IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Toh Yi Ting
17 July 2024

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

Summary of methodologies

- Data collection via API and webscraping
- Data analysis and visualization on data collected
- Predictive analysis with classification models

Summary of all results

- Success rate of launches have generally increased over the years
- Predictive analysis is not effective with a small sample dataset

Introduction

- SpaceX is a company looking to make space travel affordable to everyone.
- I will be looking at SpaceX's Falcon 9 rocket launches to determine the cost of a launch and whether SpaceX can reuse the first stage to maximise cost savings.
- When SpaceX can recover the first stage (rocket does not crash etc), it can pass off the cost savings to customers and price cheaper than its other competitors.
- Taking on the role of a data scientist in SpaceX, I will train a machine learning model and use public information to predict if SpaceX can reuse the first stage while also determining the price of each launch.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data collected via API and webscraping
- Perform data wrangling
  - Data is cleaned using Python libraries
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Logistic Regression, Support Vector Machine, Decision Tree, K Nearest Neighbour models

Data Collection

- Data collection is done via API and also webscraping.

Using API:

- The SpaceX launch data will be gathered from the SpaceX REST API. This API gives us data about launches, which includes information about the rocket used, payload delivered, launch specifications, landing specifications and landing outcome.
- Using this data, we will predict whether SpaceX will land a rocket or not.

Data Collection

1. Get data from SpaceX REST API
2. View results by calling .json method
3. Webscraping related Wiki pages

Using this endpoint URL(api.spacexdata.com/v4/launches/past) to get past launch data.

Responses obtained will be in the json format and to convert this JSON to a dataframe, we use the json_normalize function.

Use the Python BeautifulSoup package to webscrape some HTML tables that contain Falcon 9 launch records. Data from these tables will be converted into a Pandas dataframe for further visualization and analysis.

Data Collection – SpaceX API

- GitHub URL of the completed SpaceX API calls notebook: https://github.com/yttoh/datascicapstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Steps:

1. Request and parse the SpaceX launch data using the GET request
2. Decode the response content as a Json using .json() and turn it into a Pandas dataframe using .json_normalize()
3. Filter the dataframe to only include Falcon 9 launches
4. Handle missing values
5. Export dataset to a CSV

Then, we need to create a Pandas data frame from the dictionary launch_dict.

[28]:
```
# Create a data from launch_dict
data=pd.DataFrame(launch_dict)
```

Show the summary of the dataframe

[29]:
```
# Show the head of the dataframe

data.head()
```

[29]:

ber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	Reus
1	2006-03-24	Falcon 1	20.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	
2	2007-03-21	Falcon 1	NaN	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	
4	2008-09-28	Falcon 1	165.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	
5	2009-07-13	Falcon 1	200.0	LEO	Kwajalein Atoll	None None	1	False	False	False	None	NaN	
6	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	

Data Collection - Scraping

- GitHub URL of the completed web scraping notebook: https://github.com/yttoh/datascicapstone/blob/main/jupyter-labs-webscraping.ipynb

Steps:

1. Web scrap Falcon 9 launch records with Python BeautifulSoup from Wikipedia
2. Parse the table and convert it into a Pandas data frame

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Ou
1	1	2006-03-24	True	20.0	LEO	Kwajalein Atoll	None
2	2	2007-03-21	True	5919.16534090909	LEO	Kwajalein Atoll	None
3	3	2008-09-28	True	165.0	LEO	Kwajalein Atoll	None
4	4	2009-07-13	True	200.0	LEO	Kwajalein Atoll	None
5	5	2010-06-04	True	5919.16534090909	LEO	CCSFS SLC 40	None
6	6	2012-05-22	True	525.0	LEO	CCSFS SLC 40	None
7	7	2013-03-01	True	677.0	ISS	CCSFS SLC 40	None
8	8	2013-09-29	True	500.0	PO	VAFB SLC 4E	False
9	9	2013-12-03	True	3170.0	GTO	CCSFS SLC 40	None
10	10	2014-01-06	True	3325.0	GTO	CCSFS SLC 40	None
11	11	2014-04-18	True	2296.0	ISS	CCSFS SLC 40	True
12	12	2014-07-14	True	1316.0	LEO	CCSFS SLC 40	True
13	13	2014-08-05	True	4535.0	GTO	CCSFS SLC 40	None
14	14	2014-09-07	True	4428.0	GTO	CCSFS SLC 40	None
15	15	2014-09-21	True	2216.0	ISS	CCSFS SLC 40	False
16	16	2015-01-10	True	2395.0	ISS	CCSFS SLC 40	False
17	17	2015-02-11	True	570.0	ES-L1	CCSFS SLC 40	True
18	18	2015-04-14	True	1898.0	ISS	CCSFS SLC 40	False
19	19	2015-04-27	True	4707.0	GTO	CCSFS SLC 40	None
20	20	2015-06-28	True	2477.0	ISS	CCSFS SLC 40	None
21	21	2015-12-22	True	2034.0	LEO	CCSFS SLC 40	True
22	22	2016-01-17	True	553.0	PO	VAFB SLC 4E	False
23	23	2016-03-04	True	5271.0	GTO	CCSFS SLC 40	False

Data Wrangling

Steps:

1. Identify and calculate the percentage of the missing values in each attribute
2. Calculate the number of launches on each site
3. Calculate the number and occurrence of eac orbit
4. Calculate the number and occurence of mission outcome of the orbits
5. Create a landing outcome label from Outcome column

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

```
[3]: df.isnull().sum()/len(df)*100

[3]: FlightNumber       0.000000
     Date               0.000000
     BoosterVersion     0.000000
     PayloadMass        0.000000
     Orbit              0.000000
     LaunchSite         0.000000
     Outcome            0.000000
     Flights            0.000000
     GridFins           0.000000
     Reused             0.000000
     Legs               0.000000
     LandingPad        28.888889
     Block              0.000000
     ReusedCount        0.000000
     Serial             0.000000
     Longitude          0.000000
     Latitude           0.000000
     dtype: float64
```

	Outcome	Class
0	True ASDS	1
1	True RTLS	1
2	True Ocean	1

We can use the following line of code to determine the success rate:

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

- Add the GitHub URL of completed data wrangling related notebook: https://github.com/yttoh/datascicapstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Charts that were plotted:

01 Scatterplots (to explore relationship between variables)

- Scatterplot 1: FlightNumber vs. PayloadMass
- Scatterplot 2: Flight Number vs. Launch Site
- Scatterplot 3: LaunchSite vs PayloadMass
- Scatterplot 4: FlightNumber vs Orbit Type
- Scatterplot 5: Payload vs. Orbit Type

02 Bar chart (for comparison)

- To find out success rate of each orbit and which orbits have high success rates

03 Line chart (to observe changes over time)

- Success rate vs year

- GitHub URL of completed EDA with data visualization notebook: https://github.com/yttoh/datascicapstone/blob/main/edadataviz.ipynb

EDA with SQL

SQL Queries

1. Display the names of the unique launch sites in the space mission
2. Display 5 records where launch sites begin with the string 'CCA'
3. Display the total payload mass carried by boosters launched by NASA (CRS)
4. Display average payload mass carried by booster version F9 v1.1
5. List the date when the first successful landing outcome in ground pad was achieved.
6. List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
7. List the total number of successful and failure mission outcomes
8. List the names of the booster_versions which have carried the maximum payload mass.
9. List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
10. 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

GitHub URL of completed EDA with SQL notebook:
https://github.com/yttoh/datascicapstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

1. Created a folium Map object, with an initial center location to be NASA Johnson Space Center at Houston, Texas.
2. Create and add folium.Circle and folium.Marker for each launch site on the site map
3. Create markers for all launch records. If a launch was successful (class=1), then we use a green marker and if a launch was failed, we use a red marker (class=0)
4. Mark launch site proximity to nearest coastline, highway, etc

GitHub URL of completed interactive map with Folium map:
https://github.com/yttoh/datascicapstone/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Added a pie chart to display number of successful launches by different sites
- Added a sliding scatterplot display illustrating payload vs outcome.
- These graphs help to illustrate SpaceX Launch Records visually

Github URL: https://github.com/yttoh/datascicapstone/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

