IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Krzysztof Szczypkowski

19/11/2022

Outline

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

Executive Summary

- Summary of methodologies

Data Collection via API and Web Scraping

Data Wrangling, Exploratory Data Analysis via Visualization and SQL

Interactive Visual Analytics via Folium and Plotly Dash

Predictive Analysis via Classification Models (Logistic Regression, Support Vector Machine, Decision Tree, K-Nearest Neighbors)

- Summary of all results

EDA results, interactive analytics, predictive analytics

Introduction

- Project background and context

SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage.

- Problems you want to find answers

Trying to predict if the 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. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

Section 1

Methodology

Methodology

Executive Summary

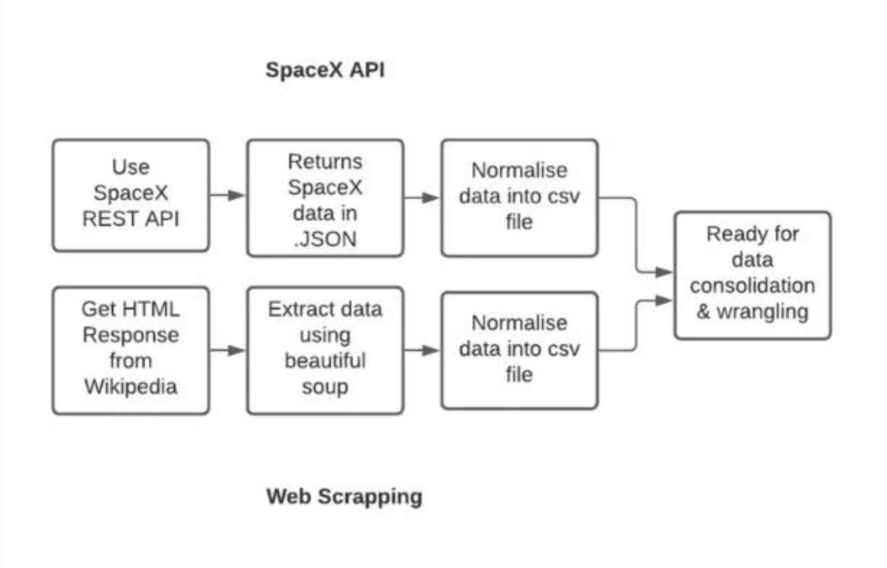
- Data collection methodology:
  - SpaceX API
  - Web Scraping Wikipedia
- Perform data wrangling
  - Collected data was enriched by creating a landing outcome label based on outcome data after summarizing and analyzing features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Logistic Regression, Support Vector Model, Decision Tree & K Nearest Neighbours Model

Data Collection

- Data was collected from SpaceX REST API:

1. Rockets
2. Launchpads
3. Payloads
4. Cores
5. Launches

- Web Scrapped data from Wikipedia



SpaceX API

Use SpaceX REST API → Returns SpaceX data in .JSON → Normalise data into csv file → Ready for data consolidation & wrangling

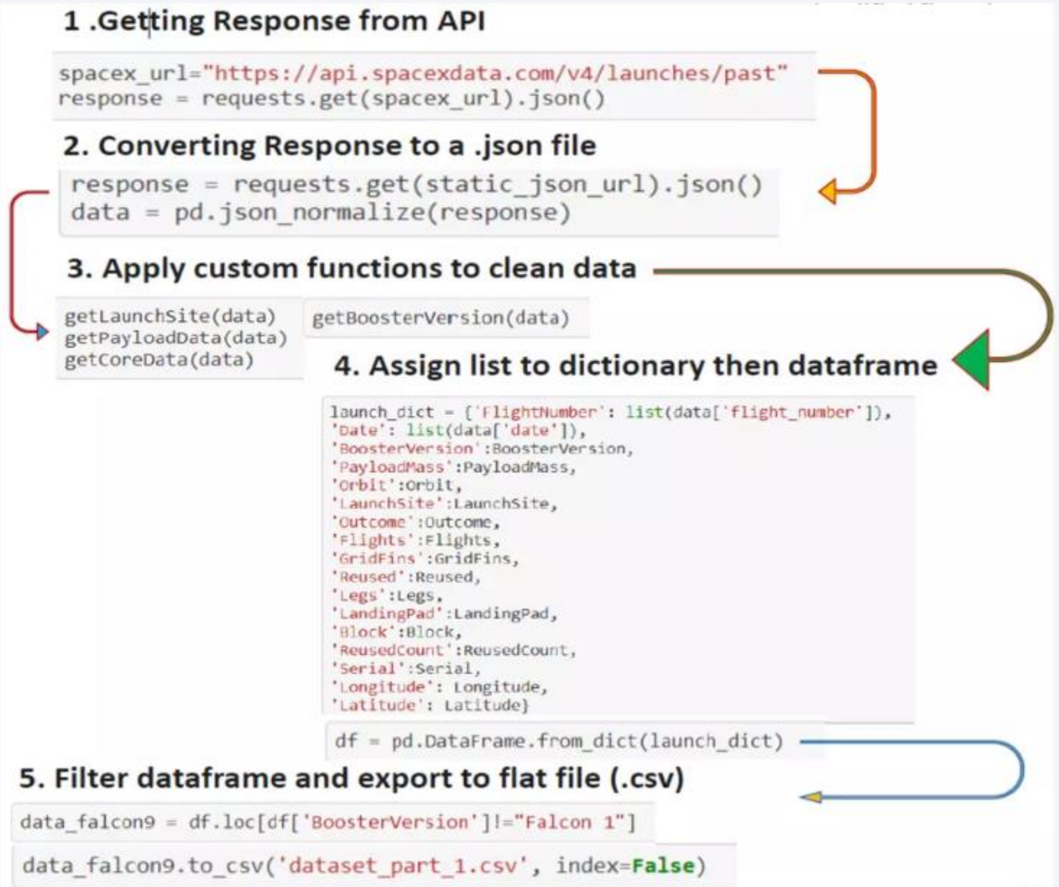
Get HTML Response from Wikipedia → Extract data using beautiful soup → Normalise data into csv file → Ready for data consolidation & wrangling

Web Scrapping

Data Collection – SpaceX API

- https://github.com/fufuthesloth/DSML_Capstone_Project/blob/main/Data_Collection_API.ipynb

- Data collection with SpaceX REST API calls



1 .Getting Response from API

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

2. Converting Response to a .json file

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

3. Apply custom functions to clean data

```
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

```
getBoosterVersion(data)
```

4. Assign list to dictionary then dataframe

```
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}
```

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

5. Filter dataframe and export to flat file (.csv)

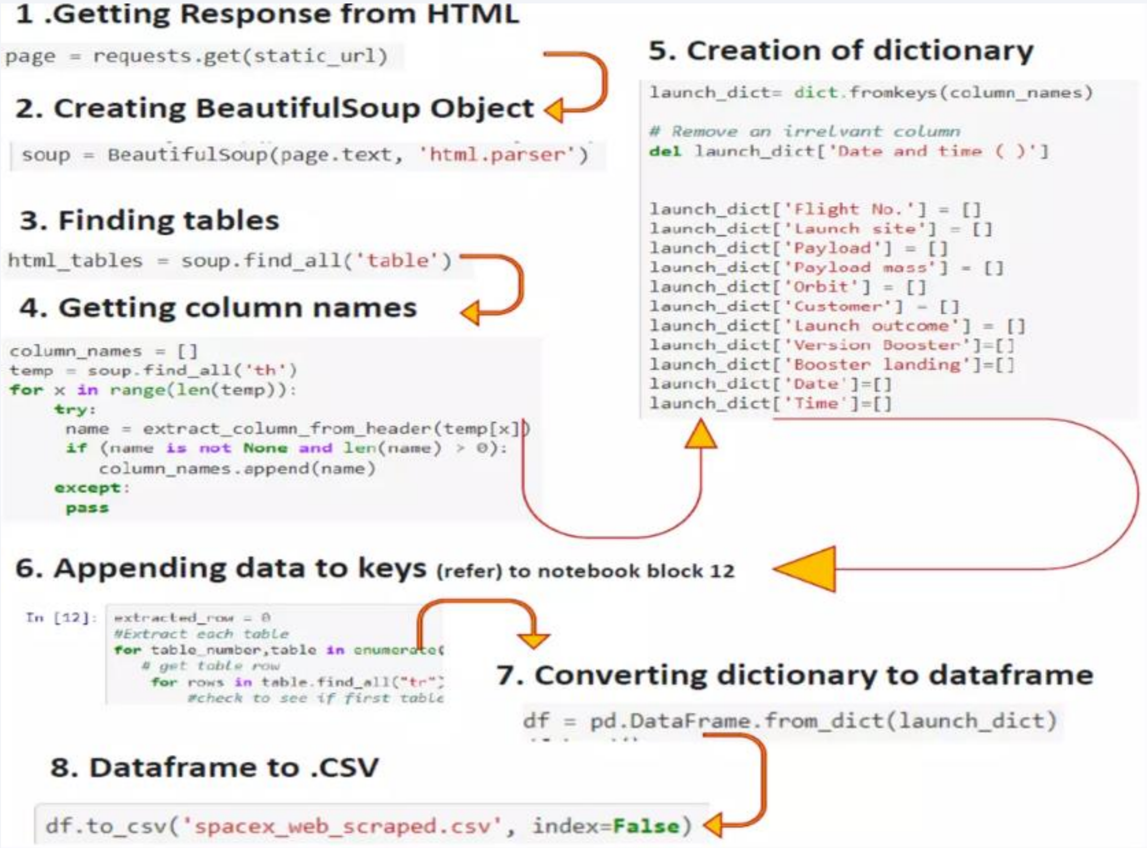
```
data_falcon9 = df.loc[df['BoosterVersion']!="Falcon 1"]
```

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

Data Collection - Scraping

- https://github.com/fufutheslot h/DSML_Capstone_Project/blob/main/Data_Collection_with_Web_Scraping.ipynb

- Web Scrapping data from Wikipedia



1 .Getting Response from HTML

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

2. Creating BeautifulSoup Object

