IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Victor Enriquez
May 3, 2023

Outline

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

Executive Summary

This project endeavors to develop a predictive model that can anticipate the success rate of landing SpaceX rockets. The successful implementation of this model holds immense potential for generating profits. The research process involved iterating and testing various models, ultimately resulting in the identification of the Decision Tree model as the one with the most accurate classification performance.

Based on our findings, we recommend conducting a pilot test to evaluate the effectiveness of this model at one of the best landing sites. This test will enable us to design improved rockets, suitable for the right landing site, orbit, booster, and payload. Phase 2 will involve implementing the model across all the landing sites. In case the pilot test results are unsatisfactory, we will review and re-evaluate the model.

Our research team collected and analyzed quality data, which needs to be effectively utilized. Failure to use this data would result in wastage of time and resources. The cost involved in conducting the pilot test is nominal in comparison to the long-term benefits of accurately predicting landing outcomes.

We urge you to take prompt action in planning and implementing the pilot test. A successful pilot test will position us favorably in designing advanced rockets and enhancing our landing success rate, ultimately leading to greater profitability.

Introduction

SpaceX is a private American aerospace manufacturer and space transportation services company with the primary goal of enabling people to live on other planets by making space travel more accessible, reliable, and affordable. However, the unpredictable nature of rocket landings has resulted in significant losses for the space industry. SpaceX has faced several failed landings, leading to the loss of valuable resources, time, and revenue. To mitigate these losses, it is essential to accurately predict the success of rocket landings. Therefore, the development of a predictive model that can anticipate the success rate of landing SpaceX rockets can significantly reduce losses incurred by failed landings and provide SpaceX with a competitive edge in the space industry.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Used SpaceX API to collect flight details, payload, outcome, launch site, etc
  - Used BeautifulSoup Library to collect flight data from wikipedia
- Perform data wrangling
  - Calculated the number of flights to different Orbits and number of landing outcomes
  - Calculated the the success rate
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Performed Hyperparameter for SVM, Classification Trees and Logistic Regression using GridSearchCV object and created plot of the Confusion Matrix.
  - Evaluated tuned hyperparameter for each model and best score from the training data
  - Selected the best model with by comparing the best score from the test data.

Data Collection

Request the Falcon9 Launch from Wikipedia

```
static_url =
"https://en.wikipedia.org/w/index.
php?title=List_of_Falcon_9_and_Fal
con_Heavy_launches&oldid=102768692
2"

data =
requests.get(static_url).text

soup =
BeautifulSoup(data,"html.parser")
```

Request SpaceX Launch Data

To make the requested JSON results more consistent, we will use the following static response object for this project:

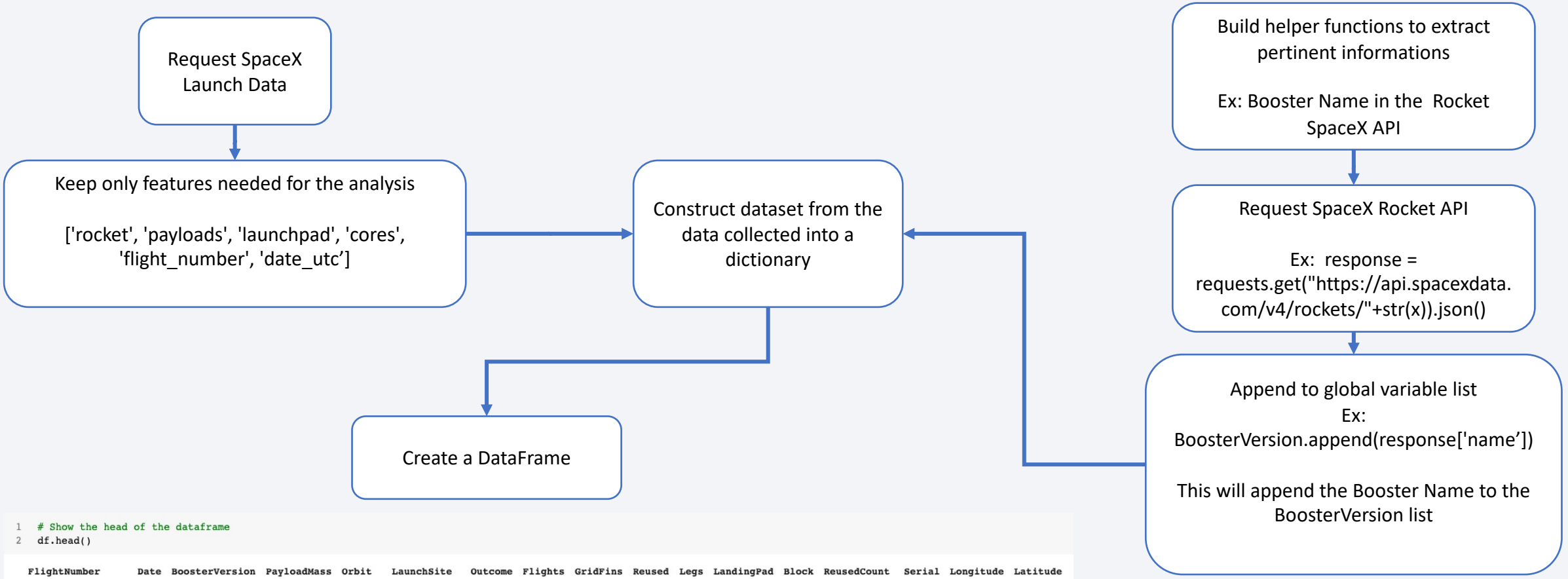
```
static_json_url='https://cf-
courses-data.s3.us.cloud-object-
storage.appdomain.cloud/IBM-
DS0321EN-
SkillsNetwork/datasets/API_call_spa
cex_api.json'

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

Request SpaceX Data

- Rocket API
- Launch Pad API
- Payload API

```
response =
requests.get("https://api.spacexdata.
com/v4/launchpads/"+str(x)).json()
```

Data Collection – SpaceX API

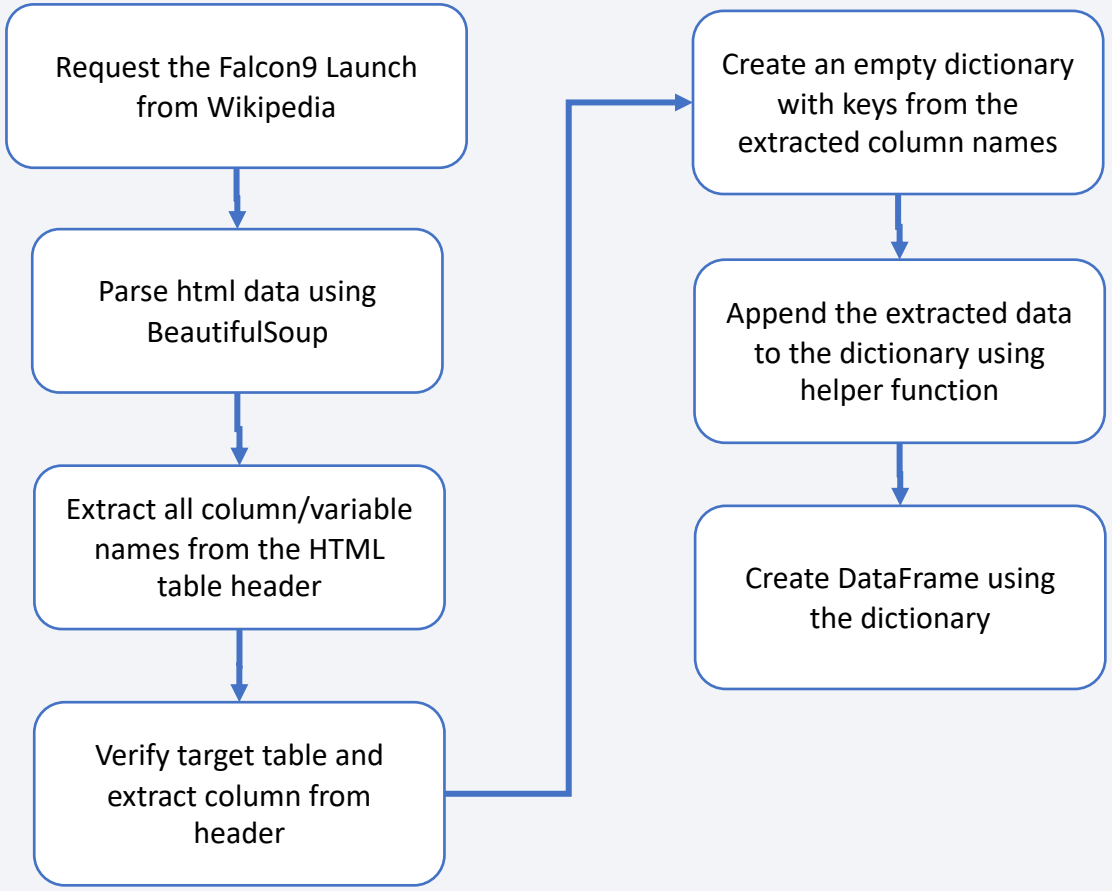



```
# Show the head of the dataframe
df.head()
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
0	1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin1A	167.743129	9.047721
1	2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin2A	167.743129	9.047721
2	4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin2C	167.743129	9.047721
3	5	2009-07-13	Falcon 1	200.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	0	Merlin3C	167.743129	9.047721
4	6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857

