IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Brett Wainwright
21 Feb 2022

Outline

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

Executive Summary

- Retrieved data from the SpaceX Wikiepedia page and SpaceX API, classifying successful landings via creation of “class” column. Performed data exploration using SQL, visualization, folium maps, and dashboards. Produced four models utilizing machine learning and visualized the accuracy score of all models. All produced similar results with an accuracy rate of 83.3%. More data is likely need to determine which model could be most accurate.
- Summary of methodologies
  - Data collection
  - Data Wrangling
  - EDA with Data Visualization
  - EDA with SQL
  - Building an Interactive Map with Folium
  - Building a Dashboard with Plotly Dash
  - Predictive Analysis (Classification)
- Summary of all results
  - Exploratory Data Analysis Results
  - Interactive Analytics Demo via Screenshots
  - Predictive Analysis Results

Introduction

- Project background and context
  - With several companies making space travel affordable for all, an era of commercial space has arrived. Currently hailed as the most successful of these companies is SpaceX, predominantly attributed to the relatively low cost of launching a rocket. While some providers are advertising a launch cost of ~US$165 million, SpaceX advertises its Falcon 9 rocket's launch cost ~US$62 million. As these savings are largely due to the ability to reuse the first stage, we will attempt to predict if the Falcon 9 first stage will land successfully.
- Problems you want to find answers
  - Correlations between rocket variables and a successful landing rate.
  - Optimum conditions for the best results and successful landing rate.

Section 1

Methodology

Methodology

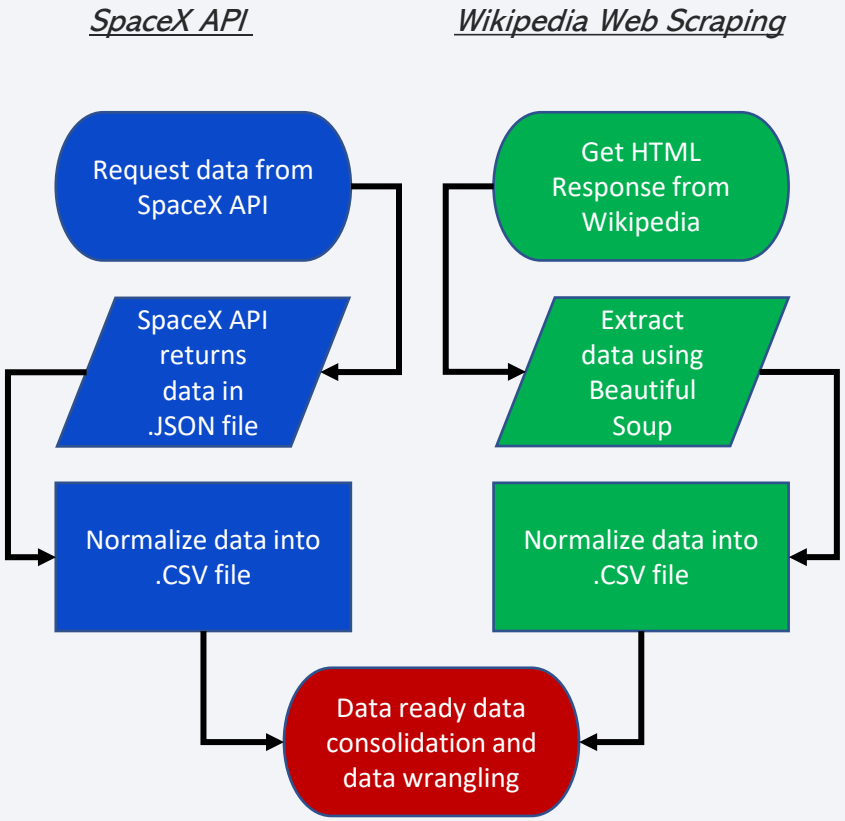
Executive Summary

- Data collection methodology:
  - SpaceX API and Web Scraping Falcon 9 and Falcon Heavy Launches Records (Wikipedia)
- Perform data wrangling
  - Convert outcomes to Training Labels with the booster successfully/unsuccessfully landed
- Perform Exploratory Data Analysis (EDA) using Visualization and SQL
- Perform Interactive Visual Analytics using Folium and Plotly Dash
- Perform Predictive Analysis using Classification models
  - Find best Hyperparameter for SVM, Classification Trees, and Logistic Regression models

Data Collection

- The data collection process includes a combination of API requests from the SpaceX API, and web scraping data from a table on the SpaceX Falcon 9 and Falcon Heavy Launches Records Wikipedia page.
  - SpaceX API Data Columns:
    - FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude
  - Wikipedia Web Scraping Data Columns:
    - Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, TimeDescribe how data sets were collected.

Data Collection Process Flowcharts:



Data Collection – SpaceX API

- Data collection with SpaceX REST calls

1) Response from API
2) Convert Response to .JSON file
3) Construct dataset
4) Clean data and export to .CSV file

[Github URL]

①
```
spacex_url="https://api.spacexdata.com/v4/launches/past"

response = requests.get(spacex_url)
```

②
```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

③
```
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}
```

④
```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = launch_df[launch_df['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'] = data_falcon9['PayloadMass'].replace(np.nan, PayloadMass_mean)
data_falcon9.isnull().sum()
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection – Wikipedia Web Scraping

- Wikipedia web scraping process

1) Getting Response from HTML

2) Creating Beautiful Soup object

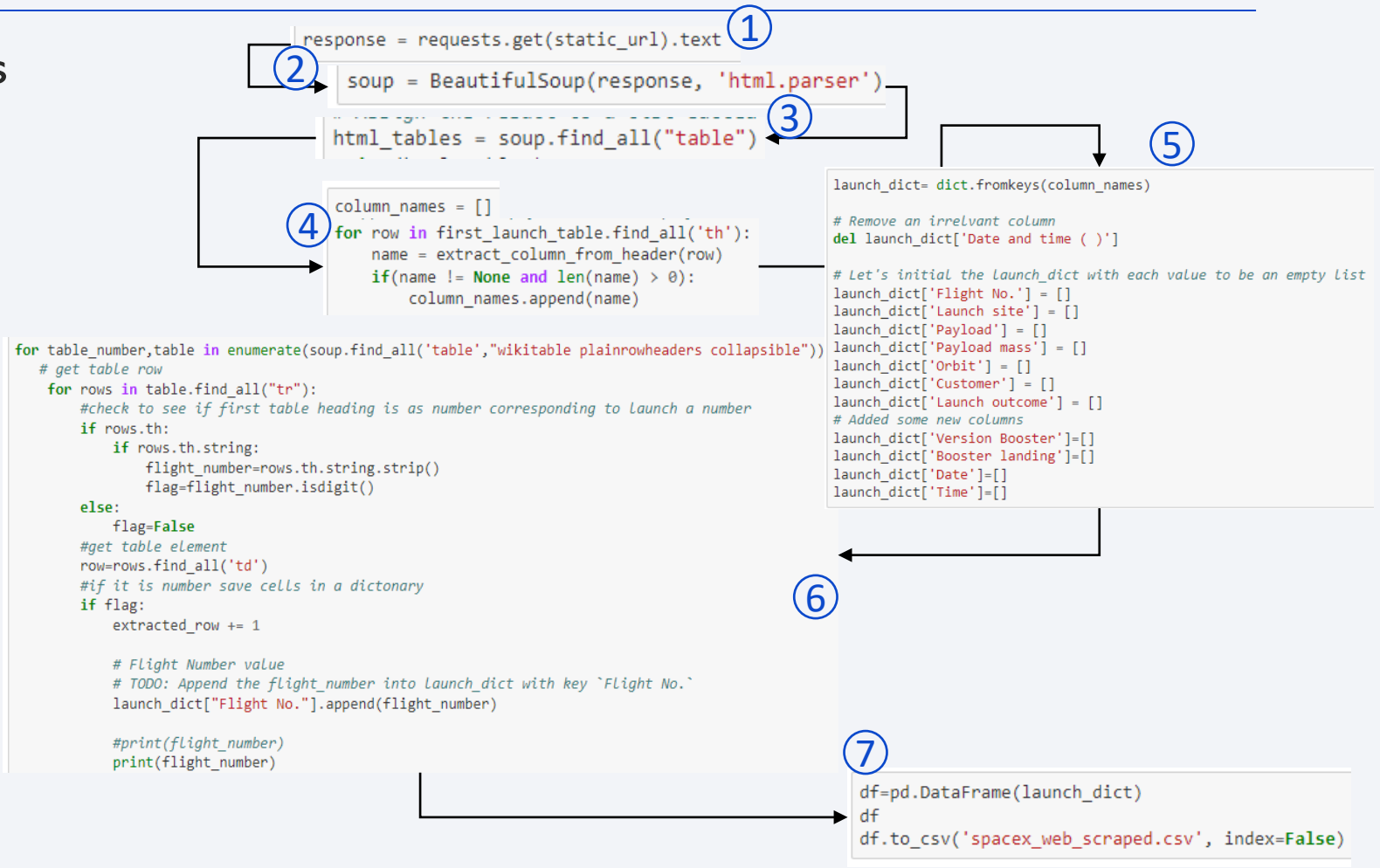
3) Finding Tables

4) Getting column names

5) Creating dictionary

6) Appending data to keys

7) Converting to data frame then .CSV file



Github URL

Data Wrangling

- EDA Analysis

1) Check null values
2) Calculate number of launches per site
3) Calculate number/occurrence per orbit
4) Calculate number/occurrence of outcome per orbit
   - Possible Outcomes:
     - True Ocean: Successfully landed to a specific region of the ocean
     - False Ocean: Unsuccessfully landed to a specific region of the ocean.
     - True RTLS: Successfully landed to a ground pad.
     - False RTLS: Unsuccessfully landed to a ground pad.
     - True ASDS: Successfully landed to a drone ship.
     - False ASDS: Unsuccessfully landed to a drone ship.
     - None ASDS / None None: Failure to land.
5) Convert results to training label
   - Successful = 1
   - Unsuccessful = 0

