IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Aniruddha Chiplunkar
08/27/2024

# Outline

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

# Executive Summary

Summary of methodologies

- Data collection
- Data wrangling
- Exploratory Data Analysis with Data Visualization Libraries
- Exploratory Data Analysis with SQL
- Building an interactive map with Folium
- Building a flexible Dashboard with Plotly Dash
- Predictive analysis using ML Classification

Summary of all results

- EDA results
- Screenshots of Interactive Analysis
- Results of Accuarcy in Predictive Analysis

# Introduction

**Project background and context**

SpaceX, founded by Elon Musk in 2002, is a private aerospace manufacturer and space transportation company. One of its most notable achievements is the development of the Falcon 9 rocket. Costing about $62 million each, it's a reusable two-stage rocket designed for the reliable and safe transport of satellites into orbit. The Falcon 9 is known for its ability to return its first stage to Earth, which brings us to our analysis. If we can predict if the first stage will land or not, we can help determine the cost for each launch. Based on data from the web, we are going to use Machine Learning techniques to predict if SpaceX will reuse the first stage.

**Problems you want to find answers**

- How do predictor variables such as launch site, payload mass, orbital path, etc affect the reuse of its first stage? Essentially, which factors have the most effect on the success of the first stage landing?
- In what conditions (on average) do we yield the best landing?
- Out of the several classification techniques learned, which one will yield the most accurate result?

Section 1

# Methodology

# Methodology

## Executive Summary

- Data collection methodology:
  - To begin, we used HTTP techniques to request data from the SPACEX API
  - To increase specificity, we used BeautifulSoup to scrape launch data from Wikipedia
- Perform data wrangling
  - Filtering desired data
  - Replacing missing values
  - Using One-Hot-Encoder to prepare data for classification
- 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

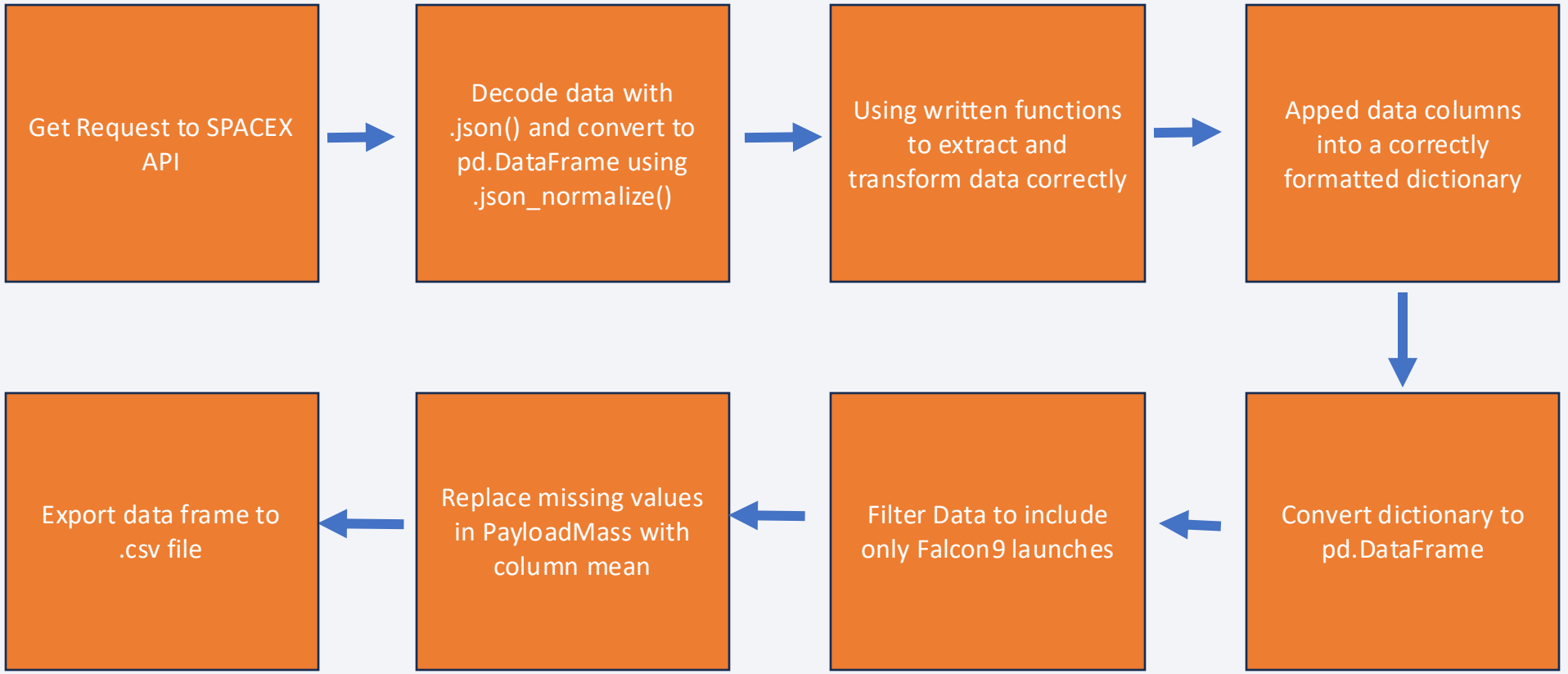
- Describe how data sets were collected.

In order to obtain a complete and concise data set, we relied on both REST API's and Web scraping. This process involved parsing through HTML nodes to find table entries and converting .json files to a Pandas Data Frame.

Data Columns obtained through SpaceX API: FlightNumber, Date, BoosterVersion, PayloadMass, Outcome, Flights, LandingPad, Block, ReusedCount, Serial, GridFins, Reused, Legs, Longitude, Latitude, Orbit, LaunchSite,

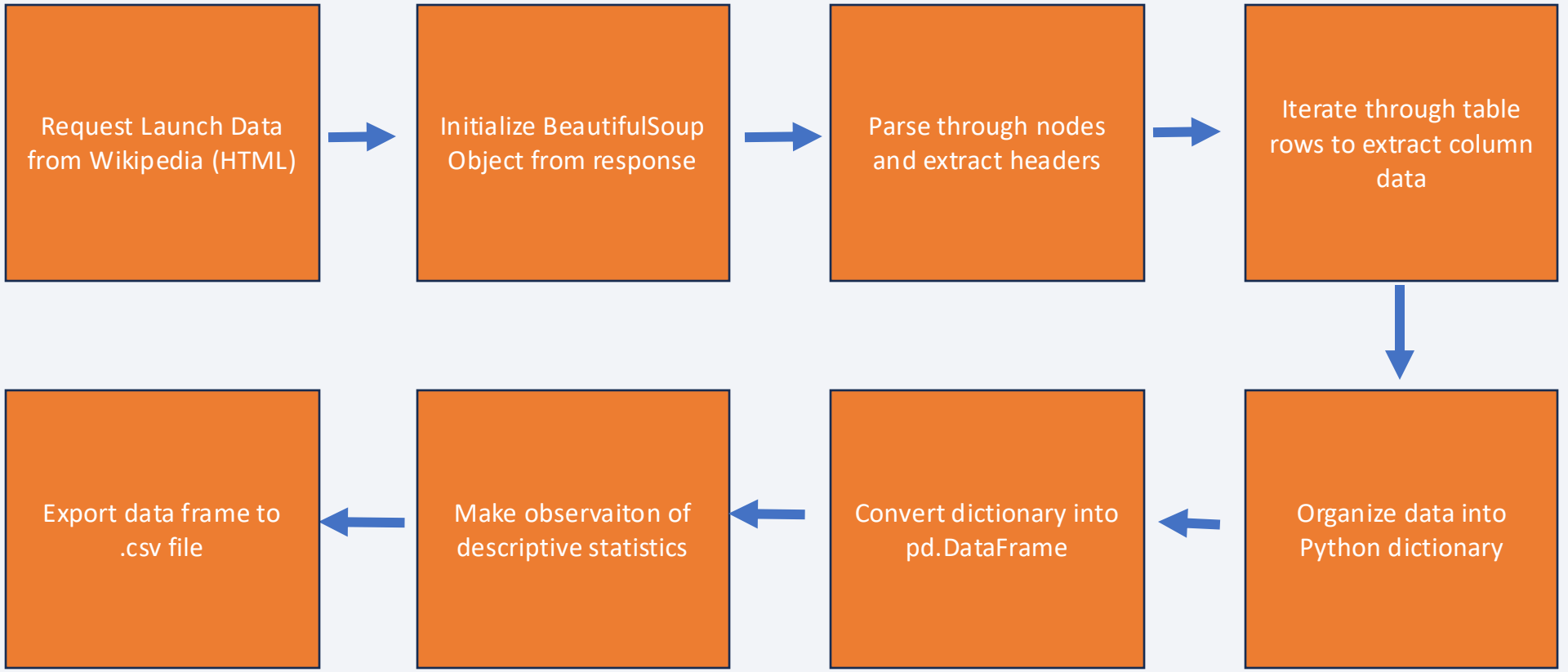
Data Columns obtained through Wikipedia Web Scraping: Flight_Number, Launch site, Payload, Payload_Mass_Kg, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time

