IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Yashar Shabbouei Hagh
13.10.2021

Outline

Executive Summary

❑**Summary of methodologies**

- Data Collecting
  - SpaceX REST API
  - Web Scrapping
- Data Wrangling
- Exploratory Data Analysis
  - SQL
  - Data Visualization
- Interactive Visual Analytics
  - Folium
  - Plotly Dash
- Predictive Analysis (Classification)

❑**Summary of all results**

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

Introduction

- **Project background and context**
  - We will predict whether the Falcon 9 first stage will successfully land or not. SpaceX, advertises Falcon 9 rocket launches for 62 million dollars; other providers charge upwards of 165 million dollars per launch; much of the savings comes from the fact that SpaceX can reuse the first stage. As a result, if we can figure out if the first stage will land, we can figure out how much a launch will cost. If another company wants to compete with SpaceX for a rocket launch, this information can be used.
  - To this end, we are going to build and train a machine learning model which can predict the successful landing of the first stage based on historical data obtained from different sources.
- **Problems to be answered**
  - What is the impact of different independent variables such as different payload mass, launch sites, F9 booster versions, orbit types on the landing success/fail rate?
  - What conditions should be considered to increase the successful landing rate?
  - Can we develop a model to predict whether the first stage of the rocket would land successfully or not?

Section 1

Methodology

Methodology

- Data collection methodology:
  - Data is collected through SpaceX REST API and Web Scraping of the wikipedia
- Perform data wrangling
  - Determining the Training Labels for supervised models by examining different attributes of the dataset
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build, tune and evaluate different types of supervised classification models

Data Collection

- SpaceX REST API
- Web Scraping Wikipedia page

- SpaceX REST API
  - The SpaceX REST API endpoints, or URL, starts with api.spacexdata.com/v4/
  - We mostly work with the endpoint api.spacexdata.com/v4/launches/past, however other endpoints such as /rickets, /launchpads, /payloads and /cores will also be used to obtain booster name, launch site information, payload information and the outcome of the landing, respectively.
  - A get request using the requests library is performed on these endpoints to obtain the launch data
  - The response is in .json() format which should be converted to a Pandas data frame through .json_normalize() function

- Web Scraping
  - Wikipedia is another data source about Falcon 9 Launch data
  - The Python BeautifulSoup package is used to web scrape some HTML tables that contain valuable Falcon 9 launch records
  - The data from these tables are parsed and converted into a Pandas data frame for further analysis

GitHub URL

Data Collection: SpaceX API

- Perform a GET request on the specified endpoint APIs
- Convert the response .json() file to a data frame
- Filter the data frame to only include Falcon 9 launches

1. Requesting rocket launch data

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

2. Decode the response content as .json() and turn it into a Pandas data frame

```
response = requests.get(static_json_url)
data = pd.json_normalize(response.json())
```

3. Apply different function methods to get Booster version, LaunchSite, PyloadData and CoreData

```
# Call getBoosterVersion
getBoosterVersion(data)
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
# Call getCoreData
getCoreData(data)
```

4. Construct dictionary file and then convert it to 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}
launch_df = pd.DataFrame({key:pd.Series(value) for key,value in launch_dict.items()})
```

5. Remove Falcon 1 launches from the data

```
data_falcon9 = launch_df[launch_df.BoosterVersion != 'Falcon 1']
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
```

6. Save the Data Frame

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

GitHub URL

Data Collection: Scraping

- Perform an HTTP GET request on the Falcon 9 Launch HTML Wiki page
- Find the target table which contains launch records
- Parse through the launch HTML table
- Create a data frame

1. Request the Falcon 9 Launch HTML page

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

2. Create a BeautifulSoup object

```
soup = BeautifulSoup(page, "html5lib")
```

3. Find the target table

```
html_tables = soup.find_all('table')
first_launch_table = html_tables[2]
```

4. Extract the Column names

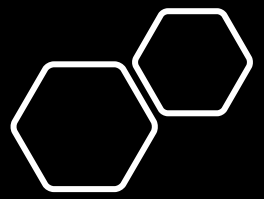
```
column_names = []
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

5. Create a dictionary file (refer to blocks 12 and 19 of the Jupyter Notebook)

6. Save the Data Frame

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

GitHub URL



Data Wrangling

- ❑ Dealing with the missing values
- ❑ Perform exploratory data analysis
- ❑ Determine training labels

1. Finding the rows with missing values

```
data_falcon9.isnull().sum()
```

2. Replacing the missing values with the mean of the column

```
# Calculate the mean value of PayloadMass column
PayloadMean = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan, PayloadMean, inplace=True)
```

3. Calculate the number of launches on each site

```
# Apply value_counts() on column LaunchSite
df["LaunchSite"].value_counts()
```

4. Calculate the number and occurance of each orbit

```
# Apply value_counts on Orbit column
df["Orbit"].value_counts()
```

5. Calculate the number and occurrence of mission outcome per orbit type

```
# landing_outcomes = values on Outcome column
landing_outcomes = df["Outcome"].value_counts()
```

6. Create a landing outcome label

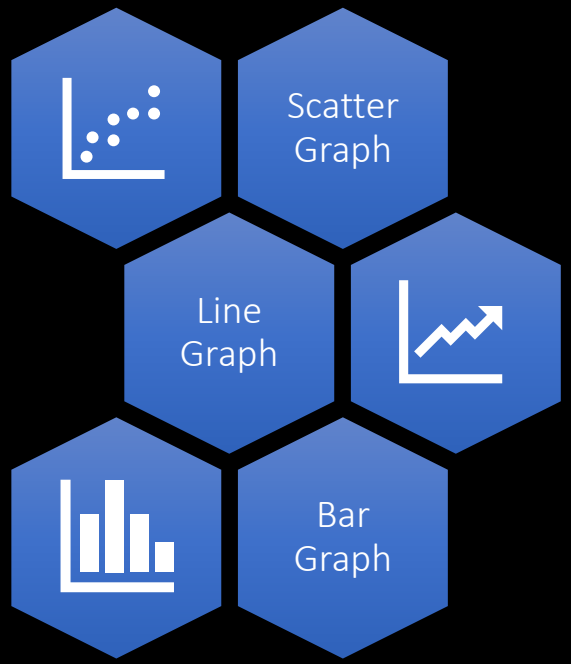
```
landing_class = []
for index, row in df["Outcome"].items():
    if row in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
df['Class']=landing_class
```

7. Save the Data Frame

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

GitHub URL

EDA with data visualization



- ***Scatter Graph*** represents values for two different numeric variables. The position of each dot on the horizontal and vertical axis indicates values for an individual data point. Scatter plots are used to observe relationships between variables to obtain how much one variable is affected by another variable
  - Flight Number vs. Payload Mass
  - Flight Number vs. Launch Site
  - Payload Mass vs. Launch Site
  - Flight Number vs. Orbit Type
  - Payload Mass vs. Orbit Type
- ***Line Graph*** is ideal for a continous dataset and when we are intrested in visualizing the data over a period. Typically, line graphs are used to track changes over short and long periods of time
  - Launch success yearly trend
- ***Bar Graph*** is a type of plot where the length of each bar is proportional to the value of the item that it represents. It is a visual tool that uses bars to compare data among categories. A bar graph may run horizontally or vertically
  - Success Rate vs. Orbit Type

GitHub URL

EDA with SQL

The list 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 URL

Build an Interactive Map with Folium

- An interactive map is used to visualize the launch site locations.
  - Extracting the latitude and the longitude coordinates of each site, Folium.map, Folium.Circle and Folium.map.marker are used to create a folium map object, add a highlighted circle area and a text label on each launch site, respectively.
  - The success/failed launches are marked with green marker color as success and red marker color as failed. Then, for each launch result, a Folium.Marker is added to the MarkerCluster
  - The distances between a launch site to its proximities are calculated and lines are drawn on the map from these proximities to the specified launch site to find out the surroundings of the launch site.
    - The distance between the coastline point and the launch site
    - The distance between the railway point and the launch site
    - The distance between the highway point and the launch site
    - The distance between the city point and the launch site

GitHub URL

Build a Dashboard with Plotly Dash

- Python has taken over the world, and Dash Enterprise is the leading vehicle for delivering Python analytics to business users

- The dashboard includes:

1. Pie Chart:
   - A circular statistical graphic divided into slices to illustrate numerical proportion
   - Visualizing launch success counts per each site or all the sites together.
   - If a specific launch site is selected, the graph shows the proportion of success versus failed attempts for the selected site.