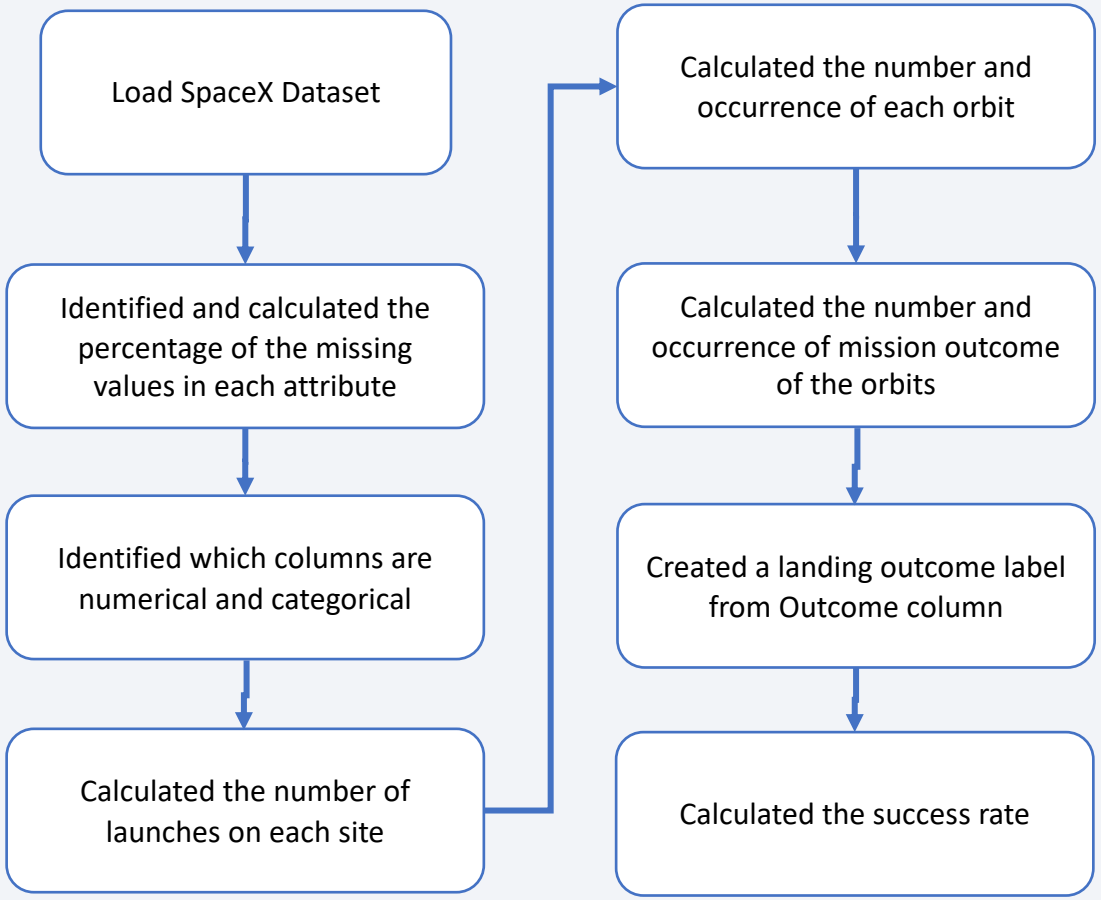
https://github.com/daVictorE/IBM_Data_Science/blob/main/10_spacex-data-collection-api.ipynb

Data Collection - Scraping



https://github.com/daVictorE/IBM_Data_Science/blob/main/10_webscraping.ipynb

Data Wrangling



https://github.com/daVictorE/IBM_Data_Science/blob/main/10_spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Catplot (Flight Number vs Payload)
  - We see that as the flight number increases, the first stage is more likely to land successfully. The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return.
- Catplot (Flight Number vs Launch Site)
  - We see that different launch sites have different success rates. Success rates are represented by the class color and values
- Scatterplot (Payload vs Launch Site)
  - We observe in the Payload Vs. Launch Site scatter point chart that VAFB-SLC Launch Site there has no rockets launched for heavy payload mass greater than 10,000.
- Barplot (Orbit and Class)
  - We see Orbits with highest success rate at ES-L1, GEO, HEO, and SSO
- Scatterplot (Flight Number vs Orbit)
  - We see that in the LEO orbit the success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.
- Scatterplot (Payload Mass vs Orbit)
  - We see that with heavy payloads Polar, LEO and ISS have positive landing rate. However, for GTO we cannot distinguish this well as both positive landing rate and negative landing.
- Line (Data and Class)
  - We observe that the success rate since 2013 kept increasing till 2020

https://github.com/daVictorE/IBM_Data_Science/blob/main/10_eda-dataviz.ipynb

EDA with SQL

- Connect SQLite3 database
  - If database does not exist, create database
  - Create a cursor (cur) object
- Read CSV data from cf-course-data using pandas read_csv method
  - Create dataframe
- Filter dataframe with landing outcome “Success (ground pad)”
- Display the names of the unique launch sites in the space mission
  - cur.execute("select distinct Launch_Site from SPACEXTBL")
- Display 5 records where launch sites begin with the string 'CCA’
  - cur.execute("select Launch_Site from SPACEXTBL where Launch_Site like 'CCA%' limit 5")
- Display the total payload mass carried by boosters launched by NASA (CRS)
  - cur.execute("select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer = 'NASA (CRS)' ")
- Display average payload mass carried by booster version F9 v1.1
  - cur.execute("select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version = 'F9 v1.1' ")
- List the date when the first successful landing outcome in ground pad was acheived.
  - cur.execute("""select Date from SPACEXTBL where "Landing _Outcome"="Success (ground pad)" """)
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - cur.execute("""select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_>4000 and PAYLOAD_MASS__KG_<6000 and "Landing _Outcome"="Success (drone ship)" """)
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - cur.execute(" select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)")
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - cur.execute(" select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)")
- 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.
  - cur.execute("""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' """)
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.
  - cur.execute("""select count("Landing _Outcome") as rank, "Landing _Outcome" from SPACEXTBL where "Landing _Outcome" like "Success%" Group by "Landing _Outcome" Order by rank desc """)

https://github.com/daVictorE/IBM_Data_Science/blob/main/10_eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Created circle with popup and marker to identify the location of NASA Johnson Space Center.
- Created circle with popup and marker to identify all four landing sites
- Built a Cluster function to identify locations of all landing outcomes. Successful outcome are shown in Green while Failure are shown in Red. The MasterCluster object prevents the overlapping of markers and provides an elegant way of dynamically aggregating the markers based on zoom level on the map.

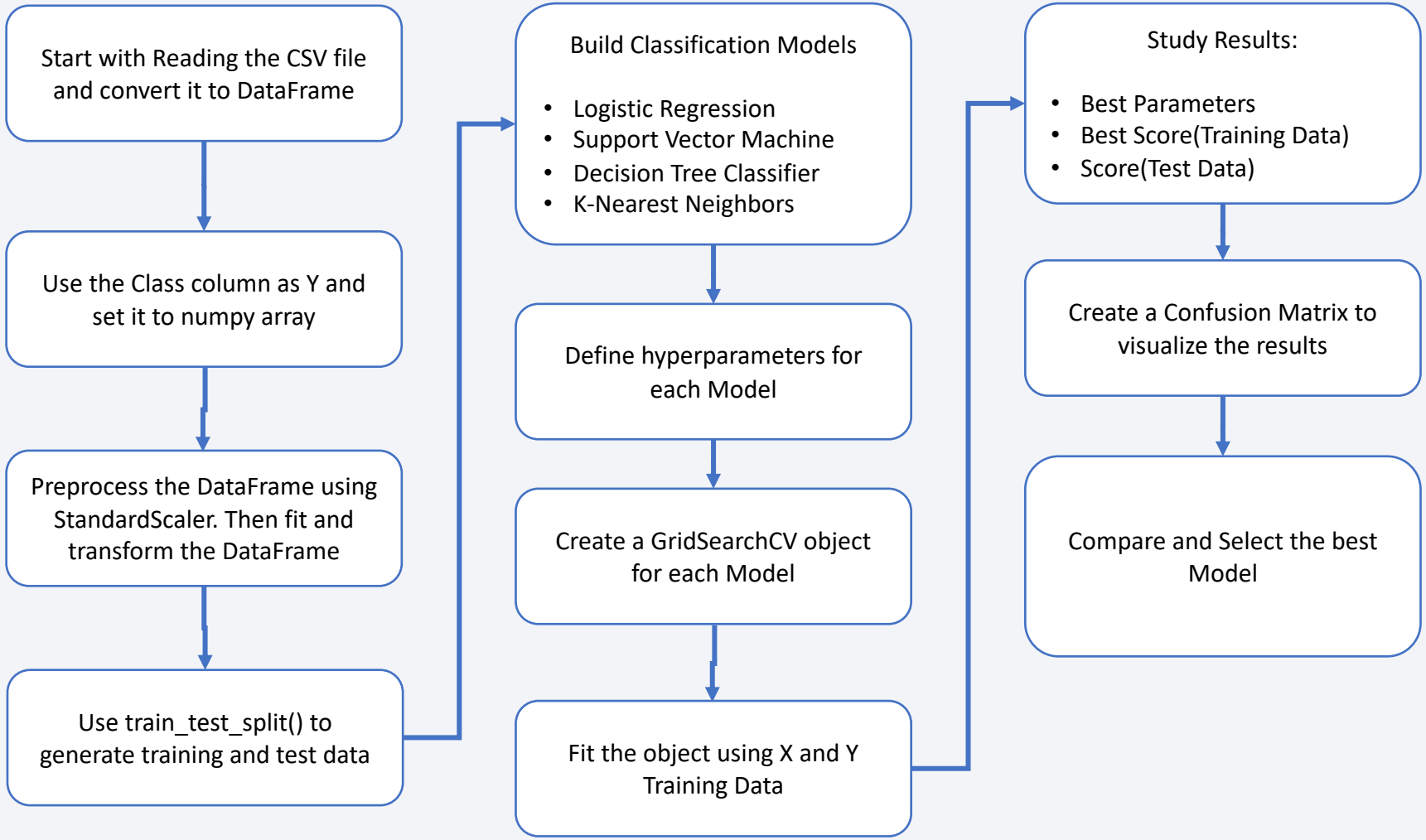
https://github.com/daVictorE/IBM_Data_Science/blob/main/10_Folium_Map.ipynb

