IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Keith McDonald
8/23/24

Outline

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

Executive Summary

- Publicly available data retrieved from SpaceX website API and Webscraping Wikipedia
- Data Evaluation included grphical analysis, SQL querries, geospatial analysis, and a Dash Board was created.
- Attributes that correlate with Successful or Failed landings were identified.
- Data was modelled using common classification models = Logistic Regression, Support Vector Machine, Decision Tree, K-Nearest Neighbors.
- The model that best predicts a Successful or Failed landing was the Decision Tree Model.

Executive Summary

- Summary of Models Evaluated and Accuracy Scores. Decision Tree1 model was the best performing with 89% accuracy.

Model	Score
Logistic Regression1	0.833
Logistic Regression with Grid Search	0.833
Support Vector Machine1	0.778
Support Vector Machine with Grid Search	0.833
Decision Tree1	0.889
Decision Tree with Grid Search	0.833
K-Nearest Neighbors1	0.778
K-Nearest Neighbors with Grid Search	0.833

Introduction

- SpaceX was the first to organization to successfully recover and reuse the 1st stage of a rocket.
- SpaceX has greatly reduced the cost of launching satellites because of their ability to recover and reuse the 1st stage of a rocket.
- The cost of launching a satellite or vehicle into orbit remains highly dependent on whether the 1st stage can be recovered.

- Can known data about a launch vehicle be used to predict whether 1st stage recovery will be successful?
- What data is needed to predict whether the 1st stage will be recovered?
- What models can be used to predict the 1st stage recovery?
- How accurate are the prediction models?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data collected from 2 sources from the period 2010 to 2020:
    - SpaceX API from https://api.spacexdata.com/v4/launches/past
    - Web Scraping from Wikipedia https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
  - Data included Launch Site, Payload Mass, Grid Fins, Legs, Booster Version, and Landing Outcome
- Perform data wrangling
  - Records with null data removed, Falcon 1 data removed
  - Evaluated data types to ensure compatible with models
  - One-Hot Encoding – Convert True/False to 1 / 0

Methodology

Executive Summary

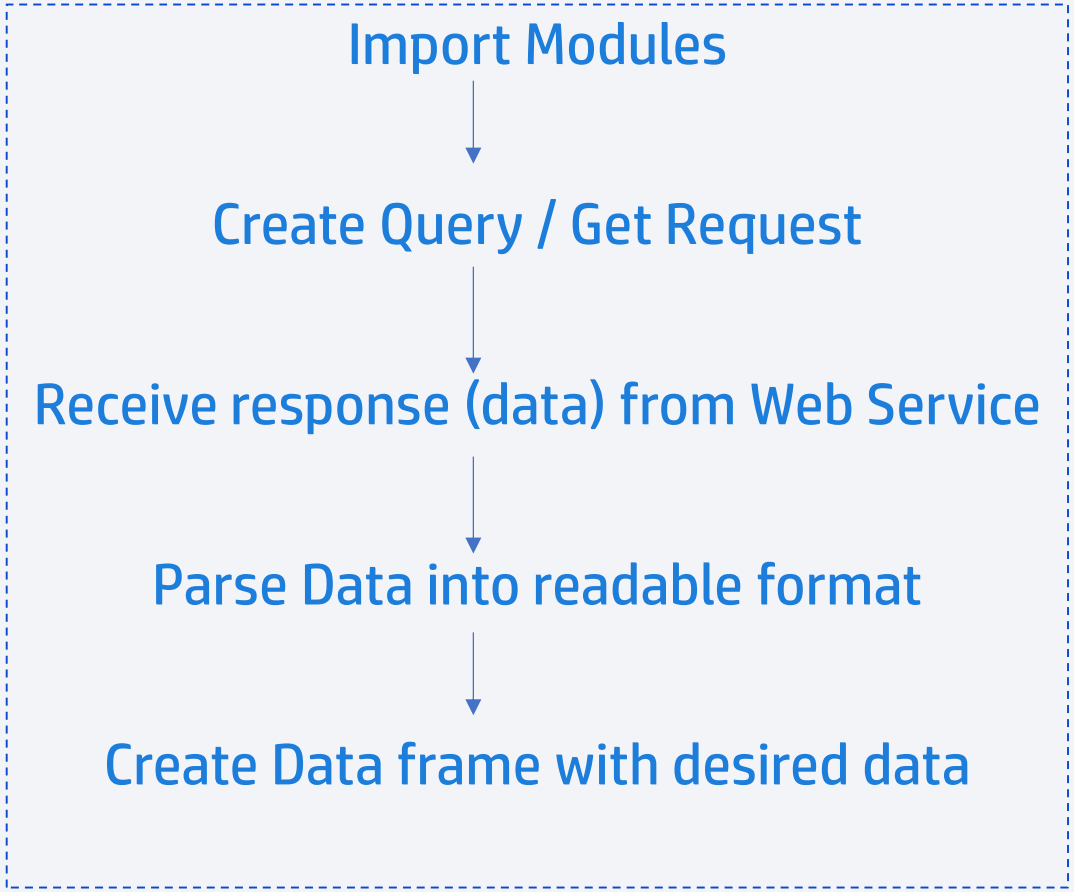
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - Load data into a Data Frames for independent and dependent variables
    - Standardize data
    - Split into Training and Test data.
    - Load data into Models
    - Evaluate models using Confusion Matrix and statistical scoring

Data Collection – SpaceX API

- Data collected from SpaceX REST API from

http://api.spacexdata.com/v4/launches/past

- Query data using requests.get(url)
- Once a successful response is received, data is parsed into readable data frame and desired data is tabulated using Pandas and Numpy modules.
- Data is saved in .csv format for future use.
- [IBM-Capstone/jupyter-labs-spacex-data-collection-api.ipynb at main · KNMcD/IBM-Capstone (github.com)]