# Data Collection – SpaceX API



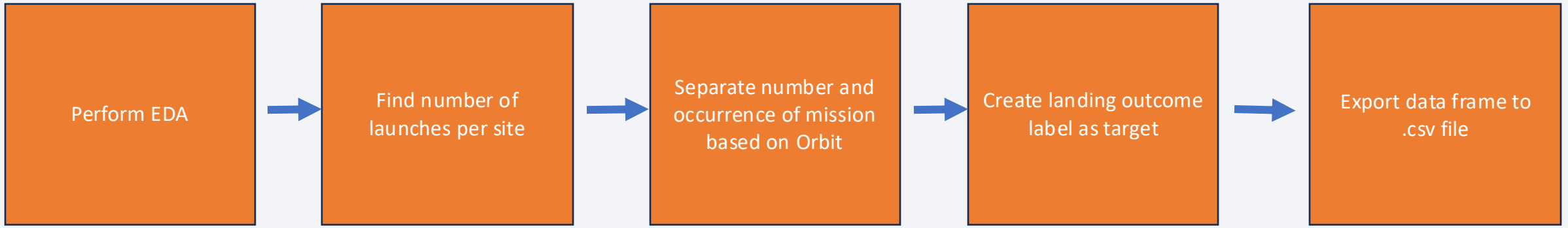

https://github.com/anichip/IBM_Capstone_SPACEX_Data_Analysis/blob/0d8423942178dcf89c3c52453ffd515c7b1a1e1e/CAP1.1_Data_Collection.ipynb

# Data Collection - Scraping




https://github.com/anichip/IBM_Capstone_SPACEX_Data_Analysis/blob/0d8423942178dcf89c3c52453ffd515c7b1a1e1e/CAP1.2_WebScraping.ipynb

# Data Wrangling




In our data set, there are several cases where the booster landing was unsuccessful. Many happen because of a natural or manufacturing accident. Here are the outcomes: True Ocean means the stage was successfully landed in an ocean while False Ocean means the stage was unsuccessfully landed in an ocean. True RTLS means the mission outcome reads "successfully landed to a ground pad", and False RTLS means the mission outcome reads "unsuccessfully landed to a ground pad." True ASDS means successfully landed on a drone ship. False ASDS means unsuccessfully landed on a drone ship.

For the purpose of classification, we convert those outcomes into training labels of “1” for success and “0” for fail.

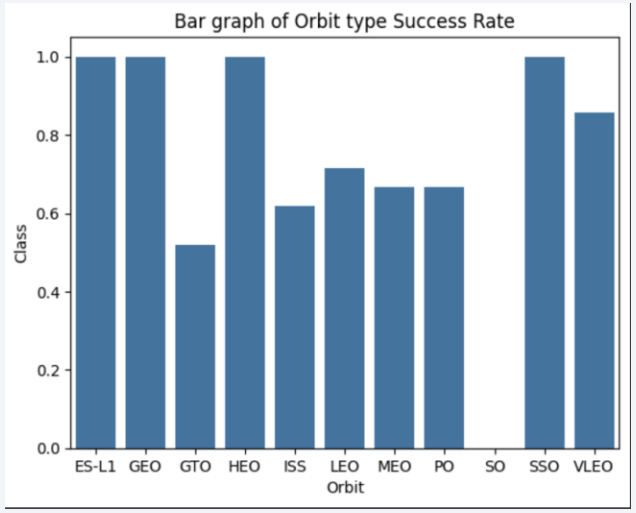


https://github.com/anichip/IBM_Capstone_SPACEX_Data_Analysis/blob/0d8423942178dcf89c3c52453ffd515c7b1a1e1e/CAP1.3_Data%20wrangling.ipynb

# EDA with Data Visualization

Using Categorical Scatter Point Plots, we identified the relationship between:

- Payload Mass vs. Flight #
- Flight # vs. Launch Site
- Payload Mass vs. Launch Site
- Flight # vs. Orbit Type
- Payload Mass vs. Orbit Type




We use Categorical Scatter plots to show how certain independent variables affect other variables. When we set the hue to "Class", we can determine which factors are more likely to yield a successful landing.

https://github.com/anichip/IBM_Capstone_SPACEX_Data_Analysis/blob/0d8423942178dcf89c3c52453ffd515c7b1a1e1e/CAP2.2-eda-dataviz.ipynb

# EDA with SQL

- Established connection with sqlite3
- *Displaying the names of the launch sites.*
- *Displaying 5 records where launch sites begin with 'CCA'.*
- *Displaying the total payload mass carried by booster launched by NASA (CRS).*
- *Displaying the average payload mass carried by booster version F9 v1.1.*
- *Showing the date when the first successful landing outcome in ground pad was achieved.*
- *Listing the names of the boosters which have success in drone ship and have payload mass between 4000 and 6000.*
- *Displaying the total number of successful and failure mission outcomes.*
- *Selecting the names of the booster versions which have carried the maximum payload mass.*
- *Using grouping to note the failed landing outcomes in drone ship, their booster versions, and launch sites names for in year 2015.*
- Used subqueries to rank landing outcomes by occurrence between the date 06/04/2010 and 03/20/2017

https://github.com/anichip/IBM_Capstone_SPACEX_Data_Analysis/blob/0d8423942178dcf89c3c52453ffd515c7b1a1e1e/CAP2.1_sqllite.ipynb

# Build an Interactive Map with Folium

The bread and butter of our Folium analysis were the Latitude and Longitude columns from our transformed dataset. We plotted a circle on each launch site, and revealed information through the hovering of one's cursor.

The MarkerCluster() and Circle() objects were used because they offer customizability and intricate details.

For readability, we assigned green points to "success" and "orange" for failure

Now that we had an interactive map, we needed to answer these questions:

*How close were the launch sites to cities? What about railways and highways?*

*What could we infer from our map?*

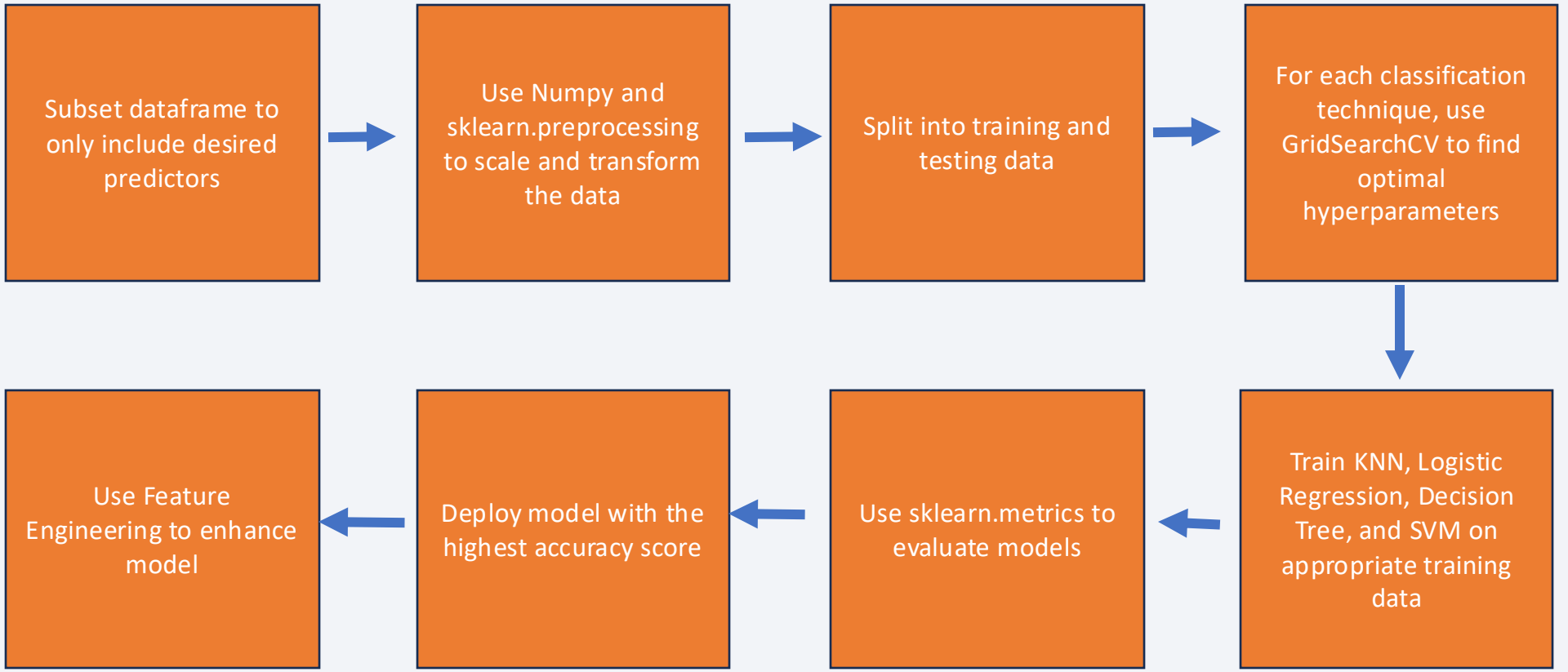
https://github.com/anichip/IBM_Capstone_SPACEX_Data_Analysis/blob/0d8423942178dcf89c3c52453ffd515c7b1a1e1e/CAP3.1_Folium.ipynb

# Build a Dashboard with Plotly Dash

- *Interactive Plotly dashboards allow the user to select and play with data findings. This makes it easy to present to stakeholders*
- *We plotted pie charts showing the total launches by a certain sites.*
- *Our visualizations included pie charts that displayed launch numbers based on site. In addition, we included scatter plots to show the relationship between desired independent and target variables.*
- *Sliders and Dropdowns allow for easy access to graphs*

https://github.com/anichip/IBM_Capstone_SPACEX_Data_Analysis/blob/0d8423942178dcf89c3c52453ffd515c7b1a1e1e/ani_spacex_dash_app.py