Build a Dashboard with Plotly Dash

- The dashboard contains a Pie Chart and a Scatter Plot.
- The Pie Chart shows the percentage of successful launches by site. You can select different launch site using the dropdown menu. This provides a good visual of successful launches when comparing all sites. You can also clearly see the break down of success vs failure when selecting a specific site.
- The Scatter plot shows the number of successful and failure launches represented by the dots on the scatter plot. The color of the dots represents the booster version. You can control the payload by adjusting the range slider.

https://github.com/daVictorE/IBM_Data_Science/blob/main/spacex_dash_app%20(3).py

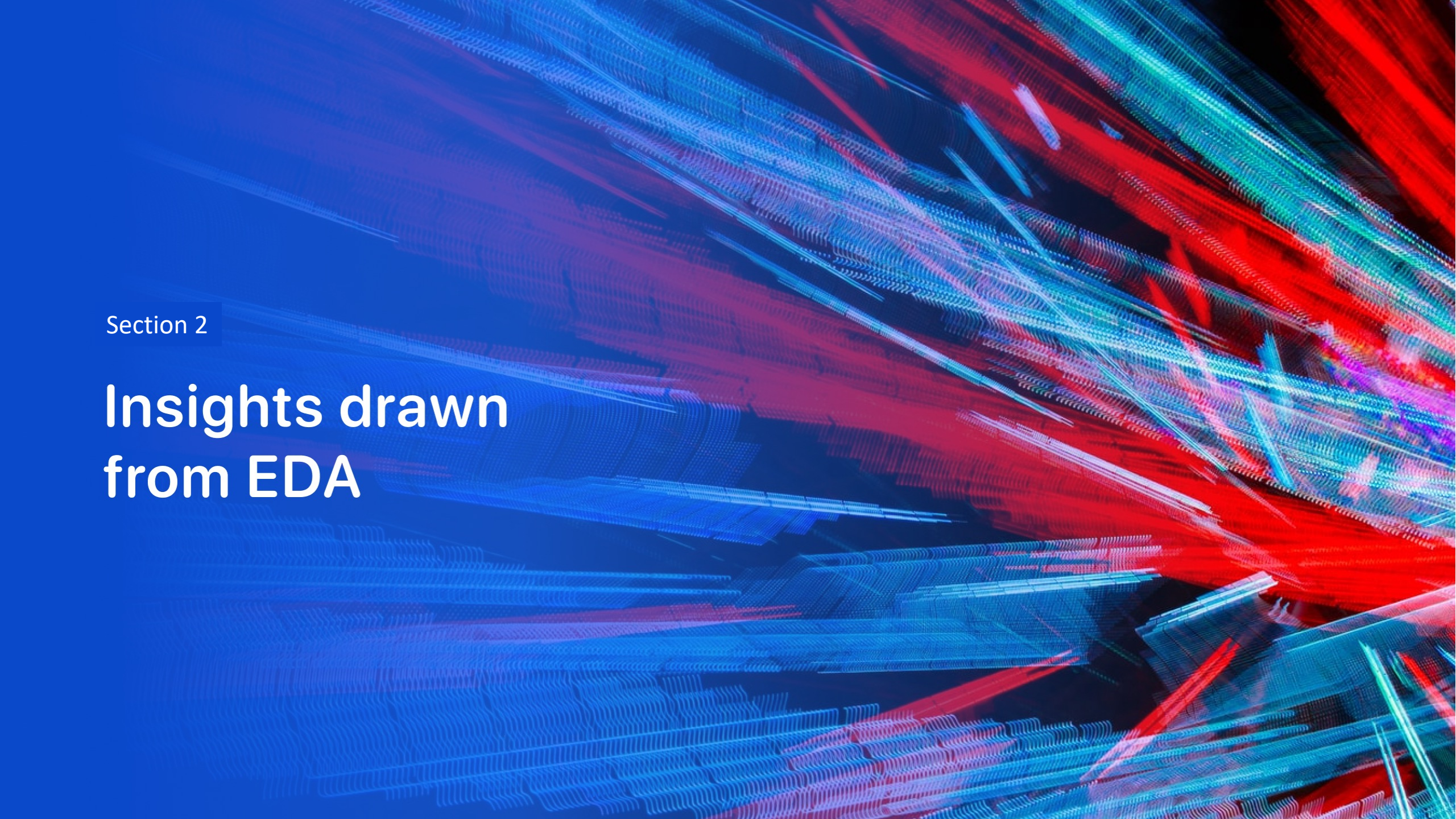
Predictive Analysis (Classification)



https://github.com/daVictorE/IBM_Data_Science/blob/main/10_SpaceX_Machine_Learning_Prediction.jupyterlite.ipynb