2. Scatter Graph:
   - Visualizing the relationship between two variables
   - Visually observe how Payload Mass may be correlated with mission Outcome for selected site(s)
   - A range slider is provided to select the range of desired Payload Mass

GitHub URL

Predictive Analysis

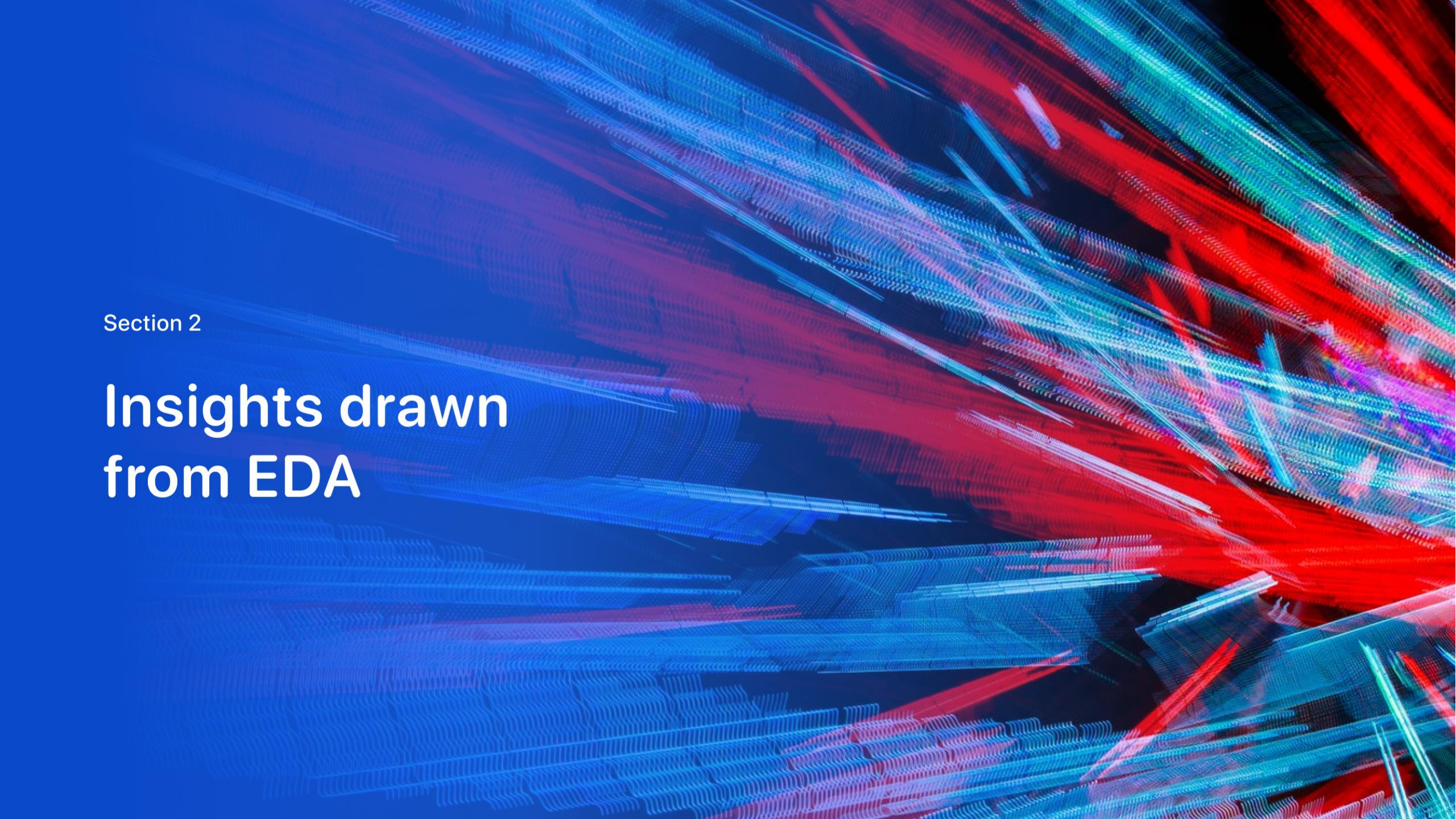
Build	1. Load the Data Frame 2. Indicate the features dataset and the target variable 3. Standardiz and transform the data 4. Split the data into train and test sets 5. Indicate the machine learning algorithm 6. Find the best tuning paramters for the algorithm
Evaluate	1. Check th accuracy of each model 2. Plot confusion matrices for each model
Improve	1. Feature Engineering 2. Re-tune the parameters of the algorithm
Find	1. The model with the highest performance score is choosen