```
soup = BeautifulSoup(page.text, 'html.parser')
```

3. Finding tables

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

4. Getting column names

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

5. Creation of dictionary

```
launch_dict= dict.fromkeys(column_names)

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


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'] = []
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6. Appending data to keys (refer) to notebook block 12

```
In [12]: extracted_row = 0
         #Extract each table
         for table_number,table in enumerate(
            # get table row
             for rows in table.find_all("tr"
                 #check to see if first table
```

7. Converting dictionary to dataframe

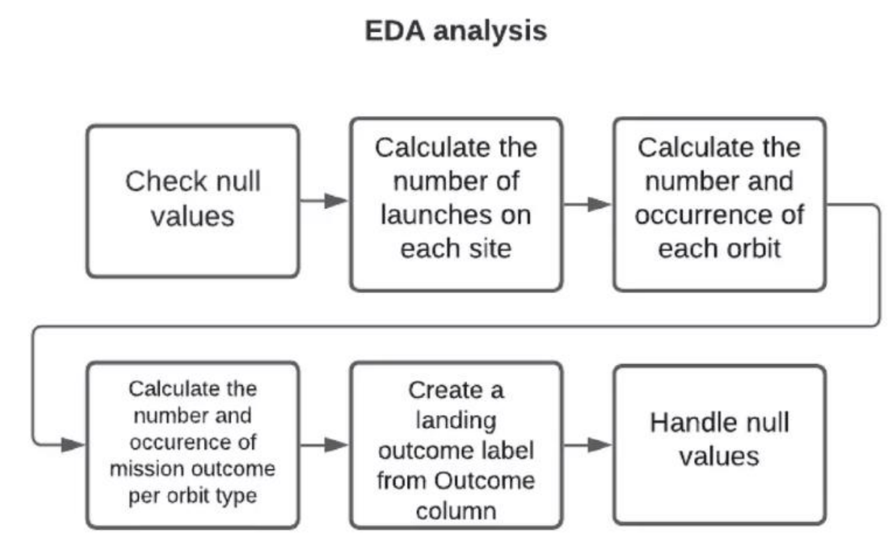
```
df = pd.DataFrame.from_dict(launch_dict)
```

8. Dataframe to .CSV

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

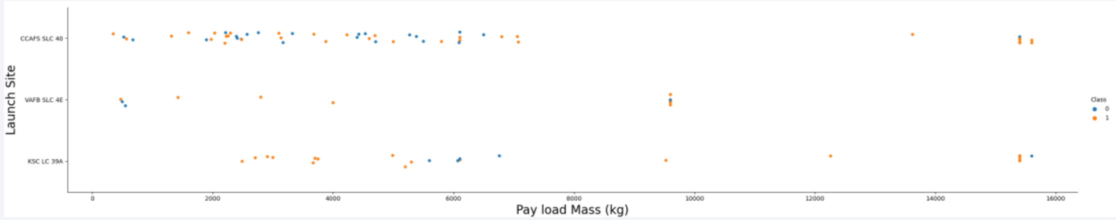
Data Wrangling

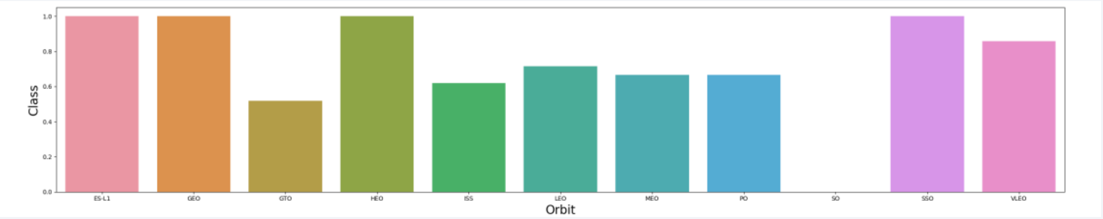
- https://github.com/fufuthesloth/DSML_Capstone_Project/blob/main/Data_Wrangling.ipynb

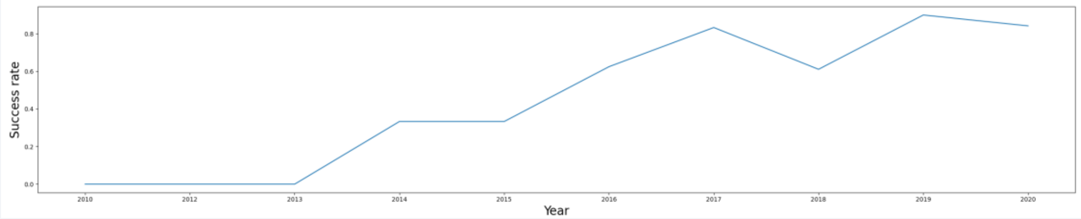

EDA analysis
Check null values
Calculate the number of launches on each site
Calculate the number and occurrence of each orbit
Calculate the number and occurence of mission outcome per orbit type
Create a landing outcome label from Outcome column
Handle null values

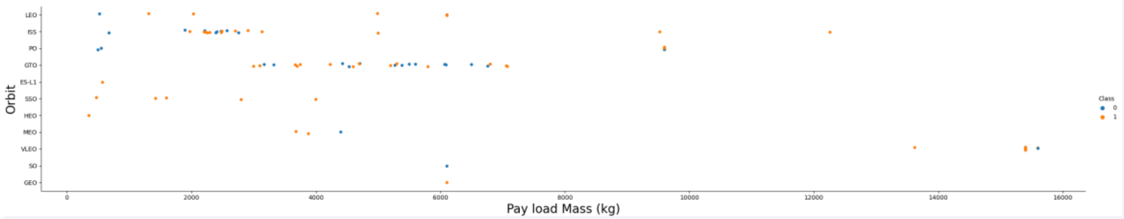
EDA with Data Visualization

- https://github.com/fufuthesloth/DSML_Capstone_Project/blob/main/EDA_with_Visualization.ipynb