[Github URL]

```
① df.isnull().sum()/df.count()*100

② # Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()

③ # Apply value_counts on Orbit column
df.Orbit.value_counts()

④ # landing_outcomes = values on Outcome column
landing_outcomes = df.Outcome.value_counts()
landing_outcomes

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

{'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}

⑤ # 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)
```

EDA with Data Visualization

- Scatter Chart
  - Scatter plots show how greatly two variables are correlated, influence of one variable on another, and is useful in larger data sets.
    - Used on the following:
      - Flight No. vs Launch Site
      - Payload vs Launch Site
      - Flight No. vs Orbit Type
      - Payload vs Orbit Type
- Bar Chart
  - Bar charts easily show comparisons between categorical data, with two axis, one numerical, the second, describing types of categories.
    - Used on the following:
      - Orbit Type vs Success Rate
- Line Chart
  - Line charts show data trends over time very clearly, and may help in predicting future results.
    - Used on the following:
      - Year vs Success Rate

Github URL

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
- Loaded the data set into IBM DB2 Database and queried the data utilizing SQL Python integration.
- These queries were utilized to gain a better understanding information within the dataset, such as:
    - Launch site names
    - Mission outcomes
    - Payload sizes
    - Landing Outcomes

Github URL

Build an Interactive Map with Folium

- Objects created and added to folium map
  - Markers indicating all launch sites
  - Markers indicating whether launches were successful or not
  - Lines indicating distances between a launch and various proximities
- These objects once placed on a map help to identify any geographical patterns about the launch sites.
  - Questions which can be answered:
    - Are launch sites in close proximity to railways? - Yes
    - Are launch sites in close proximity to highways? - Yes
    - Are launch sites in close proximity to coastlines? - Yes
    - Do launch sites keep certain distance from cities? - Yes

Github URL

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- The dashboard contains a both a pie chart and a scatter plot.
- Pie Chart – demonstrates total successful launches by sites
  - Pie charts can be selected to indicate a successful landing distribution across all launch sites or to indicate the success rate of individual launch sites.
- Scatter Plot
  - Scatter plots can be used to observe the relationship between two variables, in this case, mission outcome and Pay Load Mass (kg). Additionally, the colour coding provides further insight indicating Booster Version Categories.

Github URL

Predictive Analysis (Classification)

- EDA and determine Training Labels

1) Create a column for “Class”
2) Standardize data
3) Split data into training and test data

- Determine best parameters for:
  - Logistic Regression
  - SVM
  - Classification Trees

- Determine best method

[Github URL]

①
```
Y = data['Class'].to_numpy()
```

②
```
# students get this
transform = preprocessing.StandardScaler()
```
```
X = transform.fit_transform(X)
```

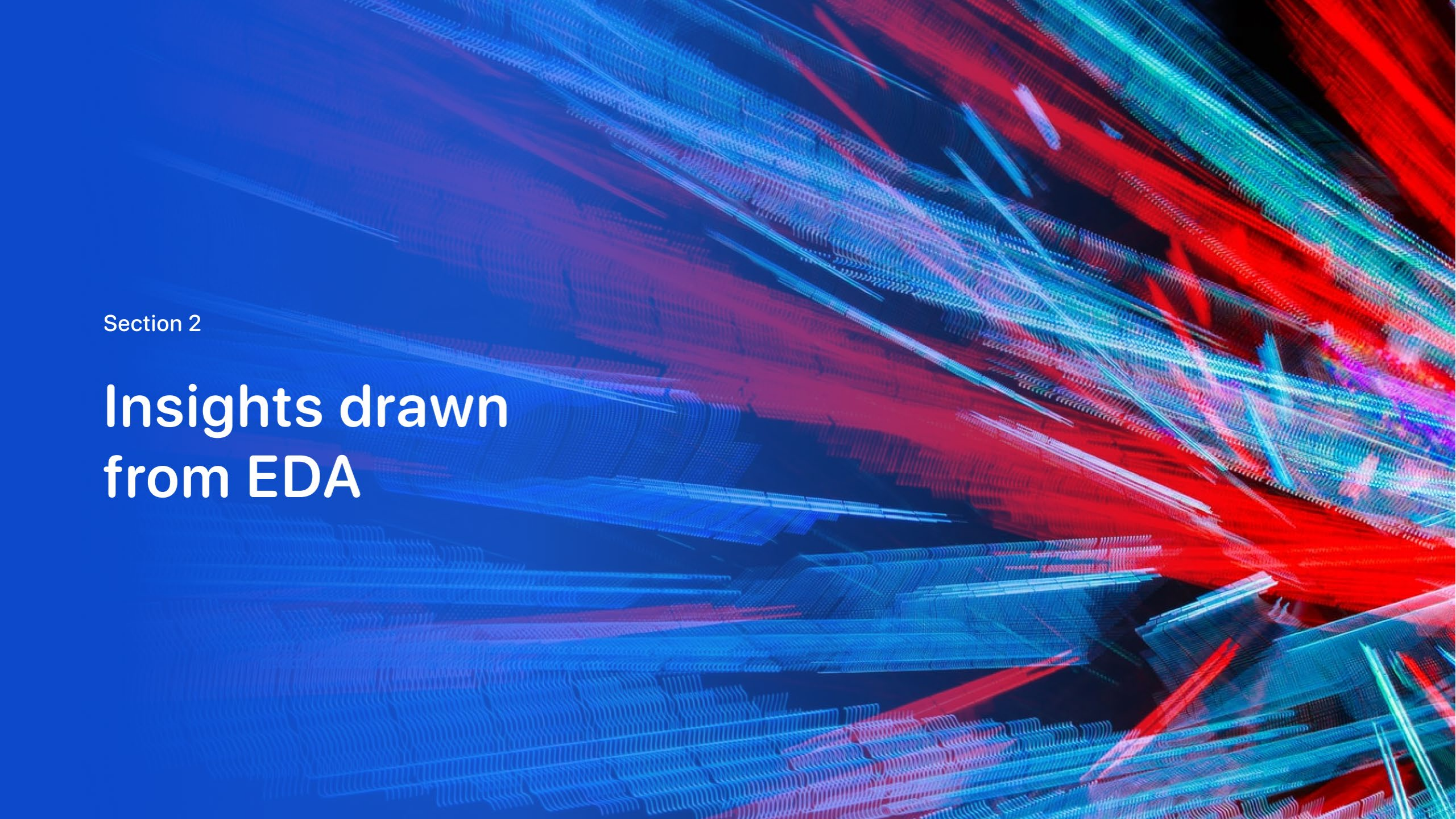
③
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)
```

④
```
parameters ={'C':[0.01,0.1,1],
             'penalty':['l2'],
             'solver':['lbfgs']}
```
```
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge
lr=LogisticRegression()
logreg_cv = GridSearchCV(lr, parameters, cv = 10)
logreg_cv.fit(X_train, Y_train)
```
```
parameters = {'kernel':('linear', 'rbf','poly','rbf', 'sigmoid'),
              'C': np.logspace(-3, 3, 5),
              'gamma':np.logspace(-3, 3, 5)}
svm = SVC()
```
```
svm_cv = GridSearchCV(svm, parameters, cv = 10)
svm_cv.fit(X_train, Y_train)
```
```
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
```
```
tree_cv = GridSearchCV(tree,parameters,cv=10)
tree_cv.fit(X_train, Y_train)
```

⑤
```
import numpy as np
import matplotlib.pyplot as plt

