IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

F Campbell
24 October 2022

Outline

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

Executive Summary

- The data required to answer the business question was collected in two methods: rocket launch data was collected using a get request from the SpaceX API, and a list of historical Falcon 9 launches was scraped from Wikipedia using the API BeautifulSoup. The data had to be cleaned, and missing values were replaced where necessary.
- Next the data was wrangled, where certain points of interest were determined including (but not limited to) determining the number of launches at each launch site and determining the landing outcomes for each launch site that were important for understanding the data. A landing outcome label was created to easily determine of a landing was successful (value of 1) or unsuccessful (value of 0).
- Using SQL, the data was explored and an interesting finding was the first successful landing date on a drone ship was 08 April 2016, whereas the first successful Ground Landing date was five months prior on 22 December 2015.
- The data was further explored by visualising different features together along with the outcome of the launch, mostly using scatter plots, to determine how important each feature is when determining the success of a landing.
- The launch sites were visually plotted and analysed using Folium, and a dashboard was created using Plotly Dash.
- Finally after exploring and understanding the data, the success rates of the first stage landings were predicted using 4 Machine Learning Algorithms: Logistic Regression, SVM, Decision Classification Tree and a K-Nearest Neighbours algorithm was run. The best hyperparameters for the four models were determined using a Grid Search method.
- To summarise the results of the four ML Models:
  - Logistic Regression had an accuracy of 84.64%, SVM had an accuracy of 84.82%, the Decision Tree had an accuracy of 87.82% and the K-Nearest Neighbours accuracy result was 84.82%.
  - The best algorithm to predict if the first stage will land successfully is the Decision Tree algorithm with an accuracy of **87.82%**

Introduction

- Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each. Much of the savings is because Space X can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be useful, as our alternate company, SpaceY, wants to bid against Space X for a rocket launch.

- This goal of the project is to create a machine learning pipeline to **predict if the first stage will land successfully**.
- In order to assist SpaceY to bid against SpaceX, the following problems will be answered:
  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data on the SpaceX rockets (Orbit, Booster Version, Pay Load Mass, Lauch Site, Outcome etc) as collected using the SapceX API. The list of historical Falcon 9 launches was scrapped from Wikipedia using the BeautifulSoup API.
- Data wrangling
  - Missing data points in Pay Load Mass values were replaced with the average Pay Load Mass, and a Class column was created as a classification variable that labels the outcome of each launch as successful (1) or unsuccessful (0)
- Exploratory data analysis (EDA) using visualization and SQL
- Interactive visual analytics using Folium and Plotly Dash
- Predictive analysis using classification models
  - How models were built, tuned, and evaluated

Data Collection

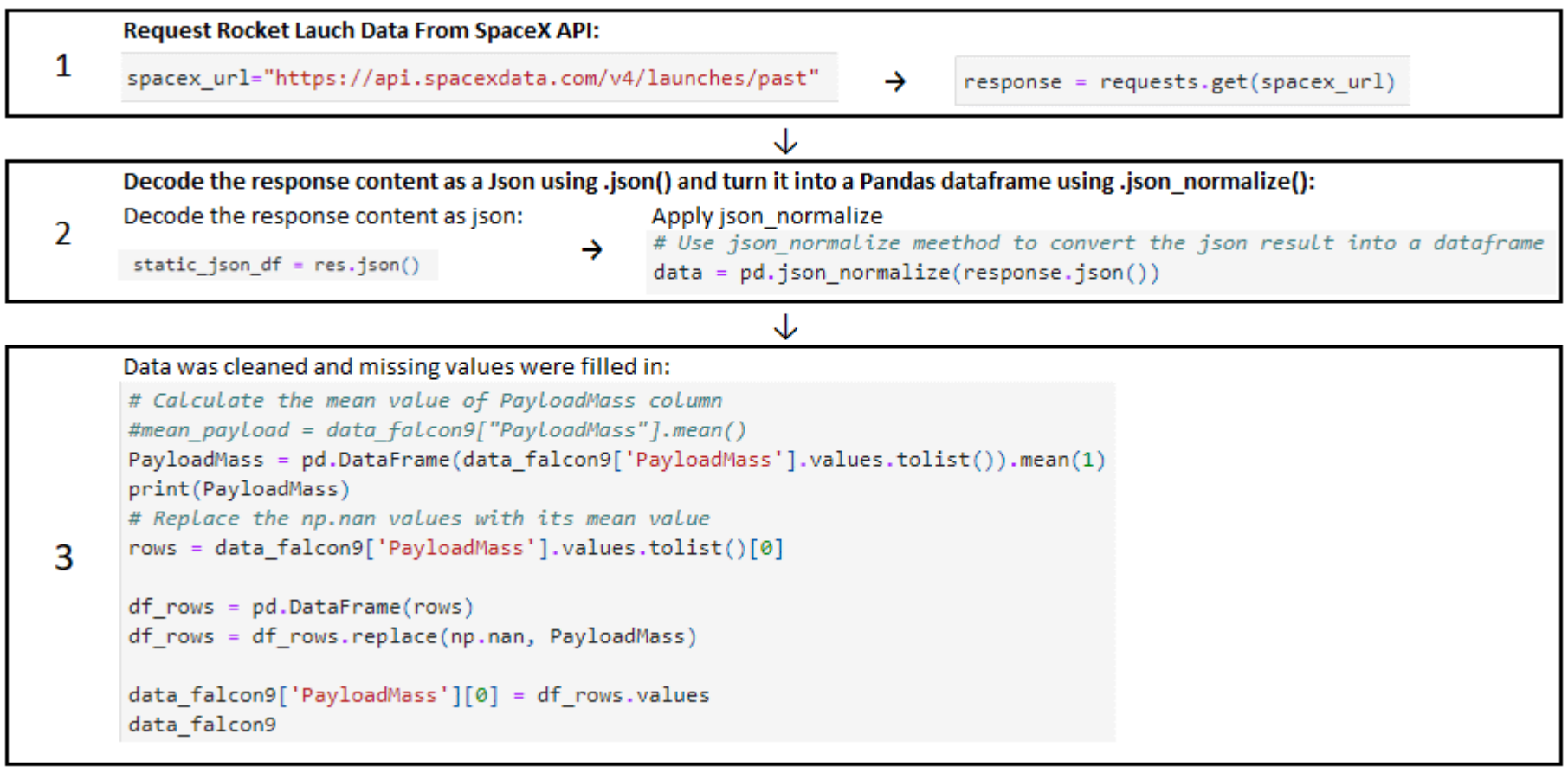
- The data was collected using various methods
  - Data collection was done using get request to the SpaceX API.
  - Next, we decoded the response content as a Json using .json() function call and turned this into a pandas dataframe using .json_normalize() function
  - The data was cleaned, checked for missing values and missing values in the Pay Load Mass column were filled in using the average Pay Load Mass.
  - In addition, Falcon 9 launch records were web scraped from Wikipedia using the BeautifulSoup API.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection – SpaceX API

The link to the notebook is:
https://github.com/fhcamp/SpaceX_project/blob/main/Data%20Collection%20SpaceX%20API.ipynb

A get request to the SpaceX API was used to collect rocket data, this was cleaned and some basic data wrangling and formatting was performed. The process followed is shown below:

Flowchart of SpaceX API calls:



Data Collection - Scraping

The link to the notebook is:
https://github.com/fhcamp/SpaceX_project/blob/main/Data%20Collection%20with%20Web%20Scraping.ipynb

Historical Falcon 9 launch data was scraped off Wikipedia using BeautifulSoup. The table was parsed and converted into a pandas dataframe. The process followed is shown below:

Flowchart of Scraping:

1	**Request Falcon 9 rocket launch page by applying the HTTP Get method** requests.get() method applied to url and the response assigned to the html_data object A BeautifulSoup object (soup) created from the above HTML response
	↓
2	**Extract all column names from the HTML table header** Find all the tables from the soup object using the find_all function and assign the result to *html_tables* Identify that the third table has the launch records ad assign this table to *first_launch_table* Iterate through the <the> elements and apply the extract_column_from_header() function to extract column name one by one
	↓
3	**Create a data frame by parsing the launch HTML tables** An empty dictionary (called launch_dict) with keys from the extracted column names in step 2 is created Using functions to loop through the rows in each column, the noise from the HTML tables are removed and the launch_dict is populated A Pandas Dataframe is creates using the columns from the lauch_dict and is filled in using the parsed data in the launch_dict
	↓
4	**Final results were exported as a csv using the to_csv command**

Data Wrangling

The link to the notebook is:
https://github.com/fhcamp/SpaceX_project/blob/main/Data%20Wrangling.ipynb

To understand the columns and to get training variables, the data was explored:

Flowchart of Wrangling:

1	The exploration included (using the .value_counts() method on the columns: a.) Calculating the number of launches at each site, b.) Each launch aims to a dedicated orbit, and therefore the number of launches for each orbit was determined c.) The unique mission outcomes and their occurrence was determined
	↓
2	A landing outcome label was created from the outcome column where 1 is a successful landing a 0 is unsuccessful. First a key was given to each type of outcome, and the bad outcomes were grouped together as a set and assigned to the variable bad_outcomes Looping through the 'Outcome' column, a list was created where: If the row in the 'Outcome' column is found in the bad_outcome set a zero is assigned Otherwise a one is assigned This variable will represent the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed successfully.
	↓
3	The results were exported as a csv via the to_csv method

EDA with Data Visualization

The link to the notebook:
https://github.com/fhcamp/SpaceX_project/blob/main/EDA%20with%20Data%20Visualisation.ipynb

To understand the relationship between potential features that impact the success of a landing, charts were used to visualize potential relationships.