# Predictive Analysis (Classification)

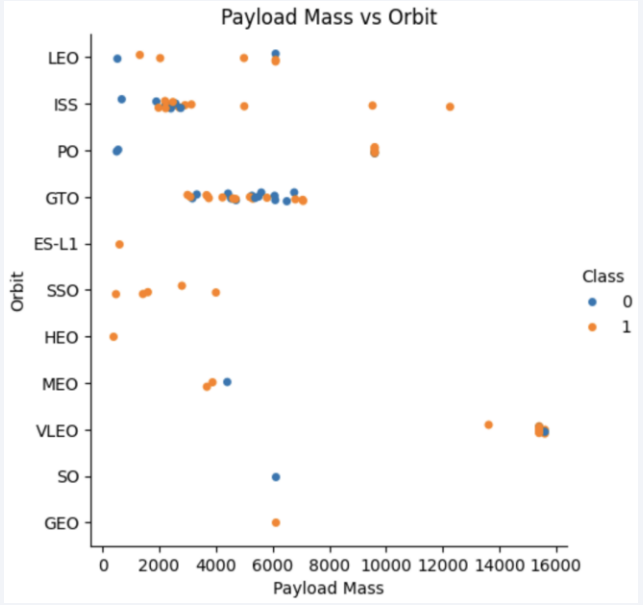


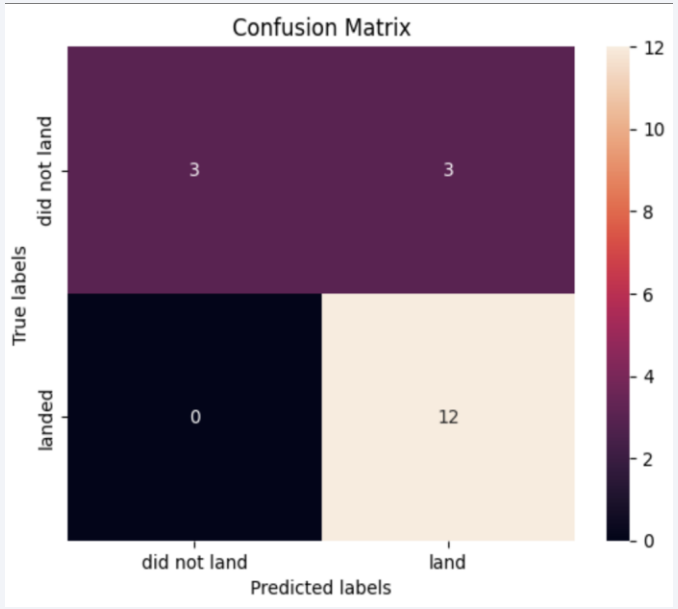

https://github.com/anichip/IBM_Capstone_SPACEX_Data_Analysis/blob/0d8423942178dcf89c3c52453ffd515c7b1a1e1e/CAP4.1-MLPred.ipynb

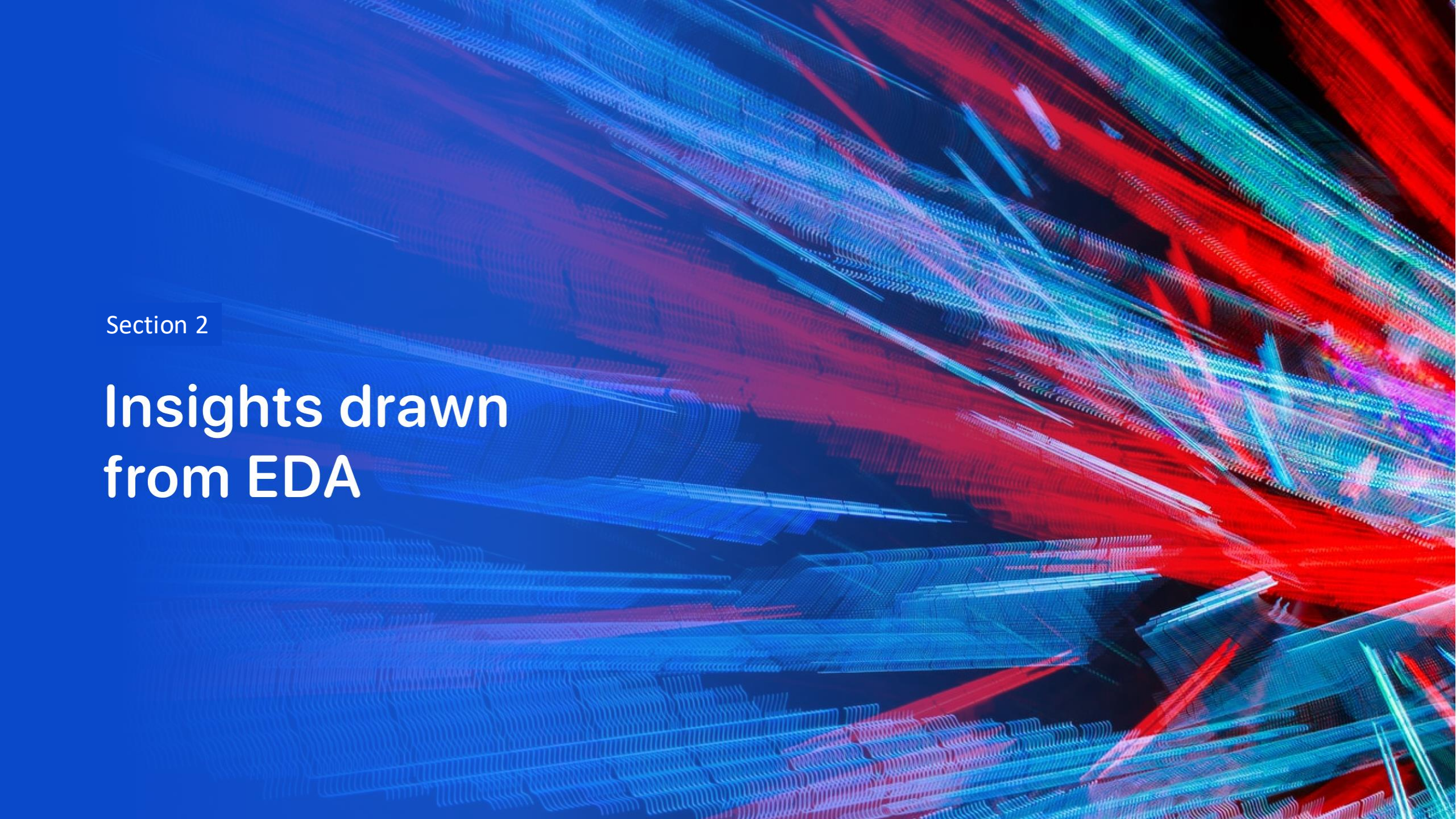
# Results

How we will display:

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



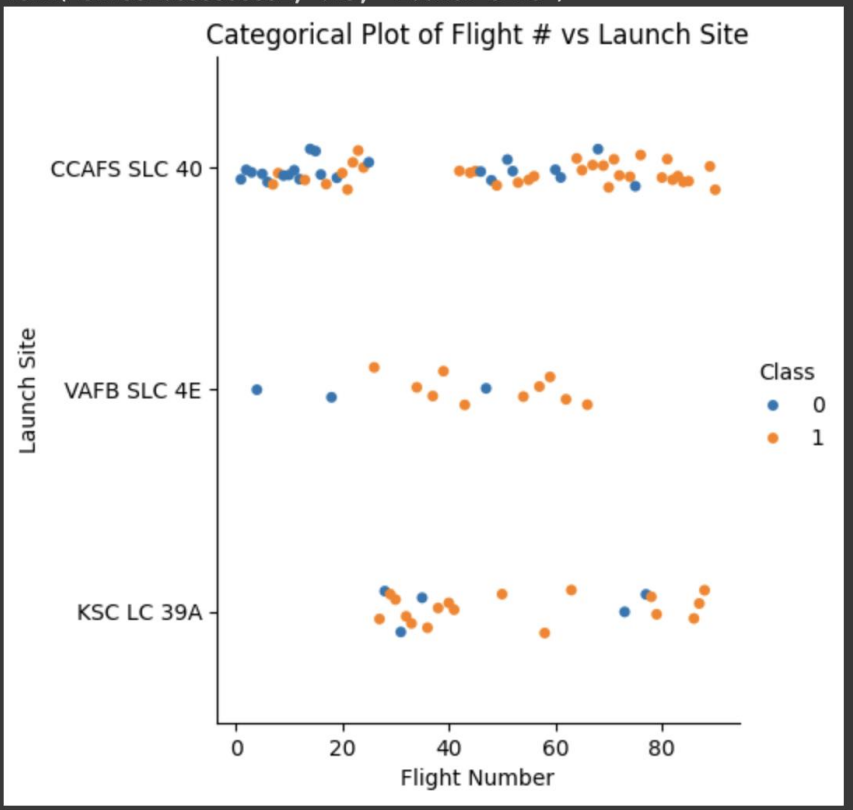


Section 2
Insights drawn from EDA

# Flight Number vs. Launch Site

- Seaborn allows us to plot the two variables against each other and distinguish data points by Class
- We use a Categorical plot because it shows compares flight variability across launch sites
- From this, we can infer that earlier flights tend to succeed, whereas later flights tend to fail.
- From our data, most launches came from CCAFS SLC 40
- KDC LC 39A appears to have the highest rate of success
- Failure is greater as flight number increases. [Flight Number is an Ordinal Numeric Variable)

```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be th
sns.catplot(data=df,x="FlightNumber",y="LaunchSite",hue="Class")
plt.title("Categorical Plot of Flight # vs Launch Site")
plt.xlabel("Flight Number")
plt.ylabel("Launch Site")
```

```
Text(45.75320833333332, 0.5, 'Launch Site')
```