plt.barh(methods, accuracy)
plt.xlabel('Accuracy')
plt.ylabel('Method')
plt.show()
```

Results

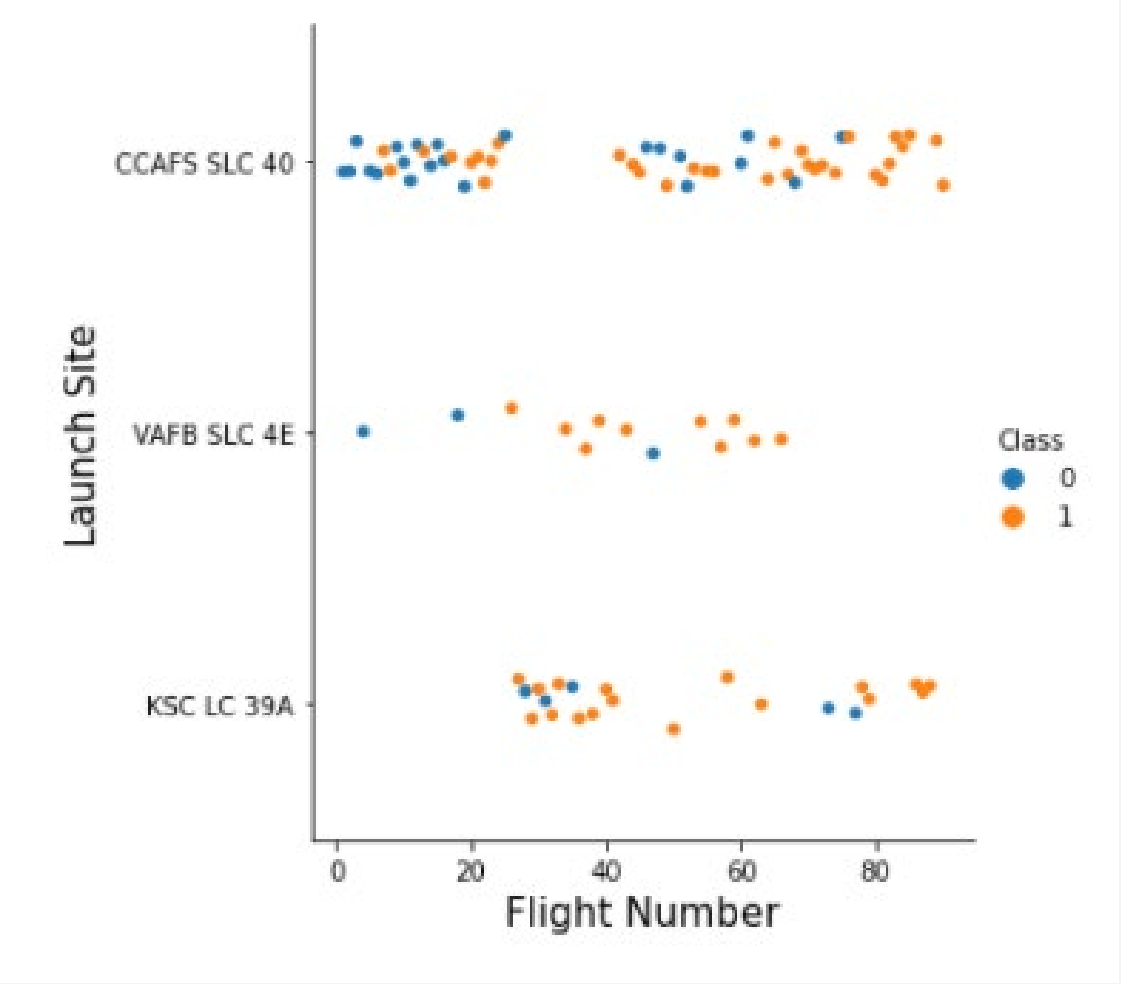
- Exploratory Data Analysis (EDA) Results
- Interactive Analytics Demo via Screenshots
- Predictive Analysis Results


Section 2
Insights drawn
from EDA

Flight Number vs. Launch Site

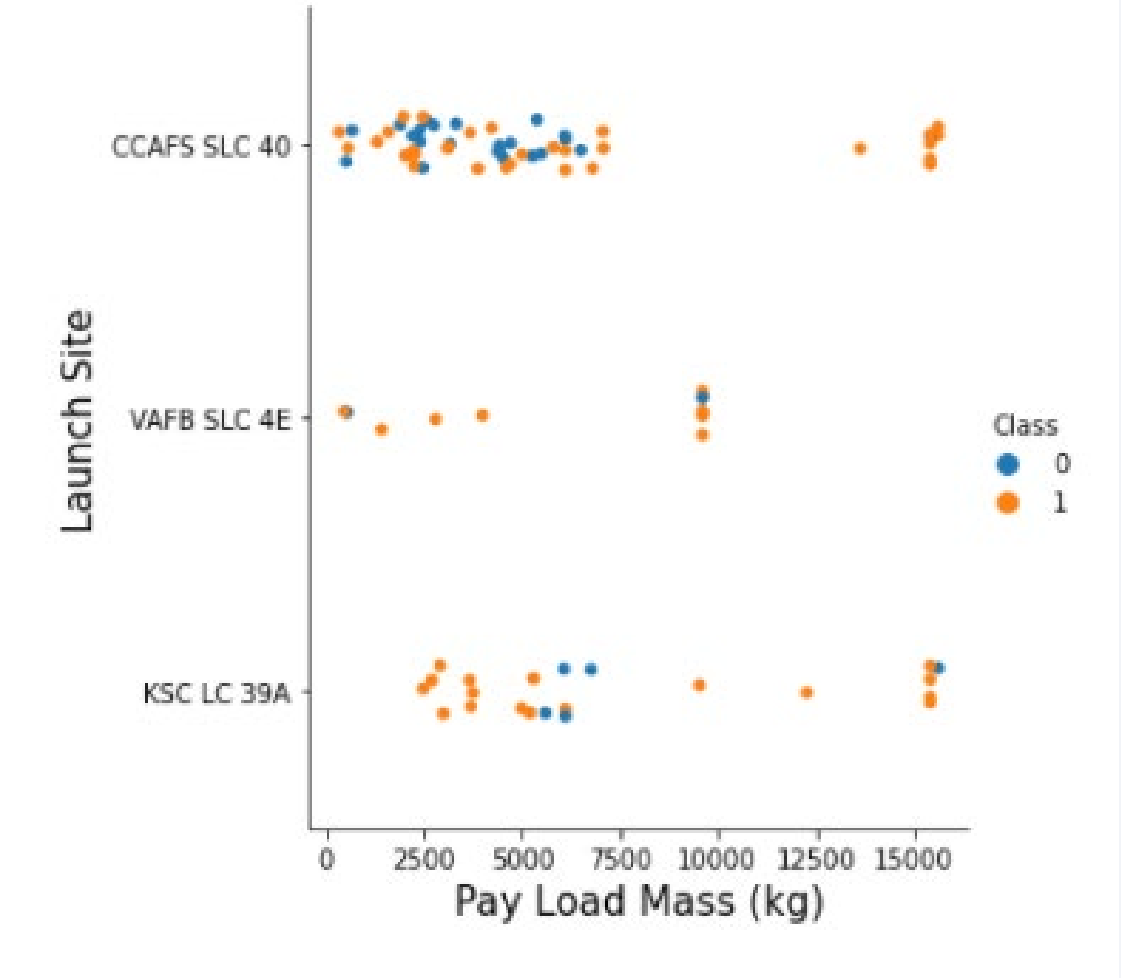
- Successful and Unsuccessful launches are represented by the Blue (Class 0) and the Orange (Class 1) points respectively.

- This scatter plot indicates that the number of successful launches increases with the number of flights, with the success rate increasing substantially after 20 flights.



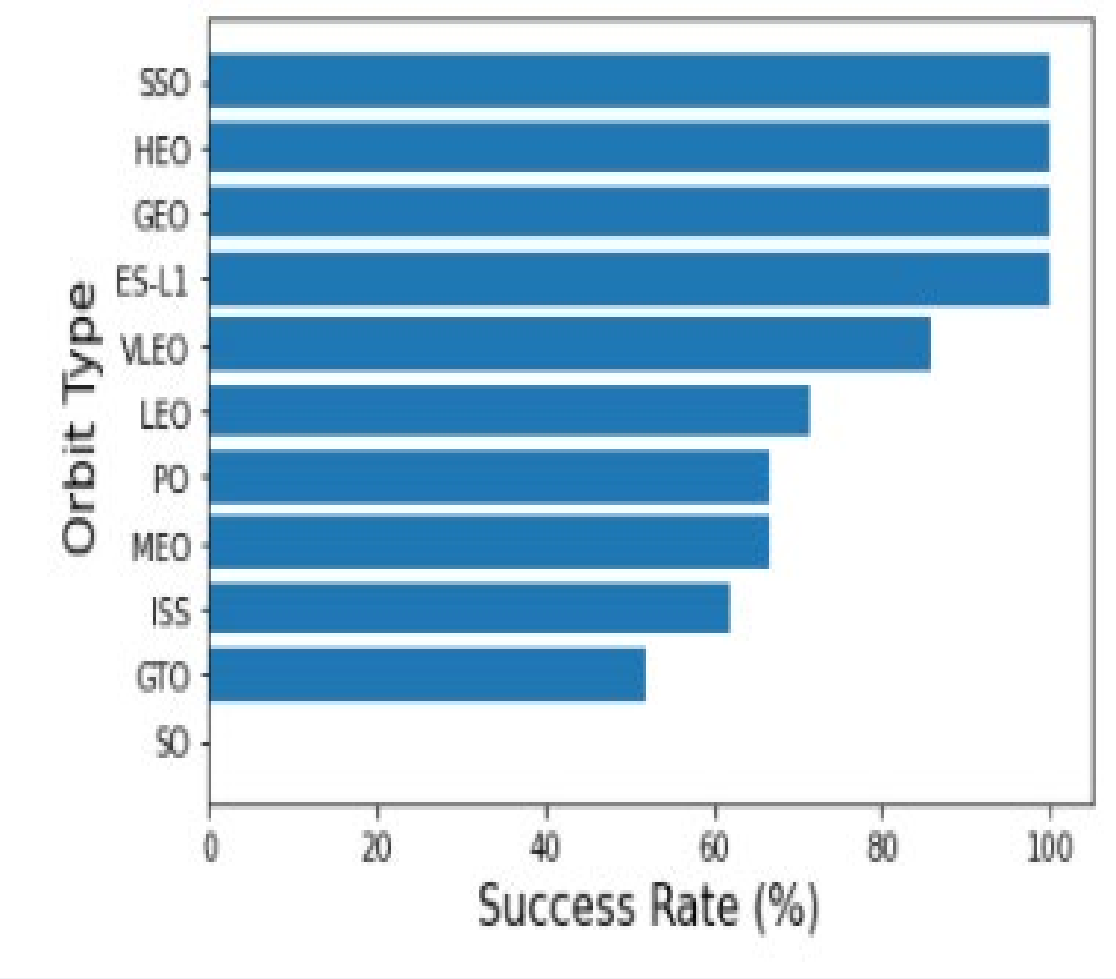
Pay Load vs. Launch Site

- Successful and Unsuccessful launches are represented by the Blue (Class 0) and the Orange (Class 1) points respectively.
- The average pay load mass appears to be 7,500 kg or less.
- CCAFS SLC 40 appears to launch the majority of the lower pay load mass launches.
- No clear apparent correlation between Pay Load Mass and Launch Site selection for successful launces.



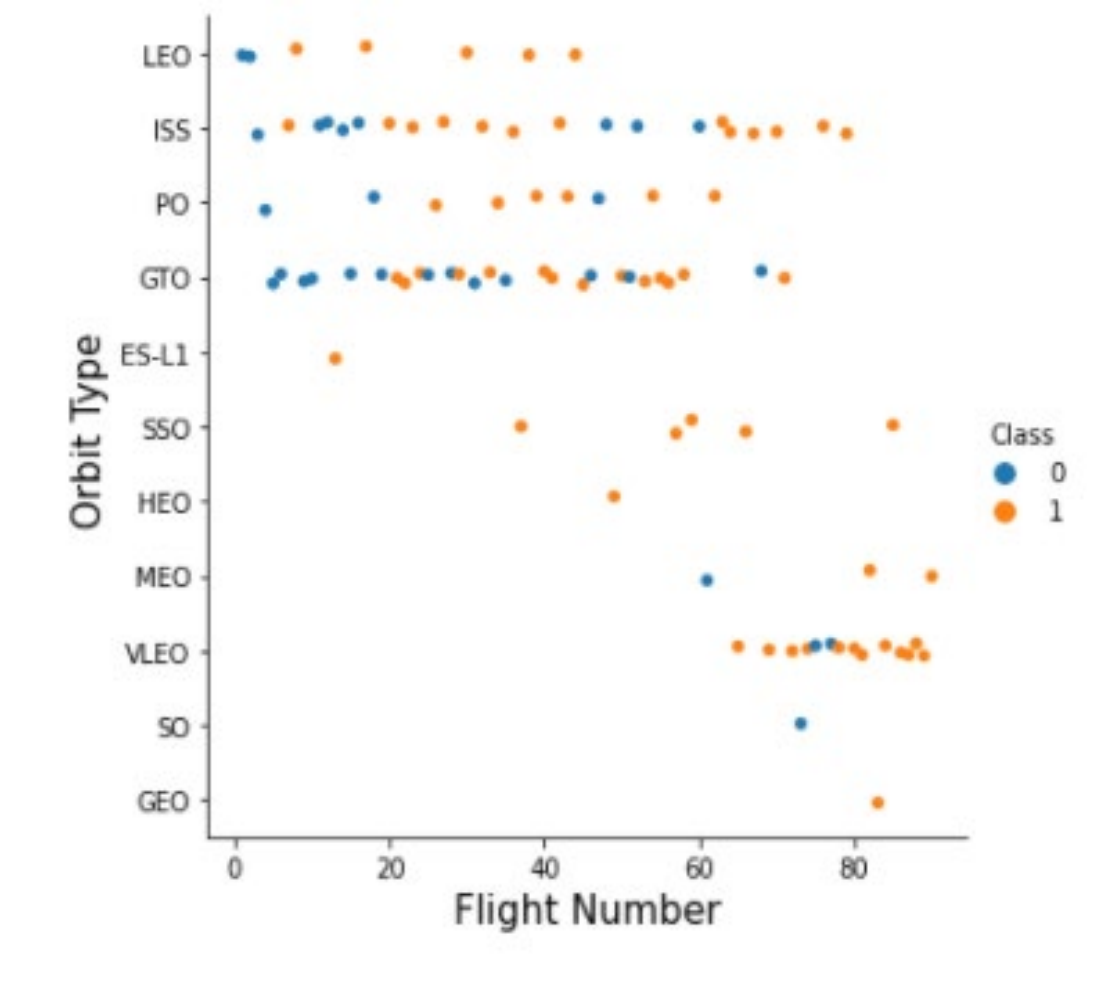
Success Rate vs. Orbit Type

