IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Kusum Subedi
01/05/2023

Outline

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

Introduction

- Many Commercial Companies Aim to make space travel affordable for everyone
  - The Best one is SPACEX Falcon 9
- Claims it only cost about 62 million dollars per person, a lot cheaper than other company

BUT HOW??

SPACEX can reuse the first stage, the most expensive stage

QUESTION FOR THIS REPORT

WHAT IS THE CHANCE IF THE FIRST STAGE LANDS SUCCESSFULLY?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

  Request to the SpaceX API using get request

  HTTP GET method to request the Falcon9 Launch HTML page(Wikipedia), as an HTTP response.

- Perform data wrangling

  Filtered the data to include only Falcon 9 launches

- 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, KNN, SVM, Decision Tree with cv=10

  Compared the accuracy for each model.

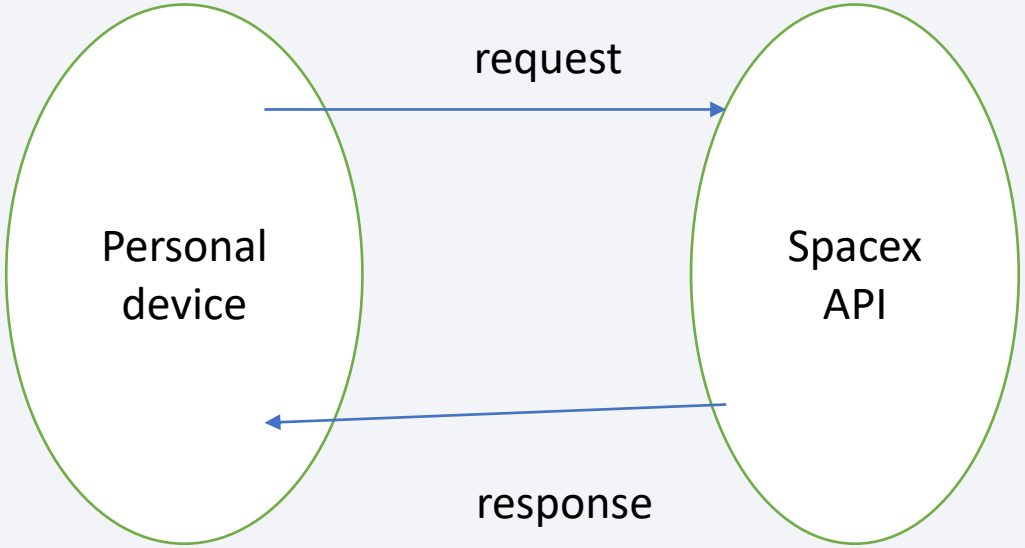
Data Collection

- From SPACEXAPI
- Webscraping from Wikipedia

NEXT SLIDES please

Data Collection – SpaceX API

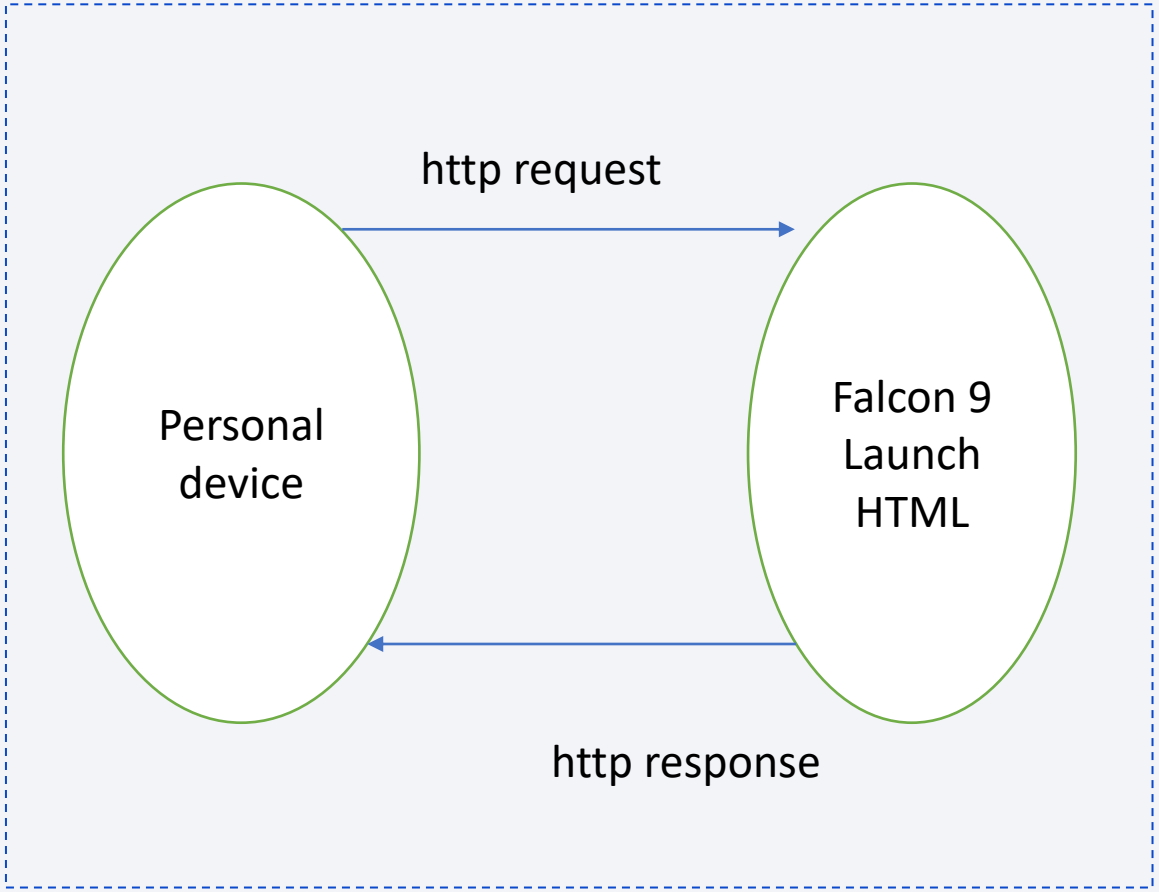
- spacex_url="https://api.spacexdata.com/v4/launches/past"
- https://github.com/skusum/Coursera-Capstone-Project-Kusum/blob/main/Lab1_dataCollectionAPI.ipynb



Data Collection - Scraping

- https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

- https://github.com/skusum/Coursera-Capstone-Project-Kusum/blob/main/Lab2_webscraping.ipynb



Data Wrangling:

https://github.com/skusum/Coursera-Capstone-Project-Kusum/blob/main/Lab3_Data%20wrangling.ipynb

Data from web scraping

Parse http response using BeautifulSoup

Create a dictionary with keys as required features

Create pd dataframe using pandas

Data from API call

Normalize the response data which is in json format
Using pd.json_normalize(response.json())

Crteate a dictionary with keys as required features

Create pd dataframe using pandas

Filter the dataframe so as to include data for Falcon 9 only

Fill the missing values with the mean of corresponding column

Find the datatypes

In both cases successful and failure were replace as class = 1 and 0 respectively

EDA with Data Visualization

https://github.com/skusum/Coursera-Capstone-Project-Kusum/blob/main/Lab5_EDA_dataviz.ipynb

Various scatter plots, line plot and bar chart were drawn for eg:

- Flight number vs PayloadMass (catplot with hue = class using seaborn)
- Flight number vs orbit type (catplot with hue = class using seaborn)
- Flight number vs Launch Site (catplot with hue = class using seaborn)
- Payload vs Orbit type (catplot with hue = class using seaborn)
- Orbit type vs SuccessRate (Bar chart)
- Success yearly rate (Line plot)

EDA with SQL

https://github.com/skusum/Coursera-Capstone-Project-Kusum/blob/main/Lab4_EDA_SQL.ipynb

