Winning Space Race with Data Science

Chidananda Rao Ramiah

07/03/2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection
  - Data Wrangling
  - Data Visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Building a dashboard with Plotly Dash
  - Predictive Analysis
- Summary of all results
  - Exploratory data analysis results
  - Interactive analytics demo
  - Predictive Analysis Results

Introduction

- Project background and context

Commercial space travel has become more feasible in recent years due to advances in technology and the increasing involvement of private companies in space activities. Companies such as SpaceX have developed reusable rockets and spacecraft, which have reduced the cost of spaceflight and made it more accessible to a wider range of customers.

Our goal is to predict if SpaceX's Falcon 9 first stage will land successfully. If we can determine if the first stage will land, we can determine the cost of a launch which should lead to valuable data during bidding wars between space travel companies.

- Problems you want to find answers
  - Correlation between the rocket's variables and the successful landing rate
  - Conditions to get optimal results and maximum successful landing rate

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API and Web Scraping from Wikipedia
- Perform data wrangling
  - Outcomes were converted into labels for training.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Finding the best hyperparameter for SVM, Classification Trees and Logistic Regression

Data Collection

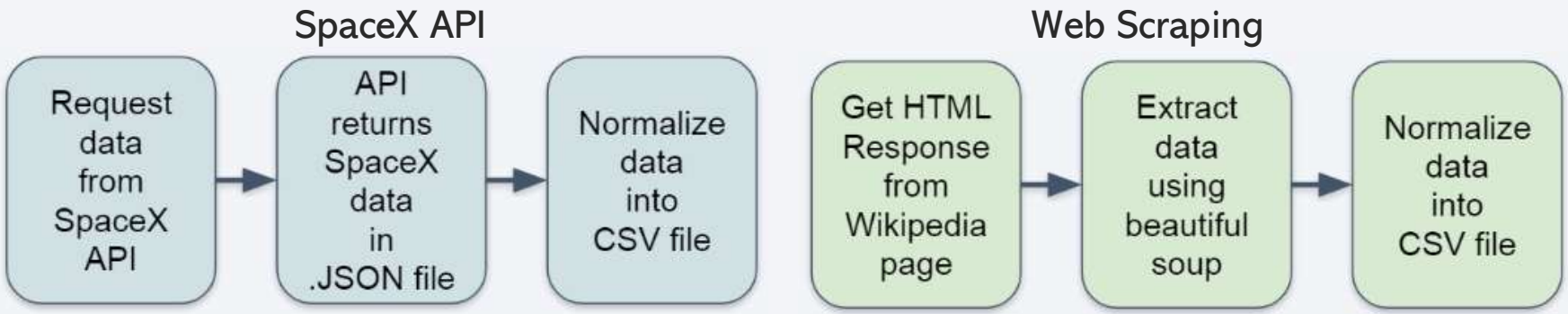
- The data collection process includes several API requests from the SpaceX API and web scraping data from the Wikipedia page of the Falcon 9 launch records.

- SpaceX API Data Columns:

FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude.

- Wikipedia Web Scrape Data Columns:

Flight No., Launch Site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time.



Data Collection – SpaceX API

1. Request rocket launch data from SpaceX API.

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

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

2. Converting response into a JSON file.

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

3. Use custom functions to clean data.

```
getBoosterVersion(data)
```

```
getLaunchSite(data)
```

```
getPayloadData(data)
```

```
getCoreData(data)
```

[GitHub Link]

4. Combining columns into a dictionary to create a data frame.

```
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}
```

5. Filtering dataframe and exporting to CSV

```
data_falcon9 = launch_df[launch_df['BoosterVersion'] == 'Falcon 9']
```

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection – Scraping

1. Getting response from HTML.

```
html_data = requests.get(static_url).text
```

2. Converting response into a JSON file.

```
soup = BeautifulSoup(html_data, 'lxml')
```

3. Finding all tables and assigning the result to a list.

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

4. Extracting columns one by one.

```
column_names = []

for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if name != None and len(name) > 0:
        column_names.append(name)
```

5. Creating an empty dictionary with keys.

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6. Creating a dataframe and exporting it to CSV.

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

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


GitHub Link

Data Wrangling

- There are several cases in cases in which the booster failed to successfully land on the dataset, and sometimes it attempted to land but failed because of accident.
  - True Ocean: the mission result has successfully landed in a specific area of the ocean.
  - False Ocean: the mission result has not successfully landed in a specific area of the ocean.
  - True RTLS: the mission result has successfully landed on the ground pad.
  - False RTLS: the mission result has not successfully landed on the ground pad.
  - True ASDS: the mission result has successfully landed on the drone ship.
  - False ASDS: the mission result has not successfully landed on the drone ship.
- Converting these results into training labels:
  - 1= successful/ 0 = failure

GitHub Link

Data Wrangling

1. Calculating the number of launches per site

```
df['LaunchSite'].value_counts()
```

2. Calculating the number and occurrence of each orbit.

```
df['Orbit'].value_counts()
```

3. Calculating the number and occurrence of mission outcome per orbit type.

```
landing_outcomes = df['Outcome'].value_counts()
```

4. Creating a landing outcome label from Outcome column.

```
landing_class = []
for outcome in df.Outcome:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

5. Calculating the success rate for every landing in the dataset.

```
df["Class"].mean()
```

```
0.6666666666666666
```

6. Exporting dataset to CSV.