Results

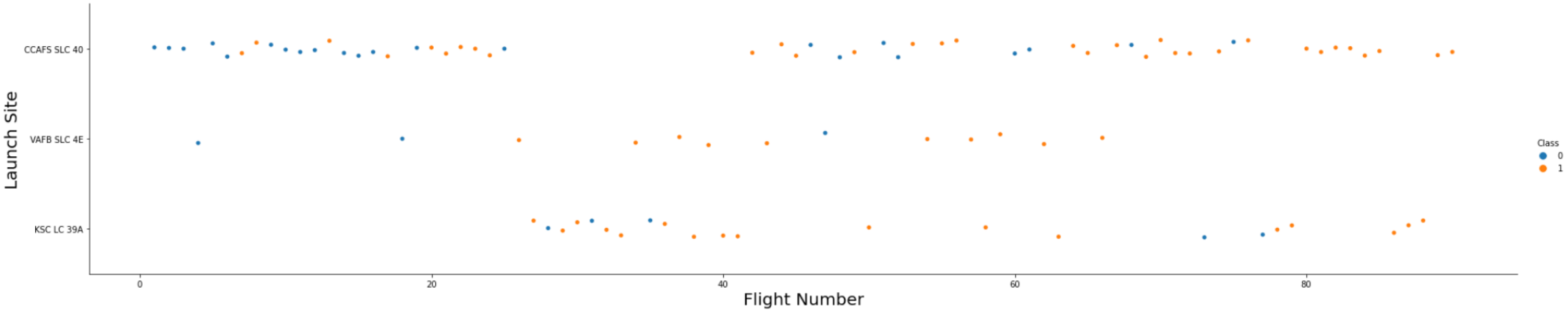

Exploratory
Data
Analysis
Interactive
Analytics
Demo
Predictive
Analysis

Section 2

Insights drawn from EDA

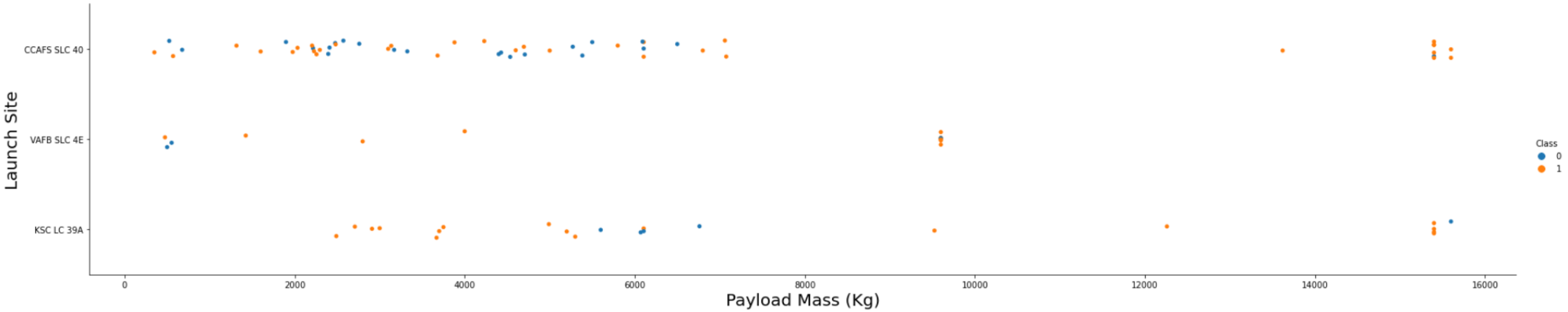


Flight Number vs. Launch Site



- The most used site is CCAFS SLC 40
- As the number of flights increased in CCAFS SLC 40, the success rate also increased
- As for the other sites, increasing the number of flight resulted in less failed outcomes

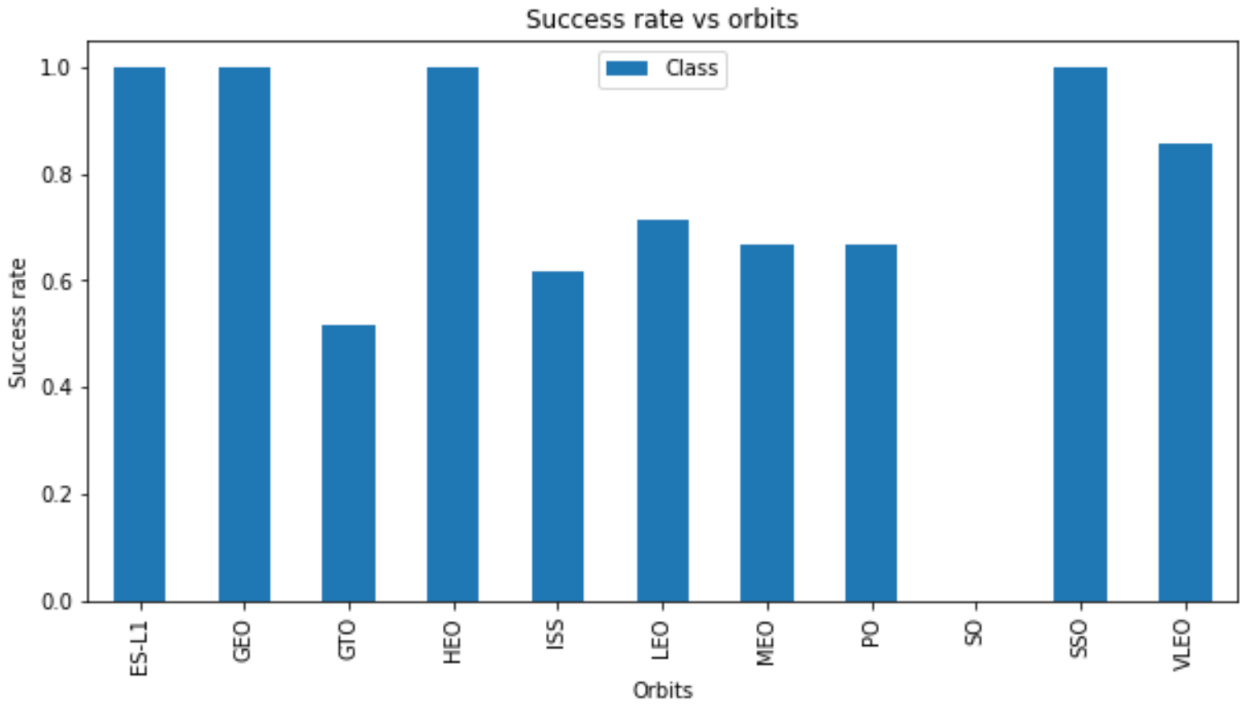
Payload vs. Launch Site



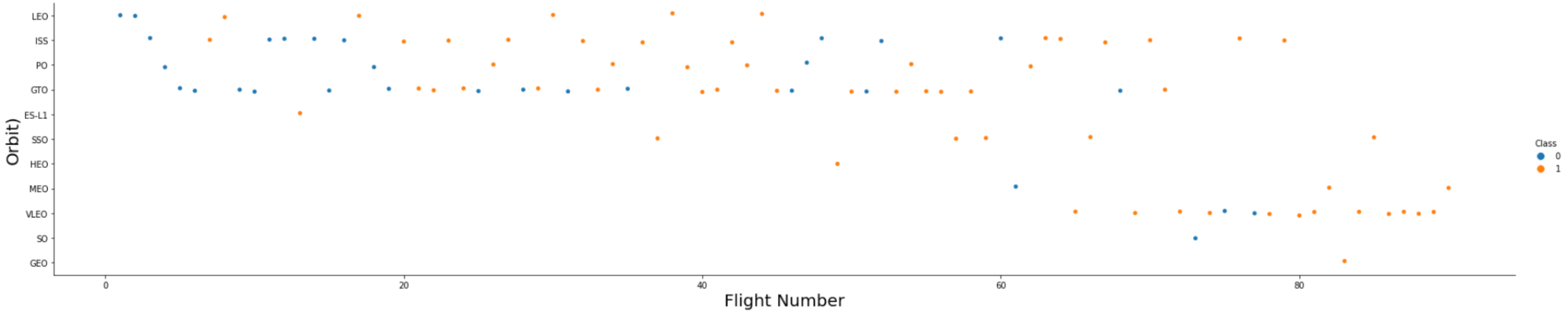
- Most launches are for payloads less than 8000 kg
- Increasing the payload resulted in successful landing; for more than 8000 kg payload, we have only 3 failing versus 15 successful landing in all sites (other features should also be considered)

Success Rate vs. Orbit Type

- The best orbits with highest success rate of one are ES-L1, GEO, HEO and SSO
- The least preferred orbit is GTO with success rate of 0.5

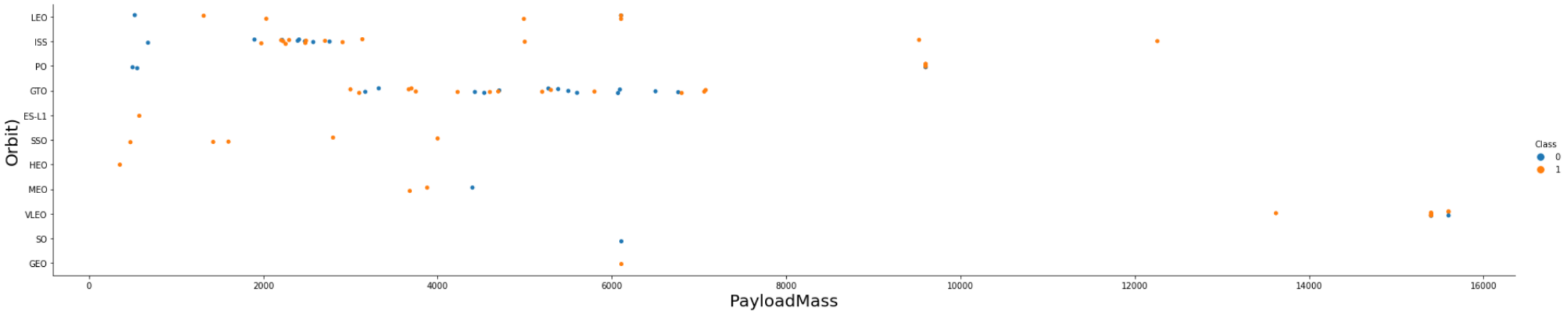


Flight Number vs. Orbit Type



- In the LEO orbit the Success appears related to the number of flights
- Overall, there seems to be no relationship between flight number when in GTO orbit

