IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Lakshmi Deeduvanu
02/09/2022

Outline

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

Executive Summary

- Methodologies Summary:
  - Data Collection
  - Data Wrangling
  - EDA using visualization and SQL
  - Interactive visual analytics using Folium and Plotly Dash
  - Predictive analysis using classification models
  - How to build, tune, evaluate classification models
- Summary of all results
  - Exploratory Data Analysis
  - Interactive Analysis
  - Model Evaluations

Introduction

Background and Context

Space Y Data Science project is focussed on learning from Space X Rocket Launch data. Space X is making space travel affordable for everyone. Space X has Falcon 9 rocket launches for 62 million, much less than the 165 million from other companies. This is because it can reuse the first stage. If we can determine if the first stage will land, we can estimate the cost of a launch. The data findings from Space X will help create better models that can help predict outcomes on future rocket launches. The report lists data findings for Space X company based on public dataset. This report is on Space Y Rocket Company competing with Space X.

The problems we want to find solutions to is:

- Understand and Analyze the data
- Identify the key “features” that provide correlation between the data and outcome.
- Analyze best algorithm that should be used for the outcome prediction.
- Predict Outcome if Space X will attempt to land a rocket successfully or not

Section 1

Methodology

Methodology

- Data collection:
  - Space X Public Dataset: JSON response from the REST API
    - https://api.spacexdata.com/v4/launches/past
  - Web Scraping from WIKI pages: Python BeautifulSoup package to get rocket launches from websites.
    - https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- Data wrangling:
  - Convert data from SpaceX API and Web Scraping into Pandas Data Frame.
  - Check for missing data per each column and replace Nan values with mean of a column.
  - Only keep the Falcon 9 Rocket Launch data.
  - Identify data types are categorical and numerical. Use One Hot Encoder algorithm with Python Pandas data frame get dummies() method, so it can be used for analysis better.
  - Converted columns in a data set using asType to float64

Methodology

- ## Perform exploratory data analysis (EDA) using visualization and SQL

- SpaceX dataset is imported into EDA DB2 to create a table. Queried tables using Python SQL alchemy for understanding which features in the dataset are important to predict the outcome better. Also used Visualizations with Python Matplotlib, Numpy, Seaborn modules – Scatterplot, bar chart, Line plot to visualize relations between the various key features such as launch site, payload mass, orbit, year, flight number etc in predicting the mission outcome.

- ## Perform interactive visual analytics using Folium and Plotly Dash

- We use Python Folium and Dash libraries to get a visual on Launch sites and the distribution of the data in world map. Mark all Launch Sites on the map and perform analysis to the distances from the Launch site to near by locations such as a city, railroad and highway. We add markers to mark the number of launches in each site and color code them.

- ## Perform predictive analysis using classification models

- To predict the outcome of future rocket launches, we determine that we need a classification algorithm for the model. Test the 4 classification models such as KNN, Decision Tree, SVM and Logistic Regression on the data set. The dataset is first split at 0.2 percent of test data and 0.8 for train data.

- ## How to build, tune, evaluate classification models

- Model is built with train data. Each of these models are tuned to search for best parameters for the model using GridSearchCV object on the algorithm,. The model is created with the best parameters. The Test dataset is used to predict the model. The Model is also evaluated using metrics such as the Jaccard score, F1-score and Log Loss and confusion matrix.

Data Collection

- There were 2 methods to collect the data for Space X Launches for Falcon 9.

1. SpaceX REST API Endpoint with JSON

1. http://api.spacexdata.com/v4/launches/past
2. Convert the JSON into Pandas data frame using json_normalize() and apply transformations

2. Web Scraping from WIKI pages:

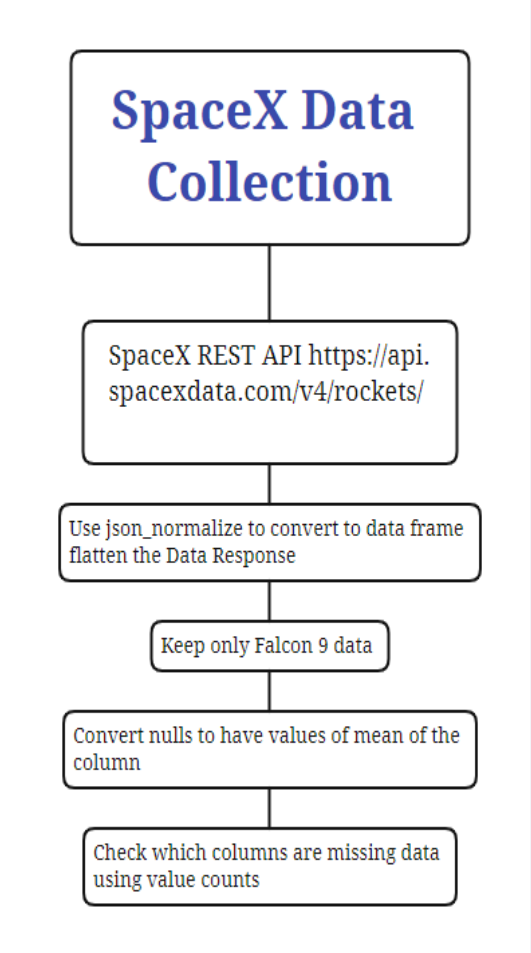
1. https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
2. Using the Python Beautiful Soup package to scrape the websites HTML tables containing Falcon 9 data.
3. Then convert that into Pandas data frame and apply transformations.
4. This info only has id numbers, so used SpaceX API endpoints to get mapping from ID to Booster, Launchpad , Payload etc.

Data Collection – SpaceX API

- Pull Space X Data from
  https://api.spacexdata.com/v4/rockets/

- Convert response to Pandas Data frame
- Filter data to only include Falcon 9 launches
- Create dictionary with required columns in the dataset.
- Check which columns are missing data
- Get the mean of the columns that have Nan values and add them to the Nan values

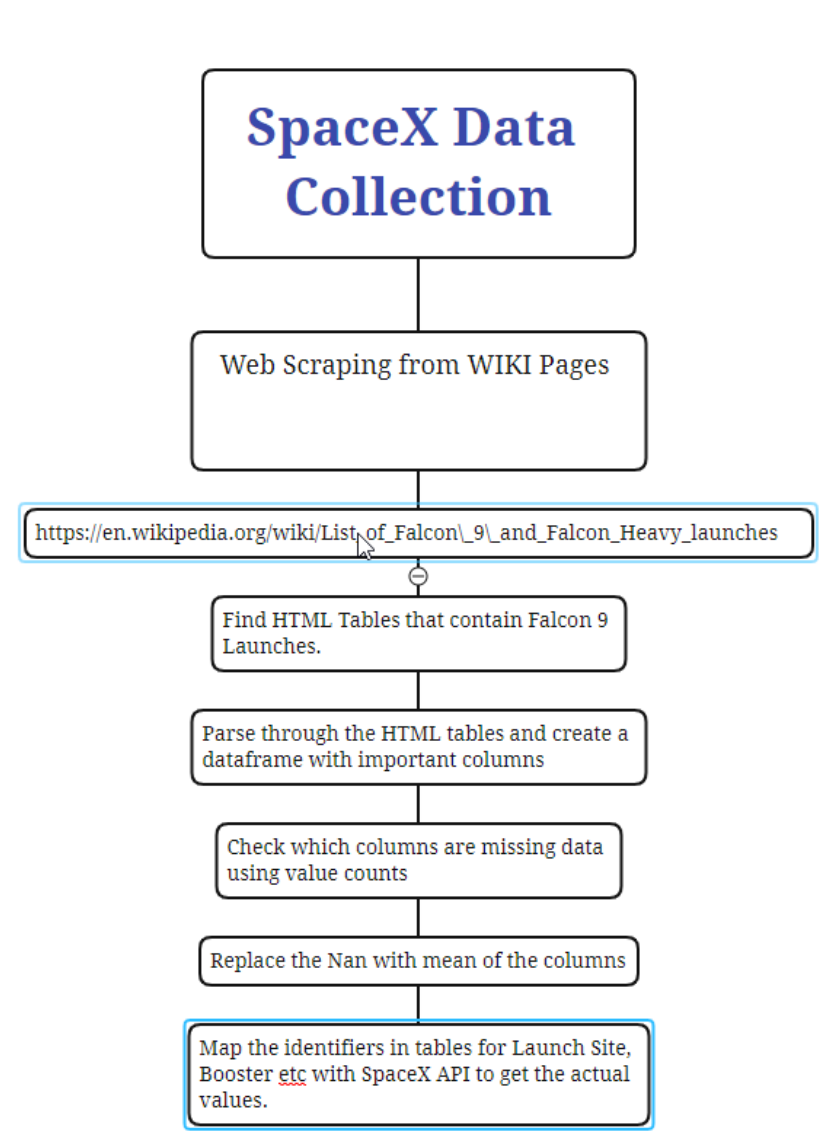
- GitHub URL
  https://github.com/lakdee/coursera_capstone_ml_project/blob/master/Data%20Science%20Final%20Capstone%20-%20Data%20Collection.ipynb