```
df.to_csv("dataset_part_2.csv", index=False)
```

GitHub Link

Data Visualization

- Scatter chart:
  - Flight Number vs. Launch Site
  - Payload vs. Launch Site
  - Flight Number vs. Orbit Type
  - Payload vs. Orbit Type
- Bar Chart:
  - Orbit Type vs. Success Rate
  - A bar chart makes the dataset comparison more explicit. One axis represents a categorical value and the other axis represents a discrete value. The goal of the chart is to indicate the relationship between the 2 axes.
- Line Chart:
  - Year vs. Success Rate
  - A Line chart shows data variables an trends very clearly and helps predict the results of data that is yet to be recorded.

GitHub Link

EDA with SQL

- Loading the dataset into the corresponding table in a Db2 database and executing SQL queries to answer the following questions.
  - Displaying the names of unique launch sites in the space missions.
  - Displaying 5 records where launch sites begin with 'CCA'.
  - Displaying the total payload mass carried by boosted launched by NASA.
  - Displaying the average payload mass carried by boosted version F9 v1.1.
  - Listing 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.
  - Listing the total number of successful and failure mission outcomes.
  - Listing the names of the booster versions which have carried the maximum payload mass
  - Listing the failed landing outcomes in drone ships, their booster versions, and launch site names in 2015.
  - Ranking the count of landing outcomes (such as Failure (drone ship) or Success(ground pad)) between the dates 2010-06-04 and 2017-03-20, in descending order.

GitHub Link

Build an Interactive Map with Folium

- Objects created and placed on the Folium Map.
  - Markers that show all launch sites on a map.
  - Markers that show the success/failed launches for each site on the map.
  - Lines that show the distances between a launch site to its proximities.
