IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Dylan S.
1/29/25

Agenda

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

Executive Summary

- A variety of preprocessing, exploratory, visual and machine learning techniques were applied to SpaceX launch data to predict future landing outcomes, identifying variables of interest such as Booster Version, Payload Mass, and Launch Site (among others). Using GridSearchCV to find the best hyperparameter tunings, we constructed four predictive models that could predict whether future SpaceX launches would land successfully.
- Of the four constructed models (K-Nearest Neighbor, Logistic Regression, Support Vector Machine and Decision Tree) we found that the decision tree classifier was most accurate in predicting launch outcomes with an accuracy of 87.7%.

Introduction

- According to Joseph Santarcangelo and the IBM Corporation, “SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch.”
- The dataset contains historical launch data from SpaceX, accessed through the SpaceX REST API and web scraping. Stored as a csv file through IBM Skills Network Cloud Storage for ease of access and manipulation.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - The data was collected using a combination of REST API and webscraping requests from https://api.spacexdata.com and https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- Performed data wrangling
  - Data was cleaned, standardized, and one-hot encoded before fitting to the models.
- Performed exploratory data analysis (EDA) using visualization and SQL
- Performed interactive visual analytics using Folium and Plotly Dash
- Performed predictive analysis using classification models
  - We created a train/test split on the data set and fit it to each tuned ML model. We assessed each model's accuracy using scikit-learn's score and best_score_ methods.

Data Collection

- The data was collected using a combination of REST API and web scraping tools (requests and BeautifulSoup libraries) to harvest data from https://api.spacexdata.com and https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Data Collection – SpaceX API

- Here is the URL from which we extracted the launch data and subsequently converted into a dataframe
- The complete notebook can be viewed here: IBM-Capstone/jupyter-labs-spacex-data-collection-api.ipynb at Data-Analytics · newmetagetright/IBM-Capstone

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

```
# Use json_normalize meethod to convert the json result into a dataframe
repo=response.json()
data=pd.json_normalize(repo)
```

Data Collection - Scraping

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup=BeautifulSoup(response.text, 'html.parser')
```

```
column_names = []
columns=first_launch_table.find_all('th')
for column in columns:
    name=extract_column_from_header(column)
    if name is not None and len(name)>0:
        column names.append(name)
```

- Firstly, we established a BeautifulSoup object to parse the html response data. Then we iterated through the table headers to grab the column names. After that, we established an empty dictionary and appended the values from the table rows of the soup object into it, so that we could transform it into an operable dataframe.

- The complete notebook can be viewed here: IBM-Capstone/jupyter-labs-webscraping.ipynb at Data-Analytics · newmetagetright/IBM-Capstone

Data Wrangling

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

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

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
```

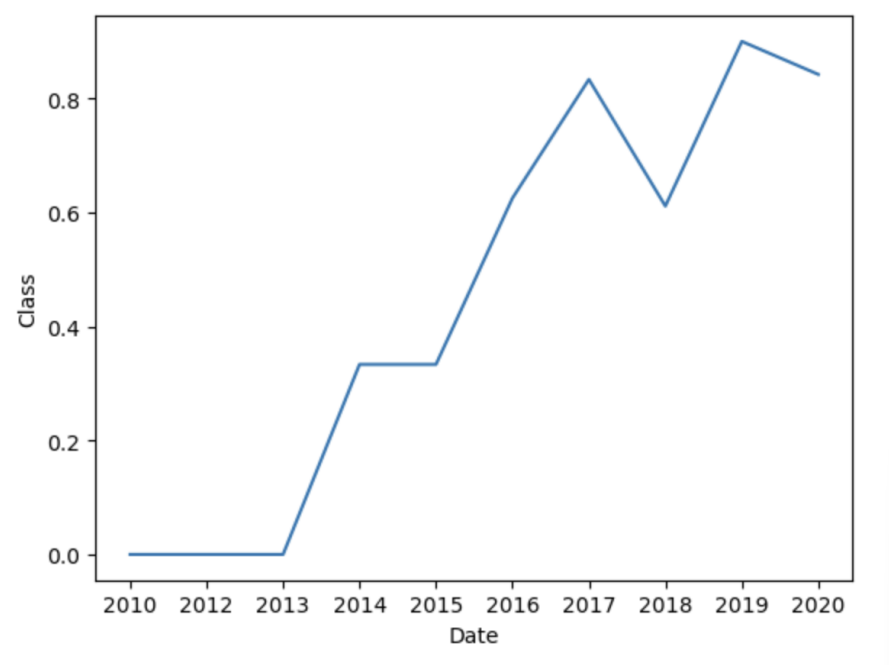
```
# landing_outcomes = values on Outcome column
landing_outcomes=df['Outcome'].value_counts()
landing_outcomes
```