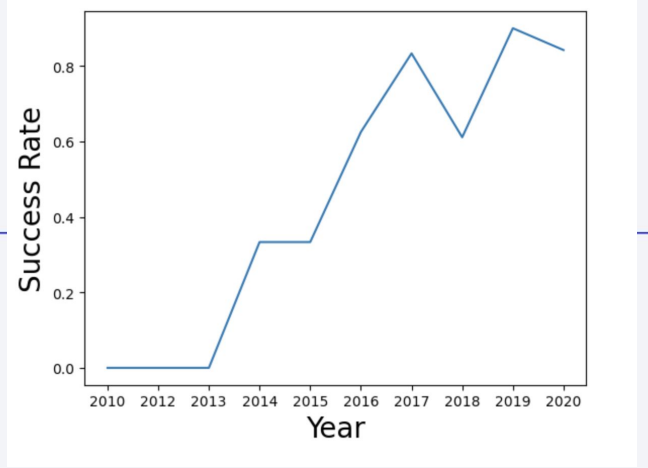
- Split the dataset into training and testing dataset
- Evaluate the training and test sets with machine learning models:
  - Logistic Regression
  - Support Vector Machine (SVM)
  - Decision Tree Classifier
  - K Nearest Neighbour (KNN)
- For all the models, hyperparameters are tuned to improve model performance and accuracy

Github URL: https://github.com/yttoh/datascicapstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

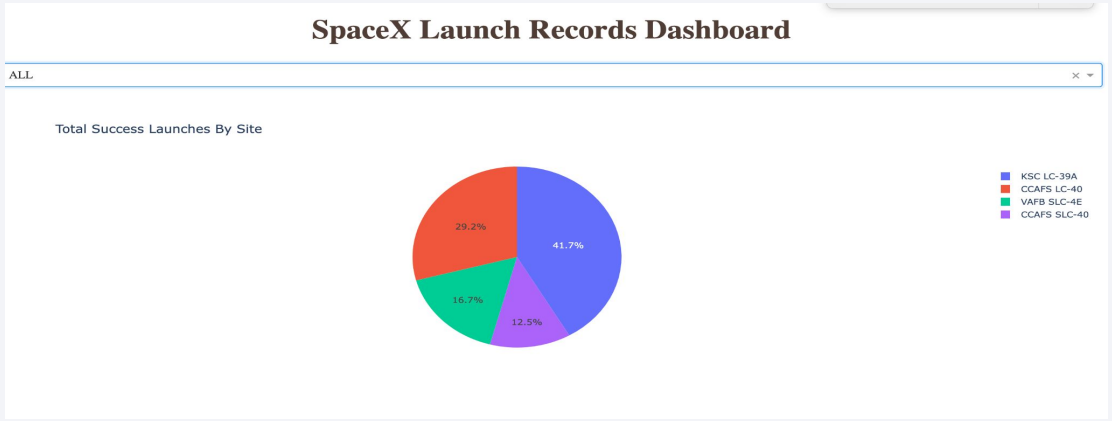
Exploratory data analysis results

- Success rate of launches has generally increased over the years



Interactive analytics demo in screenshots

- Plotly dashboard
- Folium map



Predictive analysis results

- Inconclusive on effectiveness of models due to small dataset

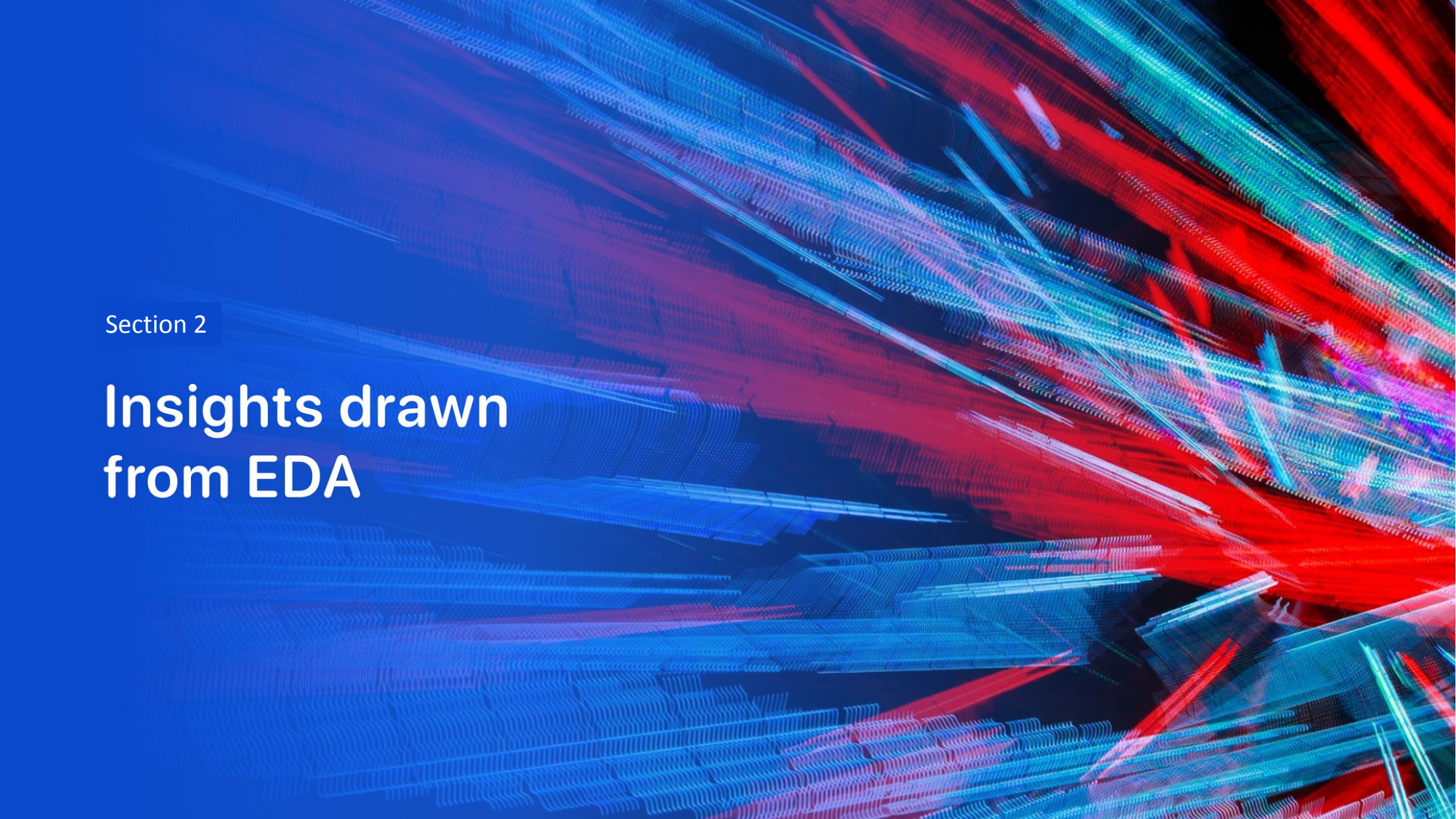
Find the method performs best:

```
print('Accuracy for Logistic Regression Method', logreg_cv.score(X_test,Y_test))
print('Accuracy for Support Vector Machine Method', svm_cv.score(X_test,Y_test))
print('Accuracy for Decision Tree Method', tree_cv.score(X_test,Y_test))
print('Accuracy for K nearest neighbour Method', knn_cv.score(X_test,Y_test))
```

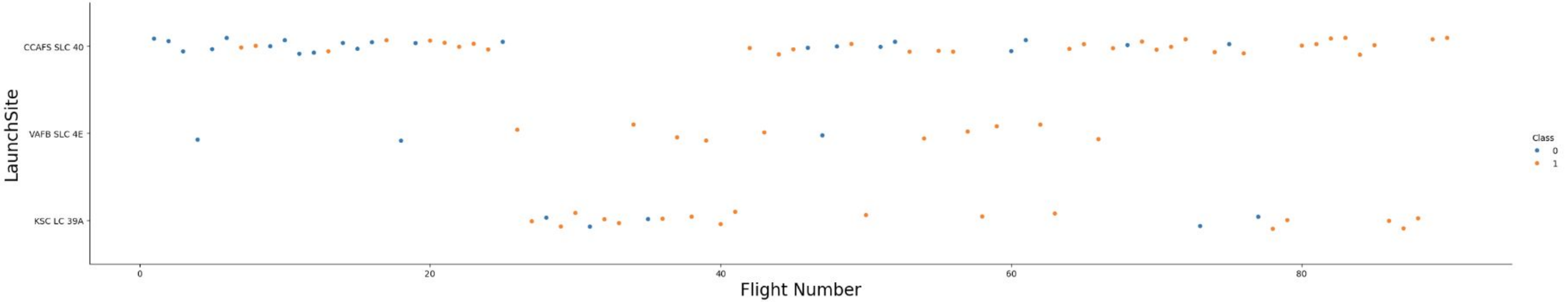
```
Accuracy for Logistic Regression Method 0.8333333333333334
Accuracy for Support Vector Machine Method 0.8333333333333334
Accuracy for Decision Tree Method 0.8333333333333334
Accuracy for K nearest neighbour Method 0.8333333333333334
```