Data Collection - Scraping

- Web scraping was done using Beautiful Soup python module, from the website
- https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- Extracted a Falcon 9 launch records HTML table from Wikipedia
- Parse the table and converted it into a Pandas data frame for further analysis

- GitHub URL
- https://github.com/lakdee/coursera_capstone_ml_project/blob/master/Data%20Collection%20with%20Web%20Scraping.ipynb



Data Wrangling

- Convert data from SpaceX API and Web Scraping into Pandas Data Frame.
- Check for missing data per each column and replace Nan values with mean of a column.
- Only keep the Falcon 9 Rocket Launch data.
- Identify data types are categorical and numerical. Use One Hot Encoder algorithm with Python Pandas data frame get dummies() method, so it can be used for analysis better.
- Converted columns in a data set using asType to float64

- GitHub URL
- https://github.com/lakdee/coursera_capstone_ml_project/blob/master/Data%20Wrangling.ipynb

EDA with Data Visualization

- Graphs Plotted:
    1. Flight Number Vs Launch Site Scatter Plot
    2. Payload Mass Vs Launch Site Scatter Plot
    3. Success Rate Vs Orbit Type Bar Chart
    4. Flight Number Vs Orbit Type Scatter Plot
    5. Payload Vs Orbit Type Scatter Plot
    6. Launch Success Yearly Trend Line Plot

- GitHub URL
- https://github.com/lakdee/coursera_capstone_ml_project/blob/master/EDA%20with%20Visualizations%20Charts.ipynb

EDA with SQL

Summary of SQL queries performed

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

- GitHub URL:

https://github.com/lakdee/coursera_capstone_ml_project/blob/master/Exploratory%20Data%20Analysis%20on%20SpaceX%20Dataset.ipynb

Build an Interactive Map with Folium

Following Map objects (markers, circles, lines) were created and added to a folium map

1. NASA Marker - using the Latitude and Longitude in Houston. Marker is marked with Yellow circle.
    1. Reason: NASA marker was added to represent the main NASA Location on the Folium Map.
2. Launch Site Marker for Space X sites are CCAFS LC-40, CCAFS SLC-40, KSC LC-39A, VAFB SLC-4E.
    1. Reason: To represent all unique Launch Sites for Space X on Folium World Map
3. Marker Cluster – Red and Green Outcome to mark the success / failed launches for each site
    1. Reason: To represent the number of SpaceX Launches per each site. The Marker Cluster is a great way to simplify data points that represent number of launches for a Launch Site, especially because many launches happen at one location.
4. Calculate the distances between a launch site to its proximities
    1. Line and distance (in KM) to represent distance between KSC LC-39A and its nearest city Titusville
    2. Line and distance (in KM) to represent distance between KSC LC-39A and its closest Railroad – NASA Railroad
    3. Line and distance (in KM) to represent distance between KSC LC-39A and its closest Highway – Kennedy Parkway North Hwy

GitHub URL

- https://github.com/lakdee/coursera_capstone_ml_project/blob/master/Data%20Visualizations.ipynb

Build a Dashboard with Plotly Dash

The Following Plots and Graphs have been added to the Plotly Dashboard.

- Launch Site Success Pie Chart
  - This interactive dashboard, provides an easy way to see which Launch Sites have better Success Outcomes.
  - The ALL option allows to quickly check the Outcome Success in each Launch Site in a Pie Chart.

- Payload Vs Launch Outcome Scatter Plot
  - This Scatter Plot in the dashboard, enables to check Success Outcomes for Payload Mass (Kg) for each Booster Version Category
  - A Ranger Slider helps to easily visualize and set the minimum and maximum values easily

- GitHub URL
- https://github.com/lakdee/coursera_capstone_ml_project/blob/master/pie-dash.py

Predictive Analysis (Classification)

1. Split the data into Train and Test Datasets
2. Repeat Steps 3 – 6 for each of the model algorithms – Logistic Regression / Decision Tree / SVM/ KNN
3. Create Model Object, then create GridSearchCV with CV =10. using train data
4. Find best parameters and best Score accuracy.
5. Create the model using the best parameters
6. Predict the test data using this model and create Confusion Matrix

- GitHub URL
- https://github.com/lakdee/coursera_capstone_ml_project/blob/master/pie-dash.py