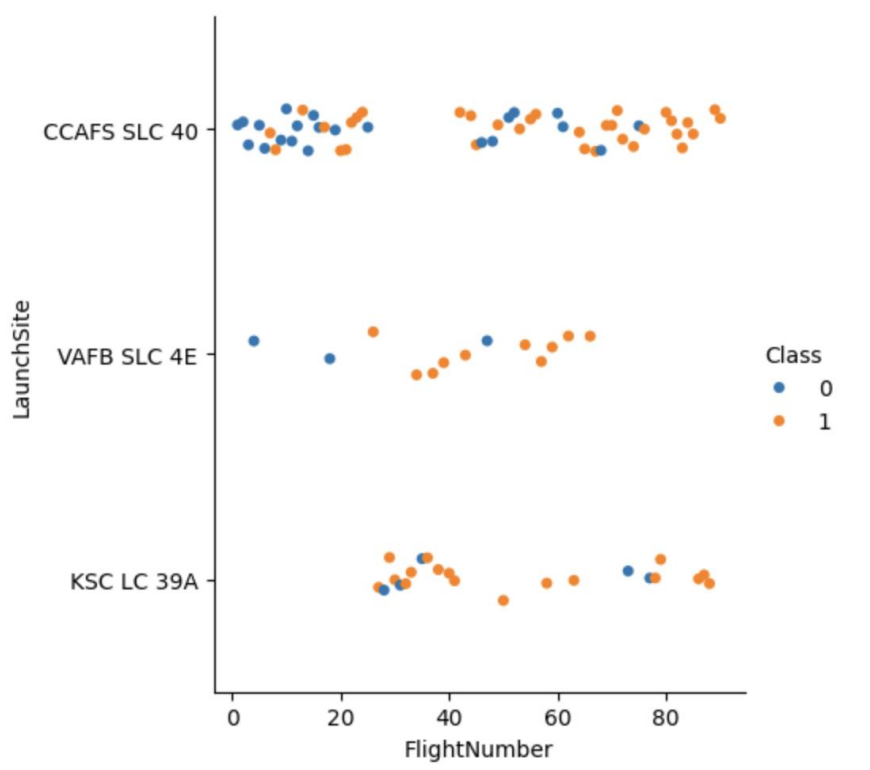
```
Outcome
True ASDS       41
None None       19
True RTLS       14
False ASDS       6
True Ocean       5
False Ocean      2
None ASDS        2
False RTLS       1
Name: count, dtype: int64
```

- Number of launches and landing outcomes were noted. We created a new dataframe column from a variable that assigns a binary value based on whether the landing outcomes were successful.
- The completed notebook can be viewed here: [IBM-Capstone/labs-jupyter-spacex-Data wrangling.ipynb at Data-Analytics · newmetagetright/IBM-Capstone]

EDA with Data Visualization

- We plotted the success rates of landing outcomes over time, as well as landing outcomes by launch site and flight number.





- This gave us a good idea of how the success rates for landing outcomes evolved over time and helped serve as a good predictor for our models.
- The completed notebook can be viewed here: [IBM-Capstone/edadataviz.ipynb at Data-Analytics · newmetagetright/IBM-Capstone]

EDA with SQL

- We queried the contents of the table and variables of interest to understand the relationships between them and landing outcomes through the use of filter statements.
- The completed notebook can be viewed here: [IBM-Capstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb at Data-Analytics · newmetagetright/IBM-Capstone]

```
%sql SELECT "Mission_Outcome", COUNT(*) AS "Total" FROM SPACEXTABLE 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

