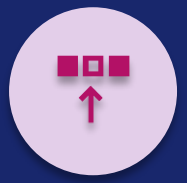
IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Neel Bhadkamkar

25-12-2021

OUTLINE



Executive Summary

Introduction

Methodology

Results

Conclusion

Appendix

EXECUTIVE SUMMARY

Summary of methodologies

- Data Collection
- Data Wrangling
- EDA with Data Visualization
- EDA with SQL
- Building an interactive map with Folium
- Building a Dashboard with Plotly Dash
- Predictive analysis (Classification)

Summary of all results

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

INTRODUCTION

- We will predict if the Falcon 9 first stage will land successfully. 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 that need to be solved-
  - Factors which influence the rockets successful landing.
  - The effect of certain rocket variables have on the rocket's successful landing rate.
  - Conditions to be met to achieve the best results and to ensure the best successful landing rate.

METHODOLOGY

METHODOLOGY

Executive Summary

Data collection methodology:

- SpaceX REST API
- Web Scraping from Wikipedia

Performed data wrangling

- One Hot Encoding was done on landing outcome column.

Performed exploratory data analysis (EDA) using visualization and SQL

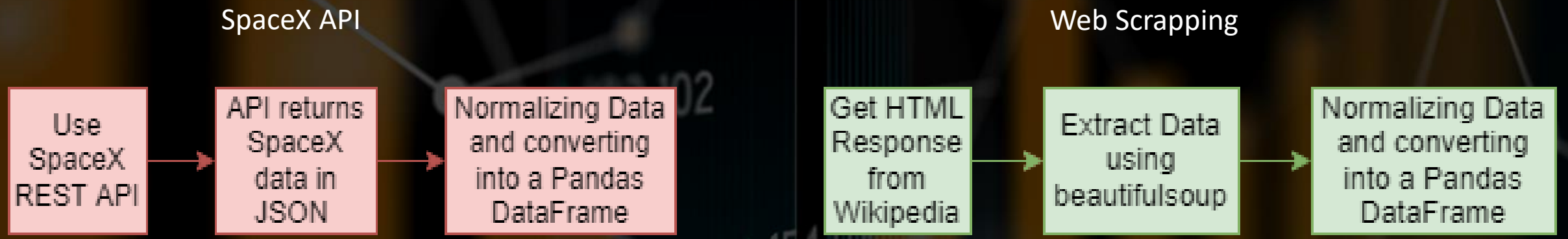
Performed interactive visual analytics using Folium and Plotly Dash

Performed predictive analysis using classification models

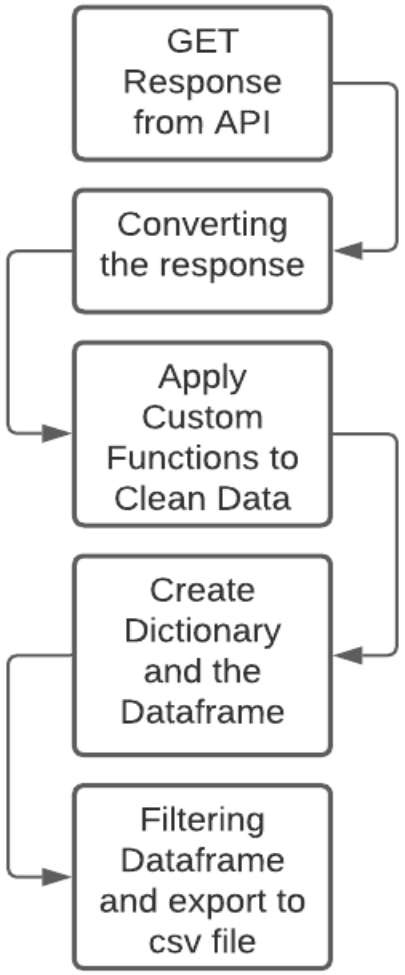
- How to build, tune, evaluate classification models

DATA COLLECTION

- The Datasets were obtained by-
  - SpaceX Launch Data was gathered from the SpaceX REST API.
  - The API provided launch data including the rocket used, payload delivered, launch specifications and landing outcomes.
  - The SpaceX Rest API endpoints all start with https://api.spacexdata.com/v4/ .
  - Another data source for obtaining Falcon 9 data is web scraping the Wikipedia Page Titled 'List of Falcon 9 and Falcon Heavy Launches' using BeautifulSoup.



DATA COLLECTION – SPACEX API




```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

```
data = pd.json_normalize(response.json())
```

```
# Call getBoosterVersion
getBoosterVersion(data)

# Call getLaunchSite
getLaunchSite(data)

# Call getPayloadData
getPayloadData(data)

# Call getCoreData
getCoreData(data)
```

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

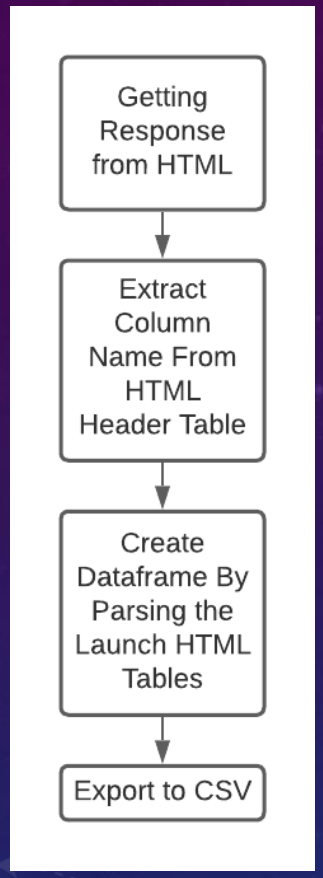
```
# Create a data from launch_dict
launch_df=pd.DataFrame(launch_dict)
```

```
data_falcon9=launch_df.loc[launch_df['BoosterVersion']!='Falcon 1']
```

```
data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```

GitHub Notebook URL

DATA COLLECTION - SCRAPING




```
x = requests.get(static_url)
```

```
html_tables = soup.find_all('table')
```

```
column_names = []

th_temp = soup.find_all('th')
for x in range(len(th_temp)):
    try:
     name = extract_column_from_header(th_temp[x])
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
     pass
```

```
column_names = []

th_temp = soup.find_all('th')
for x in range(len(th_temp)):
    try:
     name = extract_column_from_header(th_temp[x])
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
     pass
```

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
```

```
df=pd.DataFrame(launch_dict)
```

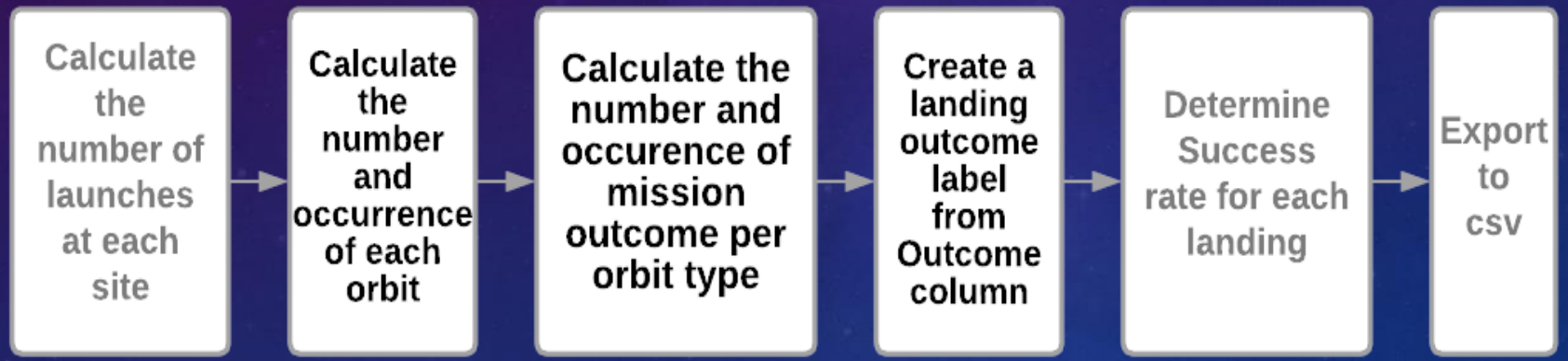
```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Notebook Link