- **Flight number vs Launch Site** (Scatter Plot): to see if there is a relationship between the launch site and the number of launches at the site and if this could indicate a successful launch. As we want to see if these are two features have an impact on the success, we chose a scatter plot.
- **Payload vs Launch Site** (Scatter Plot): to see if there is a relationship between the launch site and the weight of the payload at the launch site. We want to see if the weight of the payload at a site will impact the launch success, therefore a scatter plot was chosen.
- **Success rate of each Orbit Type** (Bar Chart): to answer the question of does the orbit type impact the success rate, it made sense to plot the number of successful launches against the orbit type in a bar chart to clearly see which orbit types have the most successful launches.
- **Flight Number and Orbit Type** (Scatter Plot): is there a relationship between the number of flights taken and the orbit type that will determine the success of the launch. This is best seen via a scatter plot.
- **Payload vs Orbit Type** (Scatter Plot): to see if there is a relationship between the payload mass and orbit type and if that impacts a successful landing, a scatter plot was again chosen.
- **Launch Success yearly trend** (Line Graph): In order to see if SpaceX has improved their technology and is increasing the number of successful launches over time. A line graph is best used to see a trend over time.

EDA with SQL

The link to the notebook:
https://github.com/fhcamp/SpaceX_project/blob/main/EDA%20with%20SQL.ipynb

- The SpaceX dataset was loaded into a PostgreSQL database and the data was explored and queried using SQL in the Jupyter notebook. The following queries were written to gain insights into the data:
  - The names of unique launch sites in the space mission.
  - Displayed 5 rows where the launch site started with 'KSC'
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The date where the first successful landing outcome on a drone ship was achieved
  - Listed the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000

EDA with SQL cont.

The link to the notebook: https://github.com/fhcamp/SpaceX_project/blob/main/EDA%20with%20SQL.ipynb

- The SpaceX dataset was loaded into a PostgreSQL database and the data was explored and queried using SQL in the Jupyter notebook. The following queries were written to gain insights into the data:
  - The total number of successful and failed mission outcomes
  - The names of the booster versions which carried the maximum payload mass
  - Since 2017, by month, all the successful landing outcomes in ground pad and their booster versions and launch site
  - Rank, in descending order, the count of the successful landing outcomes between 2010-06-04 and 2017-03-20

Build an Interactive Map with Folium

The link to the notebook is:
https://github.com/fhcamp/SpaceX_project/blob/main/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

- To identify launch sites, a folium.circle map objects was added to highlight the launch site with a circle. This was lablled using folium.map.Marker. This was done so that all launch sites across the map of America could be easily and quickly identified.
- To identify the successful or failed launches for each site. If a launch is successful it received a green marker, otherwise red.
  - A MarkerCluster was used to simplify the map as many launches will contain the same coordinates.
  - This was done so that the user can very quickly identify which launch sites have relatively high success rates.
- The distance between launch sites and it's proximities can also be calculated.
  - A MousePosition was added to the map to get coordinate for a mouse over a point on the map. This is done so that while the user is exploring the map, you can easily find the coordinates of any points of interests (such as railway) the distances between a launch site to its proximities.
  - A point of interest can be determined (for example a coastline). To identify this point, a folium.Marker can be used, and a PolyLine can be drawn between the launch site and the coastline to easily see how close the site is to the sea.
  - Distinct lines can help answer questions, for instance:
    - Are launch sites near railways, highways and coastlines.
    - Do launch sites keep certain distance away from cities.

Build a Dashboard with Plotly Dash

The link to the notebook is:
https://github.com/fhcamp/SpaceX_project/blob/main/app.py

- Pie charts were created to show the total successful launches by launch sites
  - This graph was chosen so that the user can easily identify which site had highest success rate
  - After the site with the highest success rate, the user can also view the launch success ratio of specific chosen sites
- In order to see if there is a relationship between Outcome and Payload Mass (Kg) for different booster versions, a scatter plot was created for all sites
  - A sliding scale was used so that the user can easily choose what payload they want to investigate

Predictive Analysis (Classification)

The link to the notebook is:
https://github.com/fhcamp/SpaceX_project/blob/main/Machine%20Learning%20Prediction.ipynb

Flowchart of process followed to create 4 ML models and choose best performing one:

Load the data into a Pandas Dataframes: data and x

↓

1 Create a NumPy array from the Class column and assigned it to the variable y
This is what we want to predict

↓

2 Standardesed the data in dataframe X using StandareScaler and reassigned it to the variable x uing the transform.fit_transform(X) method

↓

3
Split the data into training and testing data using the function train_test_split.
The function train_test_split is used to split the data X and Y into training and test data.
The parameter test_size set to 0.2 and random_state to 2.
The training data and test data were assigned the following labels:
X_train, X_test, Y_train, Y_test
This resulted in 18 test samples

↓

4
Run 4 ML Models, the models are trained and hyperparameters are selected using the function GridSearchCV
For each model:
A logistic regression/support vector machine/decsion tree classifier/k nearest neighbour objects were created, along with a GridSearchCV object with cv = 10.
The object was fitted to find the best parameters from the dictionary parameters.
The accuracy of the test data was calcualted using he method score
A confusion matrix was drawn to evaluate the outcome of the model

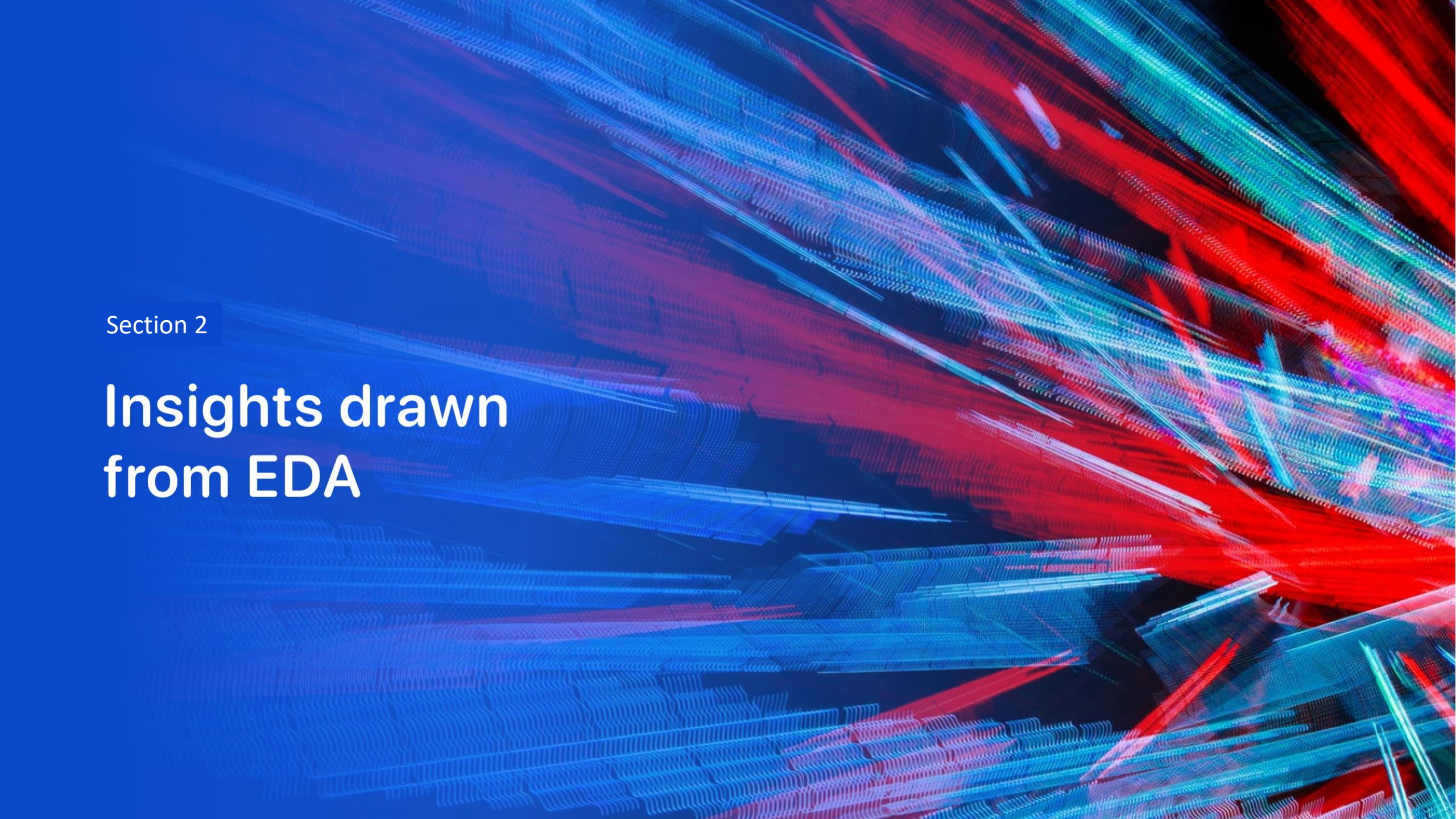
↓

5 The best performing model (method) was determed by comparing the scores of each model

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

Results

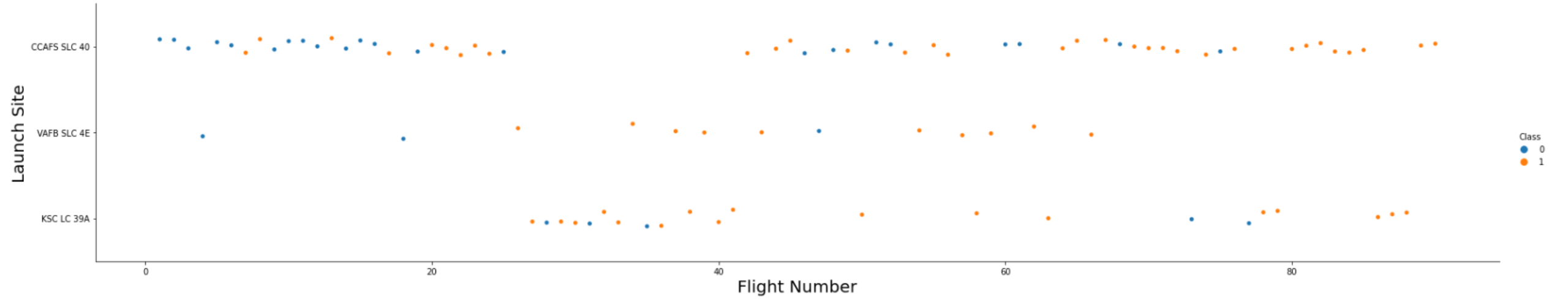
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