Following queries were performed:

- names of the unique launch sites
- ***5 records where launch sites begin with the string 'CCA'***
- ***the total payload mass carried by boosters launched by NASA (CRS)***
- ***average payload mass carried by booster version F9 v1.1***
- ***List the date when the first succesful landing outcome in ground pad was acheived.***
- ***the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000***
- ***the total number of successful and failure mission outcomes***
- ***the names of the booster_versions which have carried the maximum payload mass.***
- ***the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.***
- ***Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.***

Build an Interactive Map with Folium

https://github.com/skusum/Coursera-Capstone-Project-Kusum/blob/main/Lab6_launch_site_location_Folium.ipynb

The following tasks were performed

- Circle markers at Launch Sites with the names displayed just to see where they are
- Created marker clusters on each Launch sites with the cluster color indicating successful (green) and failed(red) landing
- Line connecting Launch Sites with its proximity showing the distance

Build a Dashboard with Plotly Dash

https://github.com/skusum/Coursera-Capstone-Project-Kusum/blob/main/Lab8_spacex_dash_app%20.py

Plotly dashboard was created with the following Features

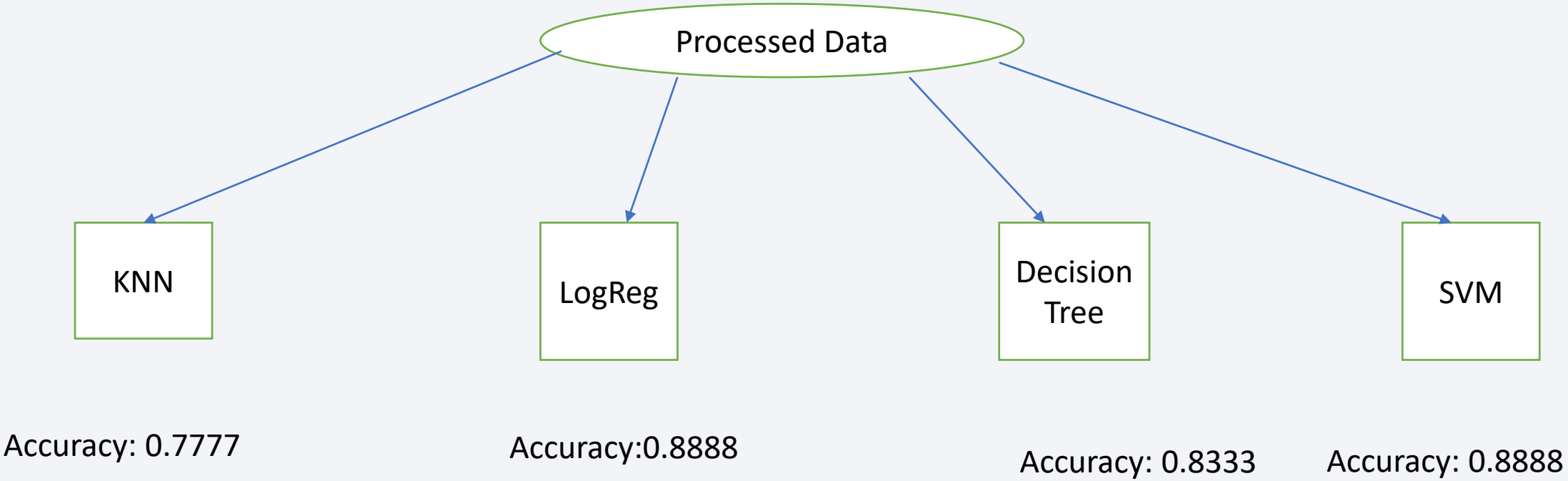
- A dropdown list with the option for each site as well as all the sites.
- Pie chart object.
- Range slider for payload Mass
- Scatter plots

When the user selects a site it shows corresponding piechart with success and failures.

When the user pick a value on Range slider and a site from the dropdown list, It displays scatter plot of successes and failures for corresponding site.

Predictive Analysis (Classification)

https://github.com/skusum/Coursera-Capstone-Project-Kusum/blob/main/Lab7_SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb



Results

- Interactive analytics demo in screenshots

- Predictive analysis results

```
logreg_score = logreg_cv.score(X_test, Y_test)
logreg_score
```

```
0.8888888888888888
```

```
svm_score = svm_cv.score(X_test, Y_test)
svm_score
```

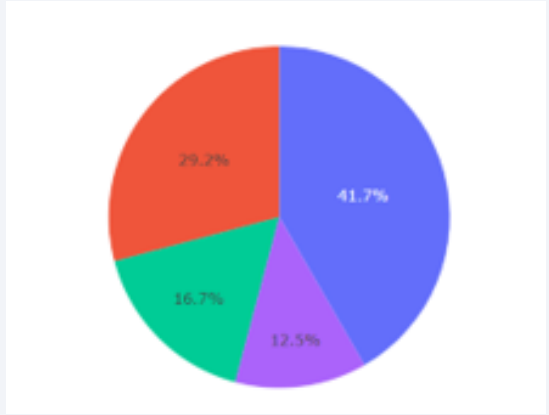
```
0.8888888888888888
```

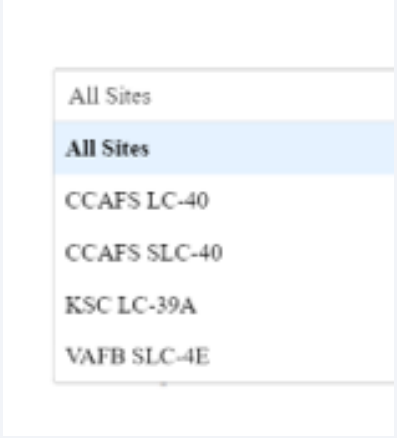
```
tree_score = tree_cv.score(X_test, Y_test)
tree_score
```

```
0.8333333333333334
```

```
knn_score = knn_cv.score(X_test, Y_test)
knn_score
```

```
0.7777777777777778
```





EDA Results (continued)

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

```
FlightNumber      0.000
Date              0.000
BoosterVersion    0.000
PayloadMass       0.000
Orbit             0.000
LaunchSite        0.000
Outcome           0.000
Flights           0.000
GridFins          0.000
Reused            0.000
Legs              0.000
LandingPad       40.625
Block             0.000
ReusedCount       0.000
Serial            0.000
Longitude         0.000
Latitude          0.000
dtype: float64
```

```
df.dtypes
```