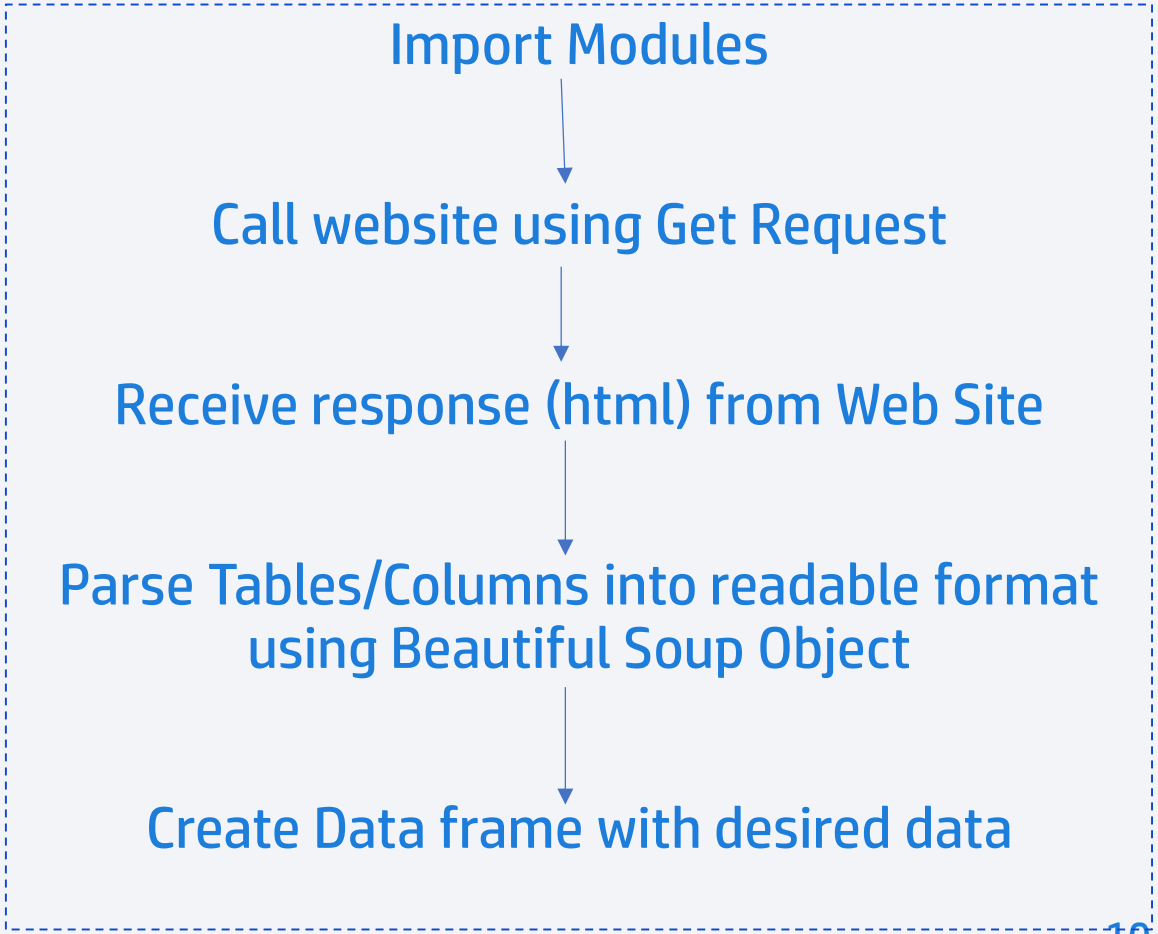


Data Collection - Scraping

- Data Scraped from website

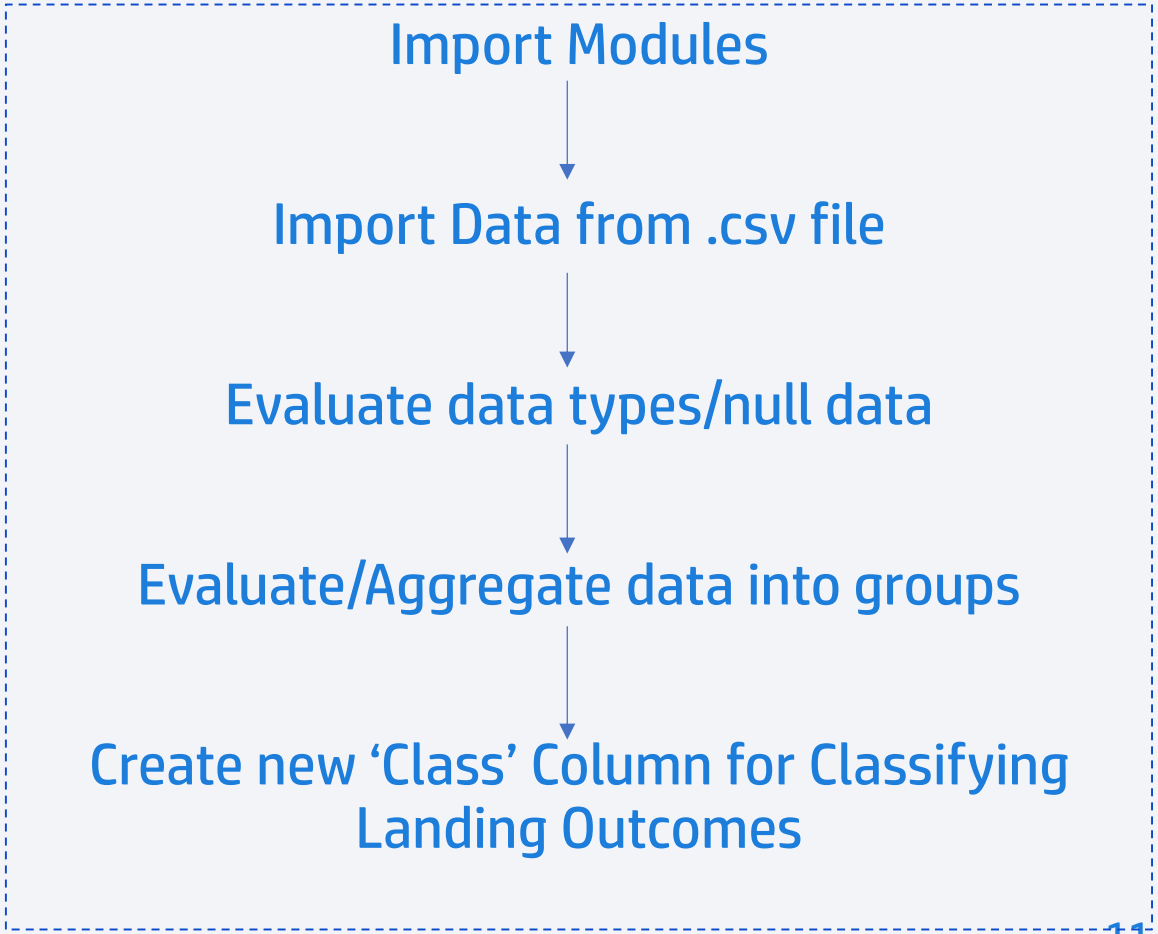
https://en.Wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

- Query data using requests.get(url)
- Once a successful response is received, data in html format is parsed using Beautiful Soup module
- Desired data is tabulated and given Column headings in a data frame using Pandas module.
- Data is saved in .csv format for future use.
- IBM-Capstone/jupyter-labs-webscraping.ipynb at main · KNMcD/IBM-Capstone (github.com)



Data Wrangling

- Data loaded from .csv file
- Evaluate data types and any missing data
- Evaluate the type/count of each
  - Launch Site
  - Orbit Type
  - Landing Outcomes
- Create a set of bad landing outcome
- Create new ‘Class’ Column for classification
  - Failed = 0
  - Successful = 1
- Save Data in .csv file
- IBM-Capstone/labs-jupyter-spacex-Data wrangling.ipynb at main · KNMcD/IBM-Capstone (github.com)



EDA with Data Visualization

- Summary of Charts used
  - Payload Mass vs. Flight Number with hue = Class
  - Launch Site vs. Flight Number with hue = Class
  - Launch Site vs. Payload Mass with hue = Class
  - Class Mean vs. Orbit
  - Orbit vs. Flight Number with hue = Class
  - Orbit vs. Payload Mass with hue = Class
  - Success Rate vs. Year
- IBM-Capstone/jupyter-labs-eda-dataviz.ipynb at main · KNMcD/IBM-Capstone (github.com)

EDA with SQL

- Summary of SQL Queries
  - Names of unique Launch Sites
  - 5 Records with Launch Site beginning with 'CCA'
  - Total Payload Mass launched by NASA (customer)
  - Average Payload Mass by booster version 'F9 v1.1'
  - Data of first successful landing on ground pad
  - Names of boosters with successful drone ship landings with Payload mass between 4000 and 6000 Kg.
  - Total number of Successful/Failed mission outcomes
  - Booster versions that carried the Maximum Payload mass
  - Months in 2015 where Landing Outcome was a drone ship failure
  - Rank Landing Outcomes by count between 6/4/2010 and 3/20/2017
- IBM-Capstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb at main · KNMcD/IBM-Capstone (github.com)

Build an Interactive Map with Folium

- Folium Map Objects used
  - Circles used to mark the Launch Site locations
  - Markers used to label the name of Launch Site locations - Vandenburg AF Base, Cape Canaveral, Kennedy Space Center, etc.
  - Marker Clusters used to mark the Launch Locations of Successful Launches (green) and Failed Launches (red)
- https://nbviewer.org/github/KNMcD/IBM-Capstone/blob/main/lab_jupyter_launch_site_location.ipynb
- IBM-Capstone/lab_jupyter_launch_site_location.ipynb at main · KNMcD/IBM-Capstone (github.com)