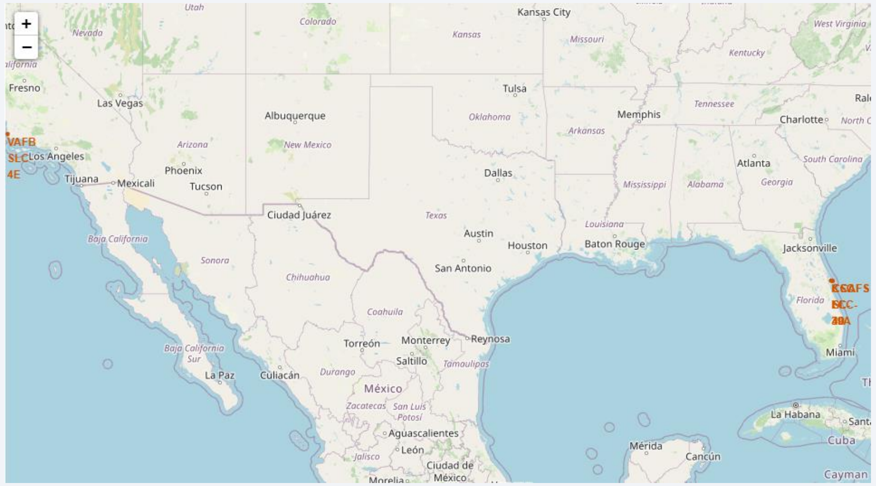


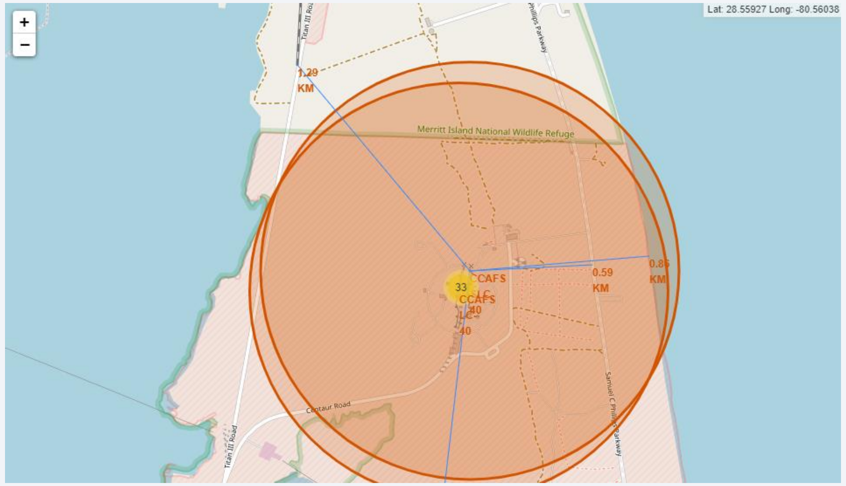
EDA with SQL

- **https://github.com/fufuthesloth/DSML_Capstone_Project/blob/main/EDA_with_SQL.ipynb**
- Display the names of the unique launch sites in the space mission.
- Display 5 records where launch sites begin with the string 'KSC'.
- 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 where the first successful landing outcome in drone ship was achieved.
- List the names of the boosters which have success in ground pad 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 records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017.
- Rank the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

Build an Interactive Map with Folium

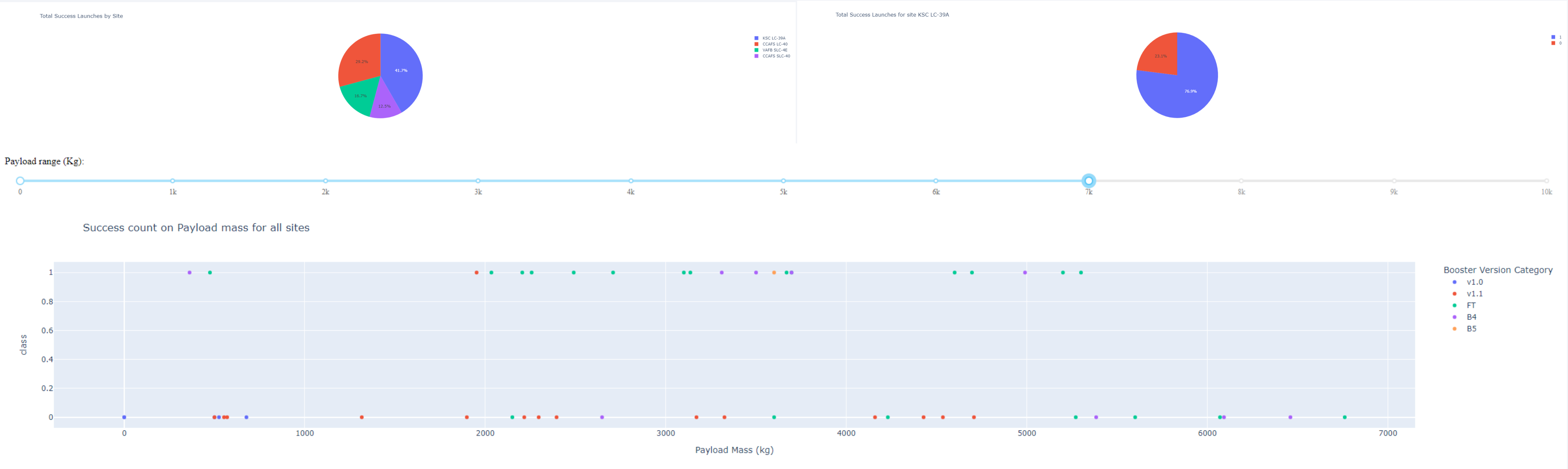
- https://github.com/fufuthesloth/DSML_Capstone_Project/blob/main/Interactive_Visual_Analytics_with_Folium.ipynb
- Marked launch sites on the map, added markers with successful/failed launches and then measured the distance to infrastructure such as railway, highway, coastline and cities in nearby proximity




+
−
Lat: 28.55927 Long: -80.56038
1.29 KM
Merritt Island National Wildlife Refuge
0.59 KM
0.86 KM
33
CCAFS LC-40
CCAFS LC-40
Titan III Road
Centaur Road
Samuel C Phillips Parkway

Build a Dashboard with Plotly Dash

- https://github.com/fufuthesloth/DSML_Capstone_Project/blob/main/spacex_dash_app.py

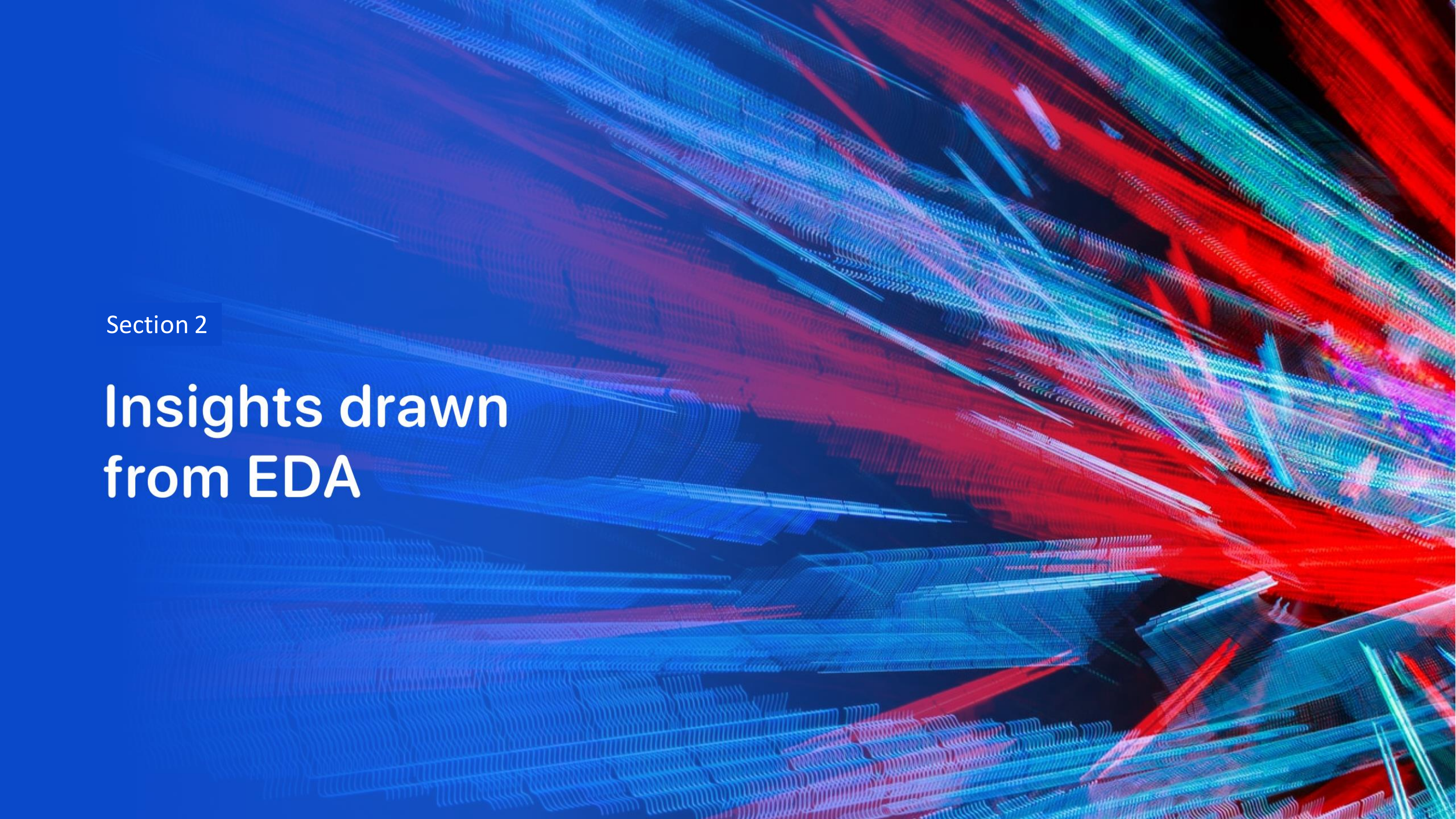


Predictive Analysis (Classification)

- https://github.com/fufuthesloth/DSML_Capstone_Project/blob/main/Machine_Learning_Prediction.ipynb
- Loaded and transformed data using NumPy and Pandas
- Split data into training and testing set
- Built different Machine Learning models and found hyperparameters using GridSearchCV
- Used feature engineering and algorithm tuning to improve the model
- Measured accuracy of the models
- Found best performing classification model