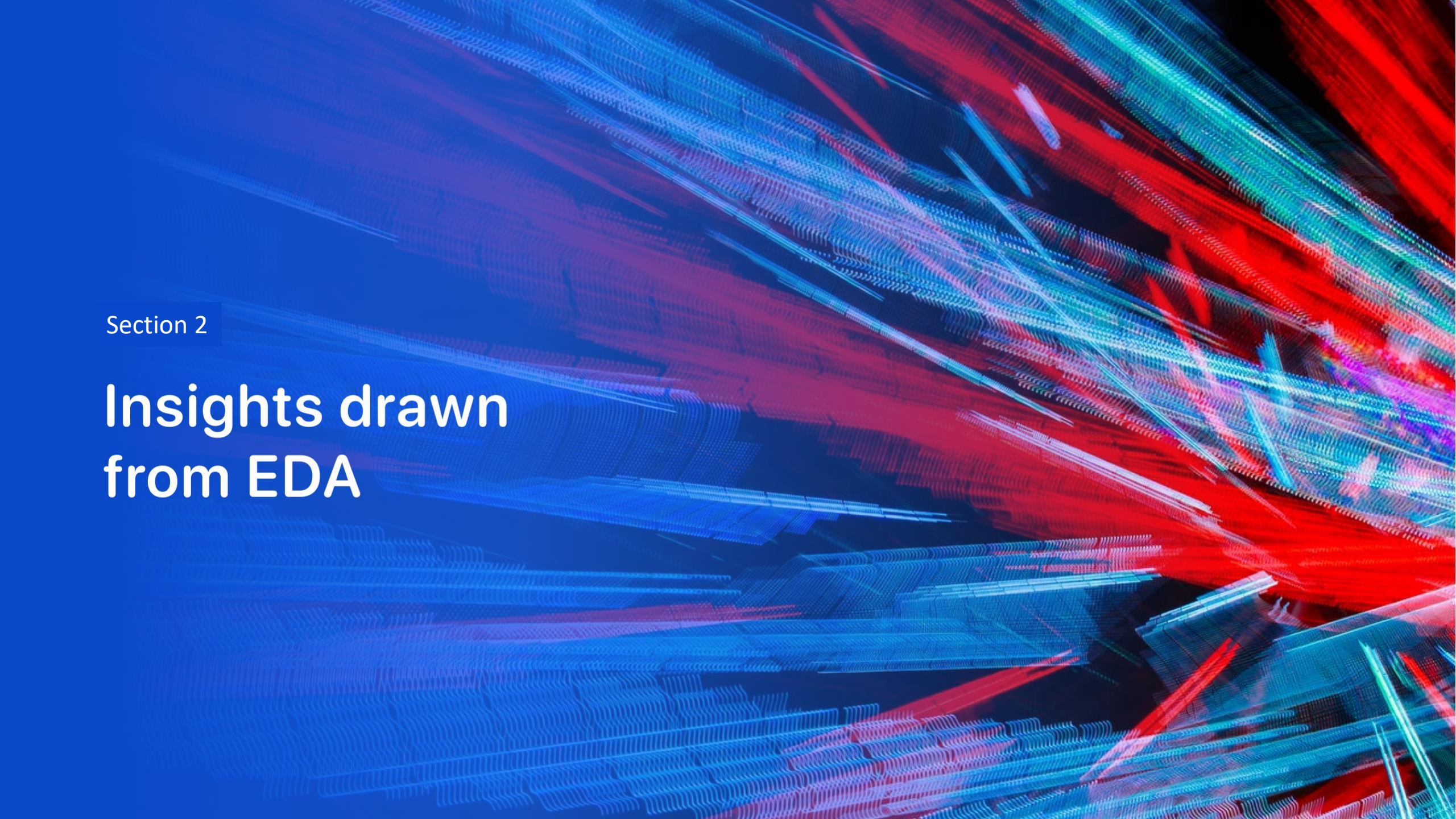
```
FlightNumber      int64
Date             object
BoosterVersion   object
PayloadMass     float64
Orbit            object
LaunchSite       object
Outcome          object
Flights           int64
GridFins           bool
Reused             bool
Legs               bool
LandingPad       object
Block           float64
ReusedCount       int64
Serial           object
Longitude       float64
Latitude        float64
dtype: object
```

```
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
Name: Orbit, dtype: int64
```