- By adding these objects, following geographical patterns about launch sites and found:
  - Are launch sites in close proximity to railway systems? Yes
  - Are launch sites in close proximity to highways? Yes
  - Are launch sites in close proximity to the coastline? Yes
  - Do launch sites keep certain distance away from cities? Yes

Interactive Maps URL

Build a Dashboard with Plotly Dash

- The dashboard application contains a pie chart and a scatter plot chart.

  - Pie Chart:

    Shows total number of successful launches by sites.

    Indicates the successful landing distribution across all sites or to indicate the success rate of each launch site.

  - Scatter Plot

    Shows the relationship between Outcomes and Payload Mass with different boosters

    Has 2 inputs namely the launch site and payload mass

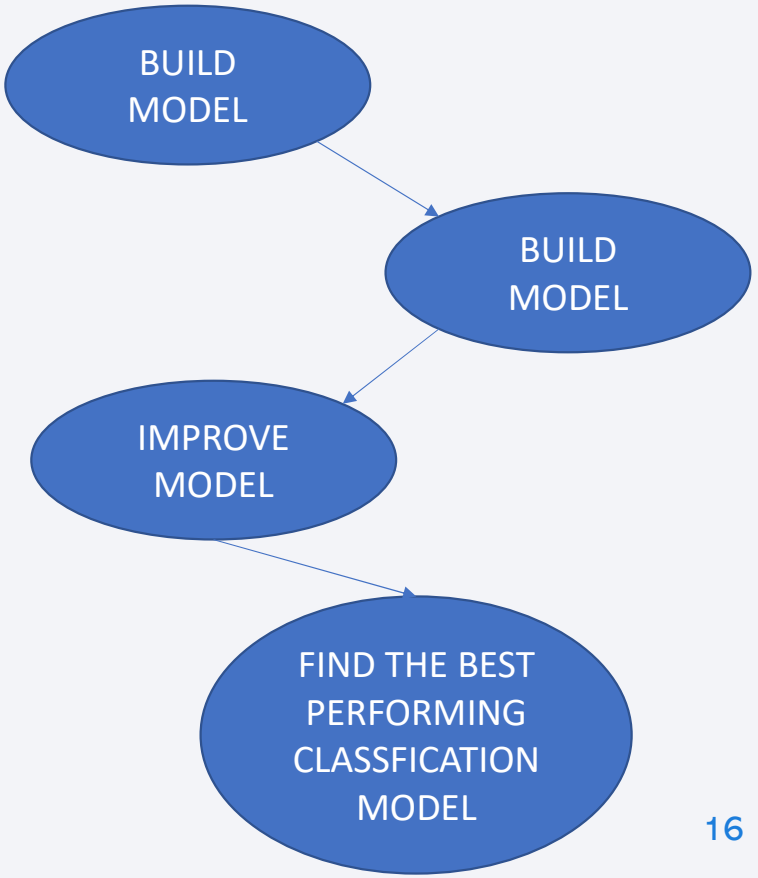
    Visualizes the dependency between payload mass, launch point and booster version

GitHub Link

Predictive Analysis (Classification)

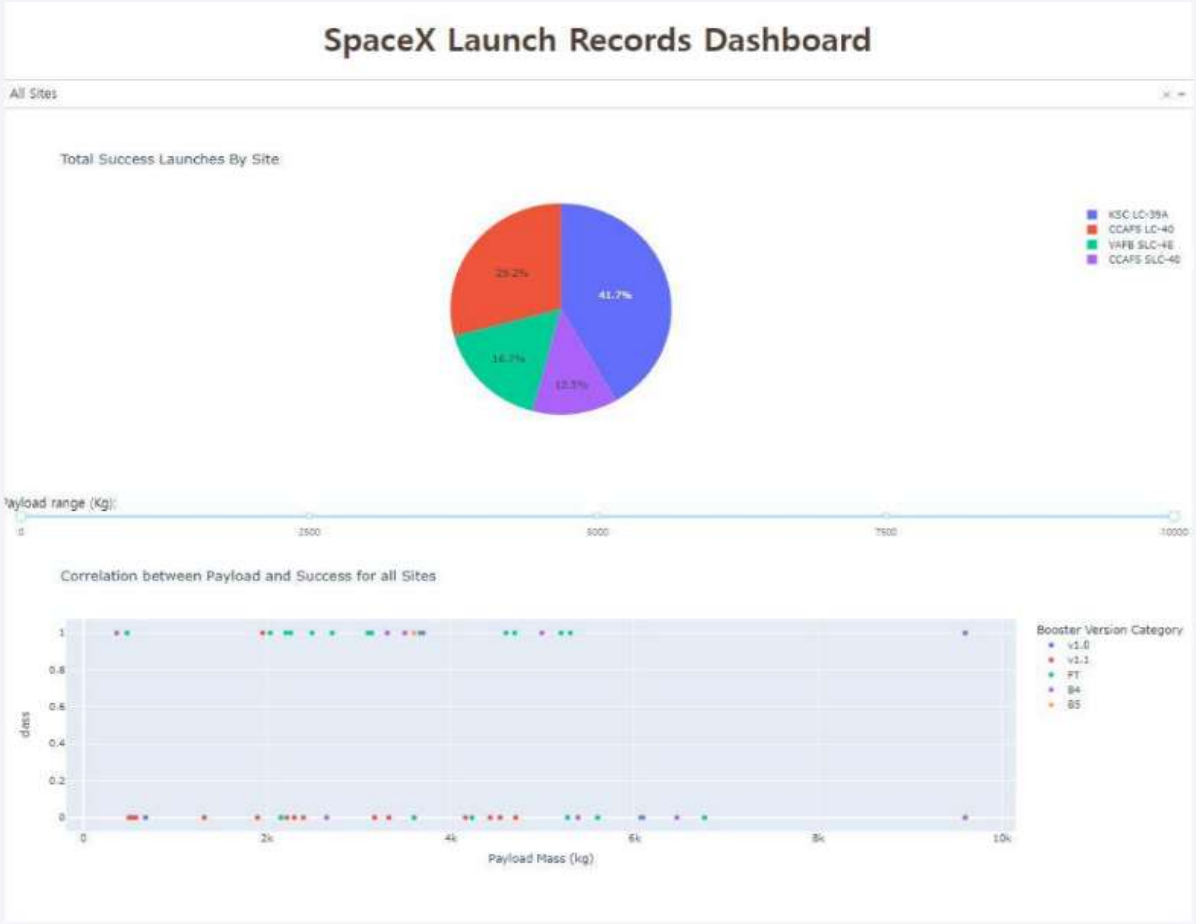
- Perform EDA (Classification) and choose training labels.
  - Separate class into column
  - Standardize the data
  - Split the data into training and test sets
- Find the best hyperparameter for SVM, Classification Trees and Logistic Regression.
  - Find the method that has the best performance using test data.

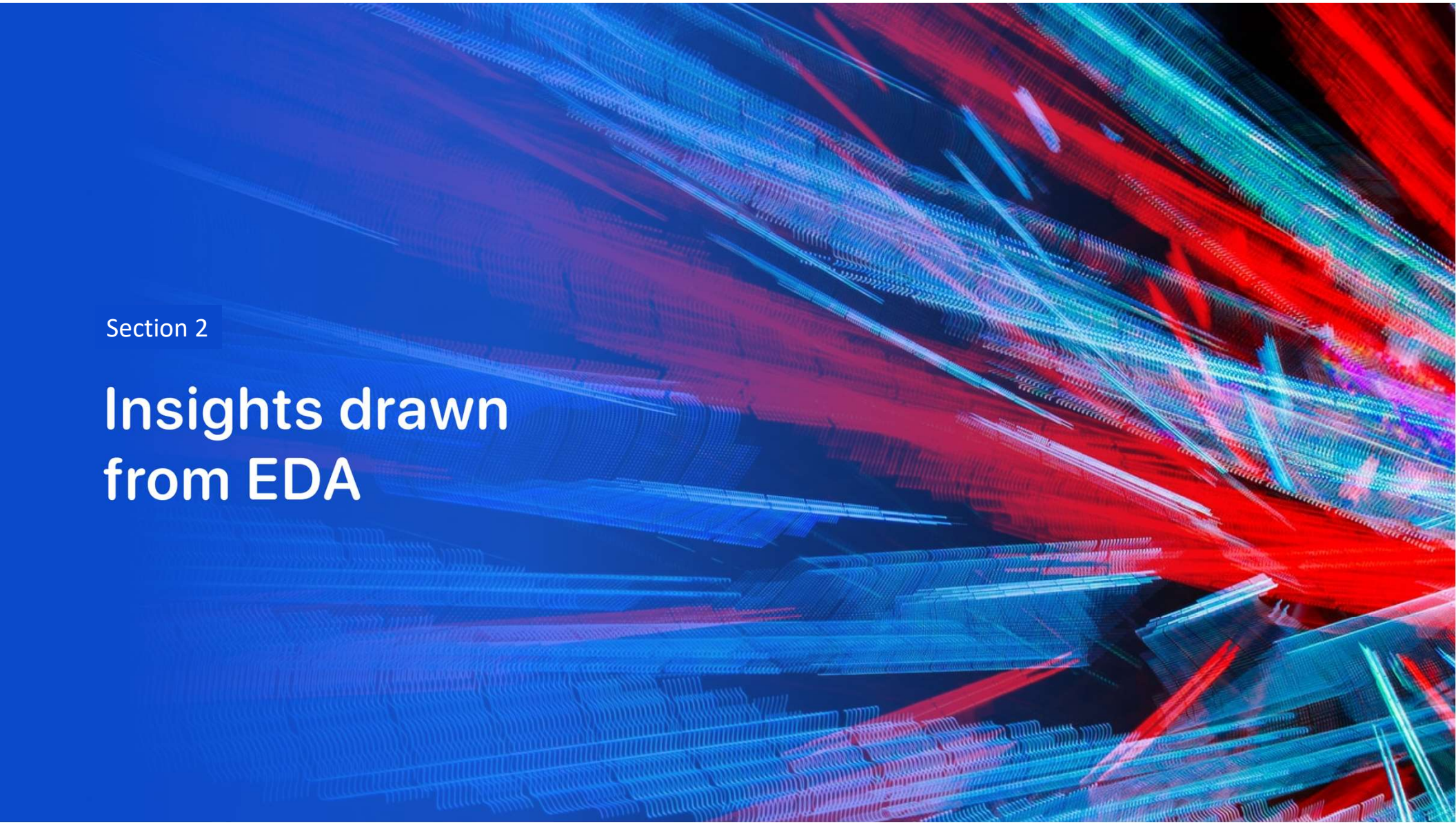
GitHub Link



Results

- The left screenshot shows the Dashboard with Plotly Dash.
- The results of EDA with visualization, EDA with SQL, Folium Map and Dashboard will be shown in the following slides.
- Comparing the accuracy of all four classification methods, all return the same accuracy of approximately 83% on the test data.



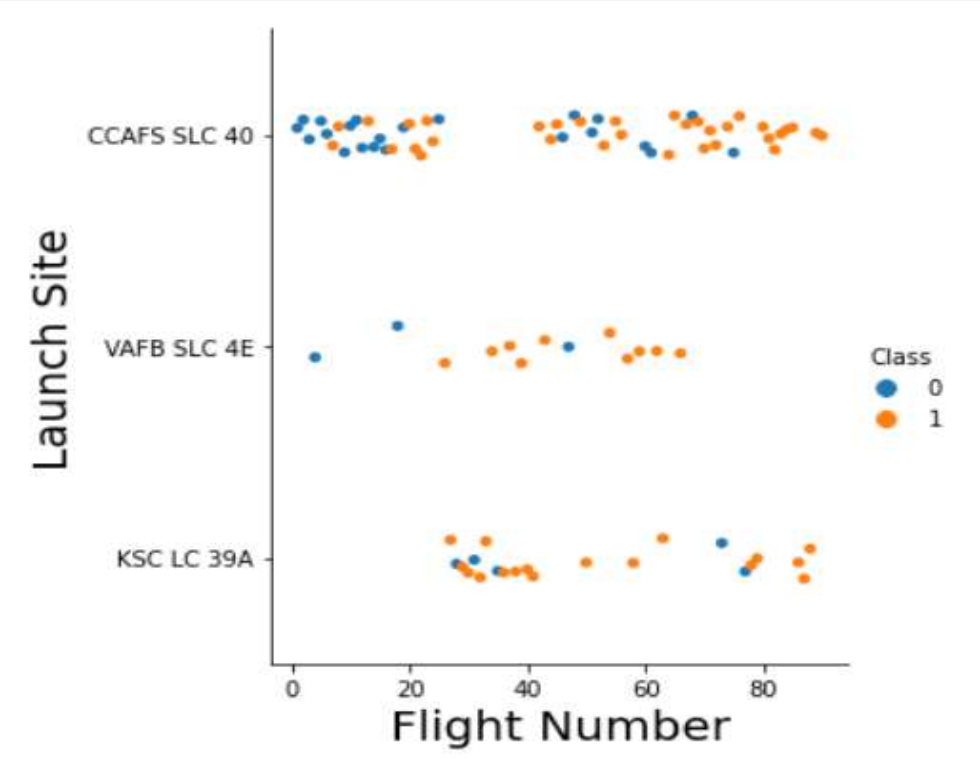


Section 2

Insights drawn from EDA

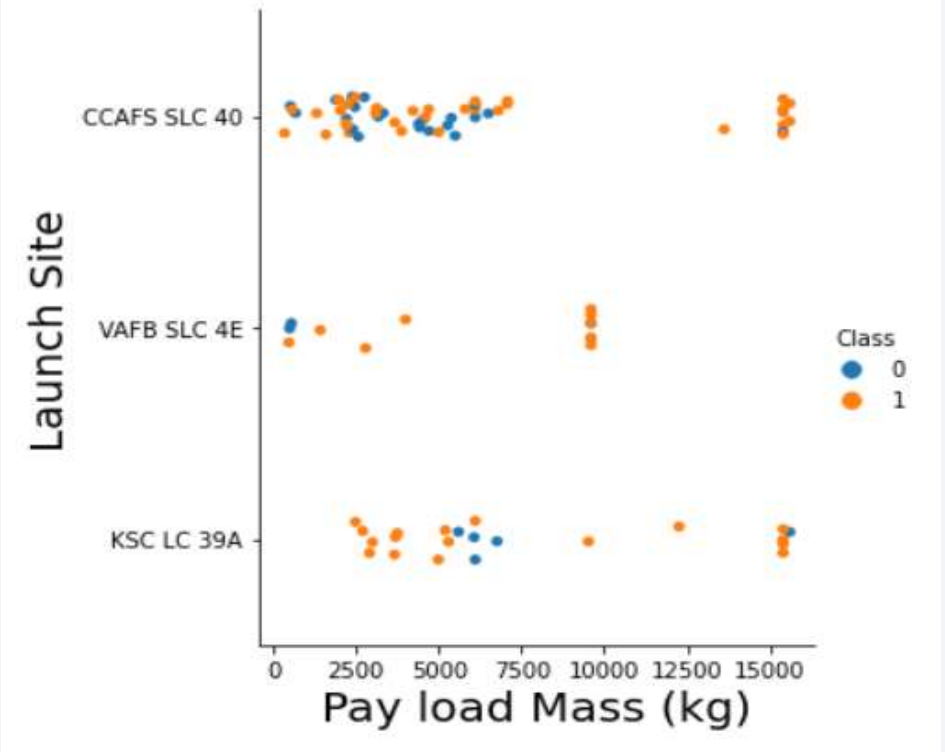
Flight Number vs. Launch Site

