Winning Space Race with Data Science

Jordan Joel Urias Paramo
2022/05/15

Outline

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

Introduction

- **Project background and context**

The commercial space age is here, companies are making space travel affordable for everyone. Virgin Galactic is providing suborbital spaceflights. Rocket Lab is a small satellite provider. Blue Origin manufactures sub-orbital and orbital reusable rockets.

Perhaps the most successful is SpaceX. SpaceX's accomplishments include: Sending spacecraft to the International Space Station. Starlink, a satellite internet constellation providing satellite Internet access. Sending manned missions to Space.

One reason SpaceX can do this is the 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.

- **Problems you want to find answers**
  - Conditions to get the best results and ensure the best successful landing rate
  - Correlations between each rocket variables and successful landing rate

Executive Summary

- We used the SpaceX and some web scraping to get information, that later was explored using Db2 for storing and pandas for manipulation, resulting in several charts giving us insight about a classification label. Later that insight was reinforced by using folium to further understand the nature of said success and give extra context about the operation.
- As a product we have an interactive dashboard and a fine tune model, using AutoML, to answer the question from the introduction.

Summary

- 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
- Summary of all results
  - Exploratory data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

Section 1

Methodology

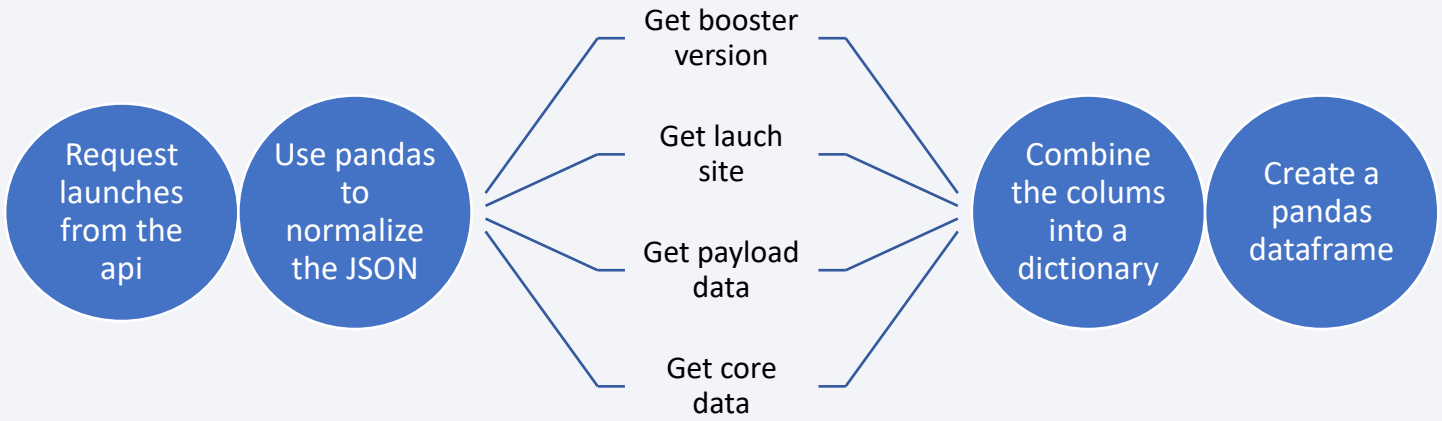
Methodology

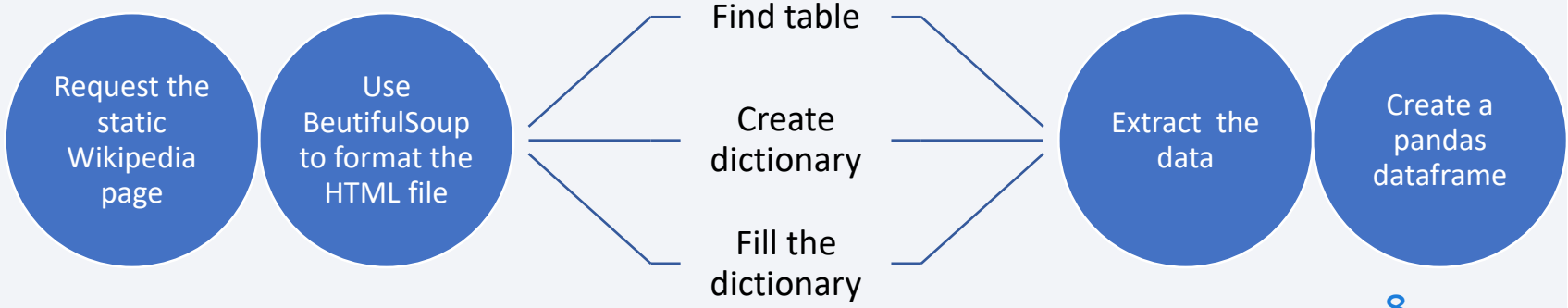
Executive Summary

- Data collection methodology:
  - SpaceX API & Web Scraping the Falcon 9 Wikipedia page
- Perform data wrangling
  - Convert outcomes into Training Labels with the booster successfully/unsuccessful 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
  - Use an autoML method (GridSearch), to find the best model and its hyperparameters

Data Collection

- The data collection process includes a combination of API requests from the SpaceX API and web scraping data from a table in the Wikipedia page of SpaceX, Falcon 9 and Falcon Heavy Launches Records





Data Collection – SpaceX API

1. Call the spaceX API

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

response = requests.get(spacex_url)
```

2. Normalize the JSON response

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

3. Get the ids details

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

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

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

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

4. Combine the colums in a dictionary

```
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}
```

5. Create a pandas dataframe

```
# Create a data from launch_dict
launch_data = pd.DataFrame.from_dict(launch_dict)
```


GitHub

Data Collection - Scraping

1. Request the static Wikipedia page

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

2. Format the html file

```
soup = BeautifulSoup(html_data, 'html5lib')
```

3. Find the table

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

4. Extract the columns

```
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if(name != None and len(name) > 0):
        column_names.append(name)
```

5. Create a dictionary using the columns

```
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
```

6. Fill the dictionary

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.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
        if flag:
            extracted_row += 1
            # Flight Number value
            # TODO: Append the flight_number into launch_dict with key `Flight No.`
            launch_dict['Flight No.'].append(flight_number)
            print(flight_number)
            datatimelist=date_time(row[0])
```

6. Create a pandas dataframe

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

[To see the full code visit this GitHub notebook]

Data Wrangling

For the data wrangling notebook, we explored 3 **features** and determinedthe **Training label**

- Features:
  - Launch Sites
  - Orbit
  - Outcome

```
# Apply value_counts() on column LaunchSite
df.LaunchSite.value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
```

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

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
```

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

```
True ASDS       41
None None       19
True RTLS       14
False ASDS       6
True Ocean       5
False Ocean      2
None ASDS        2
False RTLS       1
```

- Training label
  - Once we obtained the landing_outcomes feature, is easy to see the dichotomy of the result, True or False.

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

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

  - Therefore, the training label is created following: `1` = successful / `0` = failure, and appended to the dataframe

```
landing_class = [0 if outcome in bad_outcomes else 1 for outcome in df.Outcome]
df['Class']=landing_class
```

  - Finally, we determined the success rate

