IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

SUHAS CHALAK

1TH AUGUST 2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- Project background and context

  Space X 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 Space X 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 space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Problems you want to find answers

  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- 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

- The data was collected using various methods
  - Data collection was done using get request to the SpaceX API.
  - Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
  - We then cleaned the data, checked for missing values and fill in missing values where necessary.
  - In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

The link to the notebook is https://github.com/suhaschalak/IBM-AppliedDataScience-Capstone_FINAL/blob/main/jupyter-labs-spacex-data-collection-api%20(2).ipynb

Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.
- The link to the notebook is
- https://github.com/suhaschalak/IBM-AppliedDataScience-Capstone_FINAL/blob/main/jupyter-labs-spacex-data-collection-api%20(2).ipynb

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

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

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.\n
appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

```
# Use json_normalize meethod to convert the json result into a dataframe
data=pd.json_normalize(response.json())
```

```
# Calculate the mean value of PayloadMass column
mean=df["PayloadMass"].mean()
mean
# # Replace the np.nan values with its mean value
df["PayloadMass"].replace(np.NaN, mean, inplace=True)
df.head()
```

Data Collection - Scraping

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- The link to the notebook is
- https://github.com/suhaschalak/IBM-AppliedDataScience-Capstone_FINAL/blob/main/jupyter-labs-webscraping.ipynb

1.perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
# use requests.get() method with the provided static_url
response=requests.get(static_url)

# assign the response to a object
htmlcontent=response.content
htmlcontent
```

2.Create beautifulsoup object

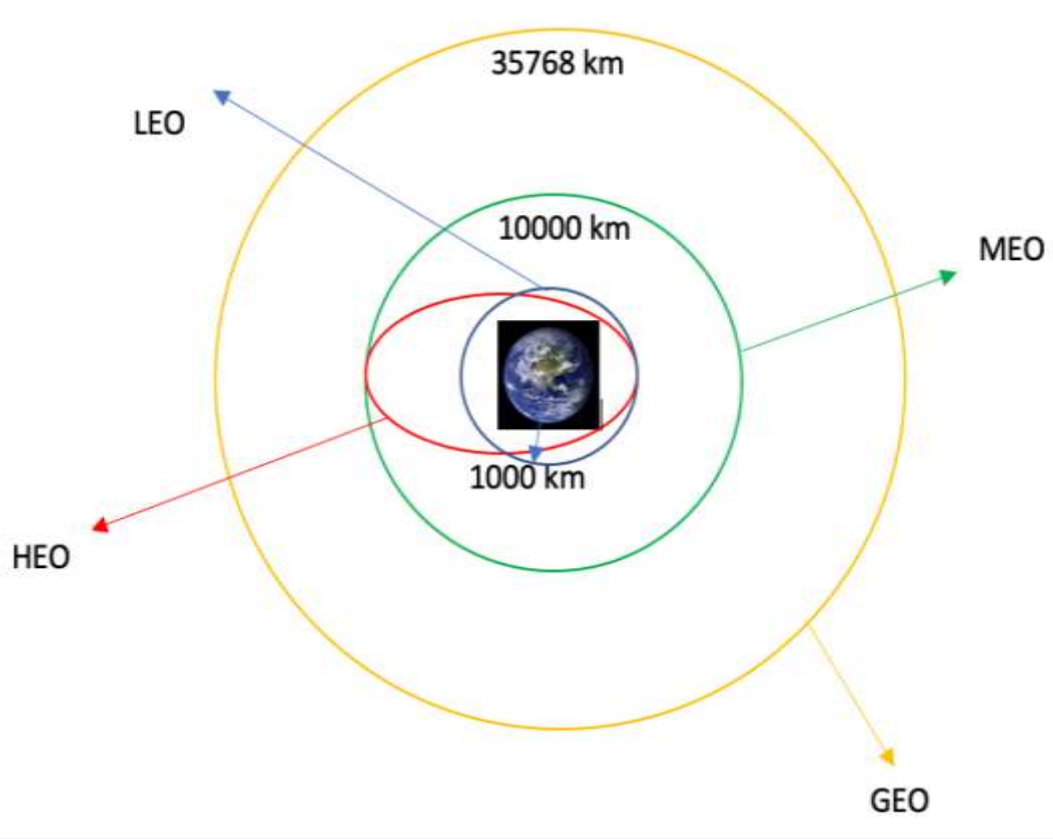
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup= BeautifulSoup(htmlcontent, "html.parser")
soup.prettify
# Use soup.title attribute
title=soup.title
title
```

3 Extract all column names from HTML table header

```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
th_elements = soup.find_all('th')

# Iterate each th element and apply the provided extract_column_from_header() to get a column name
def extract_column_from_header(th):
    # Extract text from <th> element and replace any <br> with space
    text = th.get_text(separator=" ").strip()
    return text

# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names
column_names = [extract_column_from_header(th) for th in th_elements if th.get_text().strip()]

column_names
```

4.Create a dataframe by parsing the launch HTML TABLE
5.Export data too CSV

Data Wrangling



- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- The link to the notebook is
- https://github.com/suhaschalak/IBM-AppliedDataScience-Capstone_FINAL/blob/main/labs-jupyter-spacex-Data%20wrangling%20(1).ipynb

EDA with Data Visualization

- **As part of the Exploratory Data Analysis (EDA), following charts were plotted to gain further insights into the dataset:**