- Class 0 indicates an unsuccessful launch while Class 1 represents a successful launch.
- The adjacent scatterplot indicates that the success rate of flights increased the number of flights.
- The 20th flight can be seen as a breakpoint as the success rate increased tremendously following that point.



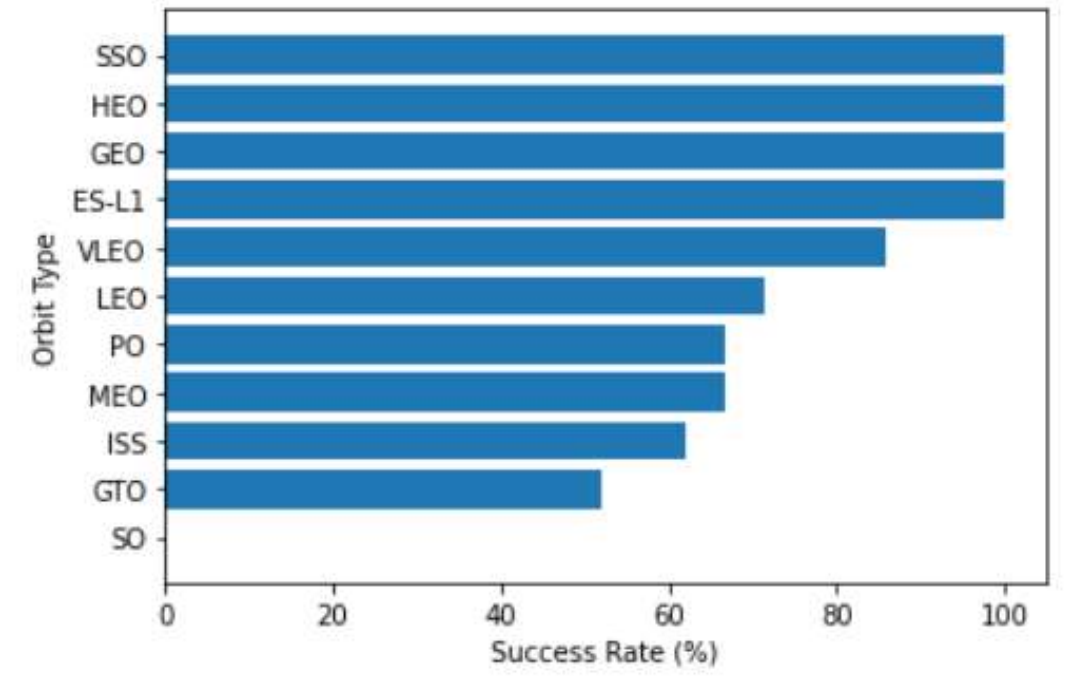
Payload vs. Launch Site

- Class 0 indicates an unsuccessful launch while Class 1 represents a successful launch.
- It may be inferred that at higher payload masses the success rate is higher but the volume of these flights is limited and thus we cannot draw a clear correlation between success rate and Payload Mass.



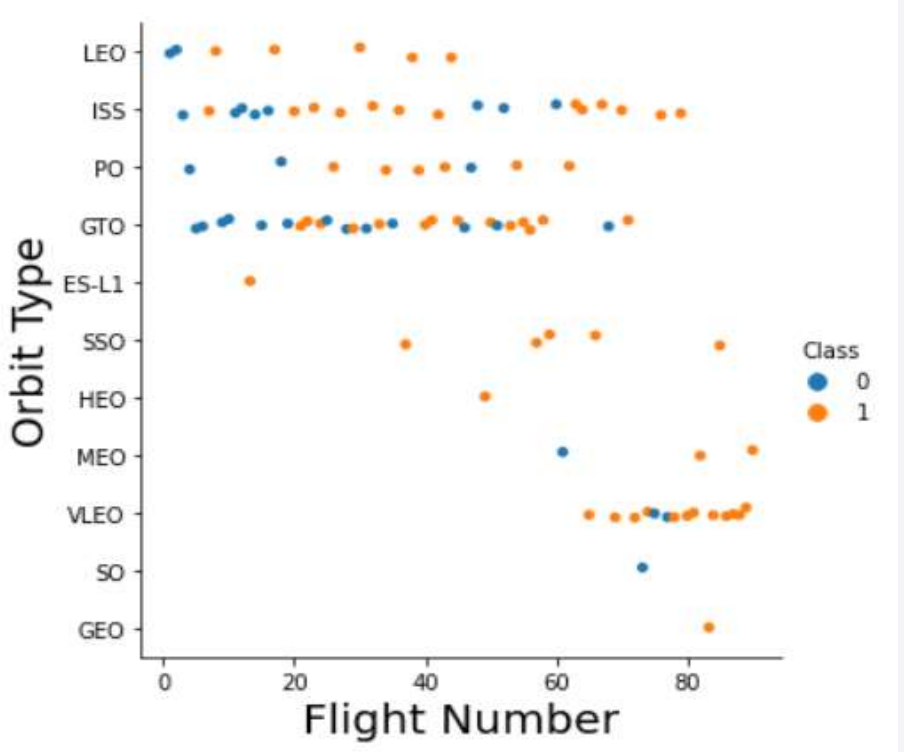
Success Rate vs. Orbit Type

- Orbits of type SSO, HEO, GEO, and ES-L1 have the highest success rates (100%).
- While, GTO exhibits a success rate of only 50% and SO exhibits the lowest success rate with 0%.



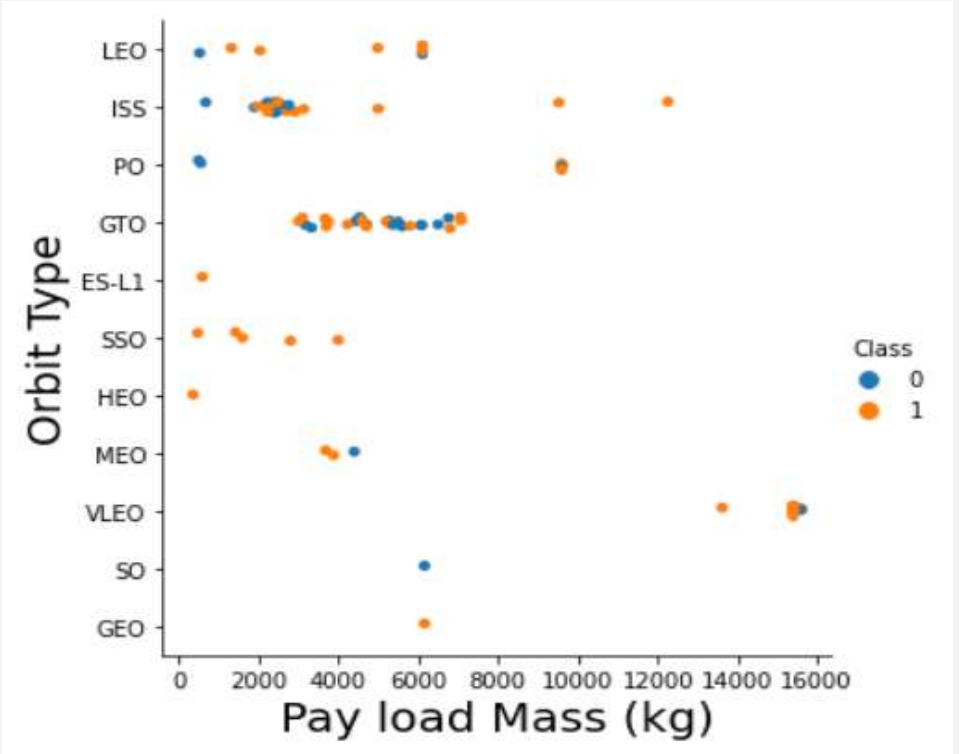
Flight Number vs. Orbit Type

- Class 0 indicates an unsuccessful launch while Class 1 represents a successful launch.
- The launch outcome shows some correlation with varying number of flights. GTO and ISS show signs of not conforming with that pattern.
- The LEO and VLEO orbits seem to have the highest success rate and seem to have been identified by SpaceX as the most effective orbit type.



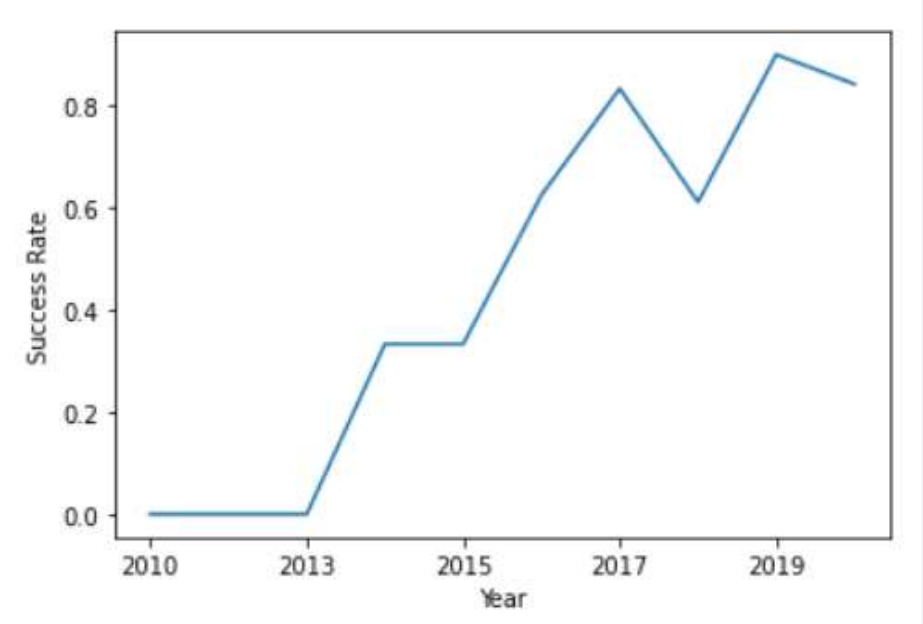
Payload vs. Orbit Type

- Class 0 indicates an unsuccessful launch while Class 1 represents a successful launch.
- The heavier payloads have mostly used the VLEO orbit and show a hig success rate.
- The GTO orbit type shows little correlation between landing success rate and payload mass.



Launch Success Yearly Trend

- We can clearly see that 2015 onwards was a time of great advancement for success rate trends.
- 2018 exhibited an unusual dip in success rate.
- Current rates exhibit about 80% of success.



All Launch Site Names

- Query :