Results

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

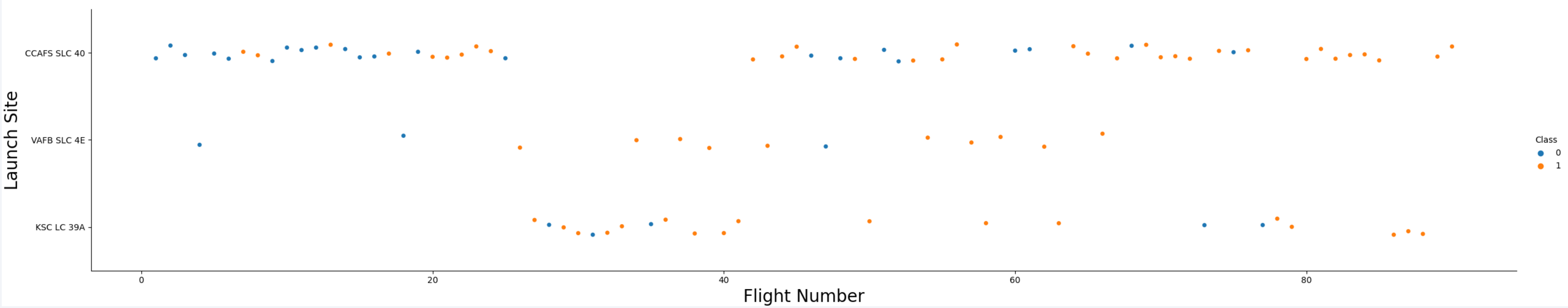


Section 2

Insights drawn from EDA

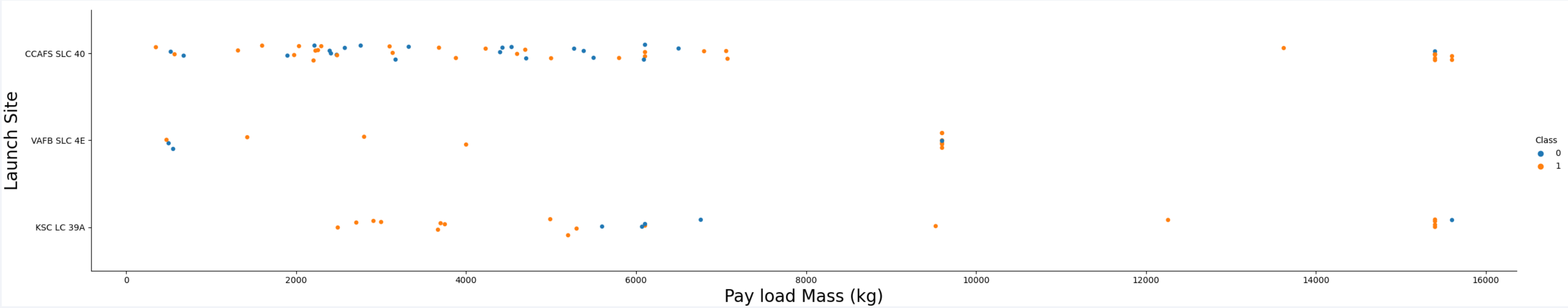
Flight Number vs. Launch Site

- From the scatter plot Flight Number vs. Launch Site, we can observe that Launch Site CCAFS SLC 40 has significantly higher amount of launches compare to other sites.



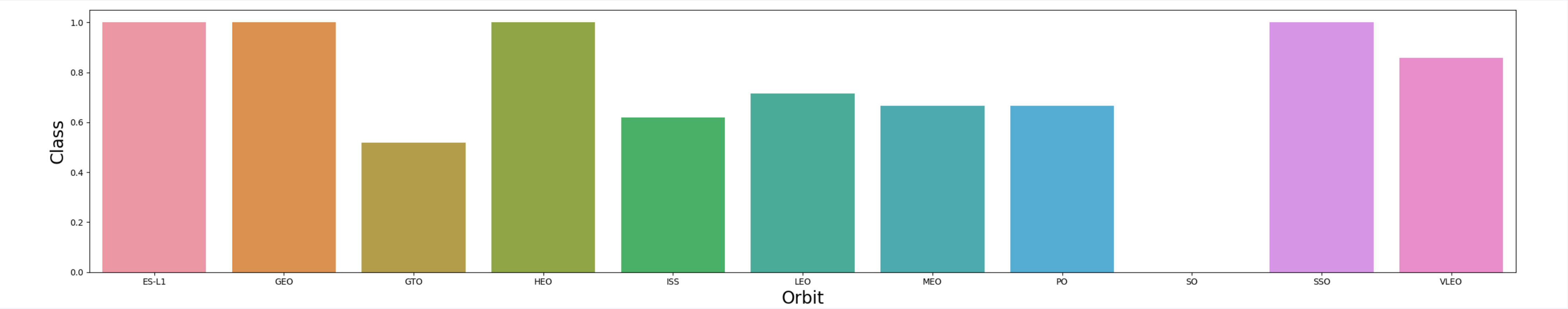
Payload vs. Launch Site

- On the scatter plot Payload vs. Launch Site chart, we can see that for the VAFB SLC 4E launch site, there are no rockets launched for heavy payload mass (greater than 10000).
- Additionally we can observe that majority of flights with lighter payloads are from CCAFS SLC 40 launch site.



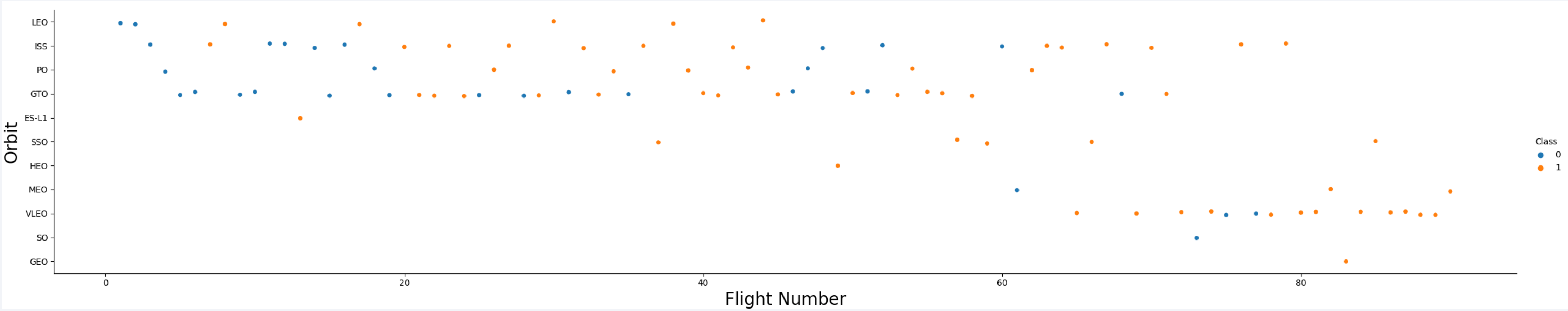
Success Rate vs. Orbit Type

- From the bar chart, we can observe that orbits ES-L1, GEO, HEO, SSO and VLEO have the highest success rate.



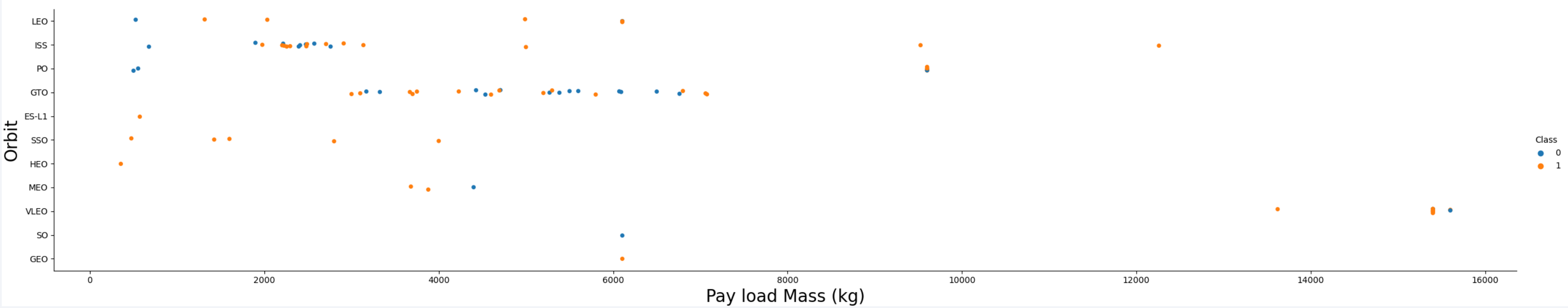
Flight Number vs. Orbit Type

- In the LEO orbit the success rate appears to be related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.
- The focus over the years seems to be changing towards launches to VLEO orbit.