Section 2

Insights drawn from EDA

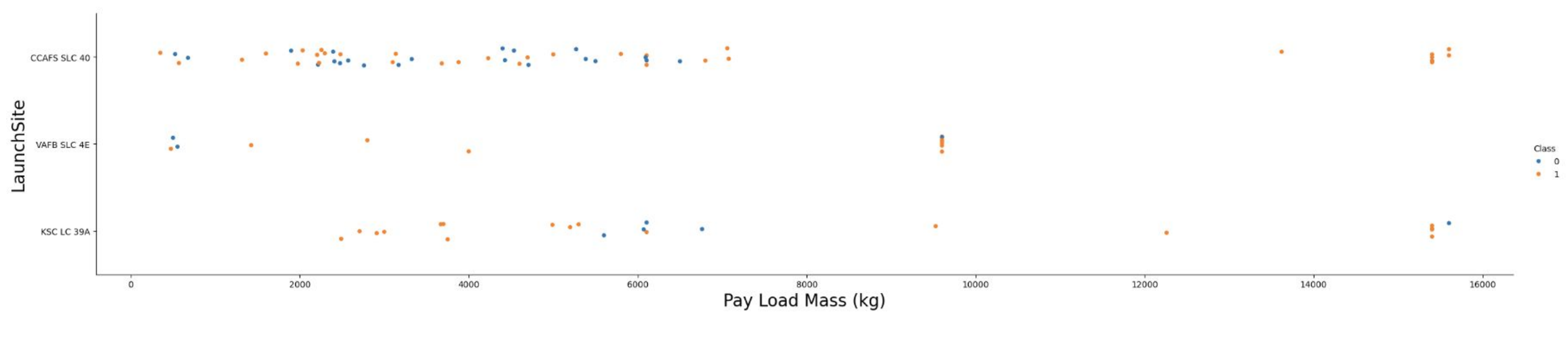


Flight Number vs. Launch Site



- CCAFS SLC 40 is the site with the most number of site launches while VAFB SLC 4E is the site with the least number of site launches.
- However, despite having the least number of launches, it seems that VAFB SLC 4E has a greater proportion of successful launches, followed by the site KSC LC39A.

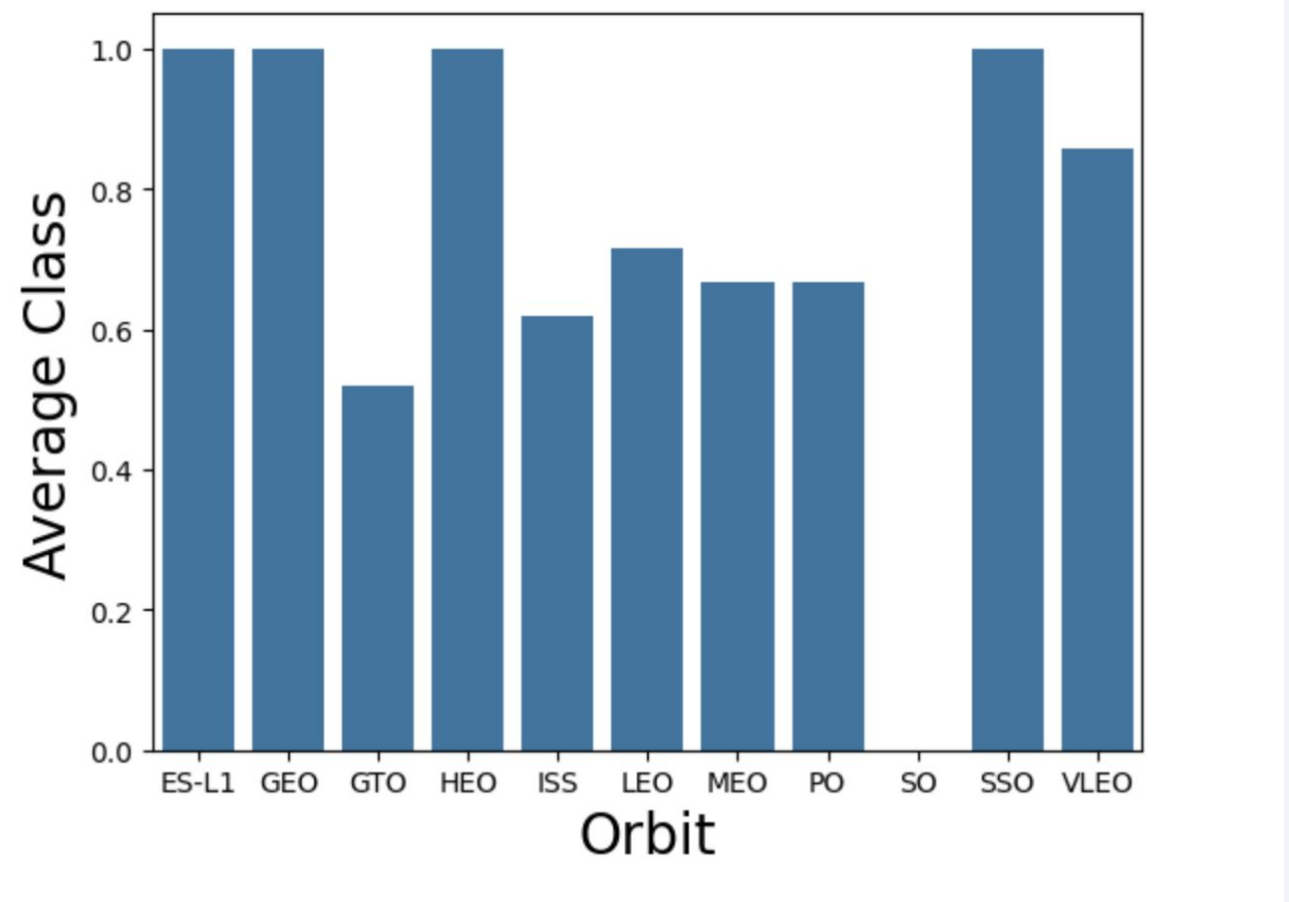
Payload vs. Launch Site



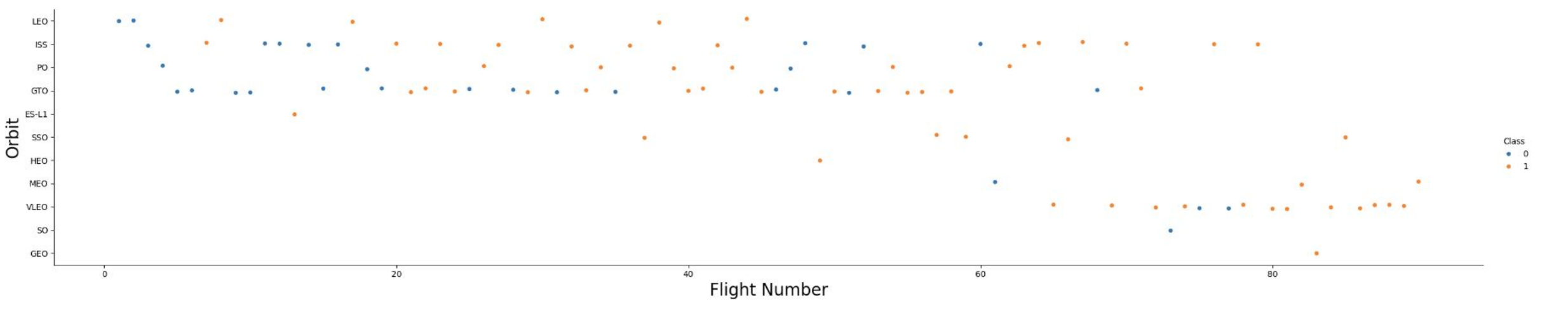
- CCAFS SLC 40 mostly have launches with a lower payload (less than 8000kg), its higher payload launches (<13000kg) have also been successful launches
- For VAFB-SLC 4E, there are no rockets launched for heavy payload mass(greater than 10000).
- KSC LC39A is most successful with launches below 5000kg payload

Success Rate vs. Orbit Type

- Orbit type ES-L1, GEO, HEO and SSO have the highest success rate (100%).
- Orbit GTO has the lowest success rate (50%).