```
SELECT DISTINCT LAUNCH_SITE
FROM SPACEXTBL
```

- Result

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

- DISTINCT selects only unique values in the Launch_Site column from the SpaceX table.

Launch Site Names Begin with 'CCA'

- Query :

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

- Result

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

- LIMIT 5 displays only 5 records from the SpaceX table.
- LIKE and the % sign, allows us to search for a Launch_Site starting with CCA.

Total Payload Mass

- Query :

```
SELECT SUM(PAYLOAD_MASS__KG_)
AS total_payload_mass_kg
FROM SPACEXTBL
WHERE CUSTOMER = 'NASA (CRS)'
```

- Result

total_payload_mass_kg
45596

- SUM() calculates the sum of the column PAYLOAD_MASS_KG_.

Average Payload Mass by F9 v1.1

- Query :

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

- Result

average_payload_mass_kg
2928.4

- AVG() calculates the average value of PAYLOAD_MASS__KG_.

First Successful Ground Landing Date

- Query :

```
SELECT MIN(DATE) AS first_successful_landing_date
FROM SPACEXTBL
WHERE LANDING_OUTCOME = 'Success (ground pad)'
```

- Result

first_successful_landing_date
2015-12-22

- MIN() finds the earliest date in the column date.

Successful Drone Ship Landing with Payload between 4000 and 6000

- Query :