```
df["Class"].mean()
```

```
0.6666666666666666
```

GitHub

EDA with Data Visualization

- **Bar chart:** Shows comparisons among discrete categories.
  - Orbit Type vs. Success Rate
- **Scatter plot:** Shows the relationship between two variables, and an additional one, because we used colors
  - Flight Number vs. Launch Site
  - Payload vs. Launch Site
  - Flight Number vs. Orbit Type
  - Payload vs. Orbit Type
- **Line chart:** Shows the variable trend in data over intervals of time
  - Year vs. Success Rate

GitHub

EDA with SQL

- The following SQL Queries were excuted using a connection to a Db2 database:
  - Displaying the names of the unique launch sites in the space mission
  - Displaying 5 records where launch sites begin with the string 'CCA'
  - Displaying the total payload mass carried by boosters launched by NASA (CRS)
  - Displaying average payload mass carried by booster version F9 v1.1
  - Listing the date when the first successful landing outcome in ground pad was achieved
  - Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - Listing the total number of successful and failure mission outcomes
  - Listing the names of the booster_versions which have carried the maximum payload mass
  - Listing the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - Ranking the count of landing outcomes (such as Failure (drone ship) or Success (groundpad)) between the date 2010-06-04 and 2017-03-20, in descending order

GitHub

Build an Interactive Map with Folium

- The following object were created and added to a folium map:
  - Markers that show all launch sites on a map
  - Markers that show the success/failed launches for each site on the map
  - Lines that show the distances between a launch site to its proximities
- After you plot distance lines to the proximities, we can answer the following questions easily:
  - 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 coastline? Yes
  - Do launch sites keep certain distance away from cities? Yes

GitHub

Build a Dashboard with Plotly Dash

- The dashboard application contains a pie chart and a scatter point chart.
  - **Pie chart:** For showing total success launches by sites
    - This chart is affected by a selector to indicate a successful landing distribution across all launch sites or to indicate the success rate of individual launch sites.
  - **Scatter chart:** For showing the relationship between Outcomes and Payload mass (Kg) by different boosters
    - This chart is affected by 2 inputs, the selector All sites/individual site & Payload mass on a slider between 0 and 10000 kg
    - This chart helps determine how success depends on the launch point, payload mass, and booster version categories.

GitHub

Predictive Analysis (Classification)

1. Select the target variable

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

2. Standarize the data

```
transform = preprocessing.StandardScaler()
X = transform.fit_transform(X)
```

3. Split the data into a training and test data

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

4. Create classification model an a GridSearch Model

```
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)
```

5. Display the best parameters and accuracy

```
print("tuned hpyerparameters :(best parameters) ",logreg_cv.best_params_)
print("accuracy :",logreg_cv.best_score_)
```

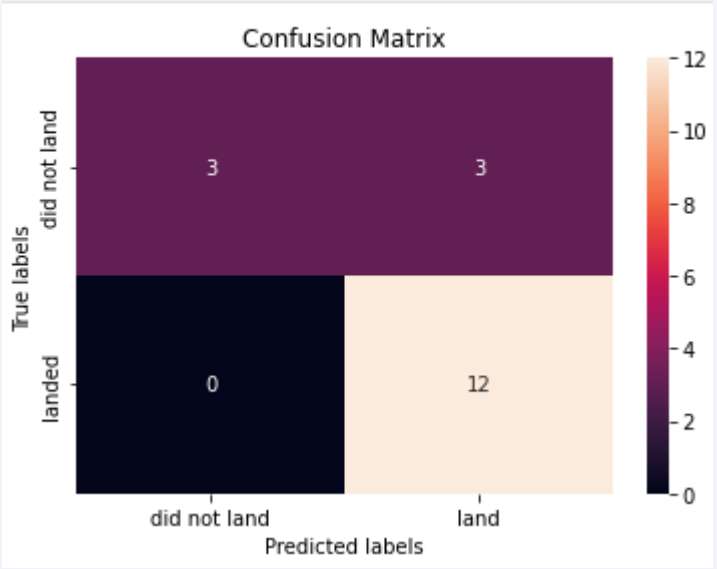
```
tuned hpyerparameters :(best parameters)  {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
accuracy : 0.8464285714285713
```

6. Calculate the accuracy using the test data

```
methods.append('Logistic regression')
accuracy.append(logreg_cv.score(X_test, Y_test))
```

7. Take a look a the confution matrix

```
yhat=logreg_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```

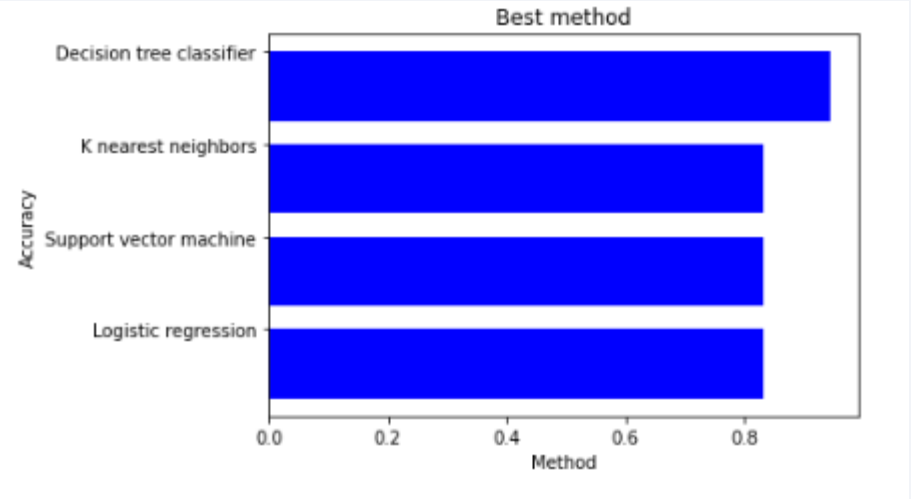


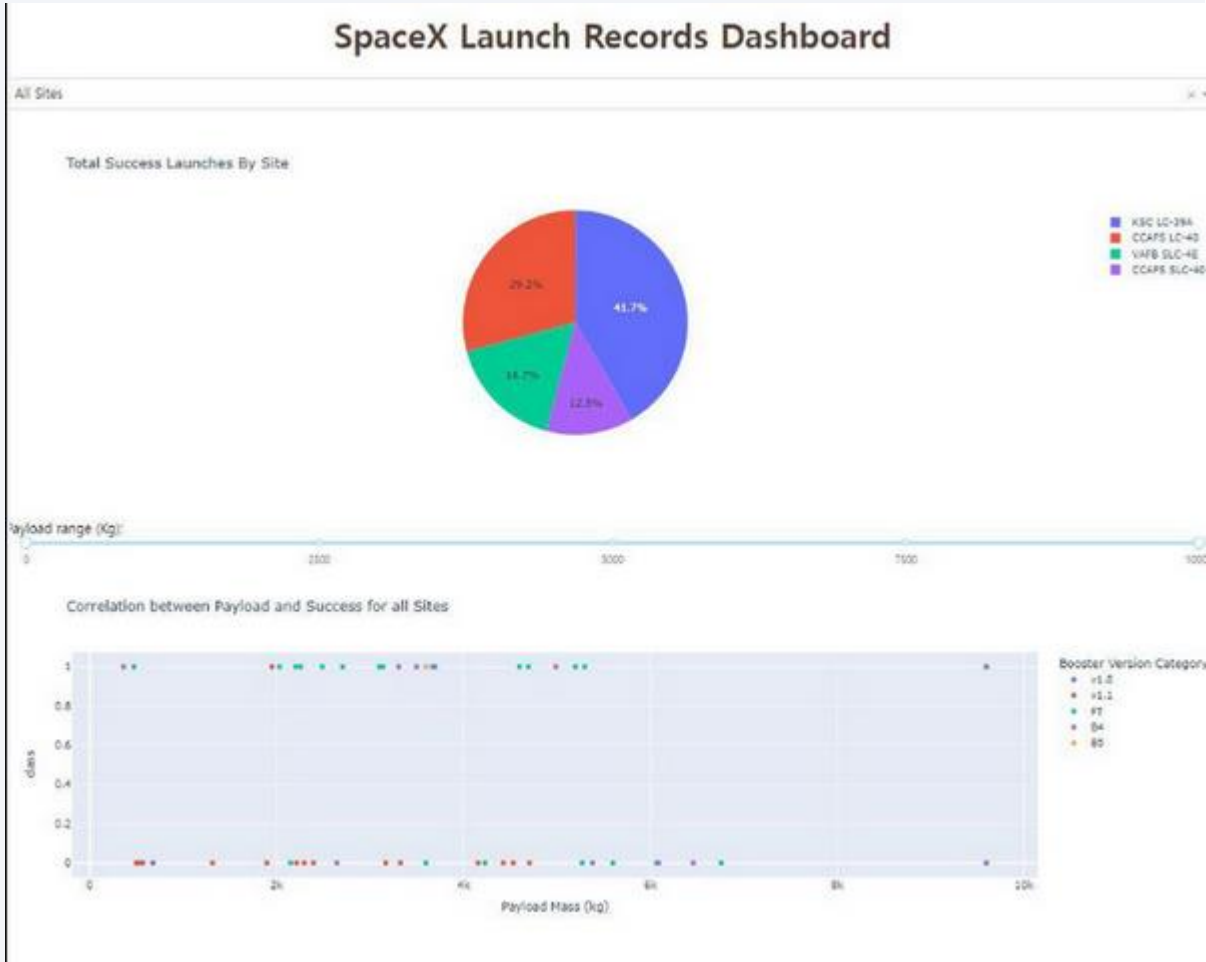
8. Repeat steps from 4 to 7, with SVM, Decision tree and KNN