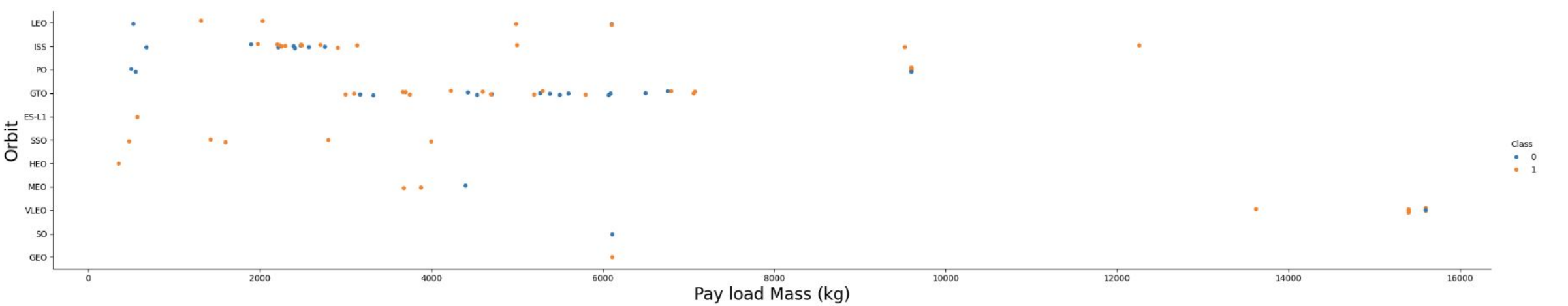


Flight Number vs. Orbit Type



- In the LEO orbit the success appears related to the number of flights, on the other hand, there seems to be no relationship between flight number when in GTO orbit, which is the orbit with the lowest success rate (50%).
- For ES-L1 and GEO orbits, they only had one launch and it was successful so their high success launch rate (100%) may be overgeneralised.
- Overall, SSO is the orbit with the best performance (flight number to success rate ratio).

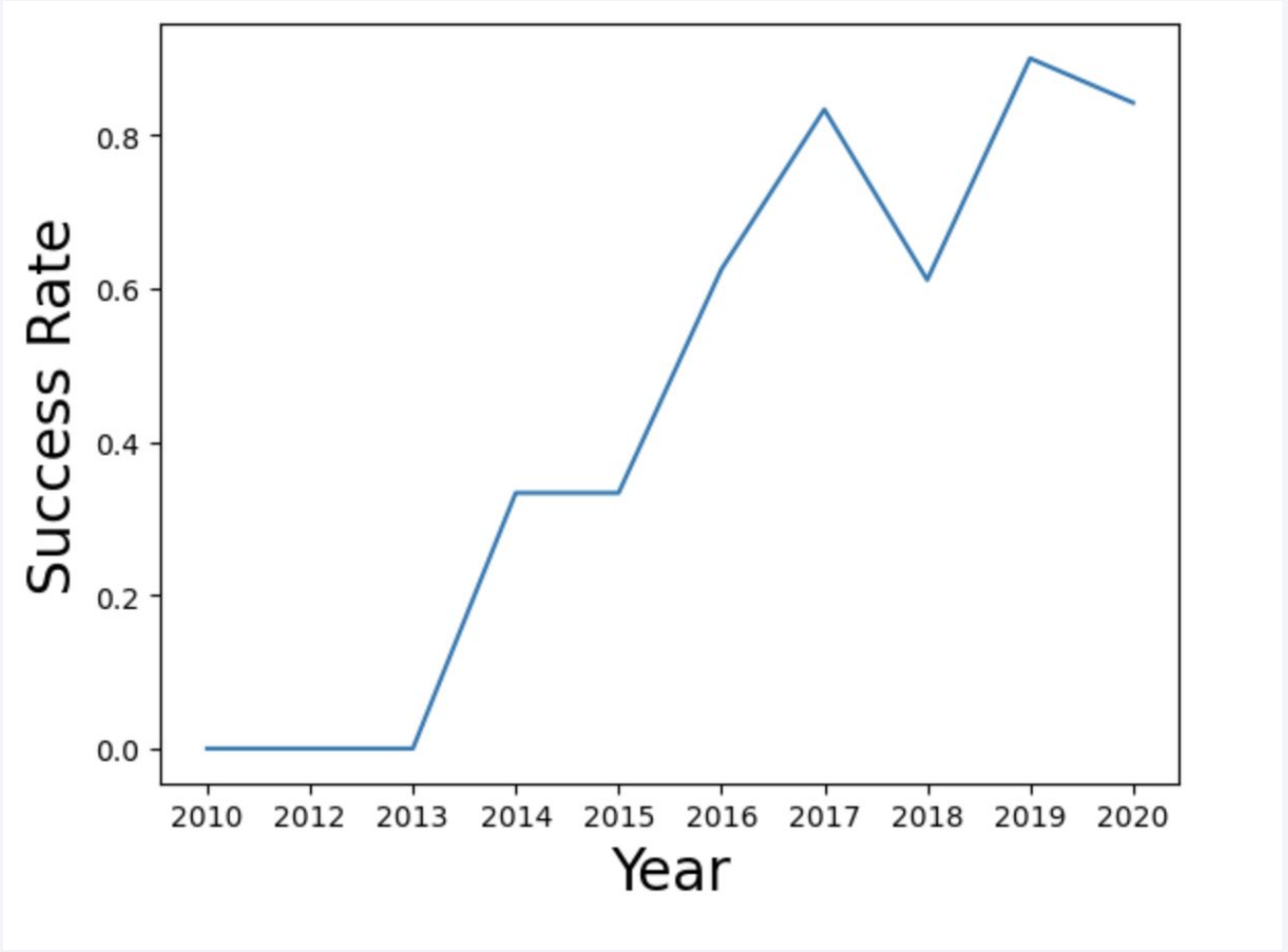
Payload vs. Orbit Type



- With heavy payload, the successful landing rate are more for VLEO, LEO and ISS.
- For lower payload (<=4000kg), SSO has the best success landing rate (100%).
- However for ISS and GTO, we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend

- Success rate generally increased over the past decade from 2010 to 2020 (increasing trend with some dips)
- From 2010 to 2013, the success rate was 0, which meant either there were no launches or all the launches had failed.



All Launch Site Names

Names of the unique launch sites:

```
In [8]: %sql Select distinct ("Launch_Site") from SPACEXTBL
 * sqlite:///my_data1.db
Done.
```

Out[8]:

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

Launch Site Names Begin with 'CCA'

5 records where launch sites begin with `CCA`:

```
%sql SELECT "Launch_Site" FROM SPACEXTBL WHERE "Launch_Site" LIKE '%CCA%' LIMIT 5
```

```
 * sqlite:///my_data1.db
Done.
```

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

Calculate the total payload carried by boosters from NASA:

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

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL
```

```
 * sqlite:///my_data1.db
Done.
```

SUM("PAYLOAD_MASS__KG_")
619967

Average Payload Mass by F9 v1.1

Calculate the average payload mass carried by booster version F9 v1.1:

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

```
%sql SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL where "Booster_Version" = "F9 v1.1"
```

```
 * sqlite:///my_data1.db
Done.
```

AVG("PAYLOAD_MASS__KG_")
2928.4

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad:

```
%sql Select MIN("Date") from SPACEXTBL where "Landing_Outcome" like '%Success%'
```

```
 * sqlite:///my_data1.db
Done.
```

MIN("Date")
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000:

%sql SELECT ("Booster_Version") from SPACEXTBL where "Landing_Outcome" = "Success (drone ship)" and "PAYLOAD_MASS__KG_" between 4000 and 60000

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1029.1
F9 FT B1021.2
F9 FT B1036.1
F9 B4 B1041.1
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

Calculate the total number of successful and failure mission outcomes:

```
%sql select Mission_Outcome, count(*) as Total from SPACEXTBL group by Mission_Outcome
```

```
 * sqlite:///my_data1.db
Done.
```

Mission_Outcome	Total
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

List the names of the booster which have carried the maximum payload mass:

```
%sql select ("Booster_Version") from SPACEXTBL where "PAYLOAD_MASS__KG_" = (select Max("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

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

2015 Launch Records

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015:

```
%%sql
SELECT SUBSTR("DATE",6,2) AS MONTH, "BOOSTER_VERSION", "LAUNCH_SITE", "LANDING_OUTCOME"
FROM SPACEXTBL
WHERE "LANDING_OUTCOME" = 'Failure (drone ship)' AND SUBSTR("DATE",0,5) = '2015';
```

```
 * sqlite:///my_data1.db
Done.
```

MONTH	Booster_Version	Launch_Site	Landing_Outcome
01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

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:

%sql select Landing_Outcome, Date from SPACEXTBL where Date between '2010-06-04' and '2017-03-20' order by Date desc