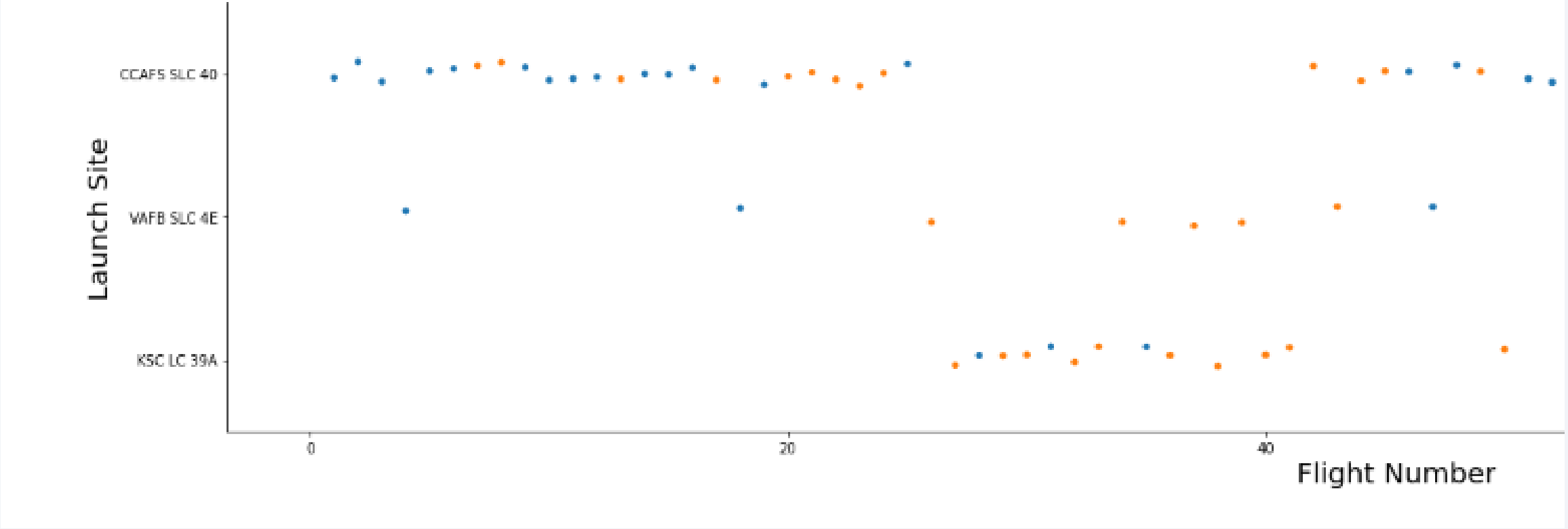
```
# Apply value_counts() on column Launch
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```


Section 2
Insights drawn from EDA

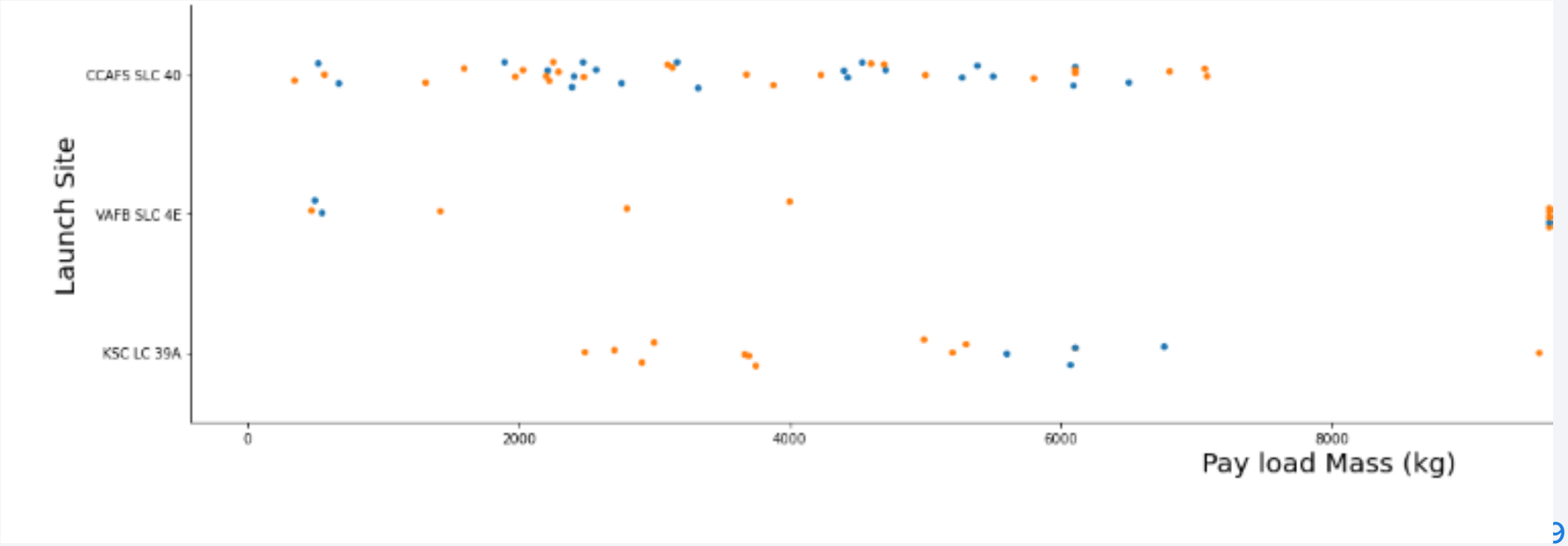
Flight Number vs. Launch Site

Looks like the Launch site CCFS SLC -40 has deployed most of the flights constantly since the very beginning



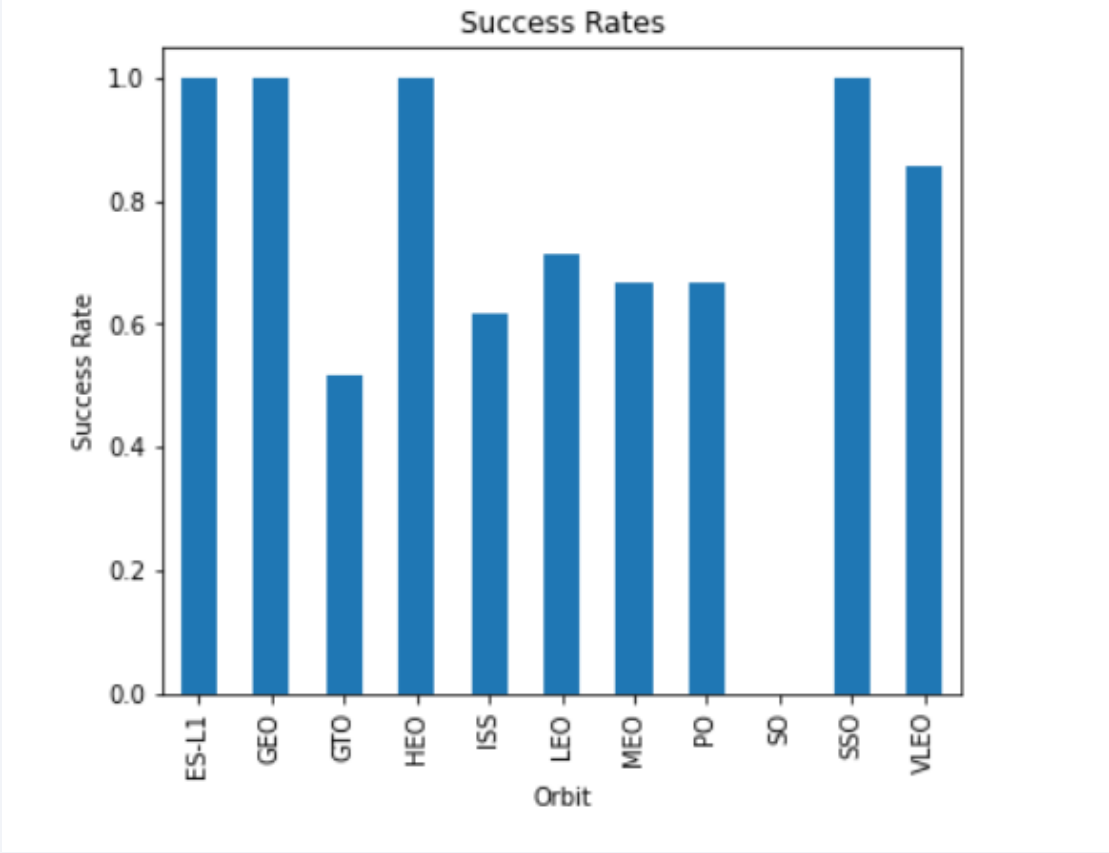
Payload vs. Launch Site

CCFS SLC -40 has deployed flights for different amounts of Pay load Mass and almost all of the flights have payload mass less than 8000 kg



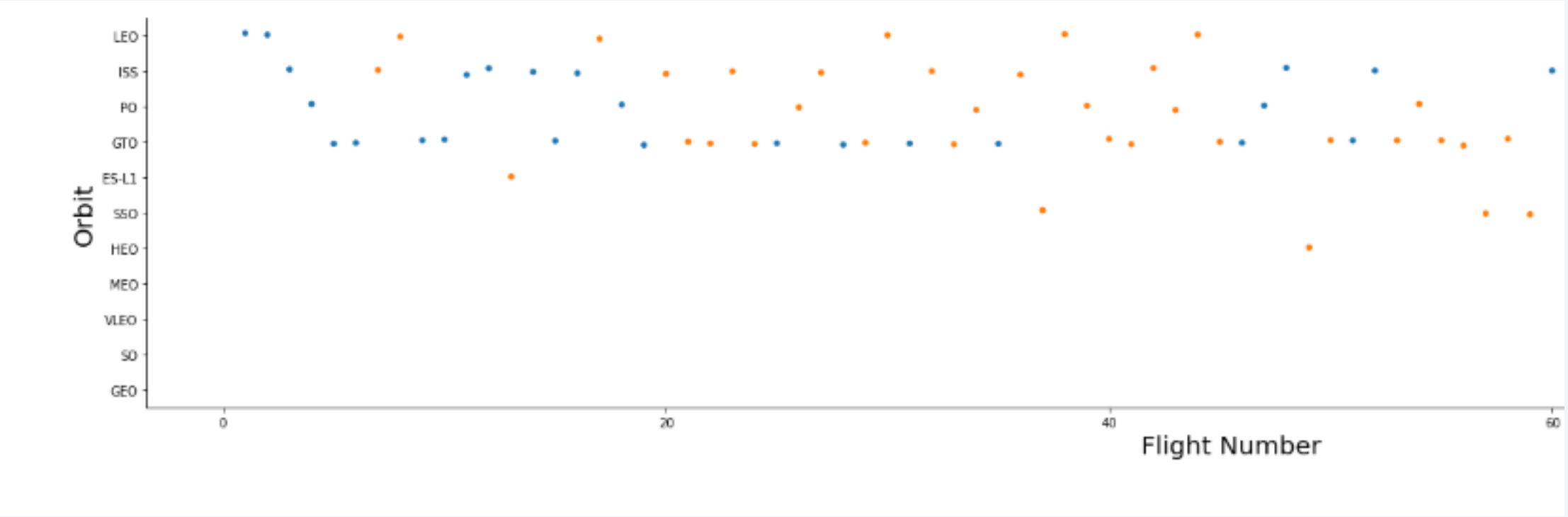
Success Rate vs. Orbit Type

Orbits ES-11, GEO, HEO, Sso have the highest success rate



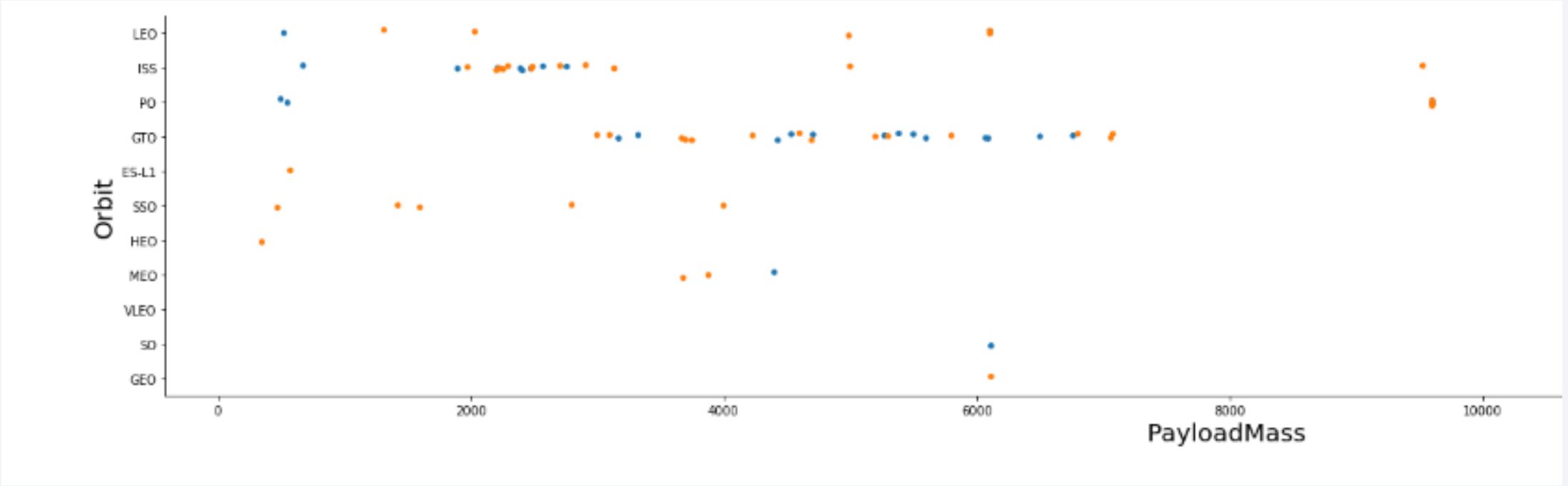