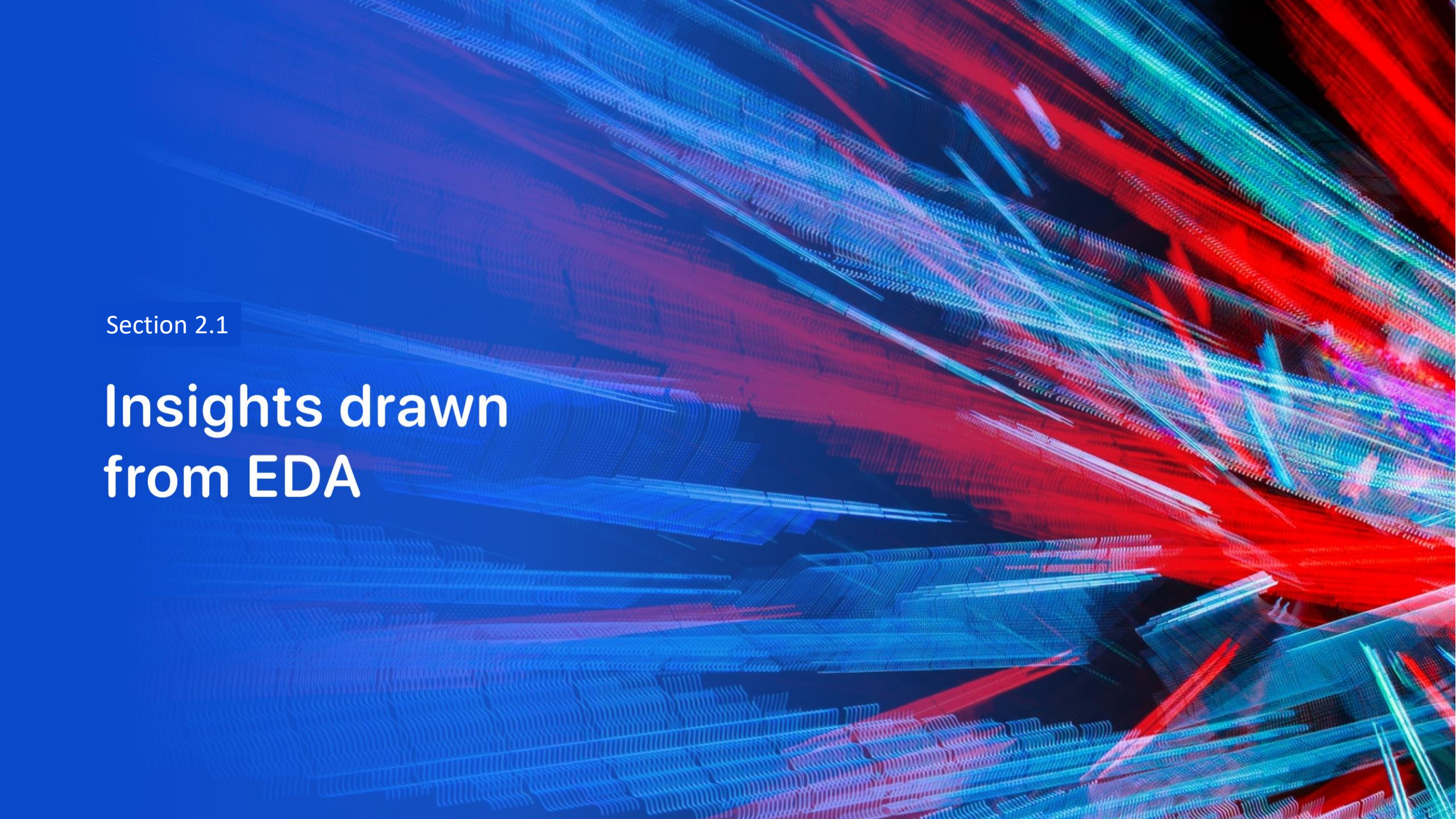
9. Finally, Select the best performer

Results

- The size of the payload directly affects the success rate of the mission.
- The orbit and launch site show correlation with the success of the mission but we need to further investigate how is related to the order of the launch (**Earliest missions were less successful**)
- The best model is the decision tree, the other 3 are tied.

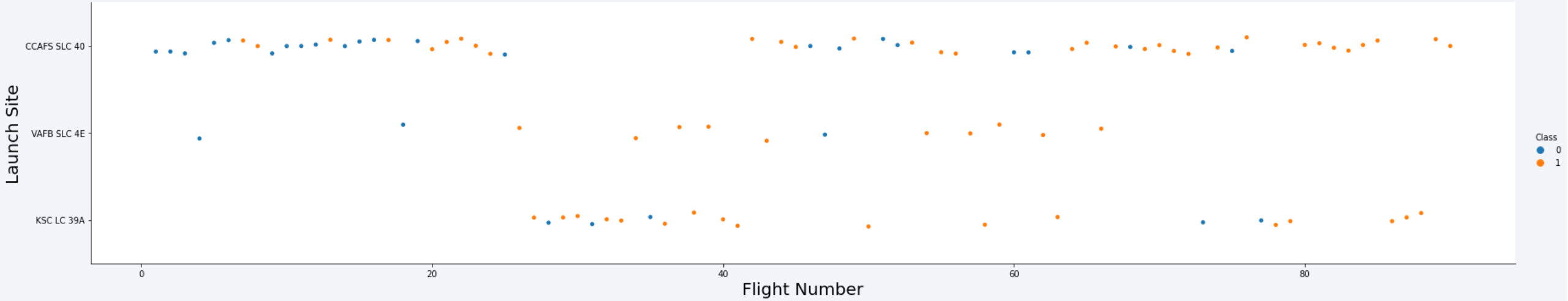





Section 2.1
Insights drawn from EDA

Flight Number vs. Launch Site

This figure shows that the success rate increased as the number of flights increased.

