Winning Space Race with Data Science

Ramiro Asturias

March 21st 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Collected and cleaned data from SpaceX API and HTML Table from Wikipedia
  - Exploratory Data Analysis (EDA) to find patterns in the data and determine what would be the label for training supervised models' outcomes of the landings that our model will be predicting
  - EDA with visualizations and Feature Engineering
  - Built interactive Maps to visualize landings and interactive Dashboards to validate correlations
  - Perform exploratory Data Analysis and determine Training Labels
    - Create a column for the class
    - Standardize the data
    - Split into training data and test data
    - Test best model for prediction with best parameters based on training data
- Summary of all results
  - Interactive and predictive analysis
  - Based on models' performance Decision Tree and KNN models with 88% accuracy are the best

Introduction

SpaceX, a company that is thriving by making space travel affordable for everyone because their rocket launches are relatively inexpensive.
SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars;
other providers cost upwards of 165 million dollars each, much of the savings is because
SpaceX can reuse the first stage.

Space Y that would like to compete with SpaceX founded by Billionaire industrialist Allon
Musk.

We want determine the price of each launch by gathering information about Space X and creating dashboards for Space Y.

We will also determine if SpaceX will reuse the first stage.
Instead of using rocket science to determine if the first stage will land successfully, we will train a machine learning model and use public information to predict if SpaceX will reuse the first stage.

Section 1

Methodology

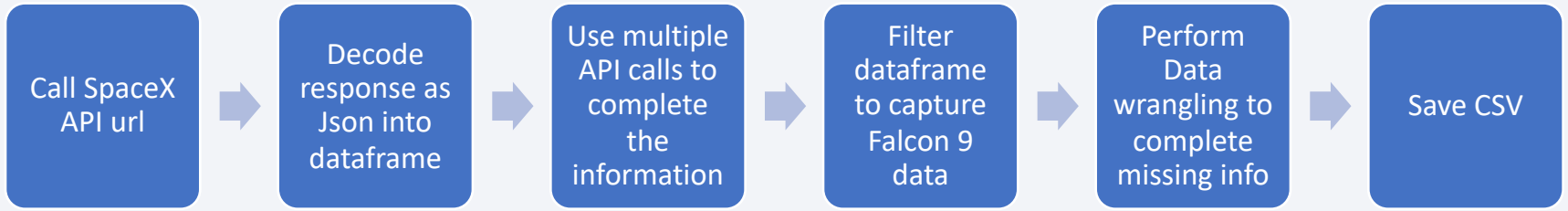
Methodology

Executive Summary

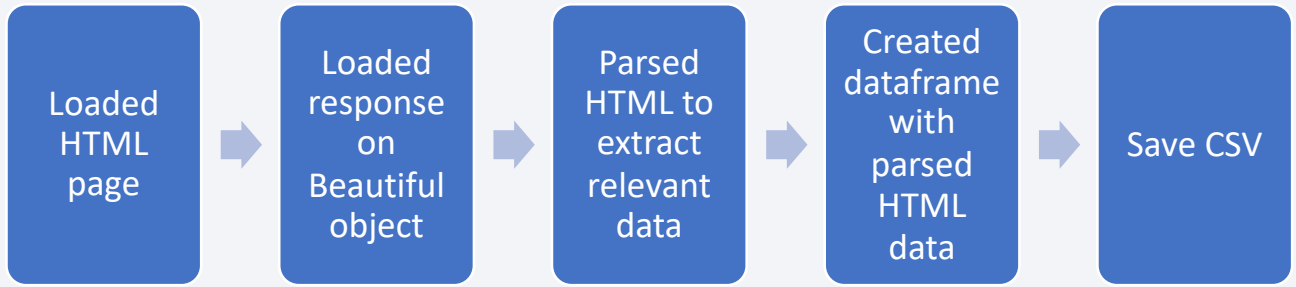
- Data collection methodology:
  - SpaceX Rest API for rocket, launch platform, payloads and core information
  - Web Scrapping from Wikipedia for launch historical records
- Perform data wrangling
  - Created an outcome label “class” to determine the result of the launch (0 fail, 1 success) based on the landing outcome calculated column
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Created column for class, standardize data, split into training and test data. Found the best Hyperparameter for SVM, CT, LR and then compare their scores.

Data Collection

- Describe how data sets were collected.
  - SpaceX API was collected using requests library and helper functions to extract information from the responses
  - HTML Wikipedia page data was collected with BeautifulSoup library
- SpaceX API



- Wikipedia Page



Data Collection – SpaceX API

Call SpaceX API url → Decode response as Json into dataframe → Use multiple API calls to complete the information → Filter dataframe to capture Falcon 9 data → Perform Data wrangling to complete missing info → Save CSV

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

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

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

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

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

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

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

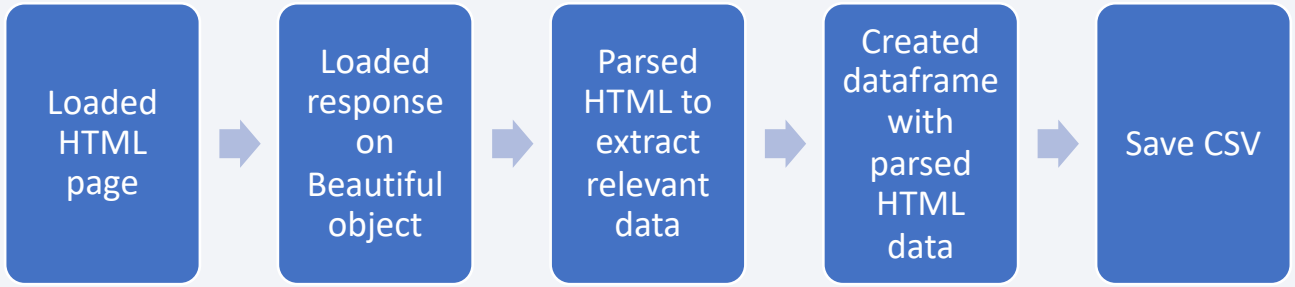
```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = data_falcon[data_falcon['BoosterVersion']=='Falcon 9']
```

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

data_falcon9.isnull().sum()

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

Notebook URL

Result Data URL

Data Collection - Scraping



First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
# use requests.get() method with the provided static_url
# assign the response to a object

response = requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
bs = BeautifulSoup(response.content,'html5lib')
```

```
# Assign the result to a list called `html_tables`

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

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

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(bs.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
```

```
df=pd.DataFrame(launch_dict)
df.to_csv('space_web_scraped.csv', index=False)
```

[Notebook URL]

[Result Data URL]

Data Wrangling

Identify missing values → Identify categorical Vs numerical columns → Calculate the number of launches for each site → calculate the occurrence of each orbit → Determine the number and occurrence of mission outcomes → Create landing label from outcome column → Create class column representing outcome of landing → Save CSV

Identify and calculate the percentage of the missing values in each attribute

```
df.isnull().sum()/df.count()*100
```

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

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

```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```

```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
bad_outcomes
```

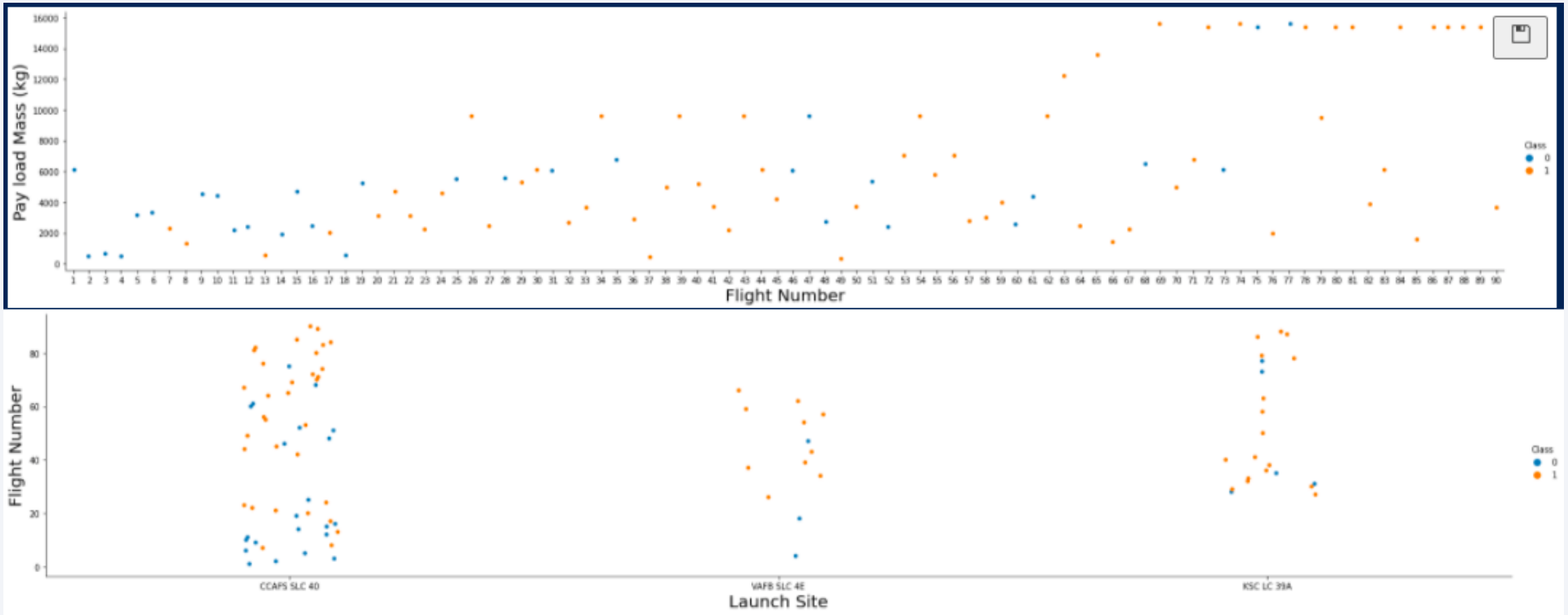
```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise

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