Flight Number vs. Orbit Type

Most of the flights were performed at Orbits: ES-11, GEO, HEO, SSO



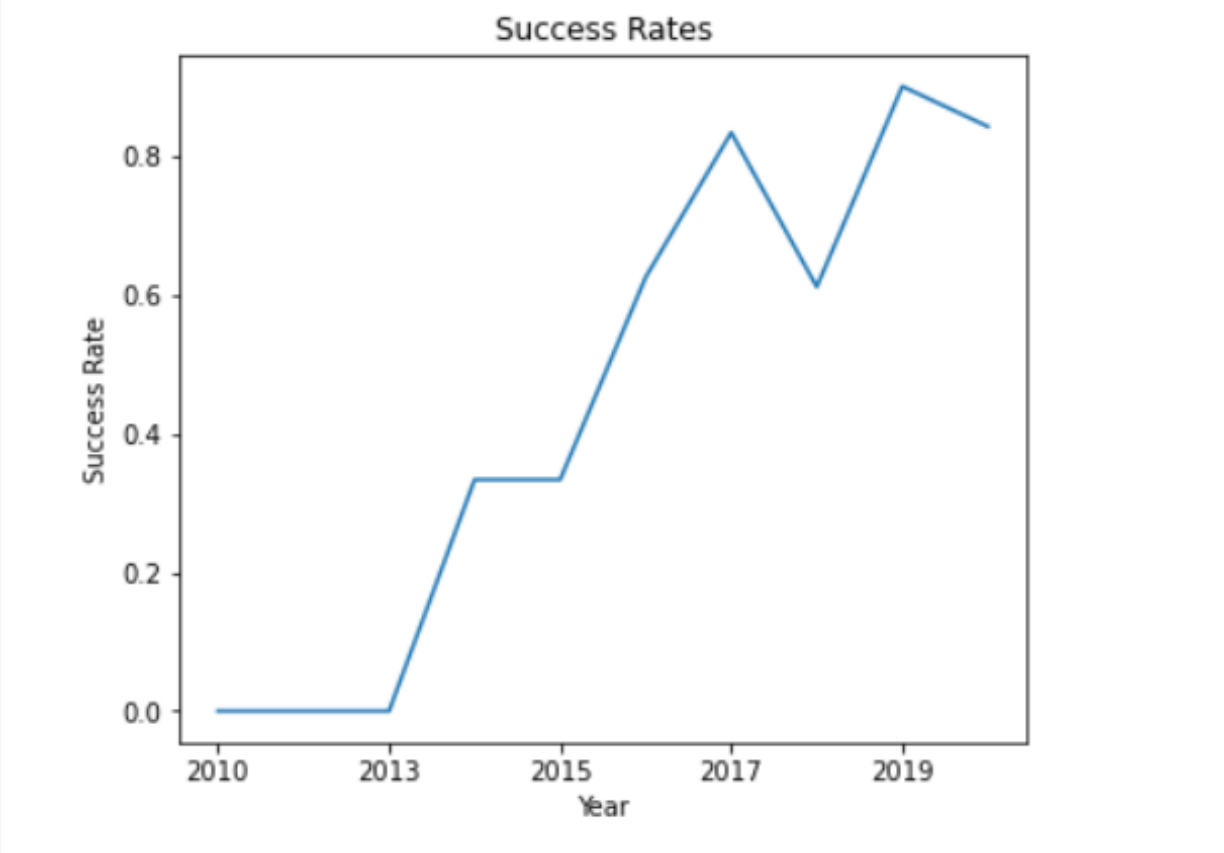
Payload vs. Orbit Type

Orbit GTO has flights with larger range of PayloadMass



Launch Success Yearly Trend

Success Rate has been increasing since the beginning higher in the following years.



All Launch Site Names

There were 55 Launches at CCFS SLC 40, 22 at KSC LC 39A and 13 at VAFB SLC 4E

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

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

Launch Site Names Begin with 'CCA'

Looks like there is something wrong in the first two launches because mass =0 does not make any sense

12]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677

Total Payload Mass