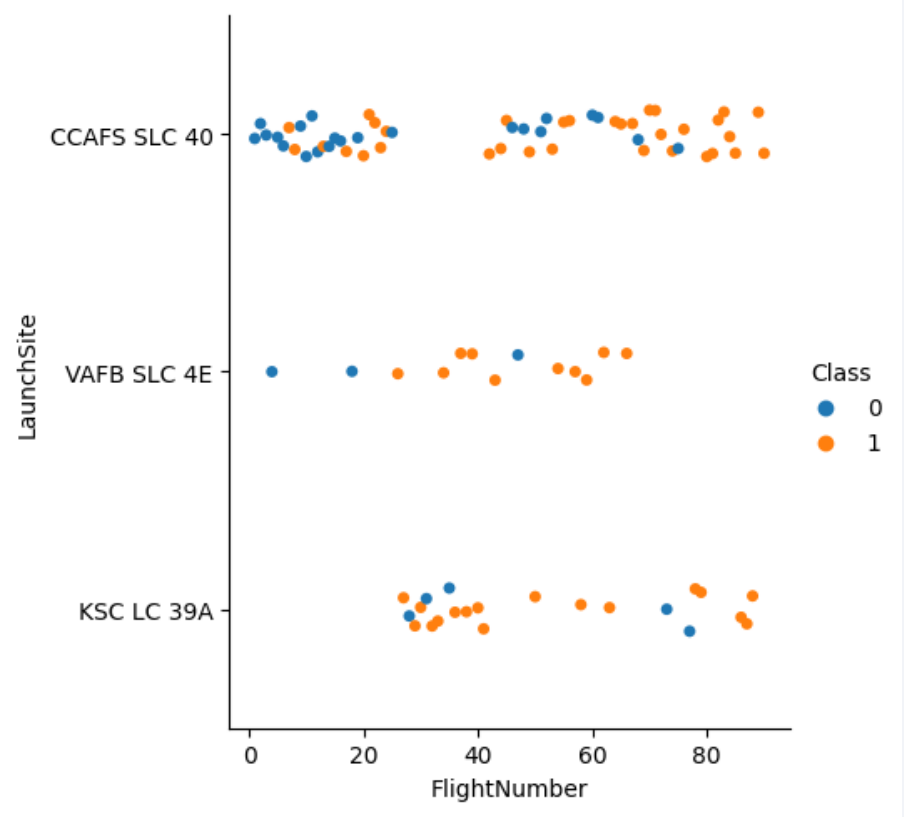
Results

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


Section 2
Insights drawn from EDA

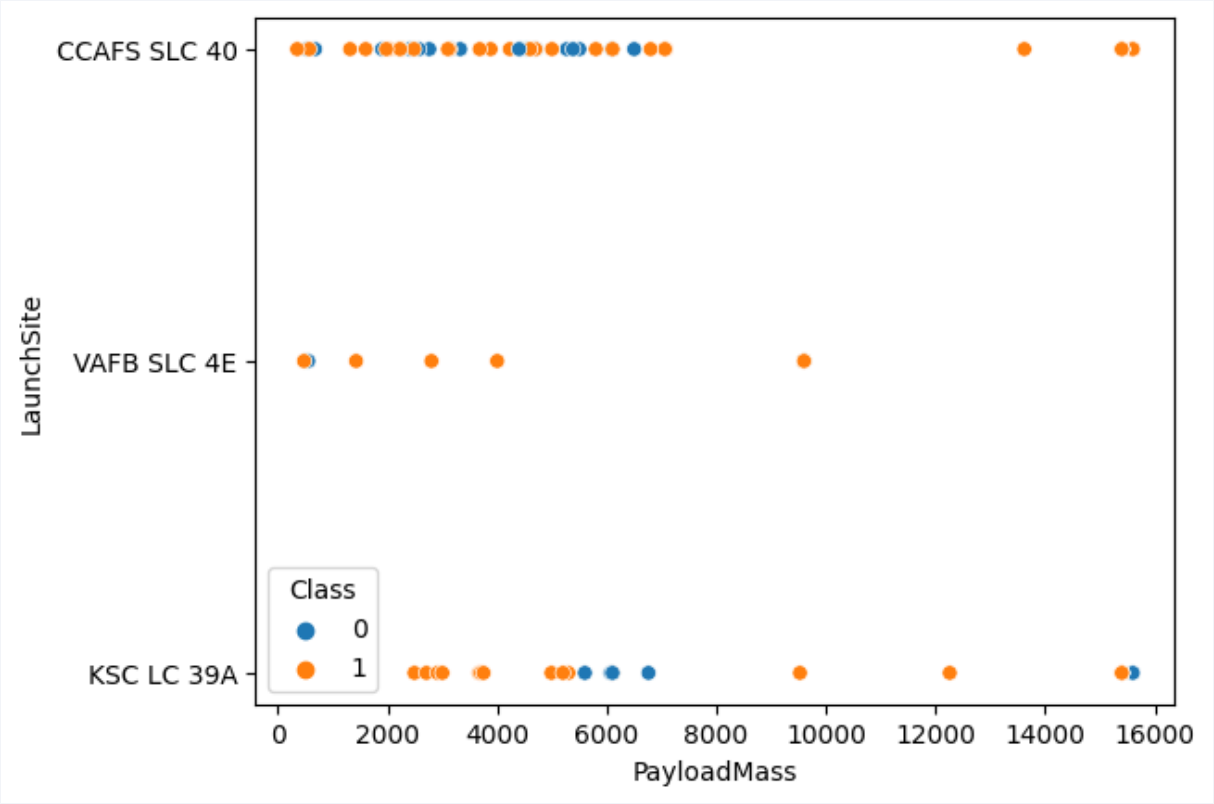
Flight Number vs. Launch Site

- Catplot (Flight Number vs Launch Site)
  - We see that different launch sites have different success rates. Success rates are represented by the class color and values



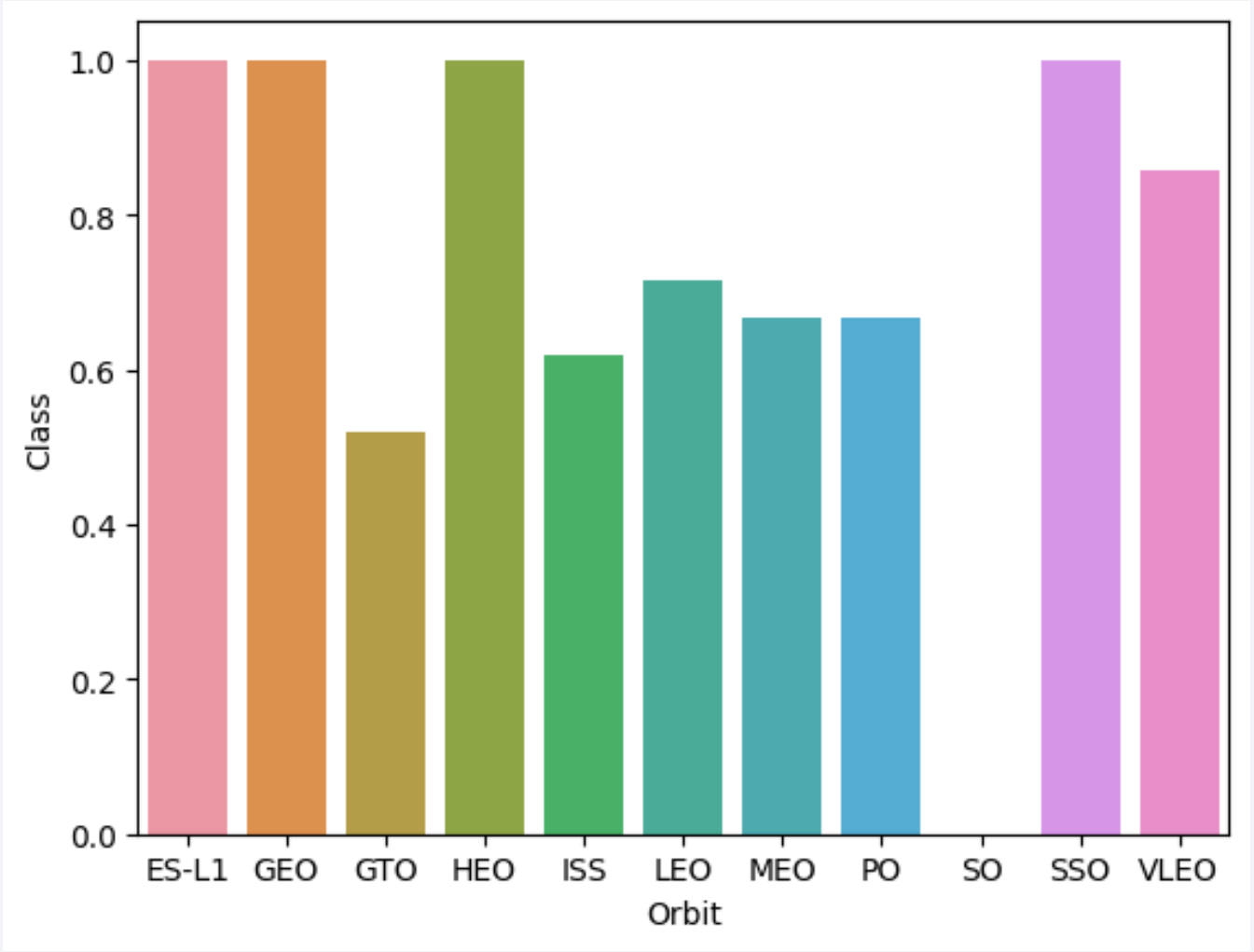
Payload vs. Launch Site

- Scatterplot (Payload vs Launch Site)
  - VAFB-SLC Launch Site has no rockets launched for heavy payload mass greater than 10,000
  - CCAFS SLC 40 has 3 successful launches for heavy load



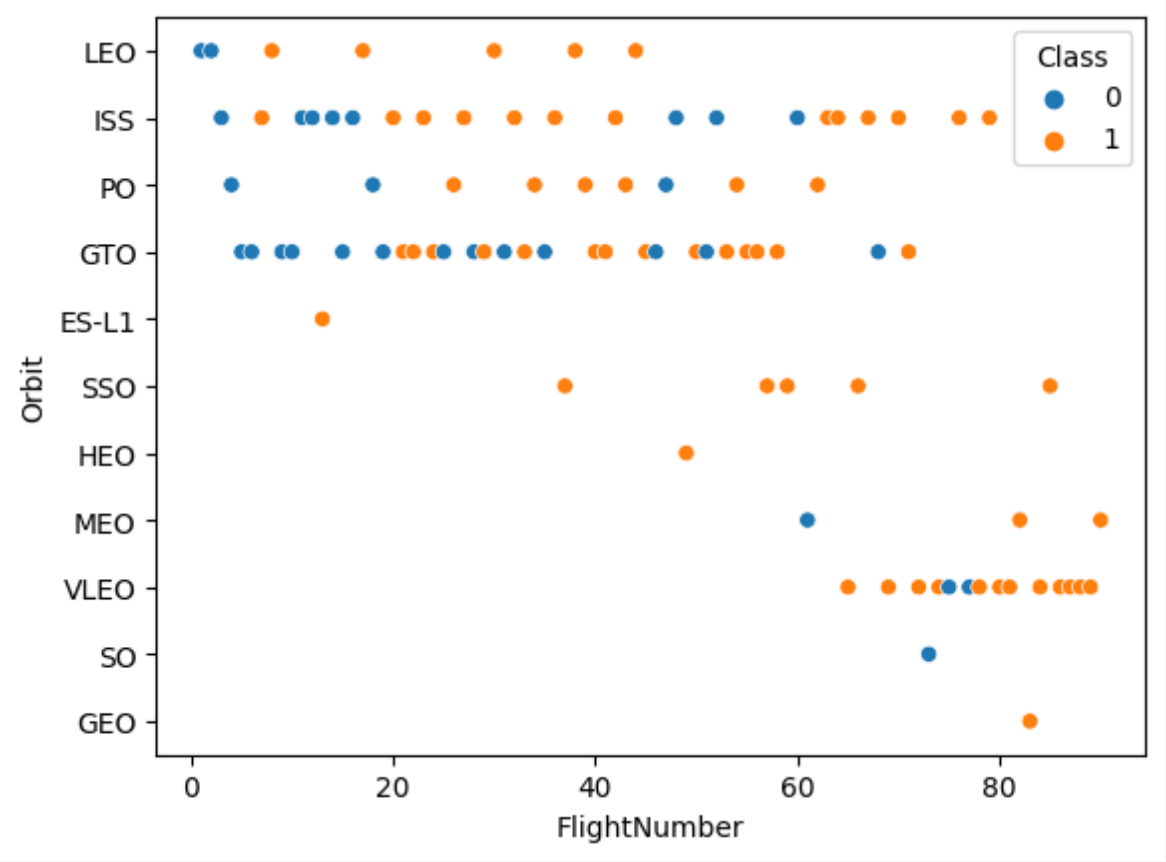
Success Rate vs. Orbit Type

- Barplot (Orbit and Class)
  - Orbits with highest success rate are ES-L1, GEO, HEO, and SSO