DATA WRANGLING

Introduction

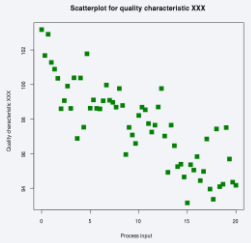
In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, `True Ocean` means the mission outcome was successfully landed to a specific region of the ocean while `False Ocean` means the mission outcome was unsuccessfully landed to a specific region of the ocean. `True RTLS` means the mission outcome was successfully landed to a ground pad `False RTLS` means the mission outcome was unsuccessfully landed to a ground pad.`True ASDS` means the mission outcome was successfully landed on a drone ship `False ASDS` means the mission outcome was unsuccessfully landed on a drone ship. We will mainly convert those outcomes into Training Labels with `1` means the booster successfully landed `0` means it was unsuccessful.



Notebook Link

EDA WITH DATA VISUALIZATION

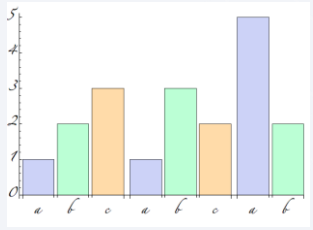
Scatter Point Plot:



- Flight V/S Payload Mass
- Payload V/S Launch Site
- Orbit V/S Flight Number
- Payload V/S Orbit Type
- Orbit V/S Payload Mass

Scatter Plots show how much a variable is correlated to another.

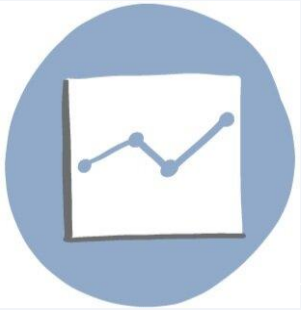
Bar Graph



- Mean V/S Orbit

A bar chart or bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent.

Line Graph



- Success Rate V/S Year

Line graphs (or line charts) are best when you want to show **how the value of something changes over time,** or compare how several things change over time relative to each other.

Notebook URL

EDA WITH SQL

Used SQL queries to gather information from the dataset.

Examples of queries performed on the dataset-

- 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 acheived.
- 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 failed landing_outcomes in drone ship, their booster versions, and launch site names for 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 Notebook URL

BUILD AN INTERACTIVE MAP WITH FOLIUM

We marked and labelled each launch site with CircleMarker using the latitude and longitude coordinates.

We marked the success and failure for each site using Green and Red Markers for success and failure respectively. This was done using MarkerCluster().

Calculation of the Distance from Launch Site to various landmarks was done to gain insight on the Launch Site location.

With them we were able to answer the following questions -

- 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

Notebook URL

BUILD A DASHBOARD WITH PLOTLY DASH

An interactive Dashboard was created using the Dash and Flask Web Framework. It's was also deployed on heroku so that it can be accessed 24/7.

Dropdown List, slider, functions to display pie chart and scatter plot were added to the Dashboard.

Pie Chart-

Pie Chart shows the total launches for all sites.

It represents data visually as a fractional part of a whole, which can be an effective communication tool for the even uninformed audience.

It enables the audience to see a data comparison at a glance to make an immediate analysis or to understand information quickly.

Scatter Plot-

It is used to see the relation between two variables.

Using it we can deduce the relation between the variables be it – strong or weak, linear or non-linear, positive or negative.

In this dashboard we have plotted to show the relation between payload mass (kg) and Outcome for the different Booster Versions.

Website Link (Heroku)

Github Notebook Link

PREDICTIVE ANALYSIS (CLASSIFICATION)

Building the Model

- Loading the dataset and creating a pandas dataframe.
- Transform the data using numpy library.
- Split the dataset into training and testing data.
- Select machine learning algorithms we want to use.
- Select the algorithm and parameters and pass it to the GridSearchCV function.
- Fit the model to the training dataset.

Evaluate the Model

- Check accuracy for each model
- Plot Confusion Matrix

Find The Best Performing Classification Model

- The best performing model is the one with the highest accuracy score.