```
df['Class']=landing_class
df[['Class']].head(8)
```

Notebook URL

Result Data URL

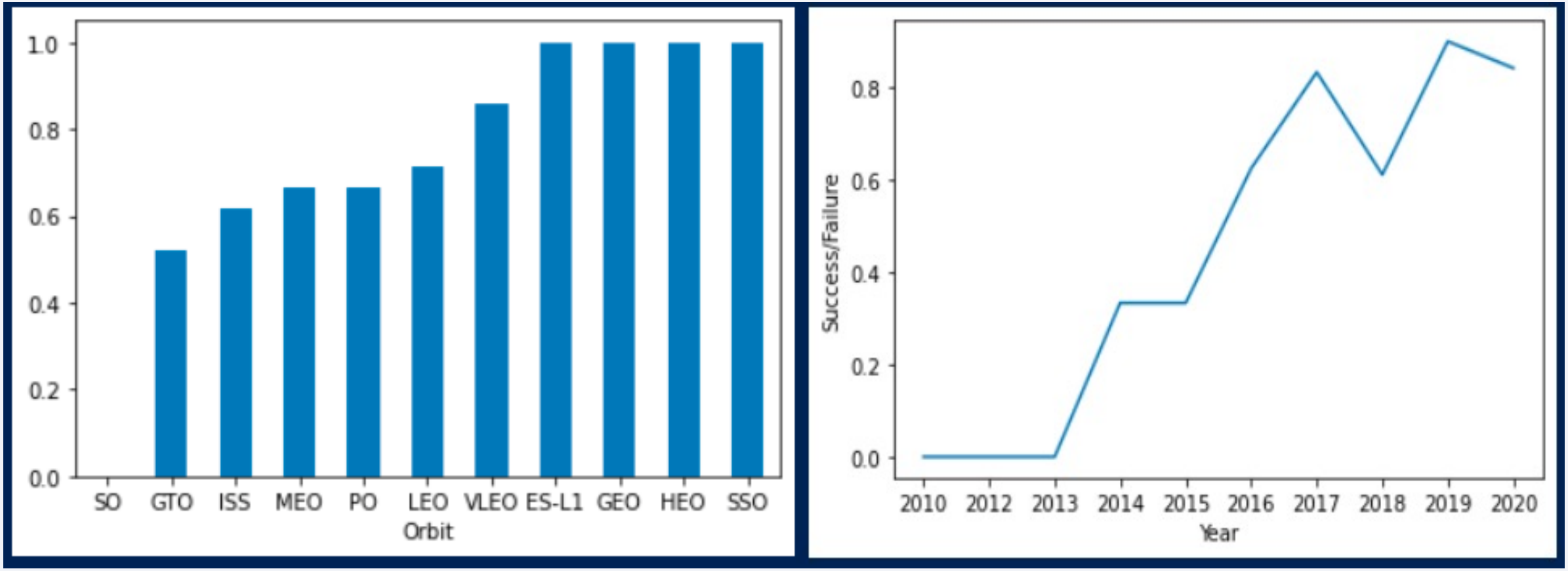
EDA with Data Visualization



- 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
- Relationship between Flight Number and Launch Site

Notebook URL11

EDA with Data Visualization



- Relationship Success Rate based on Orbit
- The average of success increases as the years go by

Notebook URL12

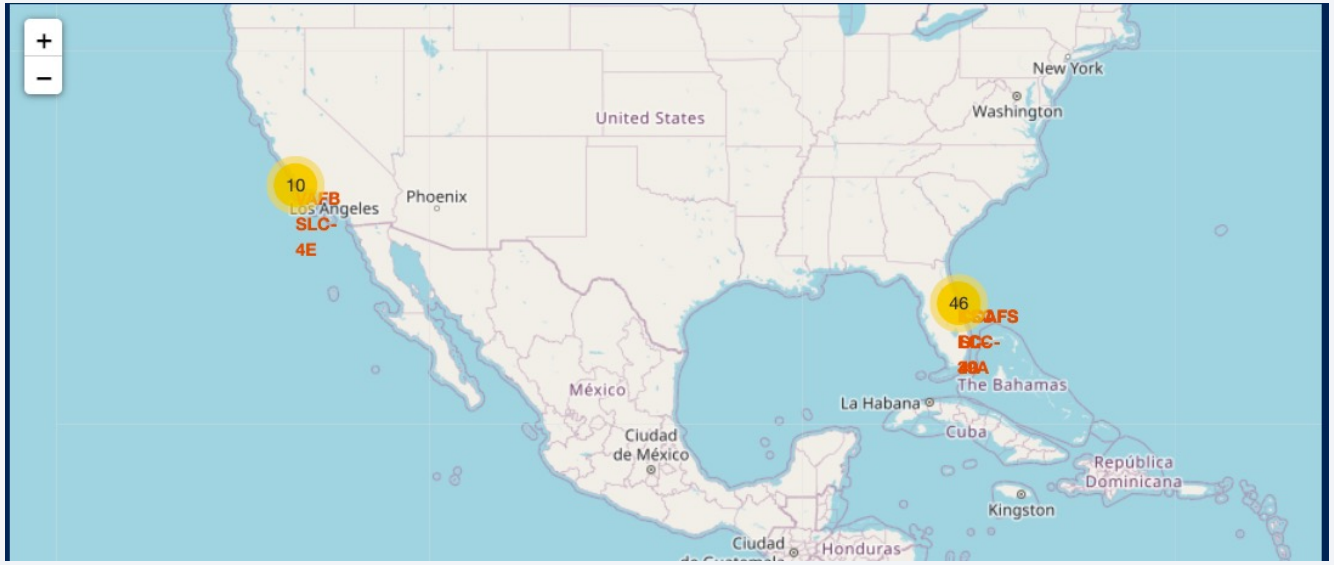
EDA with SQL

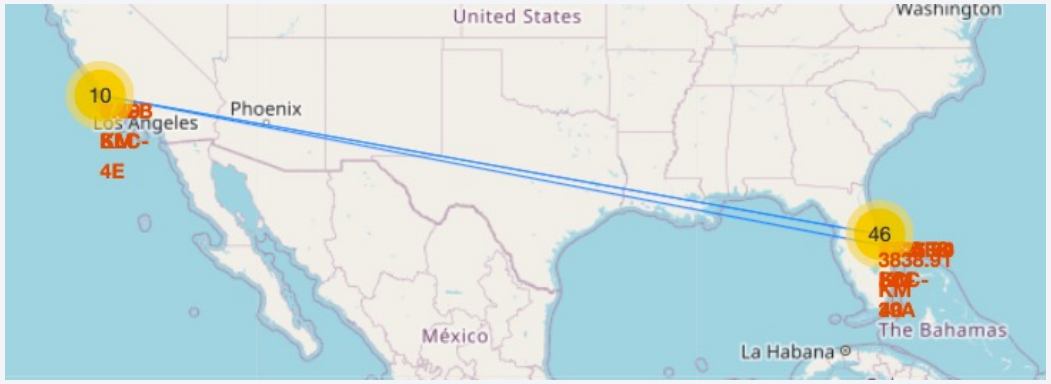
- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved.
- 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

Notebook URL

Build an Interactive Map with Folium

- Created circle markers for launch site on the map
- Marked the success/failed launches for each site on the map grouped or in clusters
- Used Lines to calculate distance between launch sites

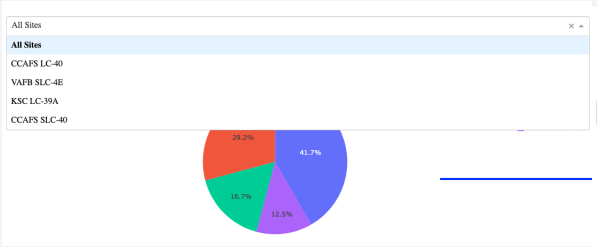


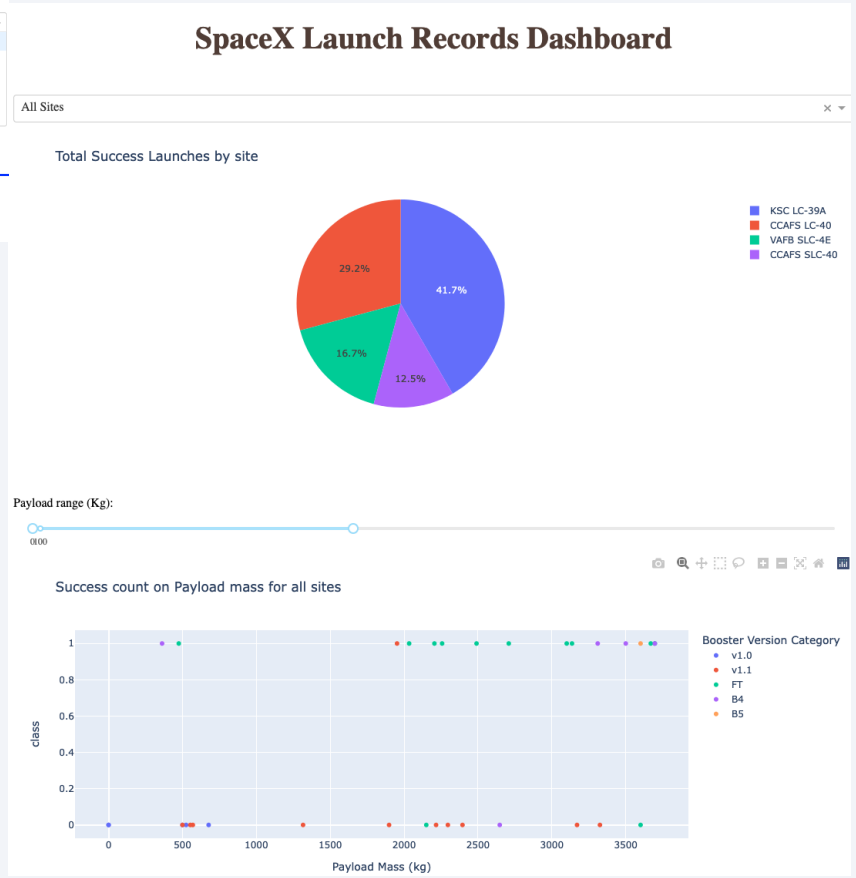