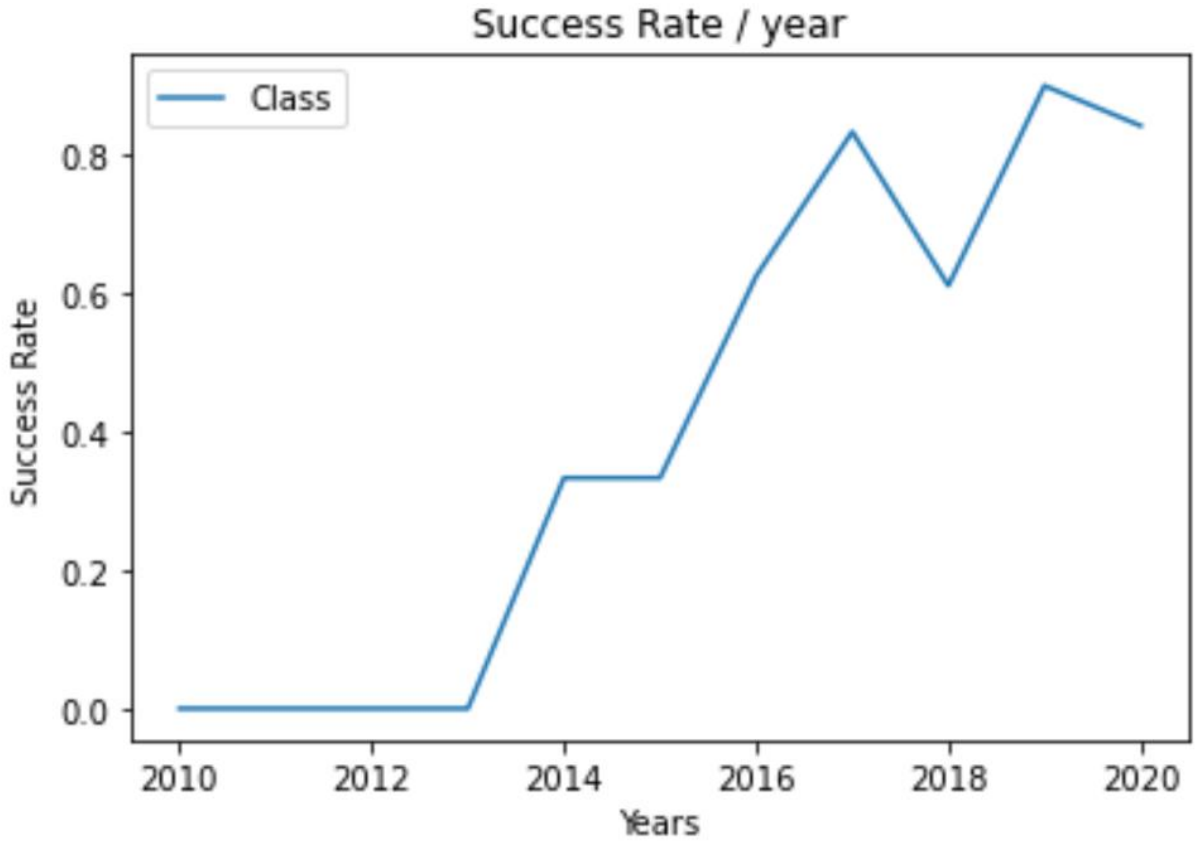
Payload vs. Orbit Type



- The increase in Payload Mass has negative impact on GTO orbit
- ISS orbit exhibits a proper outcome for high Payloads

Launch Success Yearly Trend

- The success rate since 2013 kept increasing till 2020
- The highest success rate was in 2019



All Launch Site Names

- QSL Query:
  - Select DISTINC(LAUNCH_SITE) FROM SPACEXTBL
- Explanation:
  - Using "DISTINCT" to show unique values of the Launch_Site column

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

Launch Sites Name Begin with 'CCA'

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

- QSL Query:
  - SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
- Explanation:
  - Using "LIKE" and "LIMIT" to show queries begin with 'CCA' and limit to only "5" rows

Total Payload Mass

- QSL Query:
  - SELECT SUM(payload_mass__kg_) AS SUM_OF_PAYLOAD FROM SPACEXTBL WHERE customer= 'NASA (CRS)'

sum_of_payload
45596

- Explanation:
  - Using "SUM" to calculate total values in column and filter the result through the "WHERE" condition

Average Payload Mass by F9 v1.1

- QSL Query:
  - SELECT AVG(payload_mass__kg_) AS AVG_OF_PAYLOAD FROM SPACEXTBL WHERE booster_version LIKE 'F9 v1.1%'

avg_of_payload
2534

- Explanation:
  - Using "AVG" to calculate average of values in column and filter the result through the "WHERE" condition and "LIKE" function

First Successful Ground Landing Date

- QSL Query:
  - SELECT MIN(DATE) AS DATE FROM SPACEXTBL WHERE landing__outcome= 'Success (ground pad)'

DATE
2015-12-22

- Explanation:
  - Using "MIN" to find the minimum date in the column and filter the result through the "WHERE" condition

Successful Dron Ship Landing with Payload between 4000 and 6000

- QSL Query:
  - SELECT booster_version FROM SPACEXTBL WHERE (landing__outcome= 'Success (drone ship)') AND (PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000)

- Explanation:
  - Selecting "Booster_version" from the table in which the "landing_outcome" is "Success (dron ship)" AND the "Payload_Mass__KG_" is within the specified range

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

Total Number of Successful and Failure Mission Outcomes

- QSL Query:
  - SELECT mission_outcome, COUNT(mission_outcome) AS COUNT FROM SPACEXTBL GROUP BY mission_outcome

- Explanation:
  - Selecting "mission_outcome" from the table and counting the number of each, then "Group BY" the mission outcome

mission_outcome	COUNT
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- QSL Query:
  - SELECT booster_version FROM SPACEXTBL WHERE payload_mass__kg_= (SELECT MAX(payload_mass__kg_) FROM SPACEXTBL)

- Explanation:
  - Using a "Subquery" to gather the "Max" payload mass, then finding the "booster_version" with the highest payload mass

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

- QSL Query:
  - SELECT landing__outcome, booster_version, launch_site FROM SPACEXTBL WHERE (landing__outcome= 'Failure (drone ship)') AND (YEAR(DATE)='2015')

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

- Explanation:
  - Using "YEAR(DATE)" to extract the year from the column "AND" filtering the "landing__outcome" to the values of "Failure (drone ship)"

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

❑ QSL Query:

- SELECT landing__outcome, count(landing__outcome) AS COUNT FROM SPACEXTBL
- WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
- GROUP BY landing__outcome
- ORDER BY COUNT DESC

❑ Explanation:

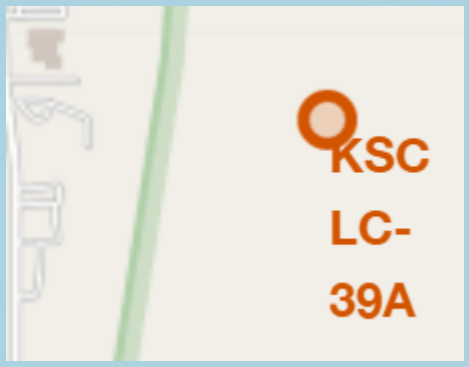
- Using "BETWEEN" to indicate the dates, "Group BY" to group the "landing__outcome" with the "COUNT" values and showing it in a descending order through "Order BY" and "DESC"

landing__outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 4

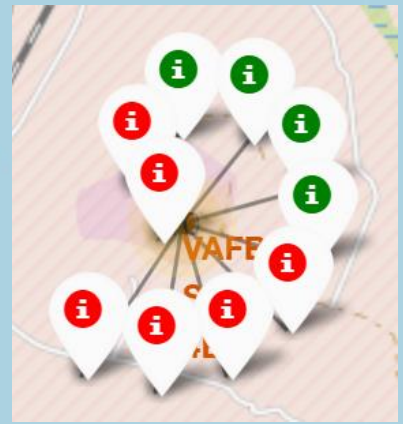
Launch Sites Proximities Analysis

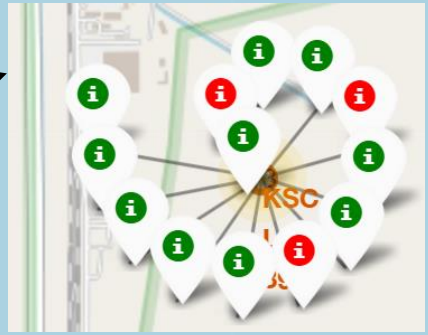
Locate each Launch Site on the Map

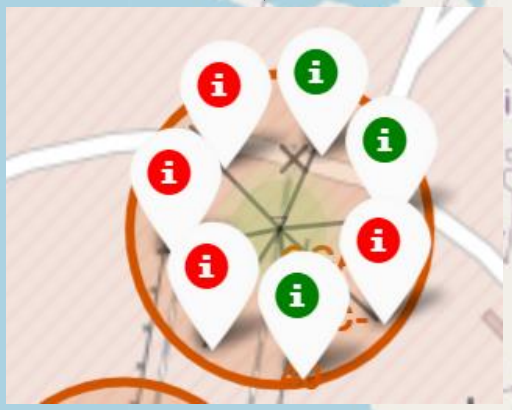


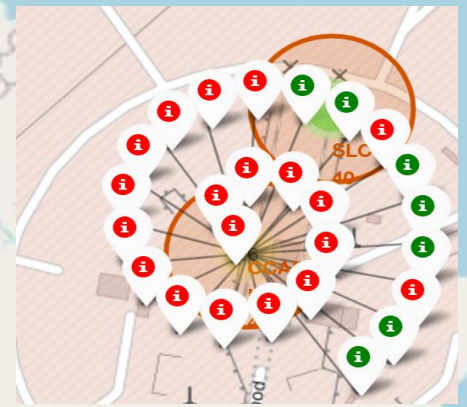
- The launch sites KSC LC-39A, CCAFS SLC-40 and CCAFS LC-40 are in proximity to the Equator line (latitude=28) while the VAFB SLC-4E is in a norther position (Lat=34)
- All sites are in close proximity to the coast