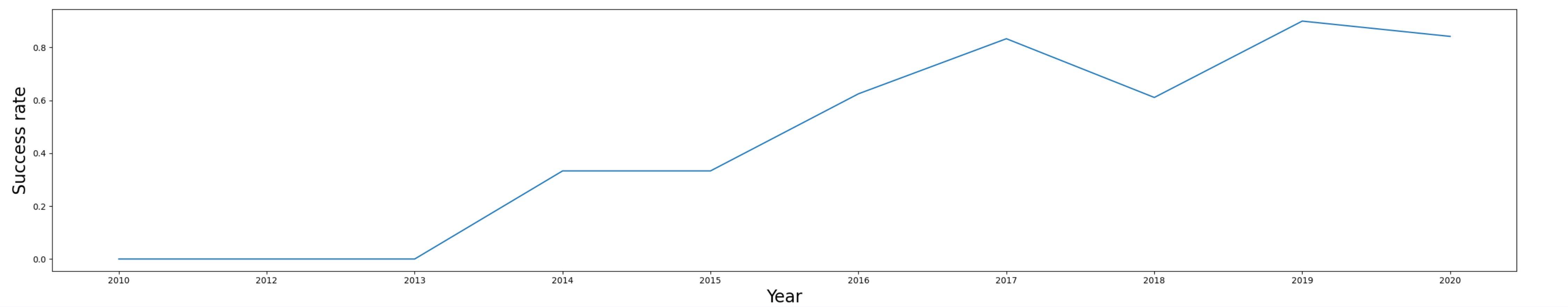
Payload vs. Orbit Type

- There is strong correlation between payloads around 2000kg and ISS orbit.
- Payloads for GTO orbit seems to be ranging from 3000 to 7000kg.
- There are only heavy payloads for VLEO orbit.



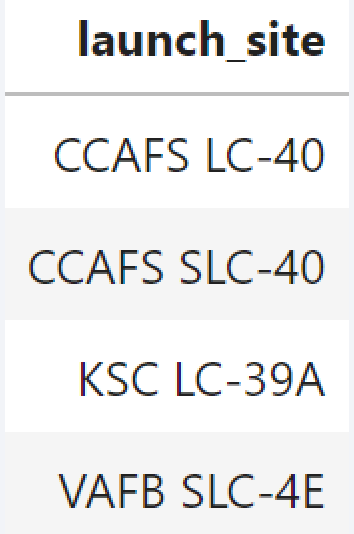
Launch Success Yearly Trend

- Success rate kept increasing since 2013 until 2020.



All Launch Site Names

- SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXDATASET
- Used SELECT DISTINCT statement to return only distinct (different) values which returned names of 4 unique launch sites.

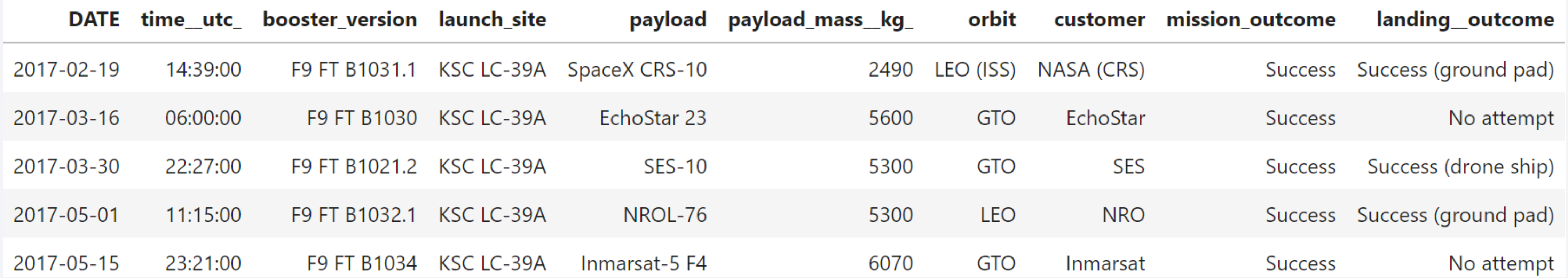


launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Launch Site Names Begin with 'KSC'

- SELECT * FROM SPACEXDATASET WHERE LAUNCH_SITE LIKE 'KSC%' LIMIT 5

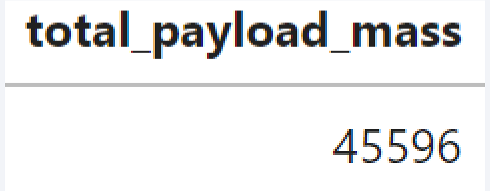
- Used statement LIKE 'KSC%' to return results only from launch sites starting with "KSC" and then limited results to 5 with LIMIT statement.



DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-03-16	06:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
2017-05-15	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

Total Payload Mass

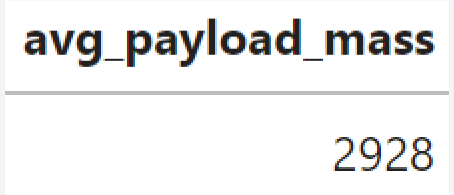
- SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD_MASS FROM SPACEXDATASET WHERE CUSTOMER = 'NASA (CRS)'
- Calculated total payload mass (in KG) via SUM statement and formatted it by adding WHERE statement to return result only for boosters from NASA.



total_payload_mass
45596

Average Payload Mass by F9 v1.1

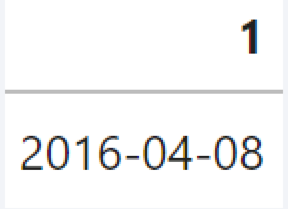
- SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS FROM SPACEXDATASET WHERE BOOSTER_VERSION = 'F9 v1.1'
- Calculated average payload mass (in KG) via AVG statement and formatted it by adding WHERE statement to return result only for booster version F9 v1.1.



avg_payload_mass
2928

First Successful Ground Landing Date

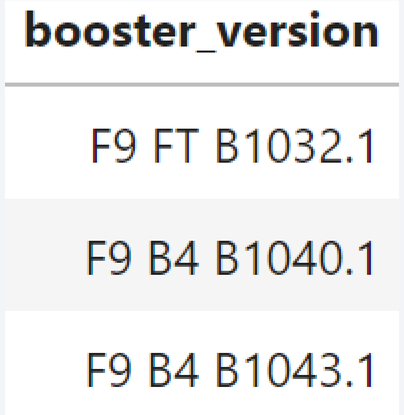
- SELECT MIN(DATE) FROM SPACEXDATASET WHERE LANDING__OUTCOME = 'Success (drone ship)'
- Used MIN statement on DATE column and formatted it via WHERE statement to find the date of first successful landing outcome on drone ship.



1
2016-04-08

Successful Drone Ship Landing with Payload between 4000 and 6000

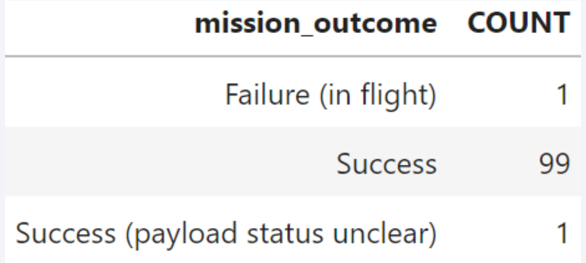
- SELECT BOOSTER_VERSION FROM SPACEXDATASET WHERE LANDING__OUTCOME = 'Success (ground pad)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
- Used WHERE, AND and BETWEEN statements to return list of the names of boosters which have successfully landed on ground pad and had payload mass greater than 4000 but less than 6000.



booster_version
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1

Total Number of Successful and Failure Mission Outcomes

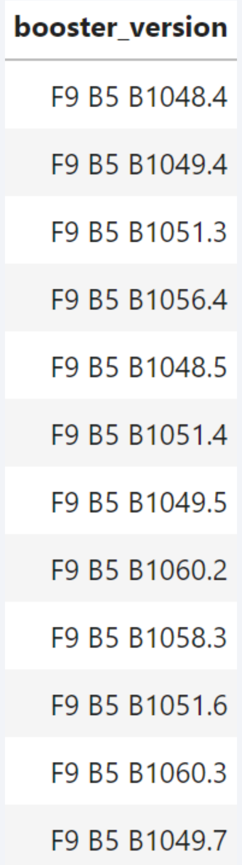
- SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS COUNT FROM SPACEXDATASET GROUP BY MISSION_OUTCOME
- Used GROUP BY statement to calculate total number of successful and failure mission outcomes.



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

Boosters Carried Maximum Payload

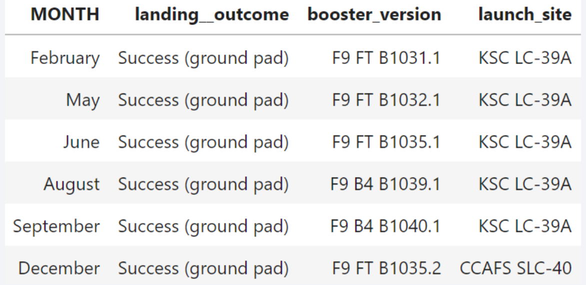
- SELECT BOOSTER_VERSION FROM SPACEXDATASET WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXDATASET)
- Used sub query with MAX statement in WHERE statement to determinate maximum payload mass and to list the names of the boosters which have carried that mass.