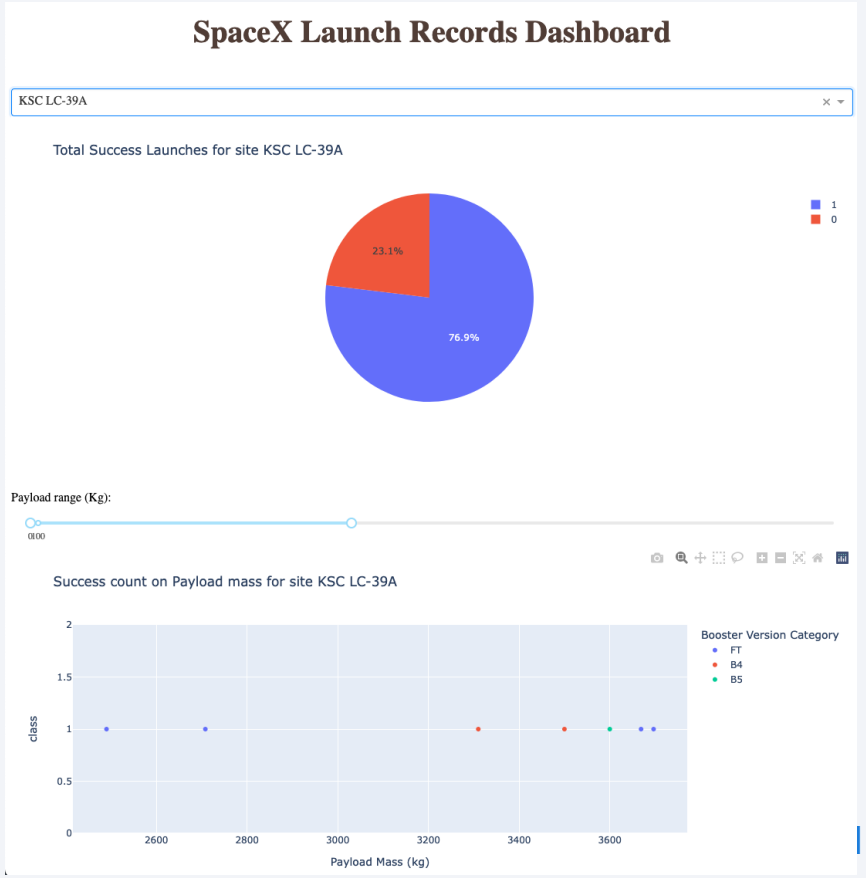
Notebook URL

Build a Dashboard with Plotly Dash

- Filter for all launching sites to see outcomes





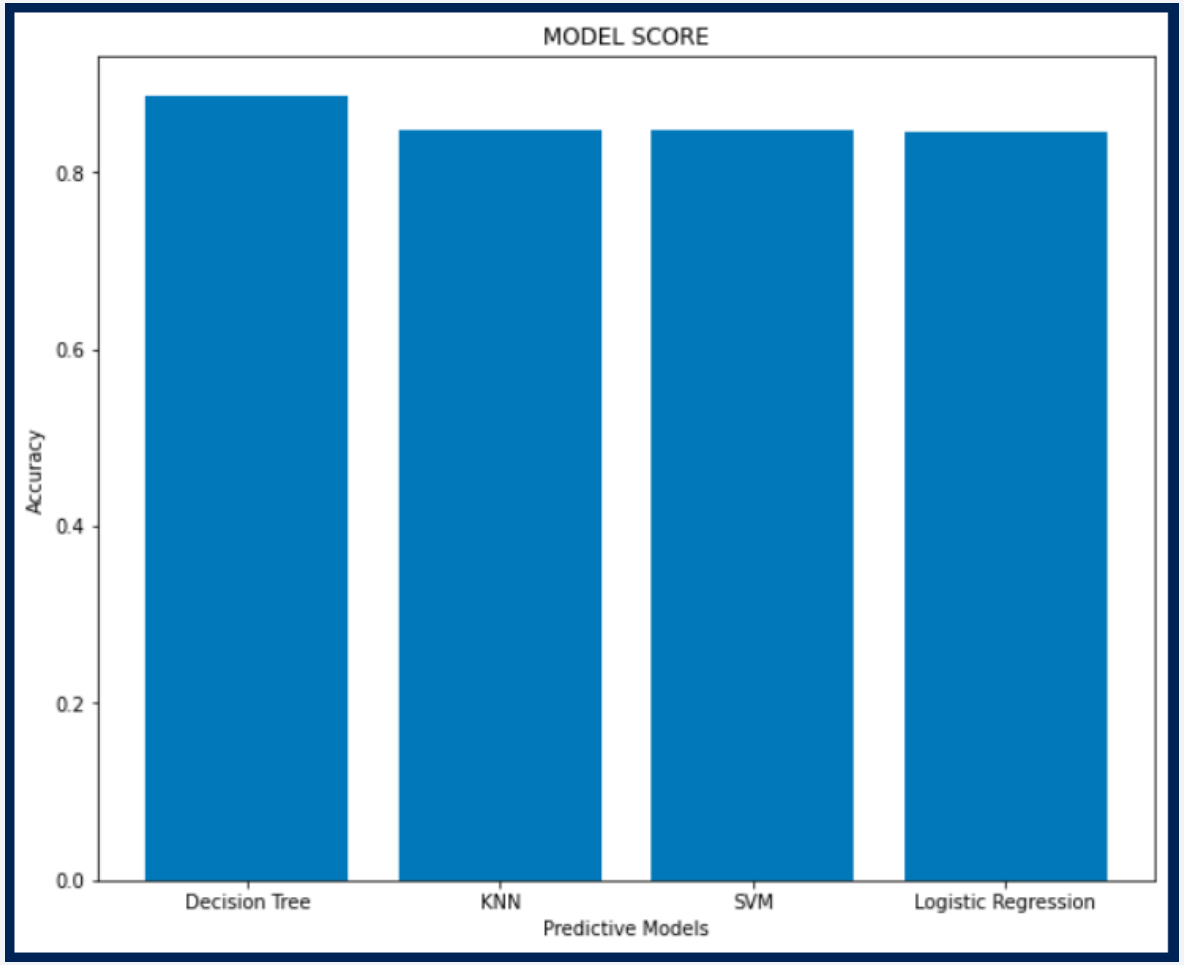


- Success count based on Payload mass

Notebook URL

Predictive Analysis (Classification)

- With the dataframe containing all data created a predicted variable Y with the **class variable**
- Standardized the independent variables X
- Separated the train dataset and the test with train_test_split function
- Used **GridSearchCV** with the configuration parameters for each of the models to find the best independent variables configurations
- Created a confusion matrix for each method to tabulate their performance
- The best model with the Hyperparameters is GT with 90% accuracy

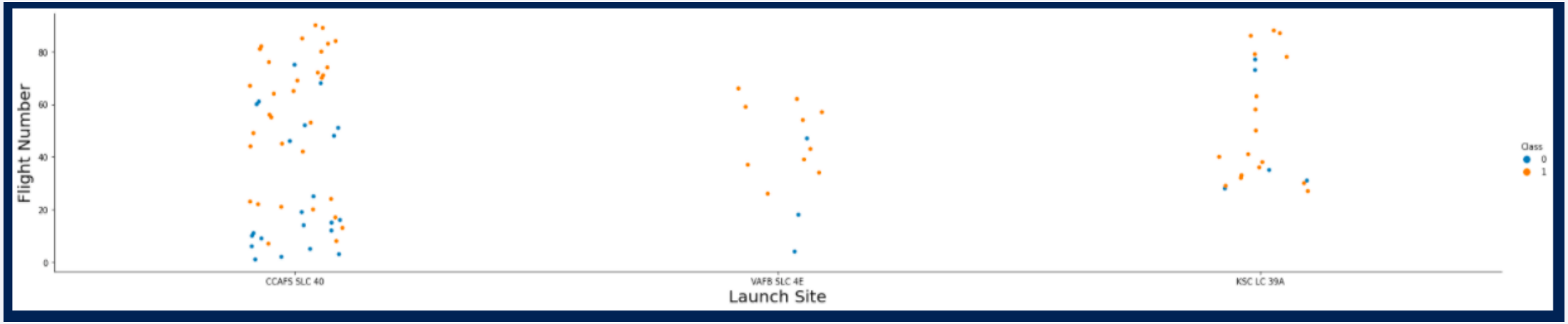


Perform EDA to determine Training Labels → Find best Hyperparameter for SVM, Classification Trees and Logistic Regression → Find the method performs best using test data