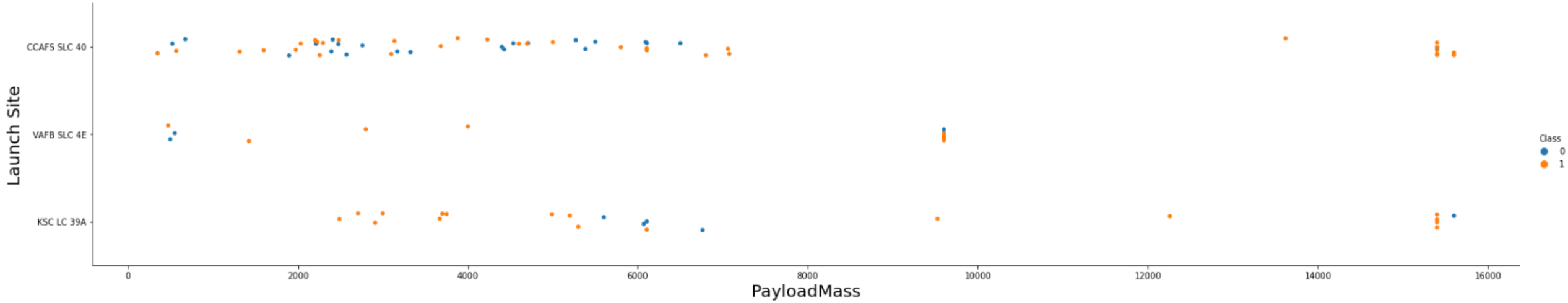
From the plot, it can be seen that:

1. the larger the flight number at a launch site, the greater the success rate at a launch site
2. Launch Site CCAFS SLC 40 has a very high success rate for approximately 70+ flights



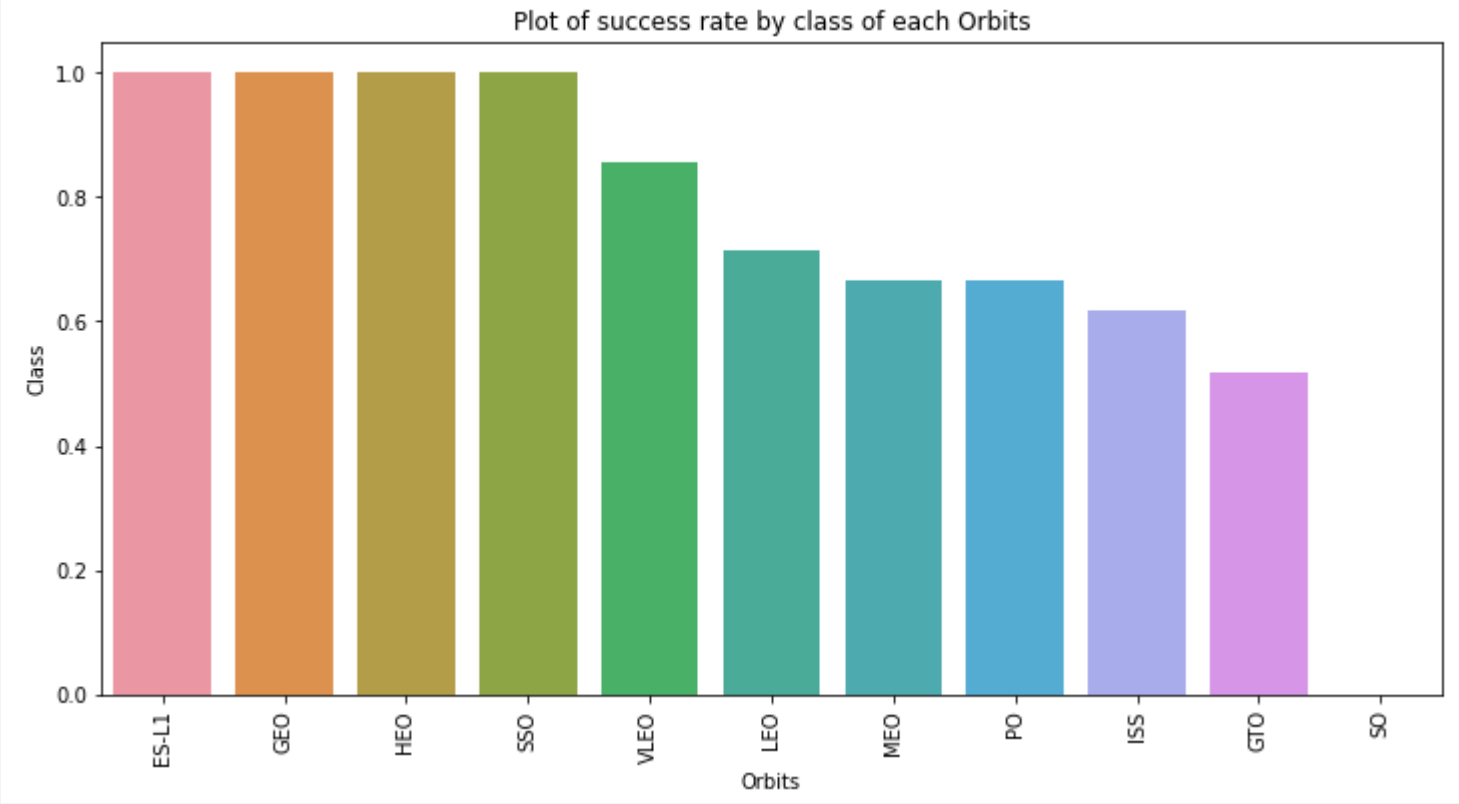
Payload vs. Launch Site

- The heavier the payload mass, the more successful the launches are, especially at launch site CCAFS SLC 40
- At site VAFB SLC 4E there are no rockets launched for heavy payload mass (greater than 10000kg).



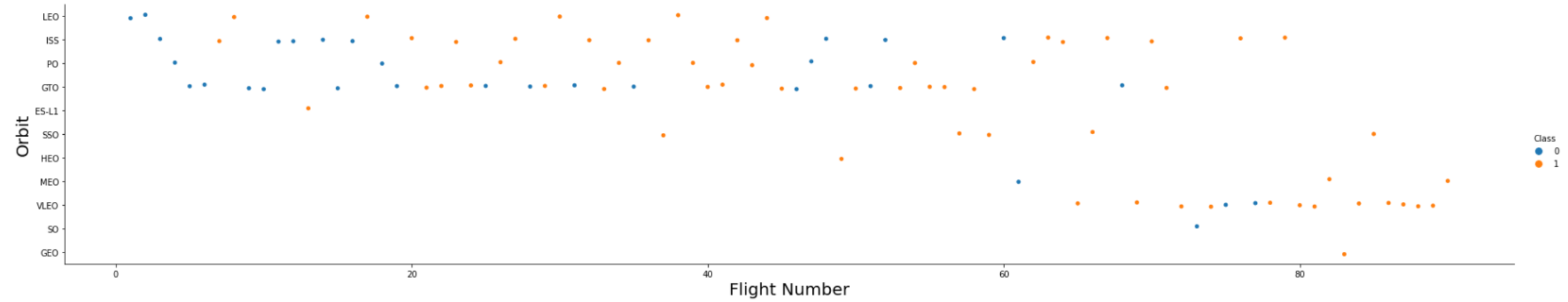
Success Rate vs. Orbit Type

- The ES-L1, GEO, HEO, SSO, VLEO orbits have the most success rates
- Orbit SO was completely unsucessful



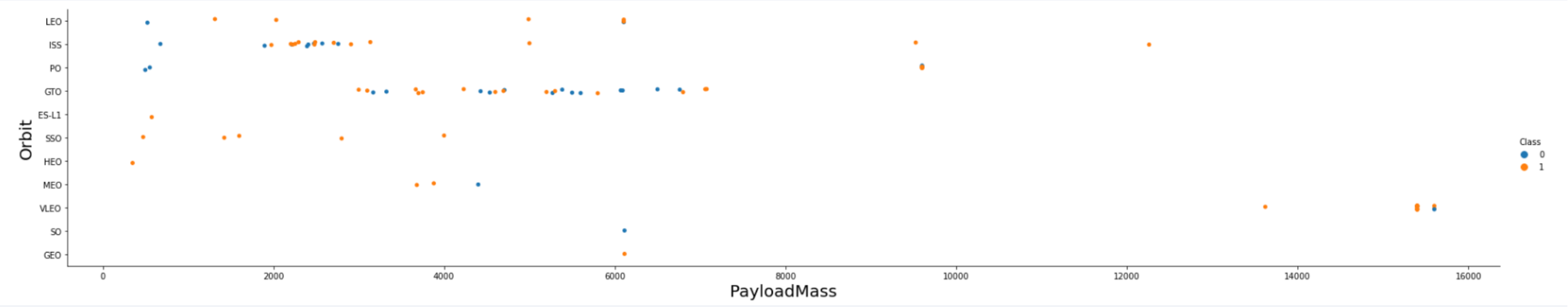
Flight Number vs. Orbit Type

LEO orbit the Success appears related to the number of flights in the LEO orbit whereas for the GTO orbit there seems to be no relationship between flight number the orbit.