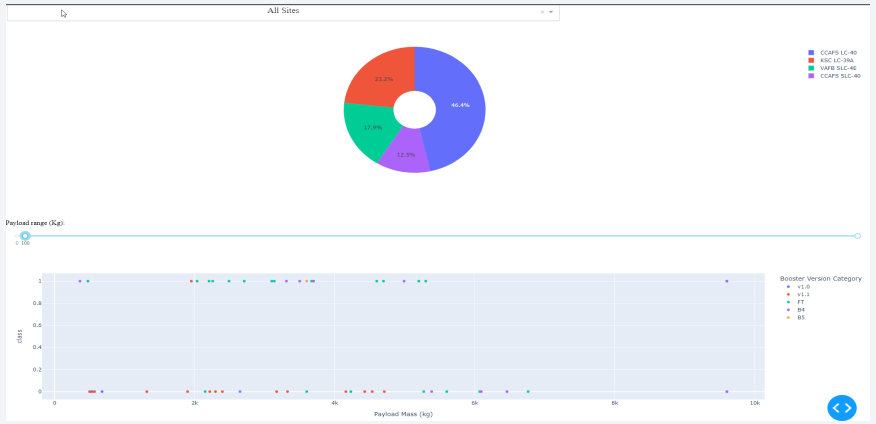
Results

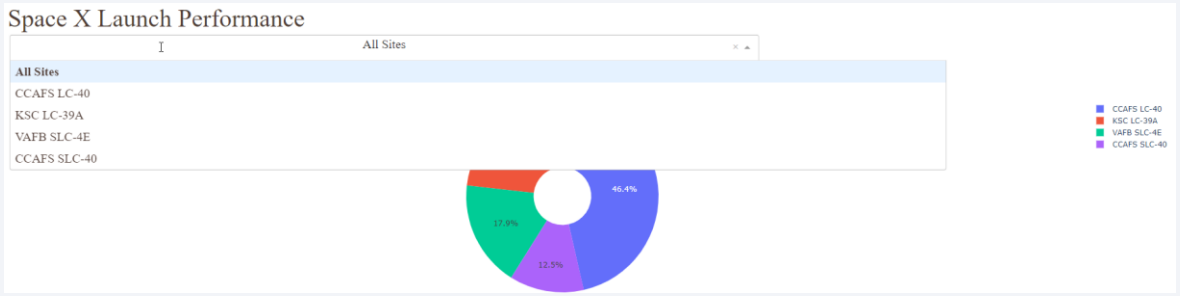
- Exploratory data analysis results

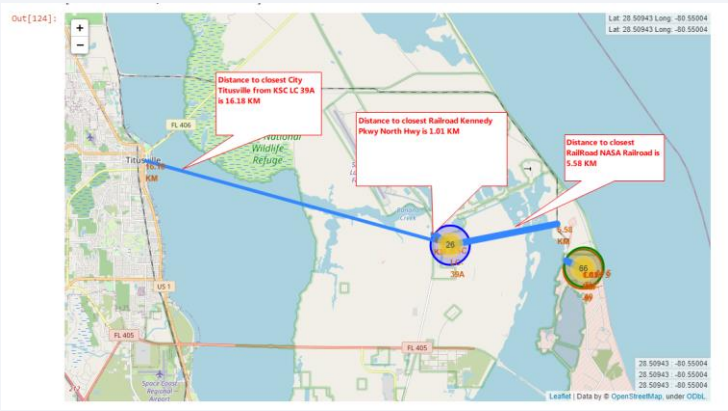
Success rate since 2013 kept increasing till 2020	Total Payload Mass carried by boosters launched by NASA CRS = 45596 KG
CCAFS Site most launches were up to 7000Kg	2015-12-22 was the first Success Launch date for SpaceX
For VAFB-SLC Launch Site there are no rockets launched for heavy payload mass(greater than 10000).	Total Number of Failure in flight = 1
The rate of success was higher for Payload Mass 7000 kg for all sites	Total Number of Success = 99
ES-L1 , GEO and HEO and SSO Orbits have a High success rates always and no failures	
GTO has equal success and failure rate	
ISS, LEO, MEO, PO Orbits have greater than average mix of both success and failure	
GTO has uniform balance of success Vs not success	
With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.	
There were no tests with ES-L1 / SSO , HEO, MEO Over 4500 Payload mass	

Results

- Interactive analytics demo in screenshots

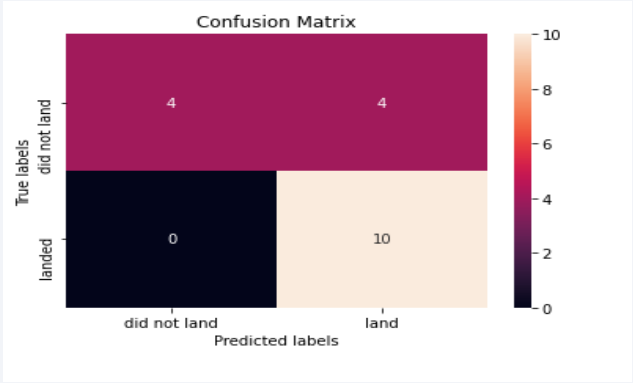




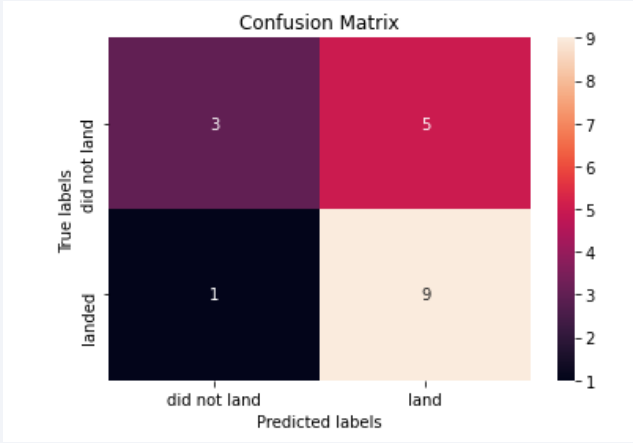


Results

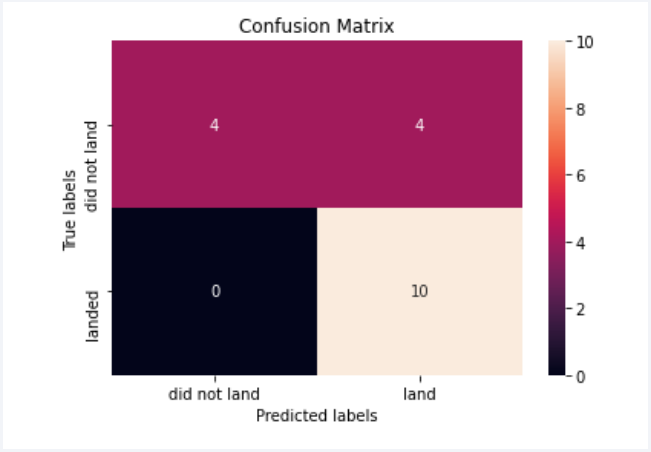
- Predictive analysis results
- KNN: 0.8767857142857143