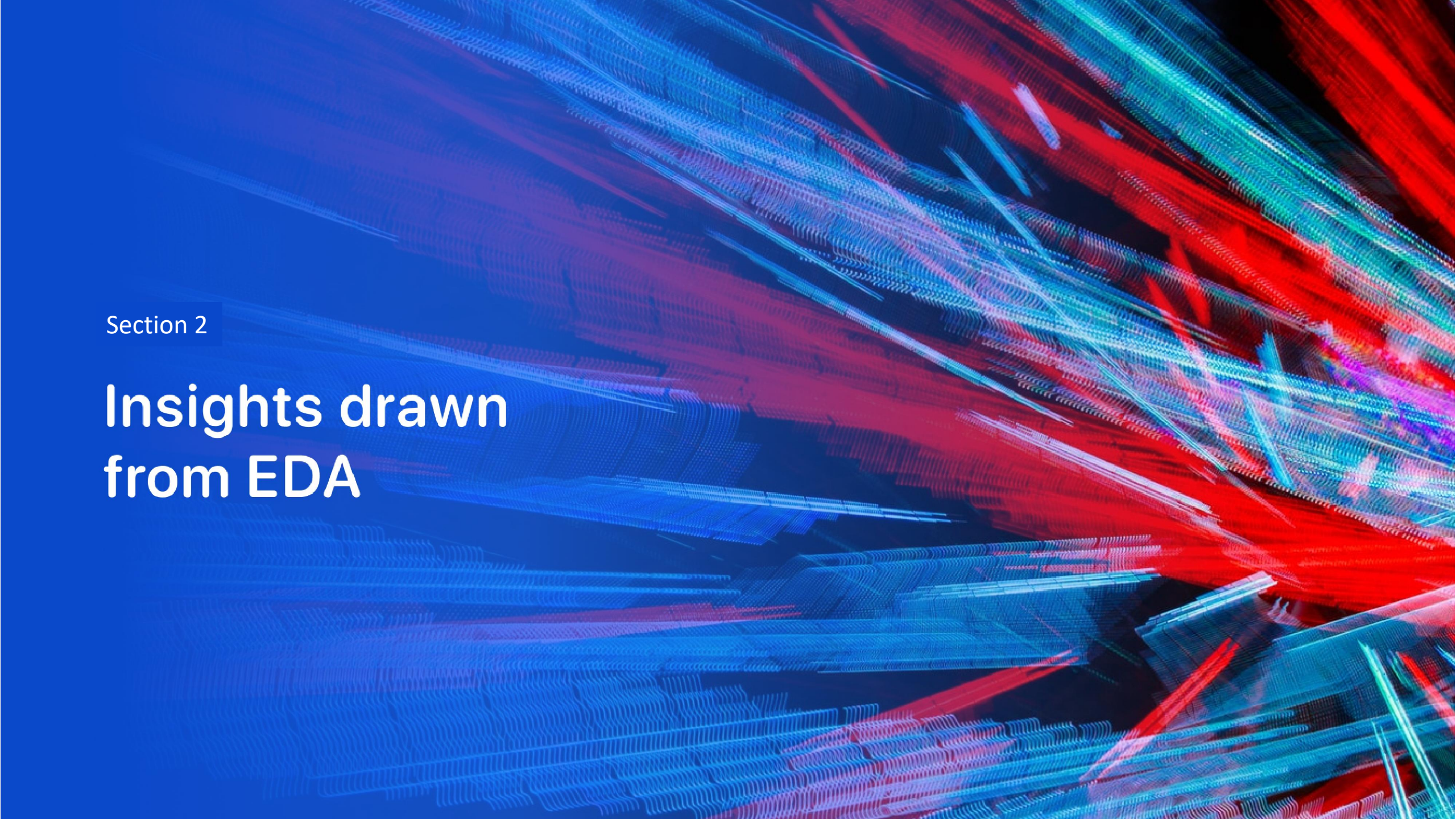
Build a Dashboard with Plotly Dash

- Dashboard Features Summary
  - Dropdown of Launch Sites
  - When All Sites selected displays Pie Chart with Percentage of Successful launches by Site
  - When specific Launch Site selected displays Pie Chart with Percentage breakdown of Successful/Failed launches
  - Scatter Chart of Successful Launches versus Payload Mass for All Sites or selected Launch Site
    - Featured with Range Slider to specify Minimum and Maximum Payload Mass
- Dropdown and Range Slider allow analysis of Successful / Failed based on Launch Site and Payload Mass.
- IBM-Capstone/dash_interactivity.py at main · KNMcD/IBM-Capstone (github.com)

Predictive Analysis (Classification)

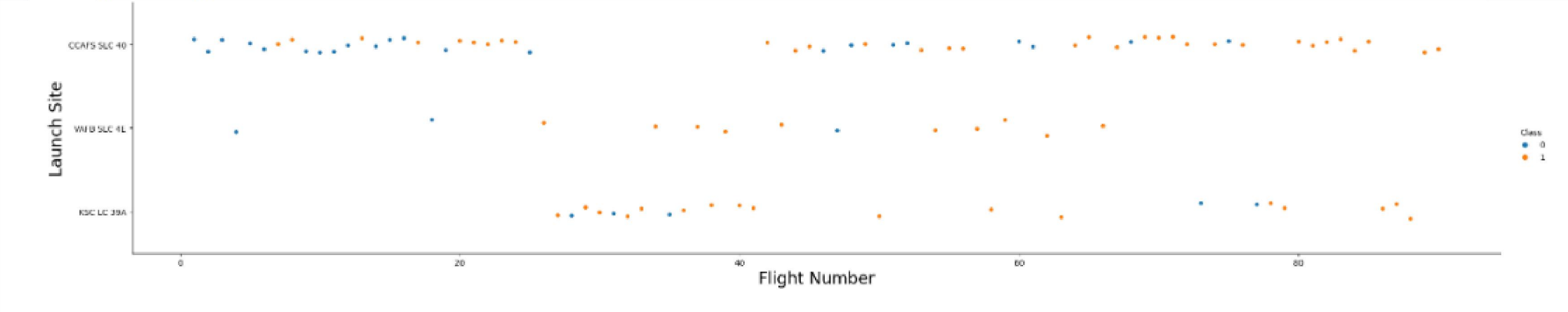
- Classification Analysis
  - Data loaded and split into Independent variables (X) and Dependent variables (Y)
  - Data Standardized using Standard Scalar function.
  - Data split into train and test data sets with 20% randomly selected as test data.
  - Training data used to train model with and without utilizing GridSearchCV function.
    - Models included - Logistic Regression, Support Vector Machine, Decision Tree Classifier, K-Nearest Neighbors,

- [IBM-Capstone/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite2.ipynb at main · KNMcD/IBM-Capstone (github.com)]


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

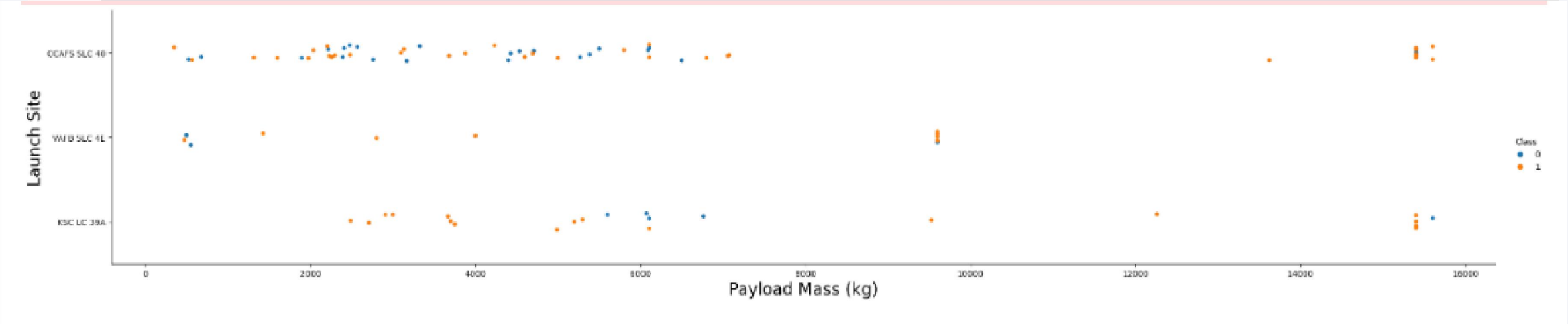
```
[4]: # Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



- Successful launches (Class = 1) indicated in orange, Failed launches (Class=0) indicated in blue.
- Later flights were more likely to be successful. Earliest flights were mostly failures.
- Largest number of flights were from Cape Canaveral with least number of flights from Vandenburg AF Base.

Payload vs. Launch Site

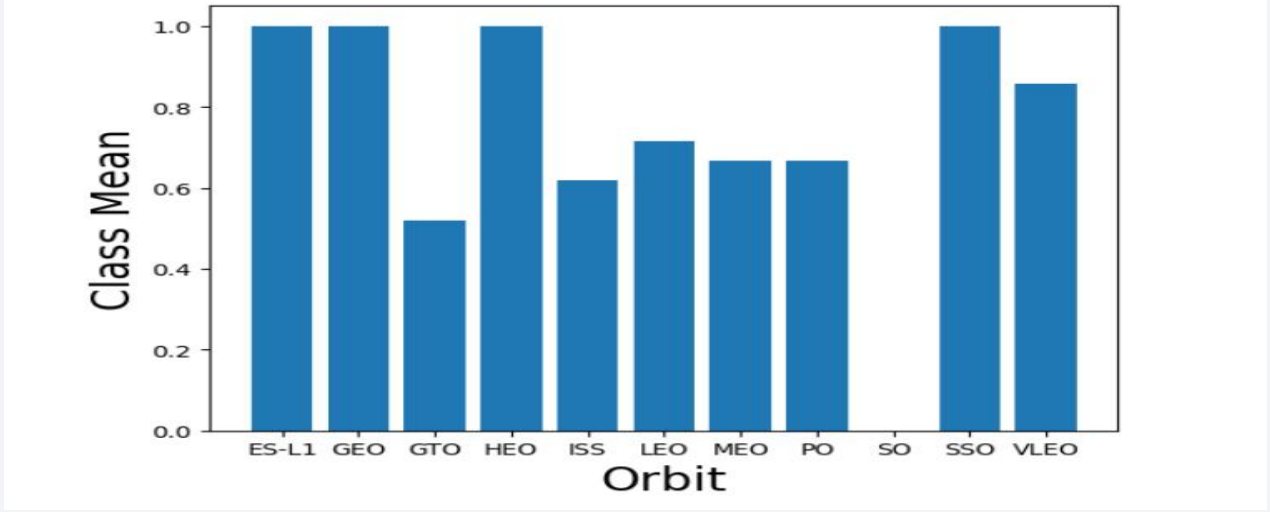
```
[5]: # Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Payload Mass (kg)",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