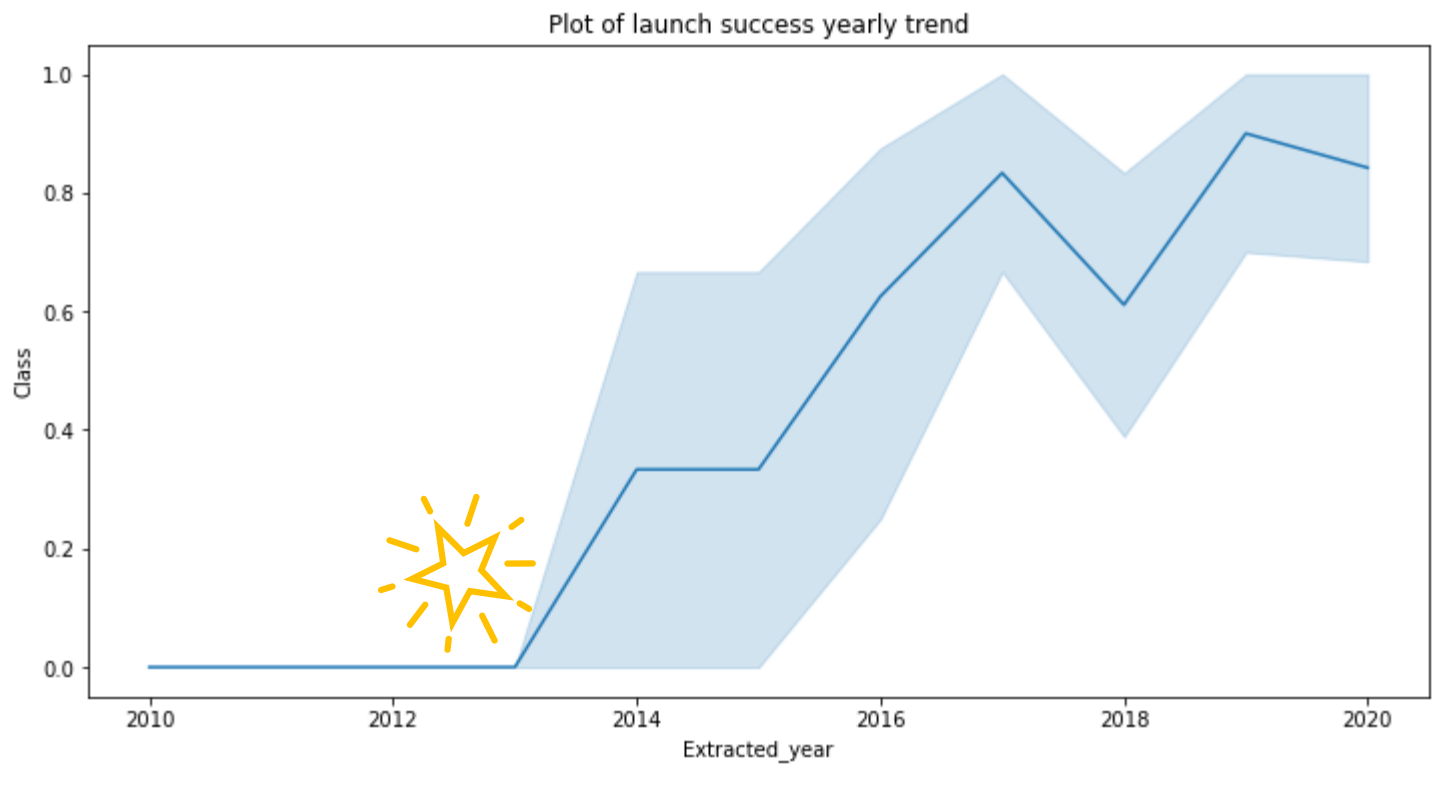
Payload vs. Orbit Type

- With heavy payloads the successful landing rate are more for Polar, LEO and ISS orbits
- GTO this cannot be distinguished as both successful and unsuccessful landing rates are seen.



Launch Success Yearly Trend

- The success rate has been increasing since 2013 until 2020



All Launch Site Names

The key word **DISTINCT** is used to show only unique launch sites from the SpaceX data

Task 1

Display the names of the unique launch sites in the space mission

```
%sql
SELECT DISTINCT
    Launch_Site
FROM spacex
```

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

Launch Site Names Begin with 'KSC'

- The key word **LIKE** in the **WHERE** clause with the syntax 'KSC%' will identify launch sites that start with KSC (the % is at the end, this ensures that the launch site name must start with KSC)
- **LIMIT** is used to show only 5 rows

Task 2

Display 5 records where launch sites begin with the string 'KSC'

```
%sql
SELECT
    *
FROM spacex
WHERE
    Launch_Site LIKE 'KSC%'
LIMIT 5
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG..	Orbit	Customer	Mission_Outcome	Landing _Outcome
19-02-2017	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
16-03-2017	06:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
30-03-2017	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
01-05-2017	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
15-05-2017	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

Total Payload Mass

- The aggregate function **SUM** was used to calculate the total payload and the **WHERE** clause specified that the customer must be from NASA (CRS)
- The total payload mass: 45596kg

Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql
SELECT
    SUM(PAYLOAD_MASS__KG_) AS Total_PayloadMass
FROM spacex
WHERE
    Customer LIKE 'NASA (CRS)'
```

Total_PayloadMass
45596.0

Average Payload Mass by F9 v1.1

- The aggregate function **AVG** was used to calculate the average payload and the **WHERE** clause specified that the booster version must be F9 v1.1
- The Average Payload Mass by F9 v1.1: 2928.4kg

Task 4

Display average payload mass carried by booster version F9 v1.1

```
%sql
SELECT
    AVG(PAYLOAD_MASS__KG_) AS AVG_PayloadMass
FROM spacex
WHERE
    Booster_Version = 'F9 v1.1'
```

AVG_PayloadMass
2928.4

First Successful Drone Ship Landing Date

- The function **MIN** was used to get the minimum date from the date column, This was given the alias FirstSuccessfull_landing_date, and the **WHERE** statement specifies that the landing outcome must be for the Success (drone ship) type
- The first successful landing date on a drone ship was 08 April 2016

Task 5

List the date where the first successful landing outcome in drone ship was acheived.

```
%sql
SELECT
    MIN(date) AS FirstSuccessful_landing_date
FROM spacex
WHERE
    Landing_Outcome = 'Success (drone ship)'
```

FirstSuccessful_landing_date
2016-04-08

This is interesting to contrast against the first successful Ground Landing, which occurred only 5 months prior on 22 December 2015.

Successful Ground Pad Landing with Payload between 4000 and 6000

A **WHERE** clause filters for boosters which have successfully landed on a ground pad and an **AND** condition was included to filter the payload mass to be greater than 4000 but less than 6000

Task 6

List the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000

```
%sql
SELECT
    Booster_Version
FROM spacex
WHERE
    Landing_Outcome = 'Success (ground pad)'
    AND PAYLOAD_MASS__KG_ > 4000
    AND PAYLOAD_MASS__KG_ < 6000
```

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

Total Number of Successful and Failure Mission Outcomes

- A **CASE WHEN** clause is used to group all the Successful launch mission outcomes and label these as ‘Success’, otherwise give the label ‘Failure’
- Each occurrence of the case clause is counted to get a mission count used the aggregate function **COUNT**. As an aggregate function is used, the whole statement is **grouped by the case clause**
- The result is 100 missions with a successful outcome, and 1 with a failure

Task 7

List the total number of successful and failure mission outcomes

```
%sql
SELECT
CASE
    WHEN Mission_Outcome LIKE 'Success%' THEN 'Success'
    ELSE 'Failure'
END AS mission_outcome,
COUNT(*) AS outcome_count
FROM spacex
GROUP BY
CASE
    WHEN Mission_Outcome LIKE 'Success%' THEN 'Success'
    ELSE 'Failure'
END
```

mission_outcome	outcome_count
Success	100
Failure	1

Boosters Carried Maximum Payload

- The maximum payload is first determined in a subquery using the aggregate **MAX** function on the payload mass column
- This subquery is inputted in the **WHERE** statement of the outer query which then selects all the boosters that have carried this max payload mass

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%sql
SELECT
    Booster_Version,
    PAYLOAD_MASS__KG_
FROM spacex
WHERE PAYLOAD_MASS__KG_ = (
                        SELECT
                            MAX(PAYLOAD_MASS__KG_)
                        FROM spacex
                        )
ORDER BY Booster_Version
```

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

2017 Successful Ground Pad Launch Records by month

Task 9

List the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017

- The **date function** was used to extract the month name from the Date column, then the landing outcome, booster versions and launch sites are selected.
- The **WHERE** statement ensures only successful ground pad landing outcomes are selected, and another date function is performed to extract the year from the date column so that we filter the data to only 2017 records
- Finally the data is **ordered by** the month number of the dates so that the result is displayed in chorological order.