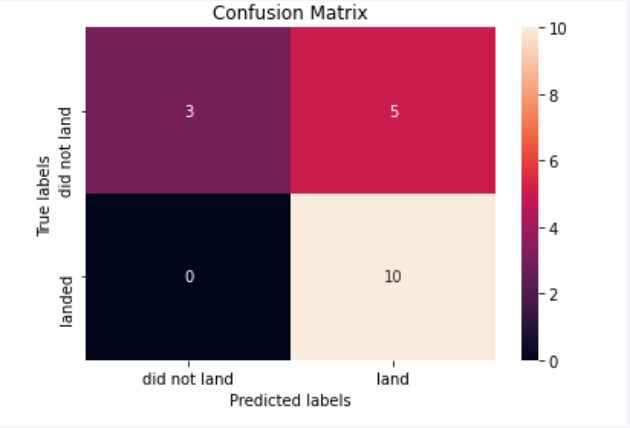
- Decision Tree = 0.9321428571428573

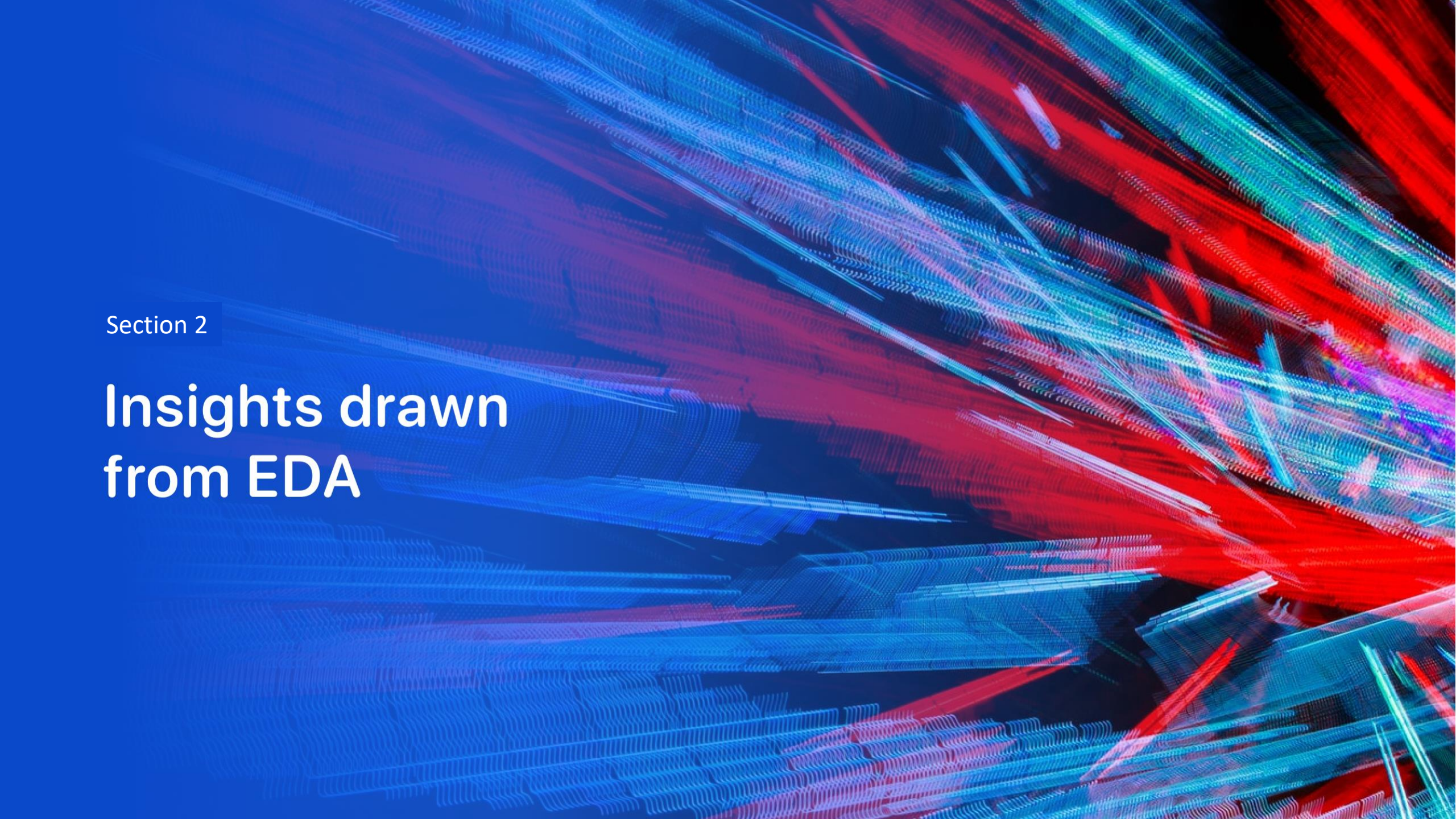


- Support Vector Machine = 0.8625

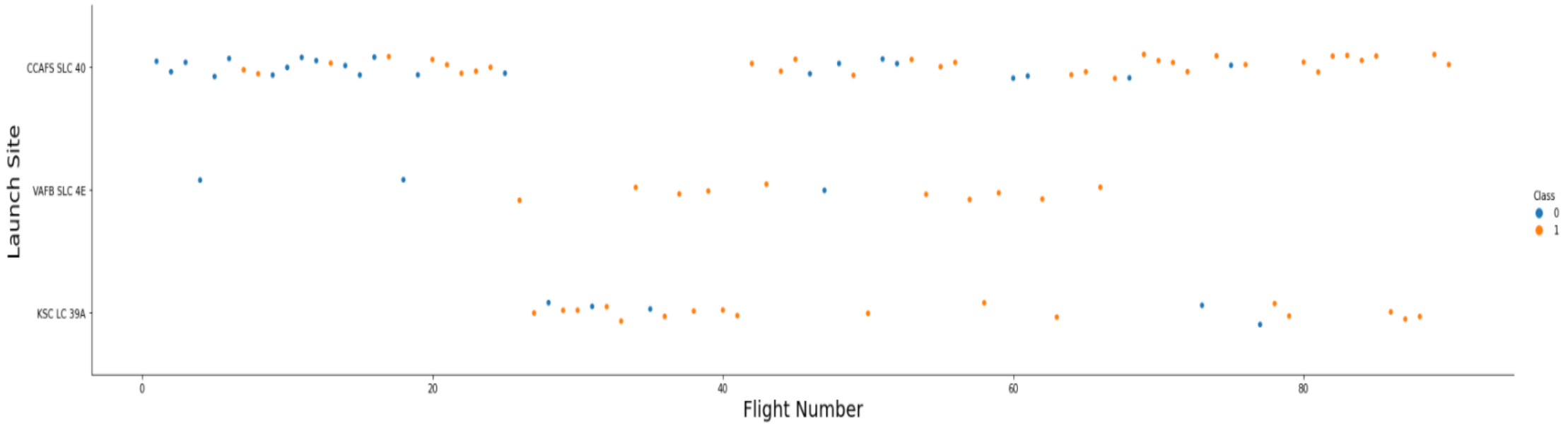


- Logistic Regression = 0.8357142857142857



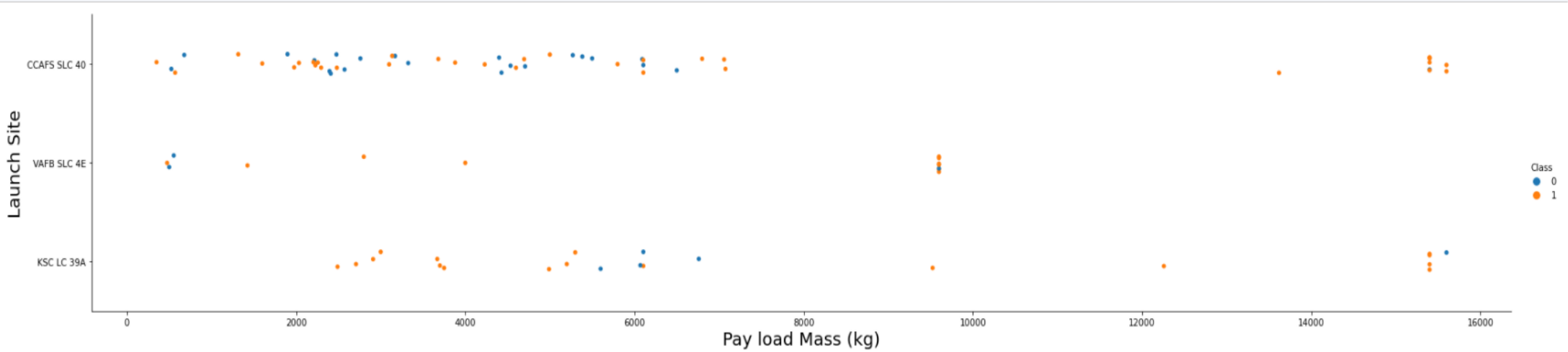

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site




```
n [ ]: #The scatter plot between Flight number and Launch site
       #1) Much more flights were launched in CCAFS SLC 40 than VAFB SLC 4E and KSC LC 39A
       #2) Num of success for Flight numbers after 50 is higher for all 3 sites especially for VAFb SLC 4E
       #3) Launch Site is not a increasing variable so  there is not much pattern between Launch Site and Flight Number
```

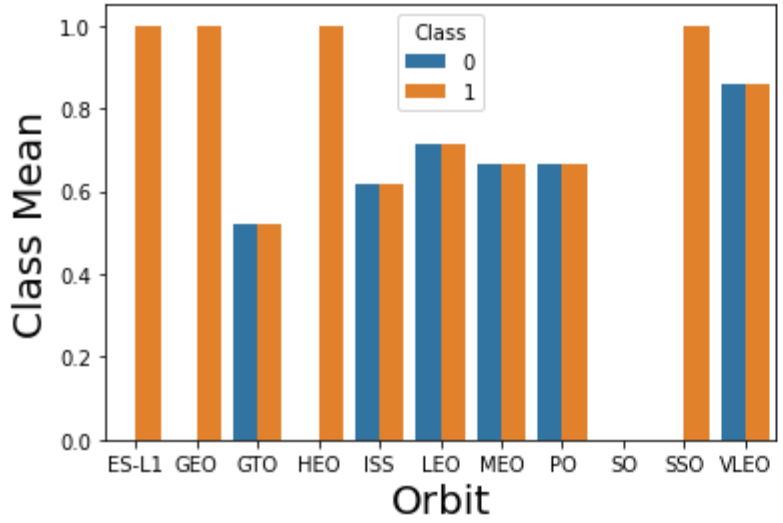
Payload vs. Launch Site



Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

[ ]:
```
#Payload MAss and Launch Site Observations
#CCAFS Site most launches were upto 7000Kg
#for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).
#The rate of success was higher for Payload Mass 7000 kg for all sites
```