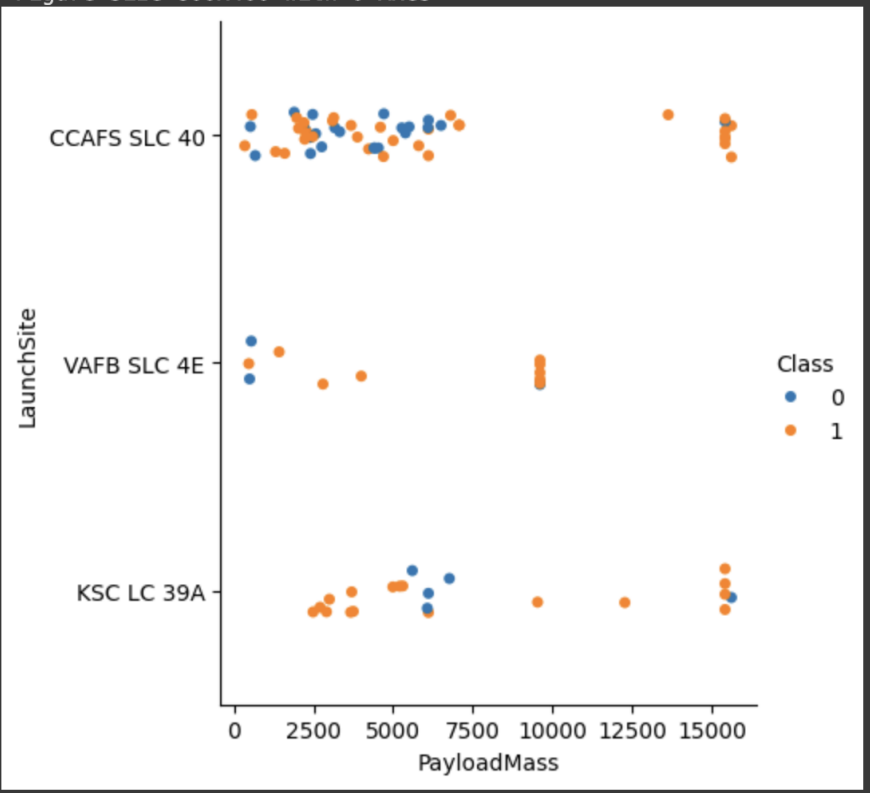
# Payload vs. Launch Site

- Here we compare a quantitative variable with a categorical variable
- As launch payloads get heavier, the number of unsuccessful landings increase
- CCAFS SLC 40 has the highest number of successes, and the launches had lower Payload Mass

We also want to observe if there is any relationship between launch sites and their payload mass.

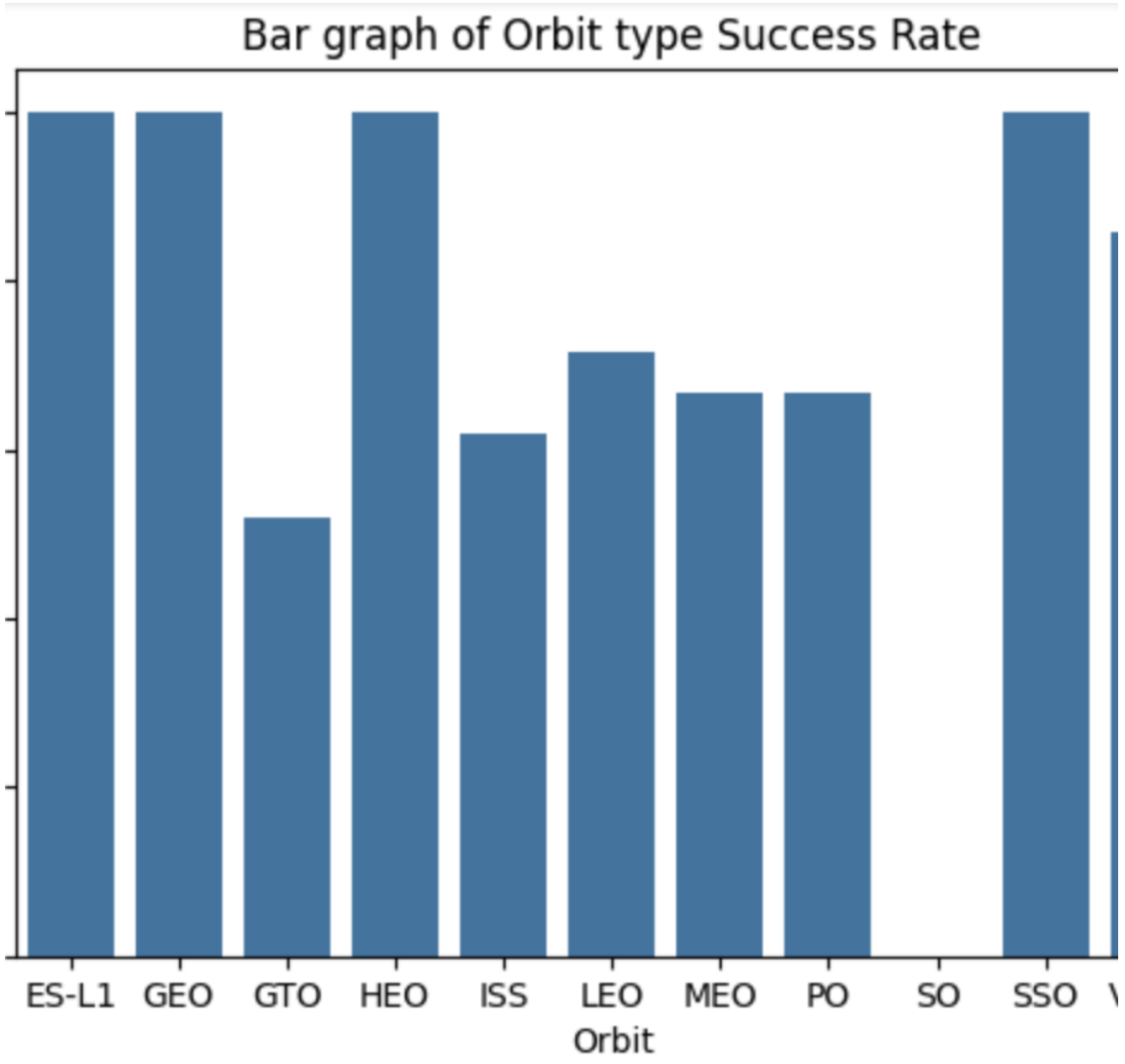
```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis t
plt.figure(figsize=(8,4))
sns.catplot(data=df,x="PayloadMass",y="LaunchSite",hue="Class")
```

```
<seaborn.axisgrid.FacetGrid at 0x796ddda3d150>
<Figure size 800x400 with 0 Axes>
```



# Success Rate vs. Orbit Type

- Bar Graphs allow us to compare frequency across categories
- Only a few launches have embarked the HEO, SSO, GEO, and ES-L1 orbital paths, and they were all successes.
- We can infer from the graph that VLEO and LEO have greater success rates compared to MEO and GTO.
    - SPACEX coordinators may consider lower orbital paths compared to elliptical geostationary orbital paths.