```
[3]: %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer = 'NASA (CRS)'
 * sqlite:///my_data1.db
Done.
```

[3]:

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

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

```
]: %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTI
```

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

]:

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

First Successful Landing was achieved in 01/01/17

```
]: %sql select min(DATE) from SPACEXTBL where "Landi
 * sqlite:///my_data1.db
Done.
```

]:

min(DATE)
01-05-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

Out[48]:

Booster_Version	Landing _Outcome
F9 FT B1022	Success (drone ship)
F9 FT B1026	Success (drone ship)
F9 FT B1021.2	Success (drone ship)
F9 FT B1031.2	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here

```
%sql  select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME like '%Success%'
```

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

count(MISSION_OUTCOME)
100

```
%sql  select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME like '%Failure%'
```

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

count(MISSION_OUTCOME)
1

Boosters Carried Maximum Payload

Out[49]:

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

2015 Launch Records

]:

Date	substr(Date, 4, 2)	Booster_Version	Launch_Site	Landing _Outcome
10-01-2015	01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
14-04-2015	04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

For this question I tried many different ways and none is working.

Please check my code.

Section 3

Launch Sites Proximities Analysis

Launch Sites

There are 4 Launch sites displayed, one in California and three in Florida. Since the three sites in Florida are very close, the markers are overlapped.

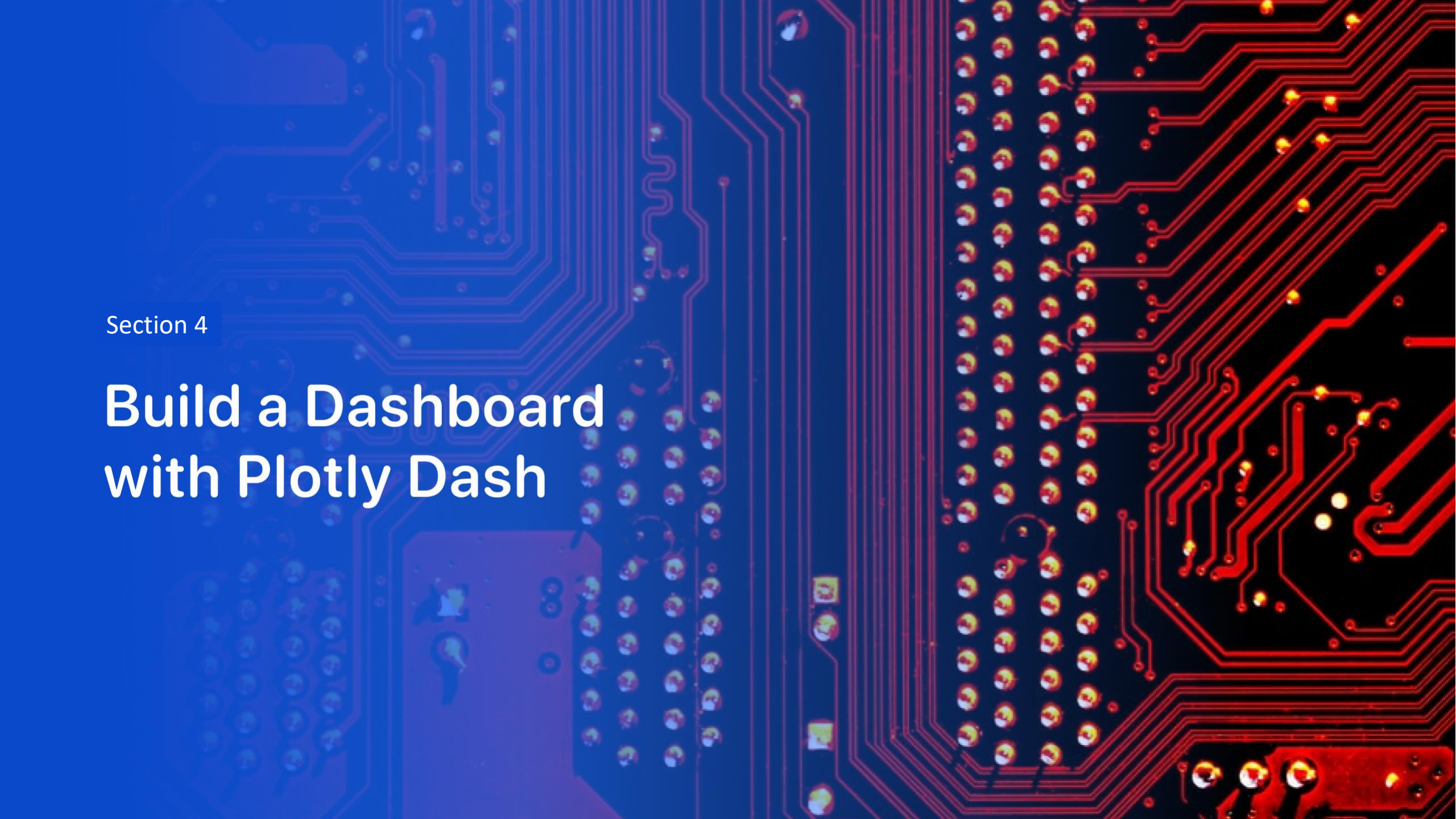
Out[16]:

Successful and Failed Landing

- Successful(green) and Failing landings(red) at Site CCAFS LC 40

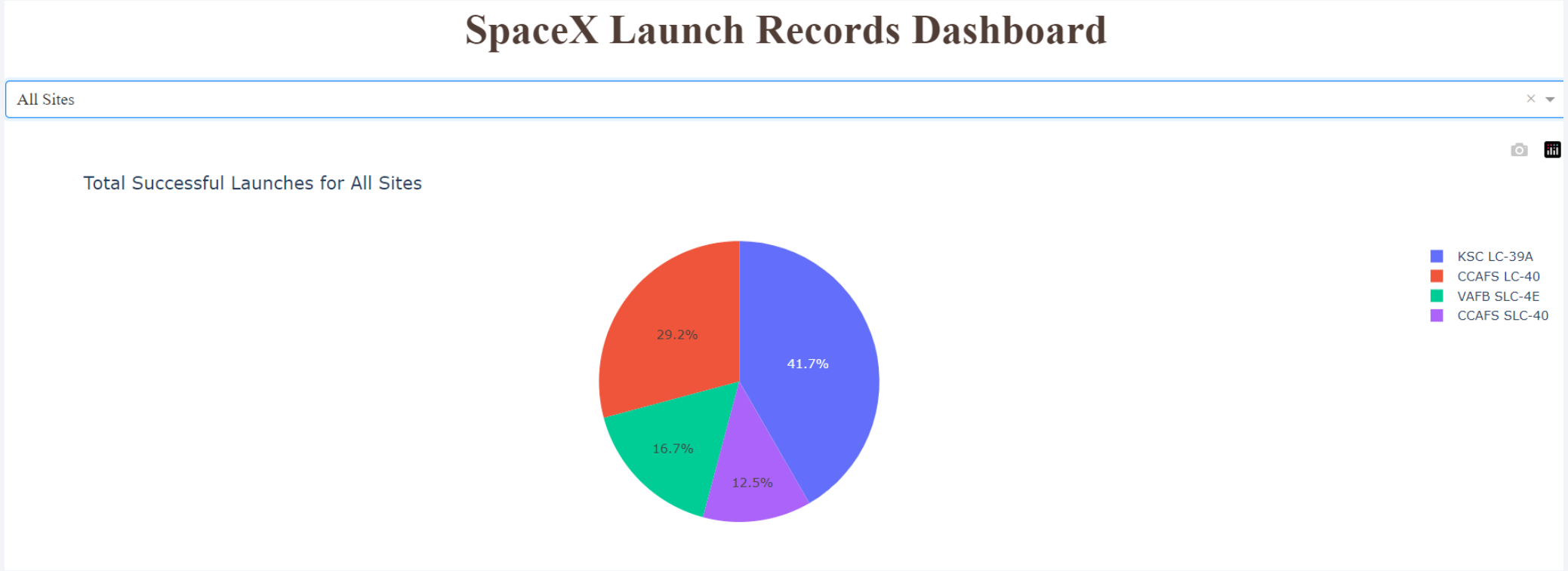
Distance from Landmarks

- The Displayed distance(0.58 KM) is distance of a Launch Site from a highway


Section 4
Build a Dashboard with Plotly Dash

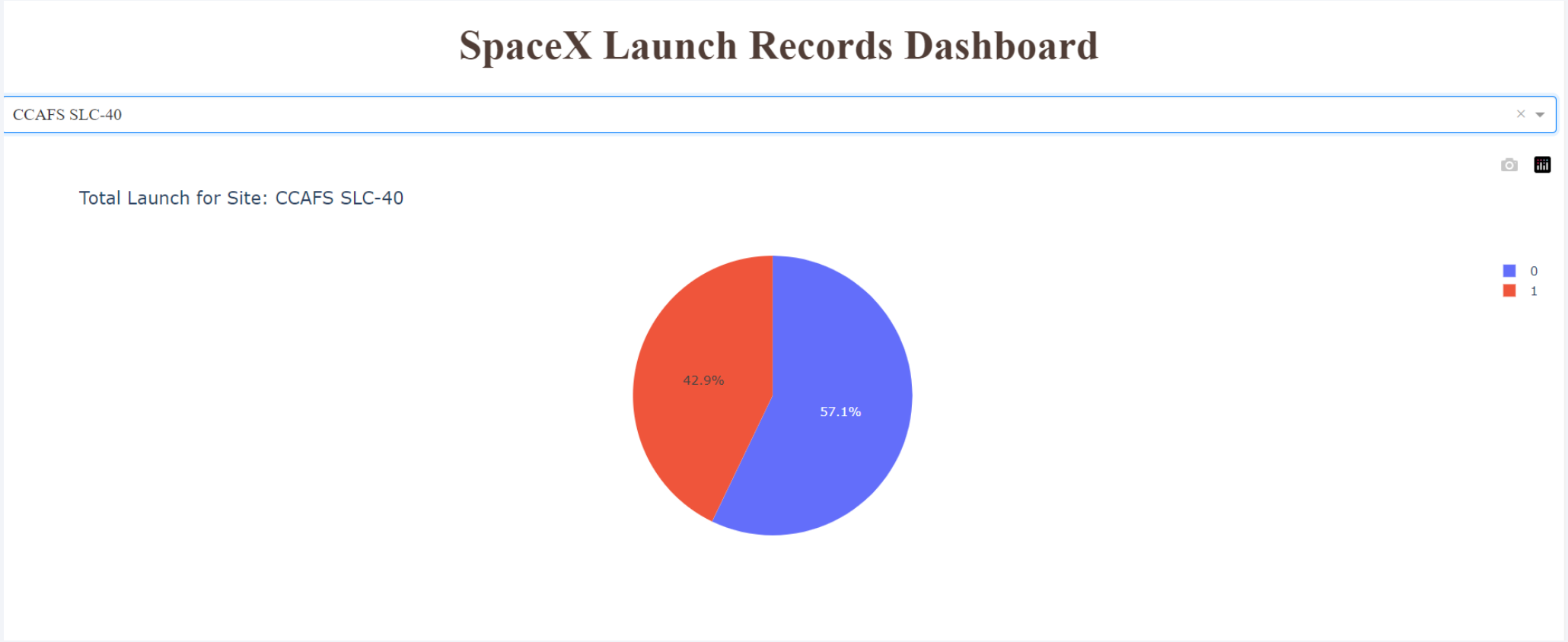
<Dashboard Screenshot 1>

Piechart for successes for all 4 sites



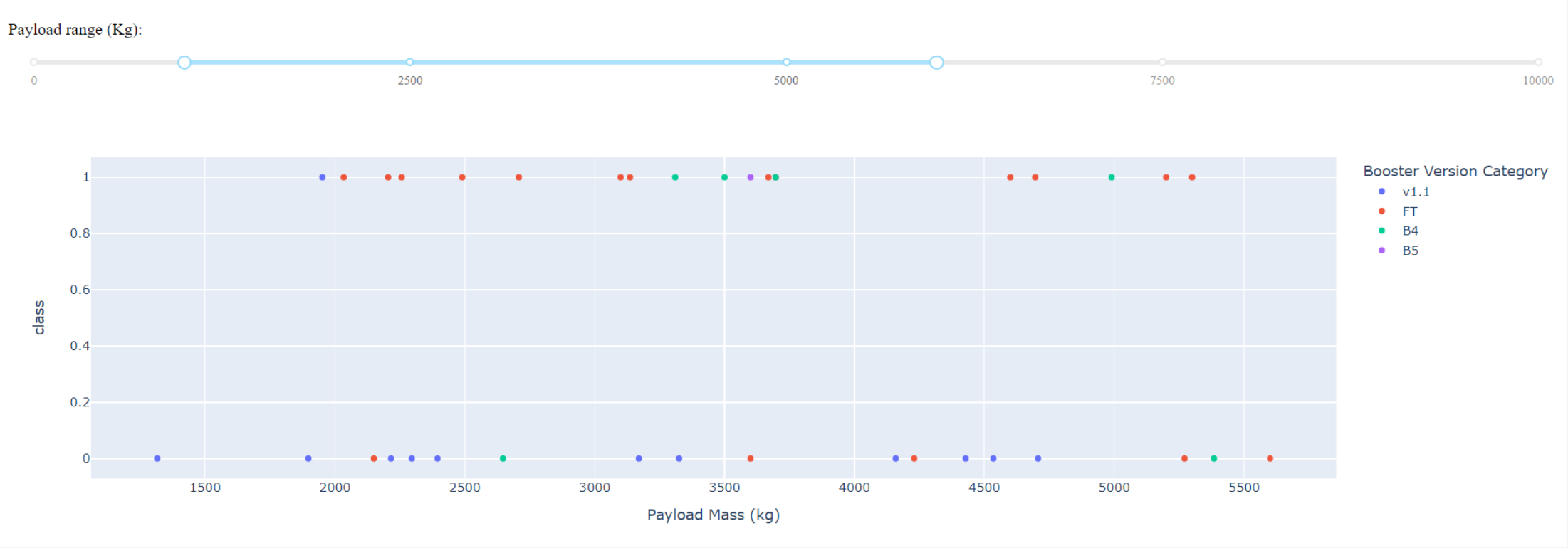
Pie Chart for Individual Sites

Piechart for one of the site with successe(blue) s and failures(red)



Class (Success/Failure) Vs. Payload Mass for different Booster Version

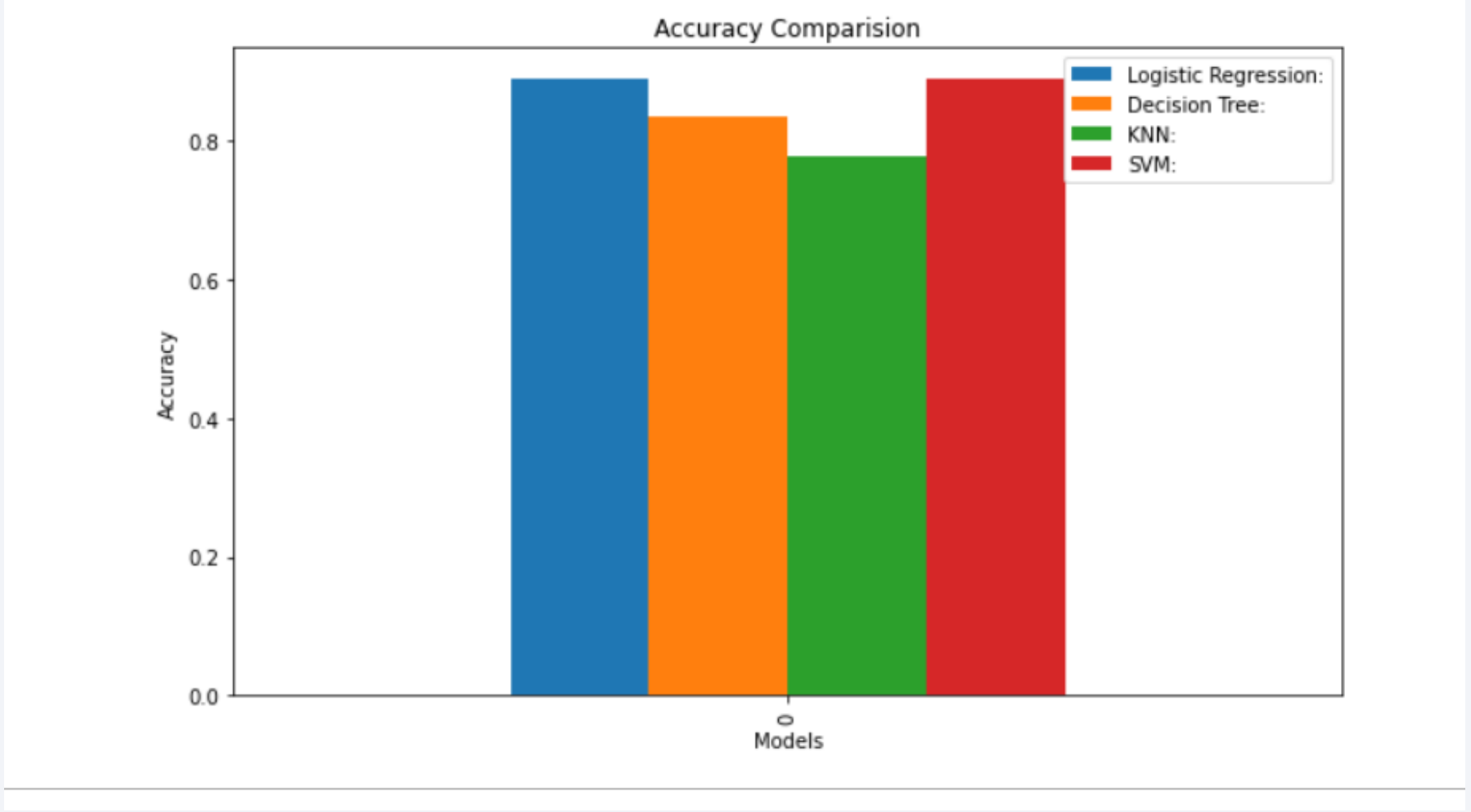
Looks like Payload mass has similar effect for all those versions



Section 5

Predictive Analysis (Classification)

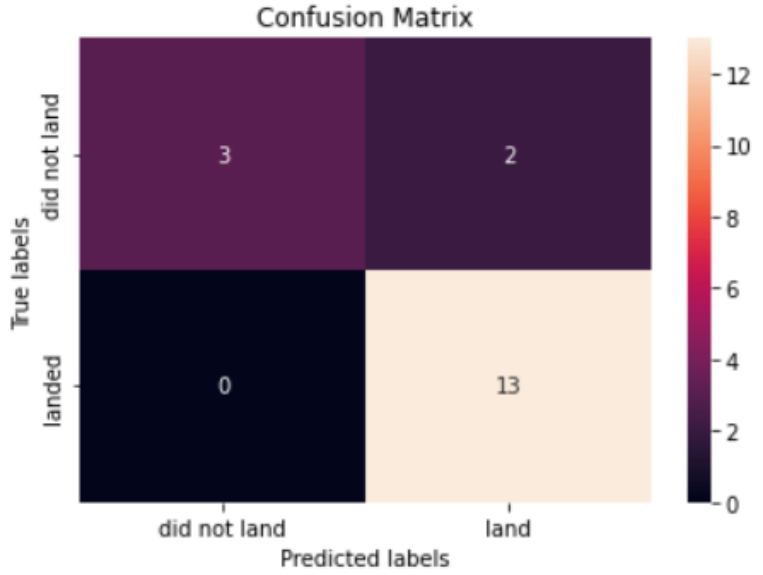
Classification Accuracy



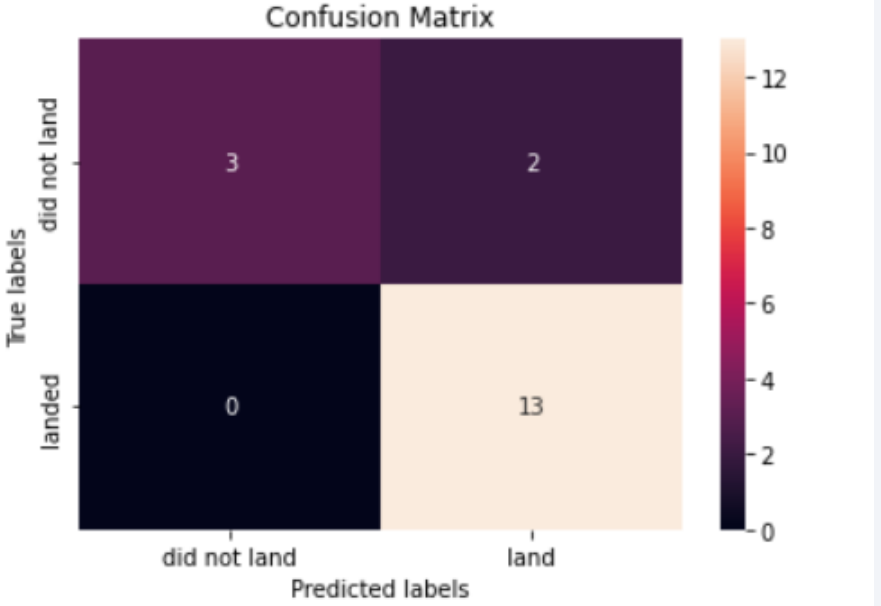
Confusion Matrix

Both LogReg and SVM classifications have exactly the same confusion Matrix