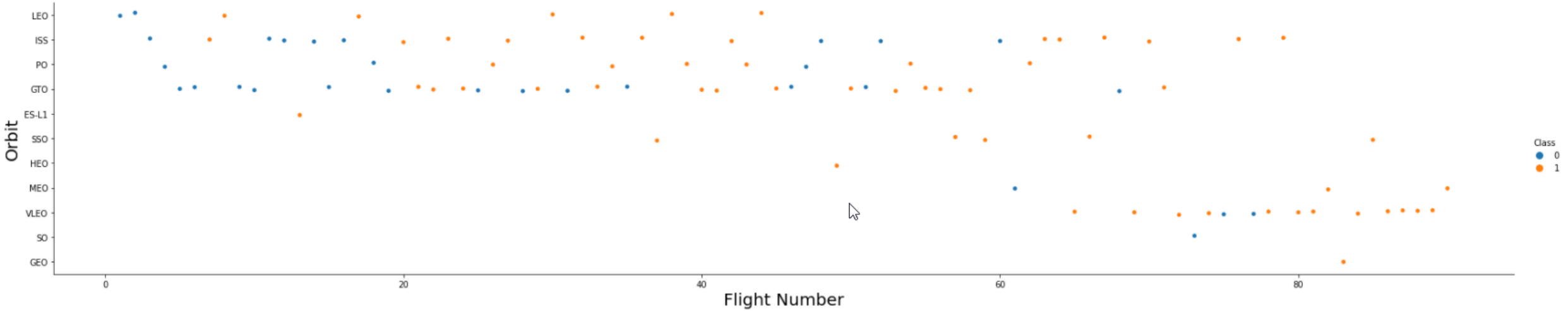
Success Rate vs. Orbit Type



Analyze the ploted bar chart try to find which orbits have high sucess rate.

```
[ ]: # Orbit Vs Mean_Class Scatter Plot Observations
     # ES-L1 , GEO and HEO and SSO Orbits have a High success rates always and no failures
     # GTO has equal success and failure rate
     # ISS LEO MEO PO Have greater than average mix of both success and failure
```

Flight Number vs. Orbit Type

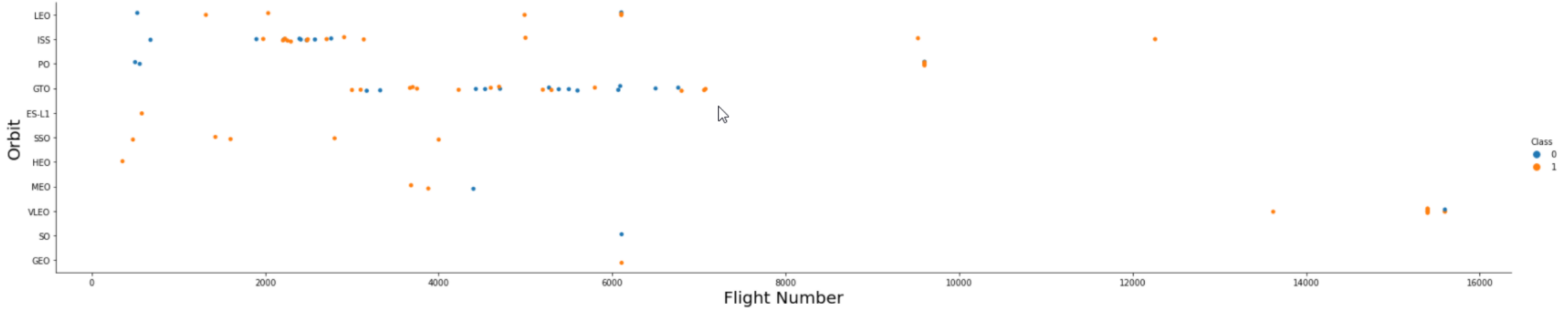


You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

```
# Flight Number Vs Orbit Scatter Plot Observations
# Number of success is greatly higher after Flight 60 for all Orbits
# Number of majority failures in Orbits is before Flight 20
```