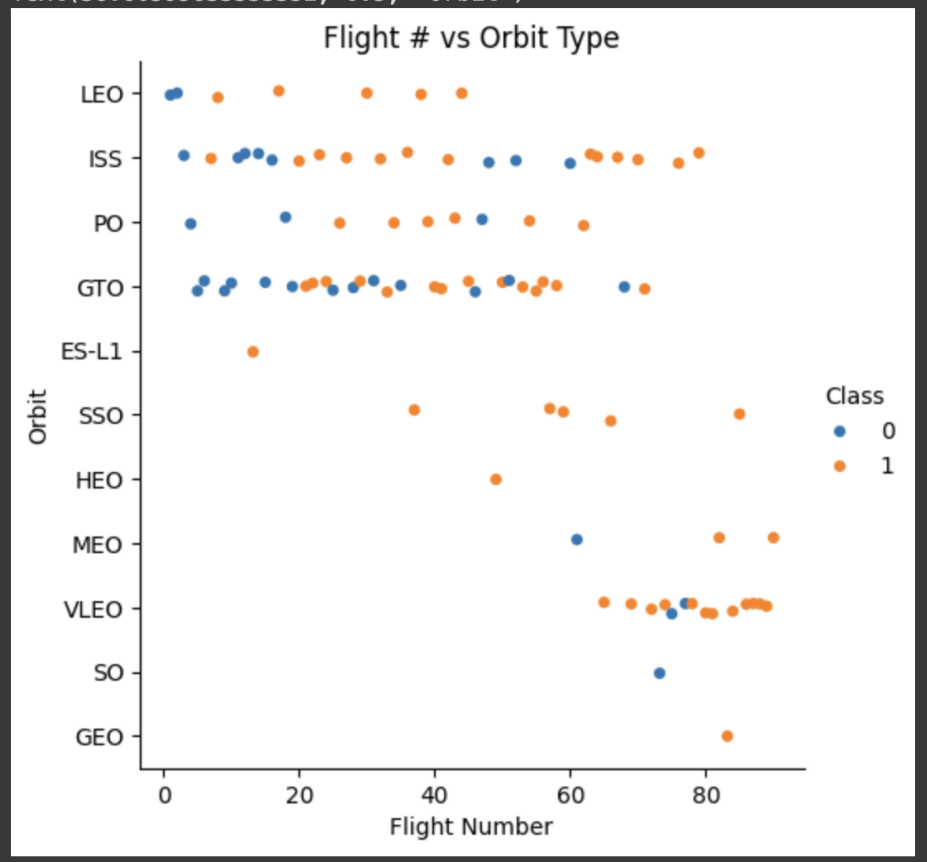


# Flight Number vs. Orbit Type

- The first several flights take geostationary orbital paths, chief among them being around the International Space Station (ISS)
- Flights that follow GTO show the highest rate of success as flight number increases.
- Launches later in the order are more likely to be unsuccessful across all orbital paths.

```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to b
sns.catplot(data=df,x="FlightNumber",y="Orbit",hue="Class")
plt.title("Flight # vs Orbit Type")
plt.xlabel("Flight Number")
plt.ylabel("Orbit")
```

```
Text(38.80595833333332, 0.5, 'Orbit')
```

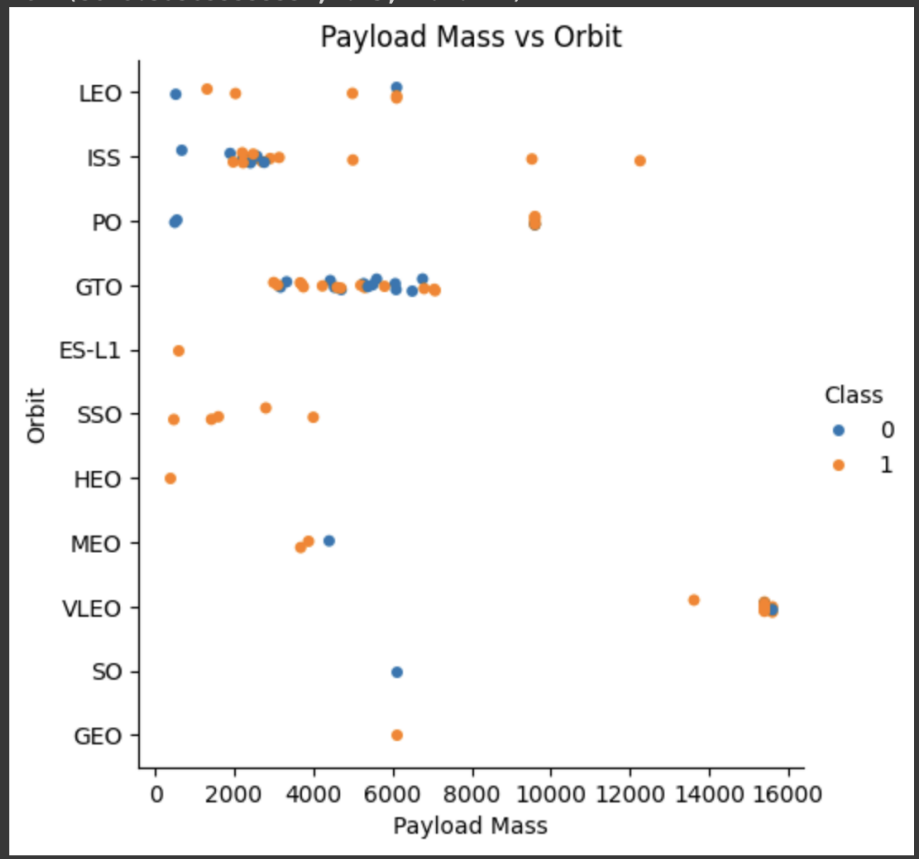


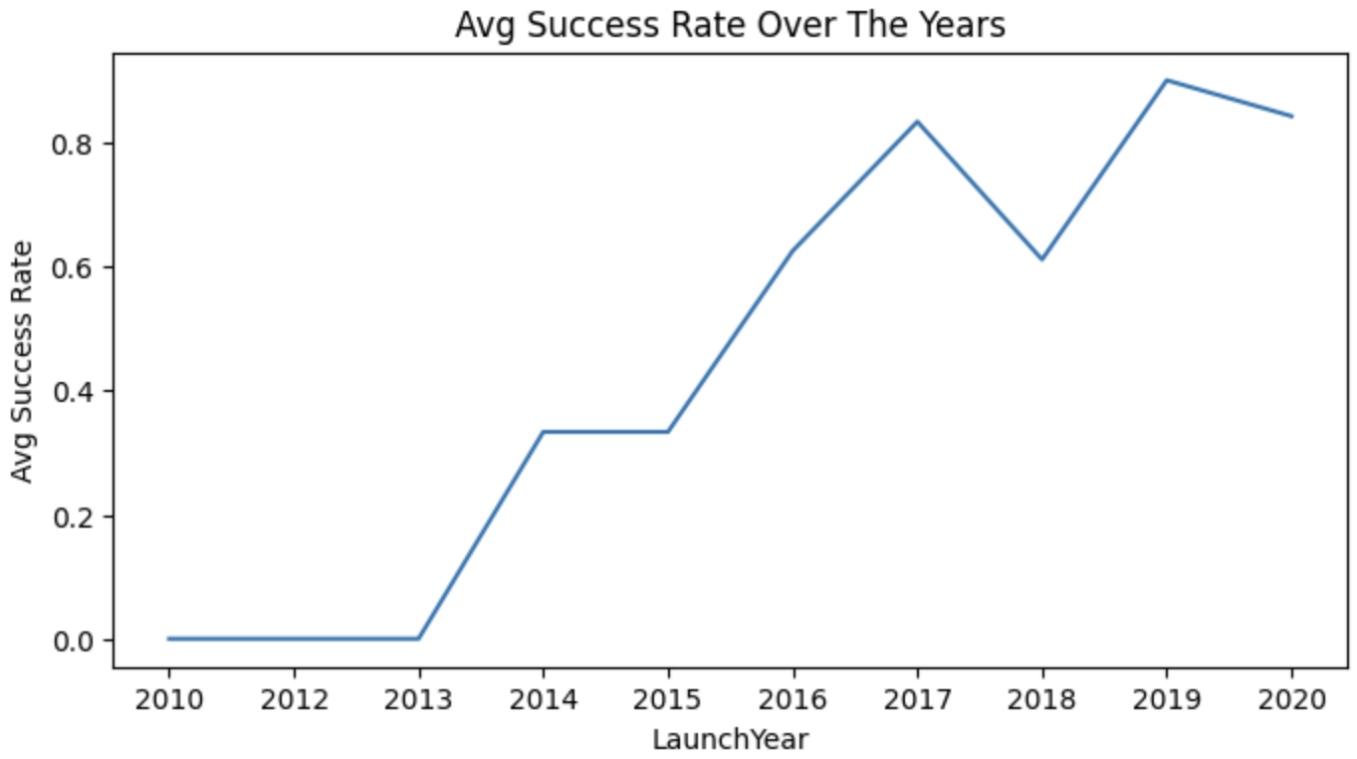
# Payload vs. Orbit Type

- The distribution is random and scattered, showing little bias
- Launches that follow the GTO path with masses between 4300 and 7500 kg yield the highest success rates.
- Overall, heavier payloads have negative effects on GTO and larger orbits, and positive effects on ISS and LEO orbits.

```
# Plot a scatter point chart with x axis to be Payload and y axis to be the
sns.catplot(data=df,x="PayloadMass",y="Orbit",hue="Class")
plt.title("Payload Mass vs Orbit")
plt.xlabel("Payload Mass")
plt.ylabel("Orbit")
```

```
Text(38.80595833333332, 0.5, 'Orbit')
```






# Launch Success Yearly Trend

As the program developed, the number of successful first stage use cases increased. With a slight plateau in 2018, the average success rate has seen a tremendous increase in between the years 2013 and 2020.