Notebook URL

Results
Exploratory data analysis results

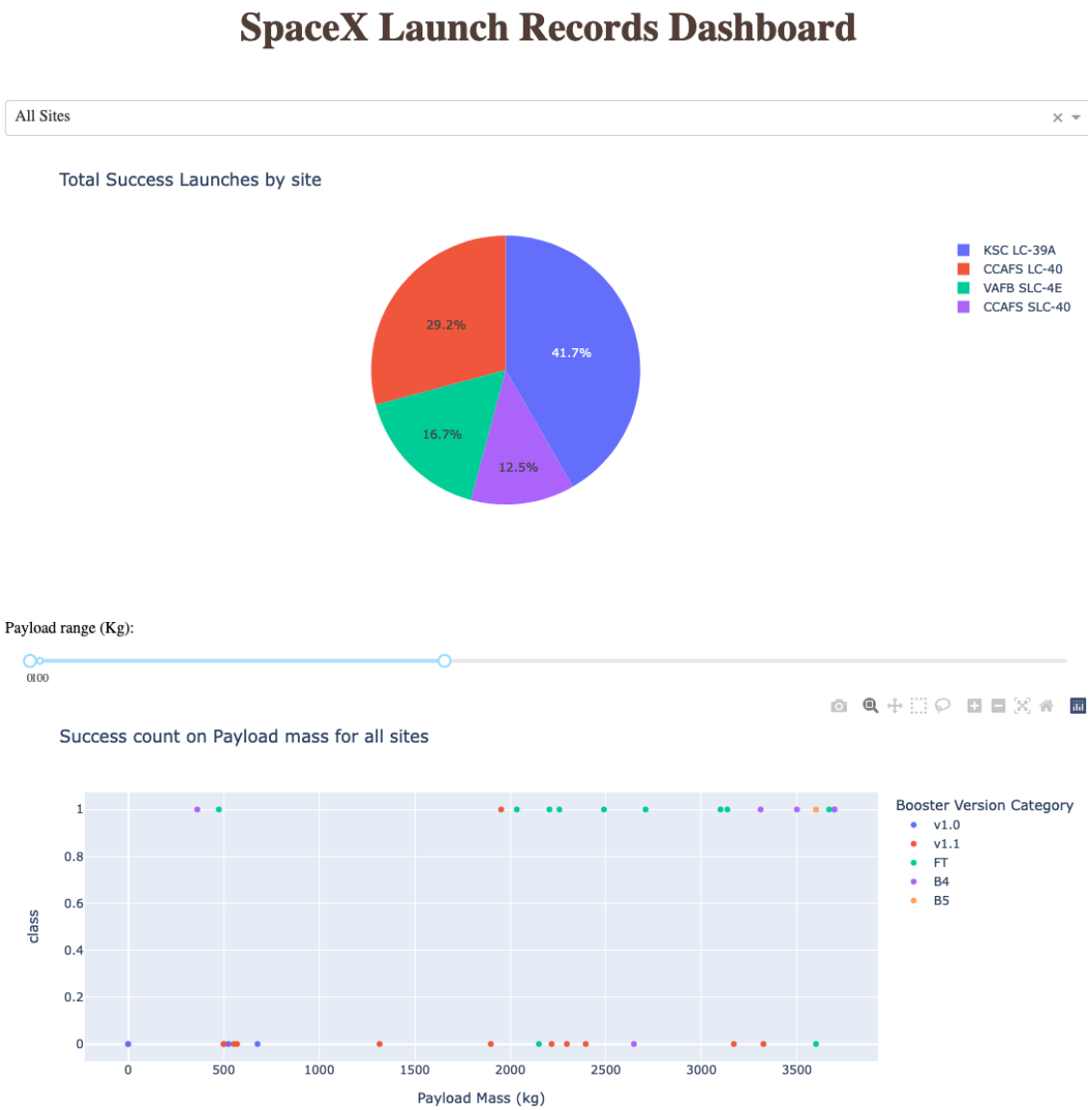
Relationship between Flight Number and Launch Site



We see that different launch sites have different success rates. **CCAFS LC-40,** has a success rate of 60% which seems lower compared to the other sites. But has almost 2 times more launches in comparation

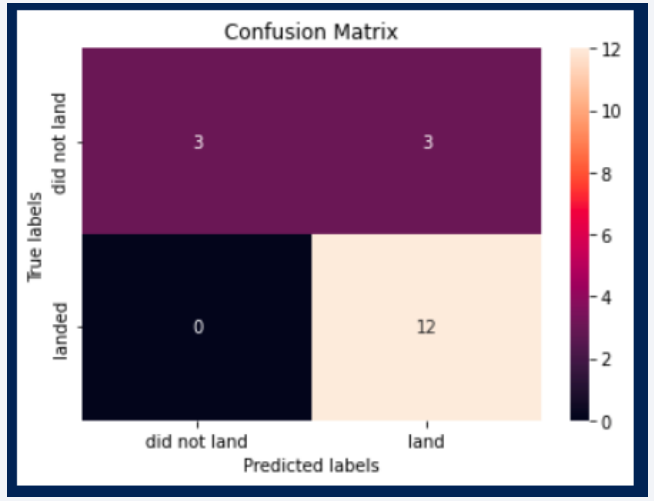
Results
Interactive analytics

- We can observe that as the payload increases the success on every site improves as well Specially for booster FT
- Booster V1.1 is a good indicator for all spaces that regardless the payload mass it will not return

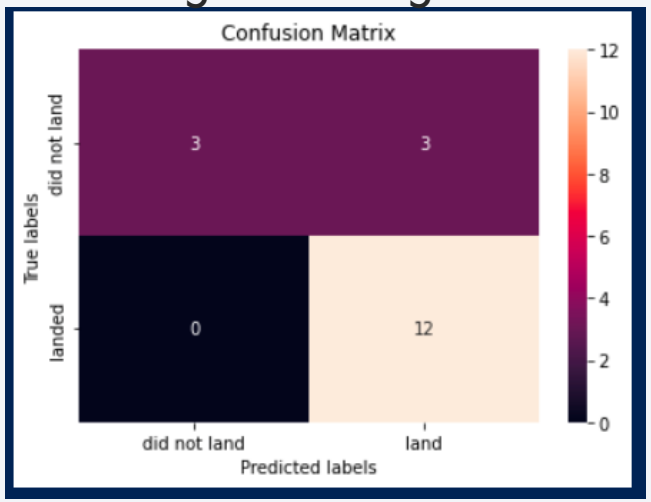


Results
Predictive analysis results

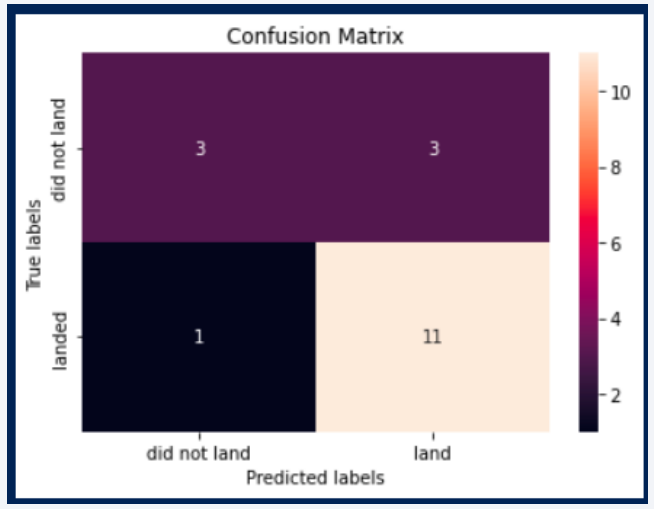
SVM



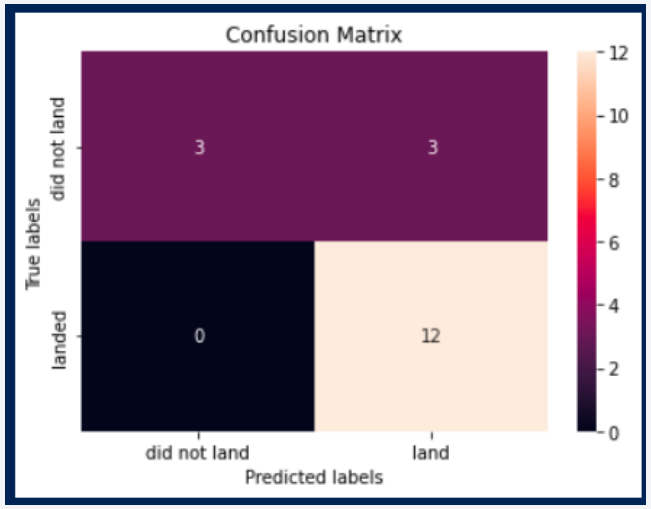
Logarithm Regression



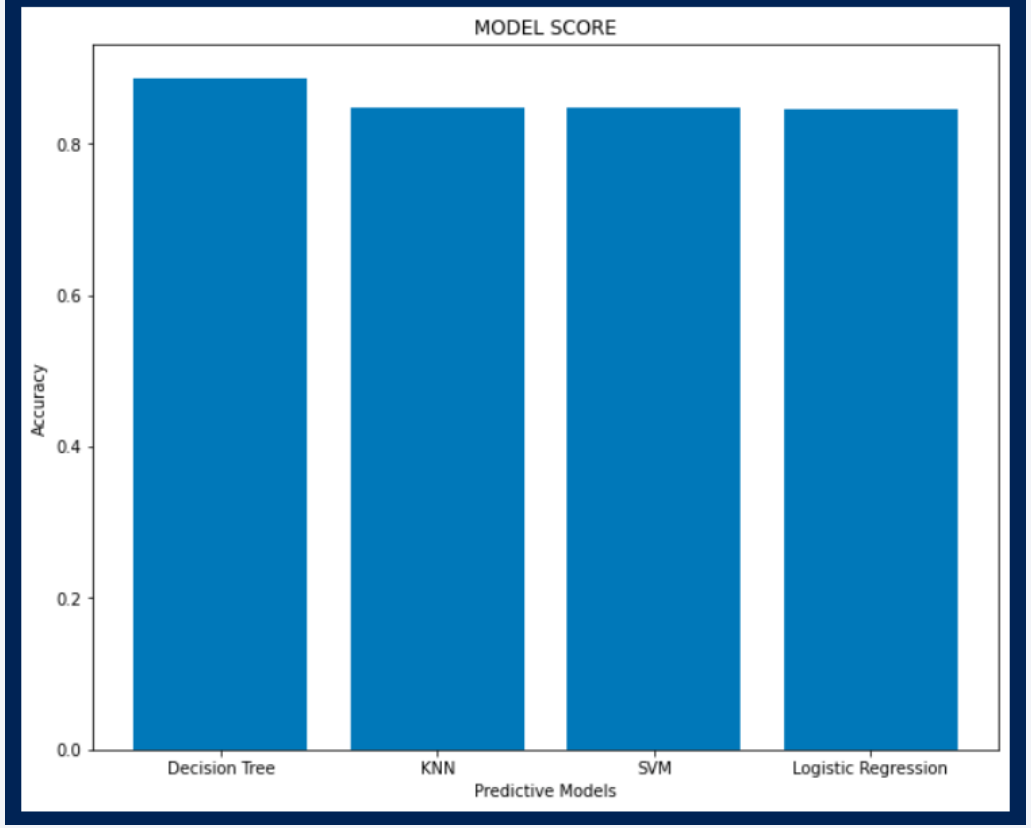
Decision Tree

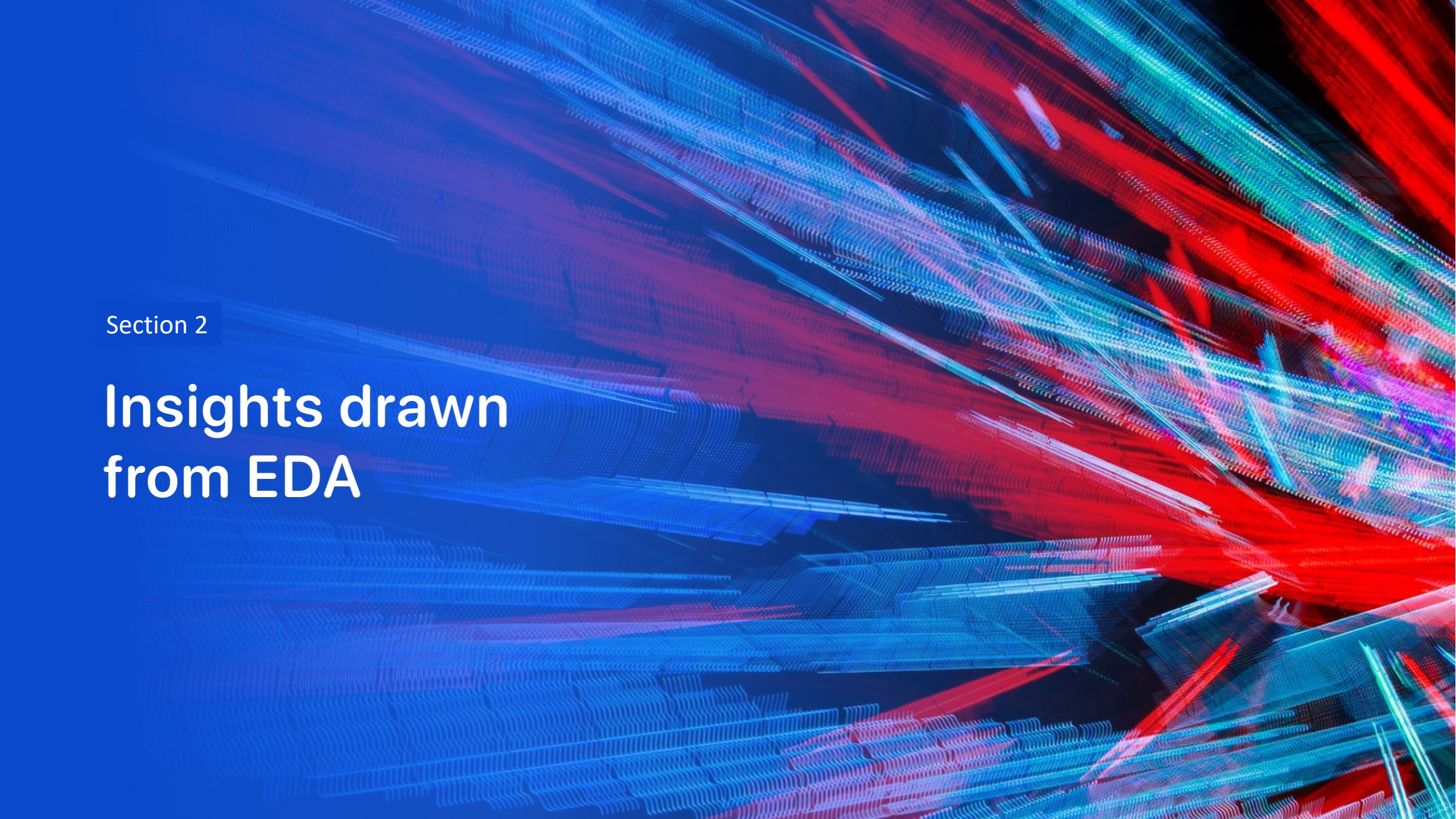


KNN

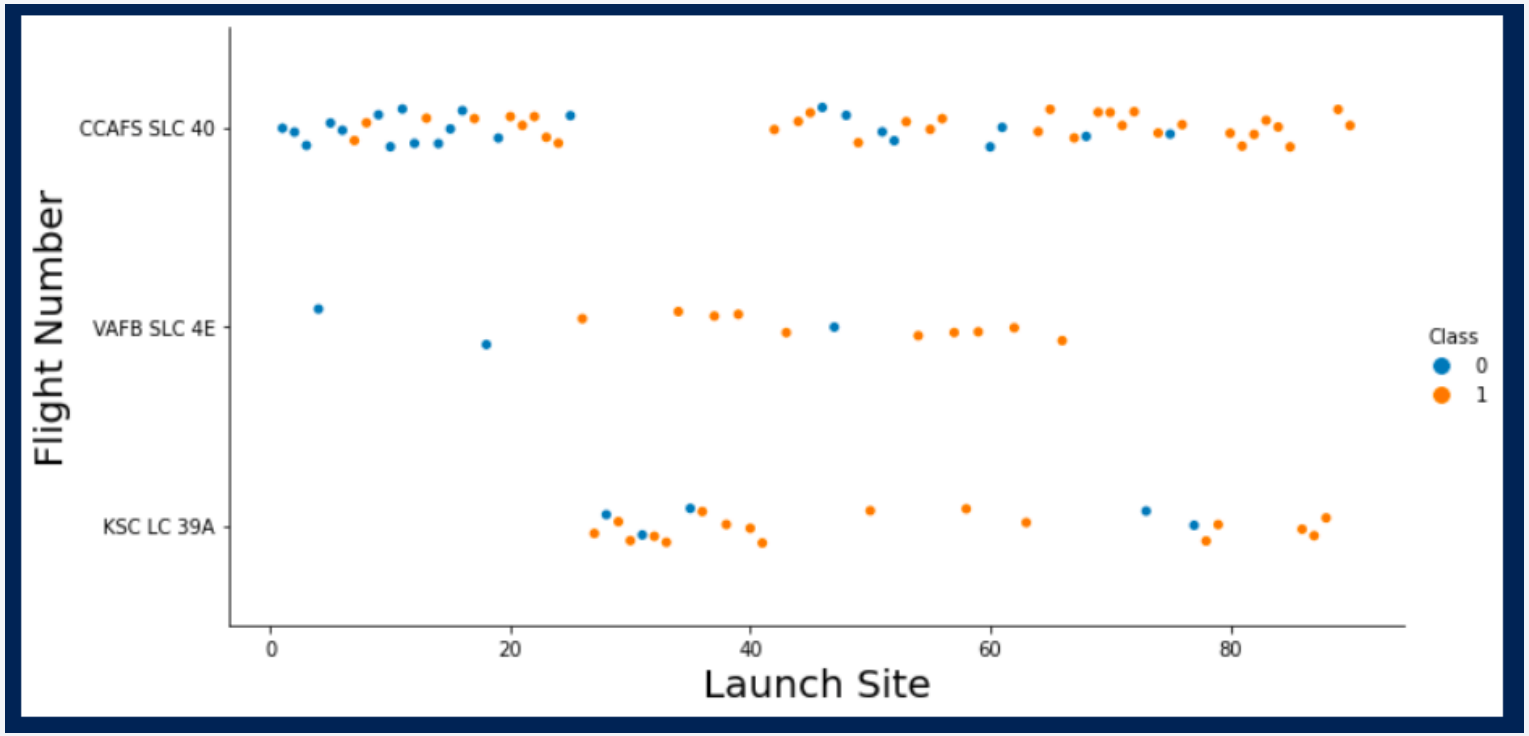