Payload vs. Orbit Type

In [38]:
```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

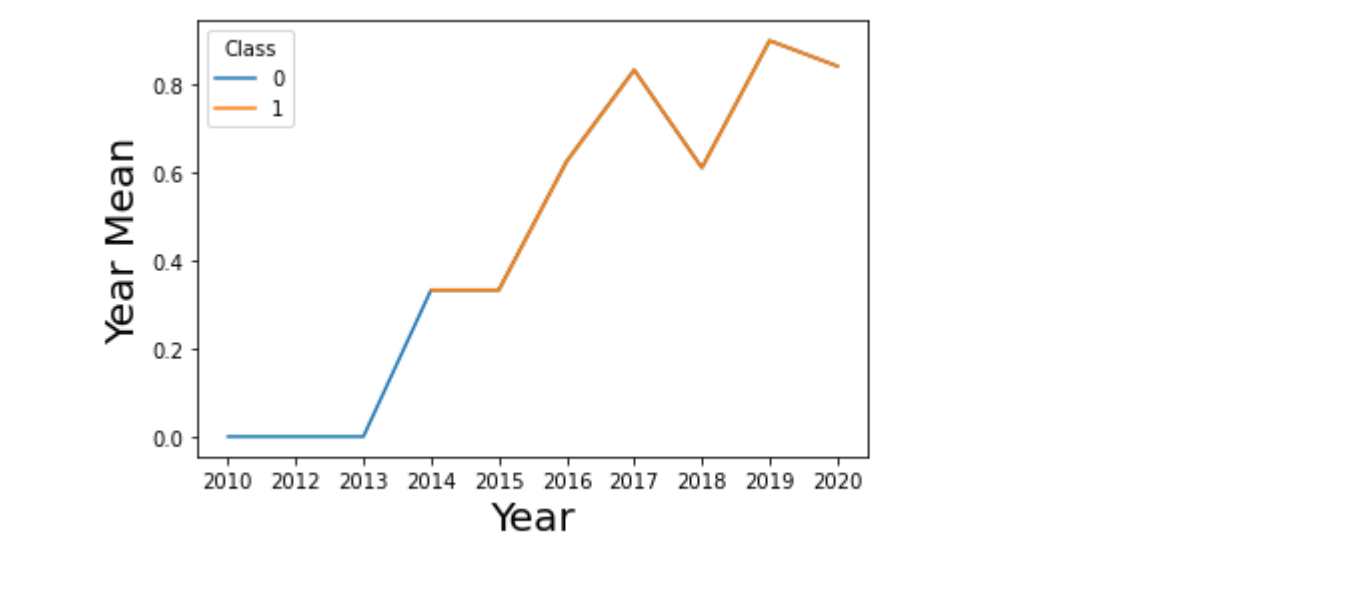


With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

In [ ]:
```
# PayloadMass Vs Orbit Scatter Plot Observations
# With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
# GTO has uniform balanace of success Vs not success
# There were no tests with ES-L1 / SSO , HEO, MEO Over 4500 Paylod mass
```

Launch Success Yearly Trend



Line Graph indicates that after 2014, the success rates was highly increased.

All Launch Site Names

- Find the names of the unique launch sites
- Used SQL DISTINCT function

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Used SQL LIKE ‘CCA%’ function

DATE	Time (UTC)	BoosterVersion	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

- Calculate the total payload carried by boosters from NASA
- Used SQL SUM function on PAYLOAD_MASS__KG_ where CUSTOMER ='NASA (CRS)

Total Payload Mass
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Used SQL AVG Function WHERE BOOSTER_VERSION like ‘F9 v1.0%’

Average Payload Mass
340

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Used SQL MIN function where Landing_outcome = ‘Success (ground pad)’

First 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

- Booster Version where LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000

Booster Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Used SQL COUNT(*) with GROUP BY MISSION_OUTCOME

Mission Outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Used SQL subquery that has MAX(PAYLOAD_MASS__KG_)

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