1. **Scatter plot:**
    - Shows relationship or correlation between two variables making patterns easy to observe
    - Plotted following charts to visualize:
    - Relationship between Flight Number and Launch Site
    - Relationship between Payload and Launch Site
    - Relationship between Flight Number and Orbit Type
    - Relationship between Payload and Orbit Type
2. **Bar Chart:**
    - Commonly used to compare the values of a variable at a given point in time. Bar charts makes it easy to see which groups are highest/common and how other groups compare against each other. Length of each bar is proportional to the value of the items that it represents
    - Plotted following Bar chart to visualize:
    - Relationship between success rate of each orbit type
3. **Line Chart:**
    - Commonly used to track changes over a period of time. It helps depict trends over time.
    - Plotted following Line chart to observe:
        - Average launch success yearly trend

https://github.com/suhaschalak/IBM-AppliedDataScience-Capstone_FINAL/blob/main/edadataviz%20(1).ipynb

EDA with SQL

- **To better understand SpaceX data set, following SQL queries/operations were performed on an IBM DB2 cloud instance:**

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

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

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

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

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

6. List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

7. List the total number of successful and failure mission outcomes

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

9. List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year2015

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

https://github.com/suhaschalak/IBM-AppliedDataScience-Capstone_FINAL/blob/main/jupyter-labs-eda-sql-coursera_sqllite%20(1).ipynb

Build an Interactive Map with Folium

Folium interactive map helps analyze geospatial data to perform more interactive visual analytics and better understand factors such location and proximity of launch sites that impact launch success rate.

- **Following map object were created and added to the map:**
  - Mark all launch sites on the map. This allowed to visually see the launch sites on the map.
    - Added 'folium.circle' and 'folium.marker' to highlight circle area with a text label over each launch site.
  - Added a 'MarkerCluster()' to show launch success (green) and failure (red) markers for each launch site.
  - Calculated distances between a launch site to its proximities (e.g., coastline, railroad, highway, city)
    - Added 'MousePosition() to get coordinate for a mouse position over a point on the map
    - Added 'folium.Marker()' to display distance (in KM) on the point on the map (e.g., coastline, railroad, highway, city)
    - Added 'folium.Polyline()' to draw a line between the point on the map and the launch site
    - Repeated steps above to add markers and draw lines between launch sites and proximities – coastline, railroad, highway, city)
- **Building the Interactive Map with Folium helped answered following questions:**
  - Are launch sites in close proximity to railways? YES
  - Are launch sites in close proximity to highways? YES
  - Are launch sites in close proximity to coastline? YES
  - Do launch sites keep certain distance away from cities? YES

https://github.com/suhaschalak/IBM-AppliedDataScience-Capstone_FINAL/blob/main/lab_jupyter_launch_site_location%20(1).ipynb

Build a Dashboard with Plotly Dash

- **Built a Plotly Dash web application to perform interactive visual analytics on SpaceX launch data in real-time. Added Launch Site Drop-down, Pie Chart, Payload range slide, and a Scatter chart to the Dashboard.**

1. Added a Launch Site Drop-down Input component to the dashboard to provide an ability to filter Dashboard visual by all launch sites or a particular launch site

2. Added a Pie Chart to the Dashboard to show total success launches when 'All Sites' is selected and show success and failed counts when a particular site is selected

3. Added a Payload range slider to the Dashboard to easily select different payload ranges to identify visual patterns

4. Added a Scatter chart to observe how payload may be correlated with mission outcomes for selected site(s). The color-label Booster version on each scatter point provided missions outcomes with different boosters

- **Dashboard helped answer following questions:**

1. Which site has the largest successful launches? KSC LC-39A with 10

2. Which site has the highest launch success rate? KSC LC-39A with 76.9% success

3. Which payload range(s) has the highest launch success rate? 2000 – 5000 kg

4. Which payload range(s) has the lowest launch success rate? 0-2000 and 5500 - 7000

5. Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest launch success rate? FT

https://github.com/suhaschalak/IBM-AppliedDataScience-Capstone_FINAL/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

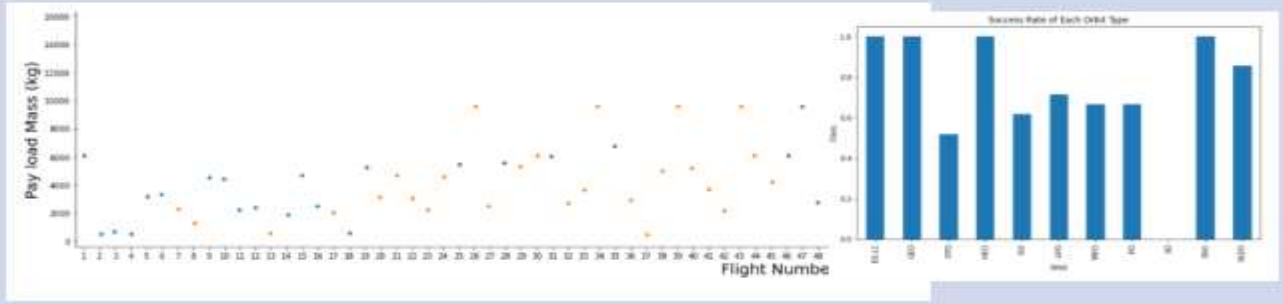
- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- The link to the notebook is
- https://github.com/suhaschalak/IBM-AppliedDataScience-Capstone_FINAL/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

Following sections and slides explain results for:

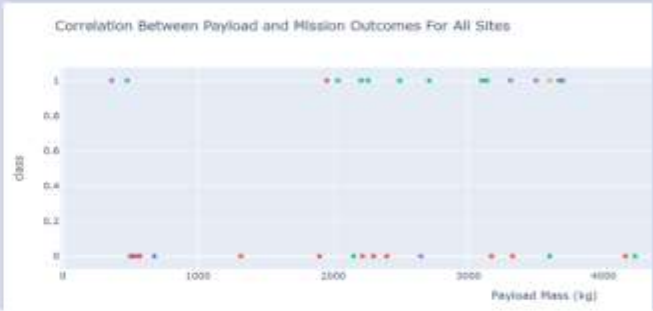
Exploratory data analysis results

- Samples:



Interactive analytics demo in screenshots

- Samples



Predictive analysis results

- Samples

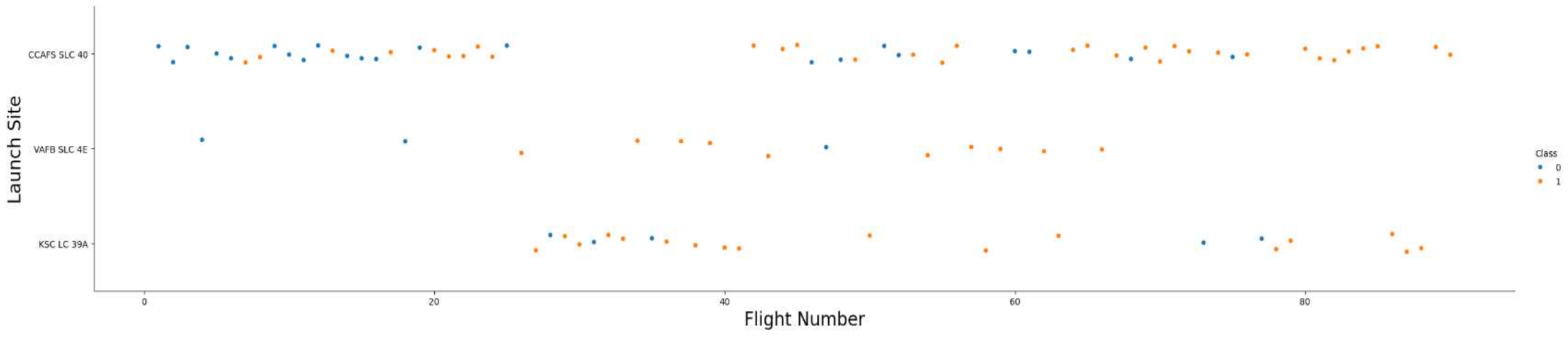
	Algo Type	Accuracy Score
2	Decision Tree	0.903571
3	KNN	0.848214
1	SVM	0.848214
0	Logistic Regression	0.846429

Section 2

Insights drawn from EDA

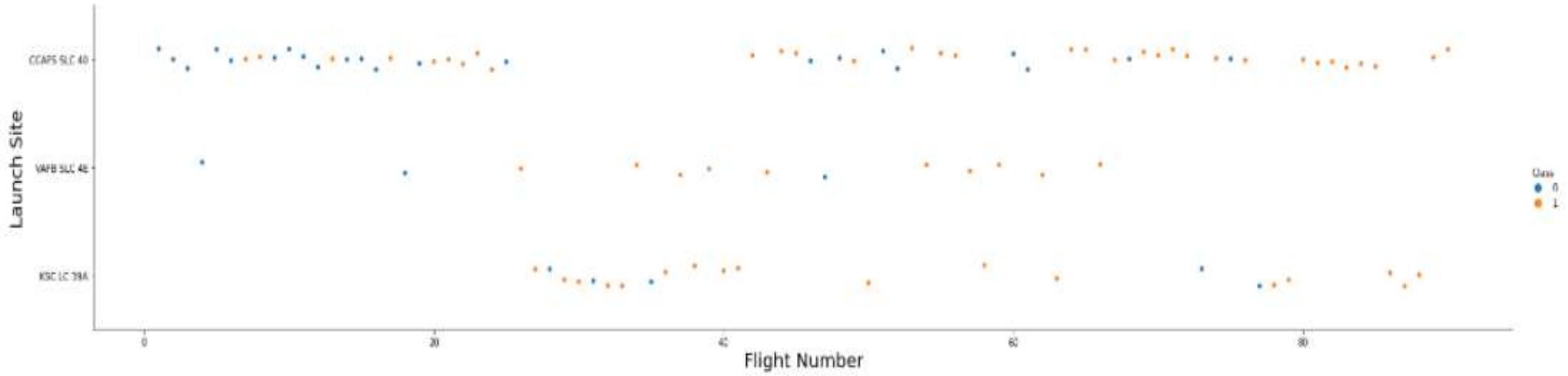
Flight Number vs. Launch Site

- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.