```
SELECT BOOSTER_VERSION FROM SPACEXTBL
WHERE LANDING_OUTCOME = 'Success (drone ship)' AND (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000)
```

- Result

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

- AND allows us to select an additional condition PAYLOAD_MASS__KG_.

Total Number of Successful and Failure Mission Outcomes

- Query :

```
SELECT MISSION_OUTCOME, COUNT(*) AS total
FROM SPACEXTBL
GROUP BY MISSION_OUTCOME
```

- Result

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

- COUNT() calculates the total number of columns
- GROUP BY groups rows that have the same Mission_Outcome to find the total of these columns.
- The result shows that SpaceX has completed nearly 99% of all of its missions.

Boosters Carried Maximum Payload

- Query :

```
SELECT DISTINCT BOOSTER_VERSION
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (
    SELECT MAX(PAYLOAD_MASS__KG_)
    FROM SPACEXTBL);
```

- We use a subquery to first find the MAX() payload and secondly searching which Boosters carry the determined maximum payload.

- Result :

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

- Query :

```
SELECT substr(Date,4,2) as Month,
LANDING_OUTCOME, BOOSTER_VERSION,
LAUNCH_SITE
FROM SPACEXTBL
WHERE LANDING_OUTCOME = 'Failure (drone
ship)' AND substr(Date,7,4) = '2015'
```

- Using WHERE and AND we state 2 conditions which need to be satisfied for our query.
- We use the substr function to obtain the month since SQLLite does not support YEAR/MONTH functions.

- Result :

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

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

- Query :