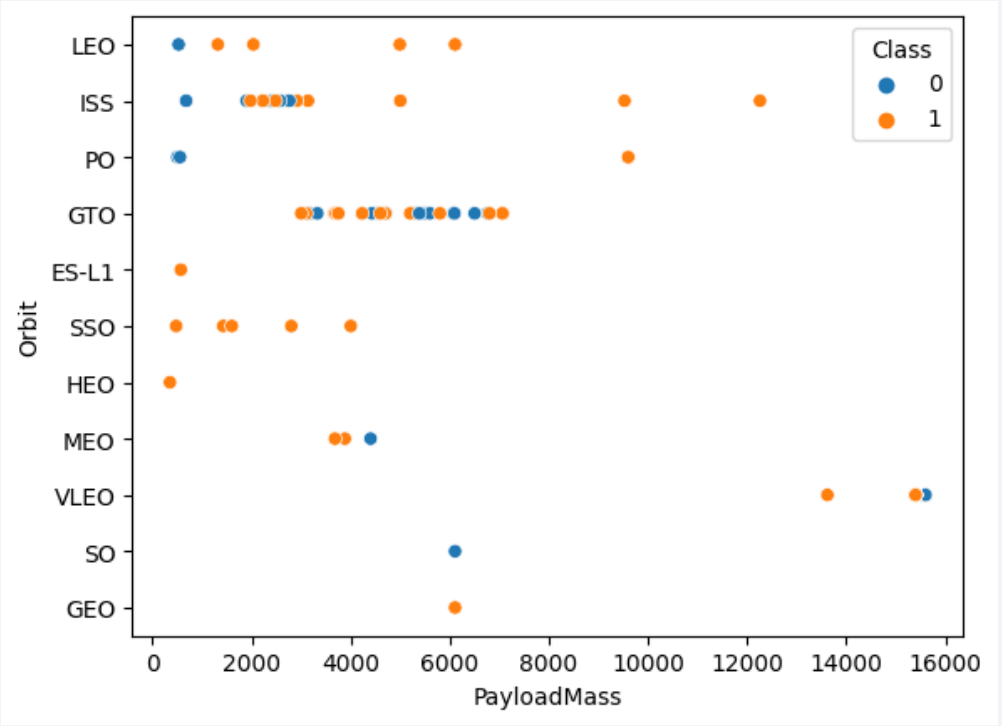
Flight Number vs. Orbit Type

- Scatterplot (Flight Number vs Orbit)
  - We see that in the LEO orbit the success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



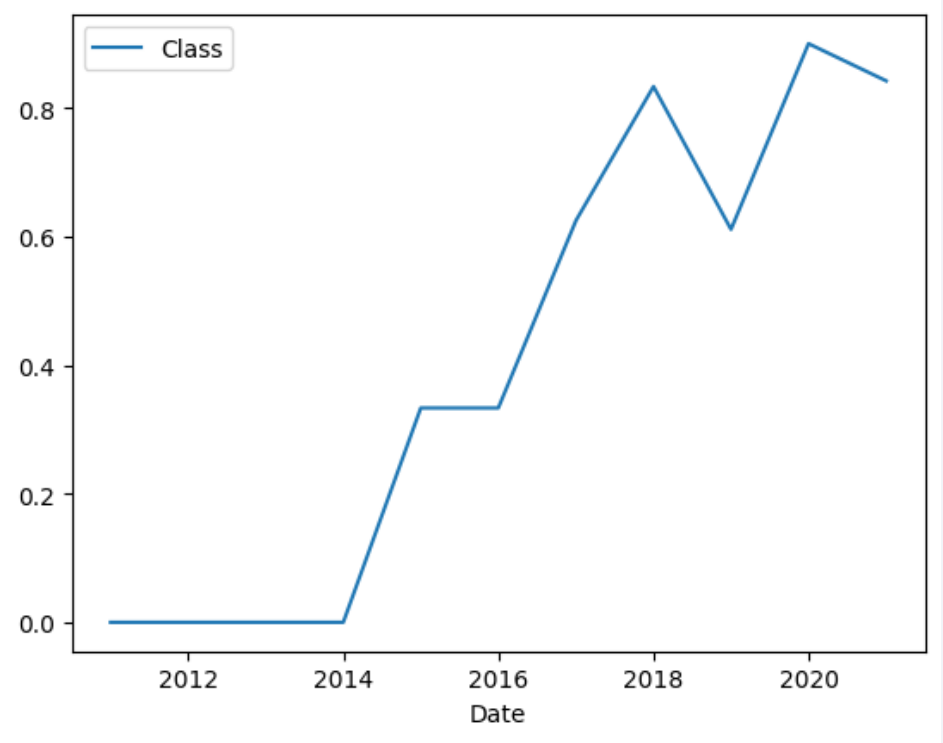
Payload vs. Orbit Type

- Scatterplot (Payload Mass vs Orbit)
  - We see that with heavy payloads PO and ISS have positive landing rate
  - GTO has a mix of successful and failure landing rate
  - ES-L1, SSO and HEO has not failure landing. Their payload is less than 6000 kg



Launch Success Yearly Trend

- Line (Data and Class)
  - We observe that the success rate since 2013 kept increasing till 2020



All Launch Site Names

- There are 4 Launch Site for SpaceX

```
1  cur.execute("select distinct Launch_Site from SPACEXTBL")
2  ls = cur.fetchall()
3  for row in ls:
4    print(row)
5
6  print(f'ls: {ls}')
```

```
('CCAFS LC-40',)
('VAFB SLC-4E',)
('KSC LC-39A',)
('CCAFS SLC-40',)
ls: [('CCAFS LC-40',), ('VAFB SLC-4E',), ('KSC LC-39A',), ('CCAFS SLC-40',)]
```

Launch Site Names Begin with 'CCA'

- Below is the list of 5 Launch Sites that begins with CCA. The first 2 have a landing outcome of “Failure (parachute)”, and the last three were “No attempt”

```
]  1  ls2 = cur.execute("select * from SPACEXTBL where Launch_Site like 'CCA%' limit 5")
   2  for row in ls2:
   3  |  print(row)
```

```
('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')
```

Total Payload Mass

- Total payload carried by NASA (CRS) was 45,596 Kg

```
plm = cur.execute("select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer = 'NASA (CRS)' ")
for row in plm:
  print(row)
```

```
(45596,)
```

Average Payload Mass by F9 v1.1

- Average payload carried by Booster_Version F9 v1.1 was 2,928 Kg

```
plm2 = cur.execute("select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version = 'F9 v1.1' ")
for row in plm2:
  print(row)
```

```
(2928.4,)
```

First Successful Ground Landing Date

- The first successful landing outcome in ground pad was 2015-12-22

```
lst1 = []
lo = cur.execute("""select Date from SPACEXTBL where "Landing _Outcome"="Success (ground pad)" """)
for row in lo:
  lst1.append(row[0])


los = pd.Series(lst1)
dt = pd.to_datetime(los, format='%d-%m-%Y')
dt.min()
```

```
Timestamp('2015-12-22 00:00:00')
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- Names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
1  bv = cur.execute("""select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_>4000 and PAYLOAD_MASS__KG_<6000 and "Landing _Outcome"="Success (drone ship)"  """)
2  for row in bv:
3    print(row)
```

```
('F9 FT B1022',)
('F9 FT B1026',)
('F9 FT  B1021.2',)
('F9 FT  B1031.2',)
```

Total Number of Successful and Failure Mission Outcomes

- There are 100 Successful Mission Outcome and 1 Failure

```
tn = cur.execute("""select "Mission_Outcome", count(*) from SPACEXTBL group by "Mission_Outcome"  """)
for row in tn:
  print(row)
```

```
('Failure (in flight)', 1)
('Success', 98)
('Success ', 1)
('Success (payload status unclear)', 1)
```

Boosters Carried Maximum Payload

- Names of the booster_versions which have carried the maximum payload mass

```
mpyl = cur.execute(" select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)")
for row in mpyl:
  print(row)
```

```
('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

- List of records which display the
  - month names for the months in year 2015
  - failure landing_outcomes in drone ship
  - booster versions
  - launch_site

```
mn = cur.execute("""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' """)
for row in mn:
  print(row)
```