- Successful launches (Class = 1) indicated in orange, Failed launches (Class=0) indicated in blue.
- Heavier Payload Mass were more likely to be Successful
- No launches from Vandenburg AF Base greater than 10,000 Kg.
- Majority of Payloads are less than 10,000 Kg.

Success Rate vs. Orbit Type

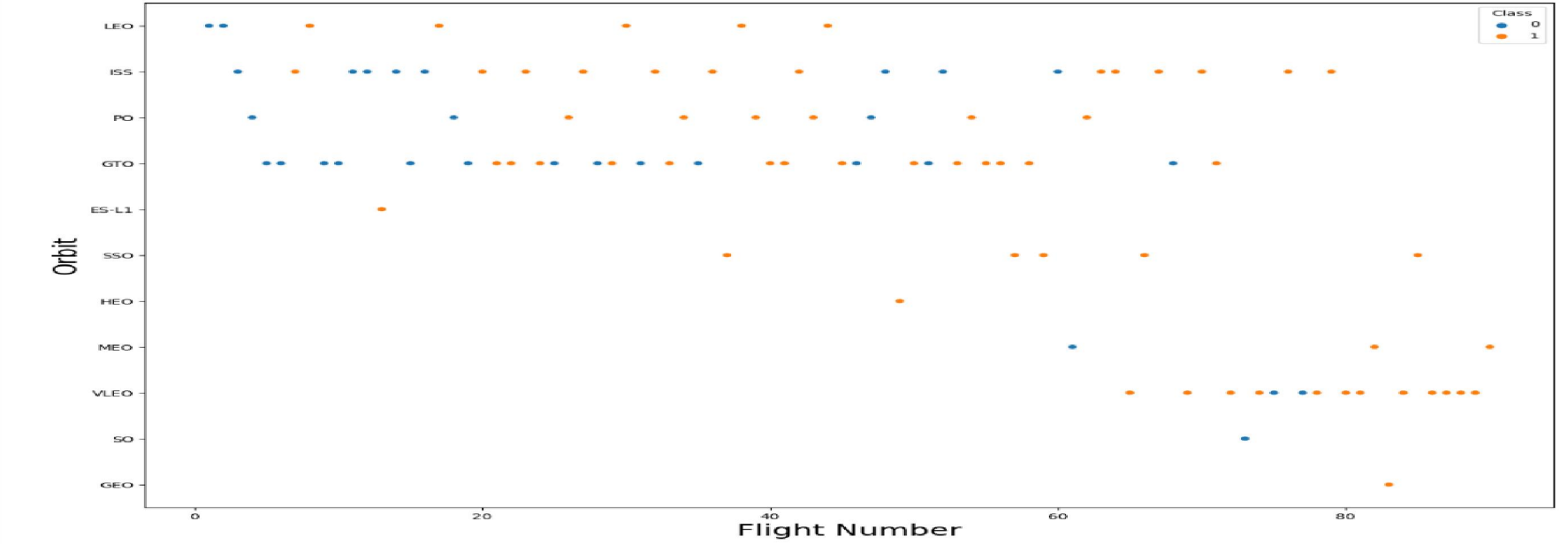
```
[8]: df_Orbitgrp = df.groupby(['Orbit'],as_index=False).mean(['Class'])
     x = df_Orbitgrp['Orbit']
     height = df_Orbitgrp['Class']
     plt.bar(x, height)
     plt.xlabel("Orbit",fontsize=20)
     plt.ylabel("Class Mean",fontsize=20)
     plt.show()
```



- Orbit types that are most likely to be Successful – ES-L1, GEO, HEO, SSO
- Orbit type that is least likely to be Successful – GTO
- No SO Orbit types were attempted.

Flight Number vs. Orbit Type

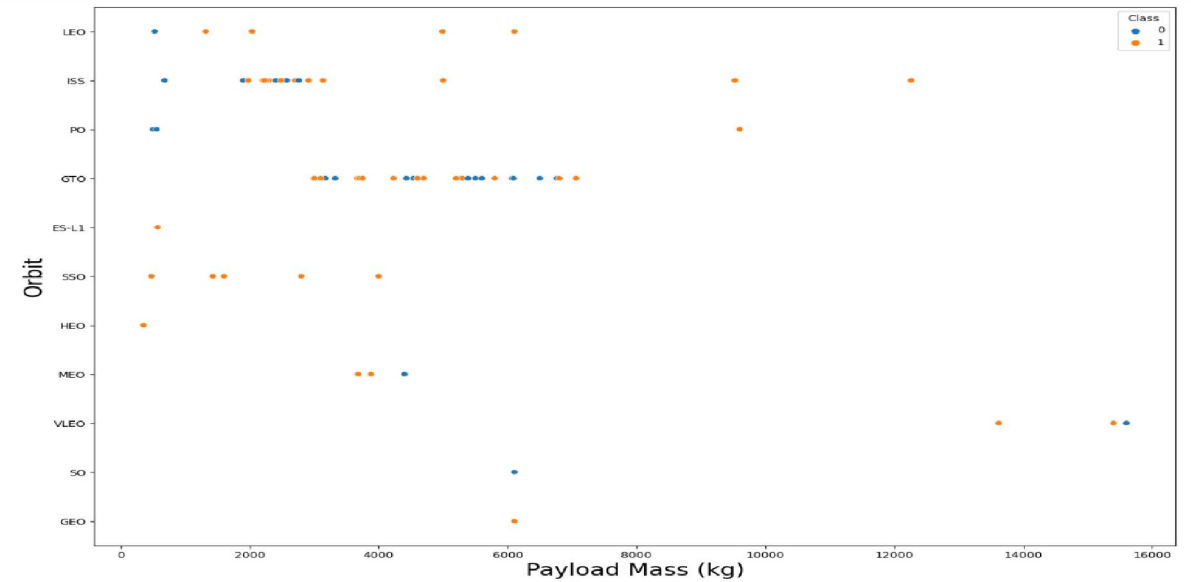
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
plt.figure(figsize=(16,12))
x = df['FlightNumber']
y = df['Orbit']
hue = df['Class']
sns.scatterplot(y="Orbit", x='FlightNumber', hue="Class", data=df)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



- Successful launches (Class = 1) indicated in orange, Failed launches (Class=0) indicated in blue.
- Later flights were more likely to be Successful. GTO Orbit least likely to be Successful