```
Accuracy for Logistics Regression Accuracy: 0.8464285714285713
Accuracy for Support Vector Machine Accuracy: 0.8482142857142856
Accuracy for Decision tree Accuracy: 0.8875
Accuracy for K nearsdt neighbors Accuracy 0.8482142857142858
```




Section 2
Insights drawn from EDA

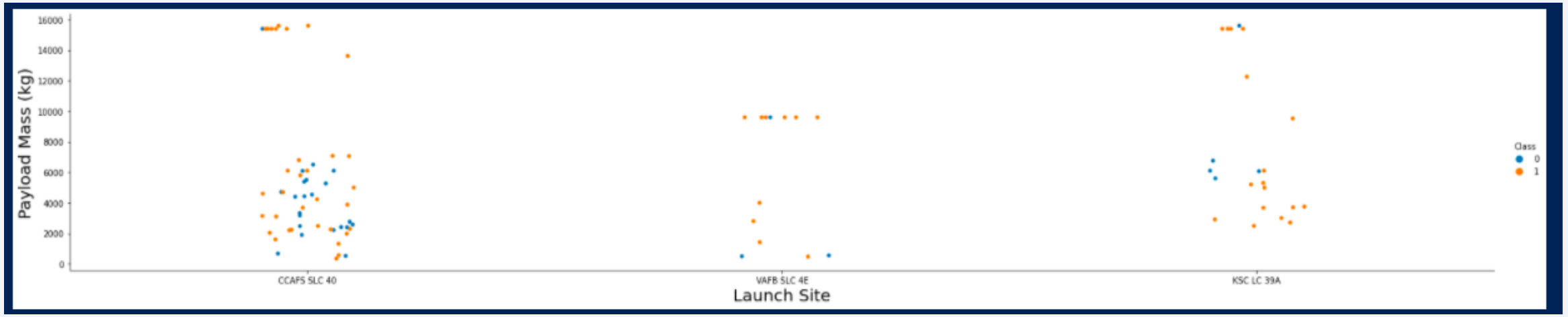
Flight Number vs. Launch Site



Launches from CCAFS SLC are higher.

VAFB is relatively consistent on success launches

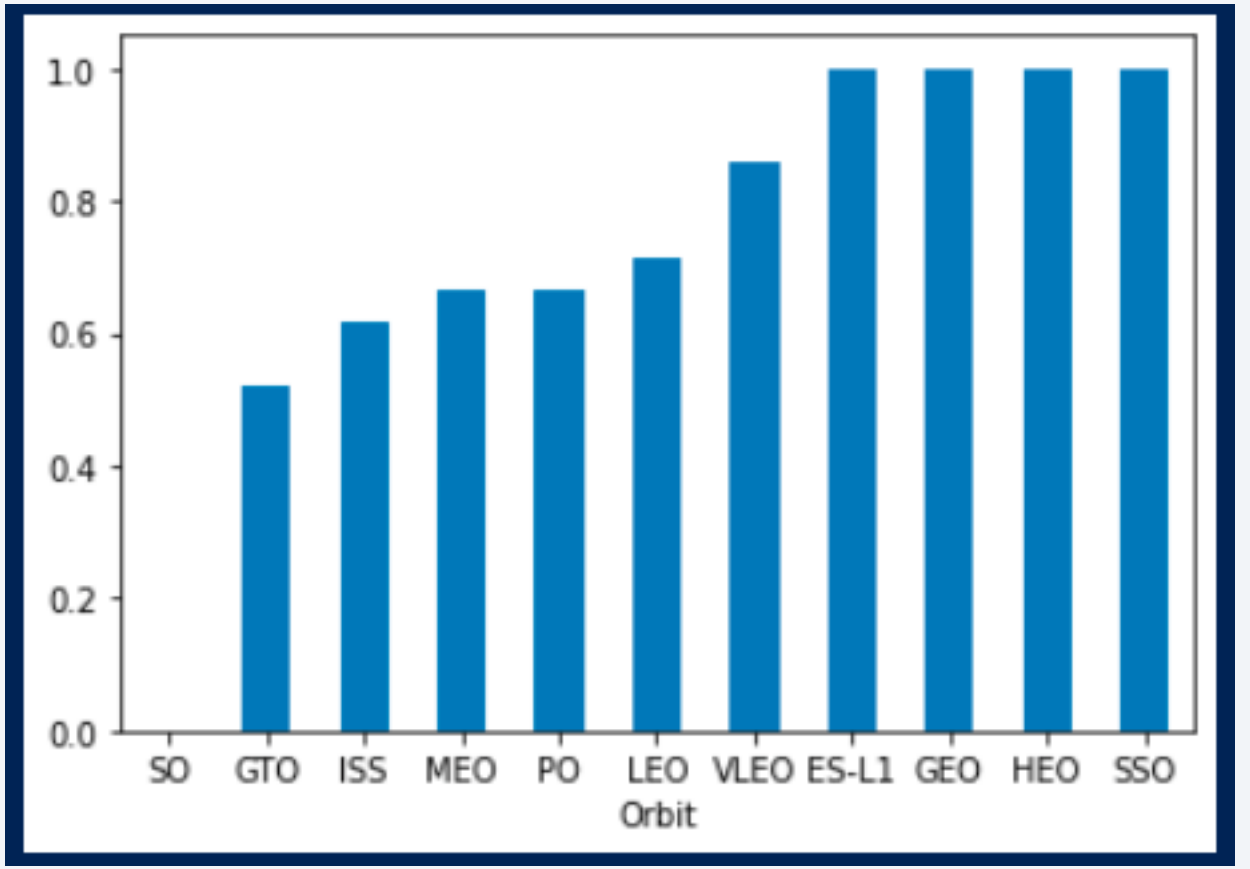
Payload vs. Launch Site



For the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000)

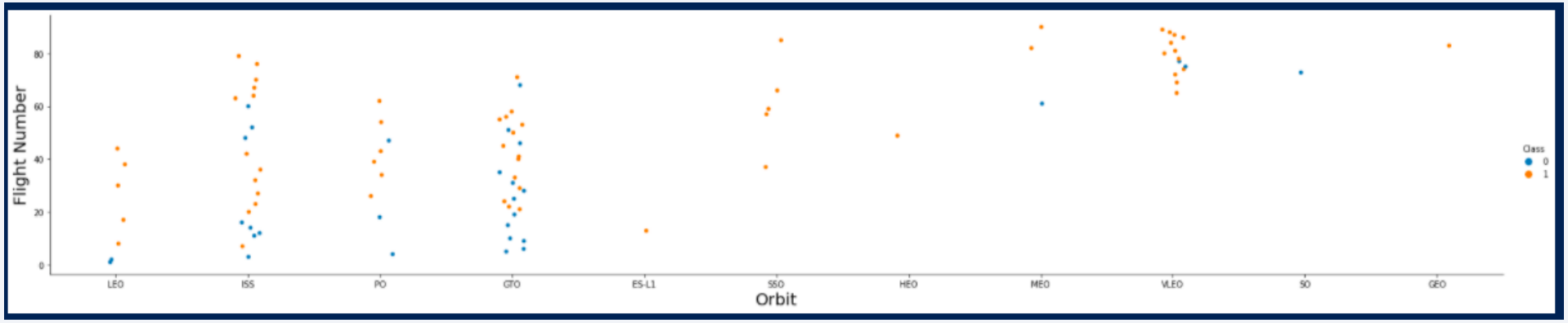
The greater the payload the more likely Phase 1 will return

Success Rate vs. Orbit Type



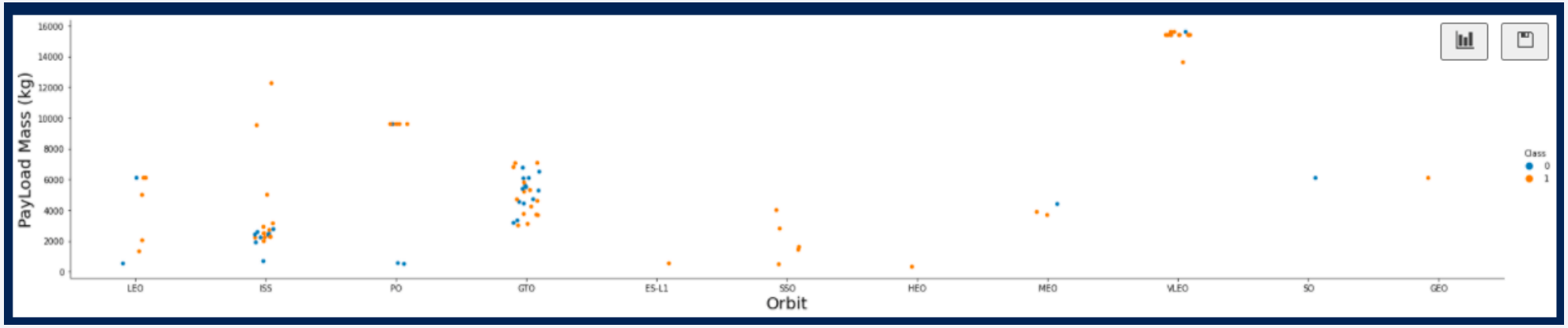
GEO HEO and SSO are the most successful orbits to launch

Flight Number vs. Orbit Type



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.

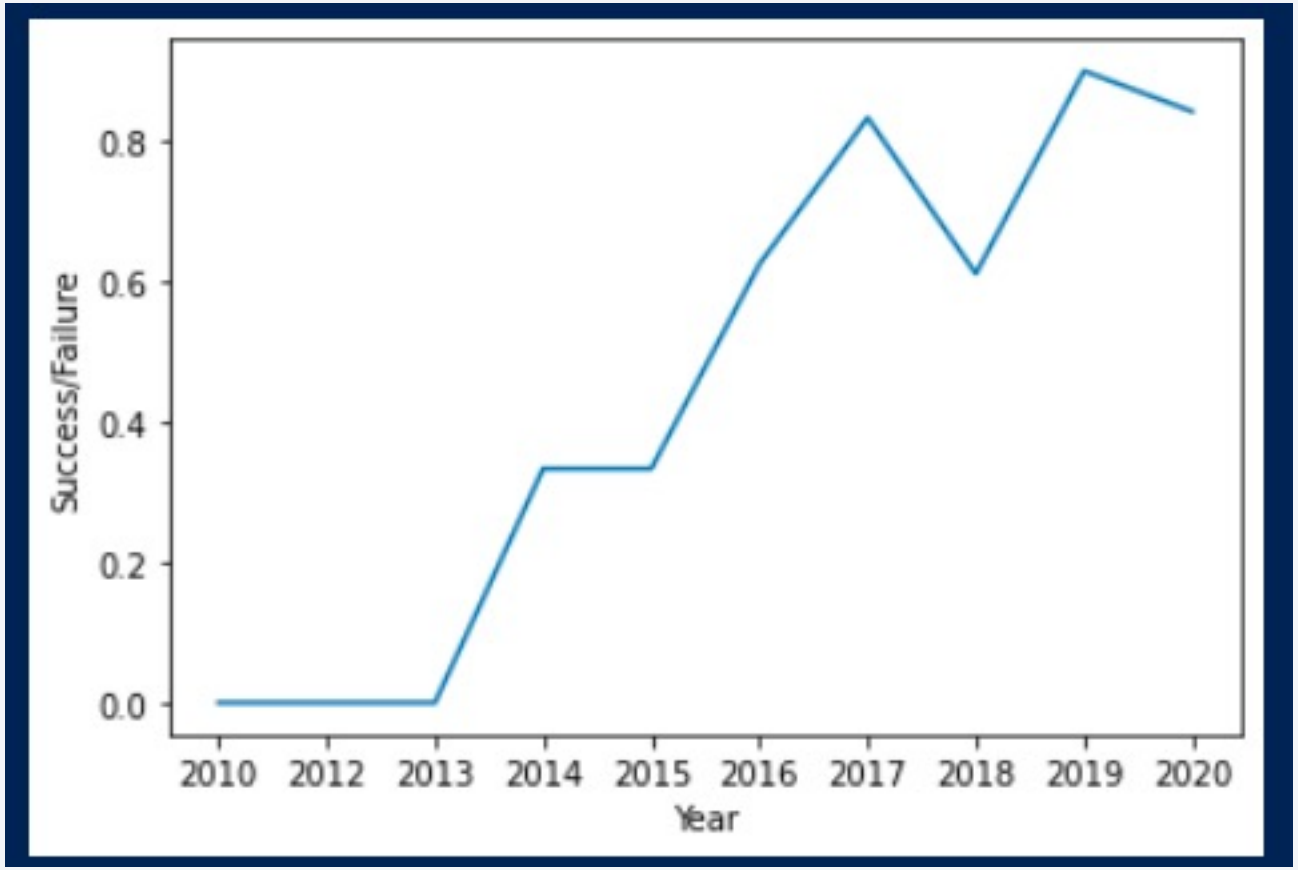
Payload vs. Orbit Type



With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend



As time passes the success rate has proportionally increased

All Launch Site Names

Select distinct(LAUNCH_SITE) from SPACEXBTL

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

Launch Site Names Begin with 'CCA'