```
%sql
SELECT
    date_format(date,'MMMM') AS month_name,
    Landing_Outcome,
    Booster_Version,
    Launch_Site
FROM spacex
WHERE
    Landing_Outcome = 'Success (ground pad)'
    AND date_format(date,'yyyy') = '2017'
ORDER BY month(date)
```

month_name	Landing_Outcome	Booster_Version	Launch_Site
February	Success (ground pad)	F9 FT B1031.1	KSC LC-39A
May	Success (ground pad)	F9 FT B1032.1	KSC LC-39A
June	Success (ground pad)	F9 FT B1035.1	KSC LC-39A
August	Success (ground pad)	F9 B4 B1039.1	KSC LC-39A
September	Success (ground pad)	F9 B4 B1040.1	KSC LC-39A
December	Success (ground pad)	F9 FT B1035.2	CCAFS SLC-40

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

- Landing outcomes and the **COUNT** of landing outcomes were selected from the data and a **WHERE** clause is used to filter for dates **BETWEEN** 2010-06-04 to 2017-03-20.
- As COUNT is an aggregate function, the data was grouped by (**GROUP BY** clause) the landing outcomes
- The **ORDER BY** clause ranks the counts in descending order.

Task 10

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

```
%sql
SELECT
    Landing_Outcome,
    COUNT(Landing_Outcome) AS Landing_Outcome_count