- Orbit types: SSO, HEO, GEO, and ES-L1 have 100% success rates.
- Only SO had a 0% success rate.



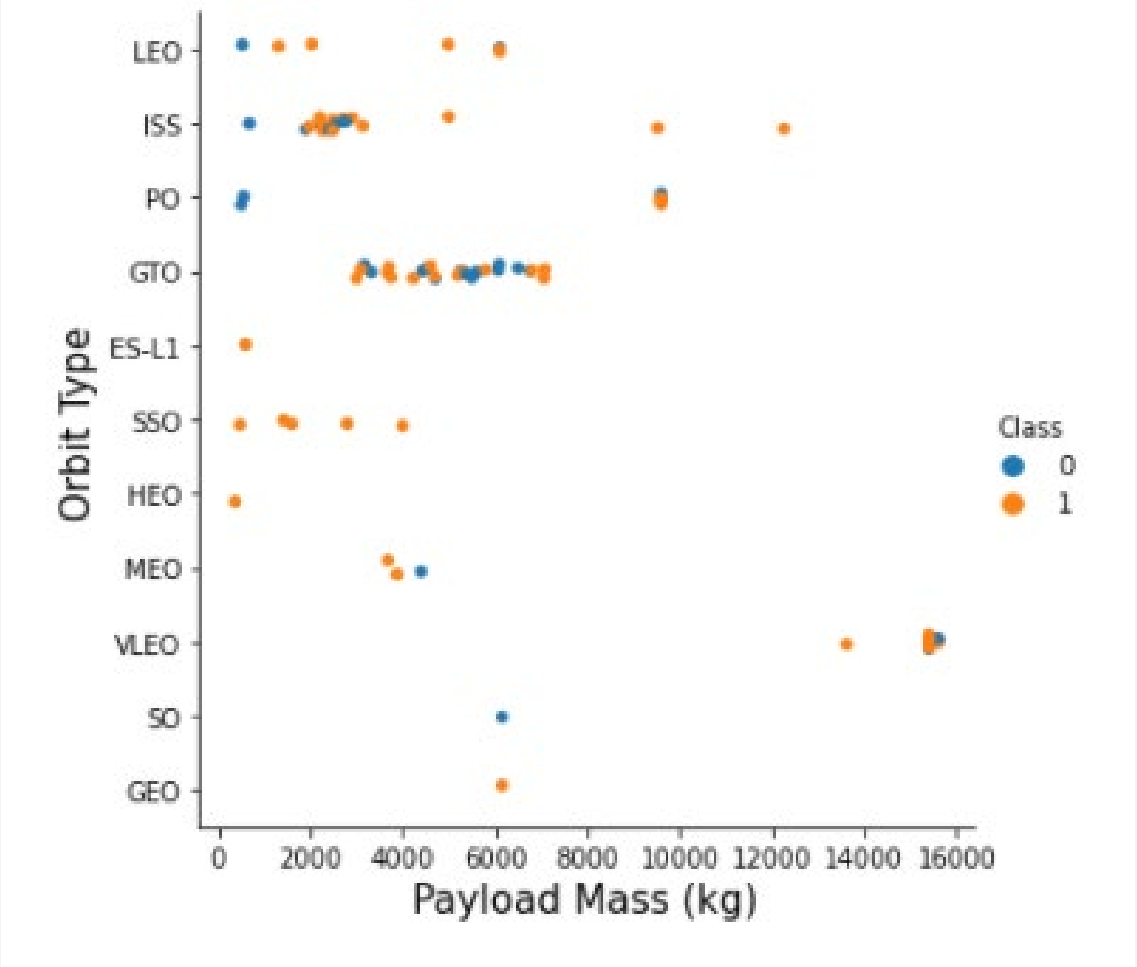
Flight Number vs. Orbit Type

- Successful and Unsuccessful launches are represented by the Blue (Class 0) and the Orange (Class 1) points respectively.
- Launch Outcome continues to appear to be correlated to an increasing number of flights, with the exception of number of flights in GTO orbit.



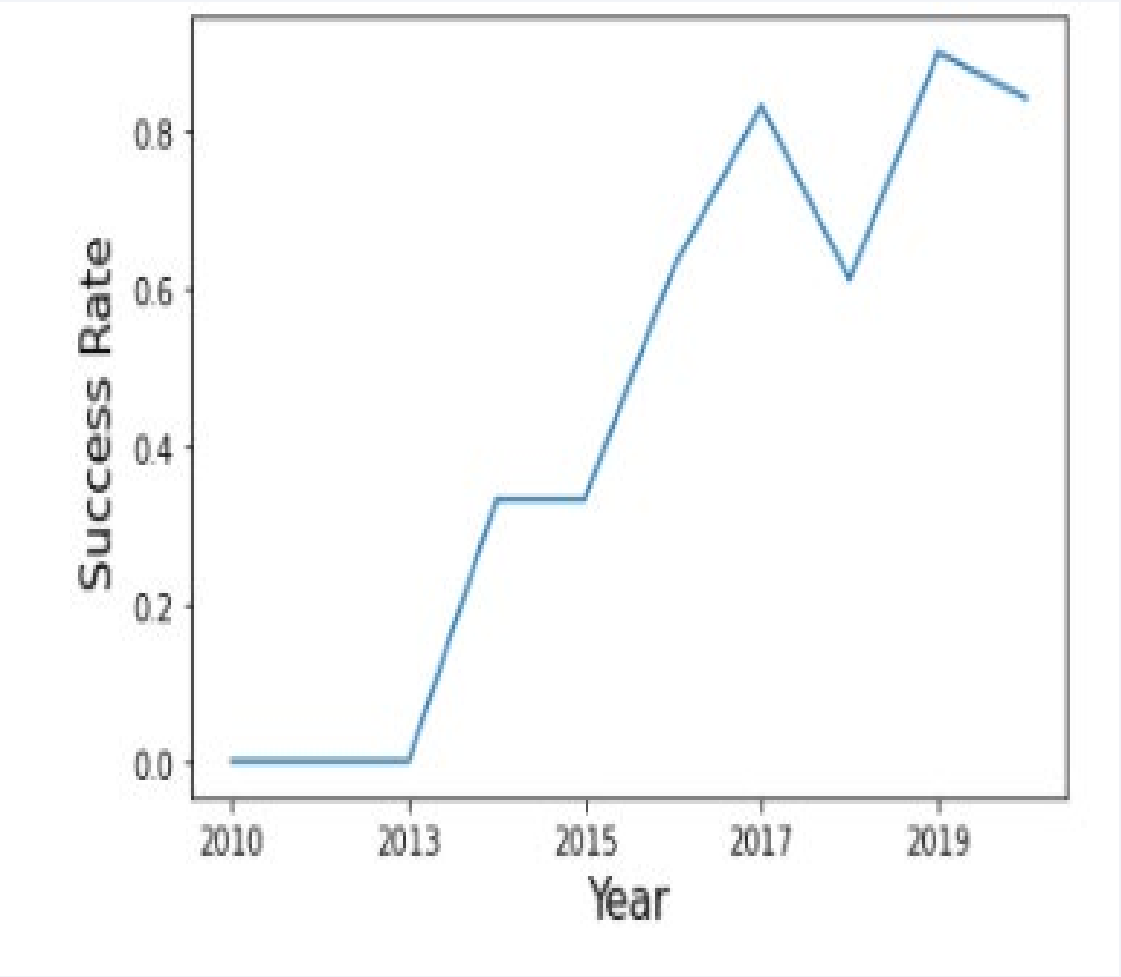
Pay Load vs. Orbit Type

- Successful and Unsuccessful launches are represented by the Blue (Class 0) and the Orange (Class 1) points respectively.
- ISS Orbits appear to have an average pay load mass around 2,000 kg, though difficult to determine whether there is any correlation with success.



Launch Success Yearly Trend

- Launch success rate has risen steadily since 2013.
- Launch success rate peaked in 2019, and is now stabilizing close to 2017 rates of around 80%.

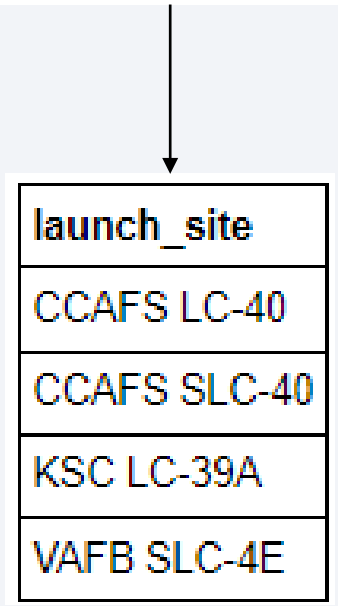


All Launch Site Names

- DISTINCT query displays only unique values

- Four Unique Launch Sites:
  - CCAFS LC-40
  - CCAFS SLC-40
  - KSC LC-39A
  - VAFB SLC-4E