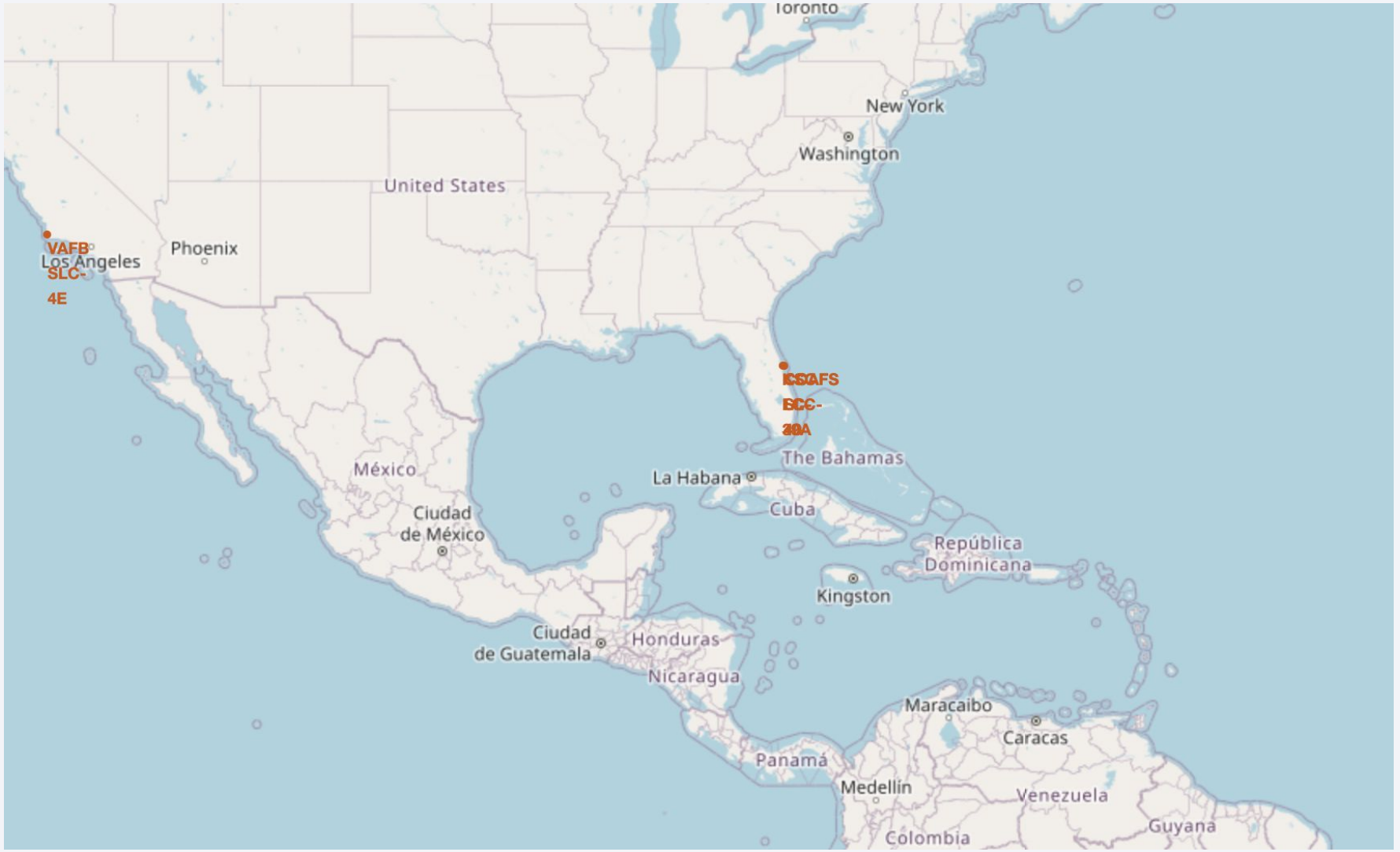
Landing_Outcome	Date
No attempt	2017-03-16
Success (ground pad)	2017-02-19
Success (drone ship)	2017-01-14
Success (drone ship)	2016-08-14
Success (ground pad)	2016-07-18
Failure (drone ship)	2016-06-15
Success (drone ship)	2016-05-27
Success (drone ship)	2016-05-06
Success (drone ship)	2016-04-08
Failure (drone ship)	2016-03-04
Failure (drone ship)	2016-01-17
Success (ground pad)	2015-12-22
Precluded (drone ship)	2015-06-28
No attempt	2015-04-27
Failure (drone ship)	2015-04-14
No attempt	2015-03-02
Controlled (ocean)	2015-02-11
Failure (drone ship)	2015-01-10

(continued)

Uncontrolled (ocean)	2014-09-21
No attempt	2014-09-07
No attempt	2014-08-05
Controlled (ocean)	2014-07-14
Controlled (ocean)	2014-04-18
No attempt	2014-01-06
No attempt	2013-12-03
Uncontrolled (ocean)	2013-09-29
No attempt	2013-03-01
No attempt	2012-10-08
No attempt	2012-05-22
Failure (parachute)	2010-12-08
Failure (parachute)	2010-06-04