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

2017 Launch Records

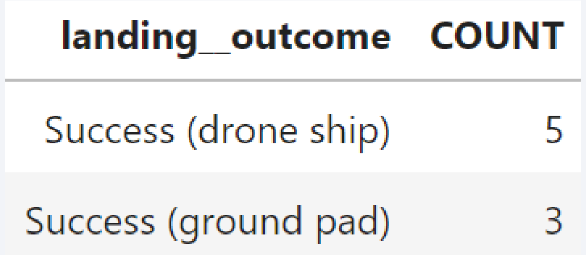
- SELECT MONTHNAME(DATE) AS MONTH, LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXDATASET WHERE LANDING__OUTCOME = 'Success (ground pad)' AND YEAR(DATE) = '2017'
- Used MONTHNAME and YEAR statements to extract relevant data and return the records which display the month names, successful landing outcomes in ground pad, booster versions and launch site for the months in year 2017



MONTH	landing__outcome	booster_version	launch_site
February	Success (ground pad)	F9 FT B1031.1	KSC LC-39A
May	Success (ground pad)	F9 FT B1032.1	KSC LC-39A
June	Success (ground pad)	F9 FT B1035.1	KSC LC-39A
August	Success (ground pad)	F9 B4 B1039.1	KSC LC-39A
September	Success (ground pad)	F9 B4 B1040.1	KSC LC-39A
December	Success (ground pad)	F9 FT B1035.2	CCAFS SLC-40

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

- SELECT LANDING__OUTCOME, COUNT(LANDING__OUTCOME) AS COUNT FROM SPACEXDATASET WHERE (LANDING__OUTCOME LIKE 'Success%') AND (DATE BETWEEN '2010-06-04' AND '2017-03-20') GROUP BY LANDING__OUTCOME ORDER BY COUNT(LANDING__OUTCOME) DESC
- Created complex argument using multiple statements to return the count of successful landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order



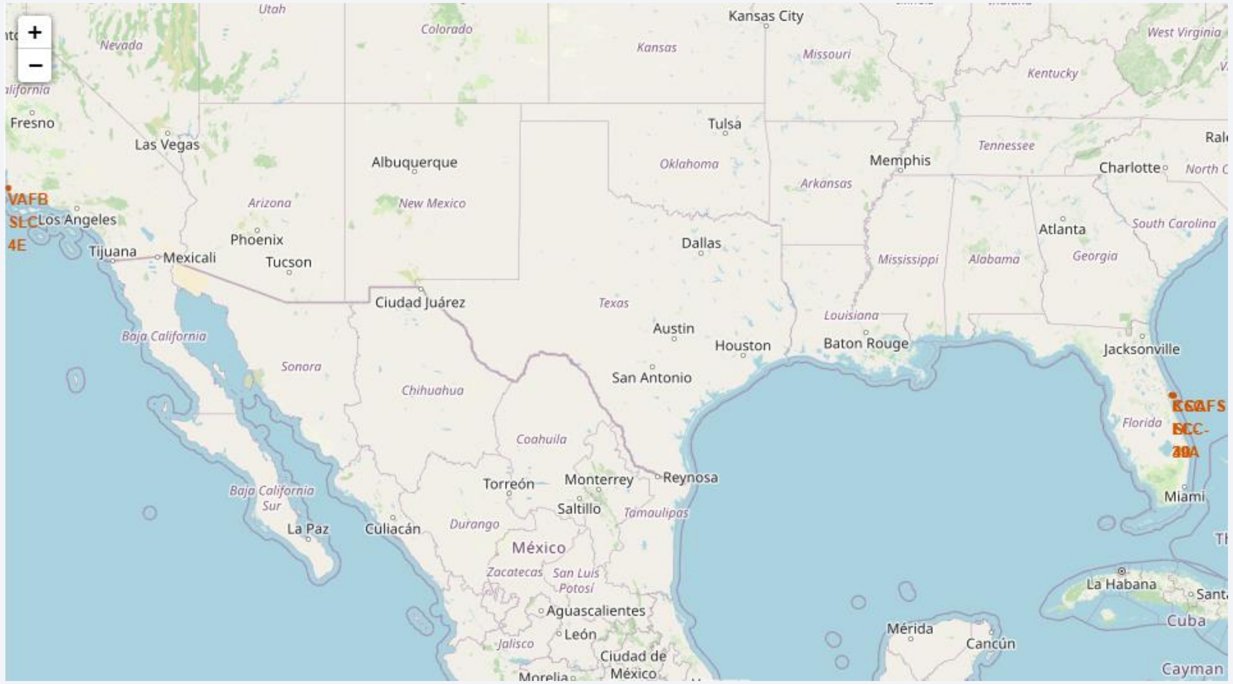
landing__outcome	COUNT
Success (drone ship)	5
Success (ground pad)	3

Section 3

Launch Sites Proximities Analysis

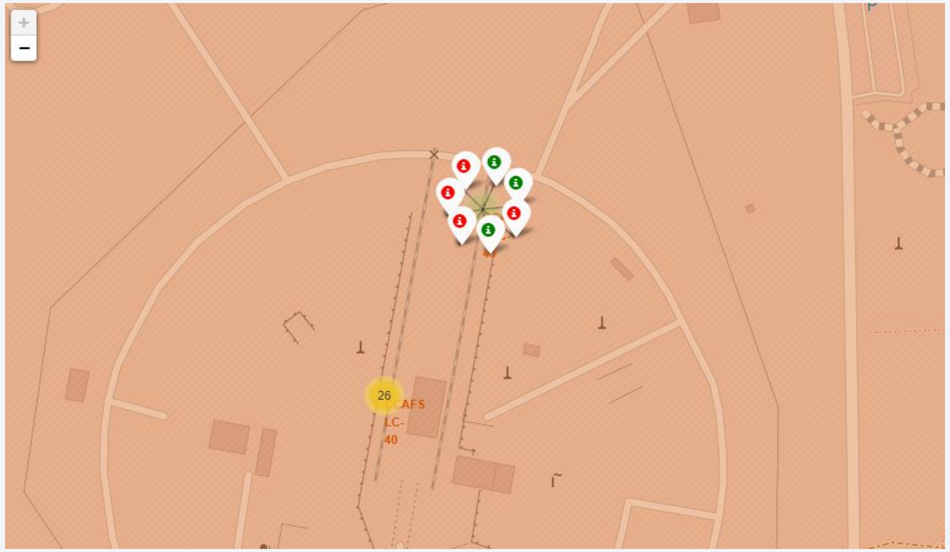
All launch sites on map

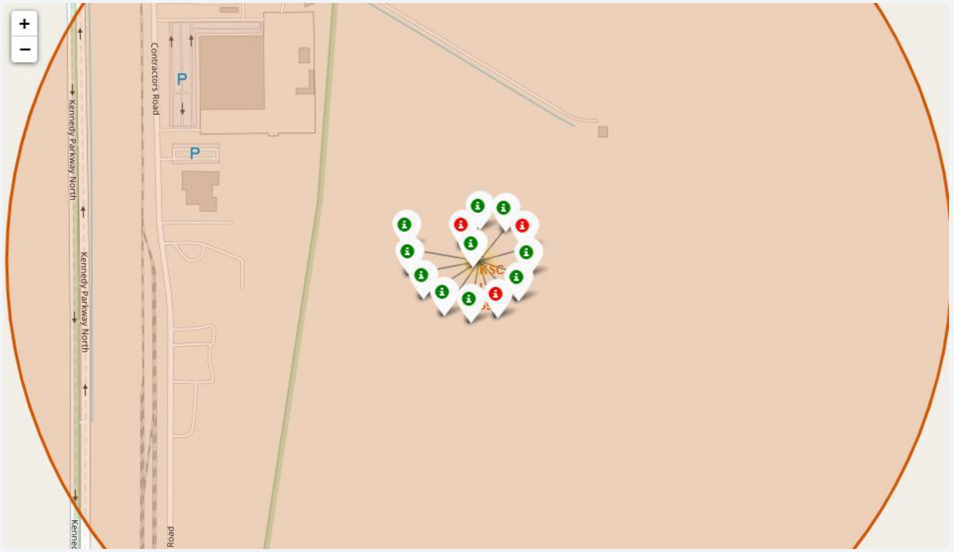
- From the map with marked all launch sites, we can observe that they are located near large bodies of water