# All Launch Site Names

Displayed the unique names of all Launch Sites from the SPACEX Table

```
%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'

All records from CCA-designated launch sites

```
%sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%';
```

```
 * sqlite:///my_data1.db
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 | 7:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | No attempt |
| 2012-10-08 | 0:35:00 | 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 |
| 2013-12-03 | 22:41:00 | F9 v1.1 | CCAFS LC-40 | SES-8 | 3170 | GTO | SES | Success | No attempt |
| 2014-01-06 | 22:06:00 | F9 v1.1 | CCAFS LC-40 | Thaicom 6 | 3325 | GTO | Thaicom | Success | No attempt |
| 2014-04-18 | 19:25:00 | F9 v1.1 | CCAFS LC-40 | SpaceX CRS-3 | 2296 | LEO (ISS) | NASA (CRS) | Success | Controlled (ocean) |
| 2014-07-14 | 15:15:00 | F9 v1.1 | CCAFS LC-40 | OG2 Mission 1 6 Orbcomm-OG2 satellites | 1316 | LEO | Orbcomm | Success | Controlled (ocean) |
| 2014-08-05 | 8:00:00 | F9 v1.1 | CCAFS LC-40 | AsiaSat 8 | 4535 | GTO | AsiaSat | Success | No attempt |
| 2014-09-07 | 5:00:00 | F9 v1.1 B1011 | CCAFS LC-40 | AsiaSat 6 | 4428 | GTO | AsiaSat | Success | No attempt |
| 2014-09-21 | 5:52:00 | F9 v1.1 B1010 | CCAFS LC-40 | SpaceX CRS-4 | 2216 | LEO (ISS) | NASA (CRS) | Success | Uncontrolled (ocean) |
| 2015-01-10 | 9:47:00 | F9 v1.1 B1012 | CCAFS LC-40 | SpaceX CRS-5 | 2395 | LEO (ISS) | NASA (CRS) | Success | Failure (drone ship) |
| 2015-02-11 | 23:03:00 | F9 v1.1 B1013 | CCAFS LC-40 | DSCOVR | 570 | HEO | U.S. Air Force NASA NOAA | Success | Controlled (ocean) |

# Total Payload Mass

Cumulative Payload Mass from NASA CRS

```
[ ] %sql SELECT SUM (PAYLOAD_MASS__KG_) FROM SPACEXTABLE  WHERE Customer IS 'NASA (CRS)'
```

```
 * sqlite:///my_data1.db
Done.
SUM (PAYLOAD_MASS__KG_)
45596
```

# Average Payload Mass by F9 v1.1

Using GROUP BY and WHERE clauses to find average payload mass of F9 boosters of version 1.1

```
%sql SELECT AVG (Payload_Mass_Kg) FROM SPACEXTABLE WHERE Booster_Version LIKE "F9 v1.1%";
```

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

| AVG (Payload_Mass_Kg) |
| --- |
| 2534.6666666666665 |

# First Successful Ground Landing Date

Displaying the earliest reporting of a Successful landing on ground pad

```
[ ]  %sql SELECT MIN (Date) FROM SPACEXTABLE WHERE Landing_Outcome = "Success (ground pad)";
```

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

| MIN (Date) |
| --- |
| 2015-12-22 |

# Successful Drone Ship Landing with Payload between 4000 and 6000

Used mathematical operations to obtain Booster Versions of rockets between 4000 and 6000 kg.

```
[ ] %sql SELECT DISTINCT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = "Success (drone ship)" AND Payload_Mass_Kg > 4000 AND Payload_Mass_Kg< 6000;

 * sqlite:///my_data1.db
Done.
Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2
```

# Total Number of Successful and Failure Mission Outcomes

Displayed the number of occurrence for each specific mission outcome in the dataset

```
[ ] %sql SELECT Mission_Outcome, COUNT (*) FROM SPACEXTABLE GROUP BY Mission_Outcome;
```

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

| Mission_Outcome | COUNT (*) |
|---|---|
| Failure (in flight) | 1 |
| Success | 98 |
| Success | 1 |
| Success (payload status unclear) | 1 |

# Boosters Carried Maximum Payload

Used subqueries to list the unique booster identifications that carry the maximum payload mass.

```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE Payload_Mass_Kg = (SELECT MAX (Payload_Mass_Kg) FROM SPACEXTABLE);
```

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

# 2015 Launch Records

Demonstrated skill with SUBSTR() function to list 2015 launch records that include month, landing outcome, booster version, and launch site

**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), Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTABLE WHERE SUBSTR(Date,0,5)='2015' AND Landing_Outcome
= "Failure (drone ship)";
```

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

| SUBSTR(Date,6,2) | 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 |

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

Used sub queries, BETWEEN clause, and GROUP BY clause to rank the frequencies of all landing outcomes in descending order