Mark the success/failed Launches for each Site



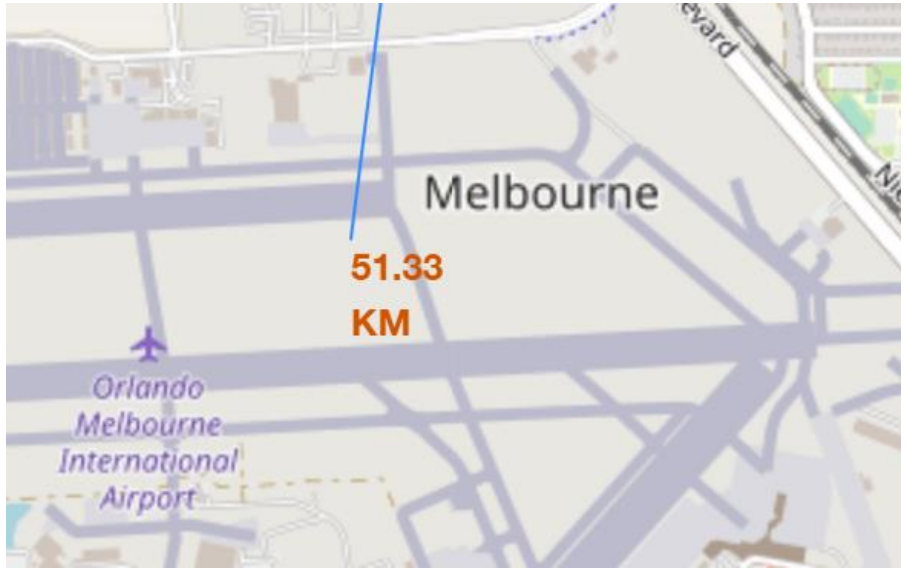


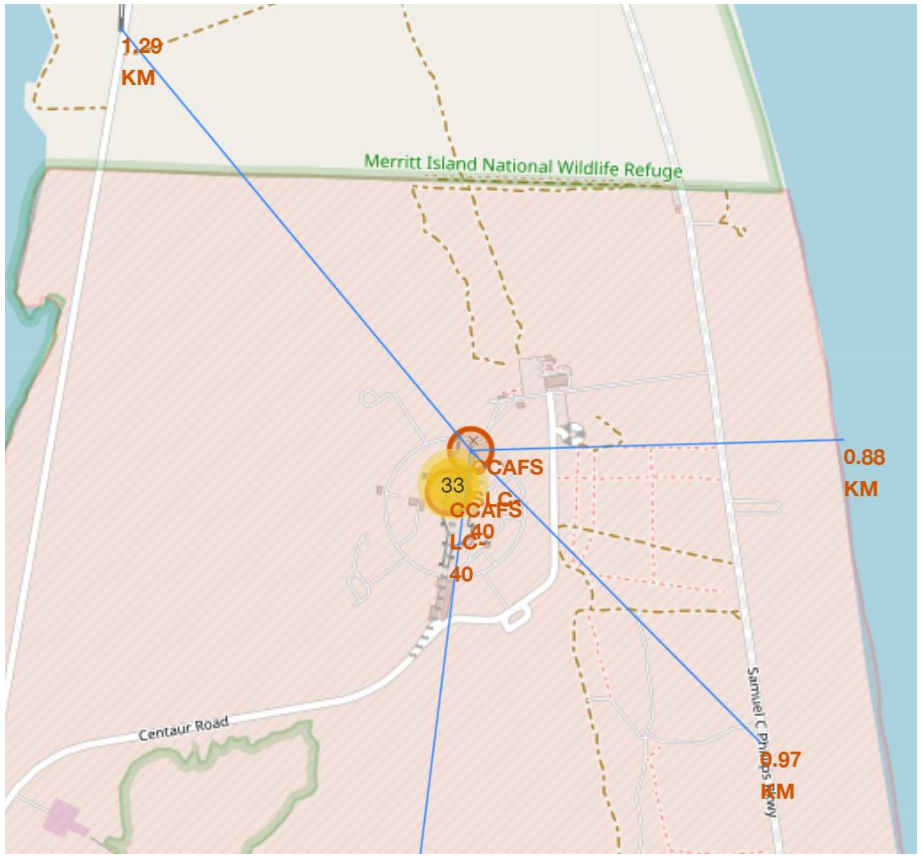




- Green marks indicate successful launches and Red markes shows failed launches

Launch Site Distance to different Proximities

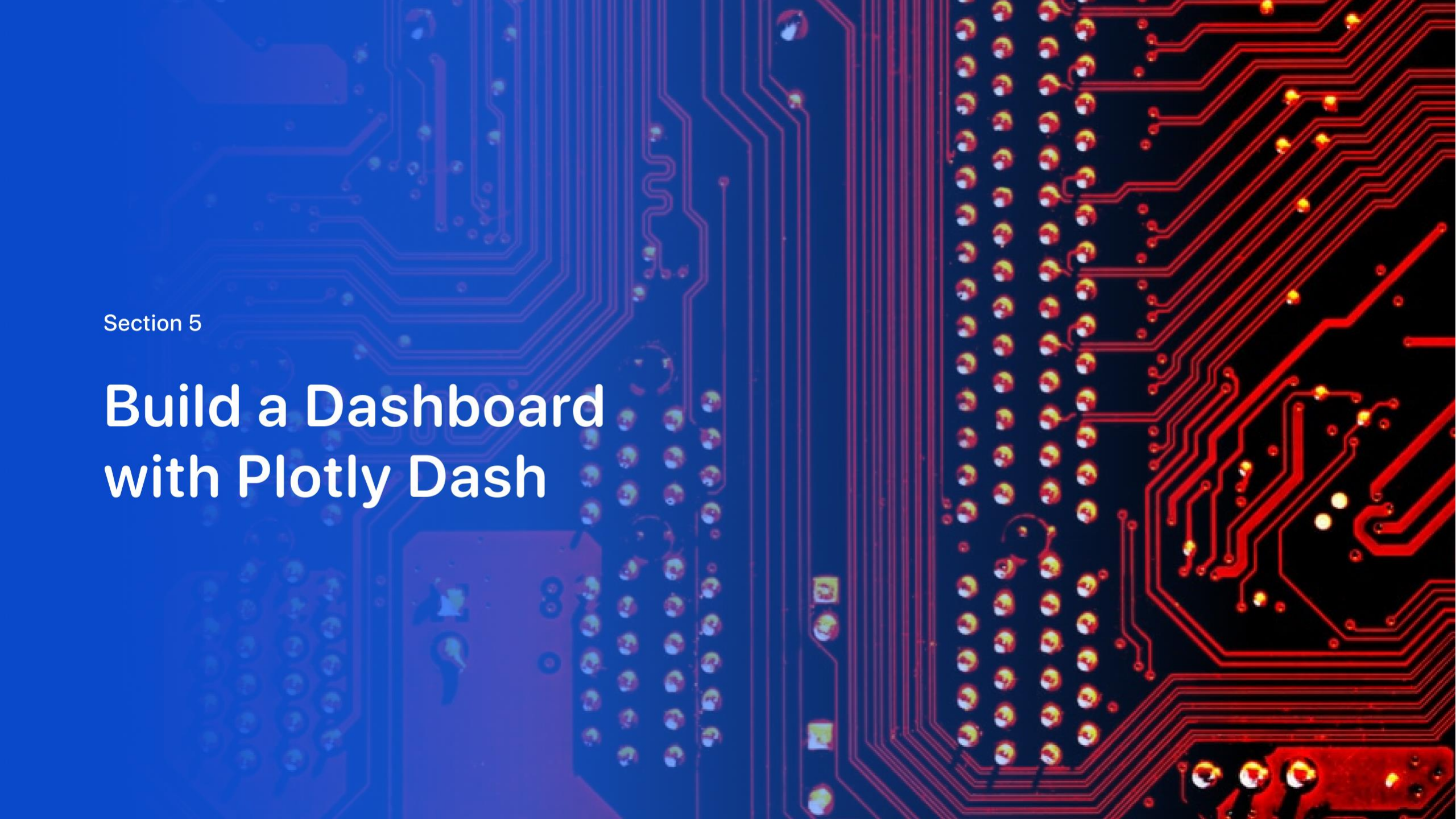




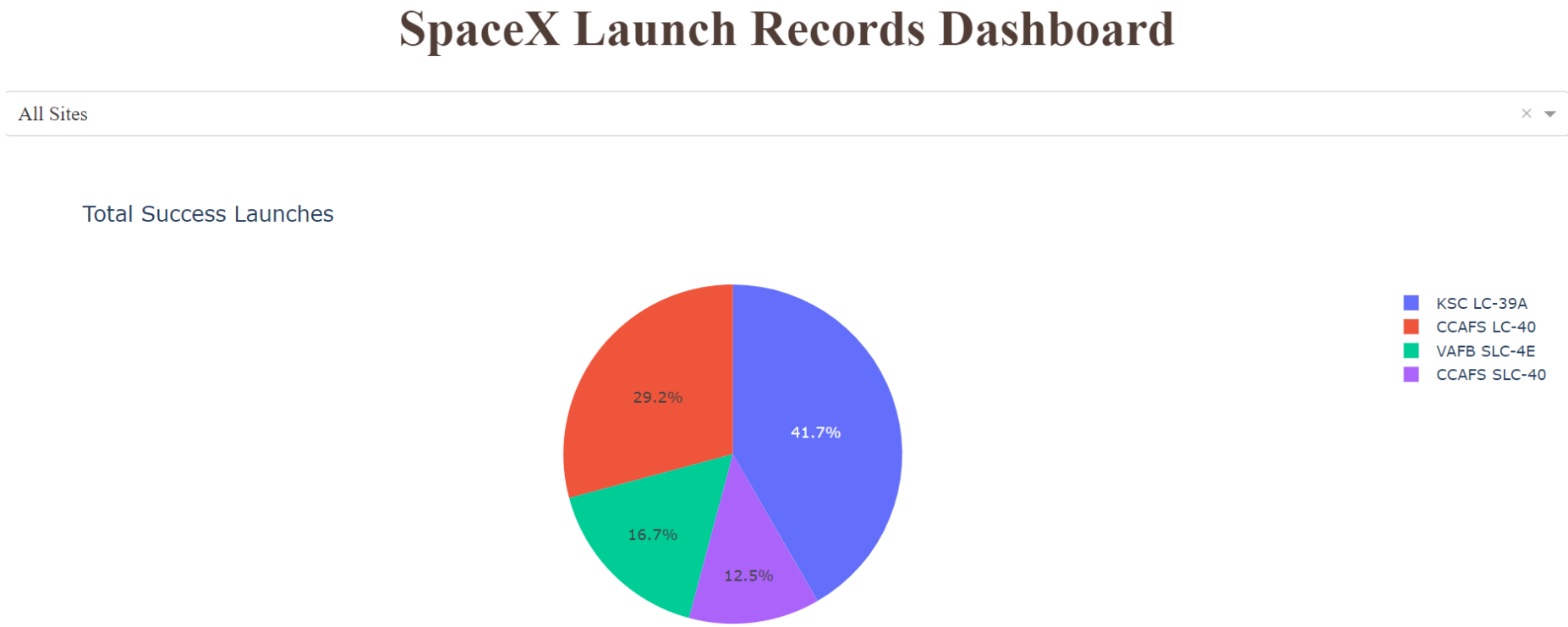
- The Launch sites are near coast and far away from cities
- For instance, the site CCAFS SLC-40 is 1.29 Km far from railway, 0.88 Km from coast, almost 10 Km far from highway and 51 Km away from the city