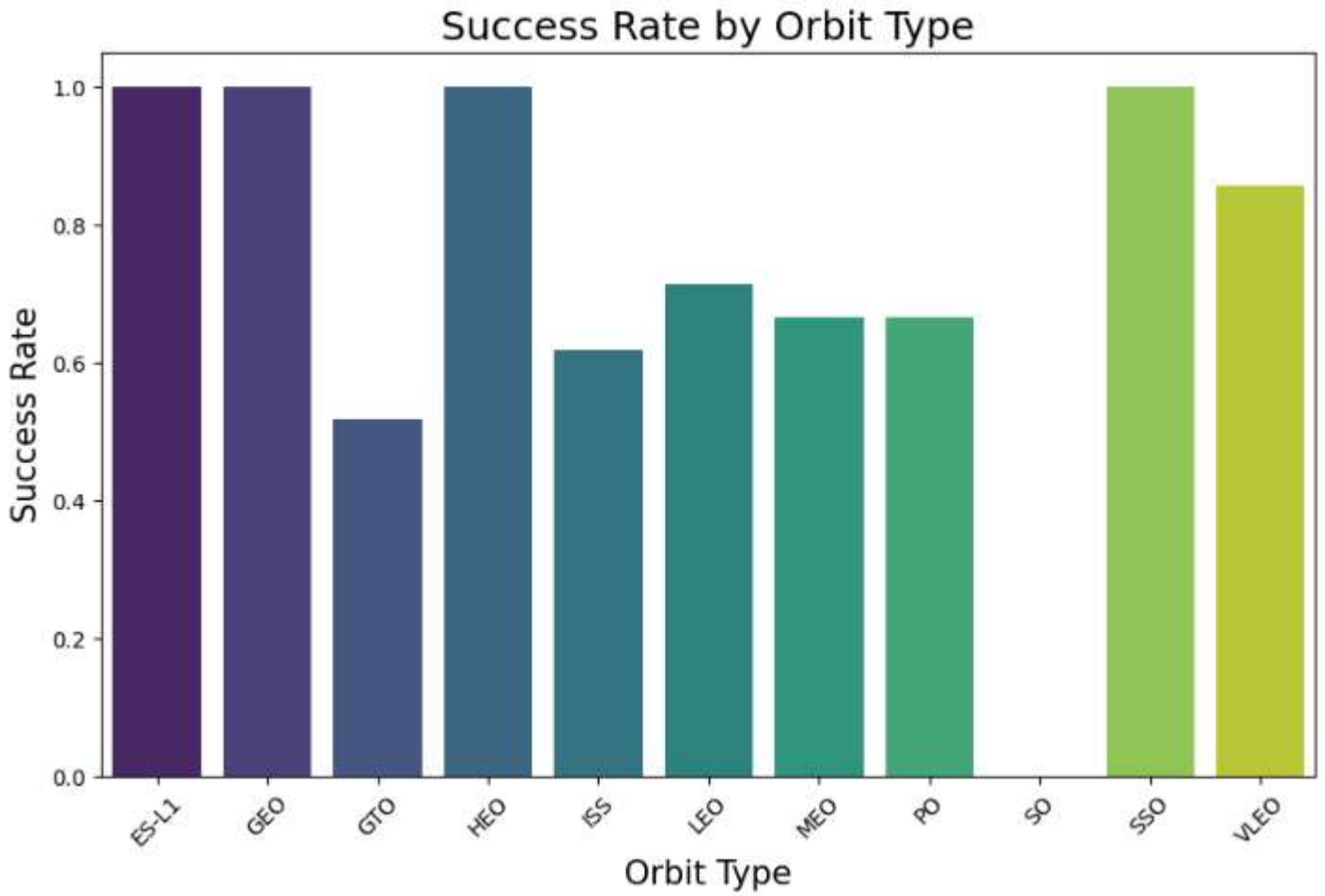
Payload vs. Launch Site

The greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket.



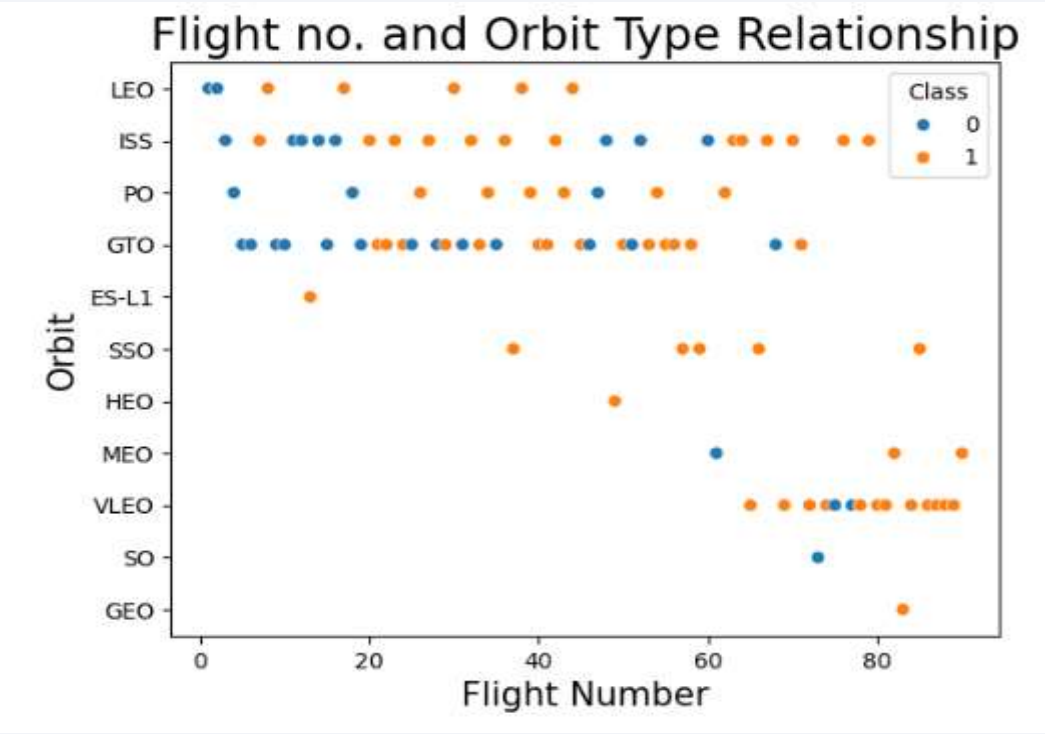
Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.