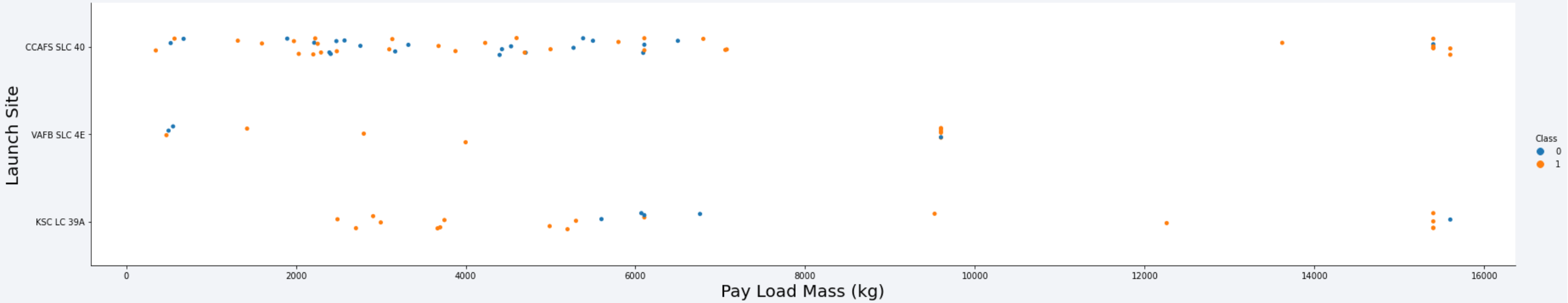


Payload vs. Launch Site

Looks like the heaviest the payload the bigger the success rate, but we need more context, maybe the ligther payload are from the earliest lauches.

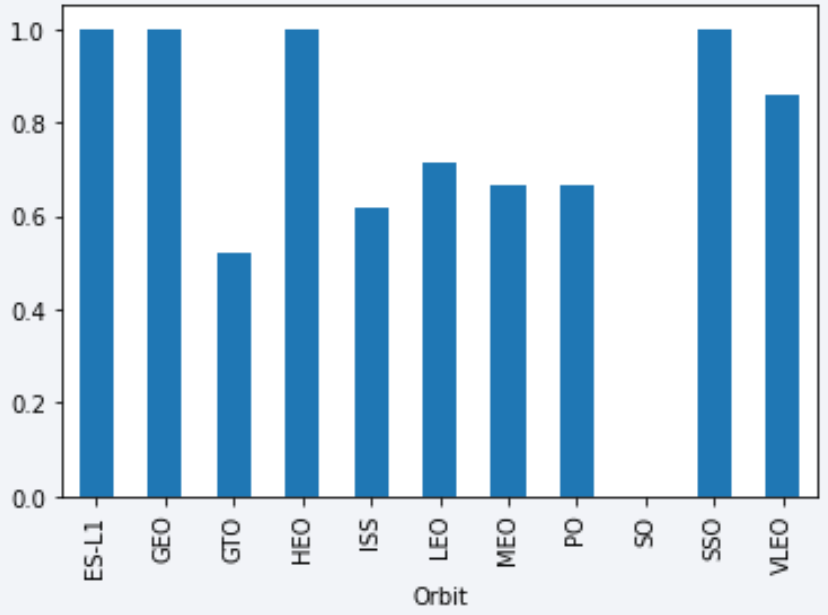
The site VAFB-SLC has not launched a payload heavier than 10,000 kg.

The site SLC 40 has the worst success rate.



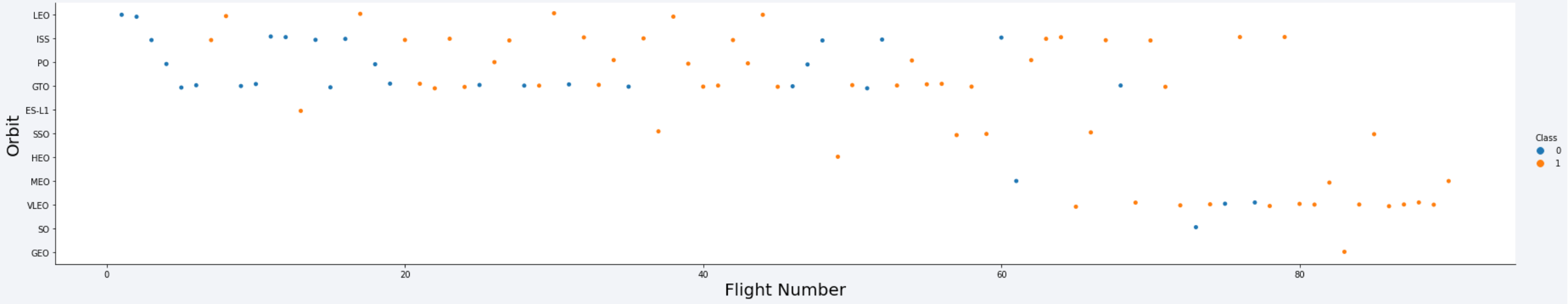
Success Rate vs. Orbit Type

- Orbit types SSO, HEO, GEO, and ES-L1 have the highest success rates (100%)
- After SO, GTO has the lowest one with 50%



Flight Number vs. Orbit Type

- 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.
- SpaceX started doing experiments in the LEO orbit, but most recently is launching payloads to the VLEO orbit.