- Select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5;

DATE	TIME__UTC_	BOOSTER_VERSION	LAUNCH_SITE
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40

Total Payload Mass

- Select sum(PAYLOAD_MASS__KG_) as total from SPACEXTBL where CUSTOMER = 'NASA (CRS)';

TOTAL
45596

Average Payload Mass by F9 v1.1

- Select avg(PAYLOAD_MASS__KG_) as total from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1';

TOTAL
2928

First Successful Ground Landing Date

- select min(DATE) from SPACEXTBL where LANDING__OUTCOME like '%Success%';

1
2015-12-22

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

Successful Drone Ship Landing with Payload between 4000 and 6000

- select BOOSTER_VERSION from SPACEXTBL where LANDING__OUTCOME = 'Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000;

Total Number of Successful and Failure Mission Outcomes

- select MISSION_OUTCOME, count(MISSION_OUTCOME) as total from SPACEXTBL GROUP BY MISSION_OUTCOME;

MISSION_OUTCOME	TOTAL
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);

BOOSTERVERSION
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5

2015 Launch Records

- SELECT MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL where EXTRACT(YEAR FROM DATE)='2015' and MISSION_OUTCOME != 'Success';

MISSION_OUTCOME	BOOSTER_VERSION	LAUNCH_SITE
Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40

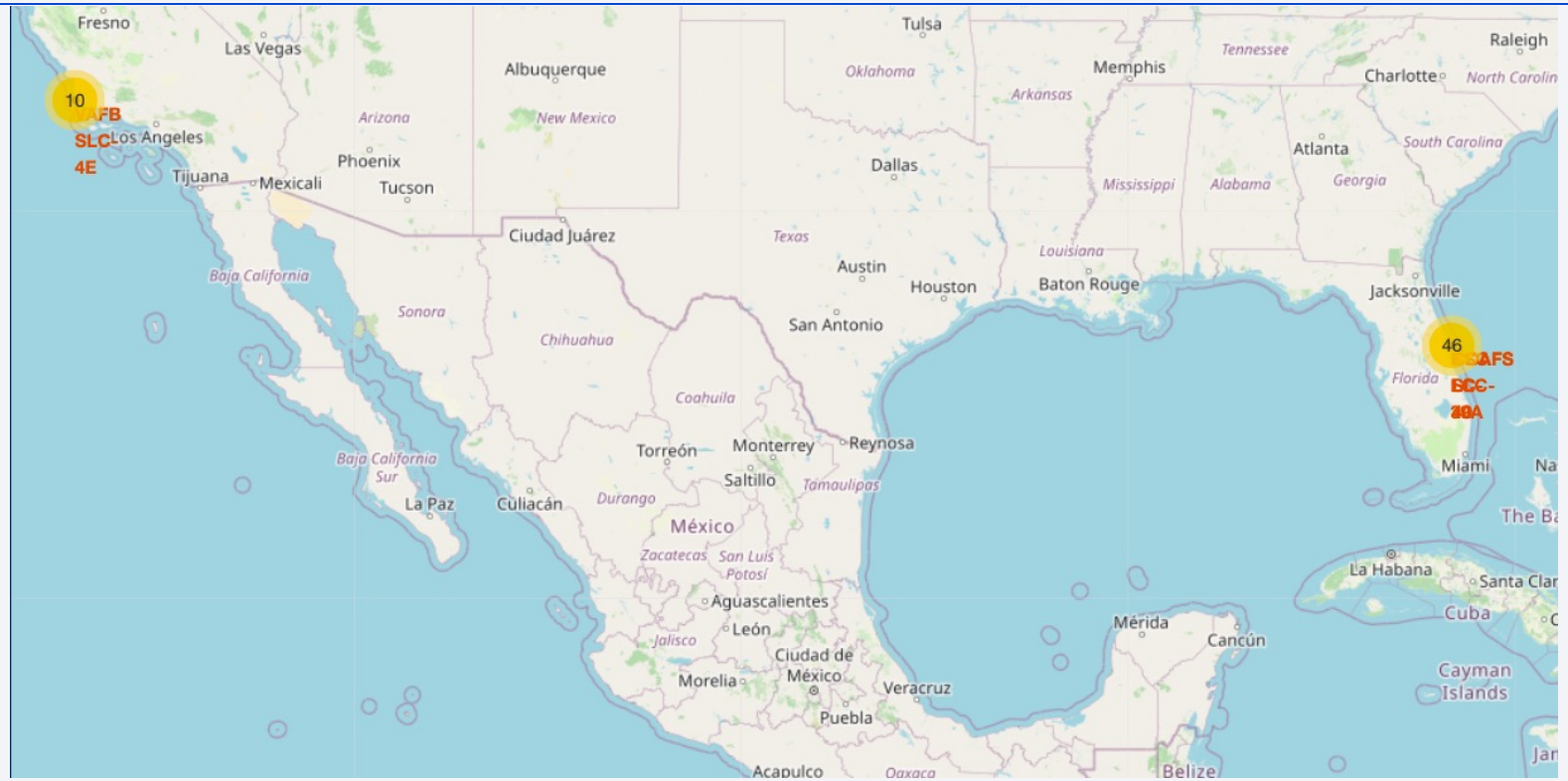
Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- select LANDING__OUTCOME, count(LANDING__OUTCOME) as total from SPACEXTBL where DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING__OUTCOME ORDER BY total desc;

LANDING__OUTCOME	TOTAL
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3

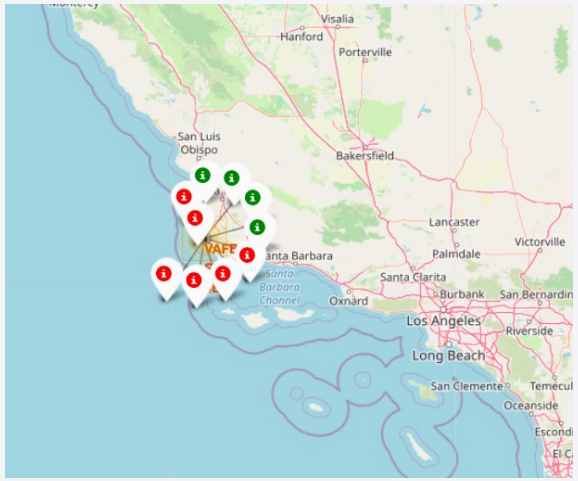

Section 3
Launch Sites
Proximities Analysis

Launch Site Locations

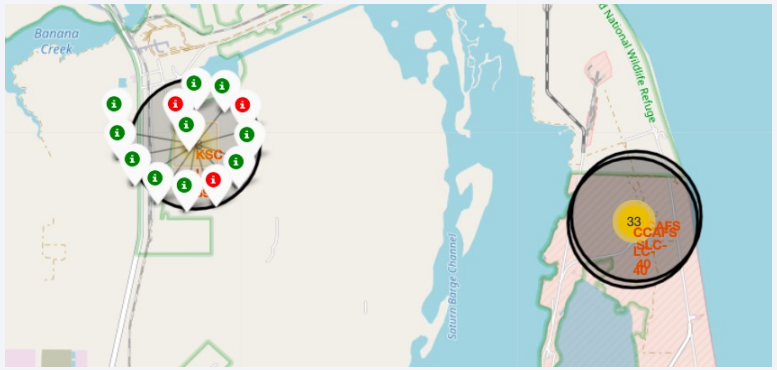


- All Launch sites are very close proximity to coasts to prevent any damage to private property or civilians in case of accident
- Locations are as close to the Equator as possible.

Launch Locations Outcomes



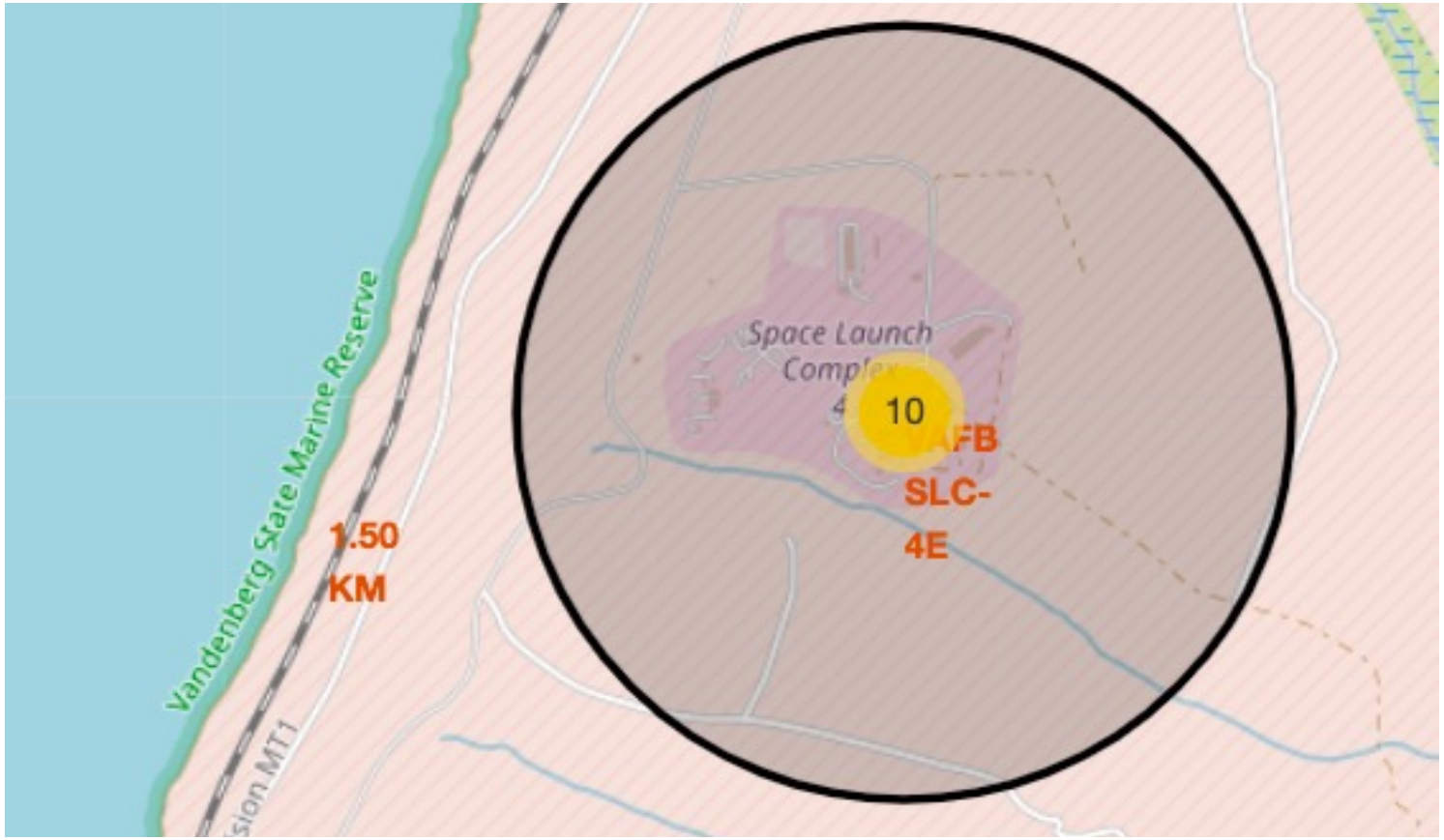
VAFB SLC-4E



KSC LC-39A

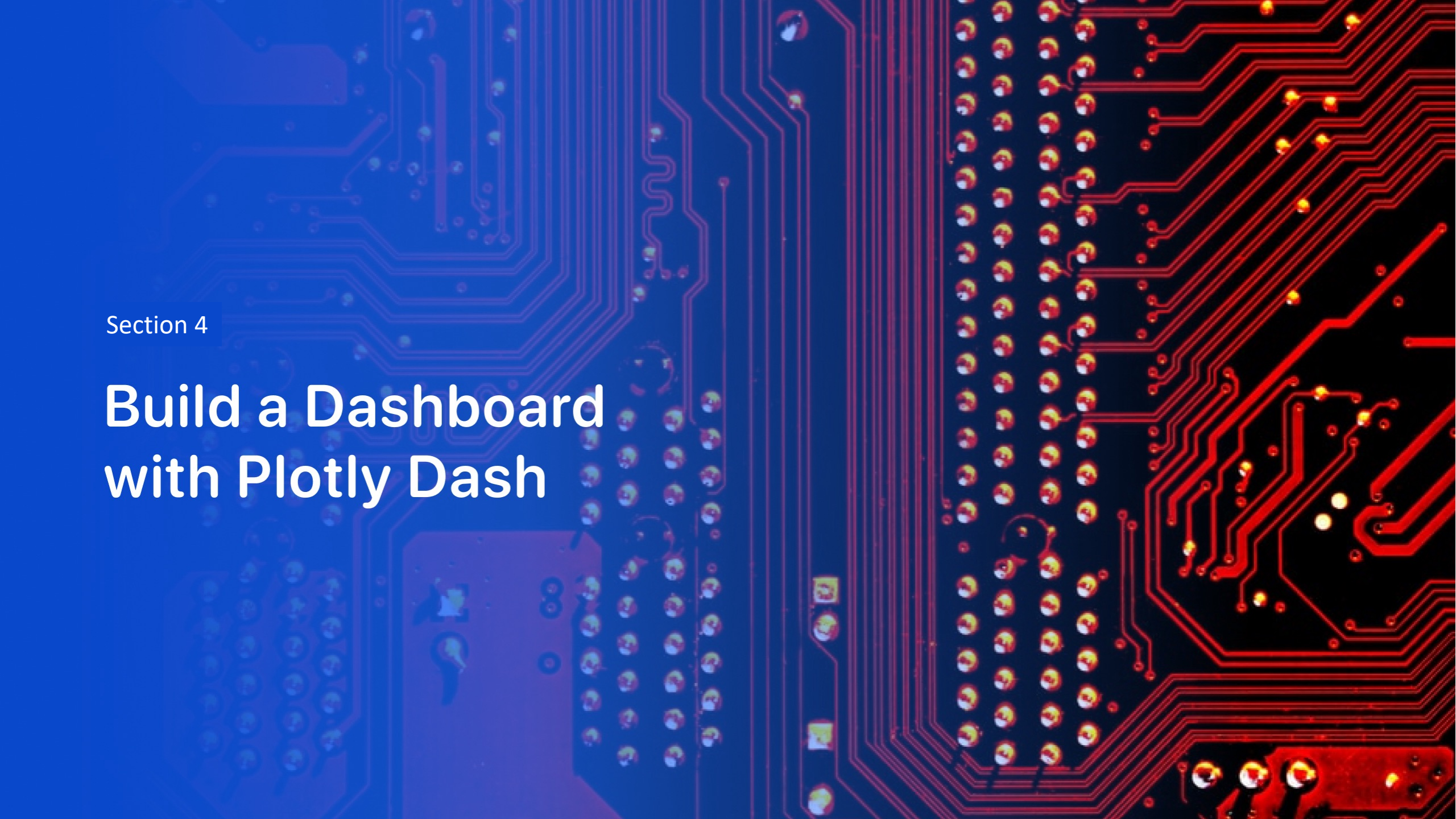
CCAFS LC-40

- Green marker indicates success and Red failure
- KSC seems to be the most successful and CCAFS where most launches happen

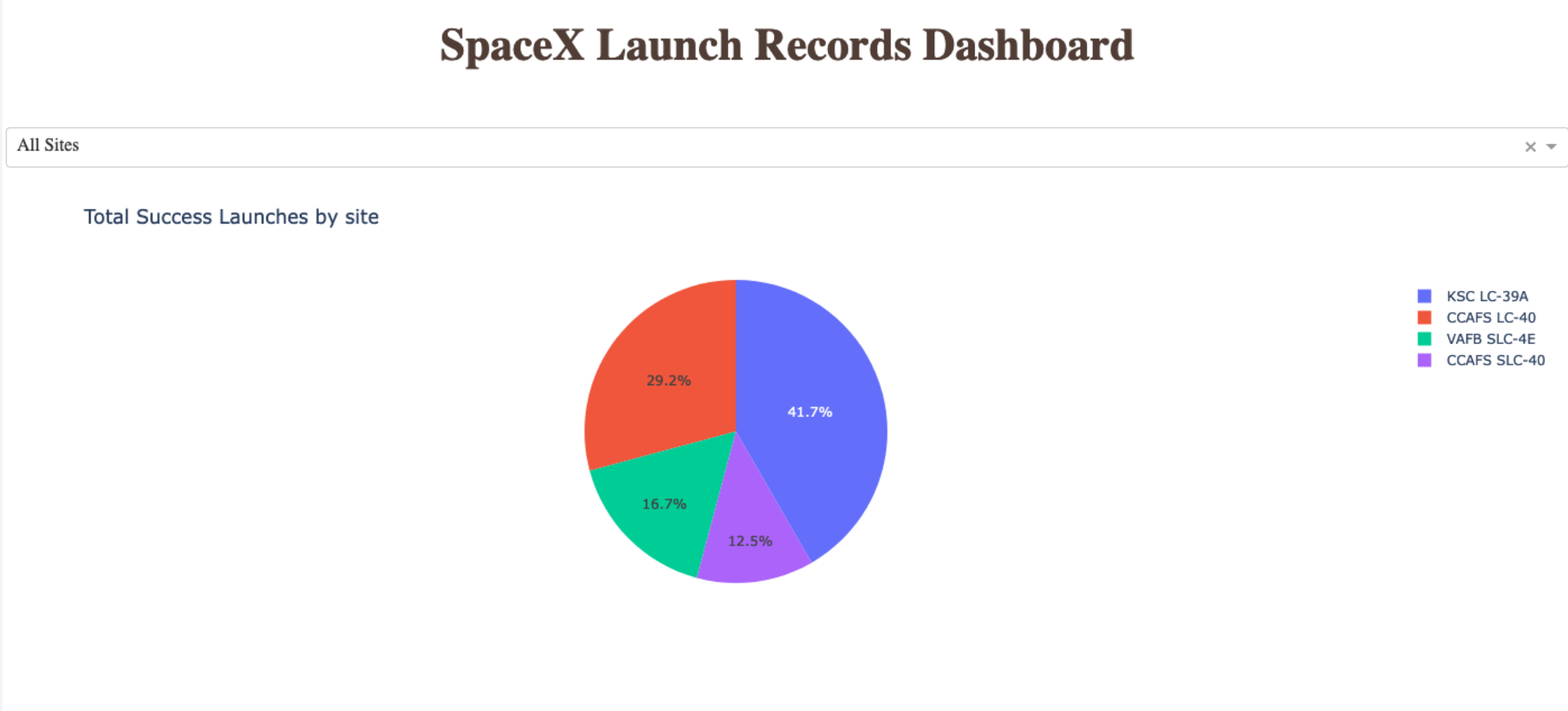


VAFB-SLC4E– Coast Lines

- Coast Line distance of 1.50 KM

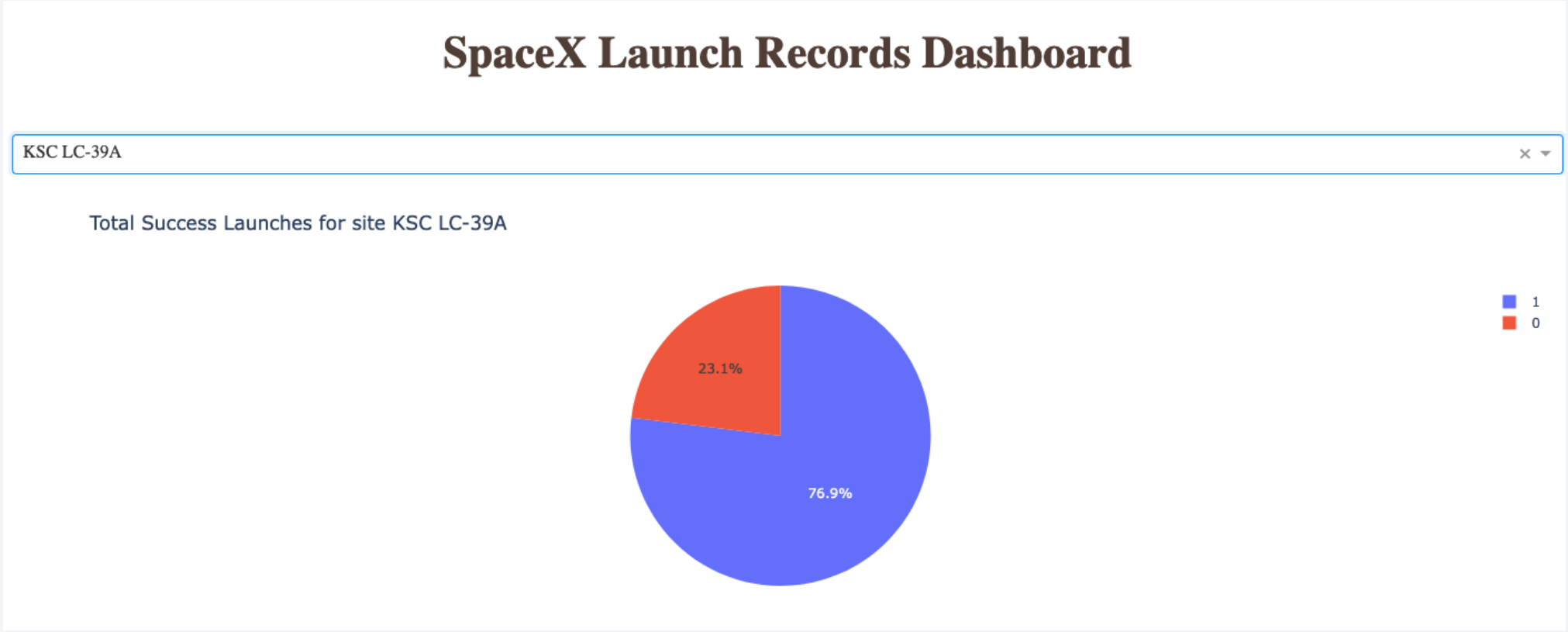

Section 4
Build a Dashboard with Plotly Dash