```
%%sql
SELECT DISTINCT LAUNCH_SITE
FROM SPACEXTBL
```



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

Launch Site Names Begin with 'CCA'

- LIMIT 5 allows for only 5 entries beginning with `CCA` to be displayed.

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

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

Total Payload Mass

- Calculated total pay load mass for NASA with SUM() and WHERE in query.

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

total_payload_mass_kg
45596

Average Pay Load Mass by F9 v1.1

- In calculating the average pay load mass carried by booster version F9 v1.1, we note it is one of the lower average pay load masses.

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

avg_payload_mass_kg
2928

First Successful Ground Landing Date

- Notes the first success ground pad landing was not until the end of 2015.

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

first_successful_landing_date
2015-12-22

Successful Drone Ship Landing with Pay Load between 4000 and 6000

- The query returns four booster versions with successful drone ship landings.

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

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

Total Number of Successful and Failure Mission Outcomes

- This query returns a 99% mission success rate.
- Also, one launch had an unclear pay load status.

```
%%sql
SELECT MISSION_OUTCOME, COUNT(*) AS total_number
FROM SPACEXTBL
GROUP BY MISSION_OUTCOME
```

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

Boosters Carried Maximum Pay Load

- According to the result, version F9 B5 B10xx.x boosters could carried the maximum pay load, 15,600 kg.

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

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

2015 Launch Records

- This query returns that there were two drone ship landing failures in 2015

```
%%sql
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

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

- The query returns that the number of successful and unsuccessful landing outcomes between 2010-06-04 and 2017-03-20 were similar.
- Additionally, failures/successes on drone ships were exactly the same.

```
%%sql
SELECT LANDING__OUTCOME, COUNT(LANDING__OUTCOME) AS total_number
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING__OUTCOME
ORDER BY total_number DESC
```

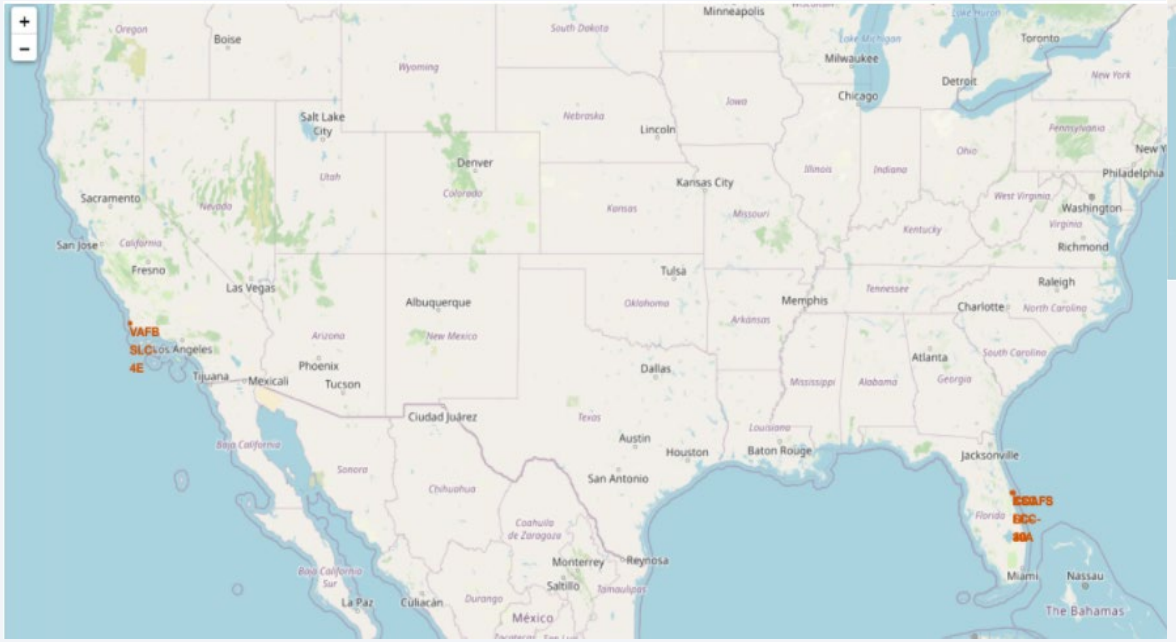
landing__outcome	total_number
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 4

Launch Sites Proximities Analysis

All Launch Sites Locations

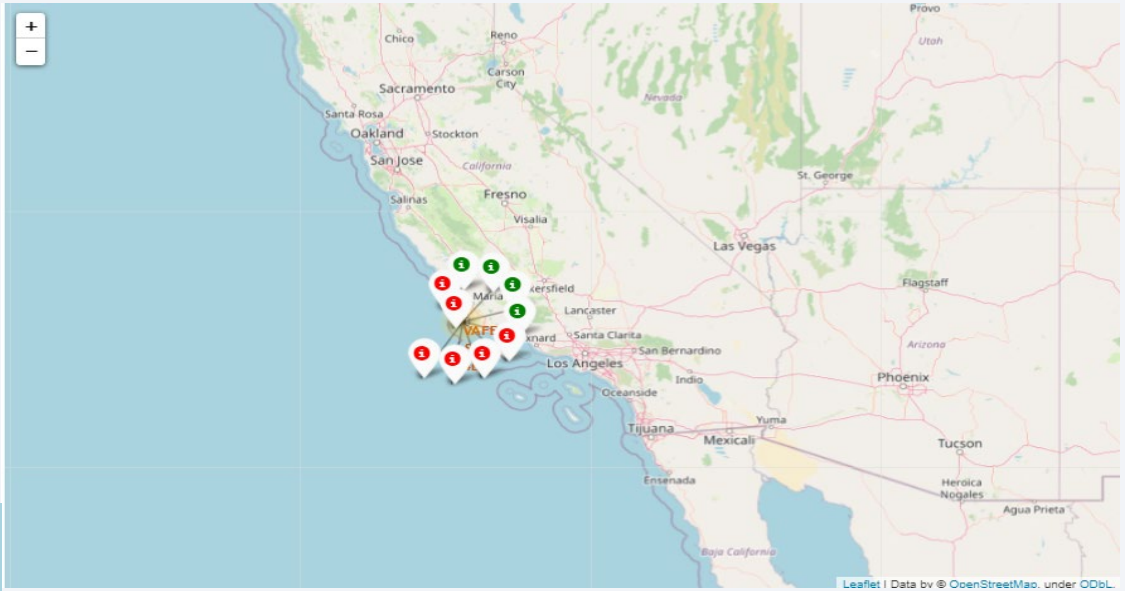
- The two maps together help indicate that all launch site locations are within the United States.