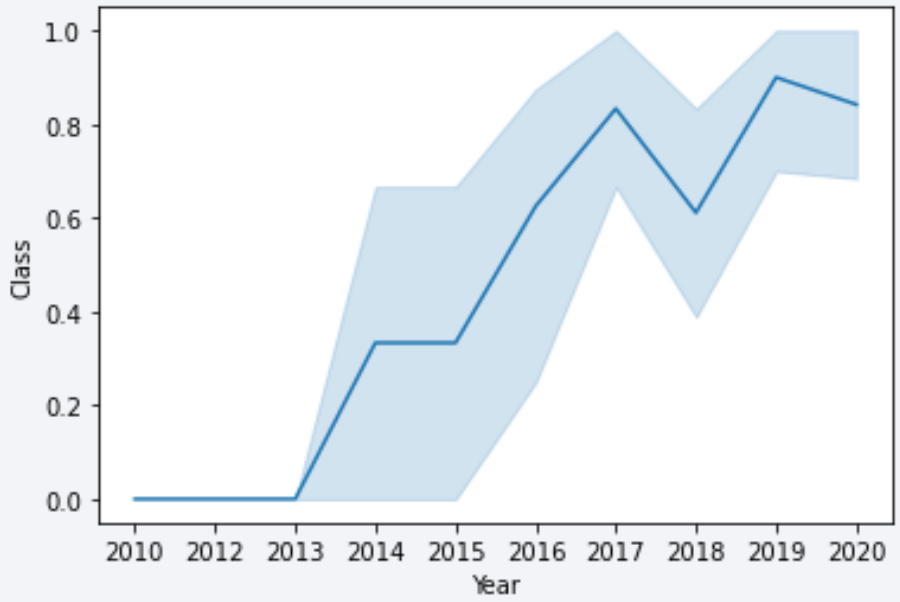


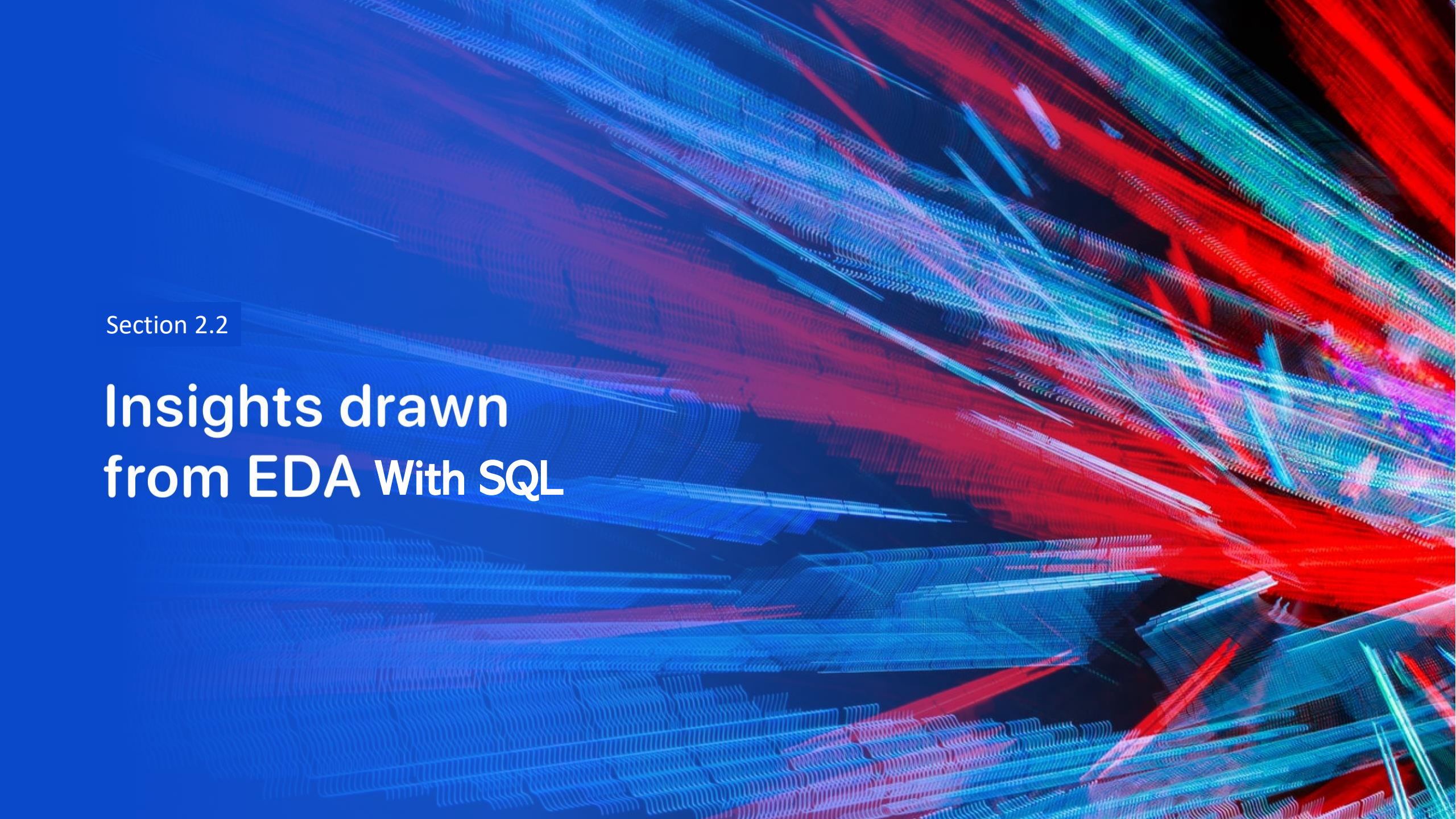
Payload vs. Orbit Type

- 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

- We can observe that the success rate since 2013 kept increasing till 2020.
- The rate decreased slightly in 2018.
- Recently, it has shown a success rate of about 80%




Section 2.2
Insights drawn from EDA With SQL

All Launch Site Names

- Query

```
SELECT DISTINCT LAUNCH_SITE
FROM SPACEXTBL
```

- Result

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

Explanation:

There is four unique launch sites

Launch Site Names Begin with 'CCA'

- Query

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

- Result

Explanation:
5 examples of missions that where launched from CCA

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Query

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

- Result

total_payload_mass_kg
45596

Explantation:
Total weigth launches for the NASA

Average Payload Mass by F9 v1.1

- Query

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

- Result

avg_payload_mass_kg
2928

Explanation:
The average payload mass of the F9 V1.1 is 2928 Kg

First Successful Ground Landing Date

- Query

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

- Result

first_successful_landing_date
2015-12-22

- Explanation
  - The 12 of December of 2015 was the date of the first success

Successful Drone Ship Landing with Payload between 4000 and 6000

- Query

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

- Result

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

- Explanation
  - The fourth booster version that have been successful at landing in that range of payload

Total Number of Successful and Failure Mission Outcomes

- Query

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

- Result

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

- Explanation
  - SpaceX has a 99% success rate

Boosters Carried Maximum Payload

- Query

```
SELECT DISTINCT(BOOSTER_VERSION)
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

- Result
- Explanation
  - Booster versions, that have carried the maximum payload mass

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

2015 Launch Records

- Query