Section 3
Launch Sites
Proximities Analysis

Launch site locations on map

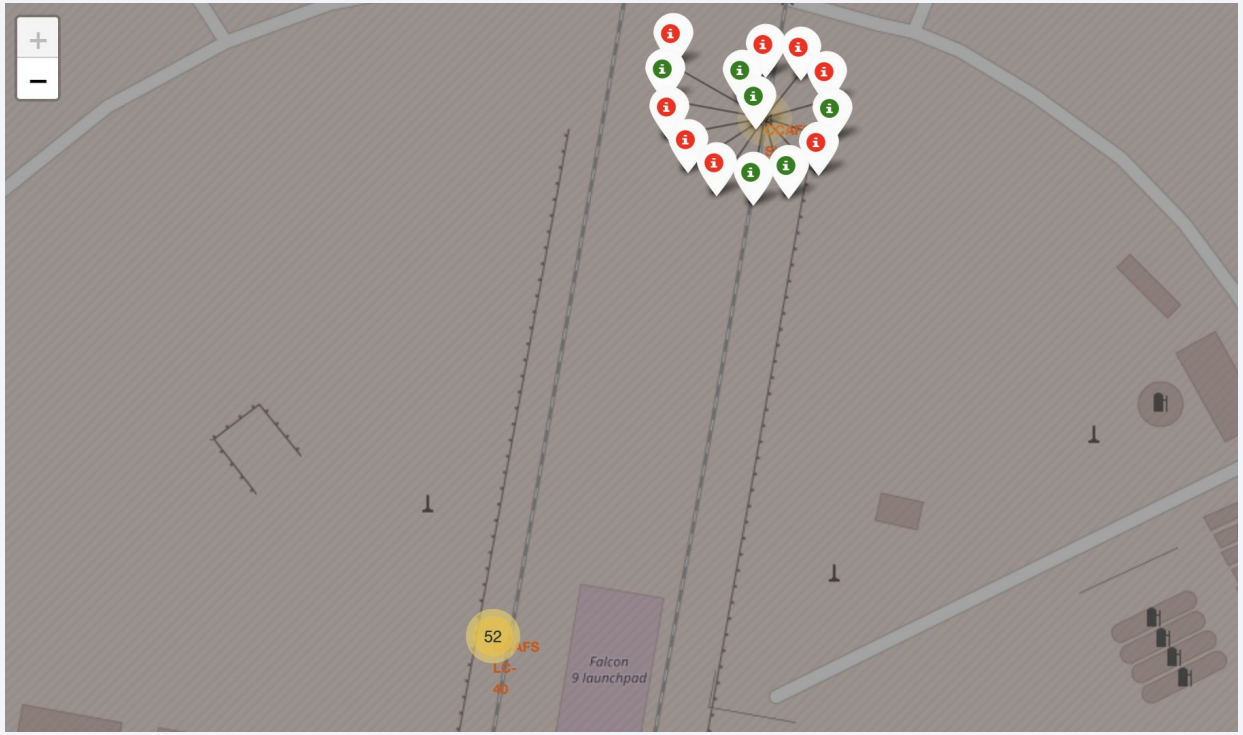


Launch sites are near to the ocean

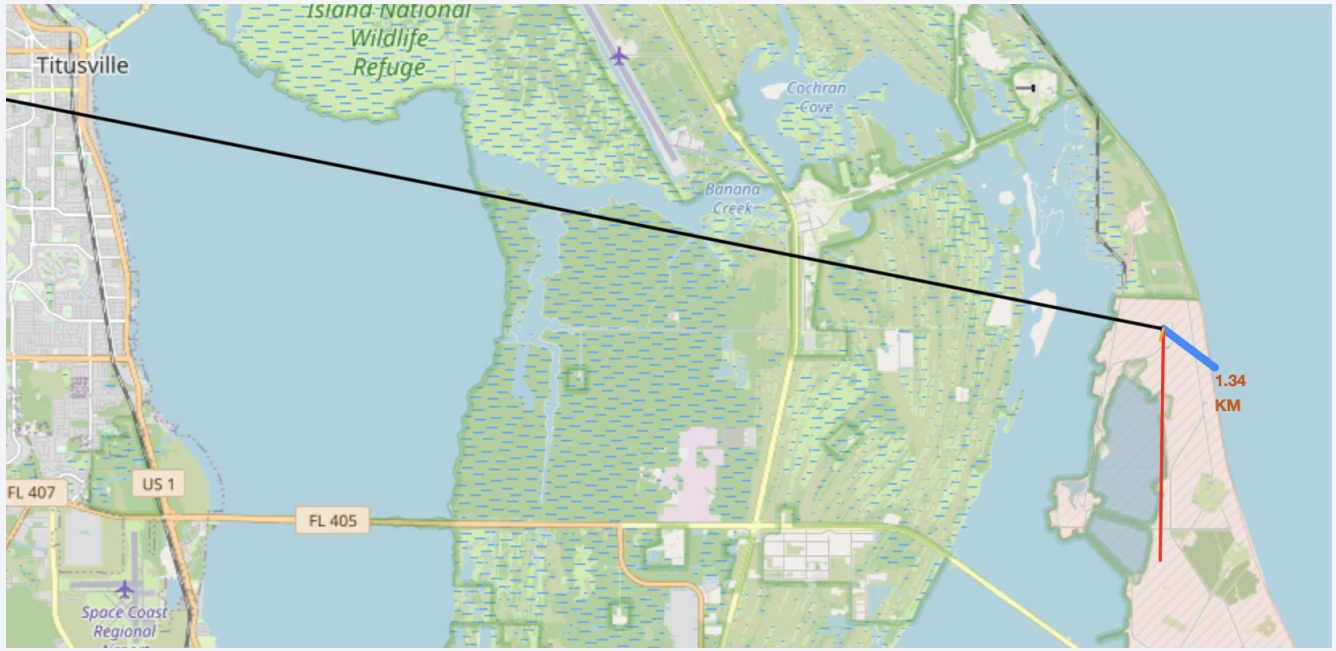
Color-labelled launch outcomes

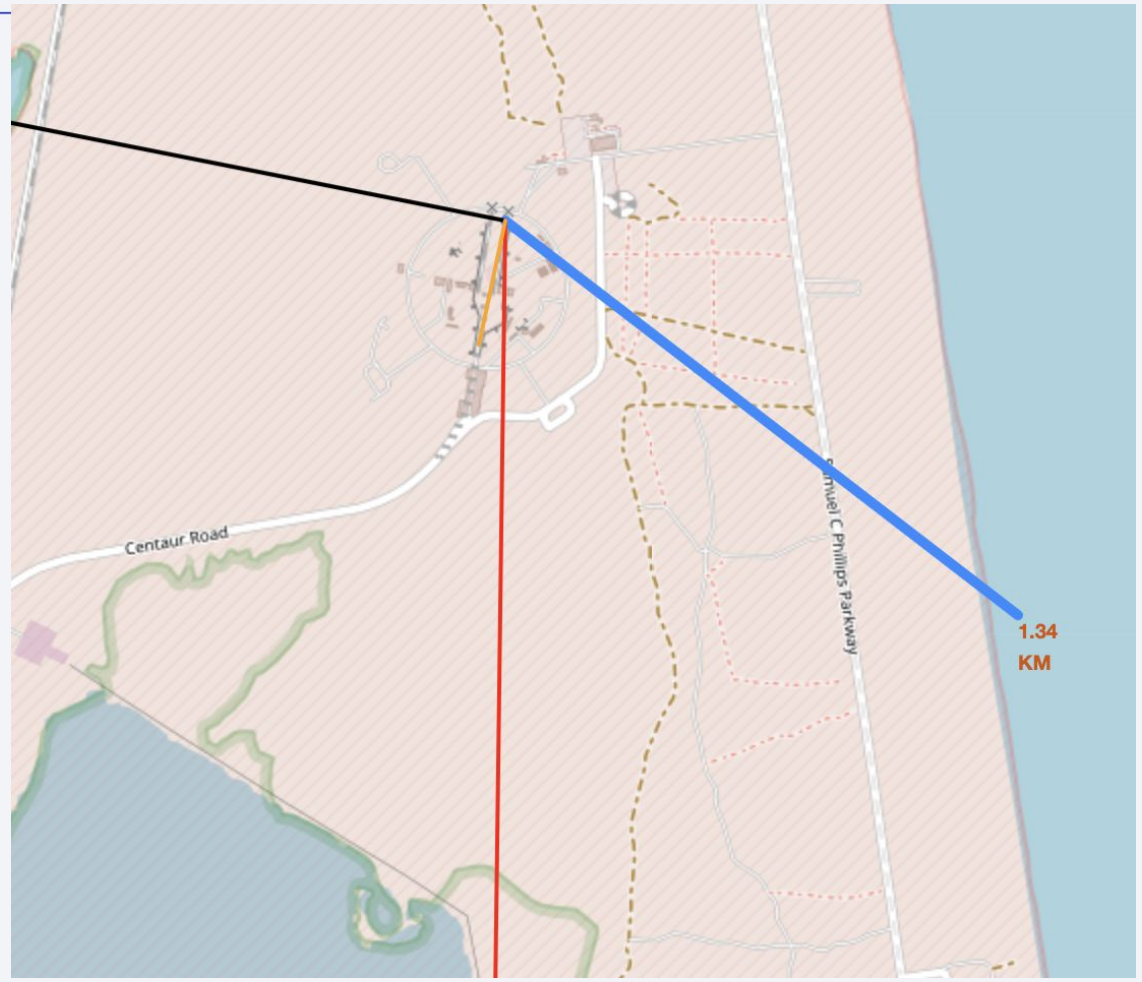
At CCAFS SLC-40:

- Green label: Successful launch
- Red label: Failed launch

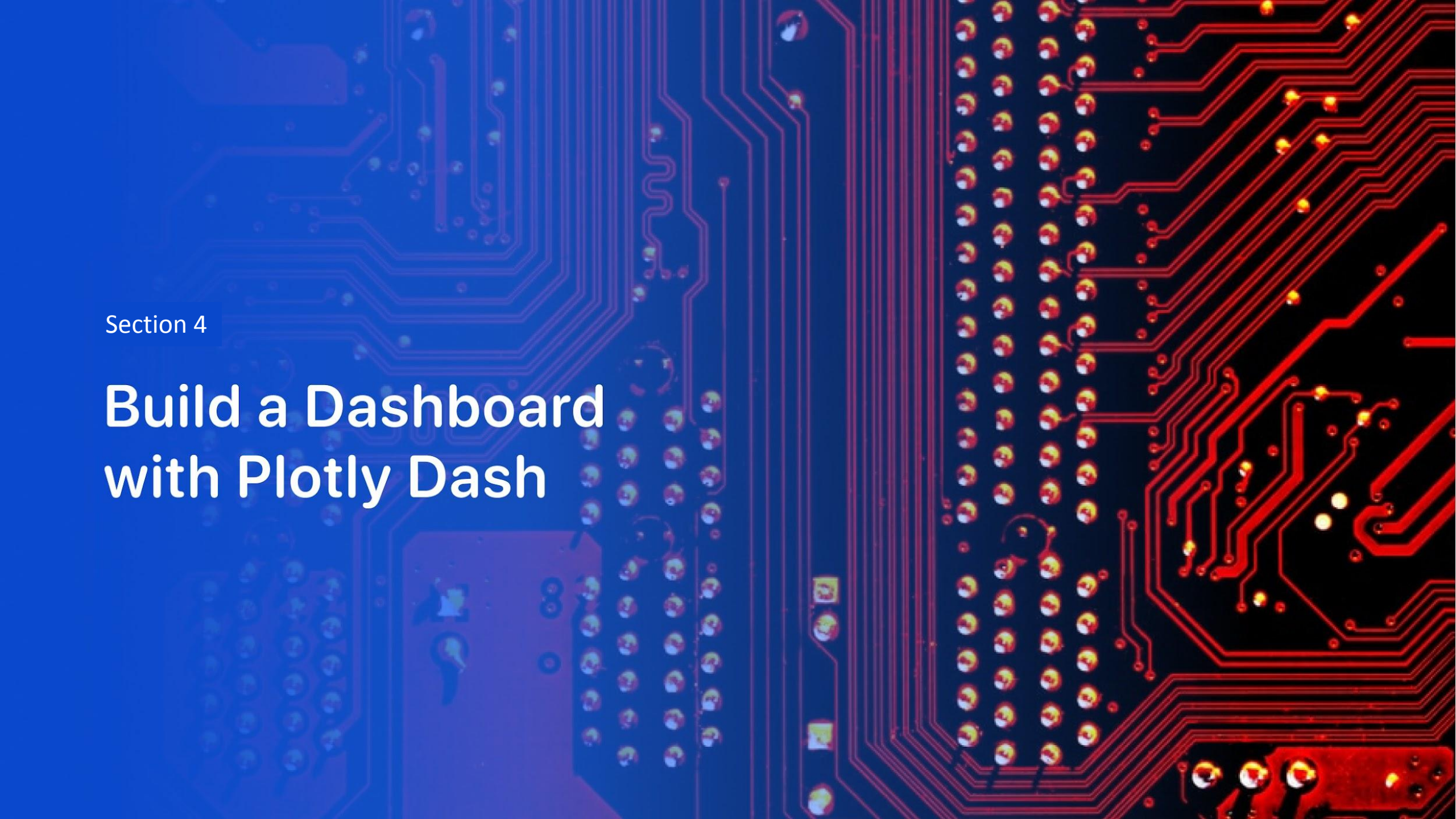


Launch site with proximities to railway, highway, coastline

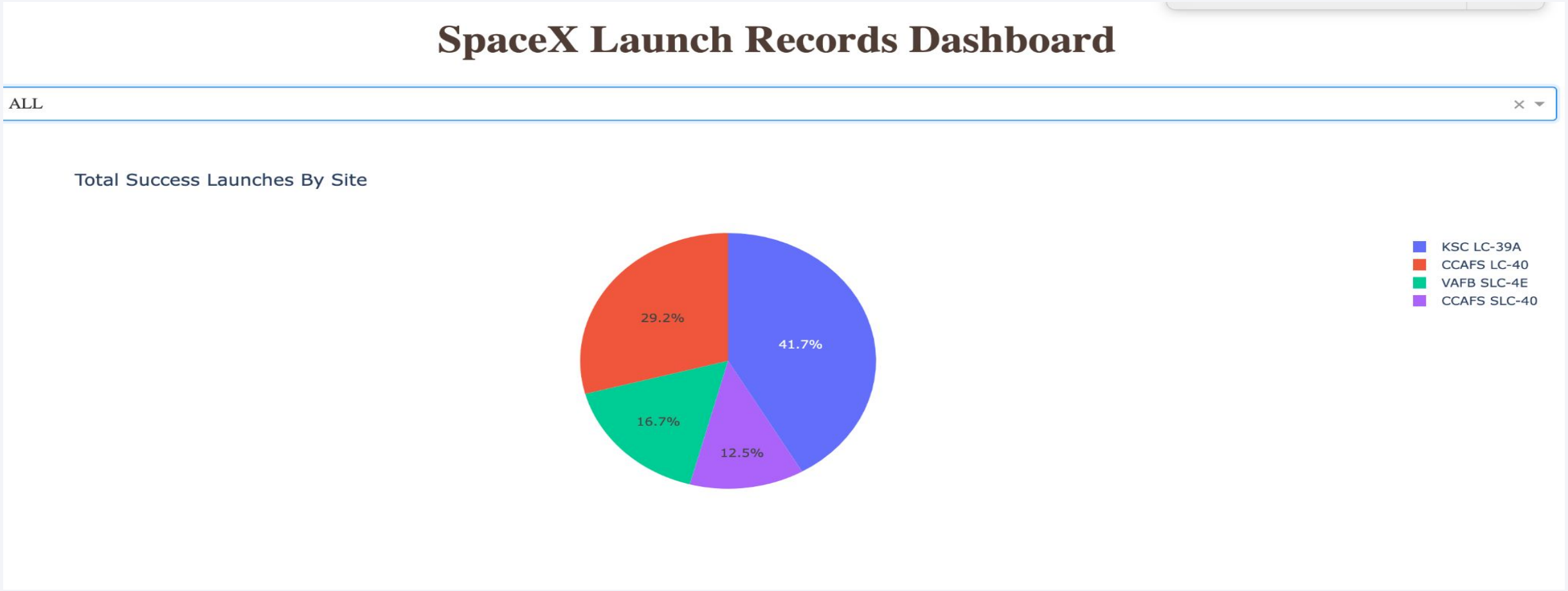




- Blue line: proximity to coastline
- Orange line: proximity to railway
- Red line: proximity to highway
- Black line: proximity to nearest city

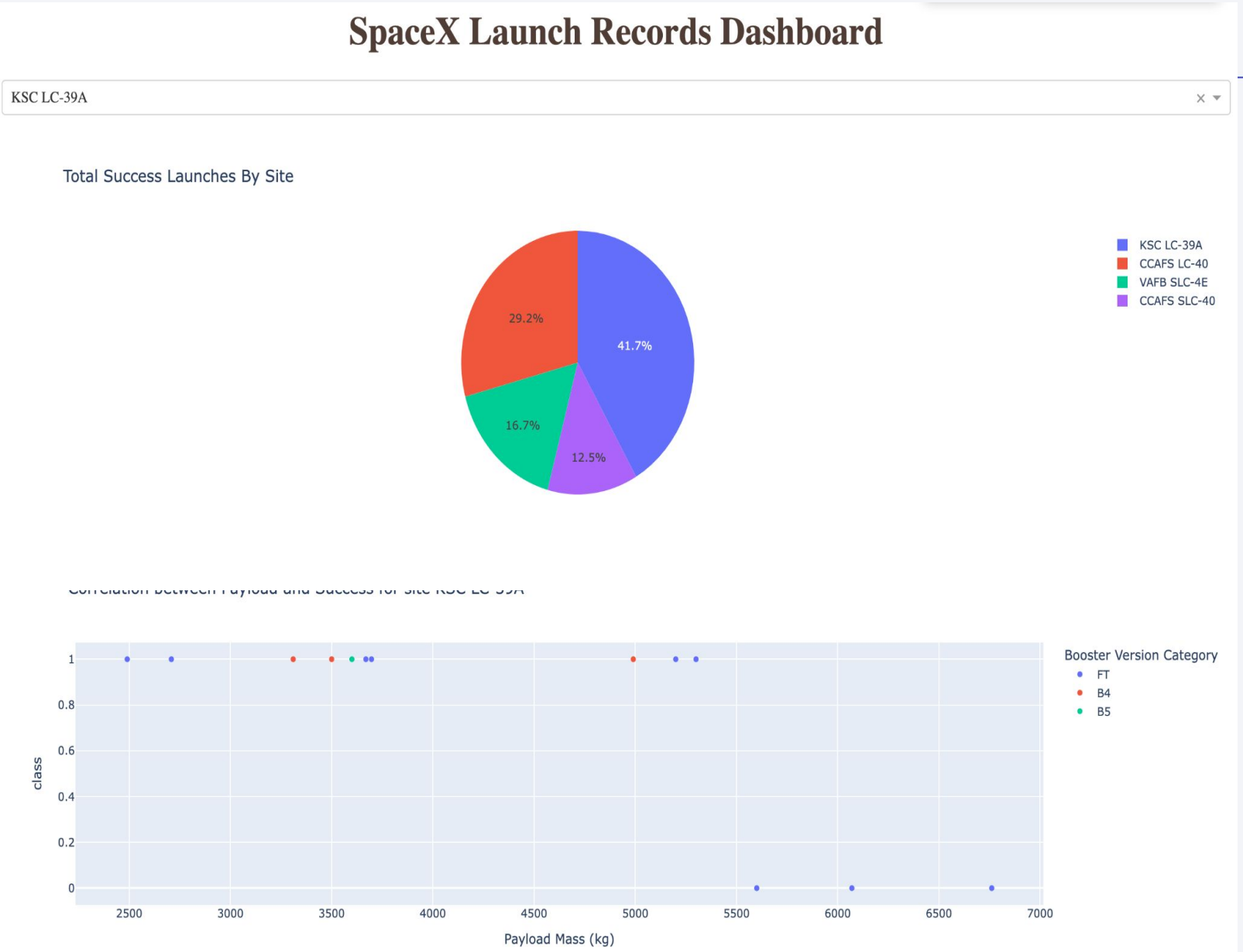

Section 4
Build a Dashboard with Plotly Dash

Number of Successful Launches by Site



- KSC LC-39A has the highest number of successful launches while CCAFS SLC-40 has the least number of successful launches.