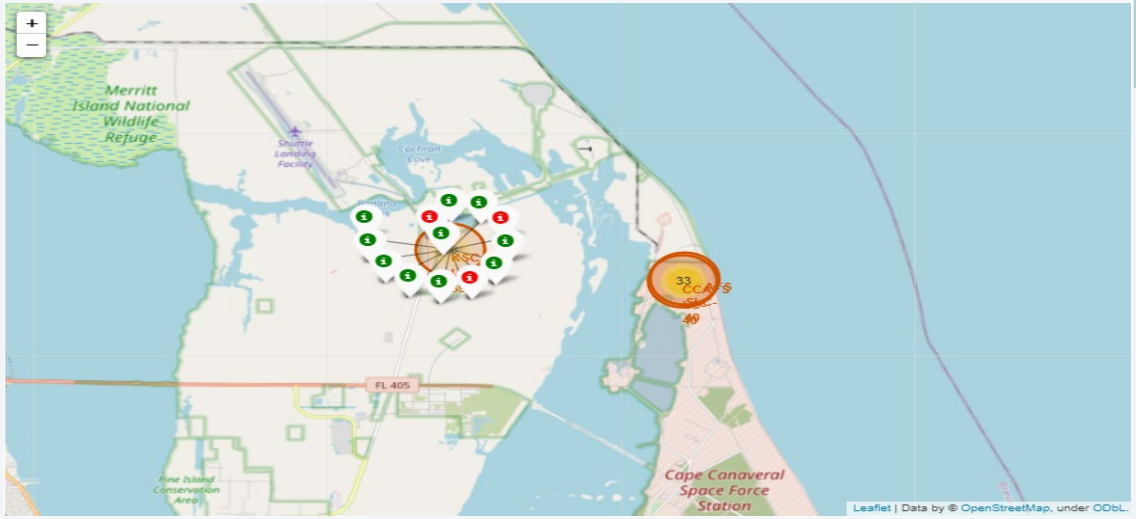


- Additionally, the launch sites are all located near the coast.

Color-Coded Launch Outcome Markers

- The map on the right shows a launch location in California, while the map below shows launch locations in Florida.

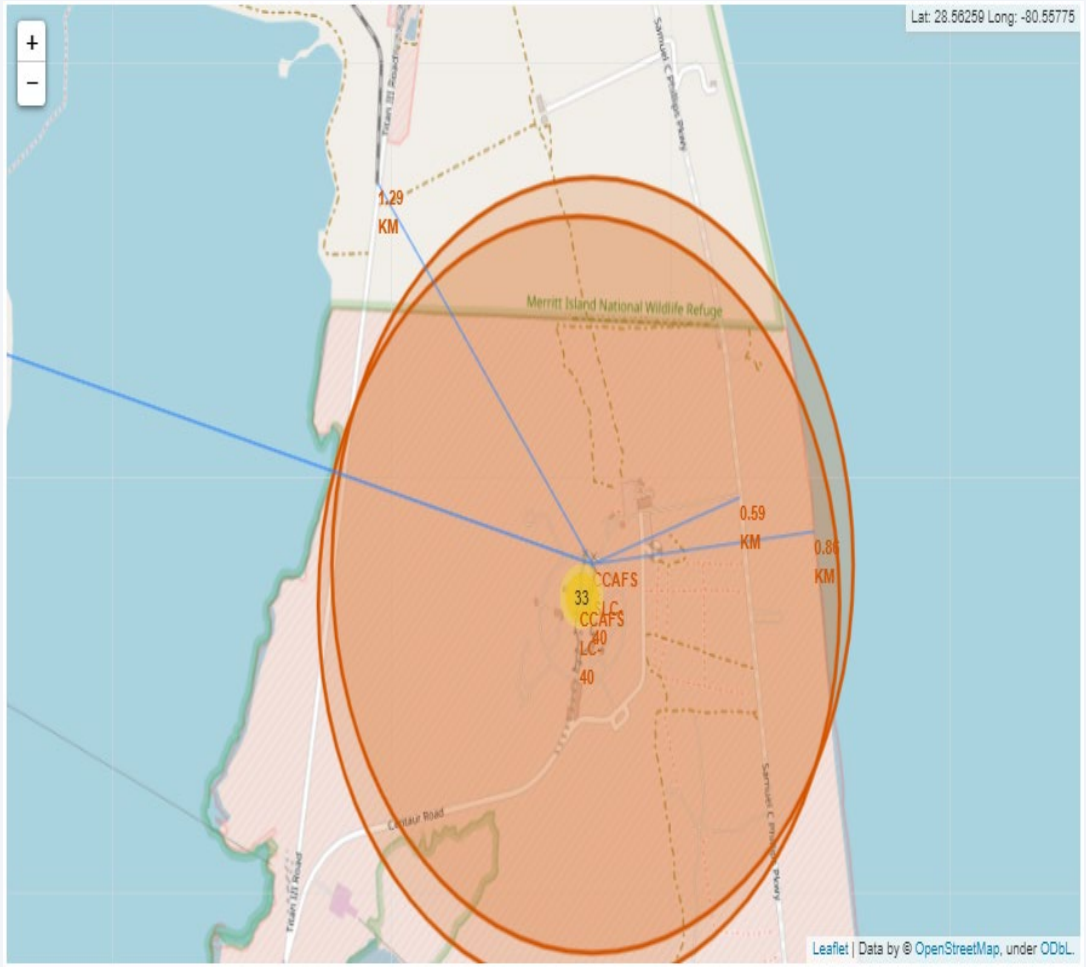


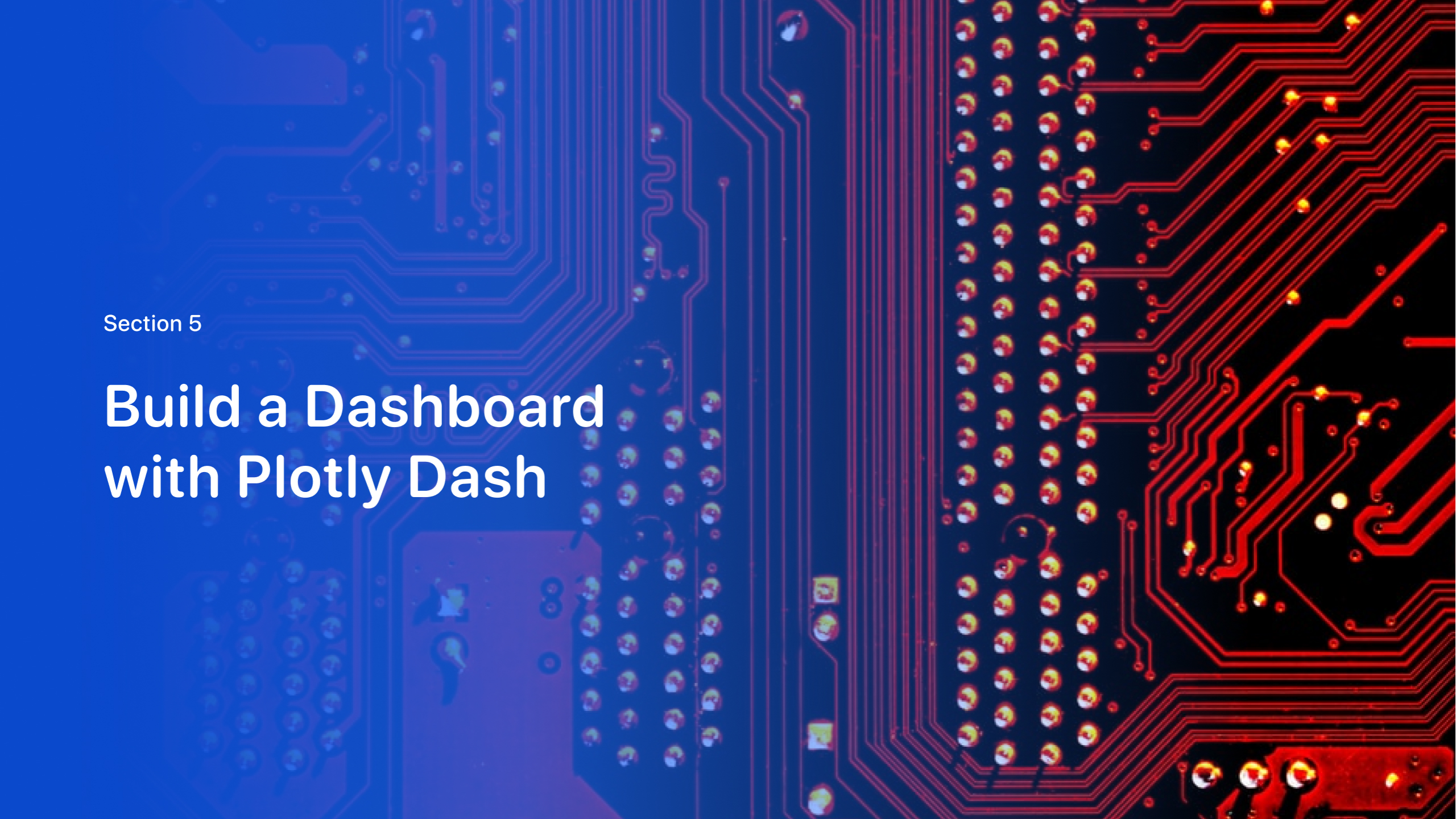


- The clusters on the Folium map can be selected to display the launch outcome, successful (green) or unsuccessful (red).