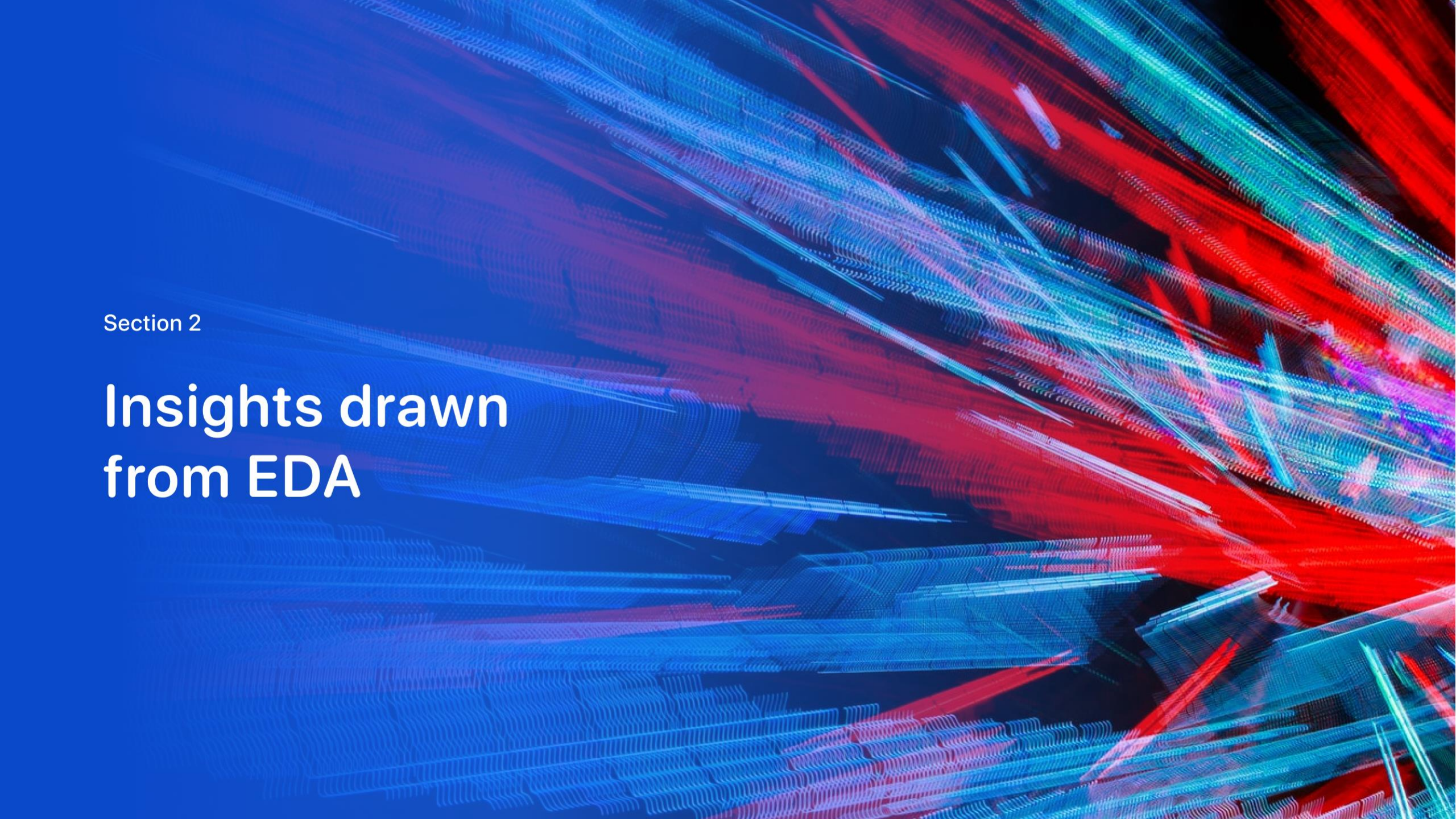
Github Notebook Link

RESULTS

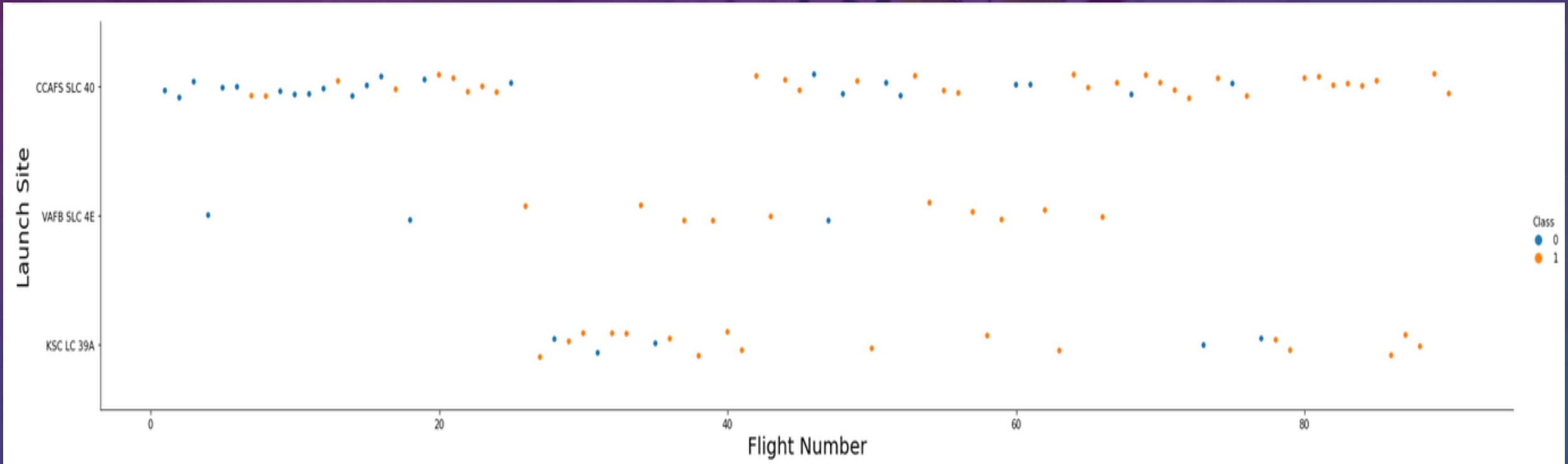
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