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Used SQL YEAR(DATE) function to get the year 2015

Landing__outcome	booster version	launch_site	DATE
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015-01-10
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015-04-14

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
- Used startdt and enddt variables using : DATE > :startdt AND DATE < :enddt GROUP BY LANDING__OUTCOME DESC

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

Section 3

Launch Sites Proximities Analysis

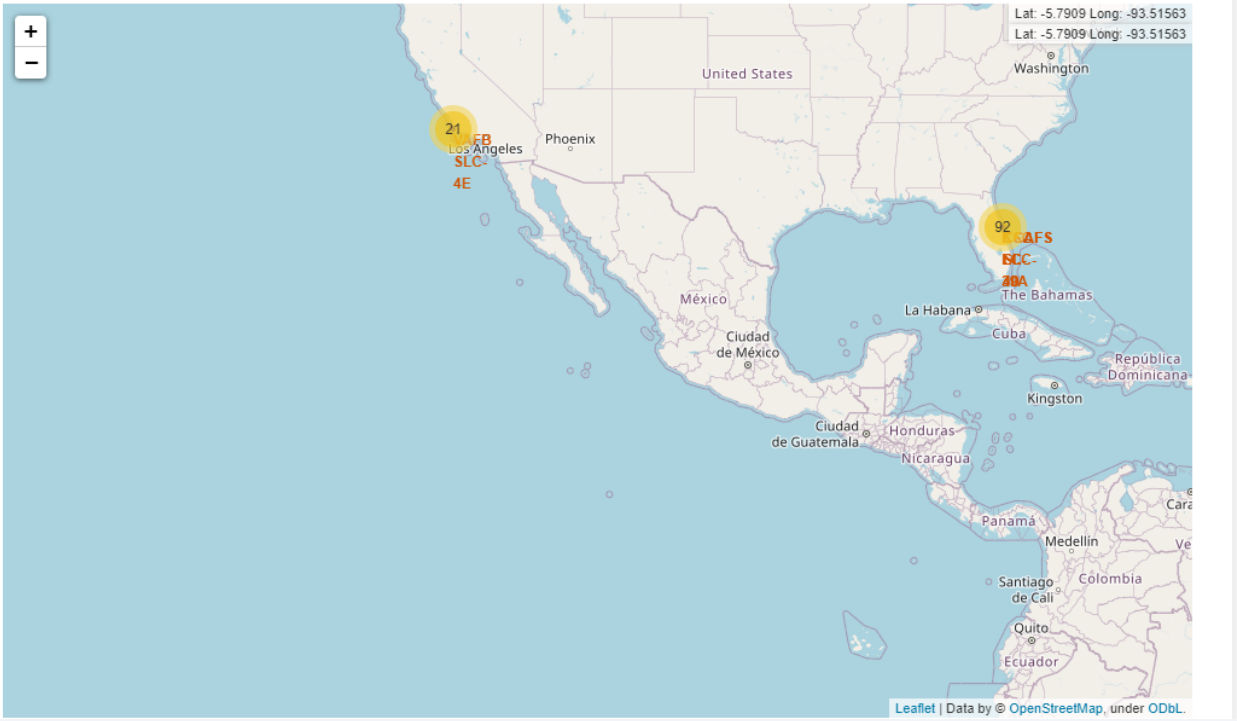
Map circling Launch Site and Label

Displays the Markers and Circles for Each Launch Site. The Circles have a radius of 1000.

Number of Launches per Location

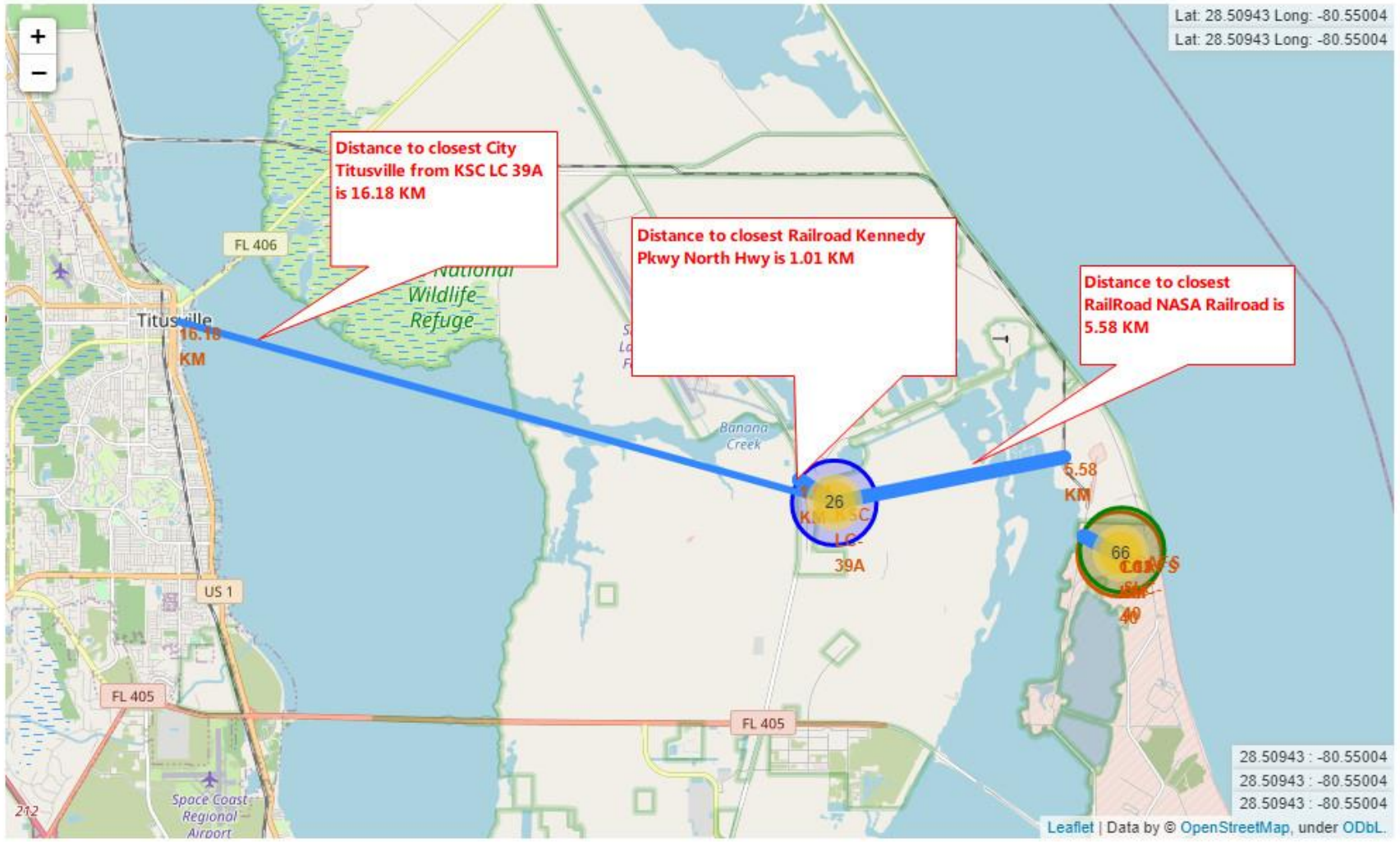
Displays the count of the number of launches per Launch site with 21 on west coast and 92 on the east coast

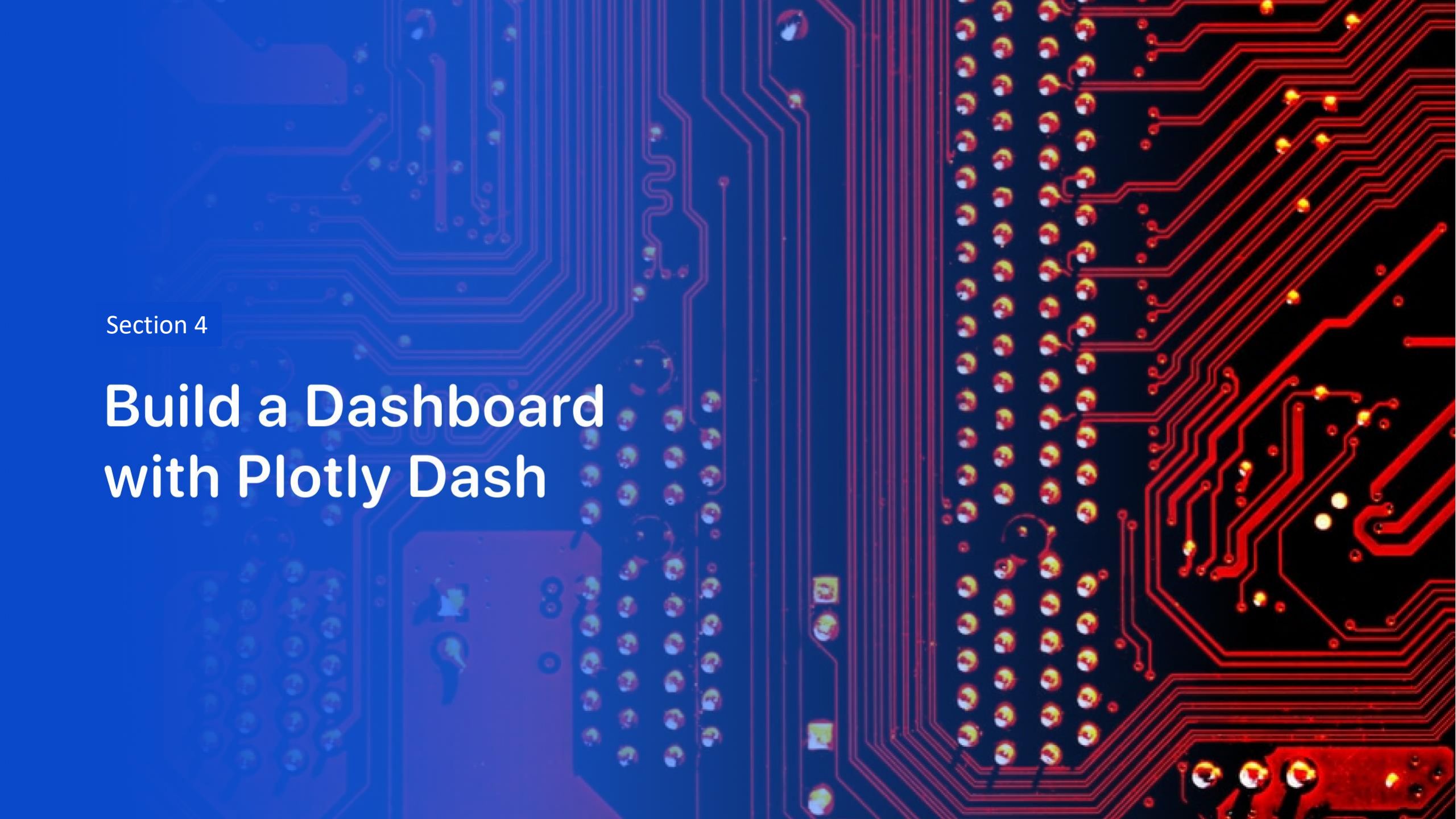
Out[100]:



KSC LC 39 A Distance Map

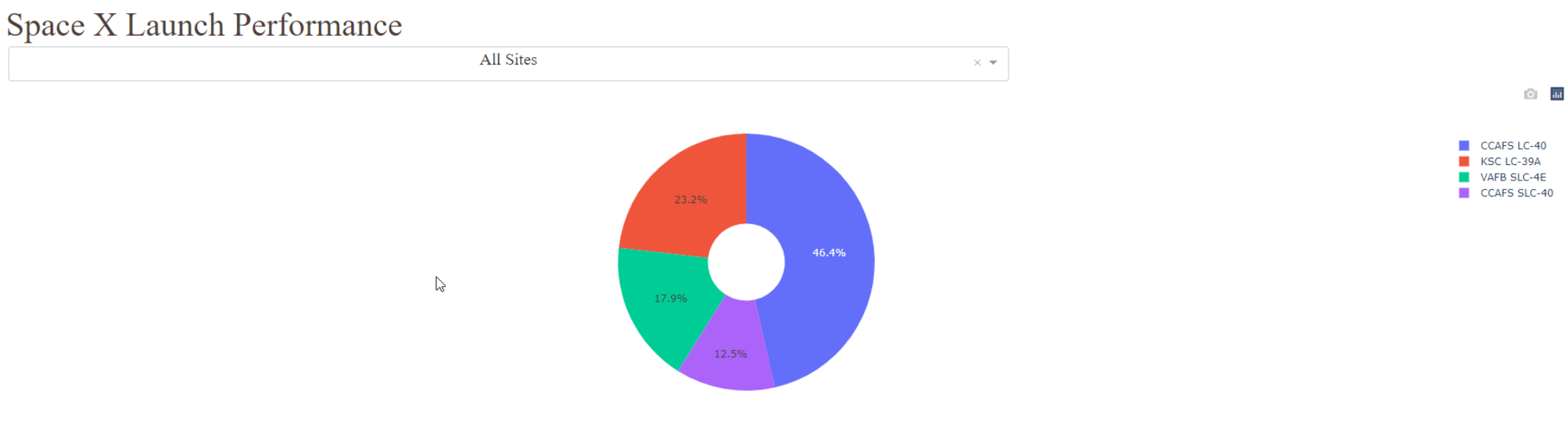
Out[124]:



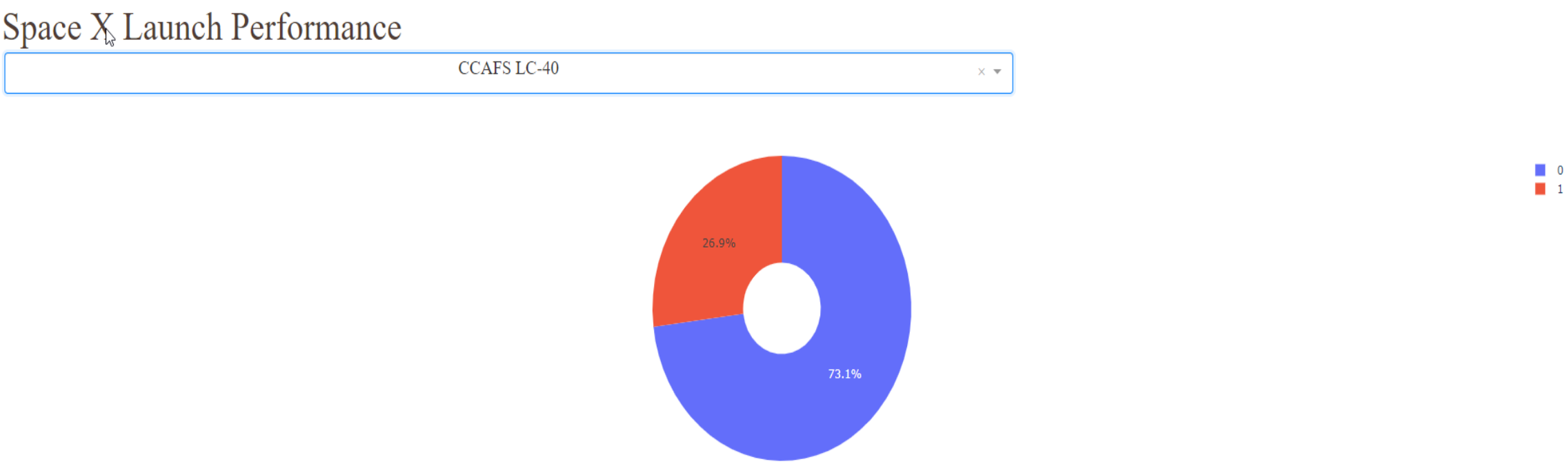

Section 4
Build a Dashboard with Plotly Dash

Launch Site Success Pie Chart

- The pie chart shows Launch Sites distribution based on the successful outcome

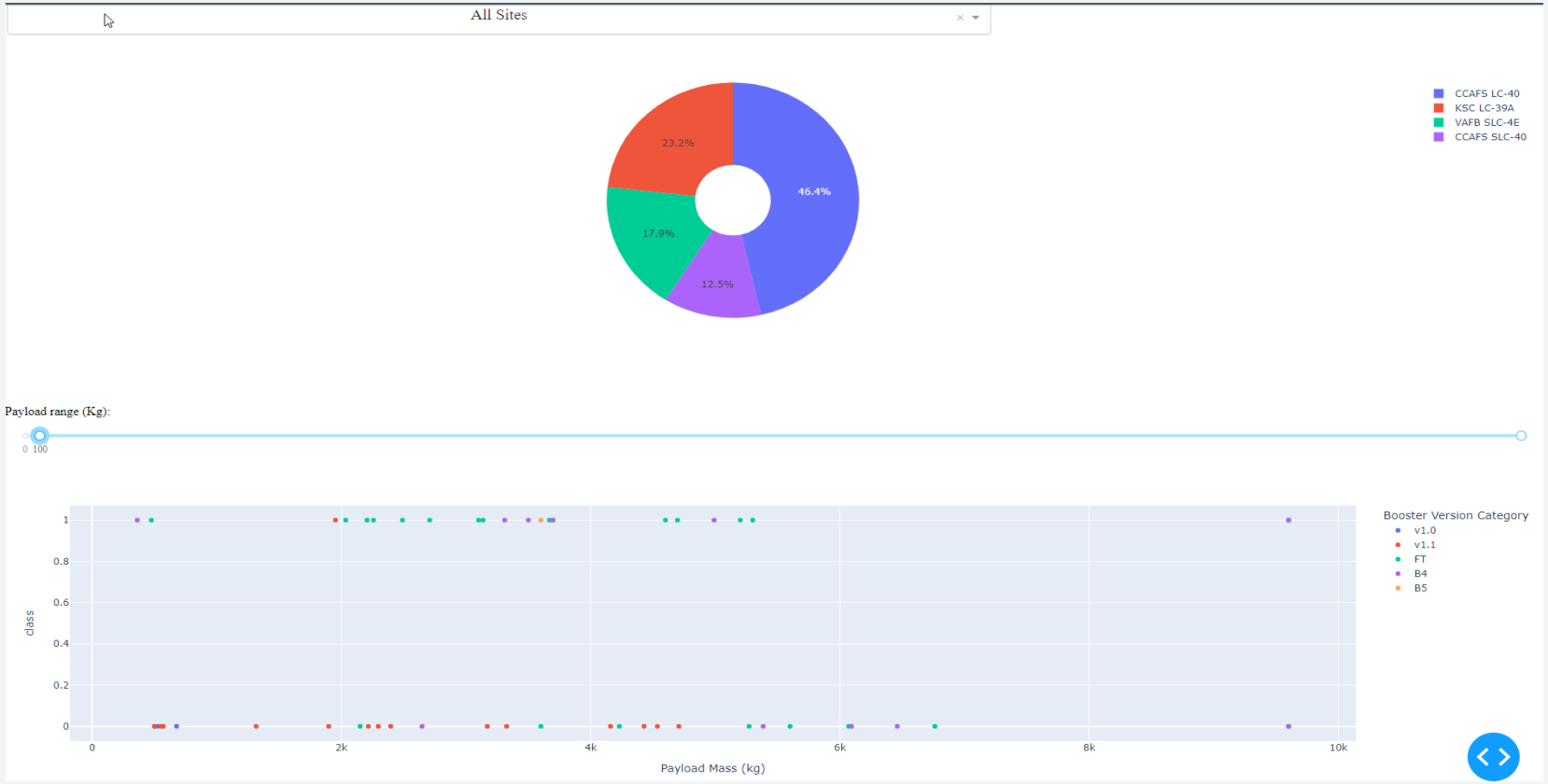


CCAFS LC-40 Distribution of Outcome



Payload Vs Launch Outcome Scatter Plot

- Scatter plot shows that for Booster Version Category of FT, had the largest success rate.



Section 5

Predictive Analysis (Classification)

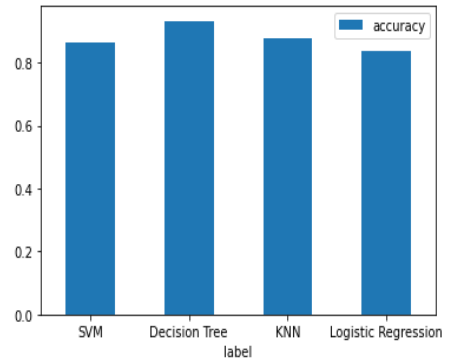
Classification Accuracy

- The accuracy is higher for Decision Tree

In [1]:

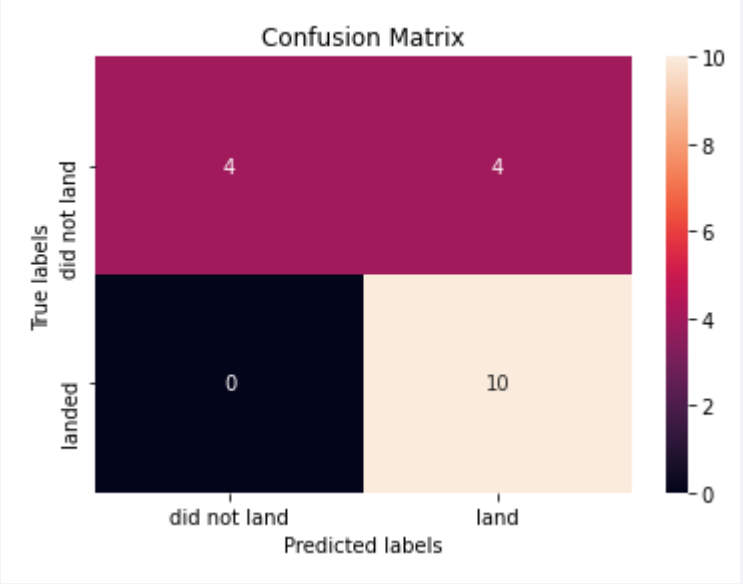
```
import pandas as pd