Launch Site Proximities

- As demonstrated in the map on the right, using Launch Site CCAFS SLC-40 as an example, the proximities to railways, highways, coastlines, and cities can be calculated.
- In investigating the launch sites, it is noted that they are close to railways and highways for supply chain and transportation purposes. In addition, the sites are closer to coastlines compared to cities in order to reduce the risk of failed launches posing threats to people.



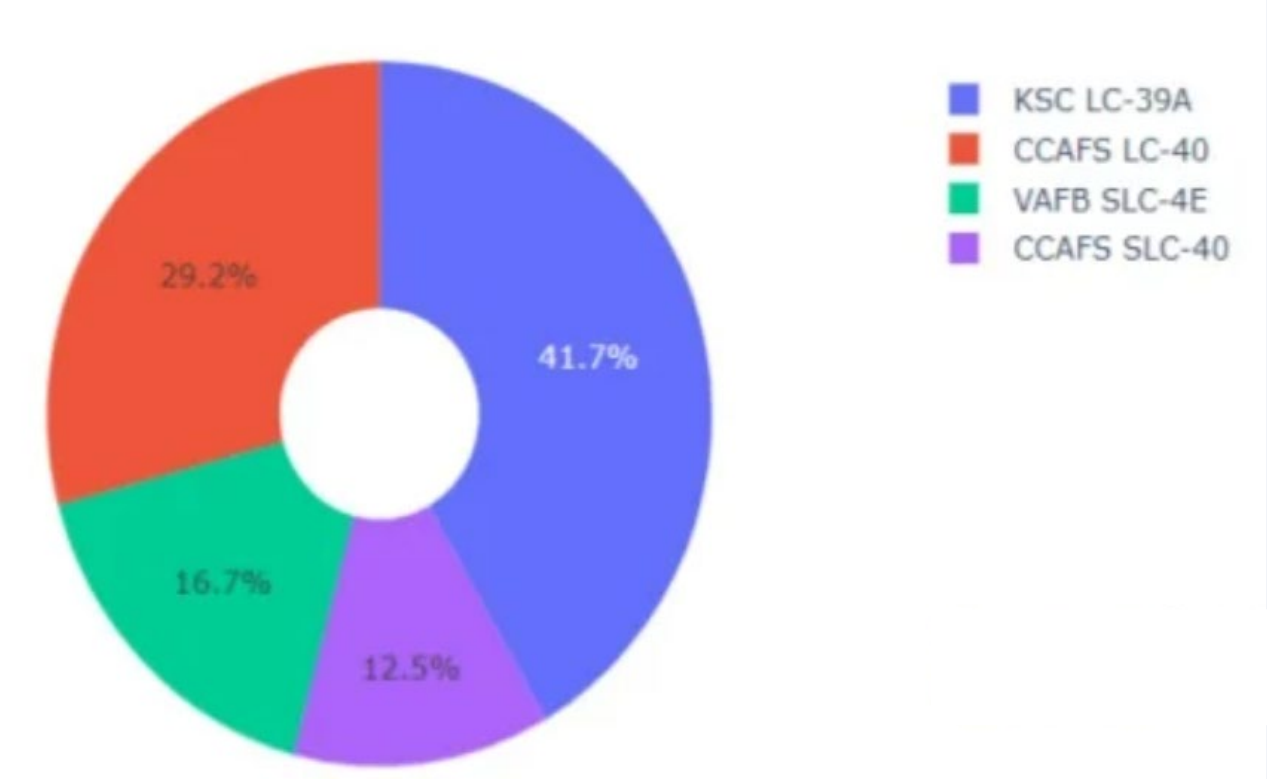


Section 5

Build a Dashboard with Plotly Dash

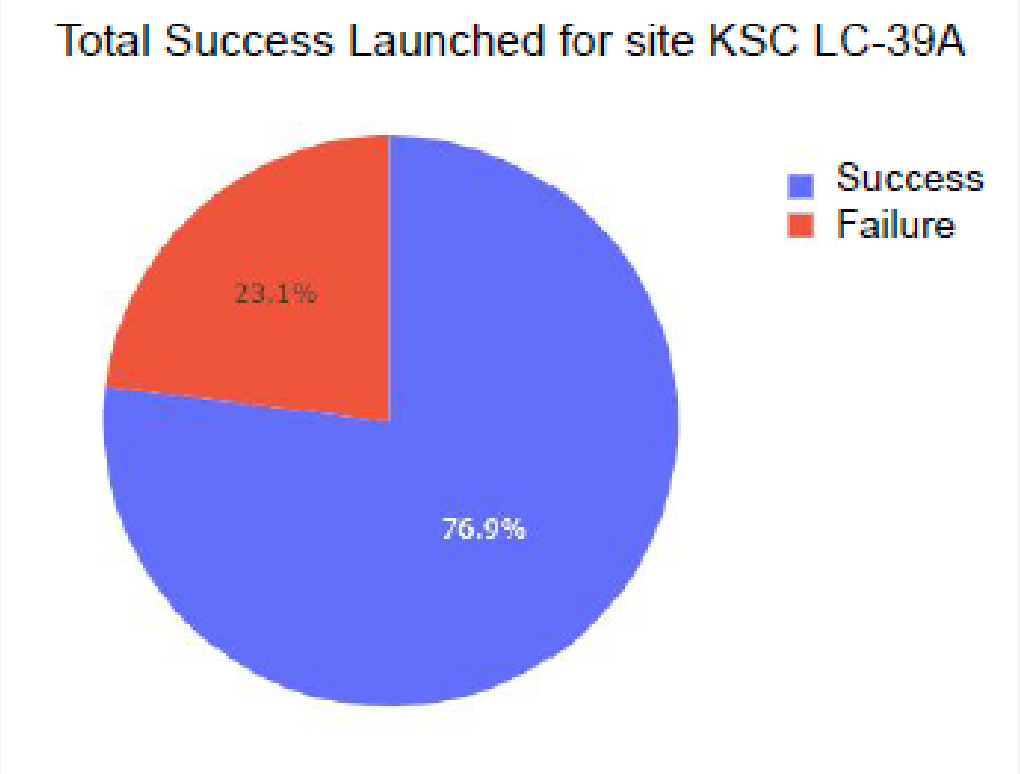
Total Successful Launches by Launch Site

- KSC LC-39A had the most successful launches of all launch sites.



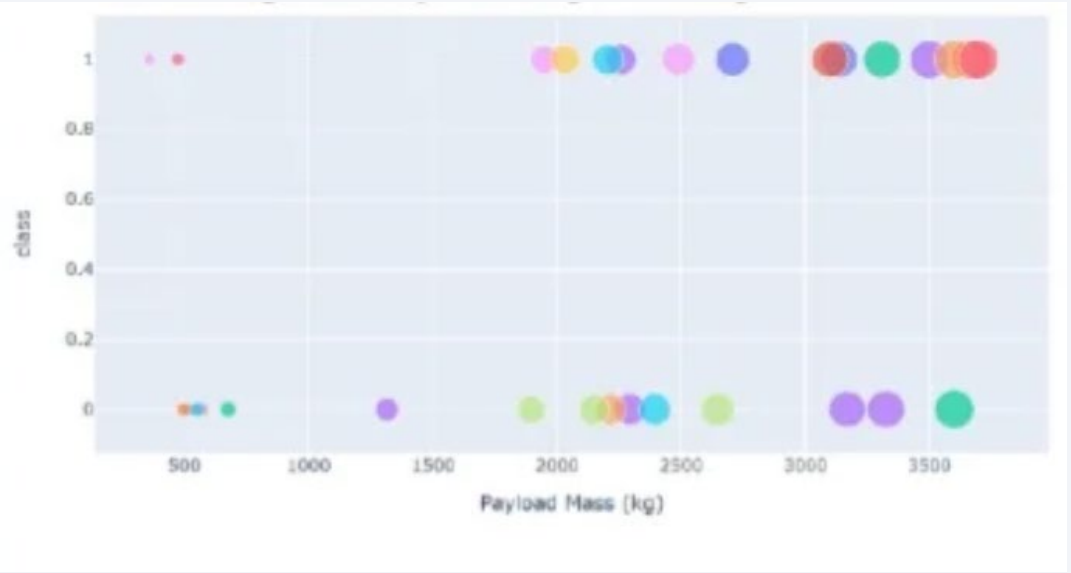
Highest Launch Success Ratio Site

- KSC LC-39A had the highest success rate of 76.9% or 10 successful landings.