```
('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

- Rank of the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

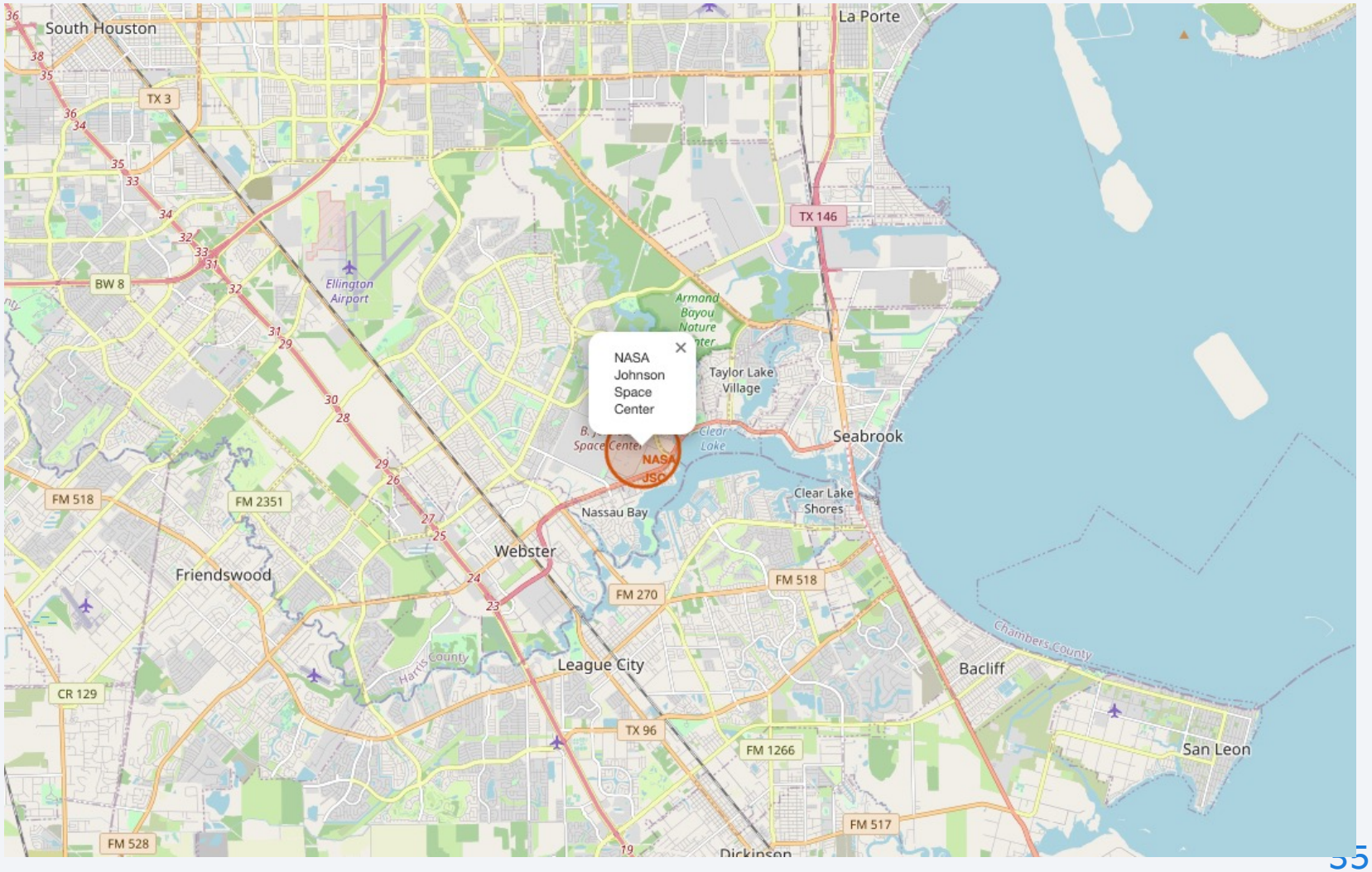
```
cur.execute("""select count("Landing _Outcome") as rank, "Landing _Outcome" from SPACEXTBL
where "Landing _Outcome" like "Success%"
Group by "Landing _Outcome" Order by rank desc  """)

rk = cur.fetchall()
for row in rk:
  print(row)

#print(rk)
```

```
(38, 'Success')
(14, 'Success (drone ship)')
(9, 'Success (ground pad)')
```


Section 3
Launch Sites
Proximities Analysis

Folium Map: NASA Johnson Space Center

Location of NASA JSC on the map



Folium Map: Launch Sites

Location of Launch Sites

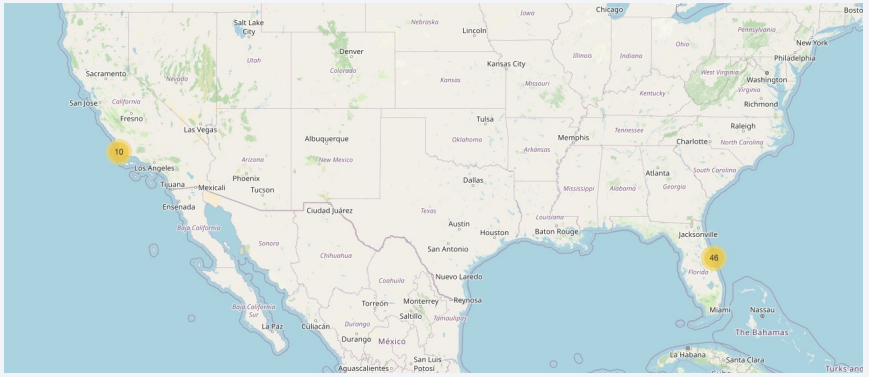
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Folium Map: Cluster