FROM spacex
WHERE
    date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY
    Landing_Outcome
ORDER BY
    Landing_Outcome_count DESC
```

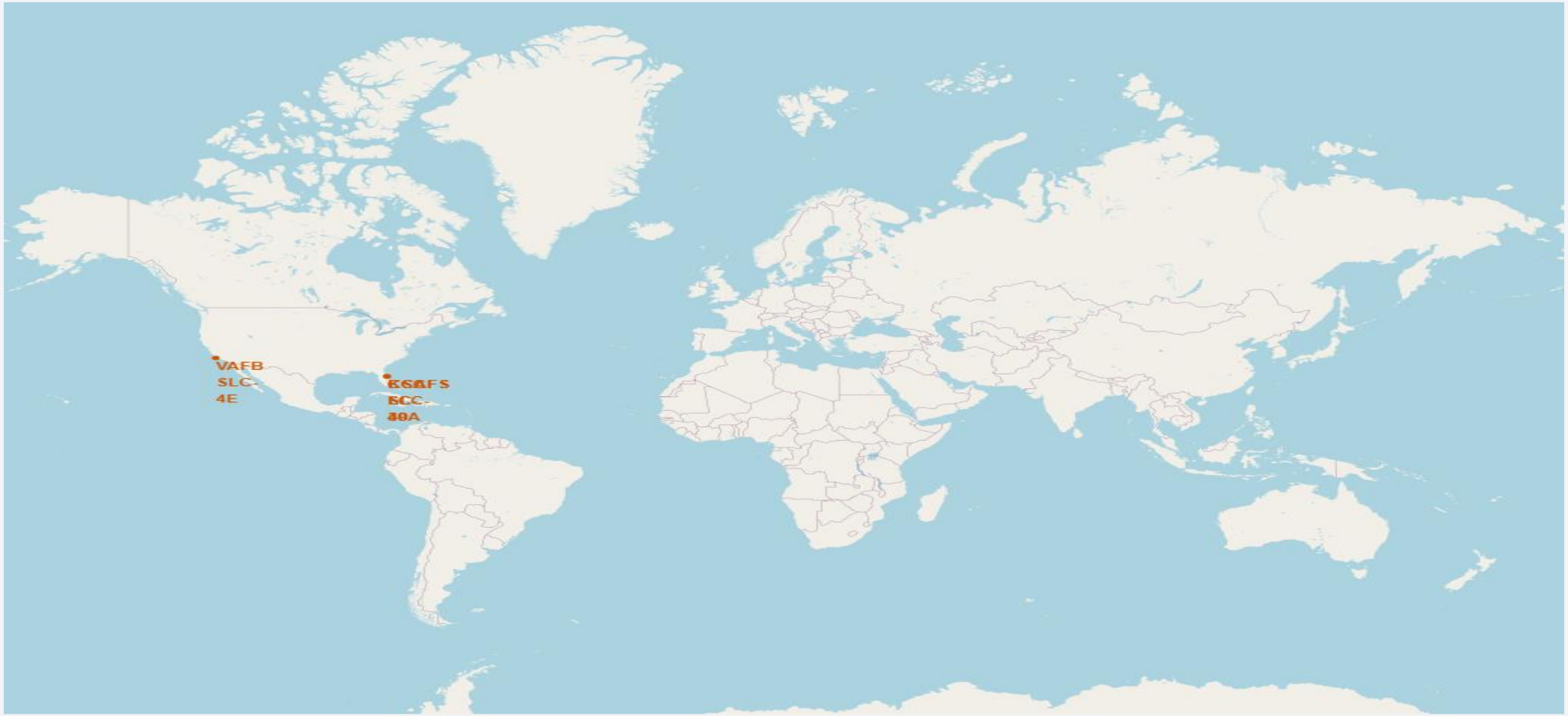
Landing_Outcome	Landing_Outcome_count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

All Launch Sites as seen Globally

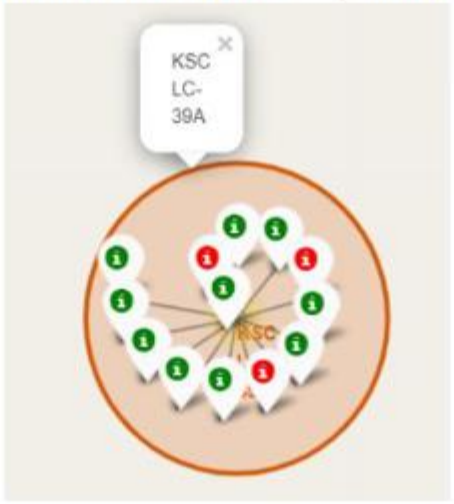
- The landing sites are in the USA, along the Florida and California coasts

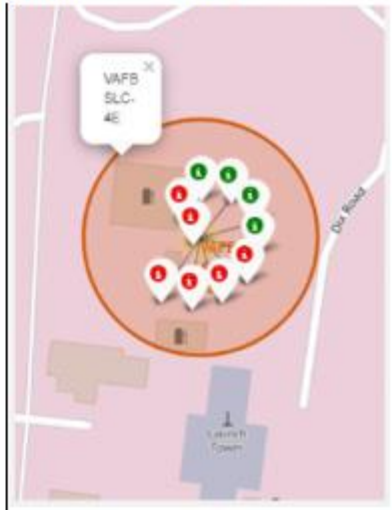


Launch Sites and Success Rates

Florida Launch Sites | **California Launch Site**

The green markers indicate the number of successful launch's, and the red the number of failed





From this it seems tat launch site KSC LC-39A has more successful launches of the three sites

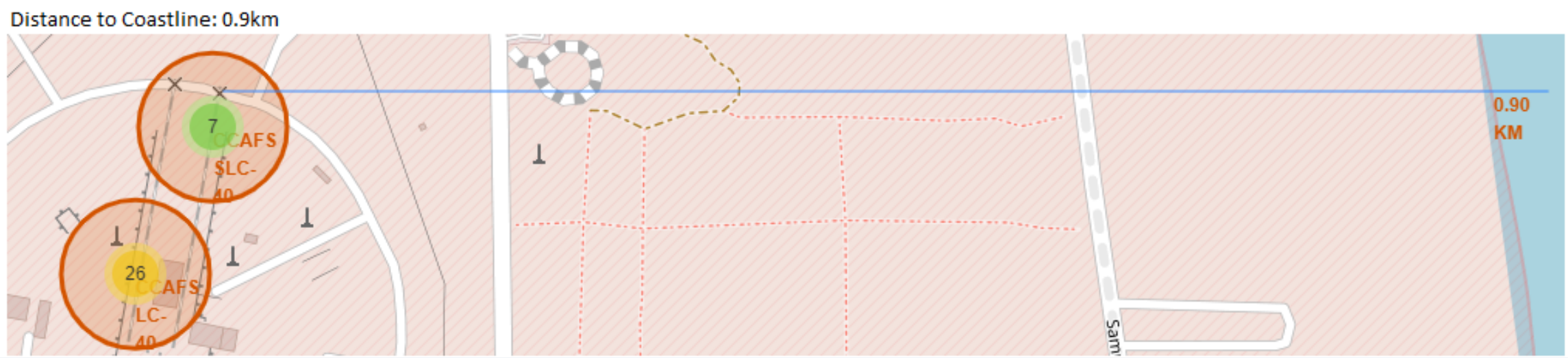
Launch site CCAFSLC - 40 has 7 successful launches, and 26 failures

The green circle indicates the 7 successful launches, while the yellow circle indicates the failed 26 launches

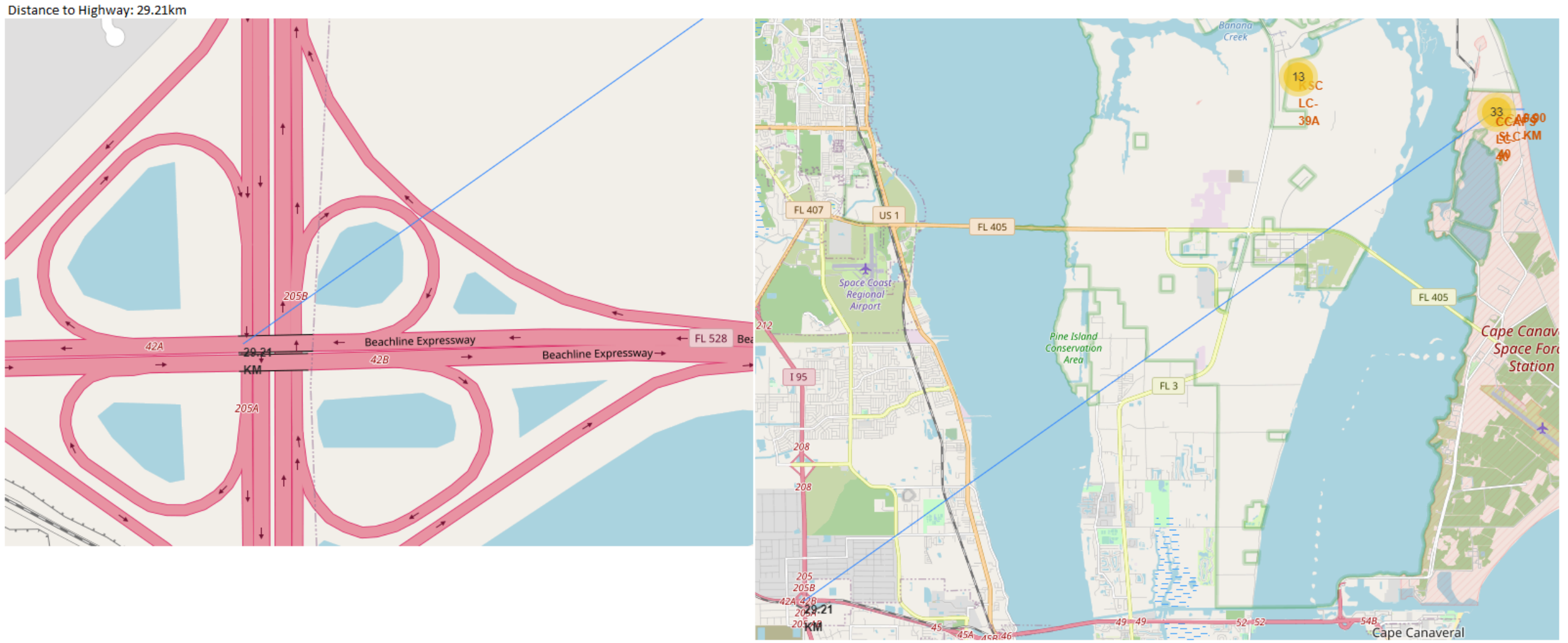
Launch Site distances to landmarks

- The following questions were answered:
  - Are launch sites in close proximity to railways? No