Payload vs. Orbit Type

```
# 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
plt.figure(figsize=(16,12))
sns.scatterplot(y="Orbit", x='PayloadMass', hue="Class", data=df)
plt.xlabel("Payload Mass (kg)",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

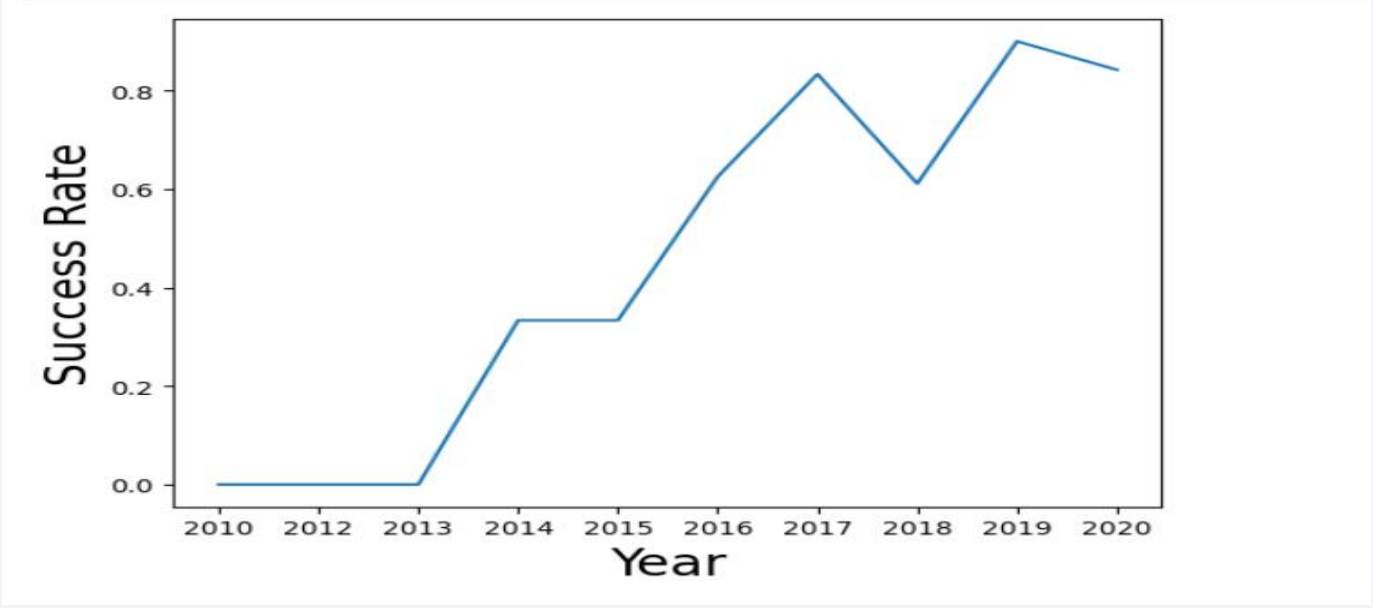


- Successful launches (Class = 1) indicated in orange, Failed launches (Class=0) indicated in blue.
- Heavier Launches more likely to be successful.

Launch Success Yearly Trend

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate

df['Year'] = Extract_year(df['Date'])
df_Success = df.groupby(['Year'],as_index=False).mean(['Class'])
x = df_Success['Year']
y = df_Success['Class']
plt.plot(x, y)
plt.xlabel("Year",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```



- Success Rate improves over time starting in 2013.
- Success Rate improves to > 80% by 2019.

All Launch Site Names

```
%sql Select DISTINCT("Launch_Site") from SPACEXTABLE
 * sqlite:///my_data1.db
Done.
```

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

- There are 4 unique Launch Sites.

Launch Site Names Begin with 'CCA'

```
%sql Select * from SPACEXTABLE where "Launch_Site" like 'CCA%' Limit 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

- First 5 records with Launch Site starting with ‘CCA’

Total Payload Mass

```
%sql Select SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE "Customer" = 'NASA  (CRS)'
```

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

SUM(PAYLOAD_MASS__KG_)
45596

- Total Payload Mass in Kg carried with NASA as the Customer.

Average Payload Mass by F9 v1.1

```
%sql Select AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE "Booster_Version" LIKE 'F9 v1.1%'
 * sqlite:///my_data1.db
Done.
```

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

- Average payload mass carried by booster version F9 v1.1 in Kilograms.

First Successful Ground Landing Date

```
%sql SELECT "Date" from SPACEXTABLE WHERE "Landing_Outcome" = 'Success (ground pad)' ORDER BY "Date" Limit 1
```

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

Date
2015-12-22

- First Successful ground landing – Query selects all Successful ground landings, then orders by date ascending, limits query to the first record.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT DISTINCT("Booster_Version") from SPACEXTABLE WHERE "Landing_Outcome" = 'Success (drone ship)' AND
("Payload_Mass__KG_" BETWEEN '4000' AND '6000')
```

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

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

- Boosters which have successfully landed on drone ship and had payload mass 4000 - 6000 Kg.

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT "Mission_Outcome", COUNT(*) FROM SPACEXTABLE GROUP BY "Mission_Outcome"
```

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

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

- Total number of successful and failure mission outcomes

Boosters Carrying Maximum Payload

```
%sql SELECT DISTINCT("Booster_Version") FROM SPACEXTABLE WHERE "Payload_Mass__KG_" = (SELECT MAX("Payload_Mass__KG_") FROM SPACEXTABLE)
```

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

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

- 12 different booster versions have carried the maximum payload mass

2015 Launch Records

```
%sql SELECT substr(Date,6,2), "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Failure (drone ship)' AND substr(Date,0,5)='2015'
```

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

substr(Date,6,2)	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- Failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

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

```
%sql SELECT "Landing_Outcome", COUNT("Landing_Outcome") FROM SPACEXTABLE GROUP BY "Landing_Outcome"
HAVING "Date" BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY COUNT("Landing_Outcome") DESC
```

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

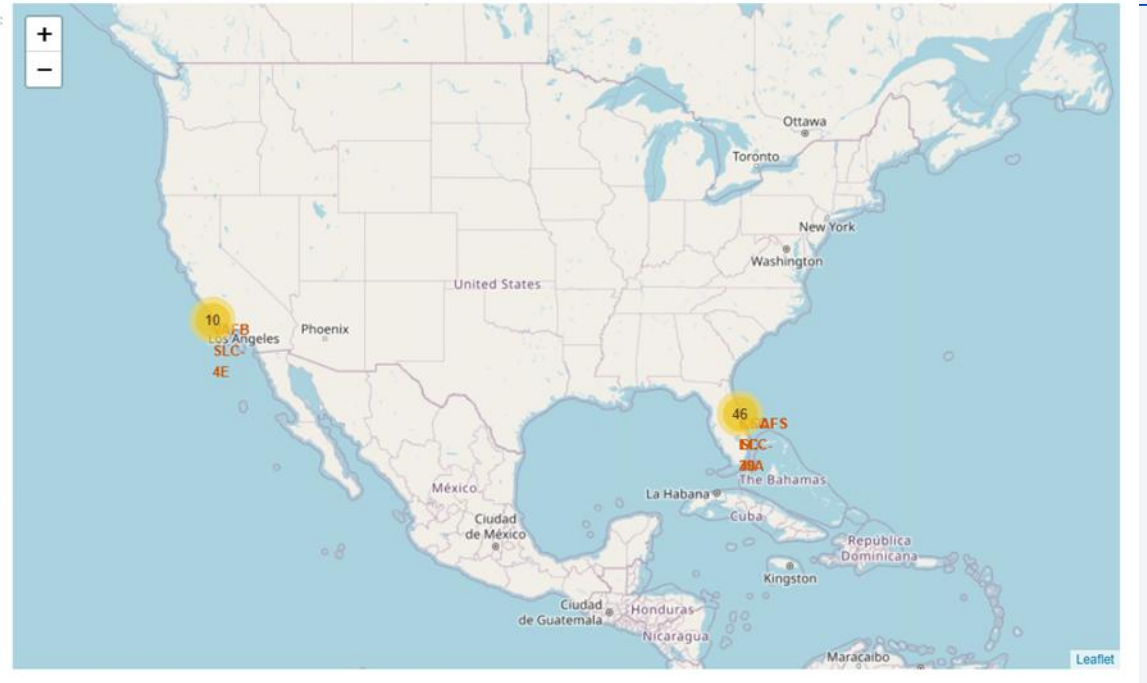
Landing_Outcome	COUNT(Landing_Outcome)
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