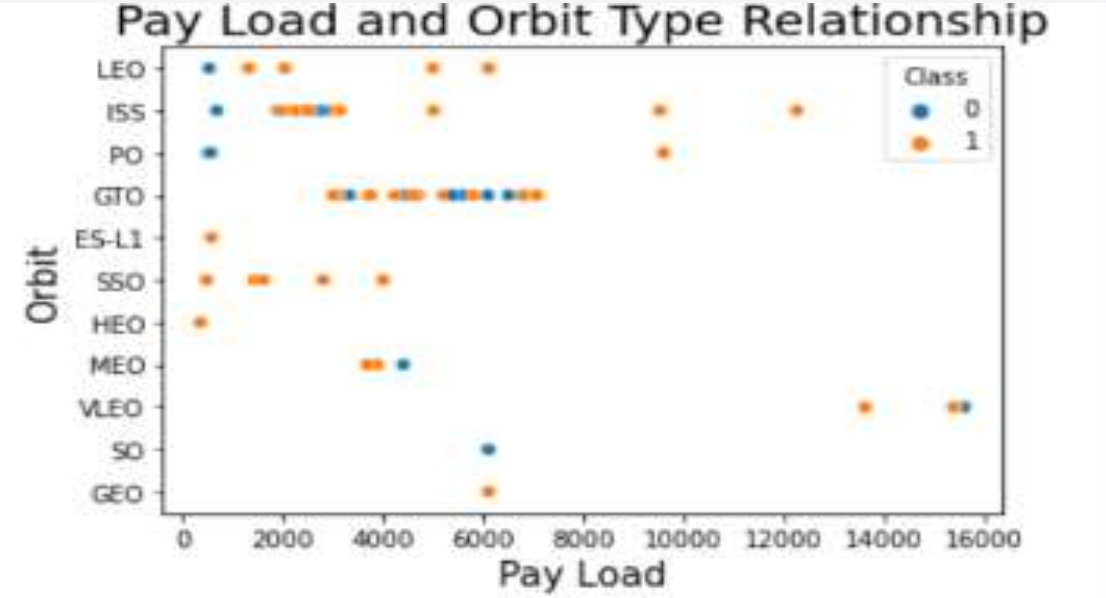
Flight Number vs. Orbit Type

- The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



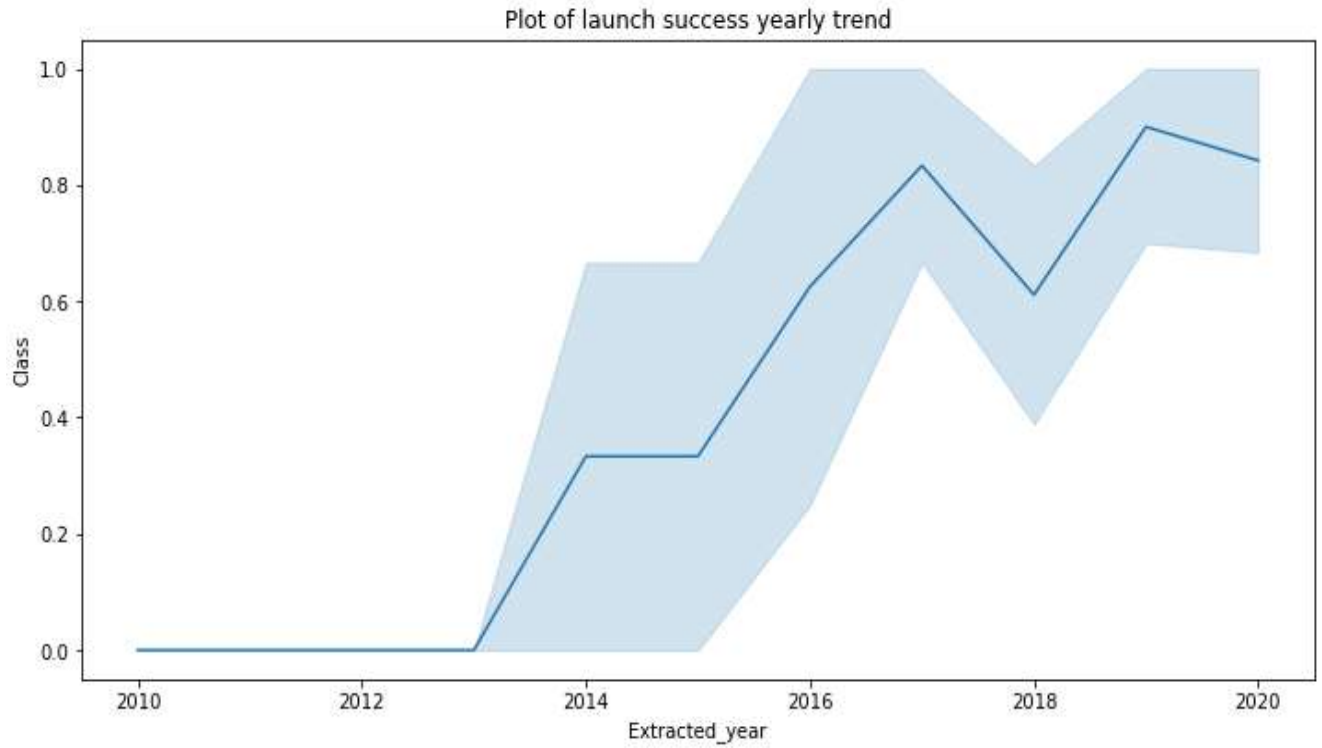
Payload vs. Orbit Type

- We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



Launch Success Yearly Trend

- From the plot, we can observe that success rate since 2013 kept on increasing till 2020.



All Launch Site Names

- We used the key word **DISTINCT** to show only unique launch sites from the SpaceX data.

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

```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL
```

```
 * 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'

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

- We used the query above to display 5 records where launch sites begin with `CCA`

Total Payload Mass

- We calculated the total payload carried by boosters from NASA as **48213** using the query below

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

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_)
FROM SPACEXTBL
WHERE Customer LIKE '%NASA (CRS)%'
```

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

SUM(PAYLOAD_MASS__KG_)
48213

Average Payload Mass by F9 v1.1

- We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

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'
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- **We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015**

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)
2015-12-22

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 Booster_Version
FROM SPACEXTBL
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

- We used the **WHERE** clause to filter for boosters which have successfully landed on drone ship and applied the **AND** condition to determine successful landing with payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%%sql
SELECT Mission_Outcome,COUNT(*)
FROM SPACEXTBL
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

- **The ‘group by’ keyword arranges identical data in a column in to group**
- **• In this case, number of mission outcomes by types of outcomes are grouped in column ‘counts’**

Boosters Carried Maximum Payload

- We determined the booster that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX()** function.