Launch success for all Sites



- LC40 and LC39A have the largest proportion of success.

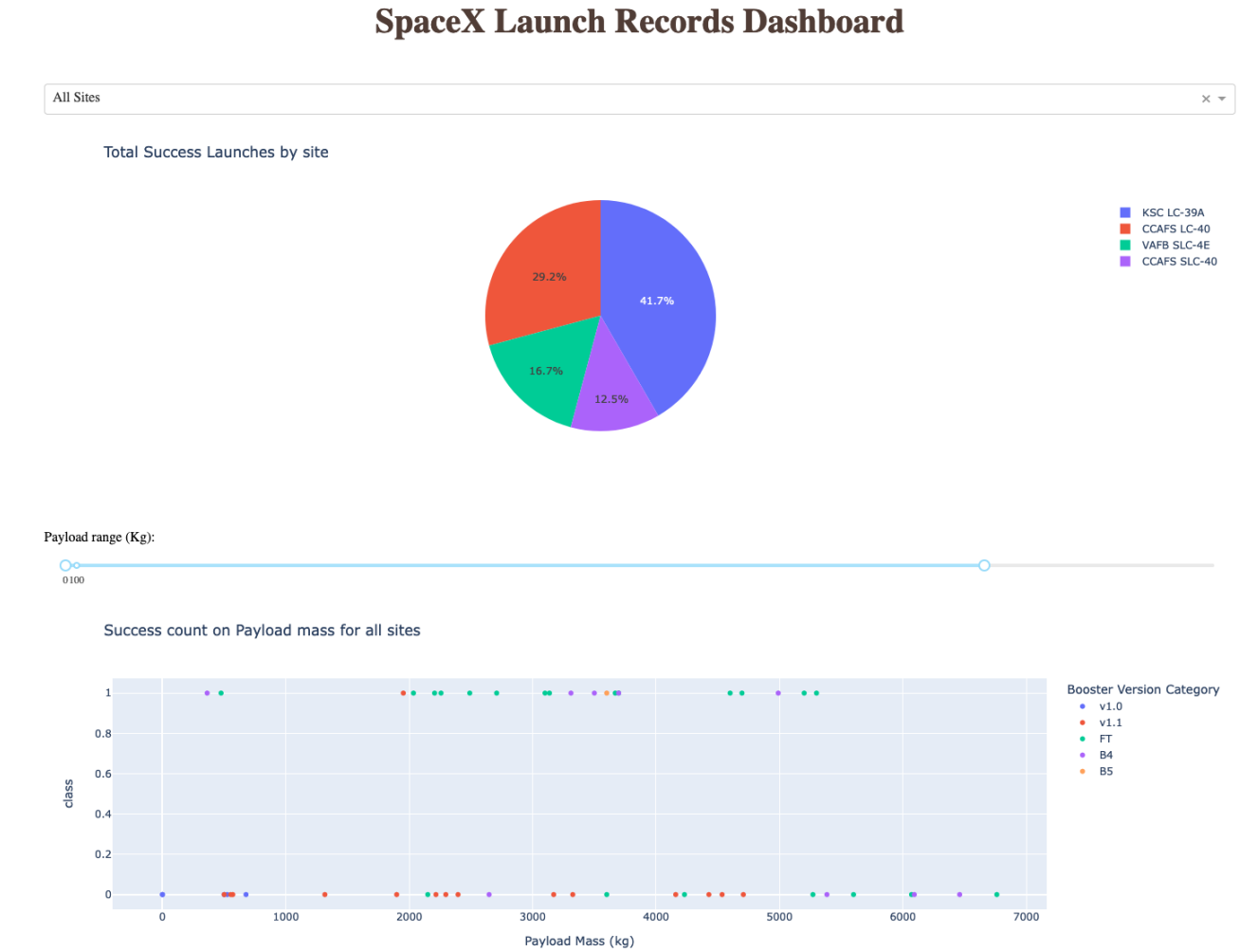
KSC LC 39 Launch Outcome overview



- LC39A have the largest proportion of success from all sites but itself the success rate is not the best.

Success by LaunchSite based on Payload Mass

- The lower the payload the better the chances to have success launches and returns..

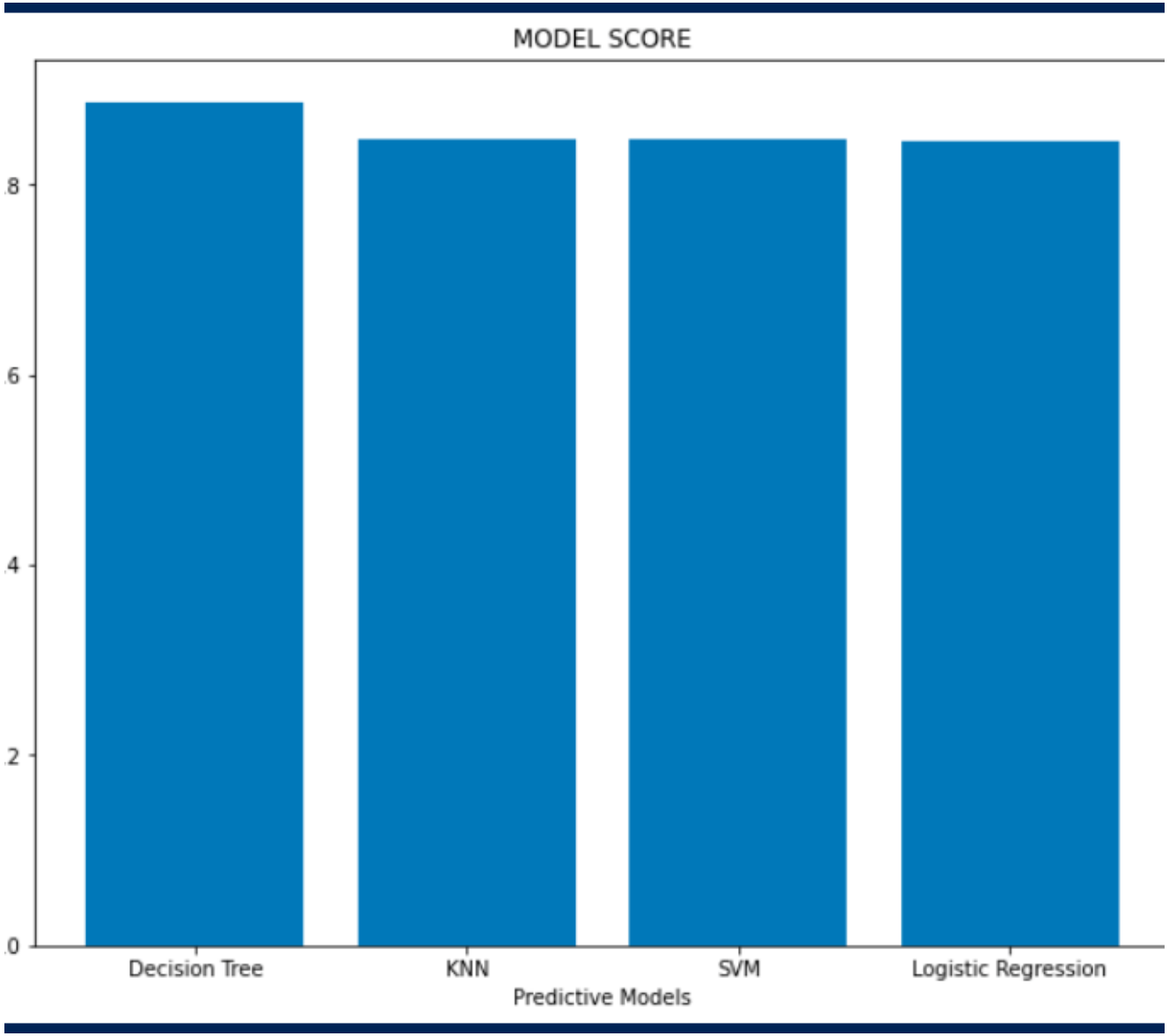


Section 5

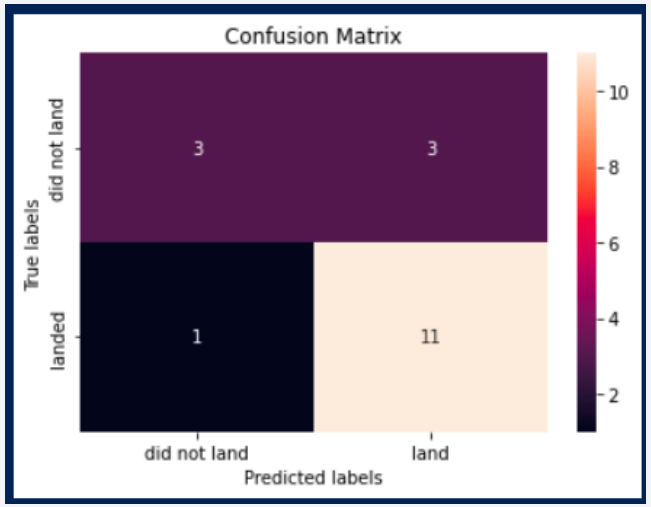
Predictive Analysis (Classification)

Classification Accuracy

- The best model with the Hyperparameters is GT with 89% accuracy



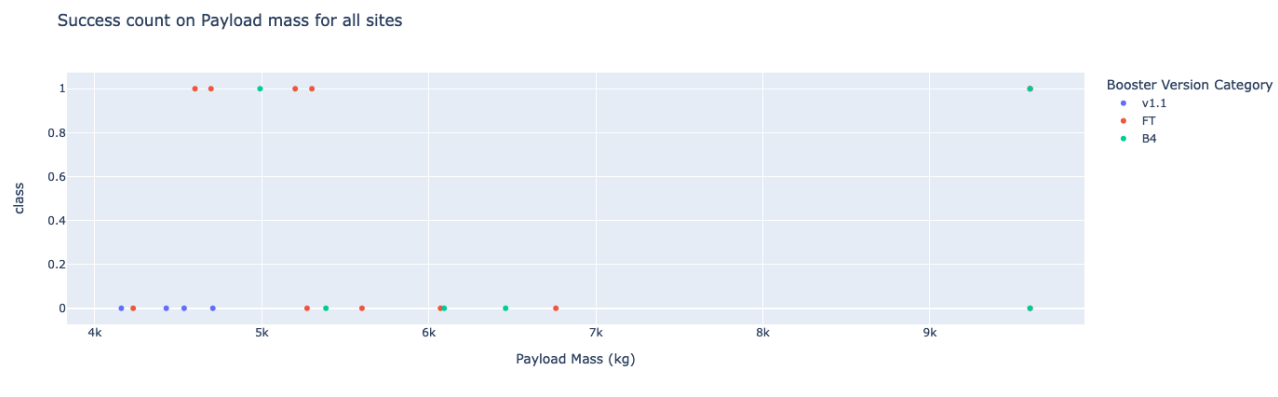
Confusion Matrix

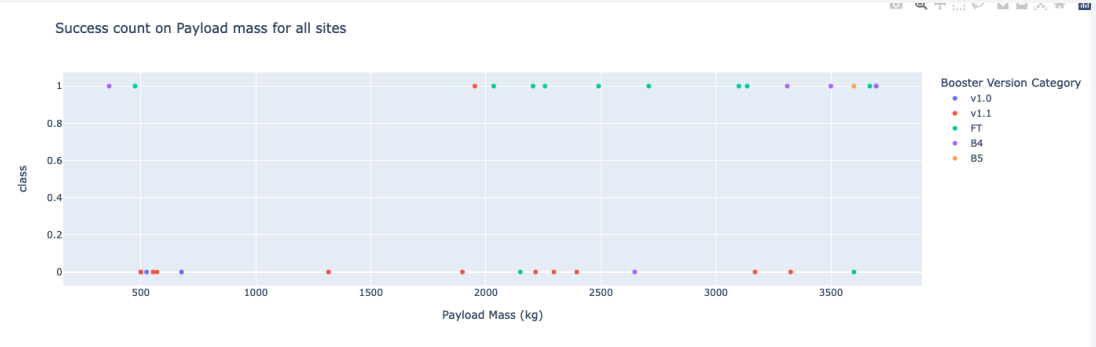


- DT model has a very good performance on predicting Landing.
- It has bad performance for predict no landing – 50% chance to get it correct. Additional training data could help to add more accuracy

Conclusions

- Worth revisiting the model every year as the more time passes more likely success rockets
- Booster and Payload mass are key indicators for success. FT booster is the best performer
- The bigger the mass less chances for success





Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!