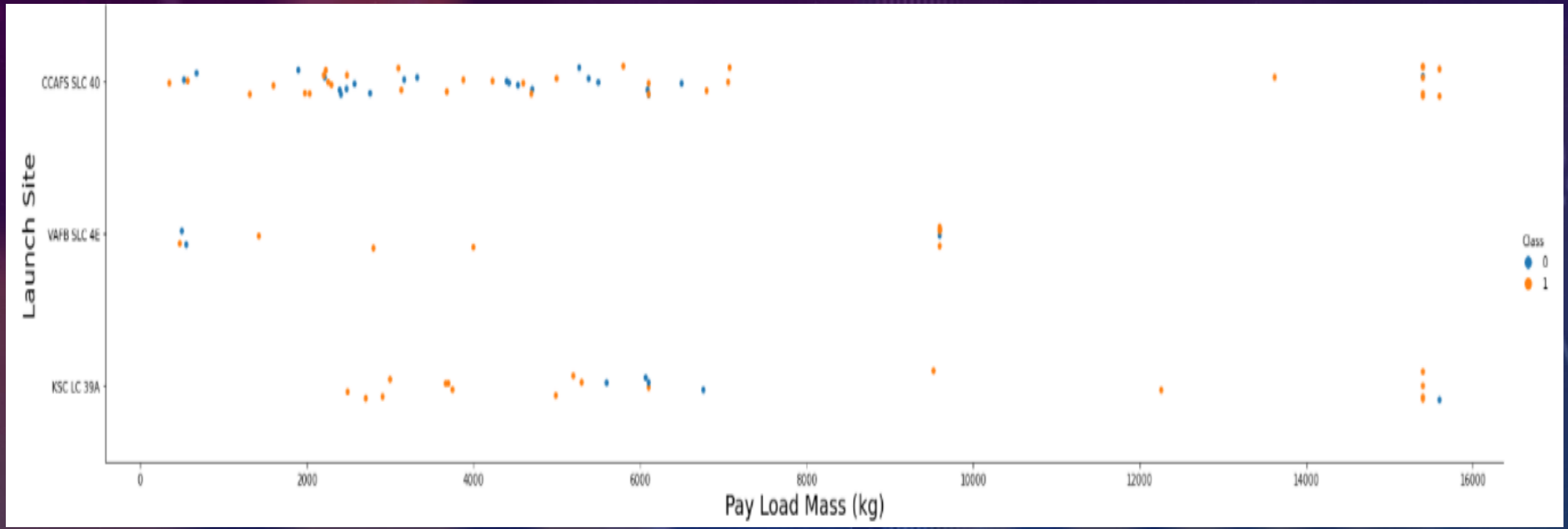


FLIGHT NUMBER VS. LAUNCH SITE



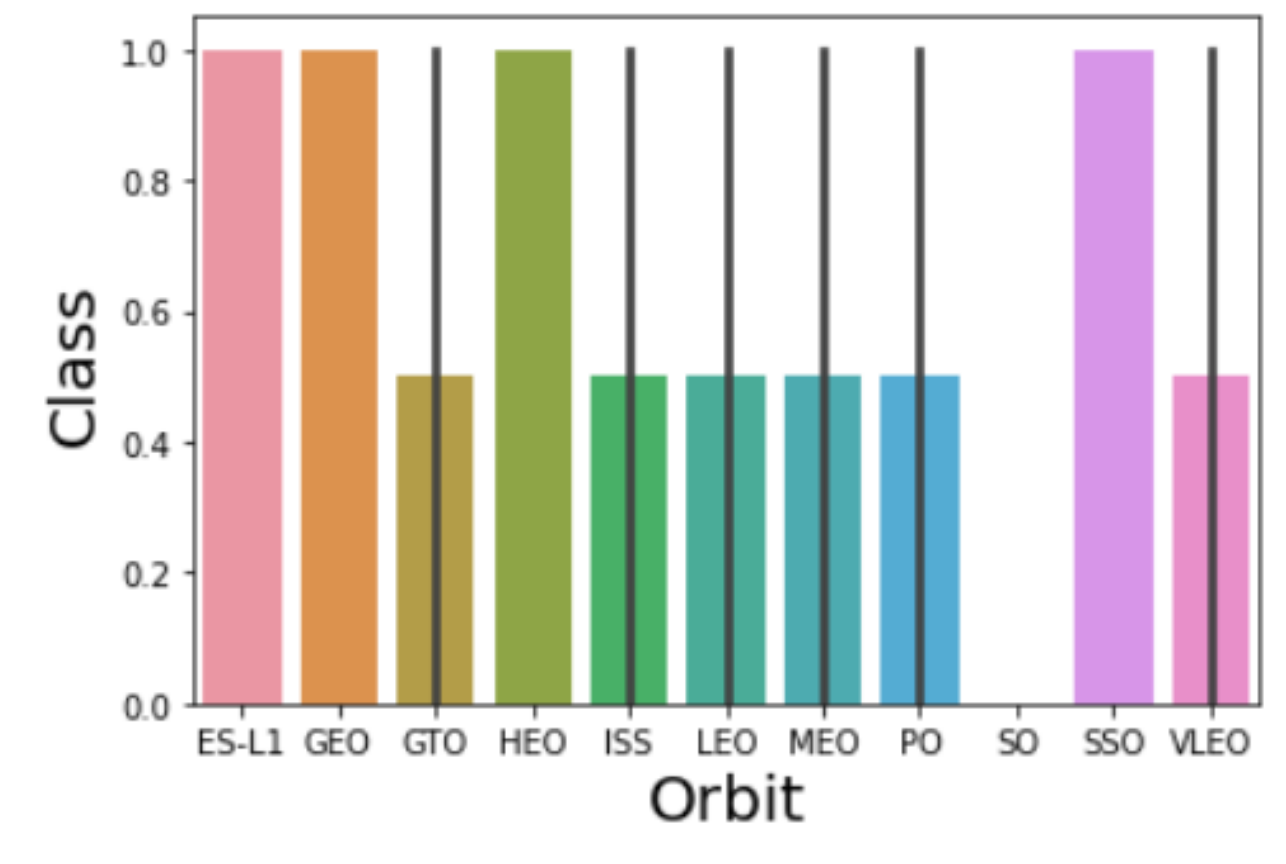
Graph suggests an increase in success rate over time. Most likely a big breakthrough occurring around flight number 20 which significantly increased success rate. CCAFS seems to be the primary launch site as it has the highest number of launches.

PAYLOAD VS. LAUNCH SITE



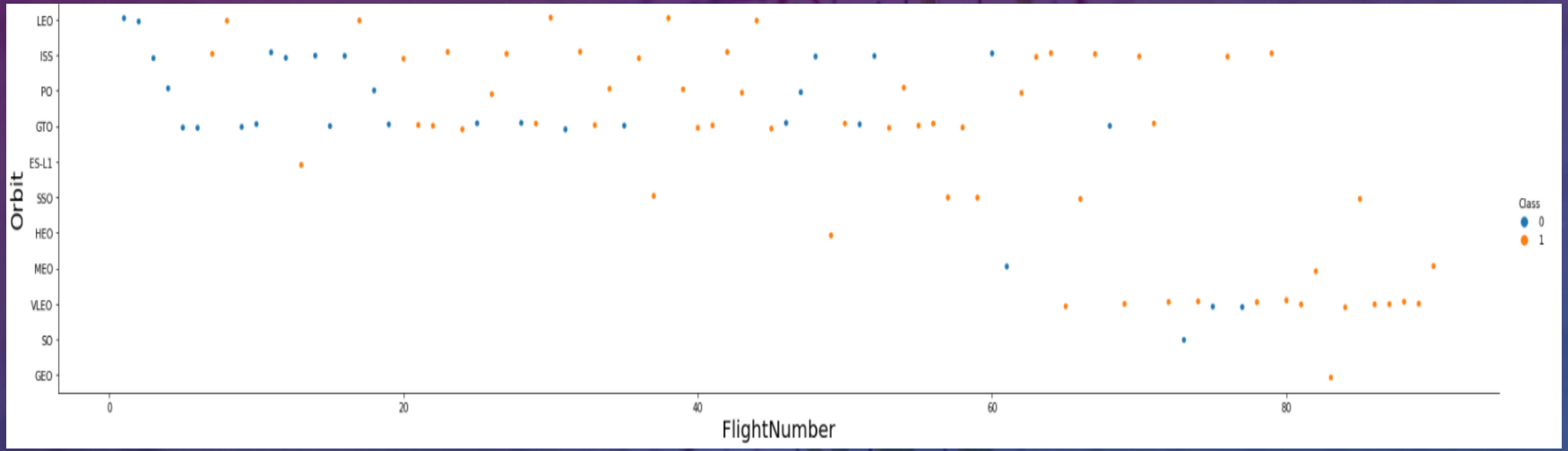
Payload mass appears to fall mostly between 0-6000 kg.
For Launch Site CCAFS SLC 40 the higher payload mass launches have highest success rate.

SUCCESS RATE VS. ORBIT TYPE



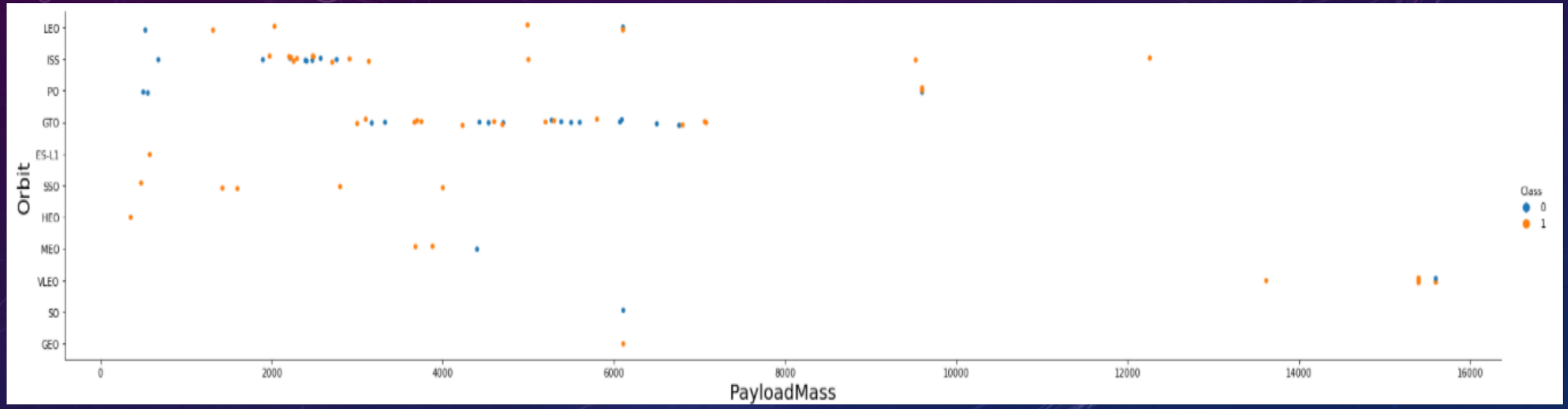
ES-L1. GEO, HEO, SSO have the highest success rate of 100%.
SO has the success rate of 0%.