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

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

```
 * 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

- We used a combinations of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

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.

```
%%sql
SELECT
     CASE substr(Date, 6, 2)
          WHEN '01' THEN 'January'
          WHEN '02' THEN 'February'
          WHEN '03' THEN 'March'
          WHEN '04' THEN 'April'
          WHEN '05' THEN 'May'
          WHEN '06' THEN 'June'
          WHEN '07' THEN 'July'
          WHEN '08' THEN 'August'
          WHEN '09' THEN 'September'
          WHEN '10' THEN 'October'
          WHEN '11' THEN 'November'
          WHEN '12' THEN 'December'
     END AS MonthName,
     Booster_Version,
     Launch_Site
FROM SPACEXTBL
WHERE Landing_Outcome = 'Failure (drone ship)'
   AND substr(Date, 1, 4) = '2015';
```

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

MonthName	Booster_Version	Launch_Site
January	F9 v1.1 B1012	CCAFS LC-40
April	F9 v1.1 B1015	CCAFS LC-40

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

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(*) AS OutcomeCount
FROM SPACEXTBL
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY OutcomeCount DESC;
```

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

Landing_Outcome	OutcomeCount
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

- We selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2010-03-20.
- We applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.

Section 4

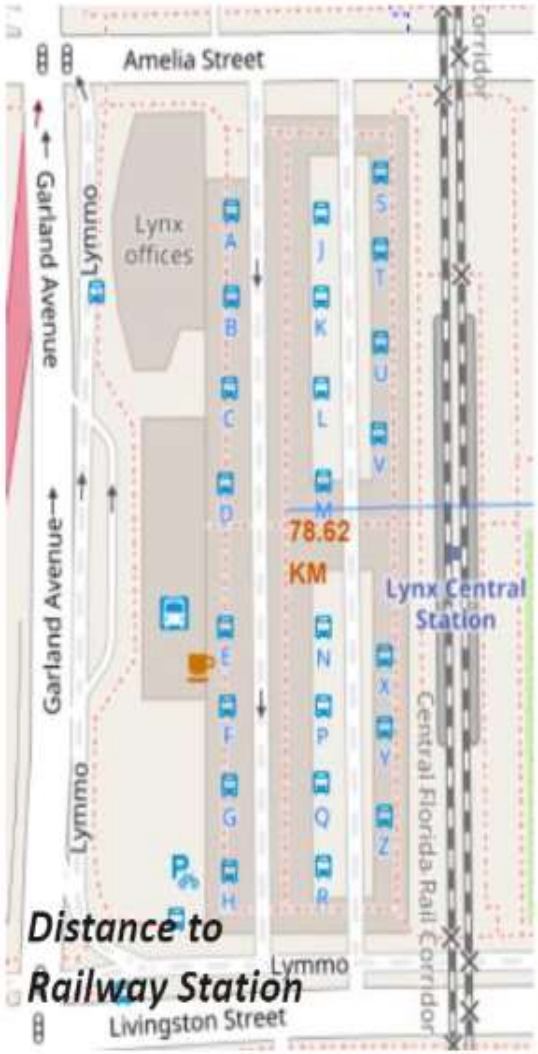
Launch Sites Proximities Analysis

All launch sites global map markers

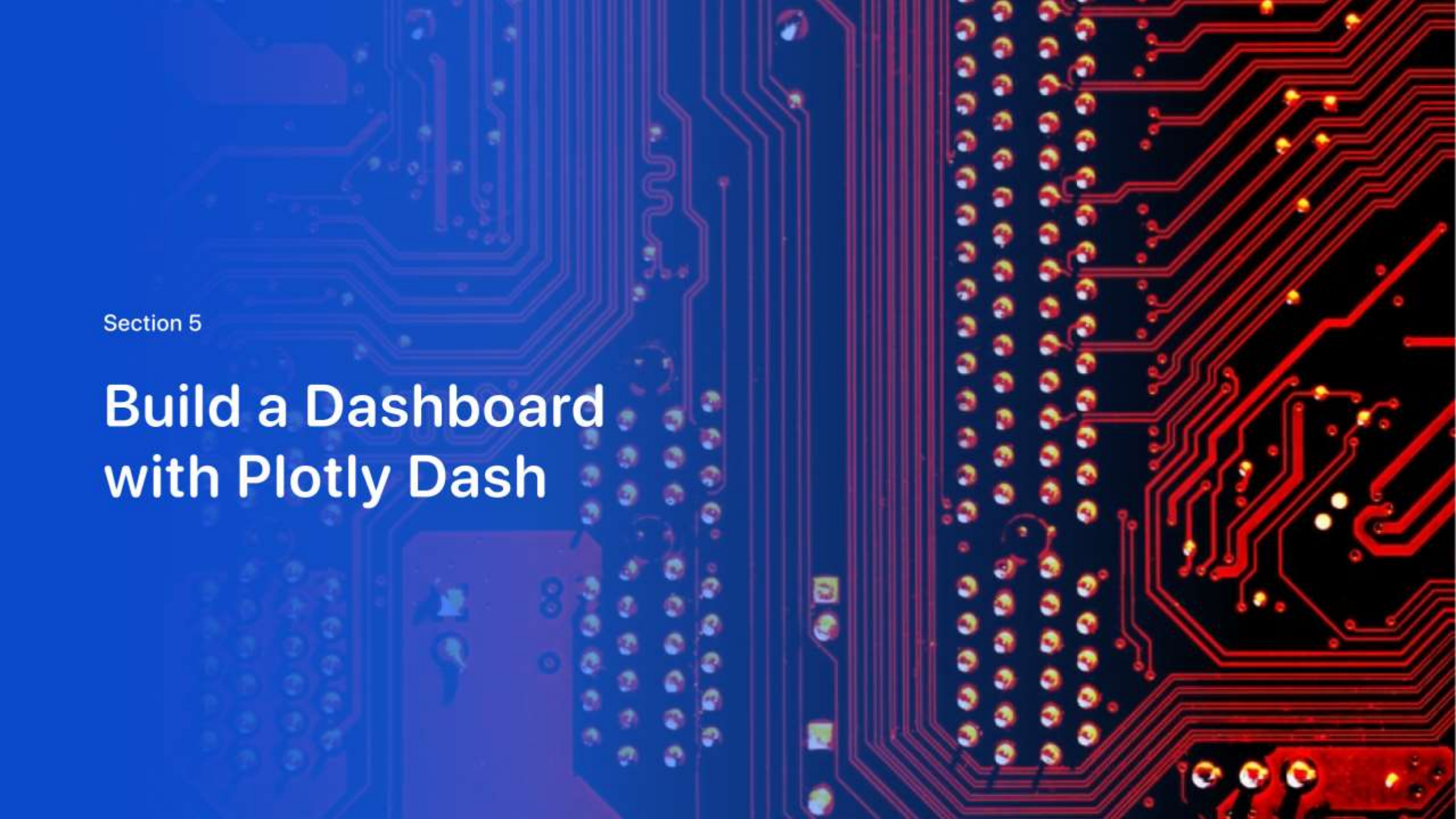
We can see that the SpaceX launch sites are in the United States of America coasts. Florida and California