- 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

Section 3

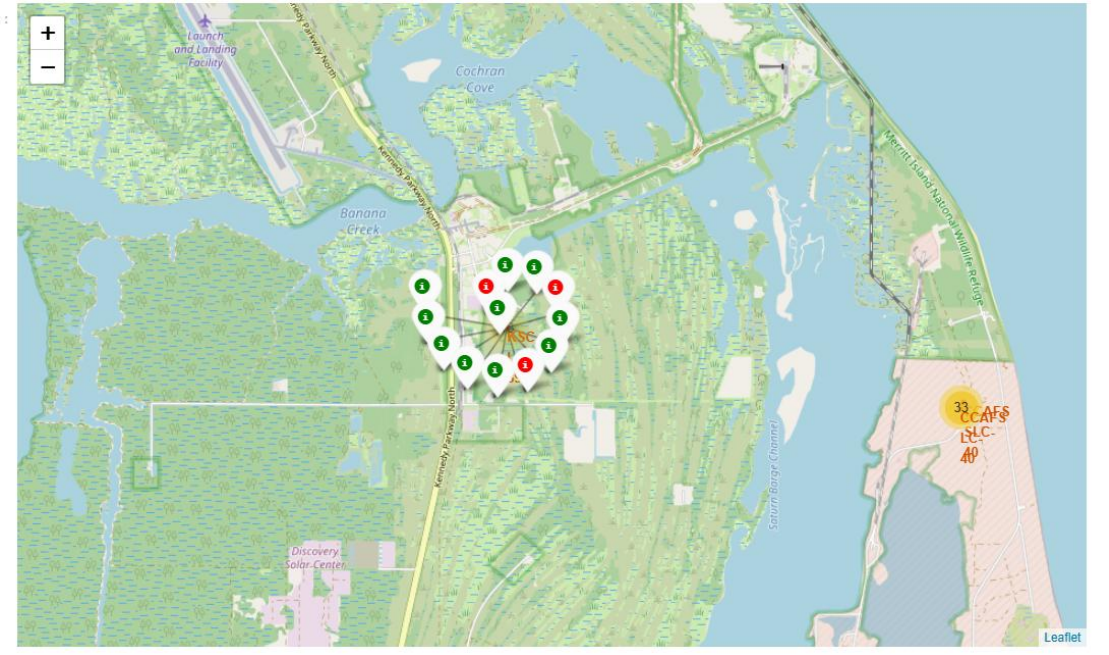
Launch Sites Proximities Analysis

Space X Launch Sites – World Map



- Map showing Marker Clusters of Launch Sites at Vandenburg AF Base, Cape Canaveral, Kennedy Space Center Pad 39A and 39B.

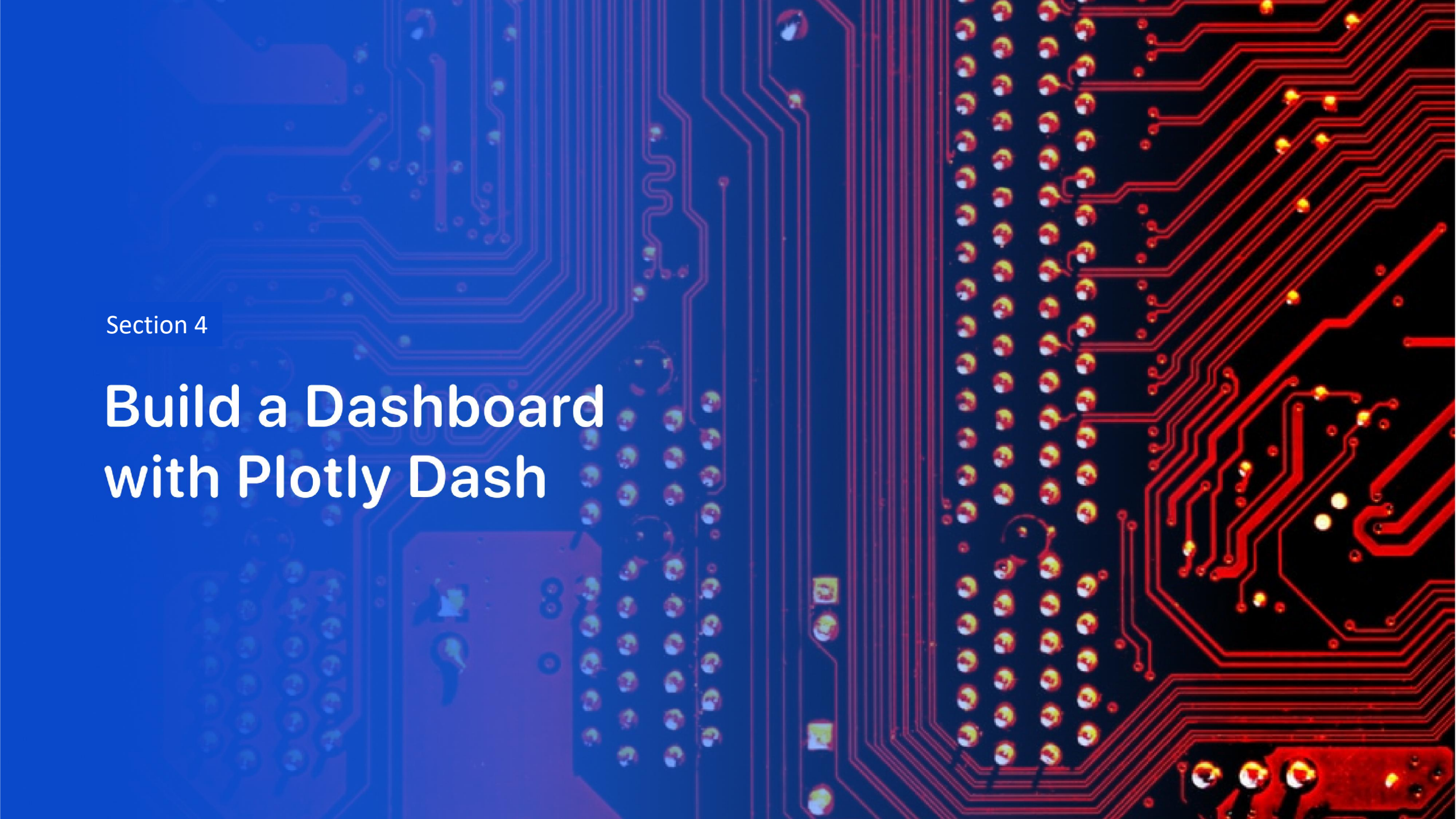
Space X Launch Sites – World Map



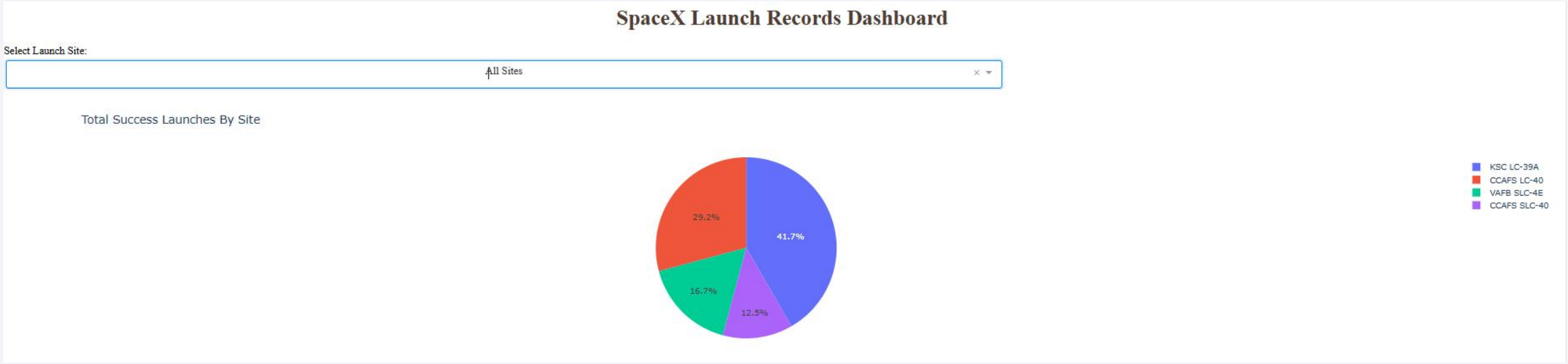
- Folium map showing Kennedy Space Center launch Site with Successful launches (green) and Failed launches (red).
- Data column for marker color was generated.
- For loop iterates through marker colors to place appropriate colored marker on the map.