FLIGHT NUMBER VS. ORBIT TYPE



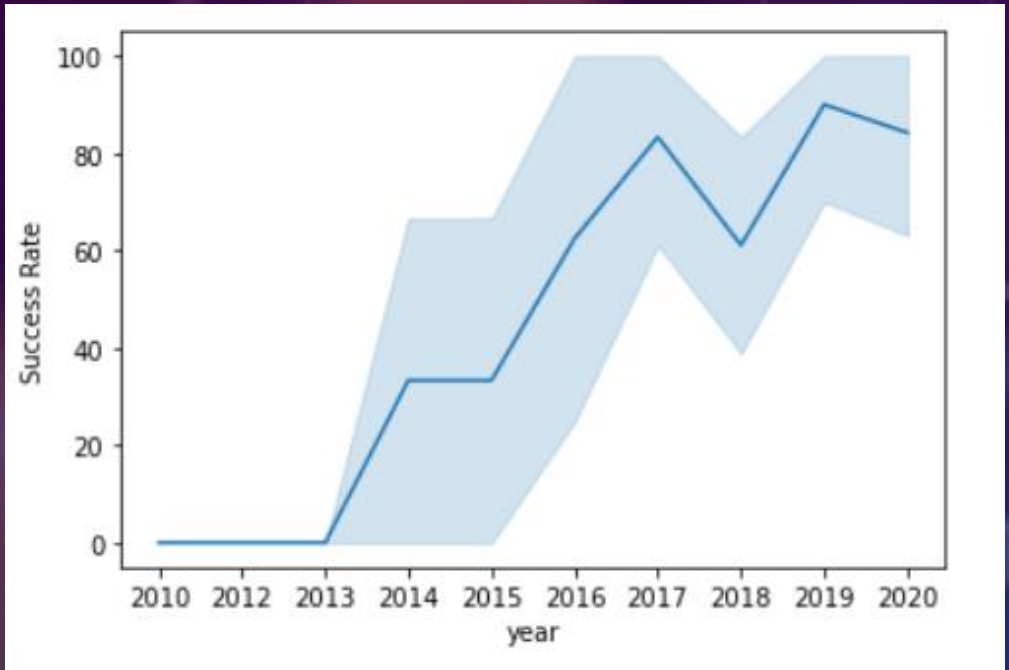
We can see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

PAYLOAD VS. ORBIT TYPE



Here we observe that Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.

LAUNCH SUCCESS YEARLY TREND



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

ALL LAUNCH SITE NAMES

```
%sql SELECT Distinct LAUNCH_SITE FROM SPACEXTBL
```

```
 * ibm_db_sa://bwj90831:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

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

Distinct Keyword is used to show all unique values in Launch_Site Column.

LAUNCH SITE NAMES BEGIN WITH 'CCA'

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

```
 * ibm_db_sa://bwj90831:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

DATE	time_utc_	booster_version	launch_site	payload	payload_mass_kg_	orbit	customer	mission_outcome	landing_outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
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 (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Limit 5 shows the first 5 results. LIKE 'CCA%' is used to get the launch site names starting with CCA

TOTAL PAYLOAD MASS

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER='NASA (CRS)'
```

```
 * ibm_db_sa://bwj90831:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

1
45596

Sum is used to the total payload mass. WHERE clause is used to so that the query only performs calculations when the customer is NASA (CRS)

AVERAGE PAYLOAD MASS BY F9 V1.1

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION='F9 v1.1'
```

```
 * ibm_db_sa://bwj90831:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

1
2928

Here AVG function is used to get the average of the column PAYLOAD_MASS_KG

WHERE clause is used to only perform calculations on the booster version F9 v1.1 entries.

FIRST SUCCESSFUL GROUND LANDING DATE

```
%sql SELECT min(DATE) FROM SPACEXTBL WHERE LANDING__OUTCOME='Success (ground pad)'
```

```
 * ibm_db_sa://bwj90831:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

1
2015-12-22

MIN function fetches the minimum date in the column Date

WHERE clause is used to only include successful landing outcomes (Ground Landing)

SUCCESSFUL DRONE SHIP LANDING WITH PAYLOAD BETWEEN 4000 AND 6000

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 AND LANDING__OUTCOME='Success (drone ship)'
```

```
 * ibm_db_sa://bwj90831:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

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

The Results are-

F9 FT B1022

F9 FT B1026

F9 FT B1021.2

F9 FT B1031.2

Between 4000 and 6000 is used to only display successful landings with payload in range 4000-6000

TOTAL NUMBER OF SUCCESSFUL AND FAILURE MISSION OUTCOMES

```
%sql SELECT COUNT(*) FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE '%Success%' OR MISSION_OUTCOME LIKE '%Failure%'
```

```
 * ibm_db_sa://bwj90831:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

1
101

Count function is used to get the count of the landings.

We can see that the total number of successful and failed missions are 101.

BOOSTERS CARRIED MAXIMUM PAYLOAD

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

```
 * ibm_db_sa://bwj90831:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

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

Max Function is used in the subquery to get the booster version with the highest payload mass.

2015 LAUNCH RECORDS

```
%sql SELECT TO_CHAR(TO_DATE(MONTH("DATE"), 'MM'), 'MONTH') AS MONTH_NAME, \
    LANDING__OUTCOME AS LANDING__OUTCOME, \
    BOOSTER_VERSION AS BOOSTER_VERSION, \
    LAUNCH_SITE AS LAUNCH_SITE \
    FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND "DATE" LIKE '%2015%'
```

```
 * ibm_db_sa://bwj90831:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

month_name	landing__outcome	booster_version	launch_site
JANUARY	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
APRIL	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