56 launches

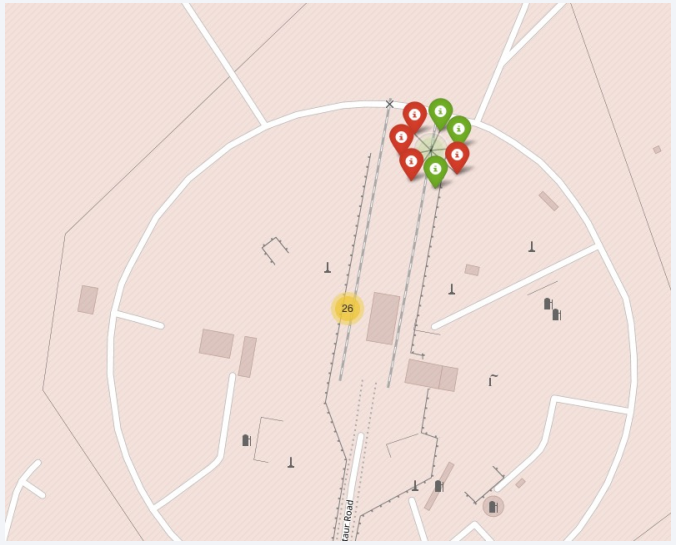
10 West Coast launches

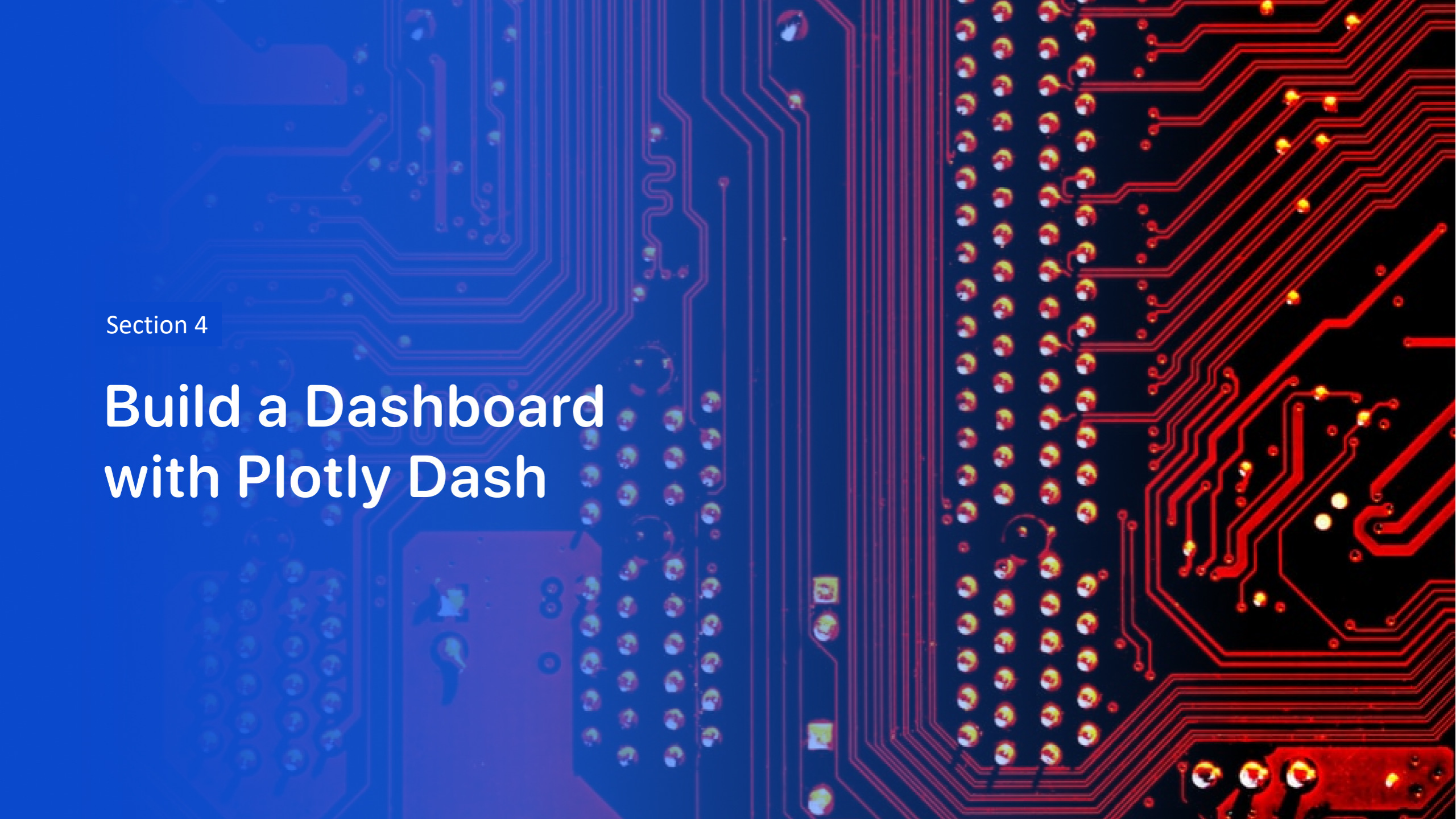
46 East Coast launches



7 launches at CCAFS SLC-40

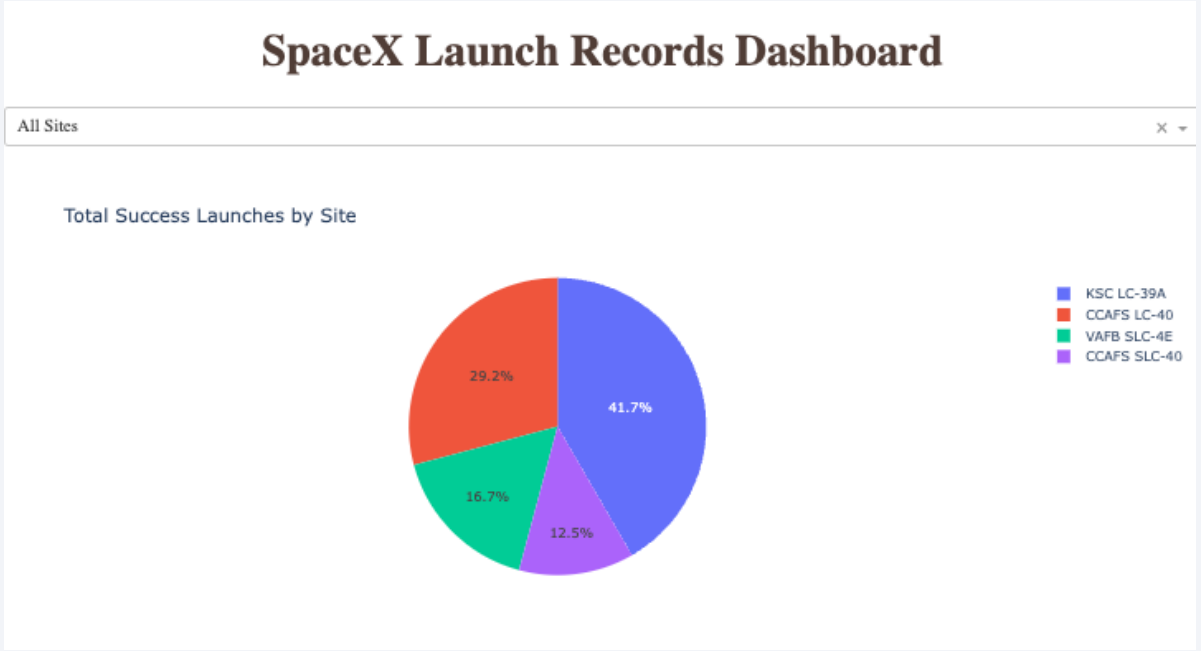
- 3 Success
- 4 Failure




Section 4
Build a Dashboard with Plotly Dash

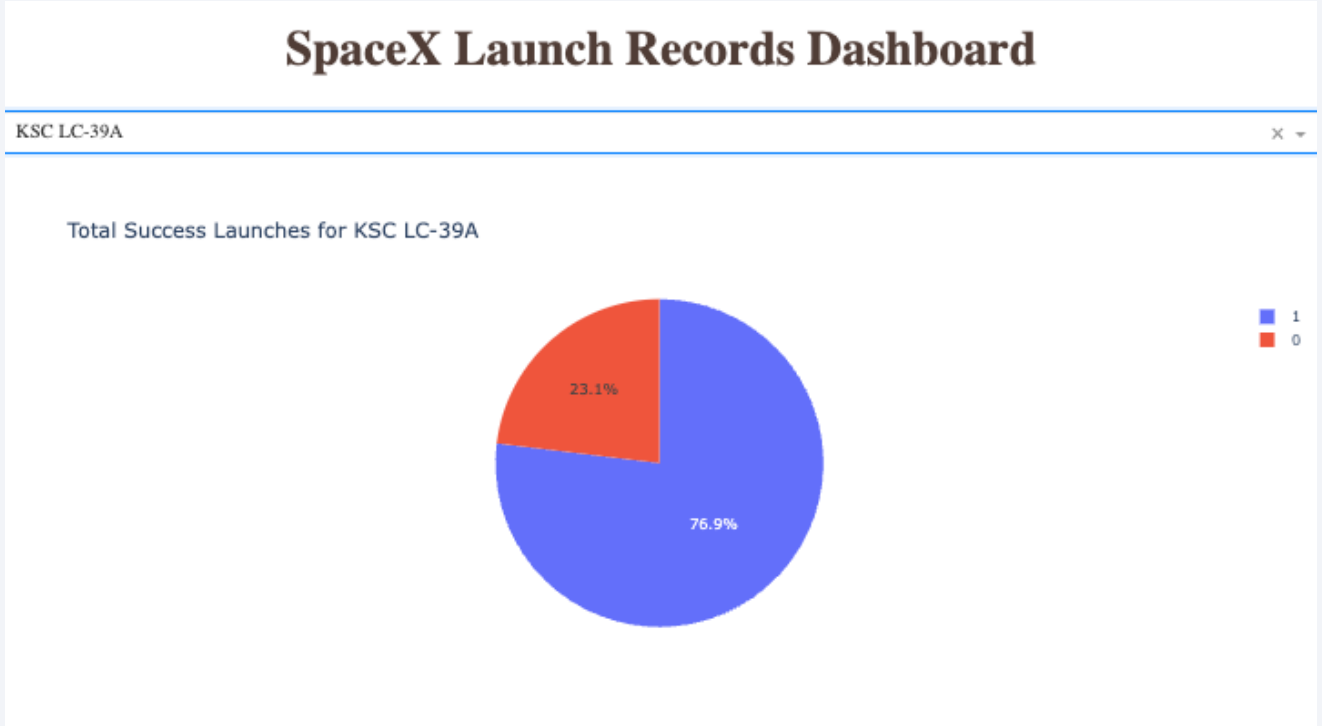
Total Success Launched by Site

Pie Chart of all launch sites showing the percentage of successful launches by site



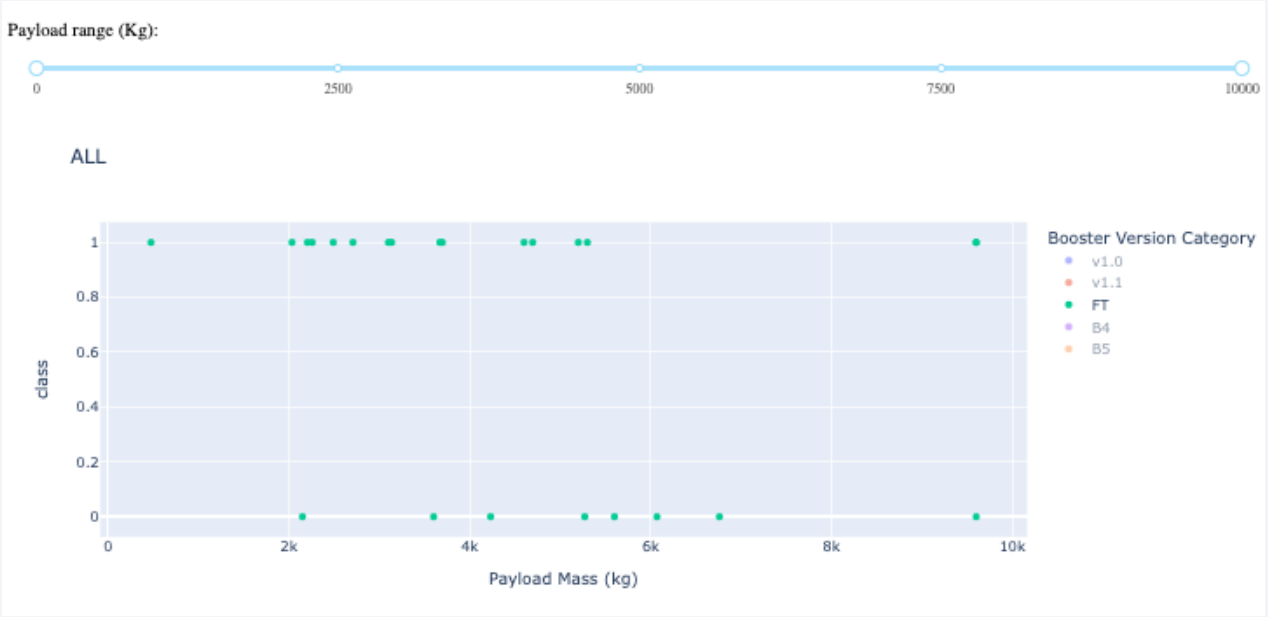
Highest Successful Launches

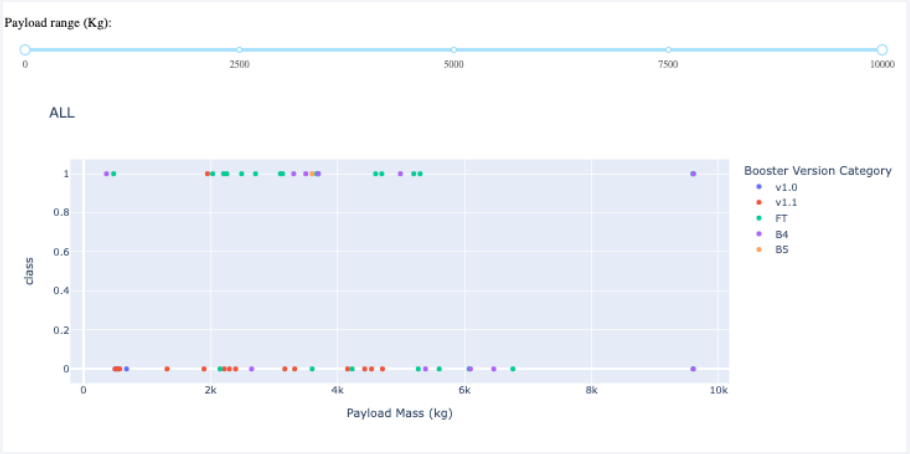
- KSC LC-39A has the highest successful launches of 76.9%