Section 4

Build a Dashboard with Plotly Dash

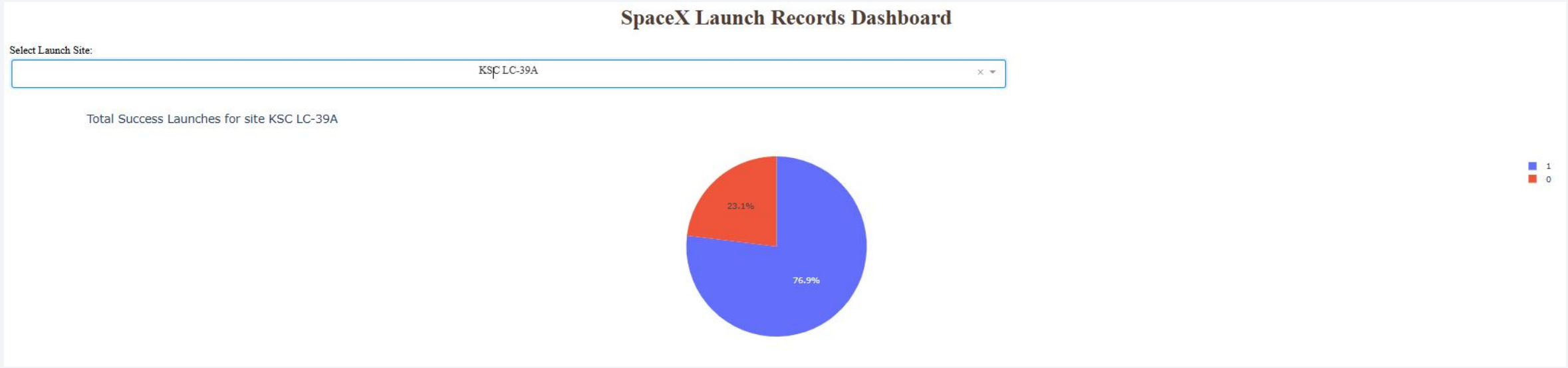


SpaceX Launch Records Dashboard



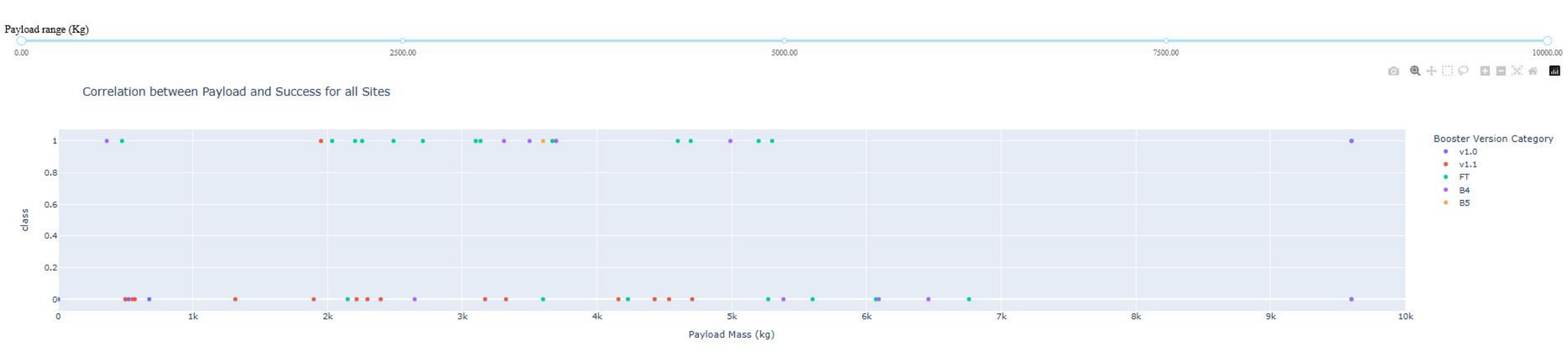
- Pie Chart showing Successful launches grouped by Launch Site
- Legend to right shows Launch Site color code.
- Largest percentage of Successful launches are from KSC-LC39A

Dashboard updated with KSC LC-39A Selected



- Pie Chart of KSC LC-39A showing percentage of Successful and Failed Launches
- Legend to right indicates Success = 1 (blue), Failed = 0 (red).

Scatter Plot of Payload Mass vs. Launch Outcome

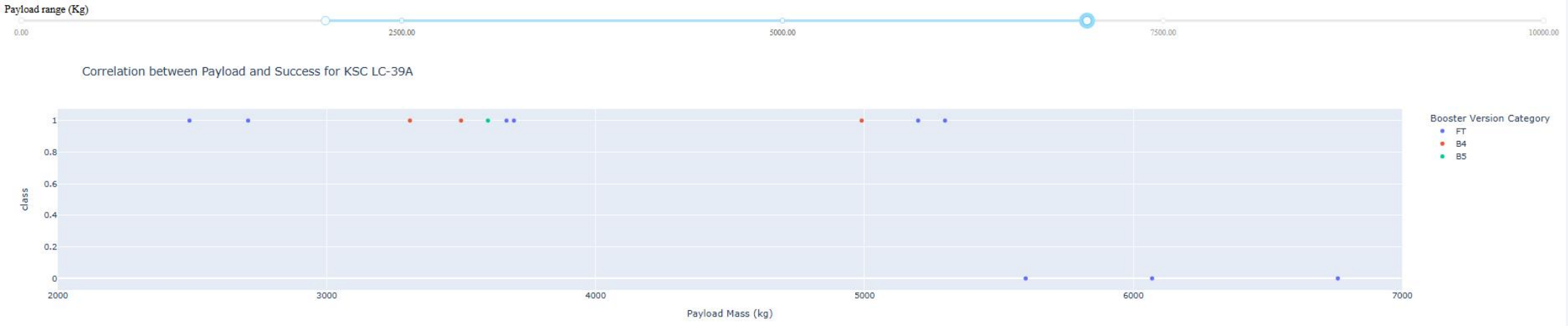


- All sites selected showing most Payload Masses are less than 7000 Kg.
- Legend on right shows Booster Version color code.
- Most Successful Booster Version is FT.

Dashboard Updated showing various Payload Ranges

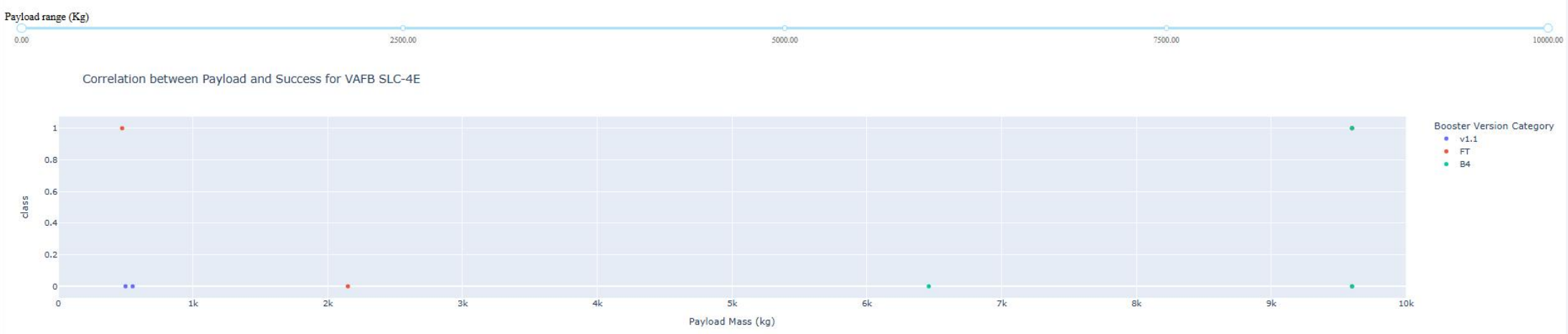
- Least Successful Payload range is approximately 5000 Kg to 7500 Kg.
- Most Successful Payload range is approximately 1500 Kg to 4000 Kg.