```
SELECT  BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND YEAR(DATE) = '2015'
```

- Result

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

- Explanation
  - There was 2 launches in 2015

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

- Query

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

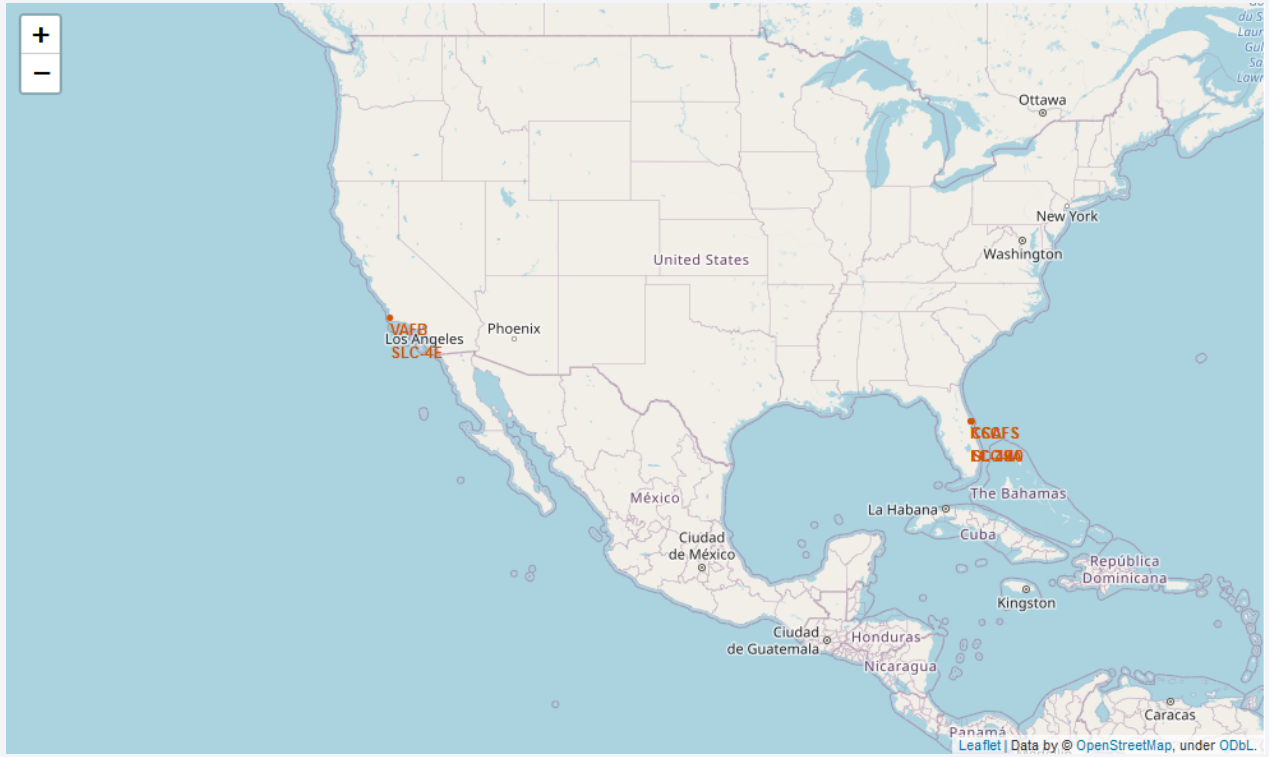
- Result
- Explanation
  - According to the results, the number of successes and failures between 2010-06-04 and 2017-03-20 was similar

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 3

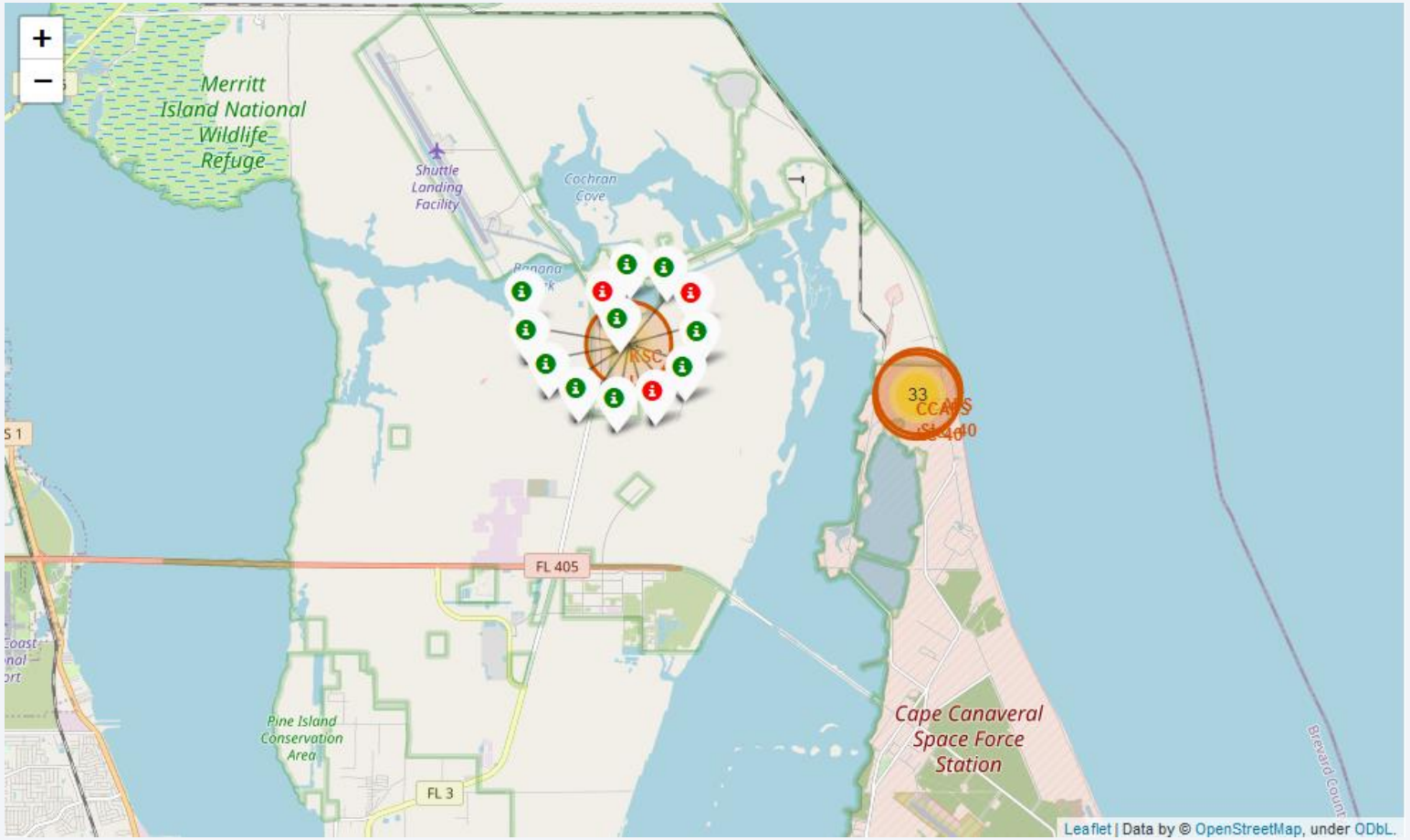
Launch Sites Proximities Analysis

Launch sites locations



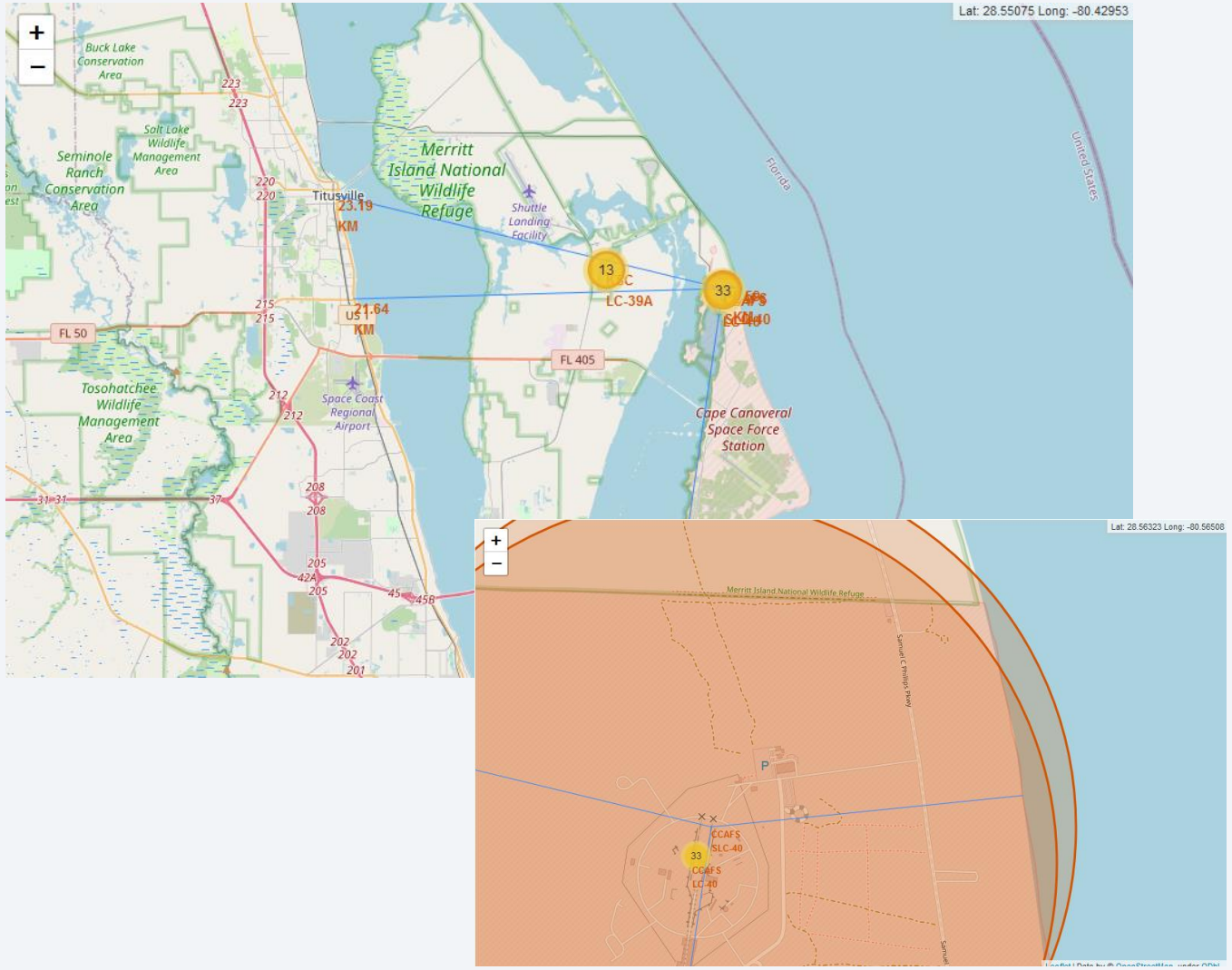
- All the launch site are in the United States
- All the sites are near the coast

Launch site success

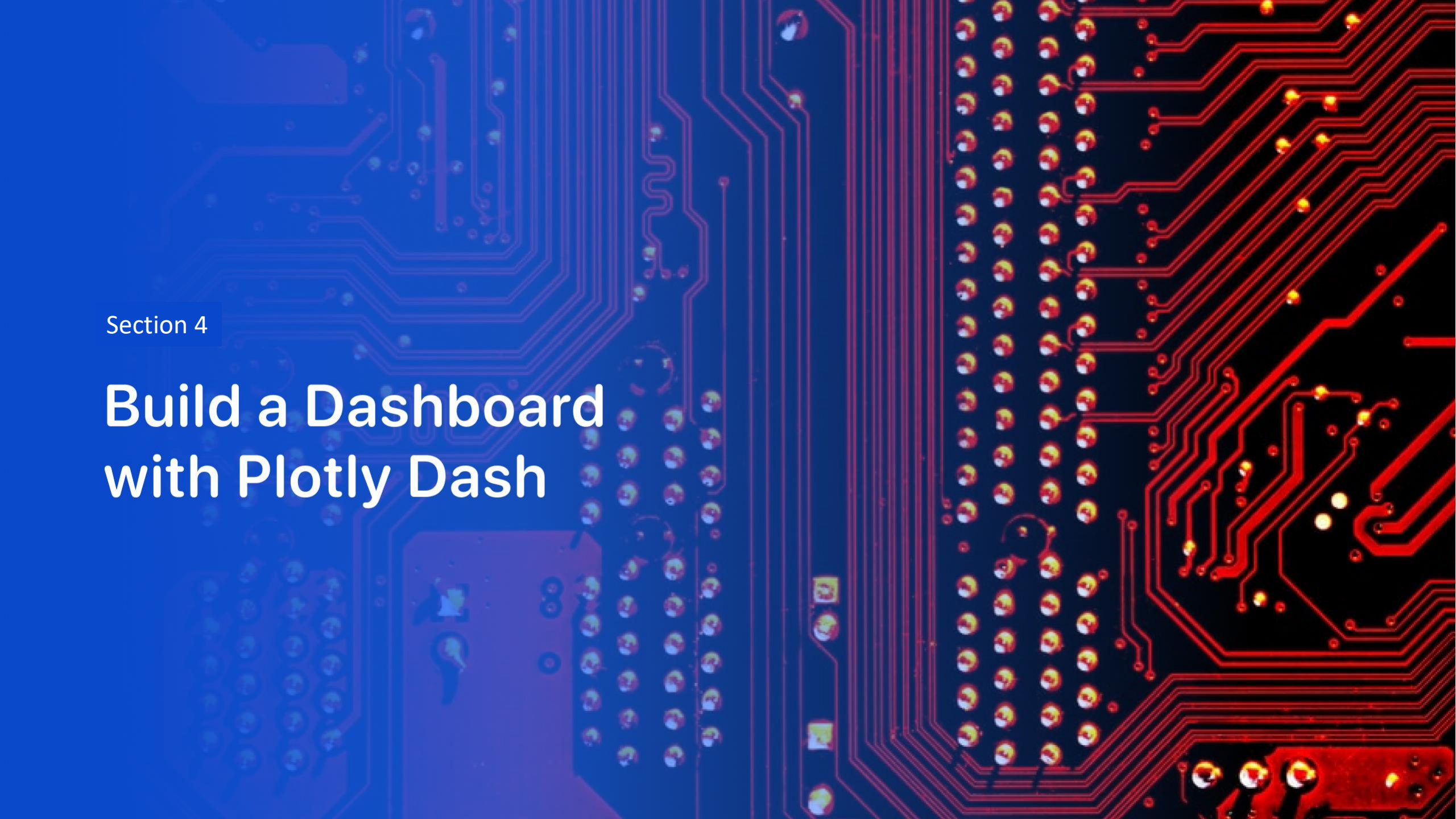


- By clicking in a site, we can see the number launches and their outcome
- For example, in the image we can observe that this launch site has a good success rate