Successful/failed launches for each site on map

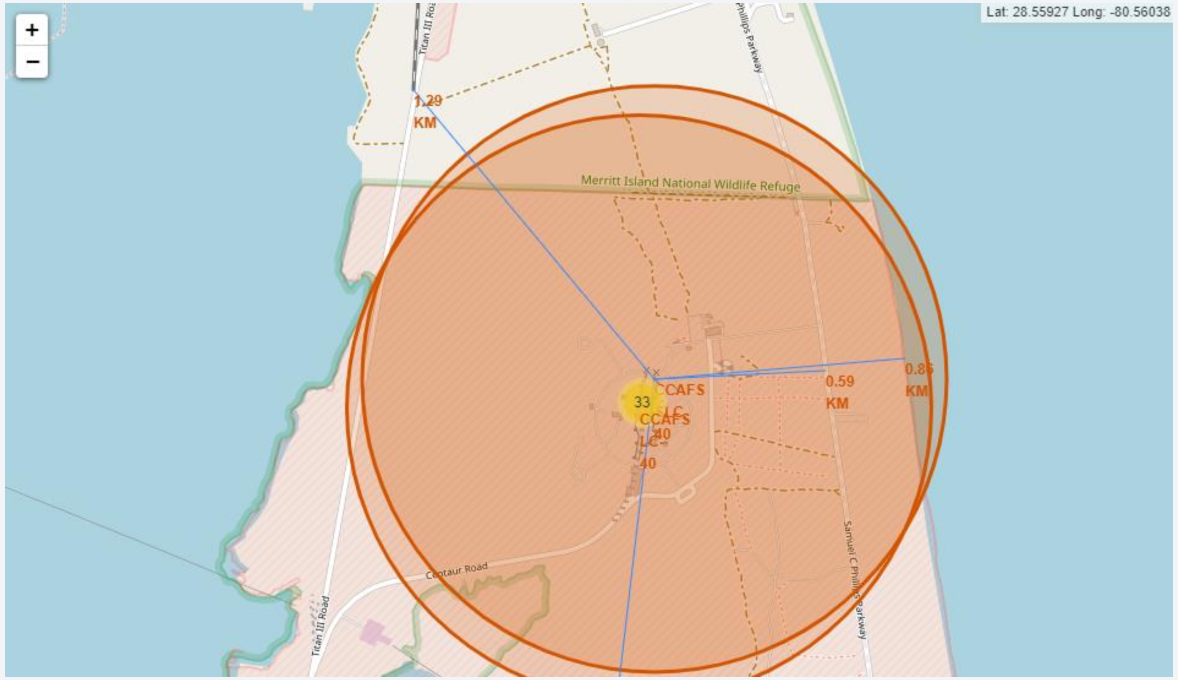
- By adding markers onto map, we can easily observe success rate of launches for each launch site





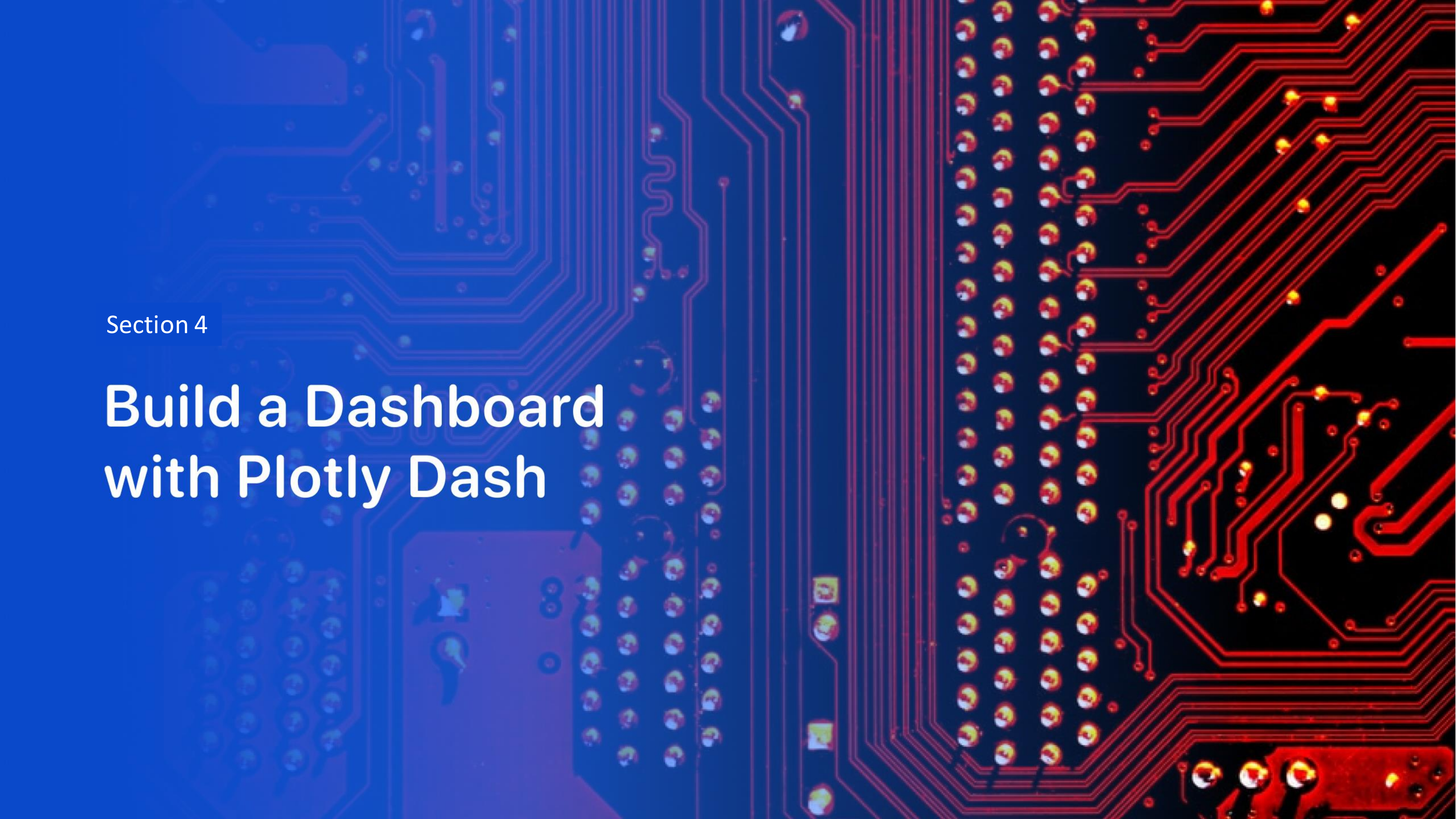
The distances between a launch site to its proximities

- By marking infrastructure near the launch site, we can observe that it's close to railway, highway, coastline and not too far from city



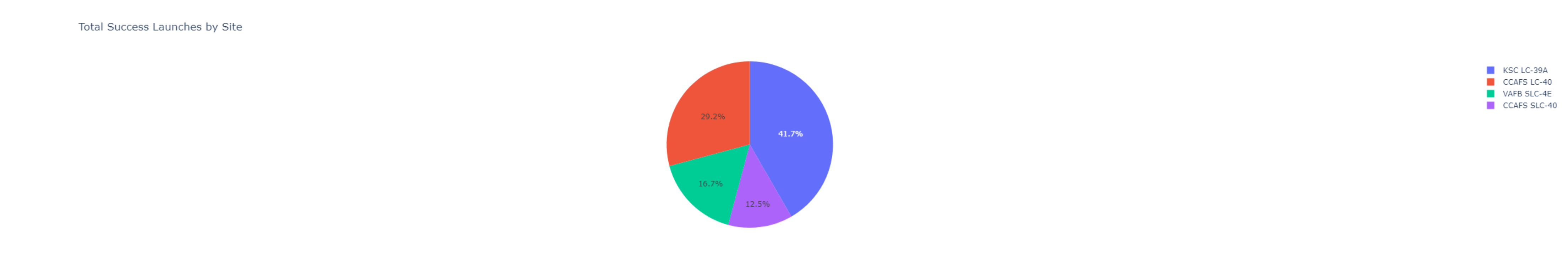
Section 4

Build a Dashboard with Plotly Dash



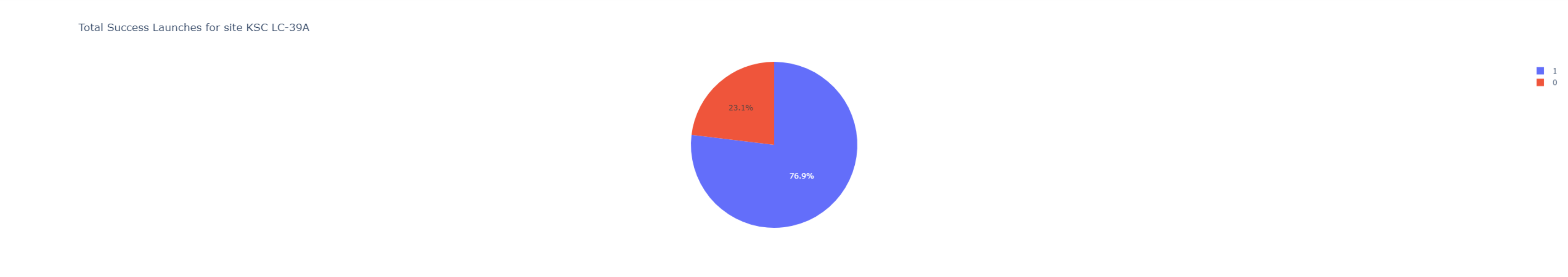
Total Success Launches by Site

- On the Total Success Launches by Site pie chart, we can observe that Launch Site KSC LC-39A has the highest success rate among all Launch Sites.