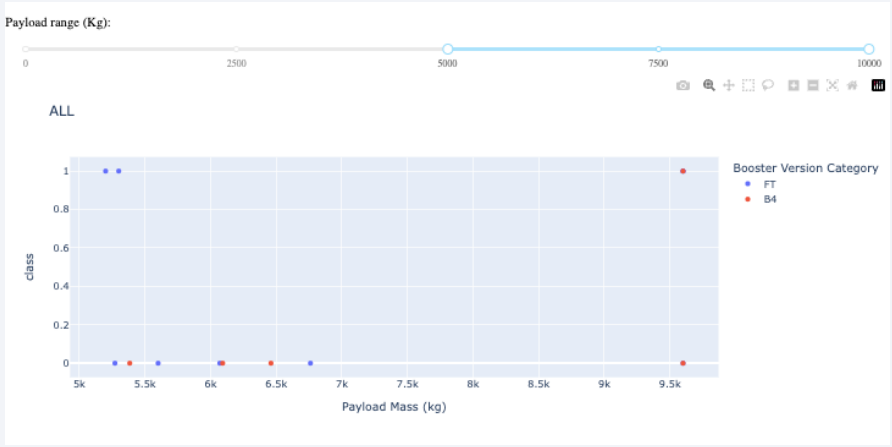


Scatterplot of Payload vs Class

- Screen shots of Payload vs Launch Outcome with different payload range
- Booster FT has more success than failure under 6000 Kg





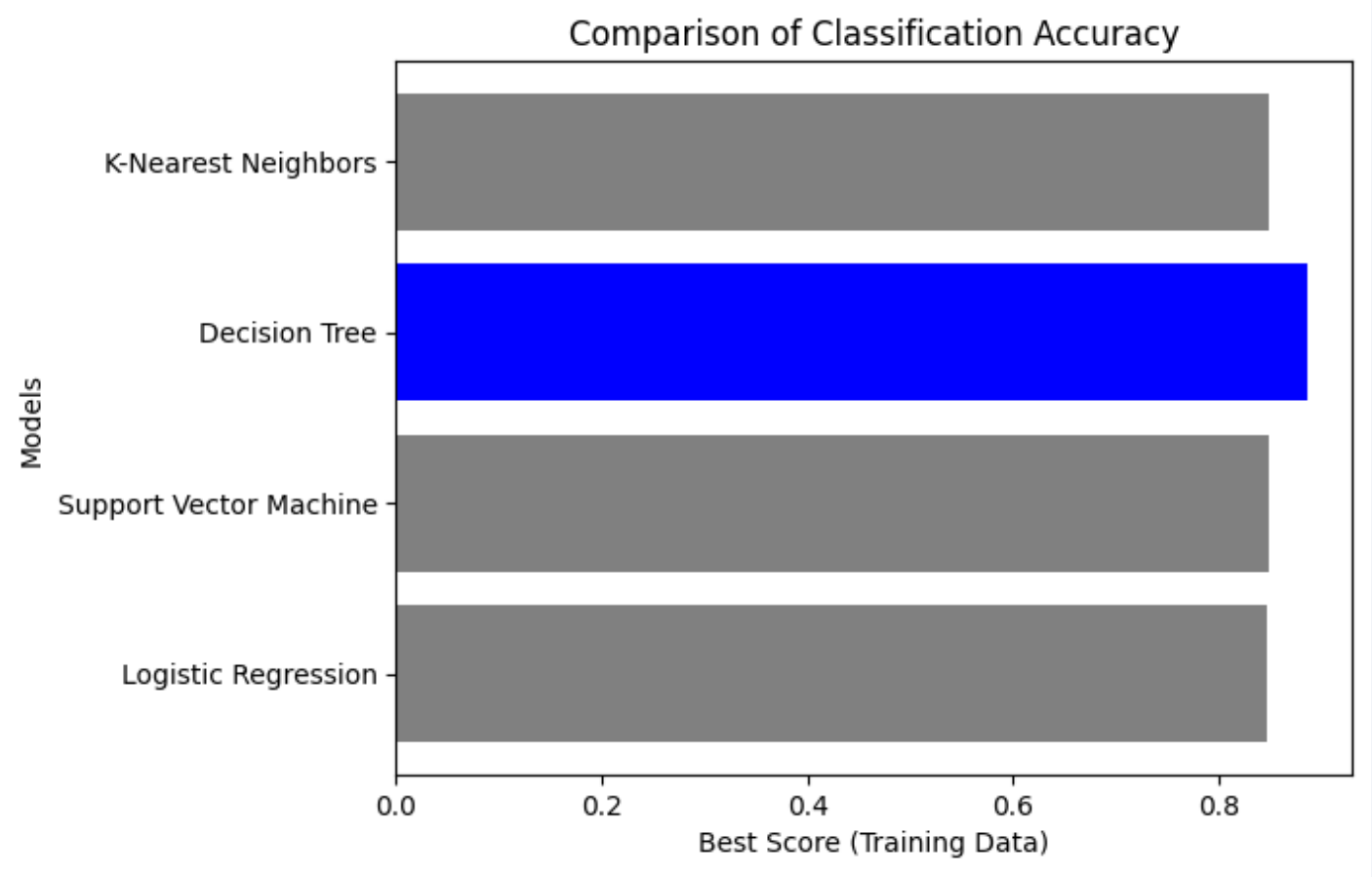


Section 5

Predictive Analysis (Classification)

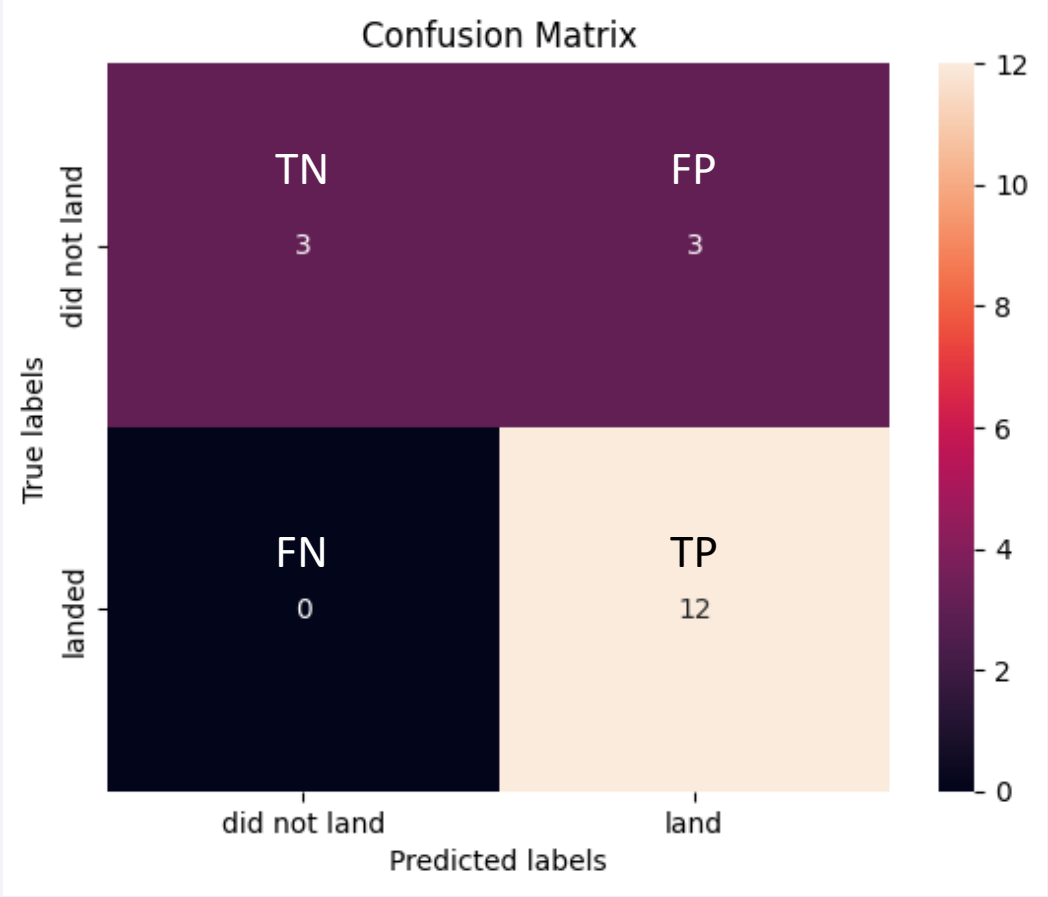
Classification Accuracy

- Decision Tree has the best classification accuracy



Confusion Matrix

- The model was able to predict all 3 landing that did not land. On the other hand, it failed to predict 3 successful landing as did not land



Conclusions

- Decision Tree if the best model in the group for predicting the landing
- All model performed well as showed in the Confusion Matrix
- Based on the test data, all models have the same Score

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

	Model	Best Score(Taining Data)	Score(Test Data)
0	Logistic Regression	0.846429	0.833333
1	Support Vector Machine	0.848214	0.833333
2	Decision Tree	0.885714	0.833333
3	K-Nearest Neighbors	0.848214	0.833333

Thank you!