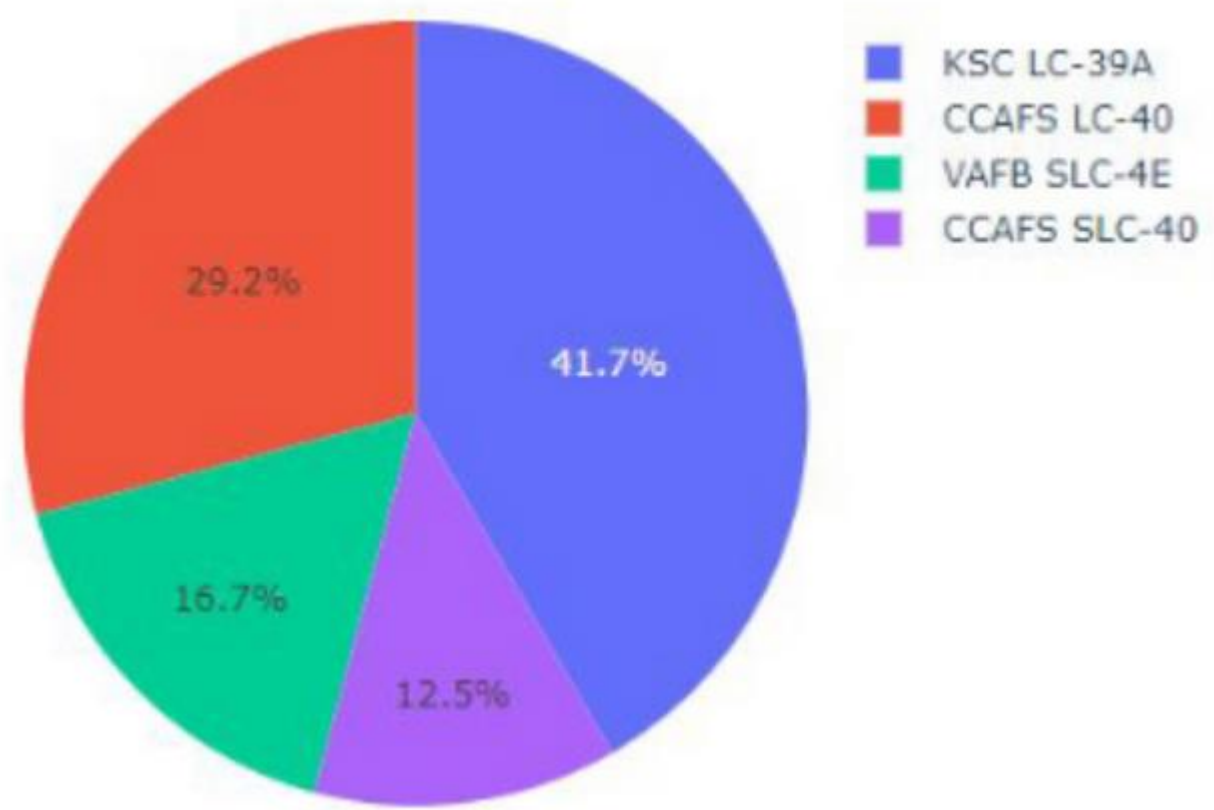
Launch site proximities



- The site are close to the coast and far way from the cities.
- We can see that there is railroad near the launch site for transporting the payload and personal

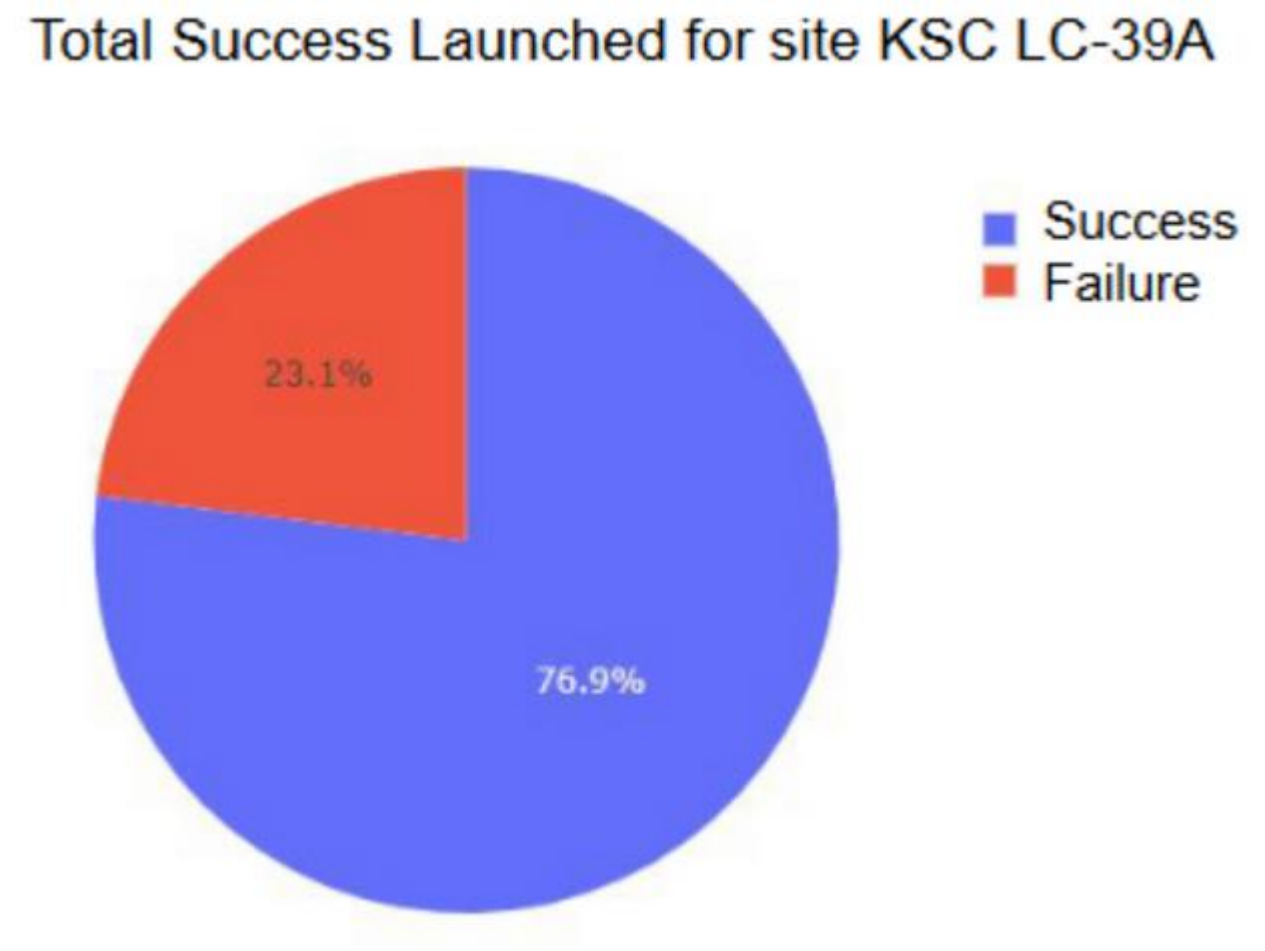

Section 4
Build a Dashboard with Plotly Dash

Launch site success



- The VAFB SLC-4E has the fewest launch success, at the same time has the fewer launches and it's the only one in the west coast
- KSLC-39A records the most launch success among all sites.

KSLC-39A Detail



- KSLC-39A has the highest success rate with 76.9%) landing successes and 23.1%. landing failures.
- Analyzing the number of flights this was the last site to be added to the roster.

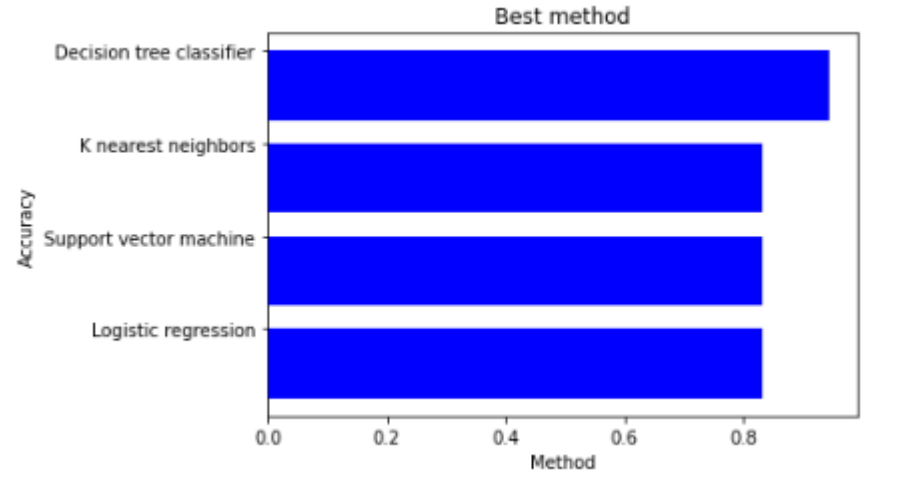
Payload mass success

- The success rate by booster version seems to be correlated to the size of the payload, the heavier the payload the lower the success rate.

Section 5

Predictive Analysis (Classification)

Classification Accuracy



- In the test set, the accuracy of 3 models was virtually the same at 83.33%, except for the decision tree with a 94.44%.
- It should be noted that the test size was small at 18.
- Therefore, more data is needed to determine the optimal model.

Confusion Matrix

- In the confusion matrix of the decision tree only one case was mislabeled.
- Overall, these model predict successful landings.

Conclusions

- As the number of flights increased, the success rate increased, and recently it has exceeded 80%
- The data seems to show that launch success rate of low weighted payloads is higher than that of heavy weighted payloads, but in reality, that failure its more related to the early launches.
- KSLC-39A has the highest number of launch successes and the highest success rate among all sites
- The model may need more training data but has a 94% accuracy score.
- The orbit type with best success are SSO, HEO, GEO, and ES-L1

Appendix

- GitHub
- Course at Coursera

Thank you!