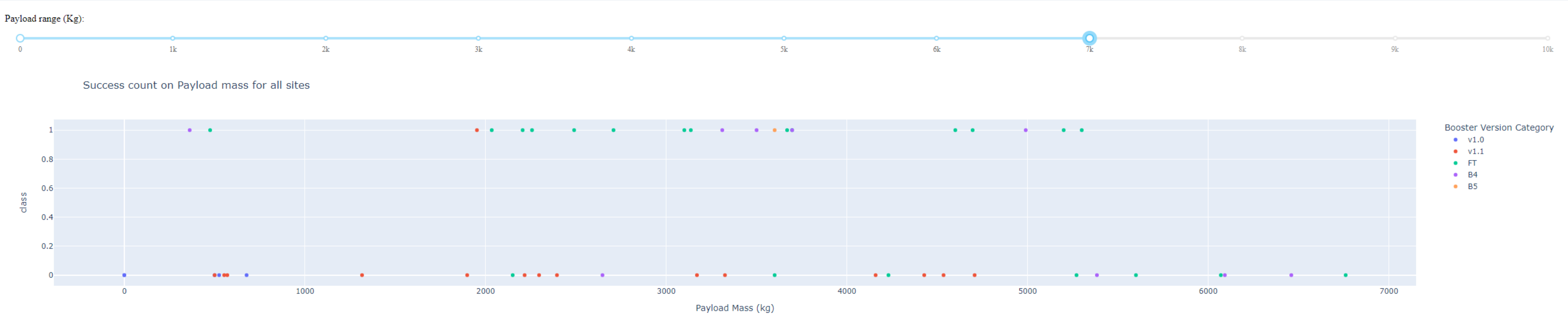
Total Success Launches for site KSC LC-39A

- On the Total Success Launches for site KSC LC-39A pie chart we can see that success rate for this Launch Site is 76.9% which means that over ¾ of launches from this site are successful.



Success count on Payload mass for all sites

- From the Payload vs. Launch Outcome scatter plot for all sites, we can deduct that launches with lighter payload have higher chance of success.

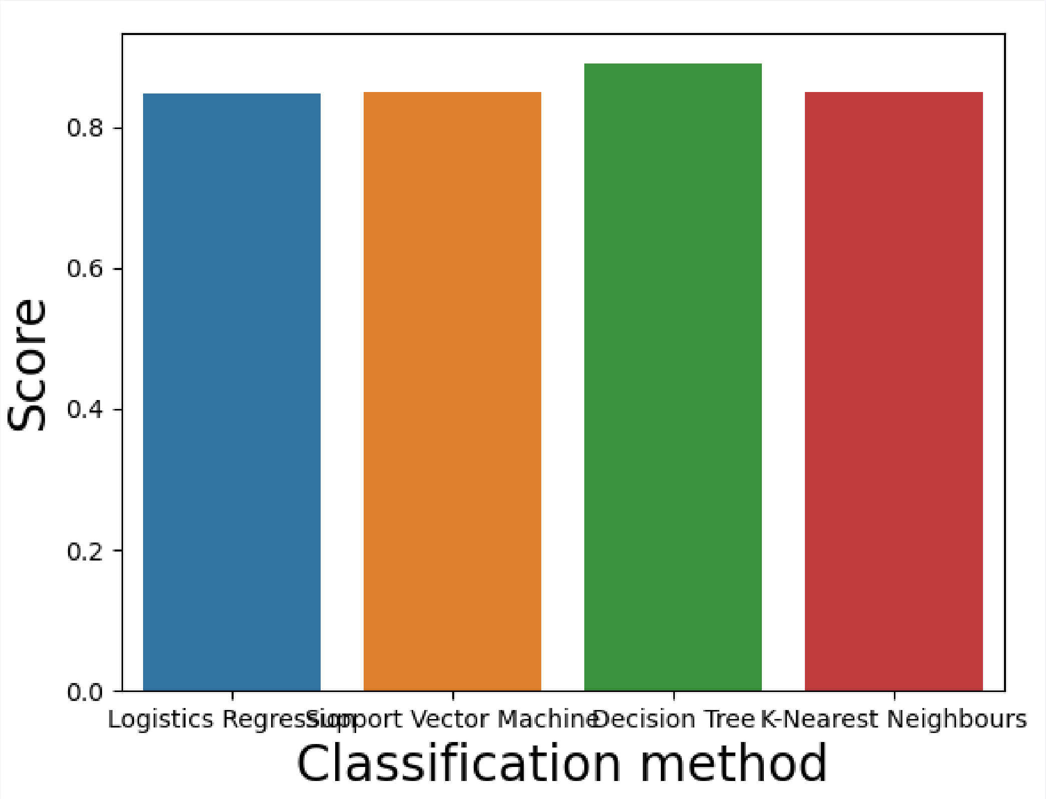


Section 5

Predictive Analysis (Classification)

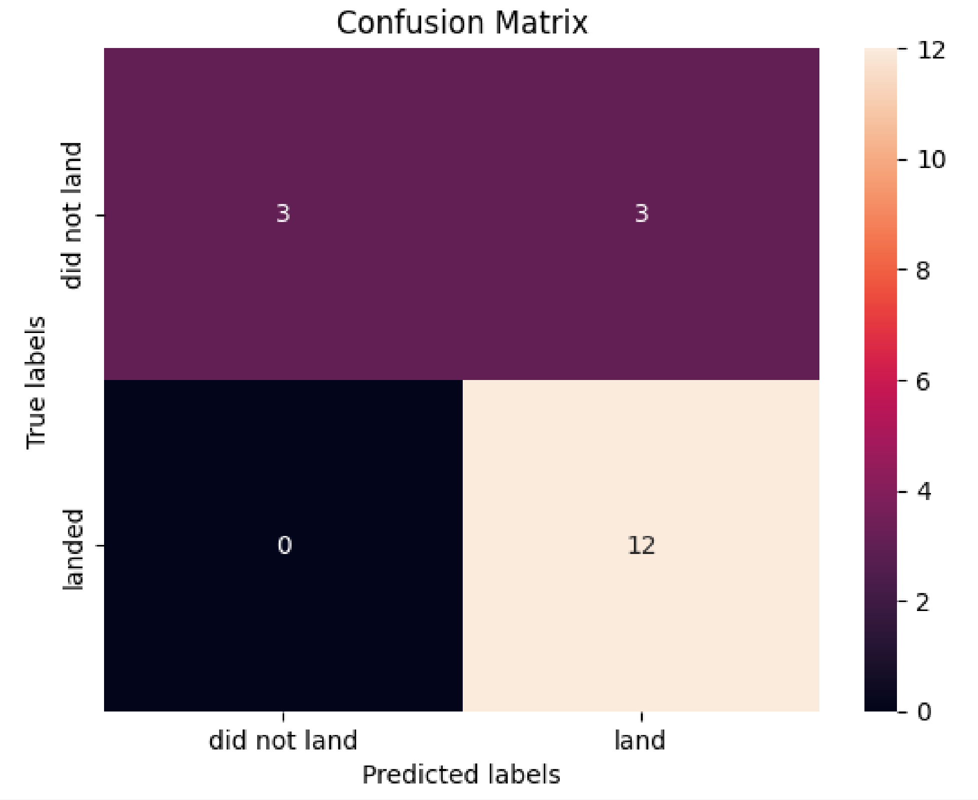
Classification Accuracy

- Predictive analysis showed that Decision Tree Classifier is the best model to predict successful landings, having accuracy over 87%.



Confusion Matrix

- Confusion matrix for Decision Tree Classifier shows that model can distinguish between different classes. The major problem is the false positives – unsuccessful landings marked as successful.



Conclusions

- Launch Site with most successful landings is KSC LC-39A
- The first success landing outcome happened in 2015, five years after the first launch.
- Two booster versions failed at landing in drone ships in 2015: F9 v1.1 B1012 and F9 v1.1 B1015.
- The number of successful landing outcomes increased over the years.
- Launch sites are often close to railway, highway, coastline and not too far from cities.
- Predictive Analysis showed that Decision Tree Classifier is the best model to predict successful landings, having accuracy over 87%.

Appendix

- https://github.com/fufuthesloth/DSML_Capstone_Project/

Thank you!