```
SELECT LANDING_OUTCOME,
COUNT(LANDING_OUTCOME) AS total, DATE
FROM SPACEXTBL
WHERE substr(Date,7,4) || substr(Date,4,2)
 || substr(Date,1,2) BETWEEN '20100604' AND '20170320' AND "Landing _Outcome" LIKE "Success%"
GROUP BY "Landing _Outcome"
ORDER BY total DESC
```

- Result:

Landing _Outcome	total	Date
Success (drone ship)	5	08-04-2016
Success (ground pad)	3	22-12-2015

- ORDER BY allows us to put the results in descending order

Section 3

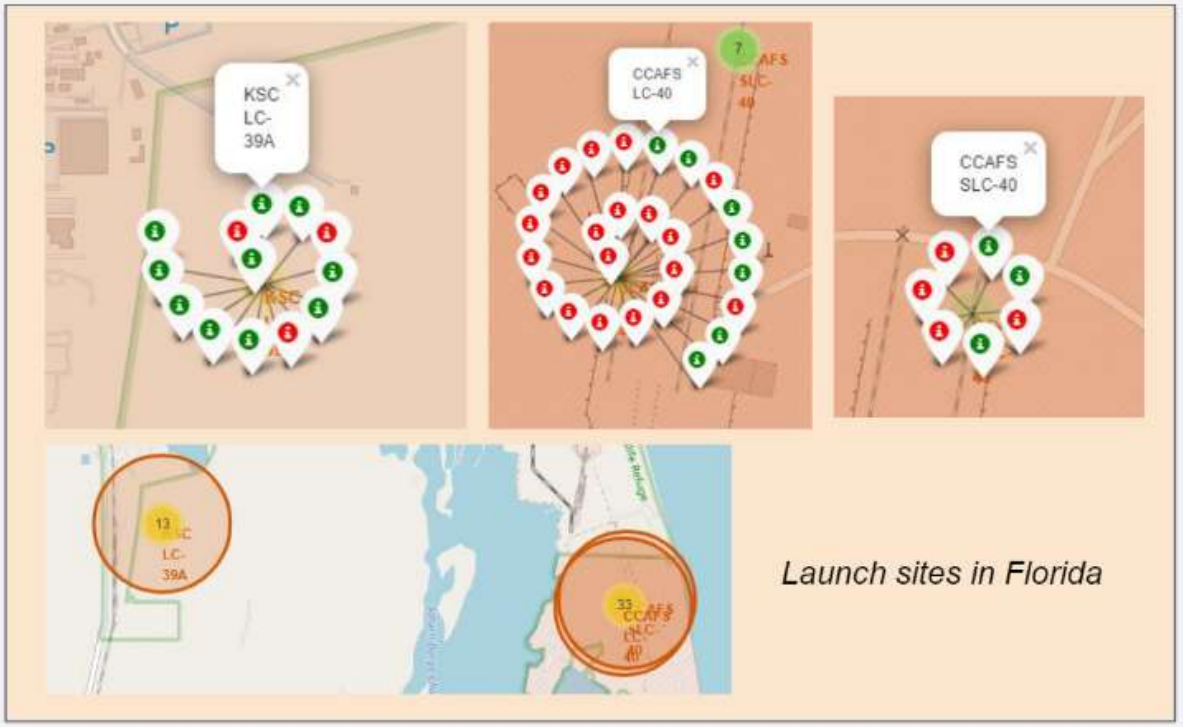
Launch Sites Proximities Analysis

All Launch Sites Location on Folium Map

- All launch sites are in the United States.
- All launch sites are near the coast.

Color-labeled Launch Outcomes

Launch site in California

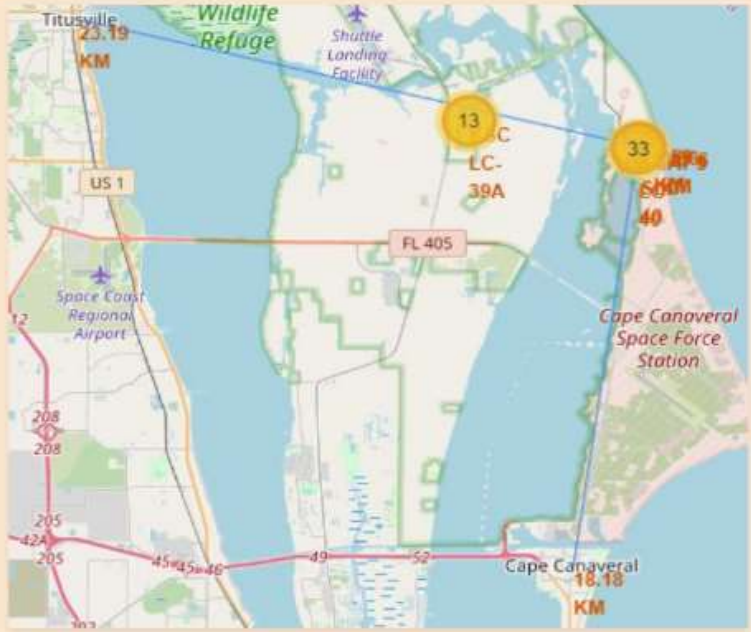


Launch sites in Florida

- The successful and failed landings are distinguished by green and red respectively.

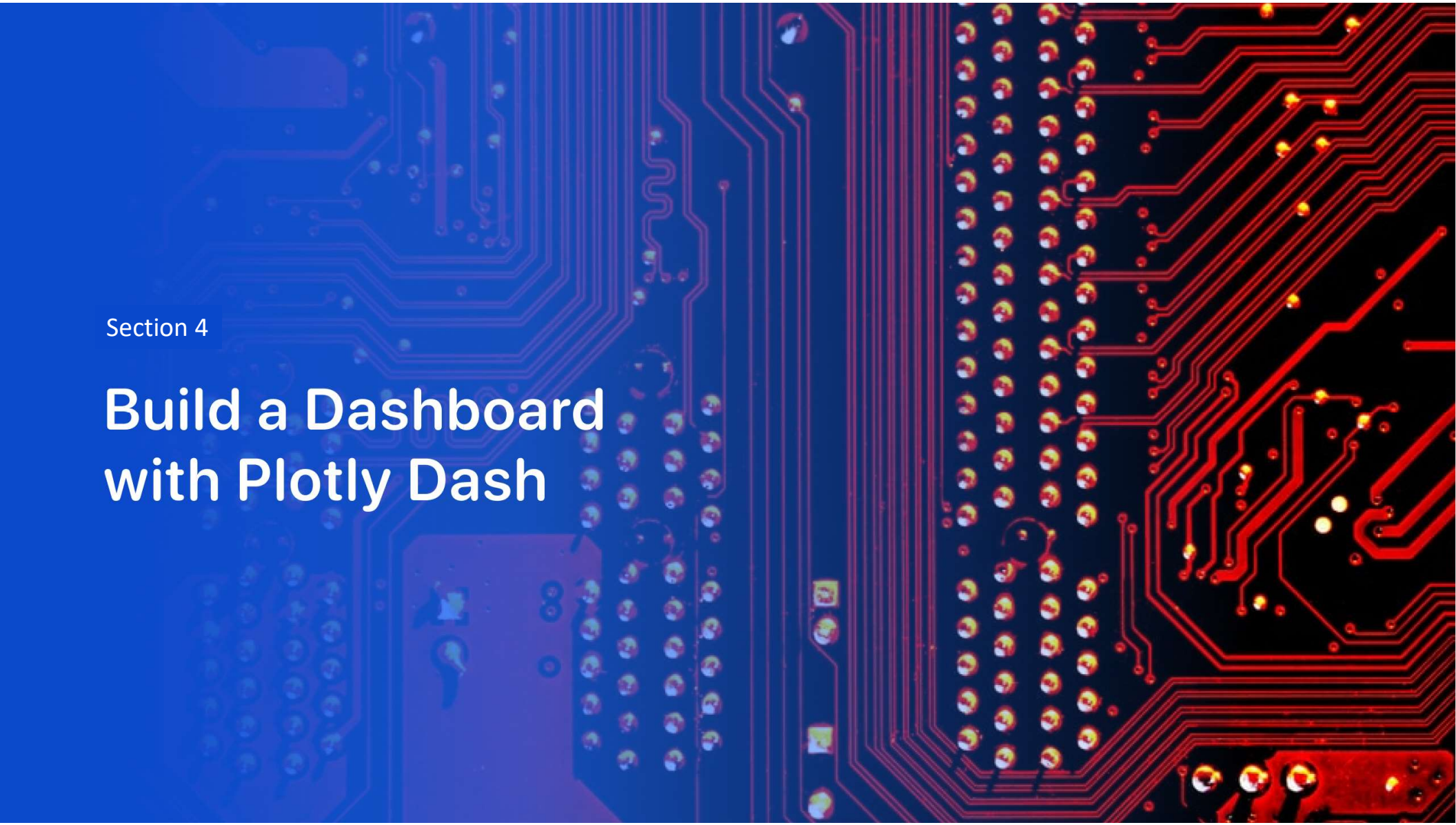
Proximity of Launch Sites

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***

- The launch sites are close to the coastline and far from cities so that launch failures do not create catastrophic events.

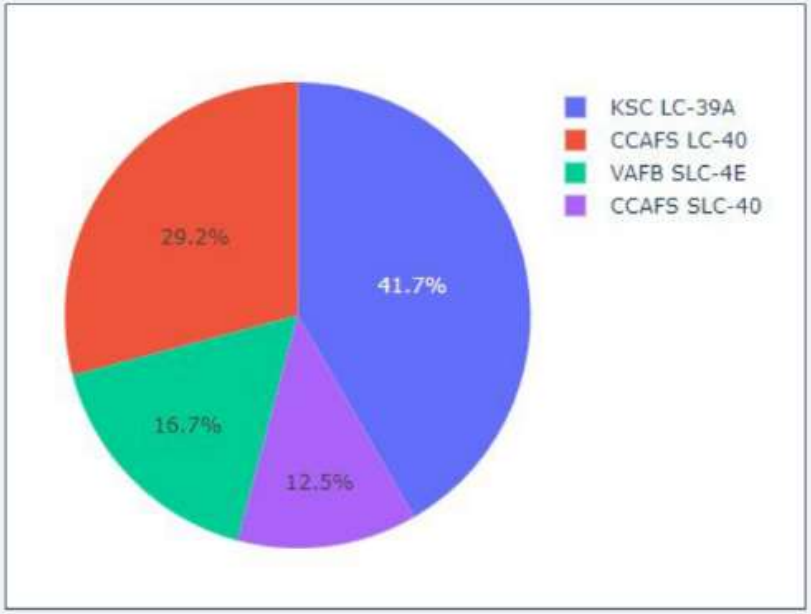


Section 4

Build a Dashboard with Plotly Dash

Successful Launches order by sites

- KSLC-39A is the site with the most launches.
- VAFB SLC-4E has the lowest launch success and is the only west coast launch site.
- It can be inferred that the east coast has favorable conditions for launches by the greater number of attempts and successes.



Launch Site with Highest Launch Success Rate

- KSLC-39A has the highest rate with 10 successful landings.

Payload vs. Launch Outcome Scatter Plots