Markers showing launch sites with color labels

Launch Site distance to landmarks



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

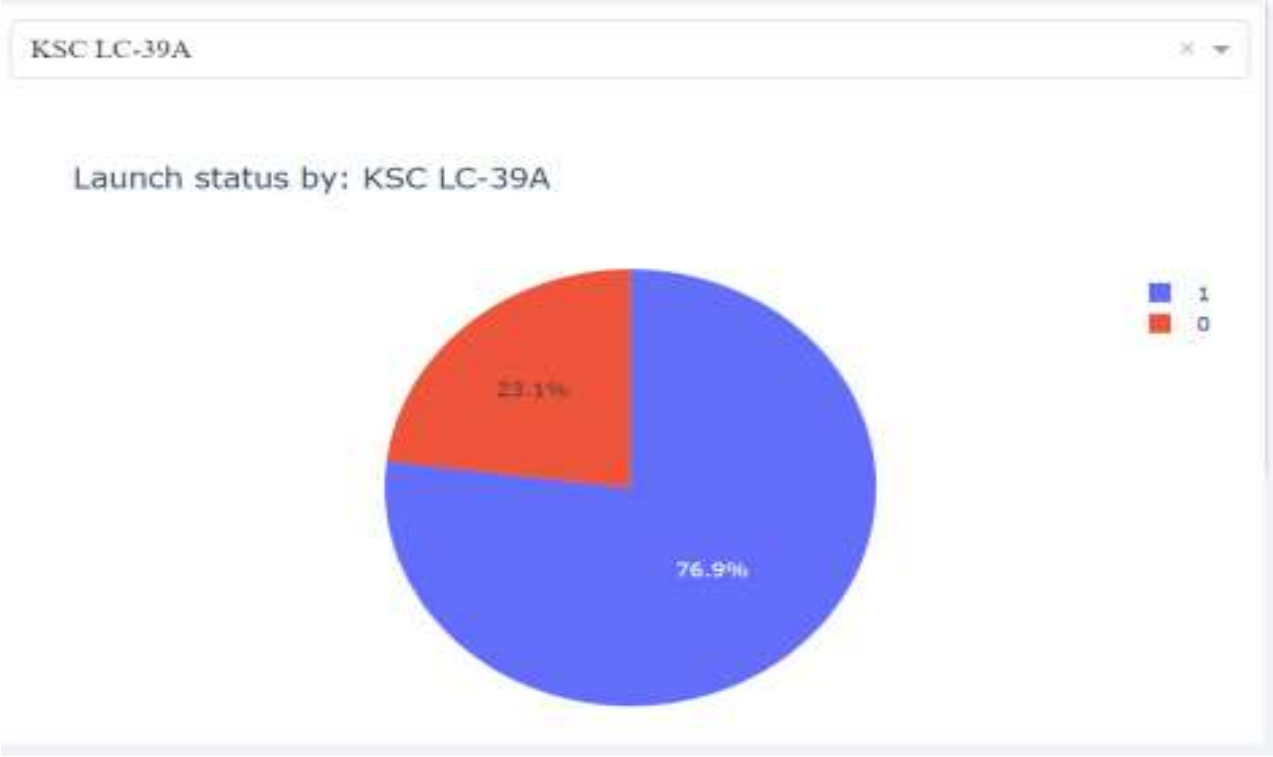

Section 5
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site

- **Launch Site ‘KSC LC-39A’ has the highest launch success rate**
- **Launch Site ‘CCAFS SLC40’ has the lowest launch success rate**

Pie chart showing the Launch site with the highest launch success ratio

- **KSC LC-39A Launch Site has the highest launch success rate and count**
- **Launch success rate is 76.9%**
- **Launch success failure rate is 23.1%**



Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider

- **Most successful launches are in the payload range from 2000 to about 5500**
- **Booster version category ‘FT’ has the most successful launches**
- **Only booster with a success launch when payload is greater than 6k is ‘B4’**