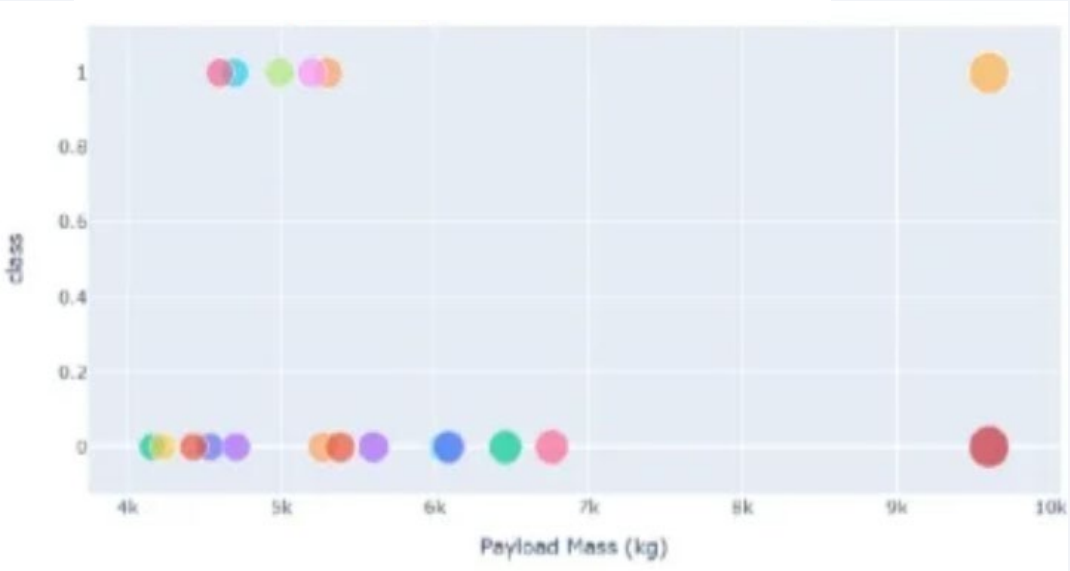


Payload vs Launch Outcome for all Launch Sites

- The success rate for lower pay load masses is higher than that of higher pay load masses.



Lower Pay Load Masses
(<4,000 kg)



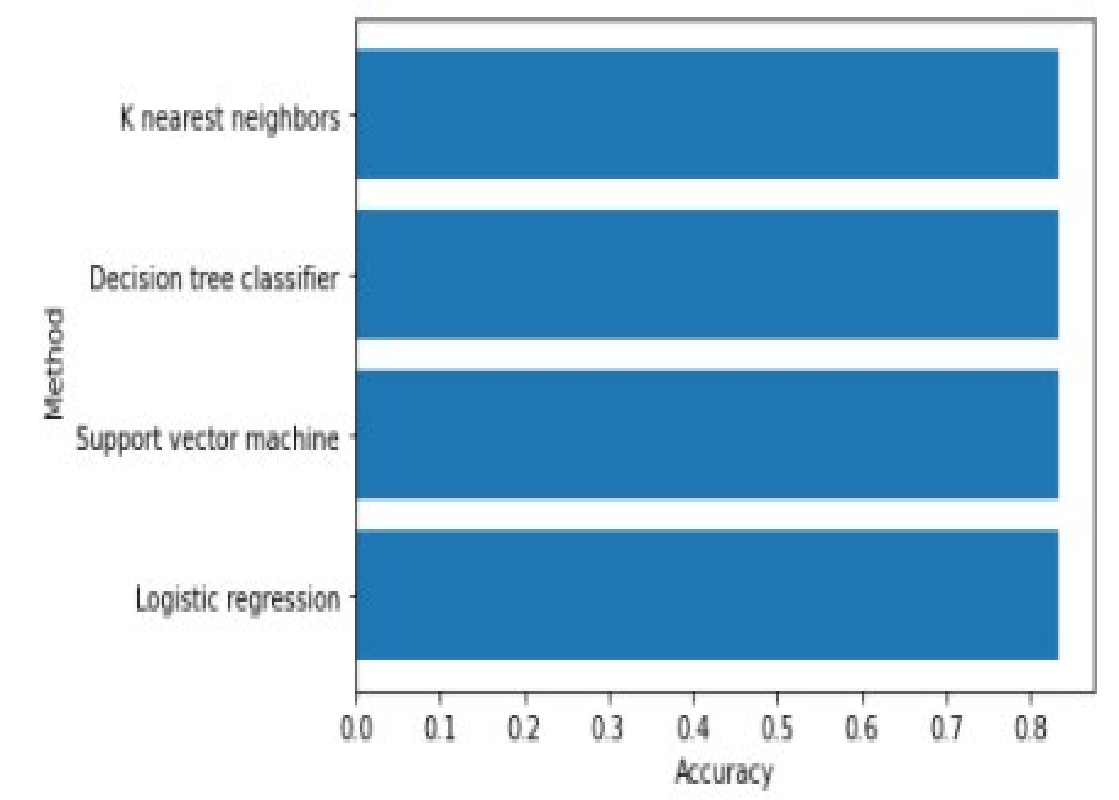
Higher Pay Load Masses
(>4,000 kg)

Section 6

Predictive Analysis (Classification)

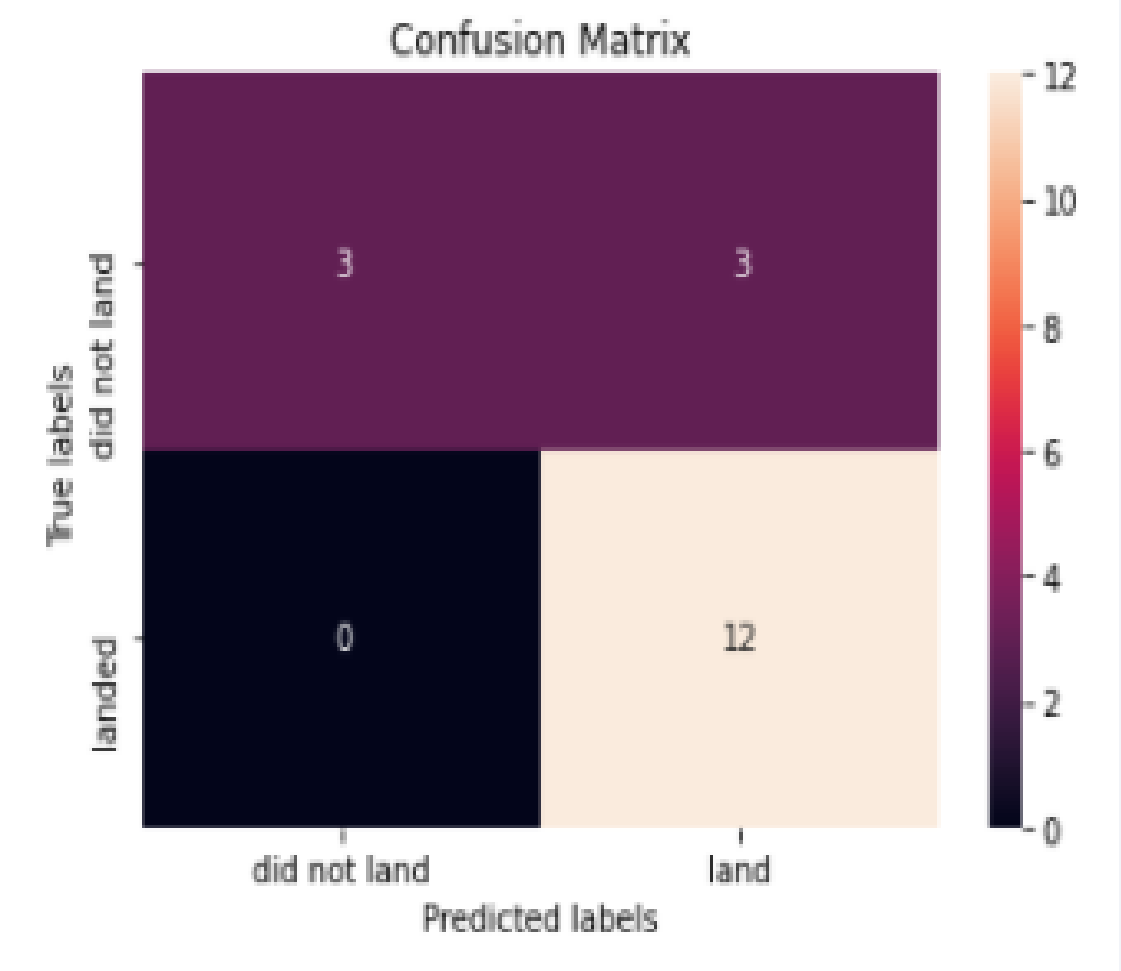
Classification Accuracy

- With a small test set, 18 samples, it is likely that more data is needed to determine the best model. After analysis, the accuracy of all models were equal, at 83.33%.



Confusion Matrix

- As all models had the same accuracy, the confusion matrix was the same for all models.
- While the models predicted successful landings, they also predicted successful landings when it did not land indicating three false positives.



Conclusions

- The success rate of launches increased over time, recently eclipsing 80%. Though orbital types: SSO, HEO, GEO, and ES-L1 had a 100% success rate.

- KSLC-39A had the highest number of launch successes and the highest success rate among all sites. However, all launch sites are close to railways, highways, and coastline, but farther from cities.

- The launch success rate of lower pay load masses is higher than that of higher pay load masses.

- All models produced the same accuracy, 83.33%, but in order to ensure the best machine learning model is truly used, more data would be required.

Appendix

- Github URL

Thank you!