```
[ ] %sql SELECT Landing_Outcome, COUNT(*) AS C FROM (SELECT * FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20') GROUP BY Landing_Outcome ORDER
```

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

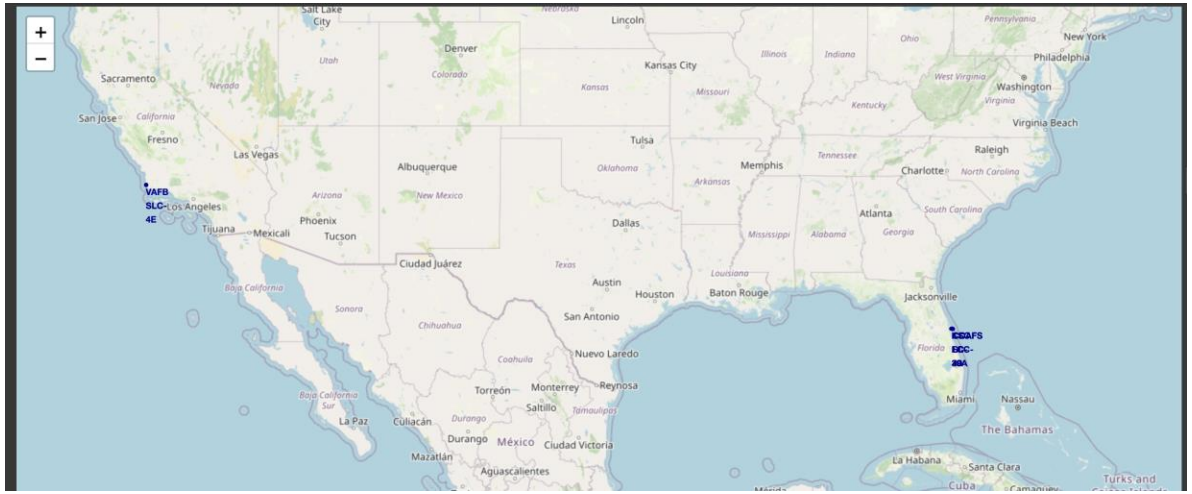
| Landing_Outcome | C |
|---|---|
| No attempt | 10 |
| Success (drone ship) | 5 |
| Failure (drone ship) | 5 |
| Success (ground pad) | 3 |
| Controlled (ocean) | 3 |
| Uncontrolled (ocean) | 2 |
| Failure (parachute) | 2 |
| Precluded (drone ship) | 1 |

Section 3

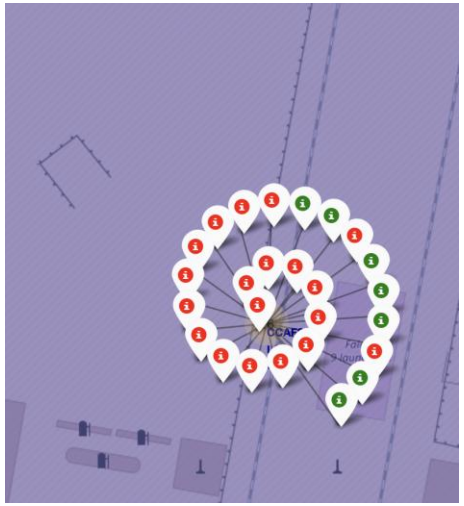
# Launch Sites Proximities Analysis

# Interactive Folium Map of all Launch Sites

- The markers were created as Folium Circle() objects, and they are colored blue to distinguish my work



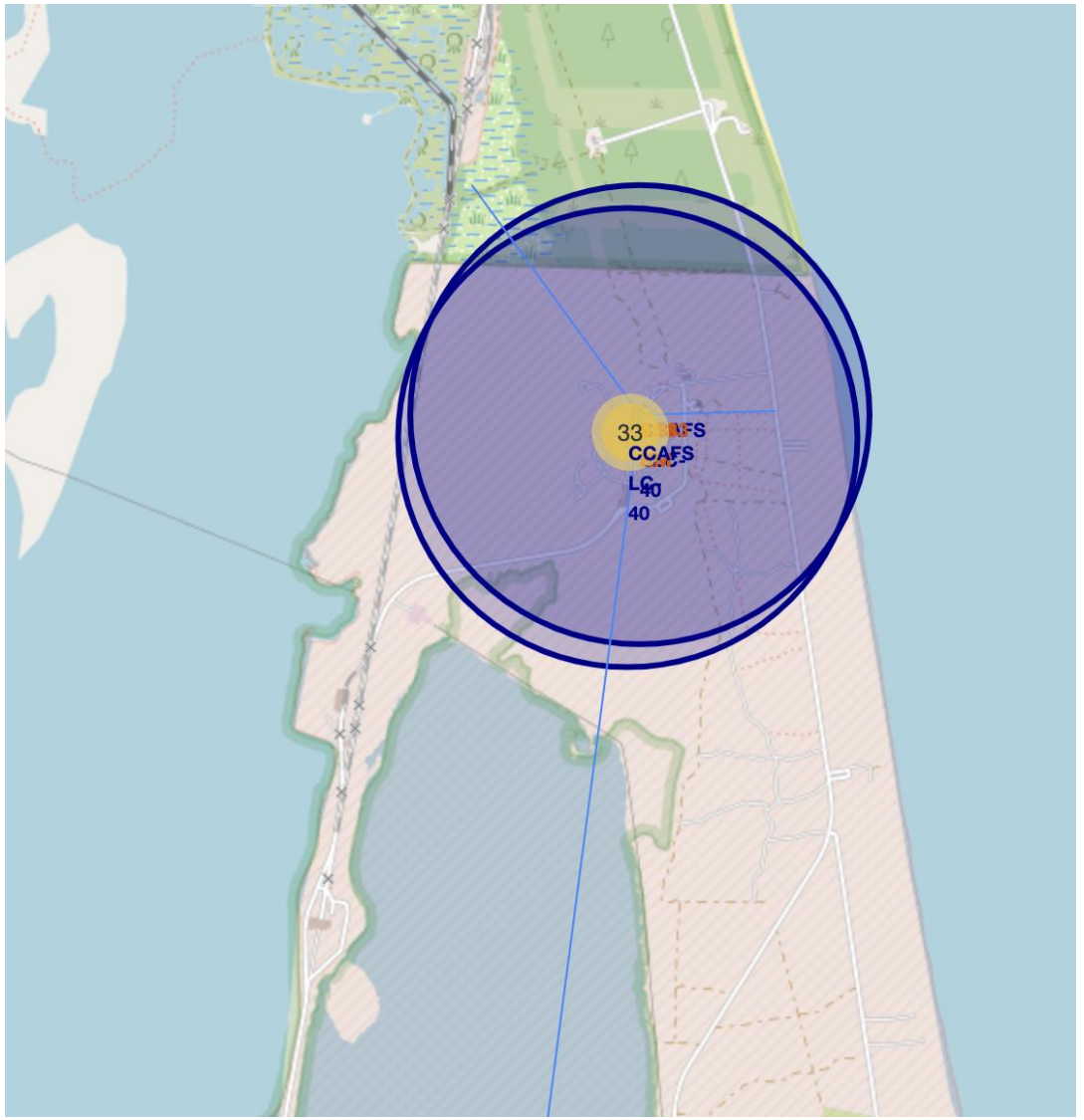
# Launch Sites marked with Color Labels

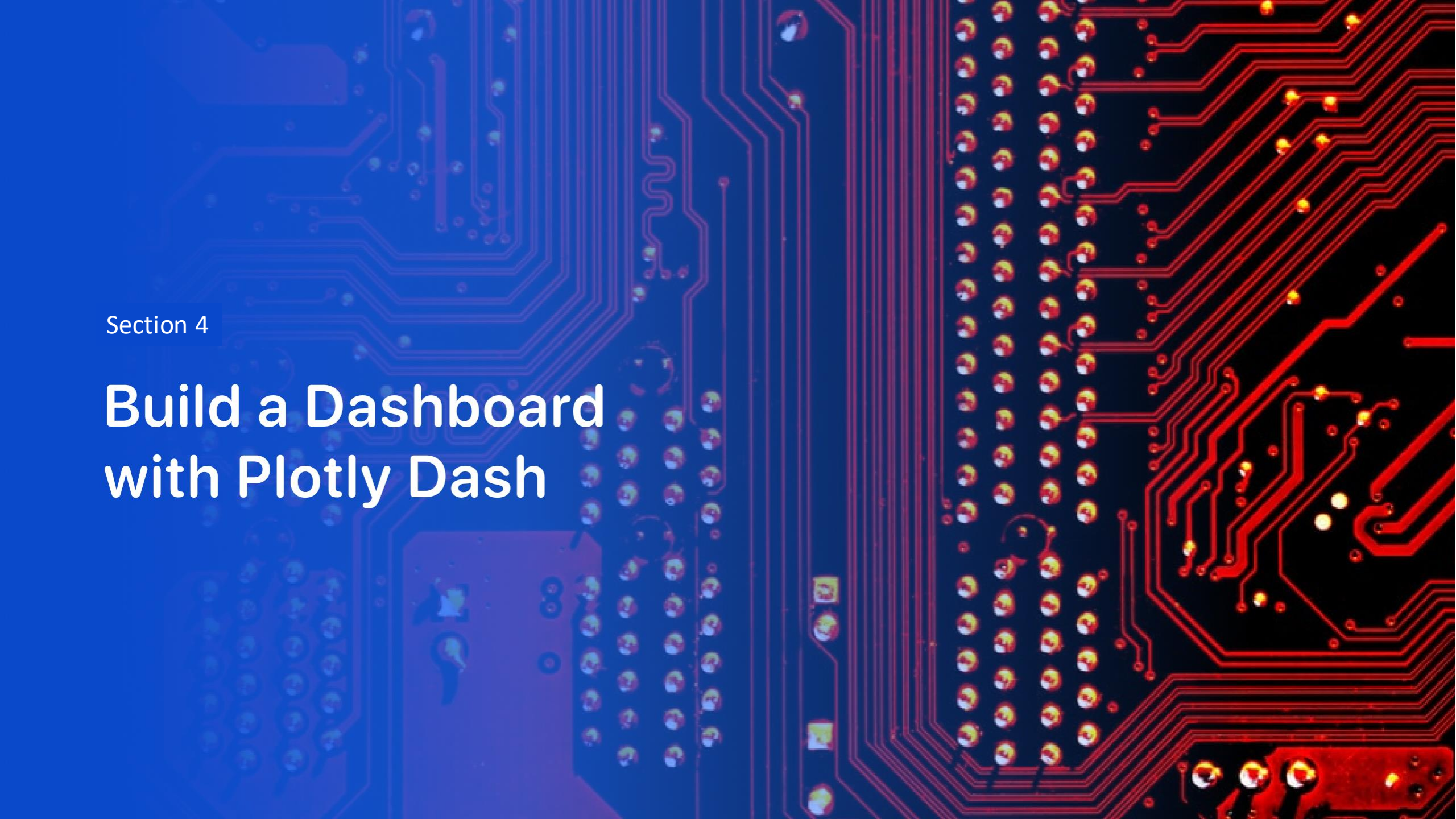


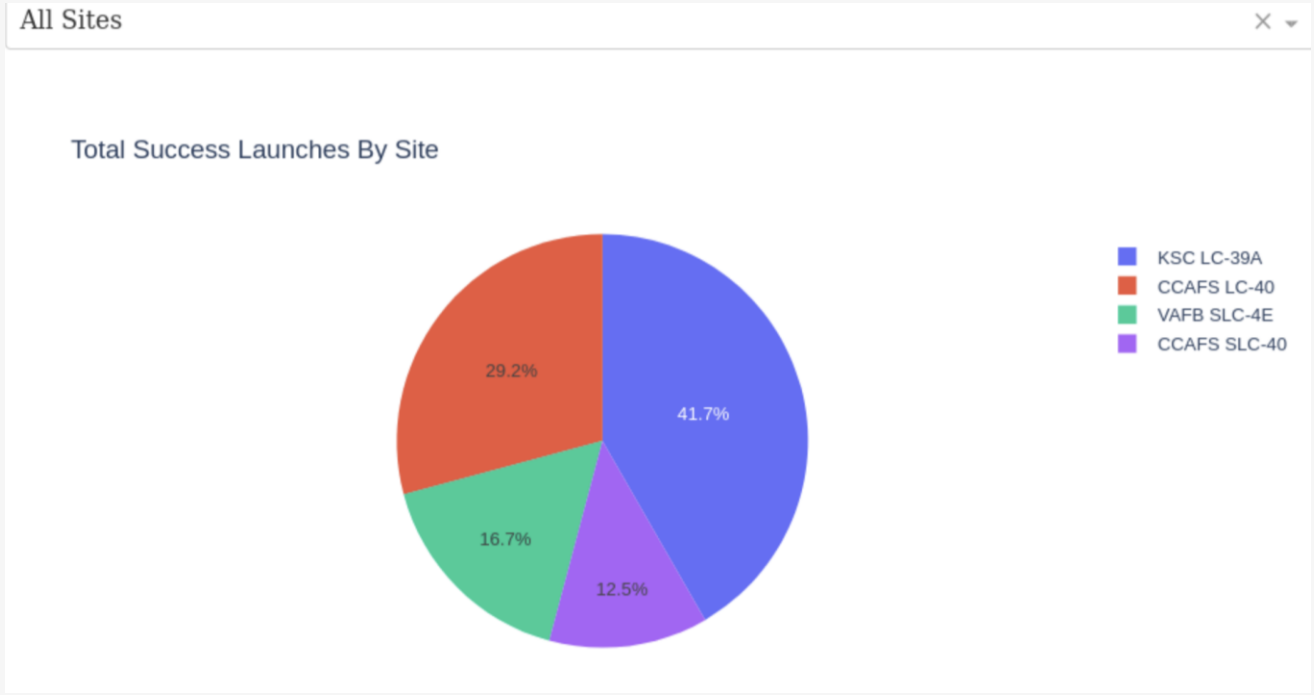
- As the user hovers over and clicks a specific launch site, details are revealed about the number of launches and their success or failure.

# CCAFS LC 40 proximity to highway, railway, and city

- Using distance equation, MarkerCluster(), and dictionaries, we plotted lines marking the distance from each civilian area to the said launch site.




Section 4
Build a Dashboard with Plotly Dash

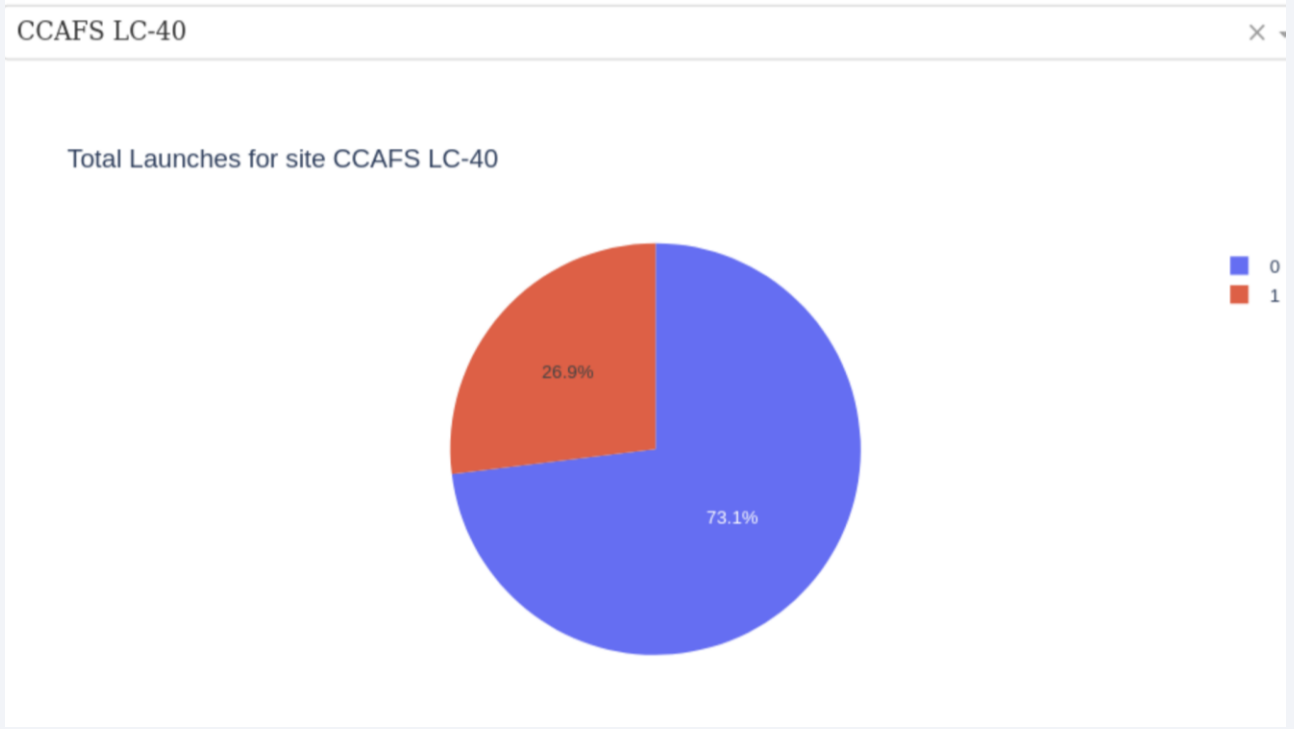



# SPACEX Launch Records Dashboard

- KSC LC-39A has reported the most successful launches

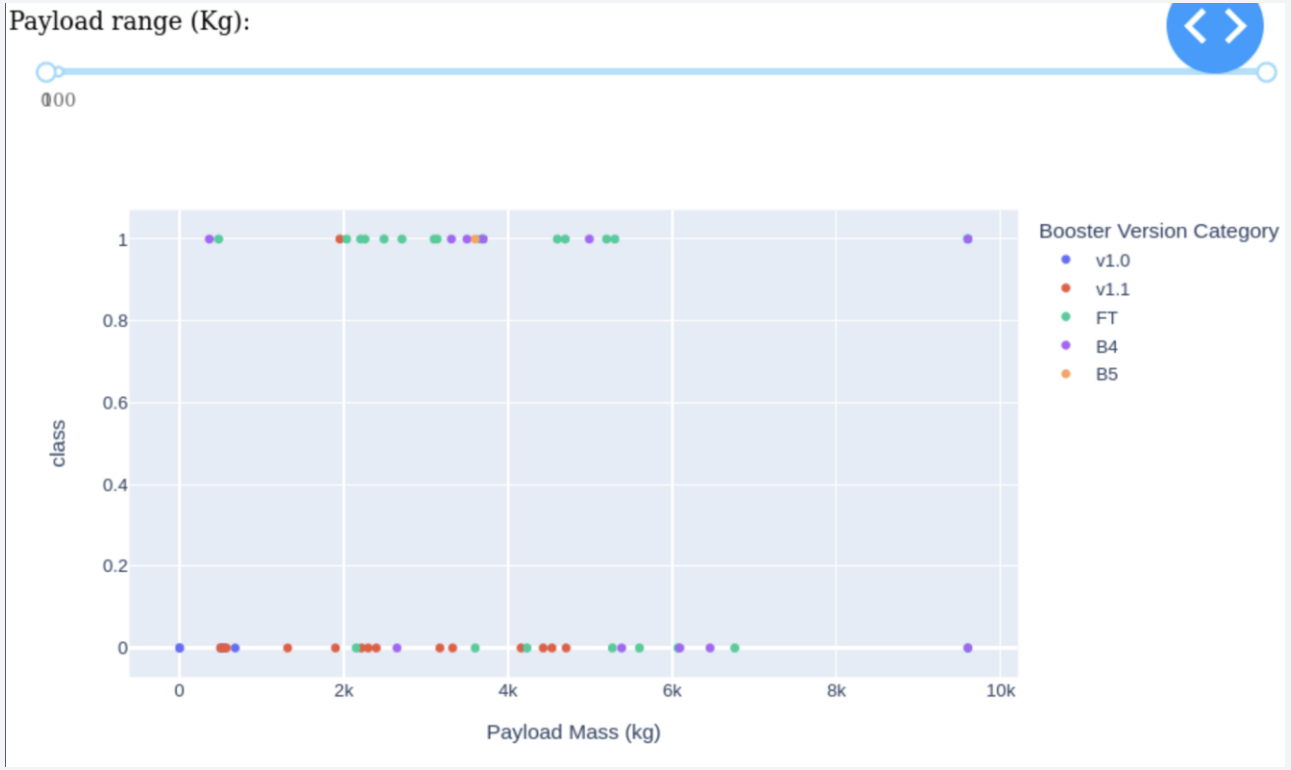
# CCAFS SLC 40 Success Rate

- Most launches in our dataset are from CCAFS SLC 40, and majority were unsuccessful