Most Successful Launch Site – KSC LC-39A



- Scatter Plot updated with Launch Site KSC LC-39A showing all Payloads ranged from 2000 Kg to 7000 Kg.
- 10 Successful launches out of a total of 13 launches.
- Booster Versions B4 and B5 had no Failed launches from KSC LC-39A.

Least Successful Launch Site - VAFB



- Scatter Plot updated with Launch Site VAFB showing all Payloads ranged from 0 Kg to 10,000 Kg.
- 2 Successful launches out of a total of 7 launches.

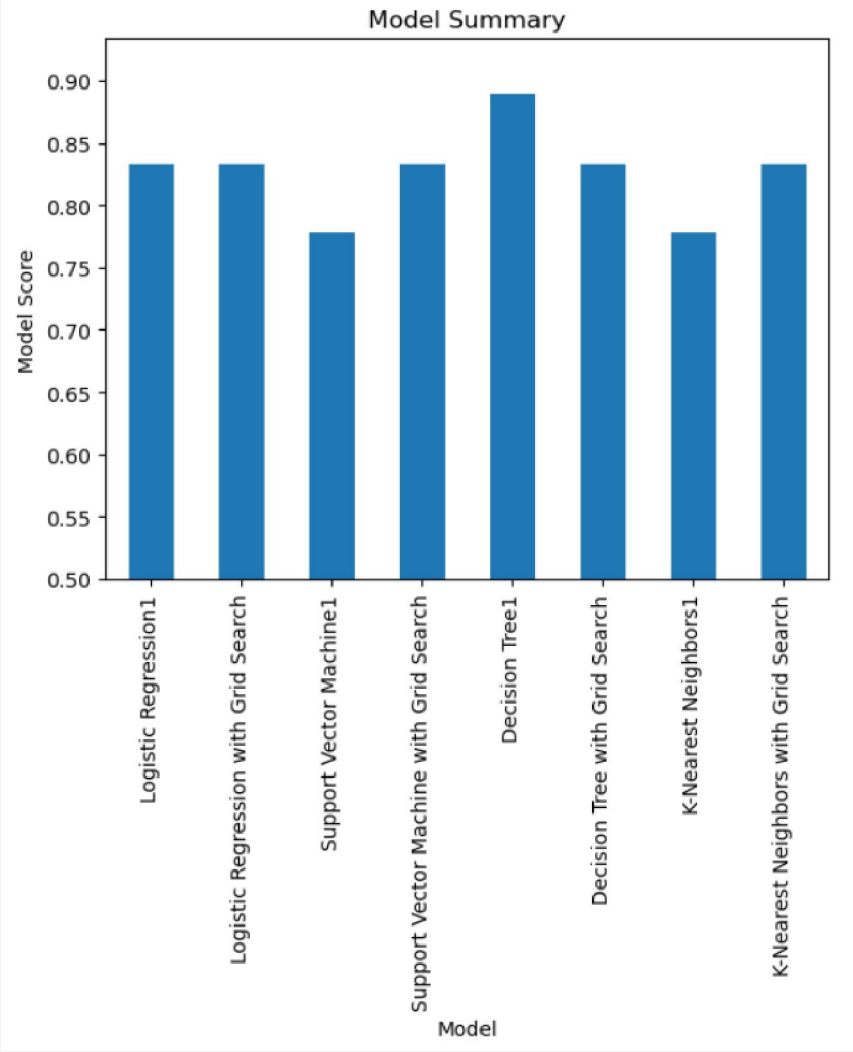
Section 5

Predictive Analysis (Classification)

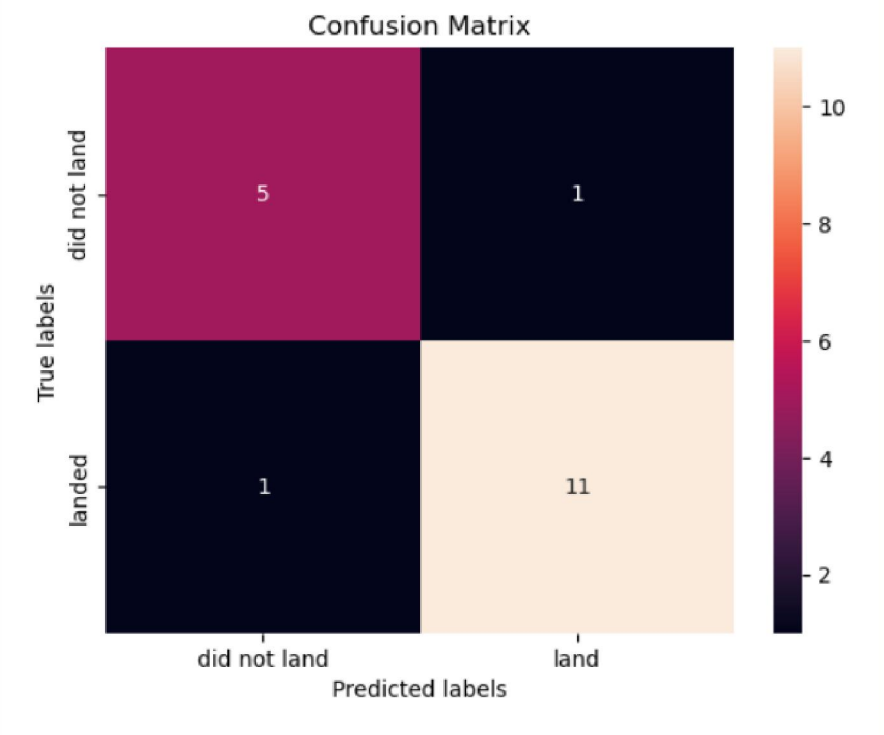
Classification Accuracy

	Model	Score
0	Logistic Regression1	0.833
1	Logistic Regression with Grid Search	0.833
2	Support Vector Machine1	0.778
3	Support Vector Machine with Grid Search	0.833
4	Decision Tree1	0.889
5	Decision Tree with Grid Search	0.833
6	K-Nearest Neighbors1	0.778
7	K-Nearest Neighbors with Grid Search	0.833

- Model Accuracy Summary table and Bar Chart
- Decision Tree Model with default parameters was the most accurate at classifying Launch Success.



Confusion Matrix




```
tree1.get_params()
```

```
{'ccp_alpha': 0.0,
 'class_weight': None,
 'criterion': 'gini',
 'max_depth': None,
 'max_features': None,
 'max_leaf_nodes': None,
 'min_impurity_decrease': 0.0,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'random_state': None,
 'splitter': 'best'}
```

- Decision Tree Confusion Matrix showing actual versus predicted results for the Decision Tree Model using the following model parameters to the right.

Conclusions

- Falcon 9 booster landing success can be predicted with approximately 89% accuracy using Decision Tree Classification Model.
- Important attributes that affect landing success are the Launch Site, Payload Mass, Orbit, and booster version.
- Landing Success has improved over time with > 80% success rate after 2019.

Appendix

```
lr1_score = round(lr1.score(X_test,Y_test),3)
logreg_score = round(logreg_cv.score(X_test, Y_test),3)
svm1_score = round(svm1.score(X_test,Y_test),3)
svm_score = round(svm_cv.score(X_test, Y_test),3)
tree1_score = round(tree1.score(X_test,Y_test),3)
tree_score = round(tree_cv.score(X_test, Y_test),3)
knn1_score = round(knn1.score(X_test,Y_test),3)
knn_score = round(knn_cv.score(X_test, Y_test),3)

model1 = 'Logistic Regression1'
model2 = 'Logistic Regression with Grid Search'
model3 = 'Support Vector Machine1'
model4 = 'Support Vector Machine with Grid Search'
model5 = 'Decision Tree1'
model6 = 'Decision Tree with Grid Search'
model7 = 'K-Nearest Neighbors1'
model8 = 'K-Nearest Neighbors with Grid Search'

Data = [[model1, lr1_score],[model2, logreg_score],[model3, svm1_score],[model4, svm_score],
        [model5, tree1_score],[model6, tree_score],[model7, knn1_score],[model8, knn_score]]

Report = pd.DataFrame(Data, columns=['Model', 'Score'])
```

```
import matplotlib.pyplot as plt


Report['Score'].plot(kind='bar')
plt.title('Model Summary')
plt.xlabel('Model')
plt.ylabel('Model Score')
plt.ylim(0.5)
plt.xticks(rotation = 90)
plt.show()
```

- Code used to Summarize and plot Model results.

Thank you!