Section 5

Build a Dashboard with Plotly Dash

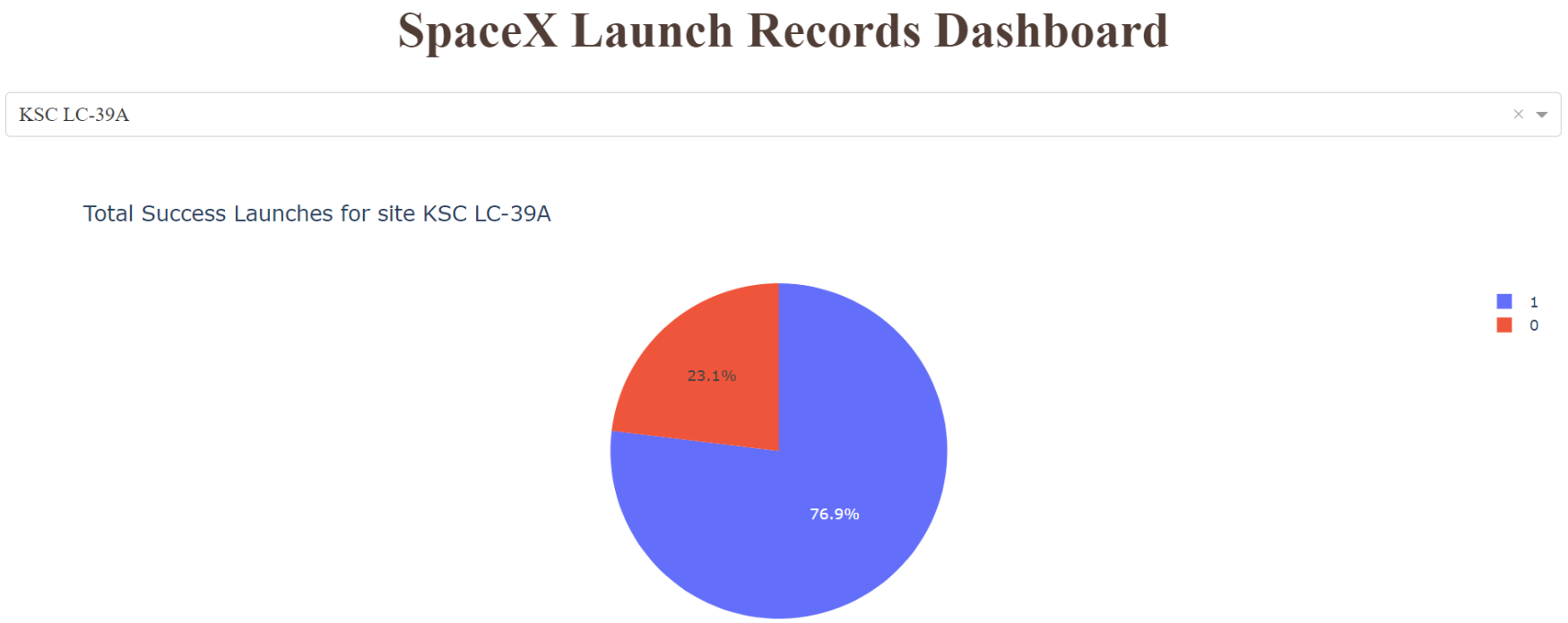


Total Success Launches for All Sites



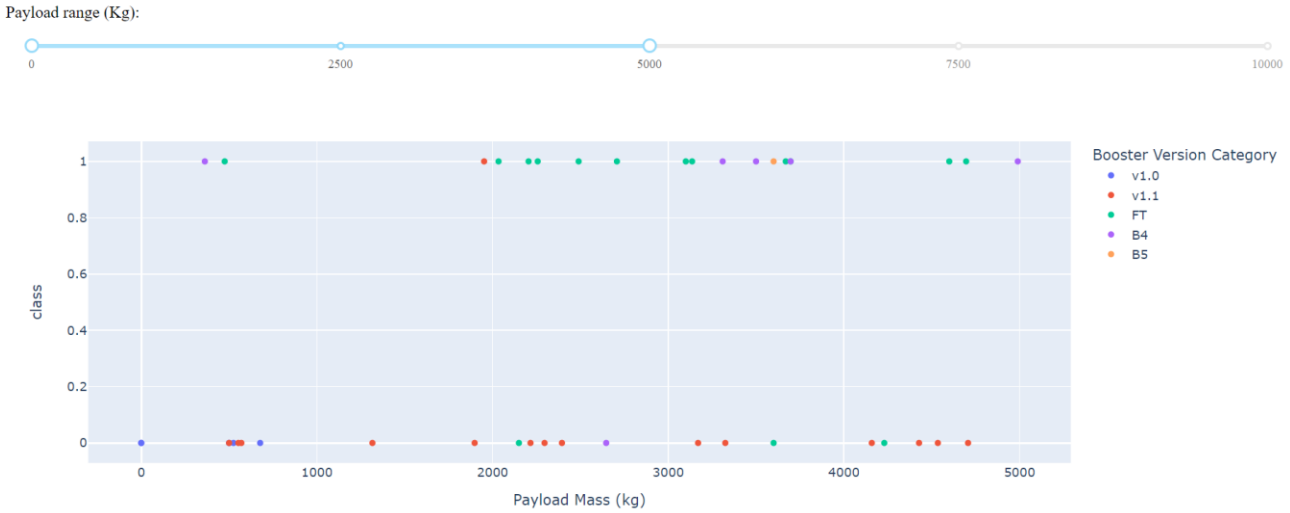
- The Launch site KSC LC-39A has the highest rate of success (41.7%)
- The Launch site CCAFS SLC-40 has the lowest rate of success (12.5%)

Launch Site with Highest Launch Success ratio

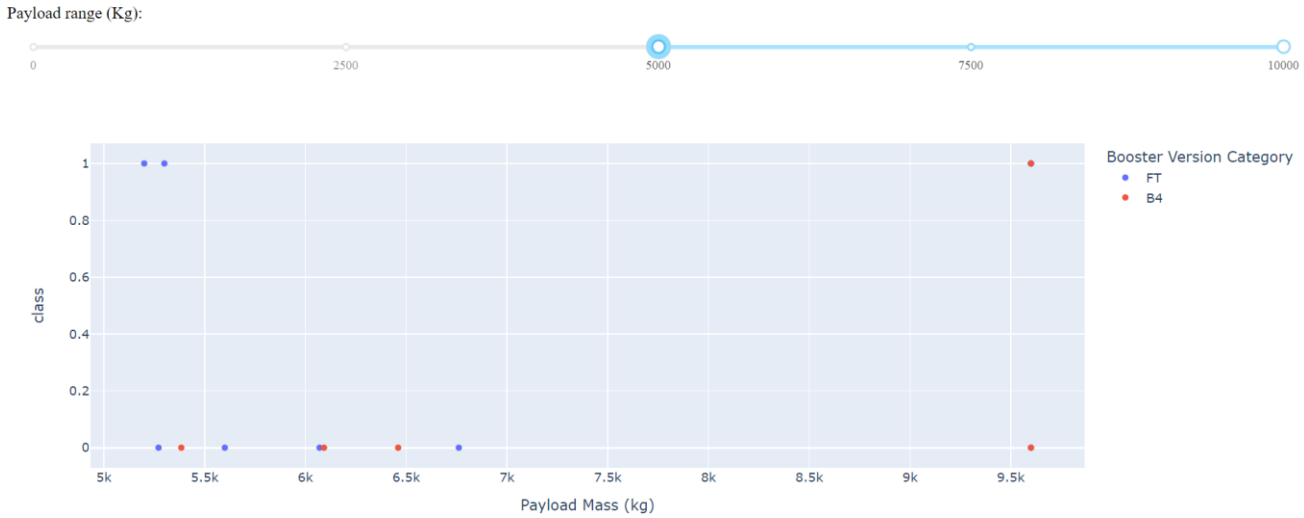


- The Launch site KSC LC-39A with the highest launch success ratio, has 76.9% success rate and 23.1% failed rate