# <Payload Range Slider >

- Users can alter the payload mass range and view which outcomes are yielded.



Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

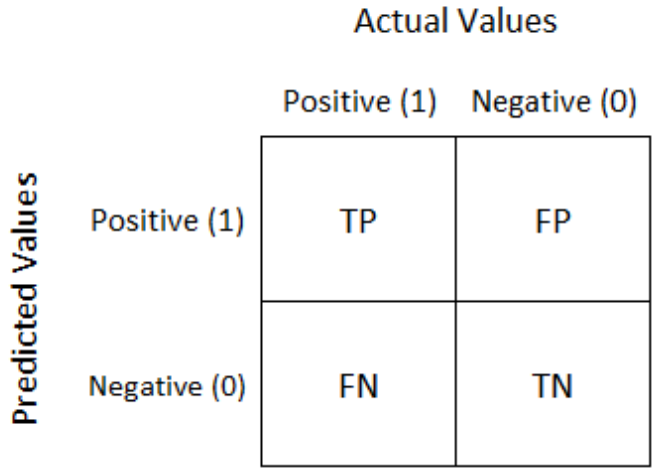
- The Jaccard Score was yielding the same accuracy, so we must rely on the best_score_ from our GridSeachCV().
- We can now see that Decision Tree Classification is the most accurate.

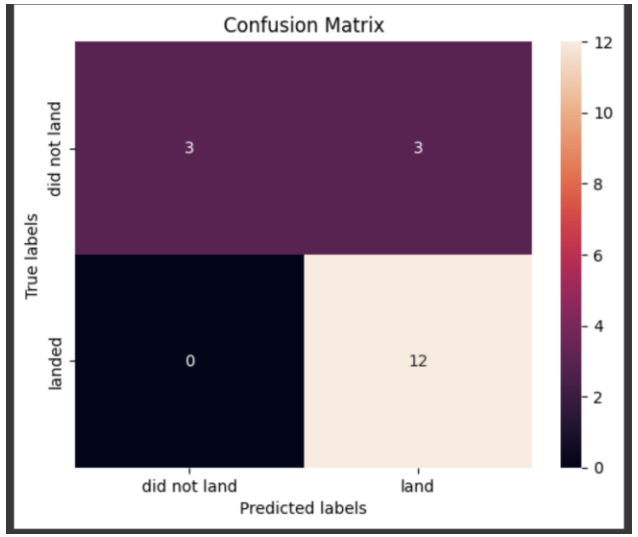
```
0s
# collect the scores
# collect the names
# dictionary
# zip loop and display
best_scores = [logreg_score,svm_score,tree_score,knn_score]
names = ["Logistic Regression","SVM","Decision Tree","KNN"]
other_dict = {"Name":names,"Score":best_scores}

for name,b_score in zip(other_dict["Name"],other_dict["Score"]) :
  print (name, b_score)
```

```
Logistic Regression 0.8464285714285713
SVM 0.8482142857142856
Decision Tree 0.8892857142857145
KNN 0.8482142857142858
```

# Confusion Matrix

- The confusion matrix from our Decision Tree model shows the most True Positives and minimal Type 1 and Type 2 errors.



# Conclusions

- The Decision Tree Classification model is the best choice for SPACEX data
- From our visualizations, we can conclude that lower payloads performed better in comparison to higher payloads
- According to our Dash Pie Charts, we can confidently say that KSC LC-39A has the most successes out of the three launch sites
- In aggregate, SPACEX saw the greatest increase in average success rate between 2015 and 2017
- SSO and VLEO orbits yield the

# Appendix

- Please find my github repository at https://github.com/anichip/IBM_Capstone_SPACEX_Data_Analysis

# Thank you!