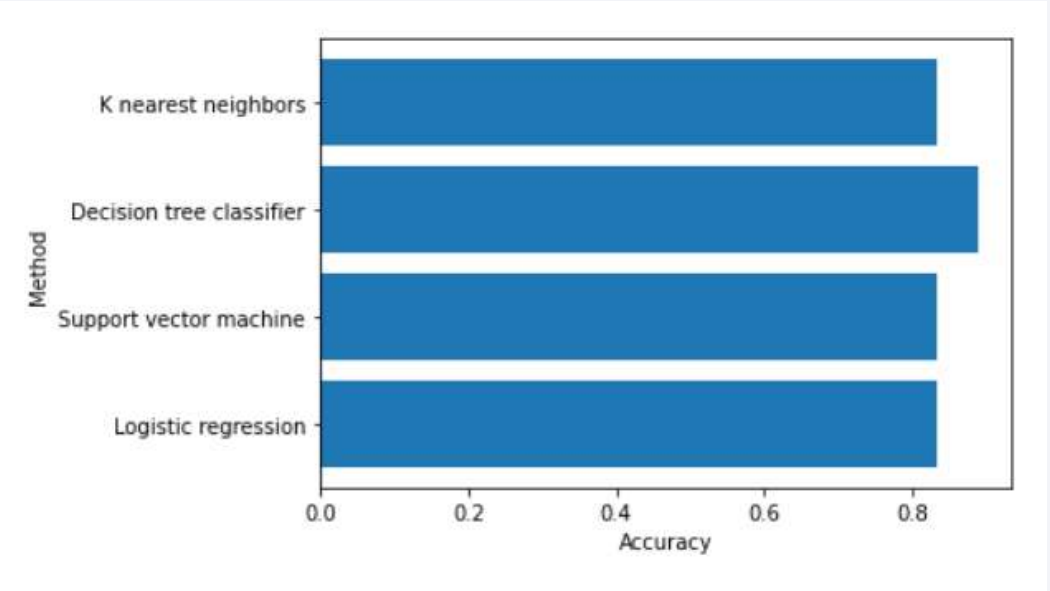
- The scatter plots show that the success rate (class 1) for low-weighted payloads is higher than that of higher-weighted payloads.

Section 5

Predictive Analysis (Classification)

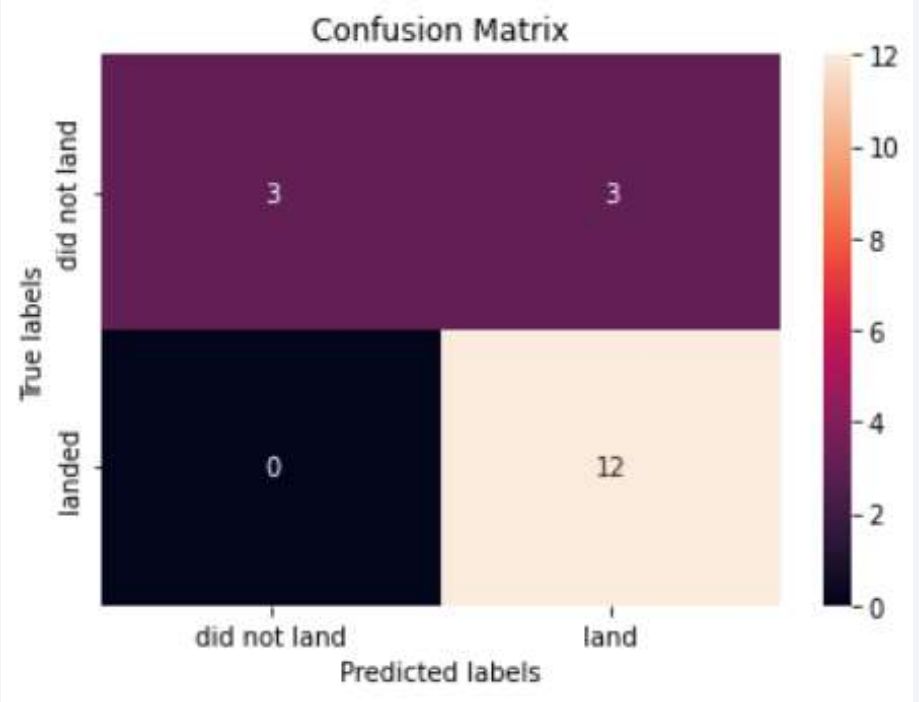
Classification Accuracy

- In the test test, all models produced an accuracy of about 83.3%.
- The test set was relatively small at 18.
- A larger training set could result in more reliable results.



Confusion Matrix

- The confusion matrix is basically the same for all models on the same test set.
- All models predicted 12 successful landings when true label was successful and 3 failed landings where the true label was failure.
- There are also 3 predictions for 3 successful landings when the reality was that those were failed landings(false positive).
- We can imply that the models correctly predict successful landings.



Conclusions

- The success rate increased with a rise in number of flights and recently stagnated at the 80% mark. This is likely due to the use of practical data to improve future launches.
- Orbit types SSO, HEO, GEO, AND ES—L1 have a success rate of 100%.
- The launch sites are close to railways, highways, and coastlines, but far from cities.
- KSLC-39A has the highest number of launch successes and the highest success rate among all sites.
- The launch success rate of low-weighted payloads is higher than that of heavy-weighted payloads.
- In this dataset, all the classification models produce the same accuracy of 83%. However, this result is based on a relatively small dataset.

Appendix

- GitHub Repository URL

Thank you!