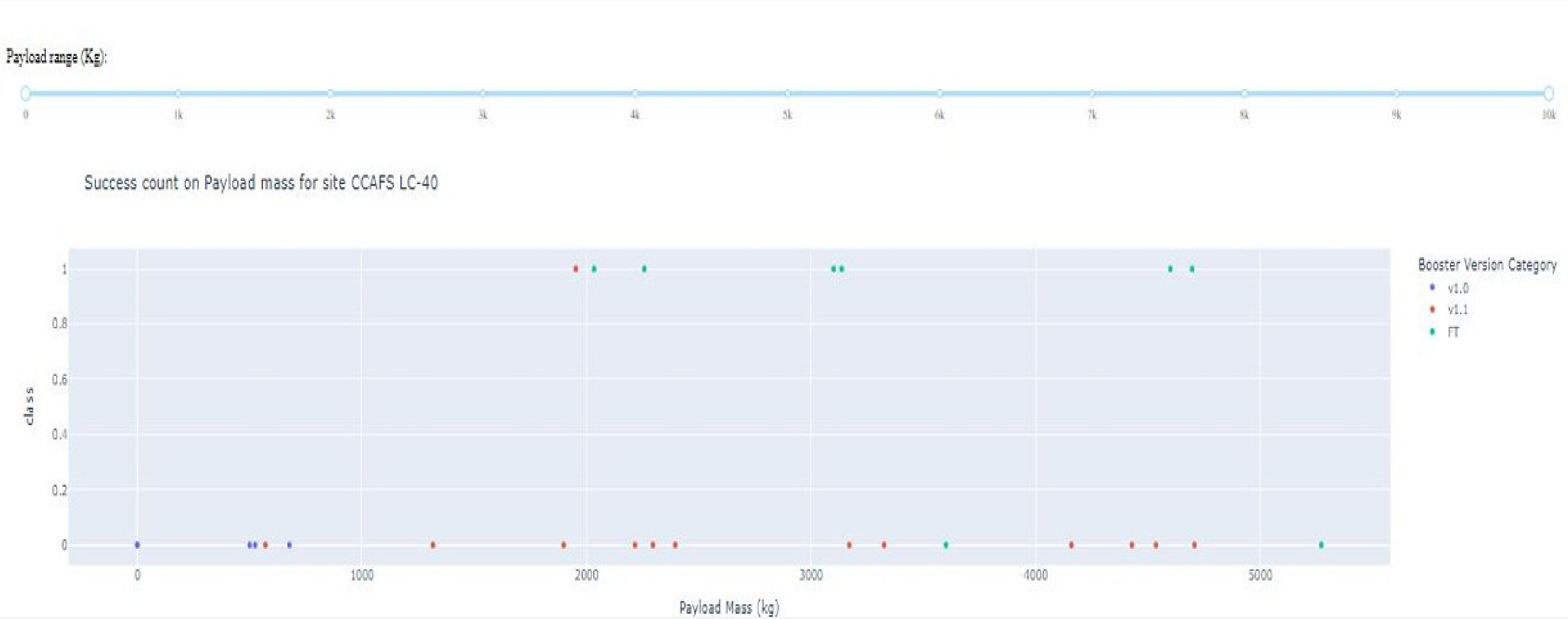
Section 4

Build a Dashboard with Plotly Dash

Pie-Chart for launch success



Scatter Plot of Payload vs. Launch Outcome for All Sites



Section 5

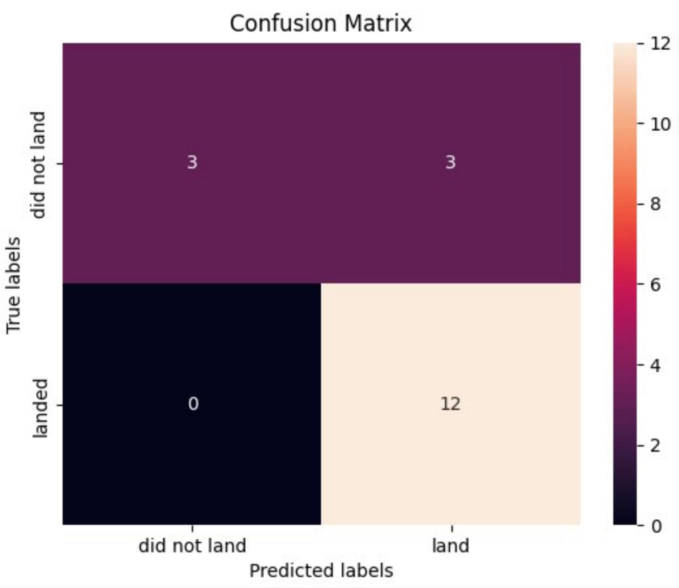
Predictive Analysis (Classification)

Classification Accuracy

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

Confusion Matrix

- All four classification models produced identical confusion matrices and demonstrated equal capability in distinguishing between the different classes. However, a significant issue across all models was the occurrence of false positives.



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%.
- We can deduce that, as the flight number increases in each of the 3 launcg 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
- If you observe Payload 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).
- 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.

Appendix

Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!