```
In [15]: yhat=logreg_cv.predict(X_test)
         plot_confusion_matrix(Y_test,yhat)
```




```
In [20]: yhat=svm_cv.predict(X_test)
         plot_confusion_matrix(Y_test,yhat)
```

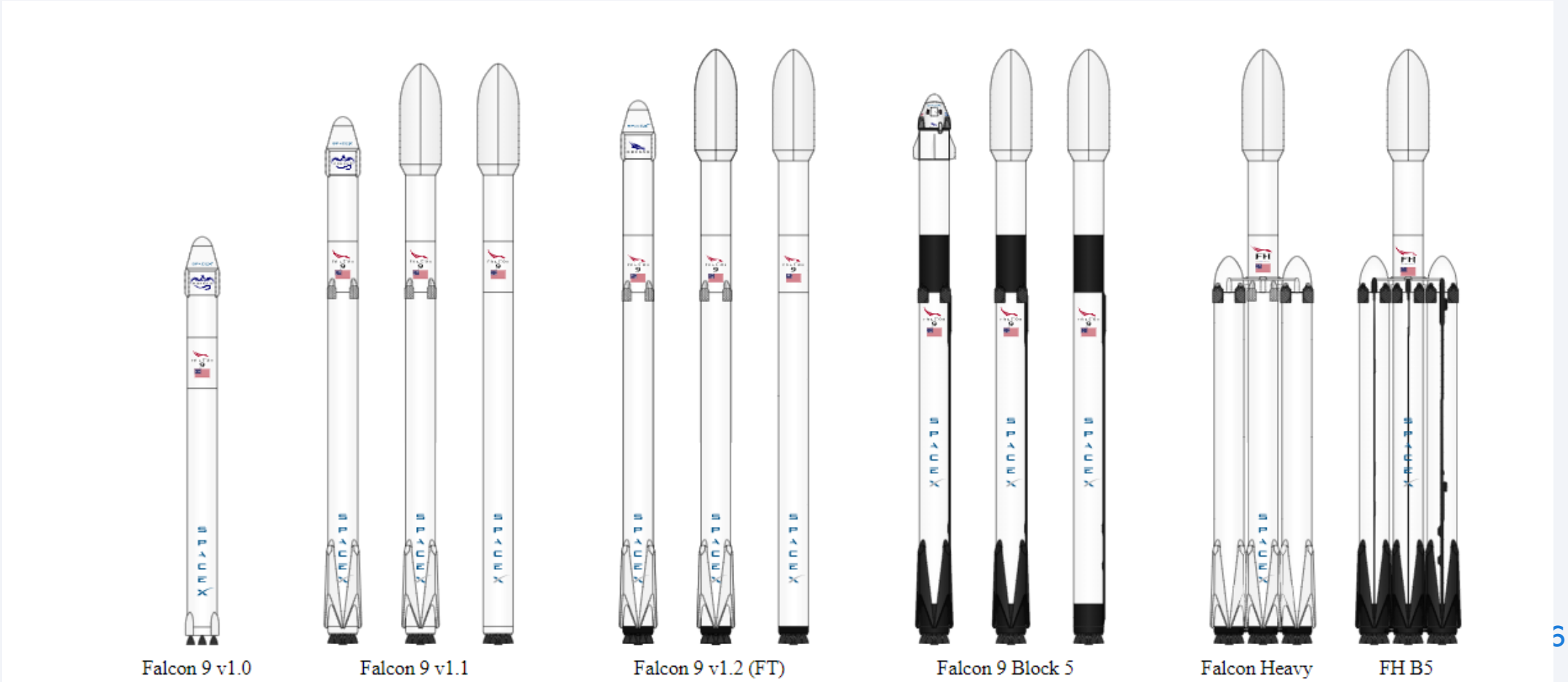


Conclusions

- Analysis was performed with the data consisted 4 Launch sites, 3 in Florida and 1 in California.
- Various classification model were applied to see the percentage of successful landings.
- Based on the results of the analysis the probability of Landing was about 88%
- Since there is a high chance of successful landing. So the first stage can be reused. Hence the cost of flight can be reduced.

Appendix

Different types of Falcon 9 Launches



Thank you!