Section 6

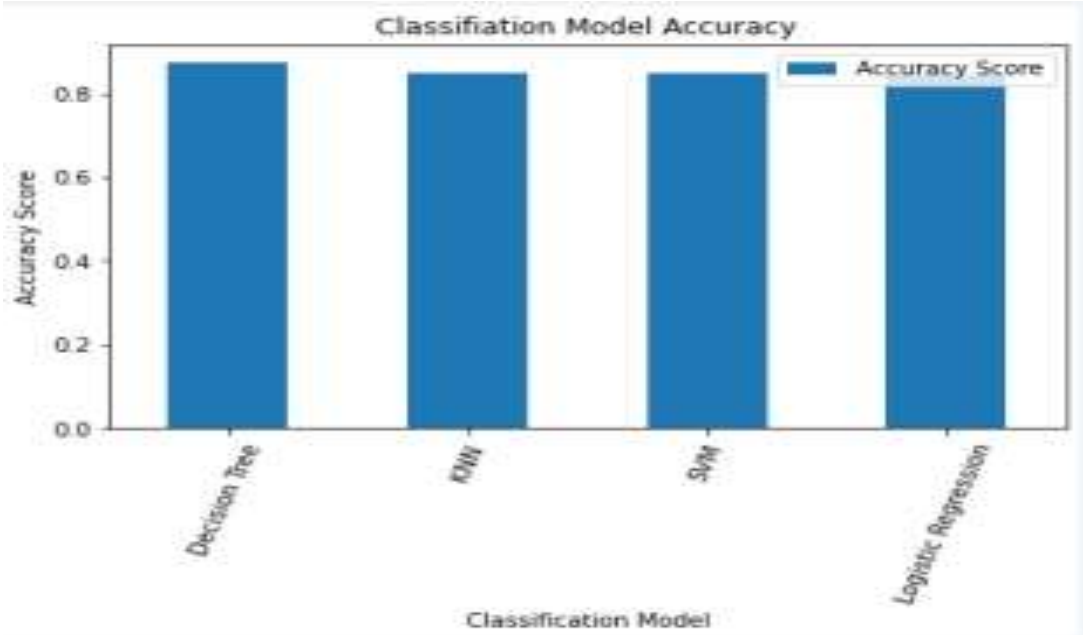
Predictive Analysis (Classification)

Classification Accuracy

- Based on the Accuracy scores and as also evident from the bar chart, Decision Tree algorithm has the highest classification score with a value of .8750

• Accuracy Score on the test data is the same for all the classification algorithms based on the data set with a value of .8333

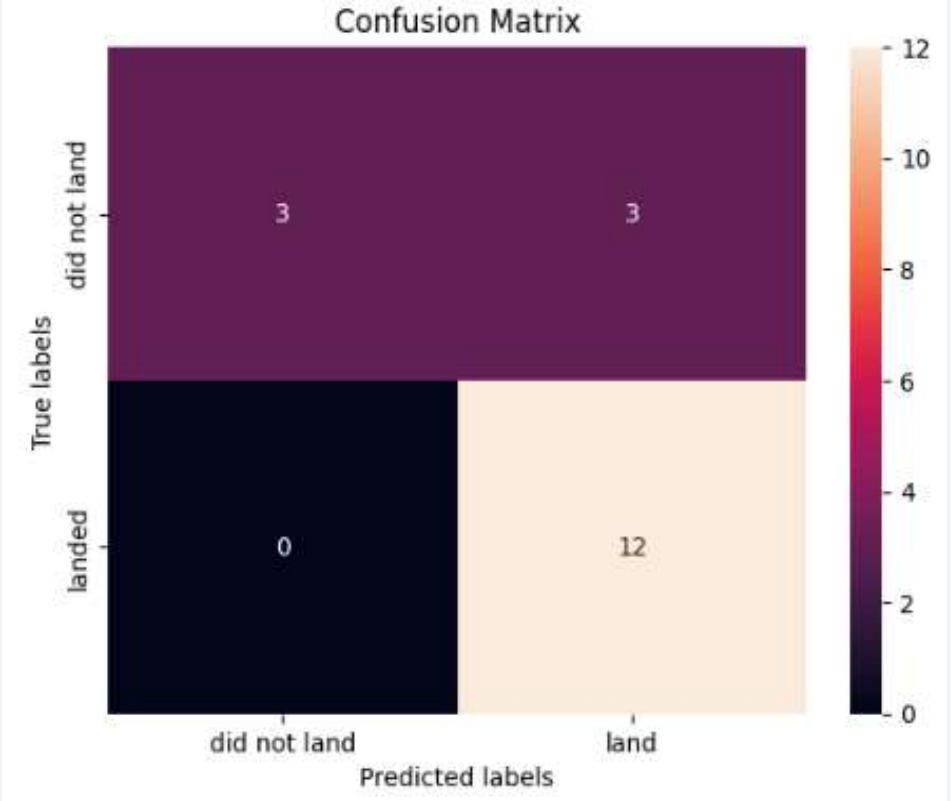
• Given that the Accuracy scores for Classfication algorithms are very close and the test scores are the same, we may need a broader data set to further tune the models



	Algo Type	Accuracy Score	Test Data Accuracy Score
2	Decision Tree	0.875000	0.833333
3	KNN	0.848214	0.833333
1	SVM	0.848214	0.833333
0	Logistic Regression	0.846429	0.833333

Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.



Conclusions

- As the numbers of flights increase, the first stage is more likely to land successfully
- Success rates appear go up as Payload increases but there is no clear correlation between Payload mass and success rates
- Launch success rate increased by about 80% from 2013 to 2020
- Launch Site ‘KSC LC-39A’ has the highest launch success rate and Launch Site ‘CCAFS SLC40’ has the lowest launch success rate
- Orbits ES-L1, GEO, HEO, and SSO have the highest launch success rates and orbit GTO the lowest
- Lunch sites are located strategically away from the cities and closer to coastline, railroads, and highways
- The best performing Machine Learning Classfication Model is the Decision Tree with an accuracy of about 87.5%. When the models were scored on the test data, the accuracy score was about 83% for all models. More data may be needed to further tune the models and find a potential better fit.

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!