The failed launch outcomes in the year 2015 are -

F9 v1.1 B1012

F9 v1.1 B1015

RANK LANDING OUTCOMES BETWEEN 2010-06-04 AND 2017-03-20

```
%sql SELECT "DATE", COUNT(LANDING__OUTCOME) as COUNT FROM SPACEXTBL \
    WHERE "DATE" BETWEEN '2010-06-04' and '2017-03-20' AND LANDING__OUTCOME LIKE '%Success%' \
    GROUP BY "DATE" \
    ORDER BY COUNT(LANDING__OUTCOME) DESC
```

```
 * ibm_db_sa://bwj90831:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

DATE	COUNT
2015-12-22	1
2016-04-08	1
2016-05-06	1
2016-05-27	1
2016-07-18	1
2016-08-14	1
2017-01-14	1
2017-02-19	1

Function Count counts the records in the column, WHERE clause is used to filter the records

Desc Orders the Count of Landing Outcomes in descending order.

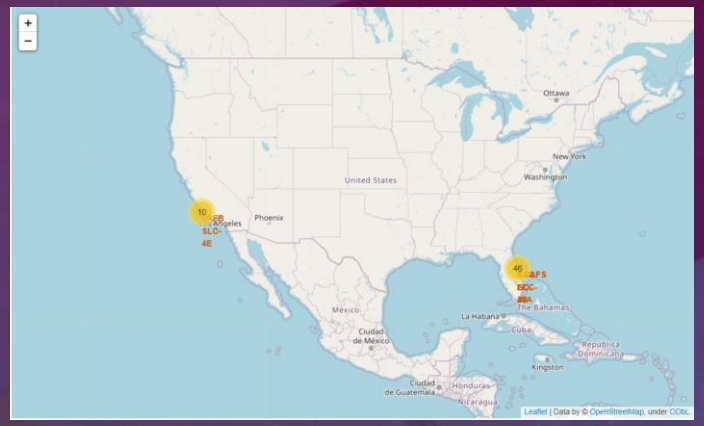
Section 4

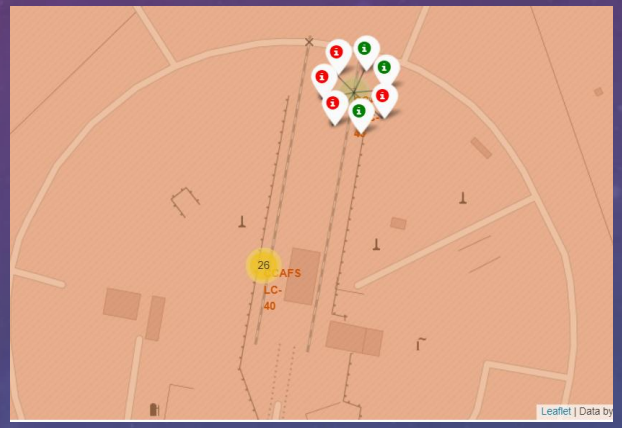
Launch Sites Proximities Analysis

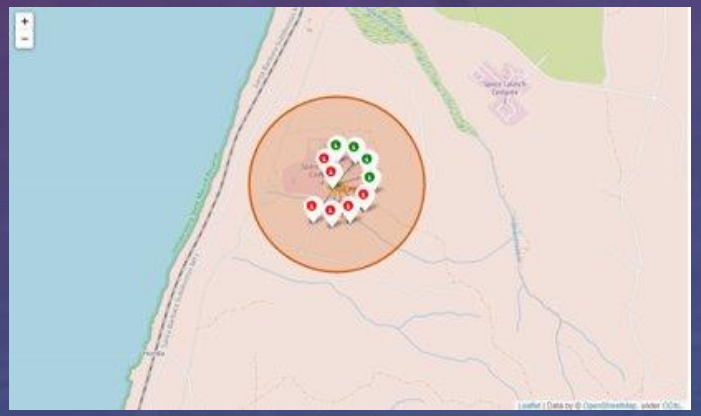
LAUNCH SITE GLOBAL MARKERS

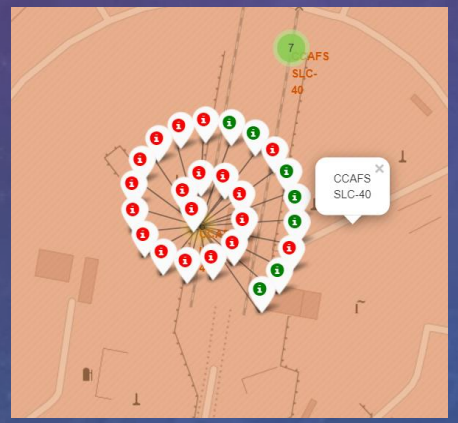
We can see that the launch sites are on the east and the west coasts of USA.

LAUNCH OUTCOMES COLORED MARKERS



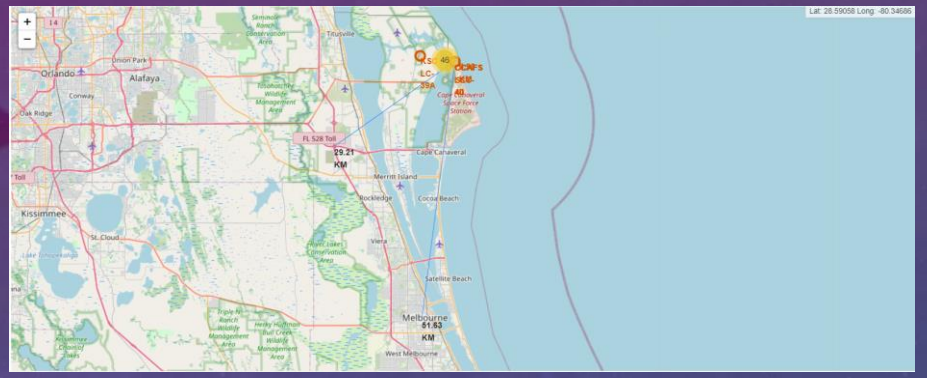