Launch Site with Highest Launch Success ratio



- With Payload Mass of smaller than 5000, the Booseter Version v1.0 has the highest success rate



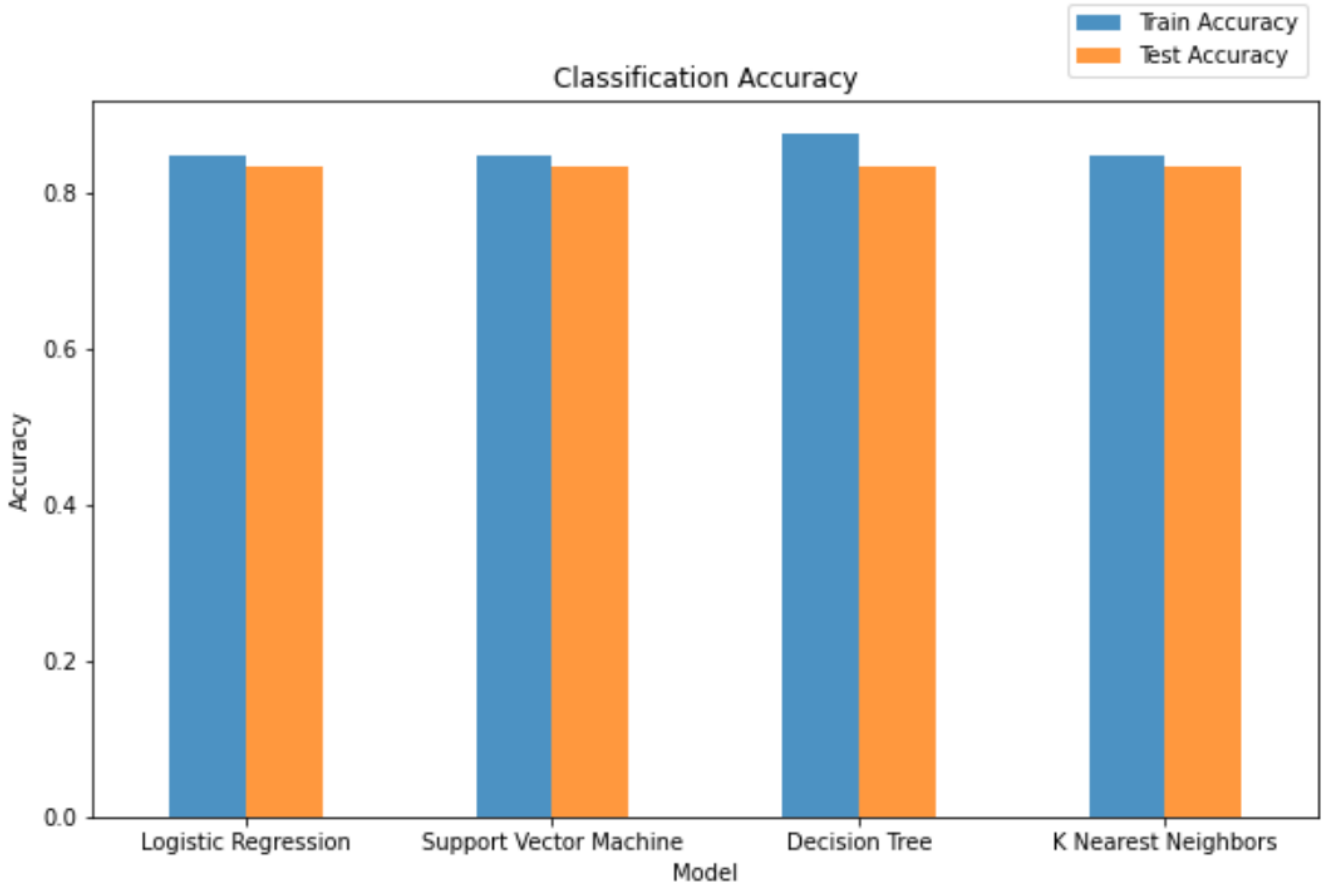
- For high Payload Mass, the booster FT and B4 has the highest success rate

Section 6

Predictive Analysis (Classification)

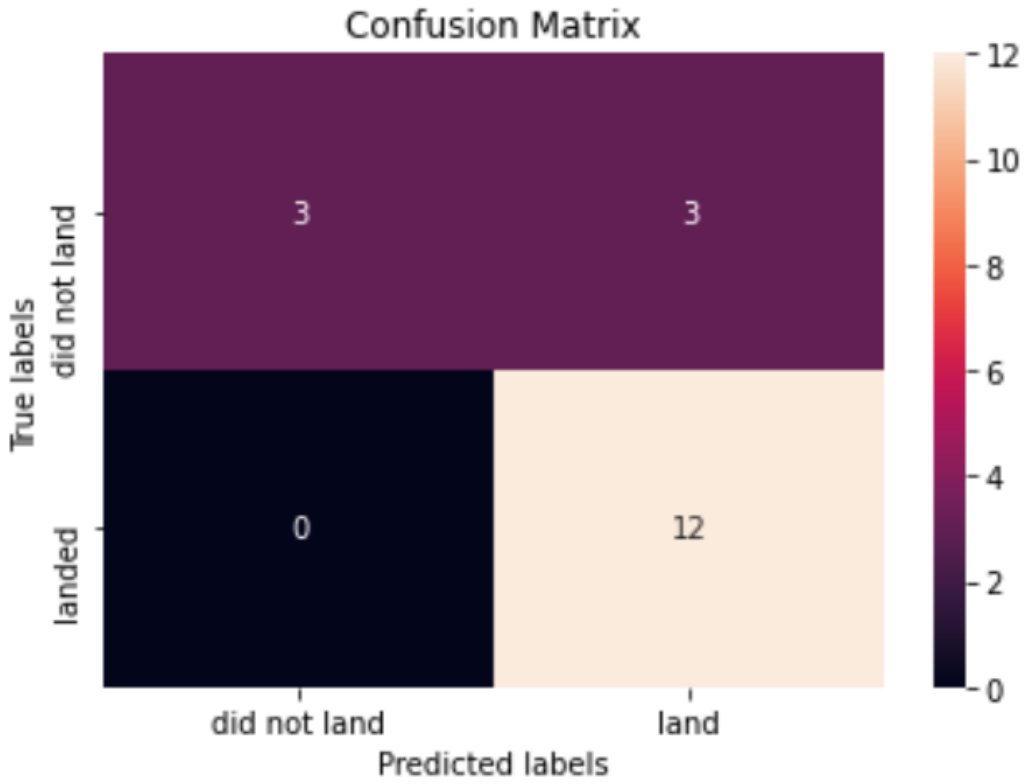
Classification Accuracy

- All algorithms has the same Test Accuracy of 0.83
- The Decision Tree method has the highest Train Accuracy of 0.875



Confusion Matrix

A common issue with all the models is the False Positive. In another words, the model predicted that the outcome is "land" but, it was "Did not land"



Conclusions

- Based on the train accuracy and the test accuracy, the "Decision Tree" model is choosen as the classification model. However, more data is needed to make the final decision.
- The KSC LC-39A has the highest success rate
- For high Payload Mass, the booster FT and B4 has the highest success rate
- ISS orbit exhibits a proper outcome for high Payloads
- As the number of flights increase, the ratio of failing is decreseing. It indicates that it is highly possible to have more successful missions in the future.

Appendix

- Wikipedia Page
  https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

- GitHub Page:
  https://github.com/YasharShabbouei/SpaceX-Data-Scinece-Capstone

Thank you!