  - Are launch sites in close proximity to highways? No
  - Are launch sites in close proximity to coastlines? Yes
  - Do launch sites keep certain distances away from cities? Yes
- Examples of distances will be shown using launch site CCAFSLC – 40 on the following slides:
  - 0.9km to the nearest coastline
  - 29.21km to a highway (Beachline Expressway)
  - 78.45km to Orlando City

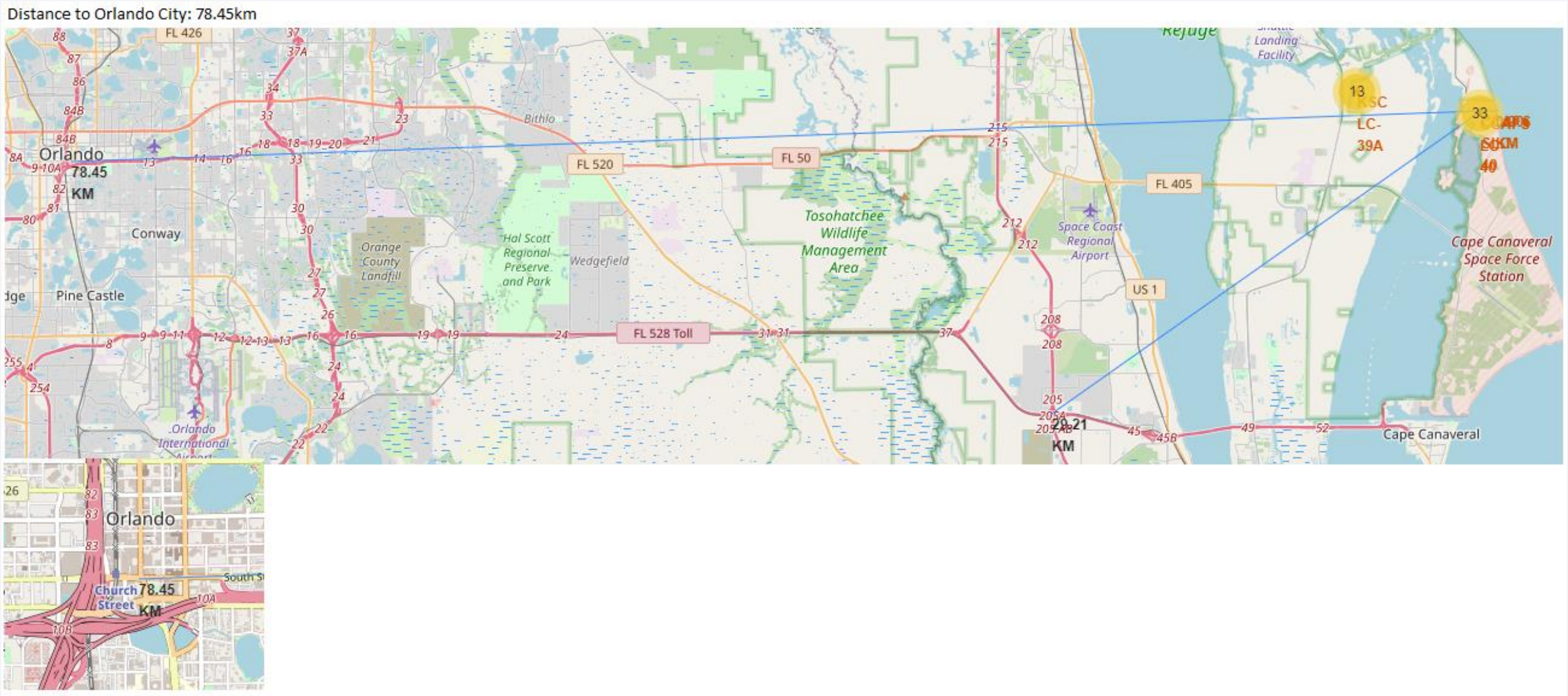
Launch Site CCAFS SLC – 40 distance to Coastline

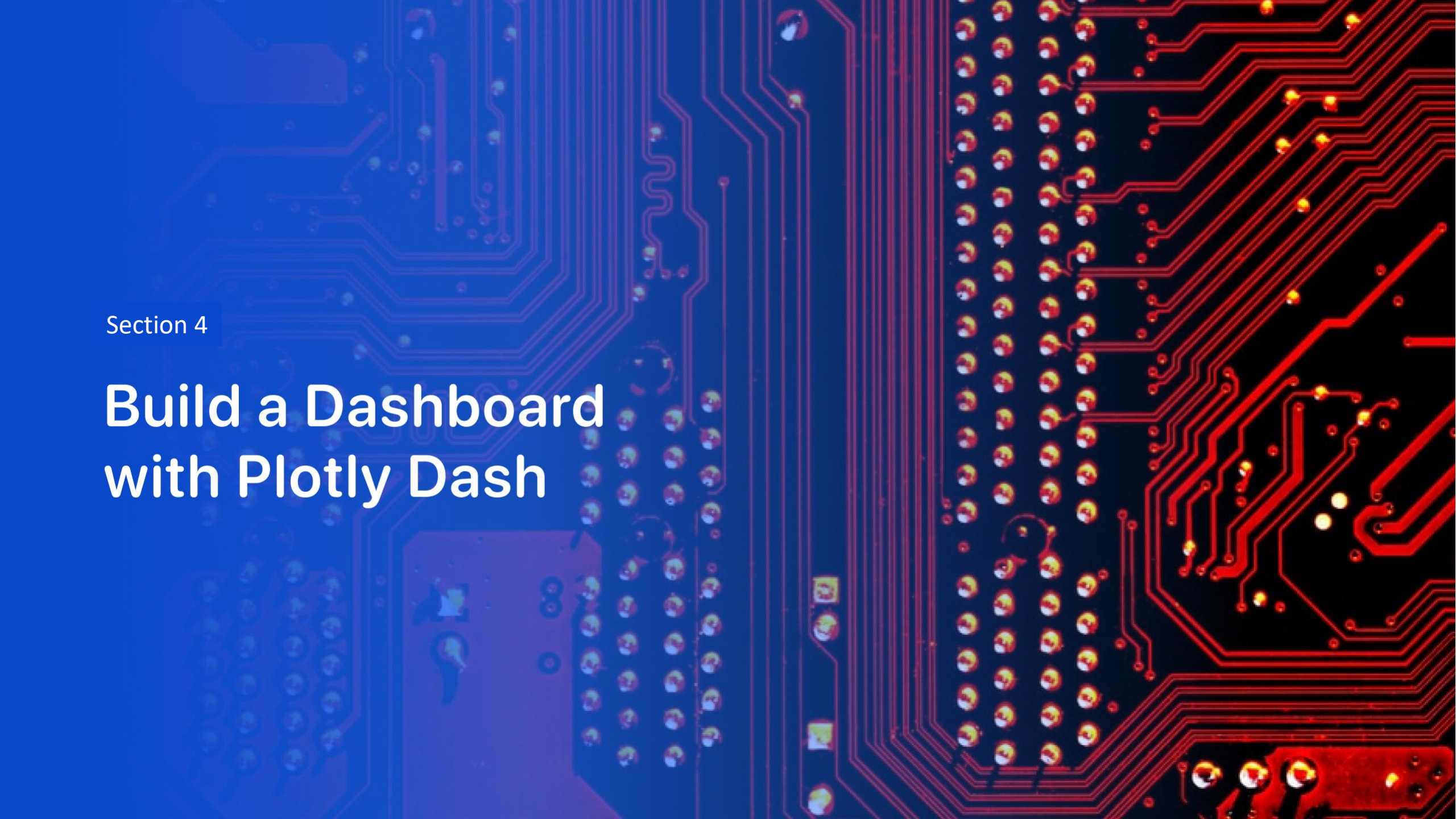


Launch Site CCAFS SLC – 40 distance to Highway



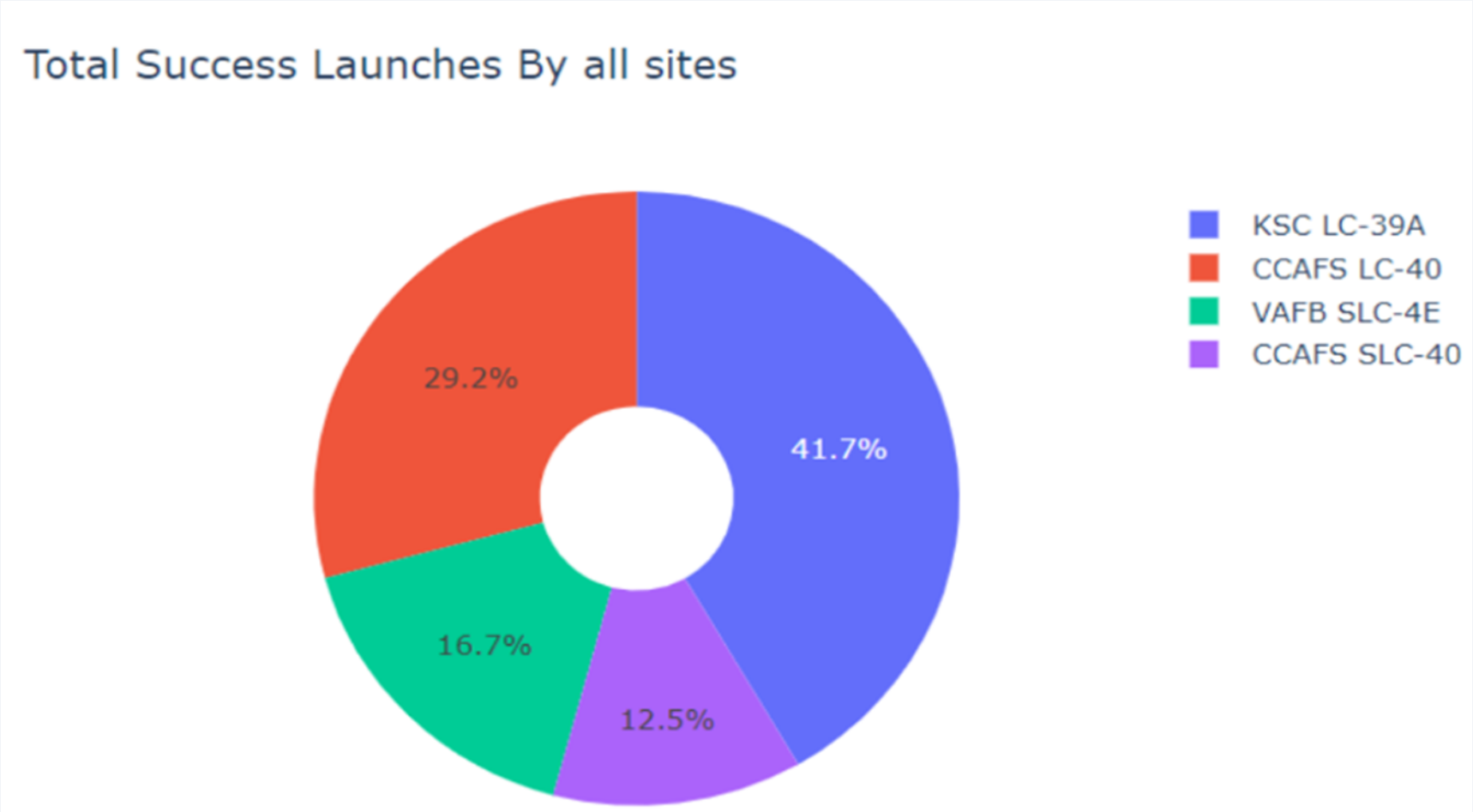
Launch Site CCAFS SLC – 40 distance to Orlando City




Section 4
Build a Dashboard with Plotly Dash

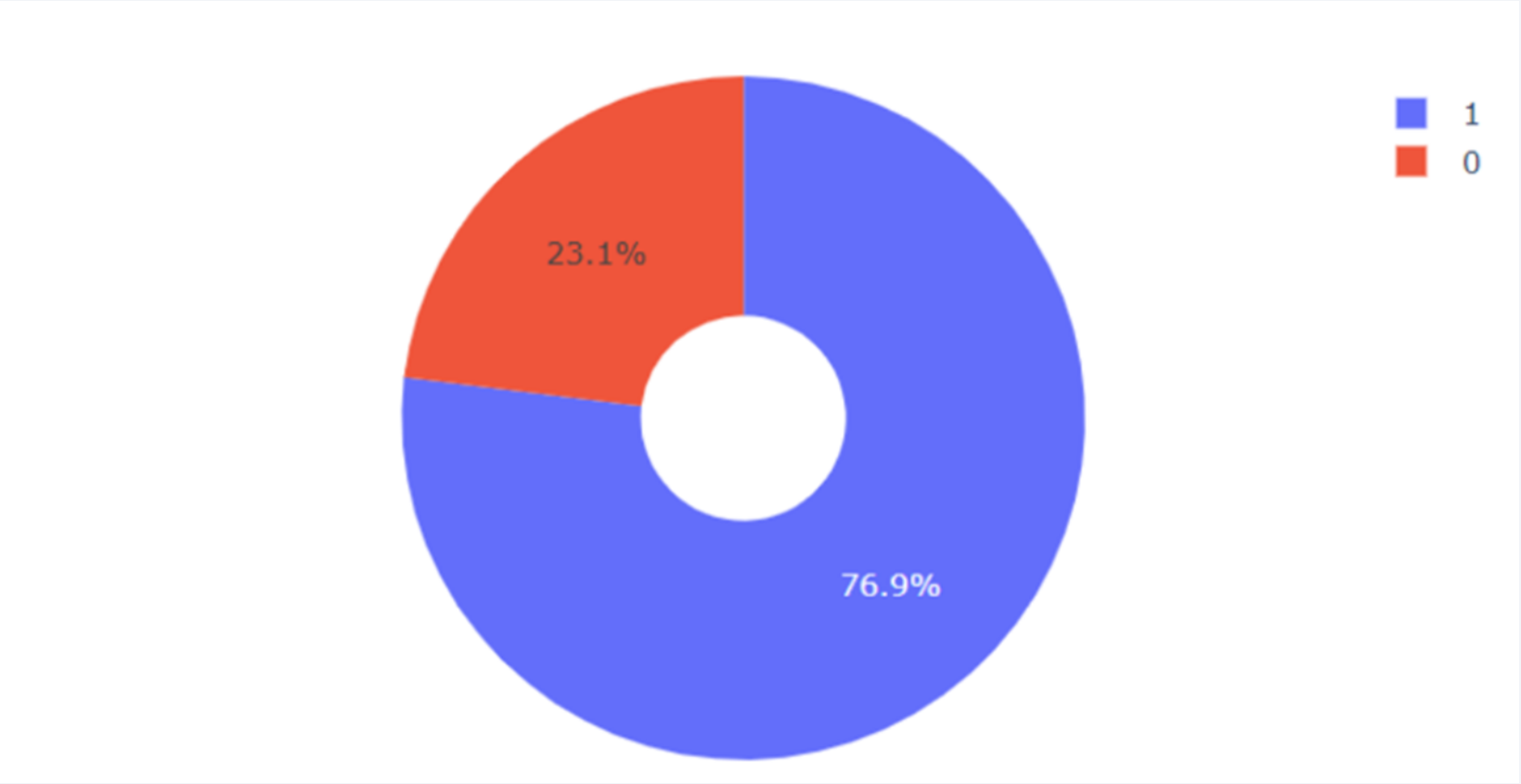
Success Rate per Launch Site

Launch Site KSC LC-39A has the highest launch success rate of 41.7% compared to the other launch sites



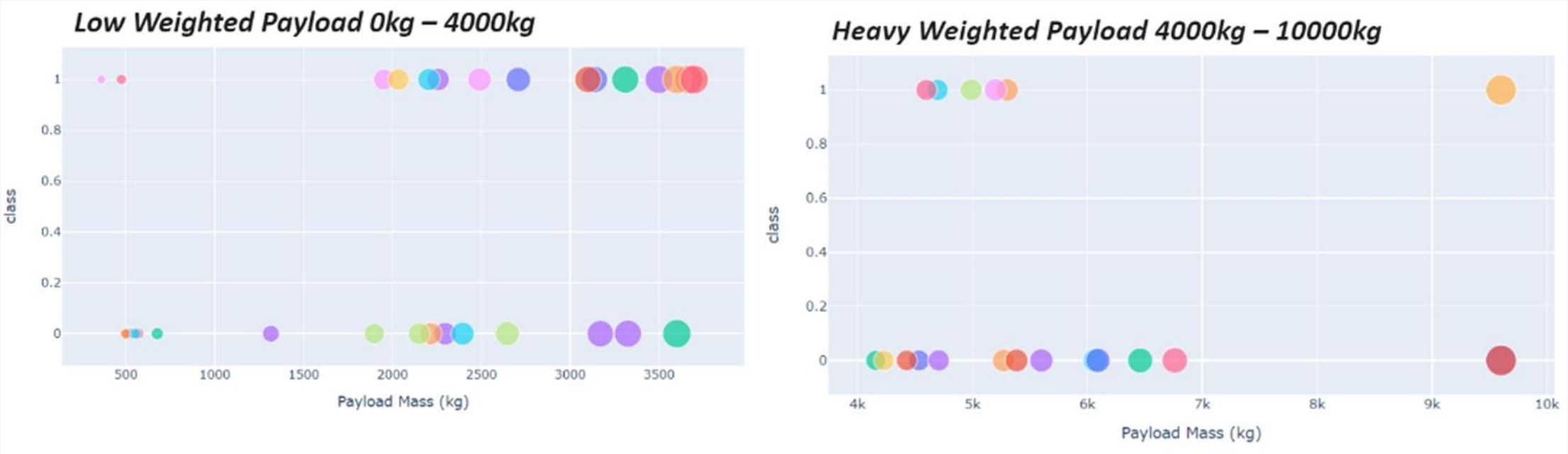
Launch Site KSC LC-39A Success Ratio

Launch Site KSC LC-39A has a success rate of 76.9%, and a failure rate of 23.1%



Payload vs Launch Outcome for all Sites

The success rates for lighter payloads is higher compared to heavier payloads.