The Cluster Markers in folium can be clicked to reveal green and red markers to show successful and failed landing outcomes respectively.

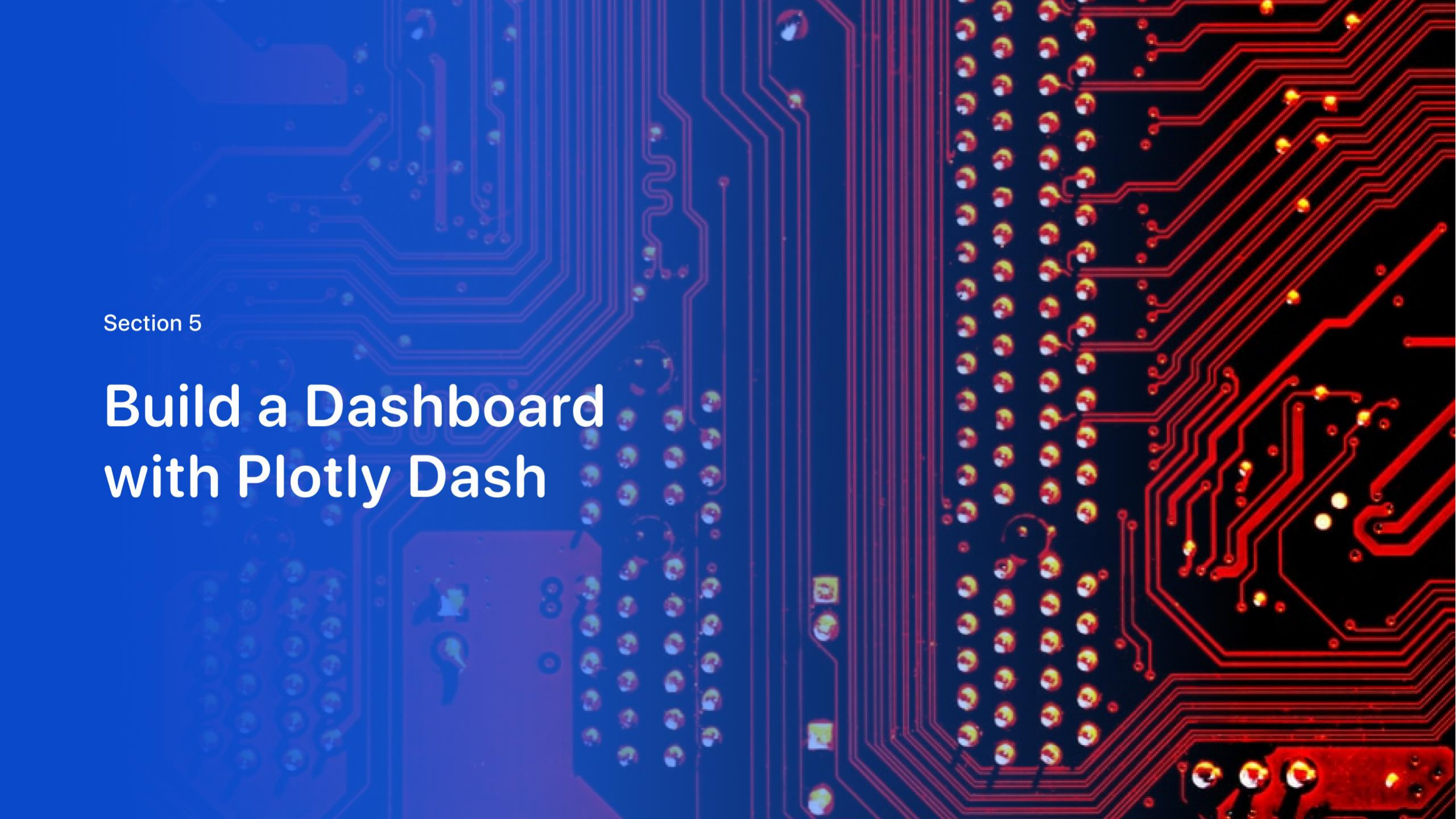
KEY LOCATIONS PROXIMITIES



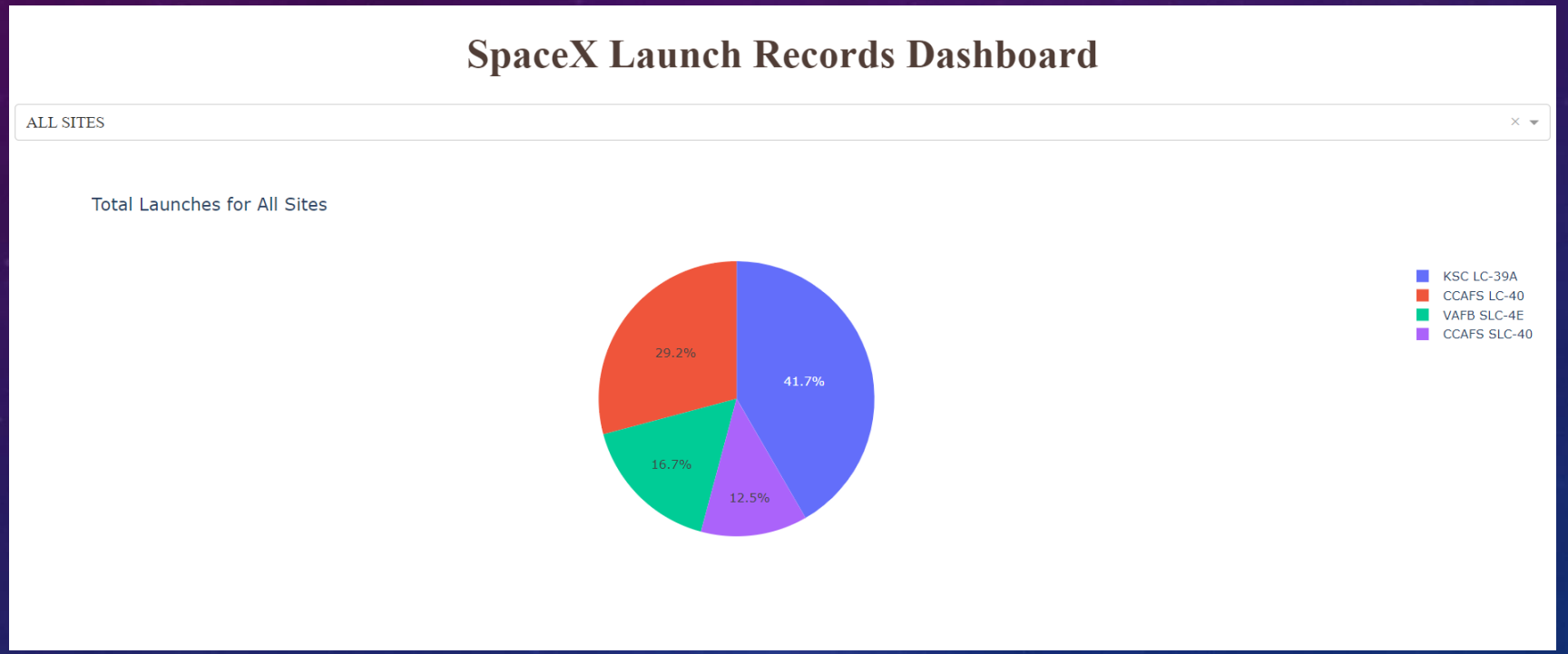
Launch sites are close to highways and coast for transportation and supply. They're also far away from cities in order to mitigate risk to bystanders should a rocket experience a catastrophic failure.

Section 5

Build a Dashboard with Plotly Dash

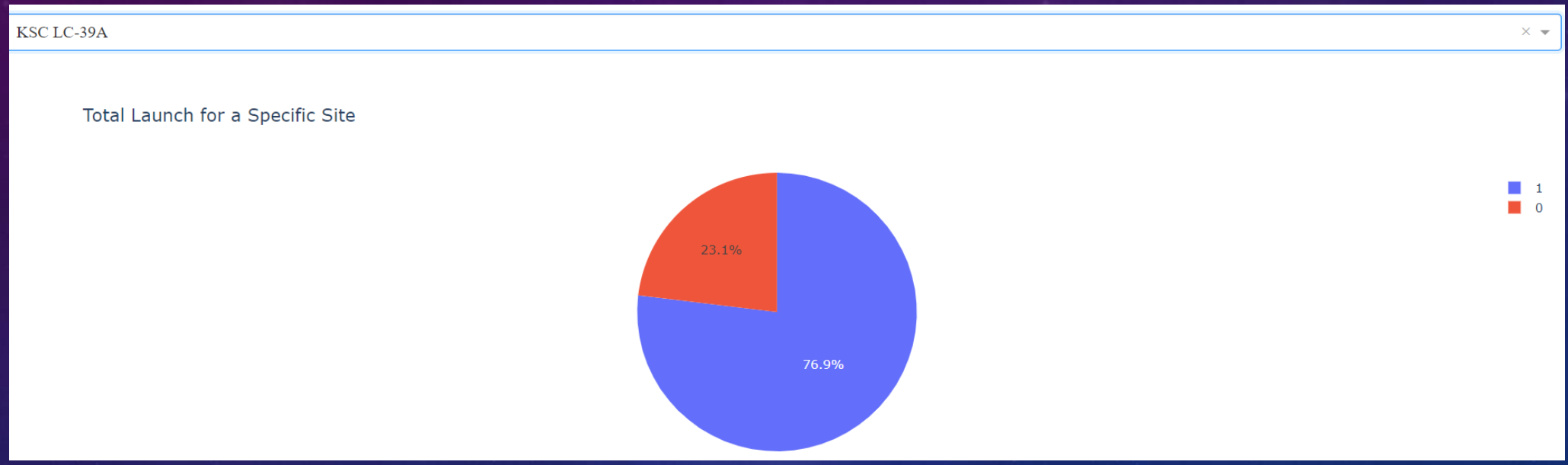


SPACEX SUCCESSFUL LAUNCHES



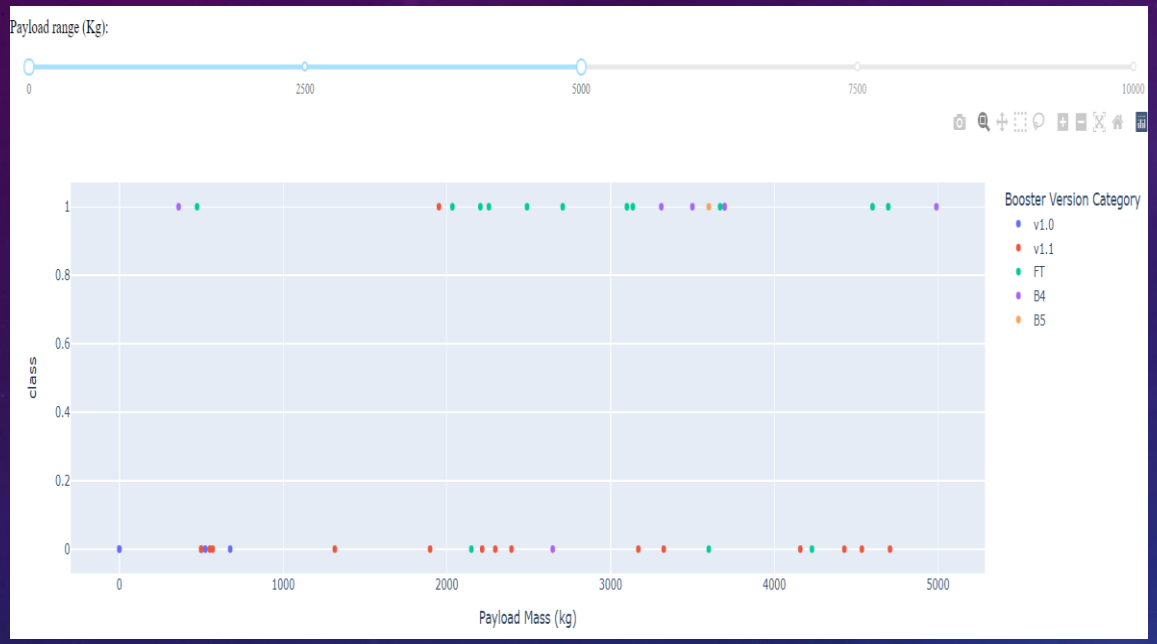
We can observe that KSC LC-39A has the most successful launches followed by CCAFS LC-40 and VAFB SLC-4E.CCAFS SLC-40 has the least number of successful launches.

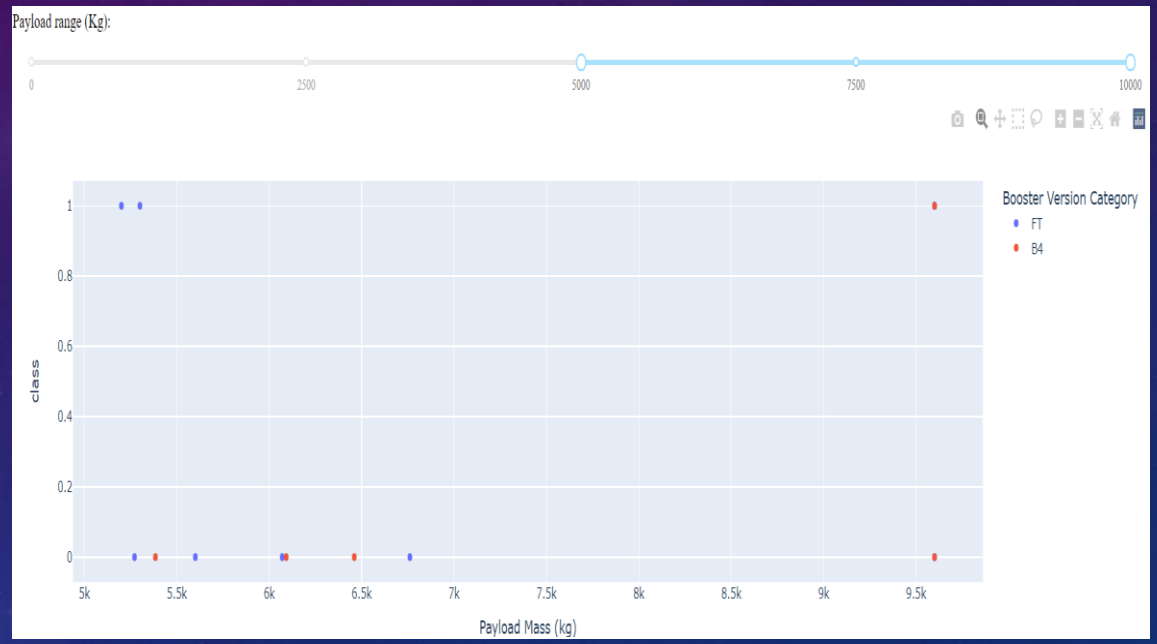
LAUNCH SITE WITH HIGHEST SUCCESS



KSC LC-39A has success rate of 76.9% and failure rate of 23.1%.

SUCCESS PAYLOAD PLOT



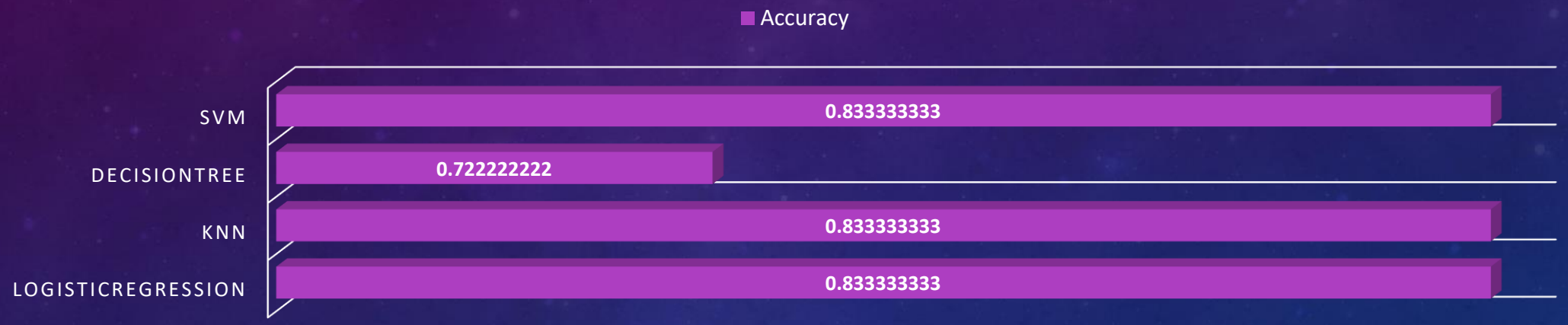


We can observe that the success rate of payload 0-5000 kg range is higher than the success rate of the 5000-10000 kg range.

Section 6

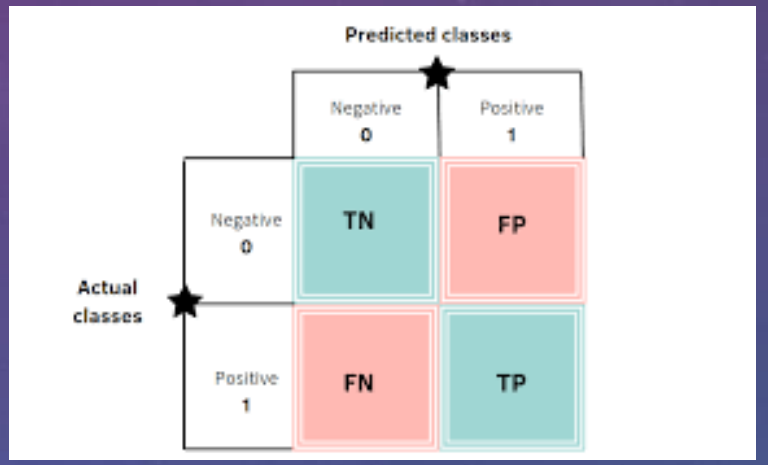
Predictive Analysis (Classification)

CLASSIFICATION ACCURACY



KNN, Logistic Regression, SVM Have the same accuracy of 0.8333
Decision Tree Model has the accuracy of 0.7222 on the test dataset

CONFUSION MATRIX



The confusion matrix for the KNN model has 3 false positives, shows us that three of the landings predicted by the model as success were in fact failures.

CONCLUSIONS

- We trained a KNN Model with an accuracy of 83.33% to predict the landing success rate for the launches for different launch sites and payloads. This is the best performing model.
- We observe that the launches from the launch site KSC LC-39A had the most success rate.
- We also observe that the lower payload launches have higher success rate than their heavier load counterpart.
- Orbits GEO,HEO,SSO,ES-L1 have the highest success rate for the launches.
- Thus We have thus used our data-driven insights to determine if the first stage of Falcon 9 will land successfully.

APPENDIX

GitHub Repository : https://github.com/Neel-XV/IBM-Data-Science/tree/master

Dashboard On Heroku : http://coursera-dash.herokuapp.com/

Thank you!