df = pd.DataFrame({'Model': ['SVM','Decision Tree', 'KNN', 'Logistic Regression'], 'accuracy':[0.8625, 0.9321428571428573, 0.8767857142857143,  0.8357142857142857]})
ax = df.plot.bar(x='Model', y='accuracy', rot=0)
```



Confusion Matrix

- Confusion Matrix of KNN Classifier Model. The TP = 4 and FP = 4 and TN = 0 and FN = 10.
  - So, the model accurately predicts:
  - 4 launches that did not land correctly,
  - 10 launches that landed correctly.
  - There was 0 incorrectly predicted successful launches
- The model incorrectly predicted 4 launches as successful, when they were unsuccessful.
- Of all the confusion Matrix for Decision Tree and SVM and Logistic Regression, KNN was the best algorithm predictions of the model from a confusion Matrix.



Conclusions

- Based on Accuracy Graph- Decision Tree had the highest accuracy for prediction.
- Based on the Confusion Matrix the best model algorithm showed similar results for KNN and SVM.
- KNN and SVM have similar high Jaccard score and high accuracy
- With this Classification Model we should be accurately be able to predict the future Success Outcome of Space X Launches.
- A successful Outcome predicts a lower cost of Launches.

Appendix

- All python Code Files have been added to this project.
- https://github.com/lakdee/coursera_capstone_ml_project

Thank you!