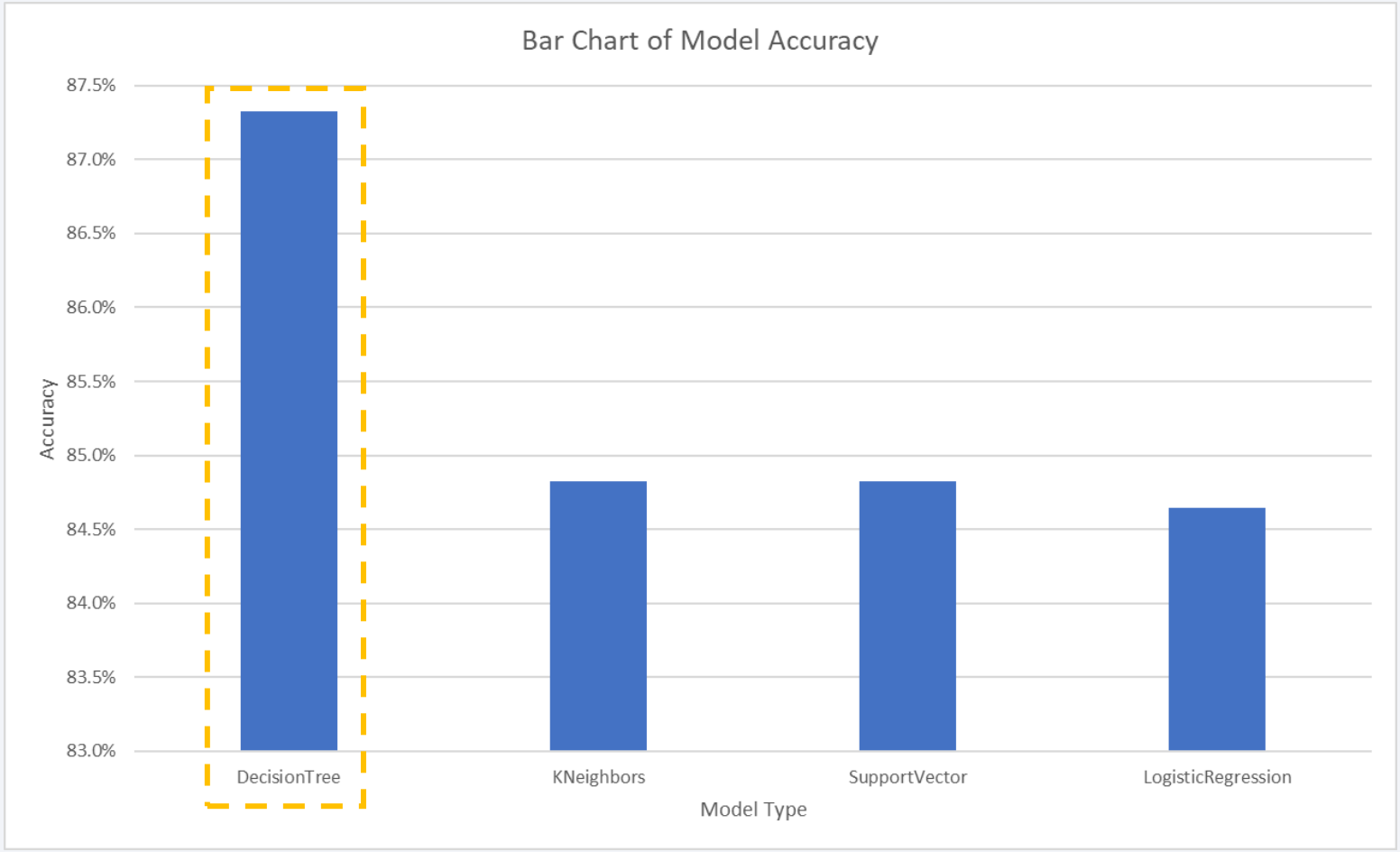


Section 5

Predictive Analysis (Classification)

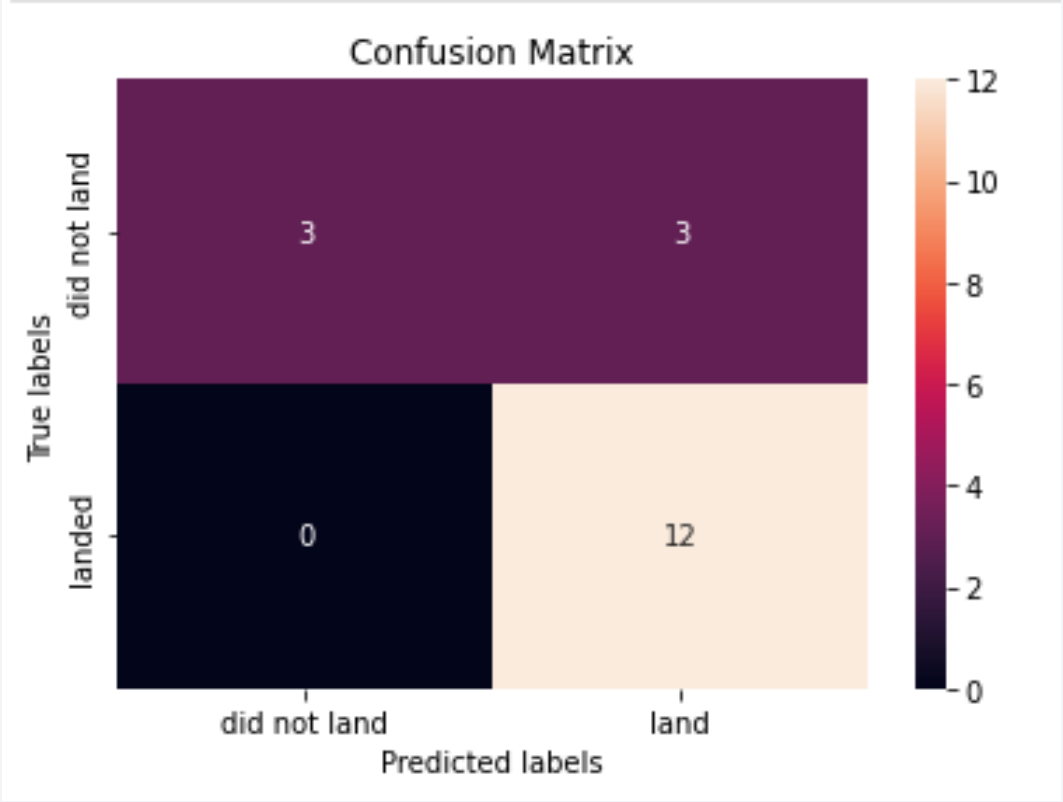
Classification Accuracy

The Decision Tree Classification model has the highest accuracy of 87.32%



Confusion Matrix of Decision Tree Model

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes.
- The matrix shows the problem with the model is the false positives – where the rocket did not land (unsuccessful) but it is predicted to have landed (successful)
- In no instance did the model predict that the rocket did not land, while in truth the rocket did land. In other words, the true positives were predicted correctly



Conclusions

- The higher the number of flights at a launch site, the greater the success rate is at a launch site.
- Launch success rates started to increase in 2013 until 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO have the highest success rates.
- KSC LC-39A has the most successful launches out of all the sites.
- The Decision Tree Classifier predicted if the first stage will land successfully with the greatest accuracy.

Thank you!