Launch Success for KSC LC-39A



- Three higher payload (>5500kg) launches failed for the site KSC LC-39A.
- Launches below 5500kg are all successful.
- B4 and B5 Booster Version have 100% successful launches while FT Booster version only has 33.3% successful launches.

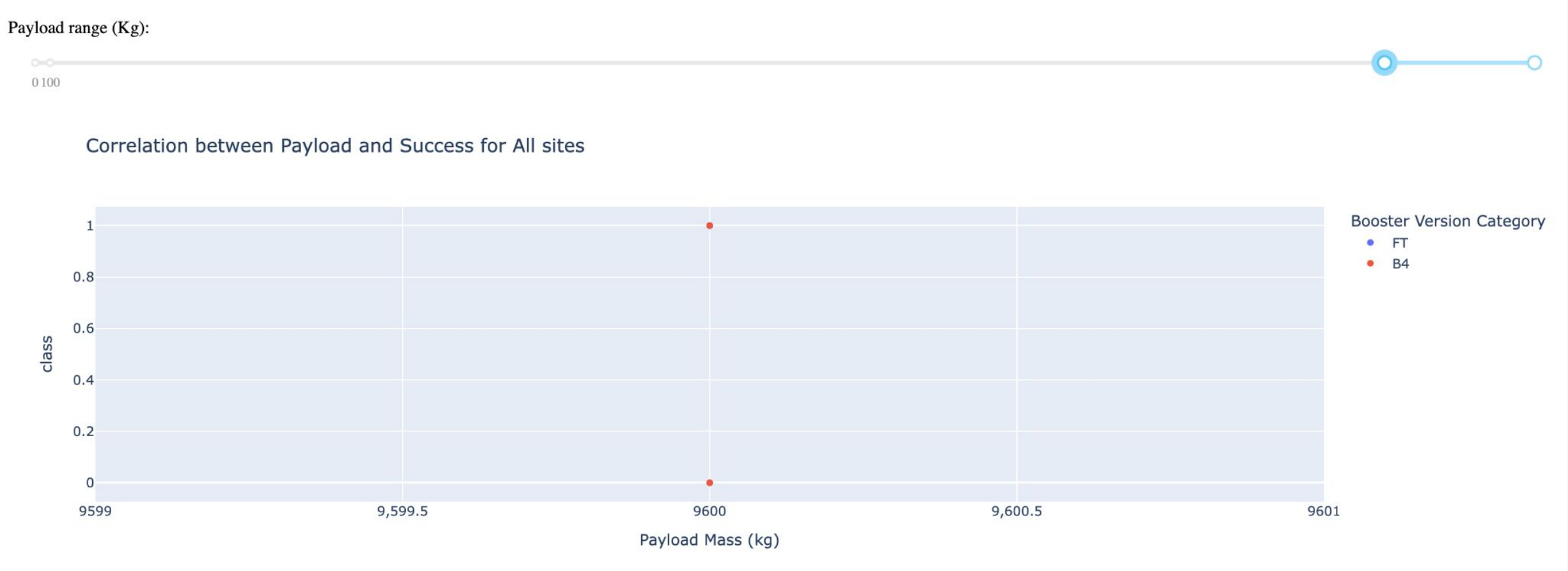
Payload vs Launch Outcomes for all sites

At low payload mass range (less than 5000kg), FT booster version has the best performance while v1.1 booster version has the least successful launches.

Payload vs Launch Outcomes for all sites

At the mid payload range (>5000kg and less than 7000kg), FT Booster Version has some successful launches (less than 50% success rate) while B4 Booster Version has no successful launches at all.

Payload vs Launch Outcomes for all sites



At the high payload range (9600kg), only B4 Booster Version has launch attempts. Its success rate is 50%.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

Find the method performs best:

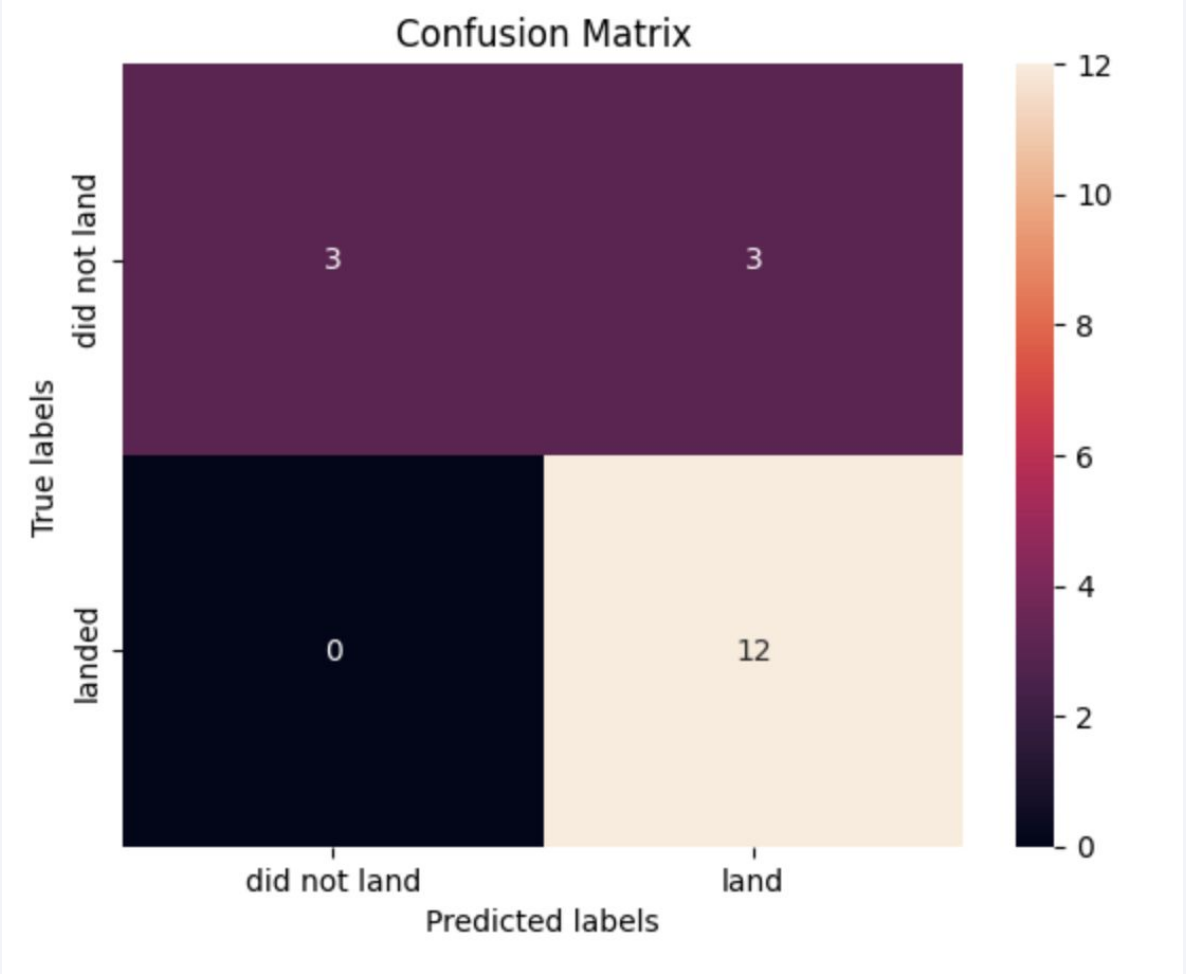
```
print('Accuracy for Logistic Regression Method', logreg_cv.score(X_test,Y_test))
print('Accuracy for Support Vector Machine Method', svm_cv.score(X_test,Y_test))
print('Accuracy for Decision Tree Method', tree_cv.score(X_test,Y_test))
print('Accuracy for K nearest neighbour Method', knn_cv.score(X_test,Y_test))
```

```
Accuracy for Logistic Regression Method 0.8333333333333334
Accuracy for Support Vector Machine Method 0.8333333333333334
Accuracy for Decision Tree Method 0.8333333333333334
Accuracy for K nearest neighbour Method 0.8333333333333334
```

- The four models (Logistic Regression, SVM, Decision Tree and KNN) all have the same accuracy (83.3%)
- The results are the same likely due to small dataset and lesser values

Confusion Matrix

- All four models have the same confusion matrix (ie. they all have the same accuracy due to small dataset and lesser values)



Conclusions

1. Launch sites success rate has generally increased over the years
2. It is harder to get successful launch for high payload mass.
3. Dataset is too small for model training so there is no best model. The four models used (Logistic Regression, Decision Tree, KNN, SVM all showed the same accuracy).
4. KSC LC-39A is the launch site with the highest success rate (41.7%). SpaceX could possibly look into the factors on why this is so.

Thank you!