```
%sql PRAGMA table_info('SPACEXTABLE')
```

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

cid	name	type	notnull	dflt_value	pk
0	Date	TEXT	0	None	0
1	Time (UTC)	TEXT	0	None	0
2	Booster_Version	TEXT	0	None	0
3	Launch_Site	TEXT	0	None	0
4	Payload	TEXT	0	None	0
5	PAYLOAD_MASS__KG_	INT	0	None	0
6	Orbit	TEXT	0	None	0
7	Customer	TEXT	0	None	0
8	Mission_Outcome	TEXT	0	None	0
9	Landing_Outcome	TEXT	0	None	0

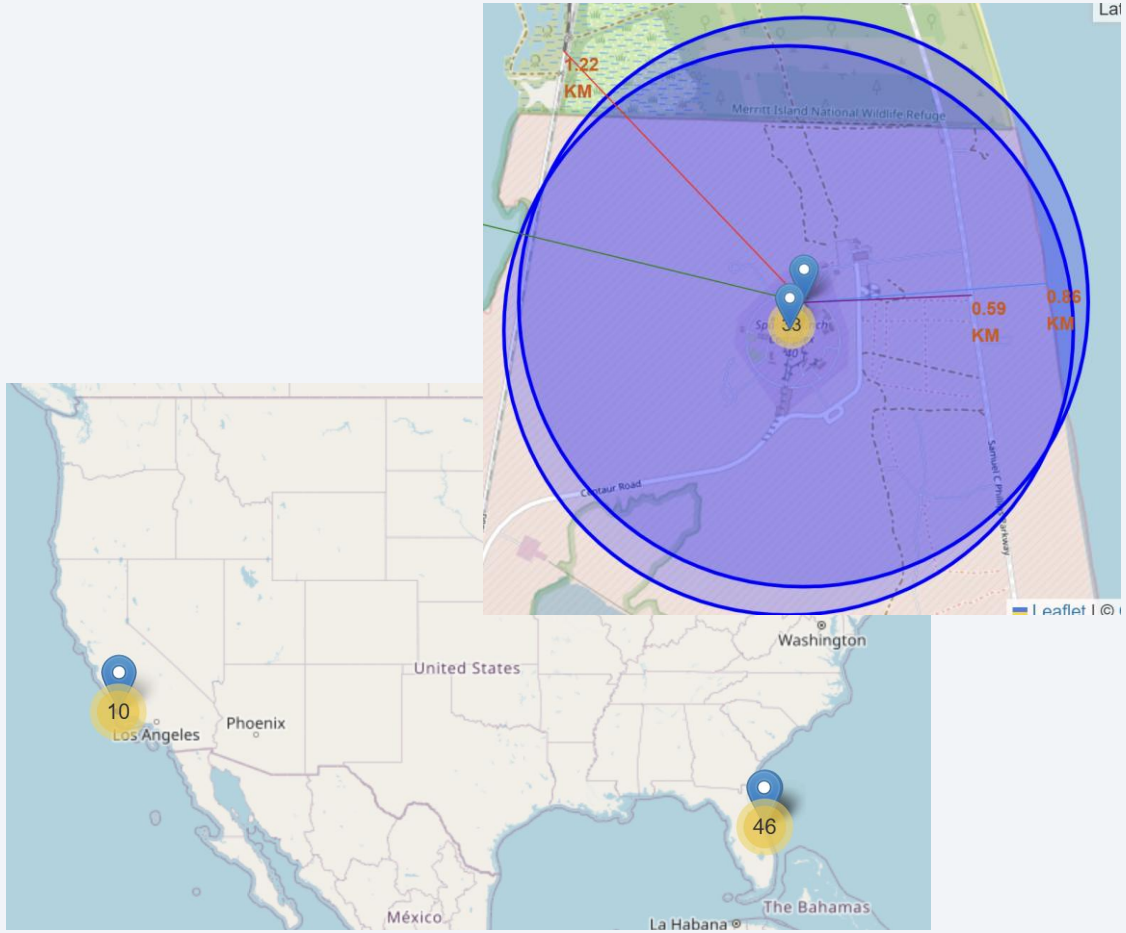
```
%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
```

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

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

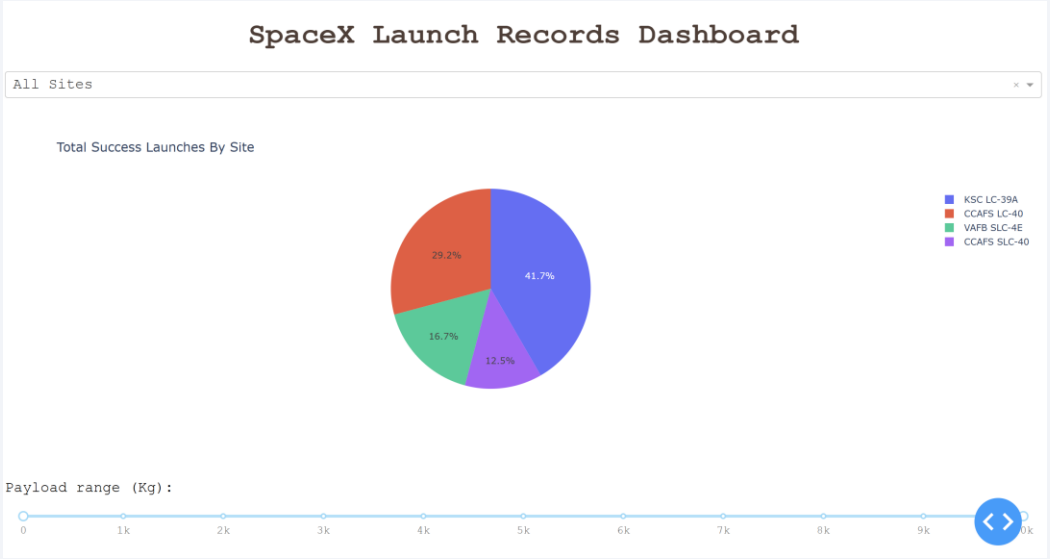
Build an Interactive Map with Folium

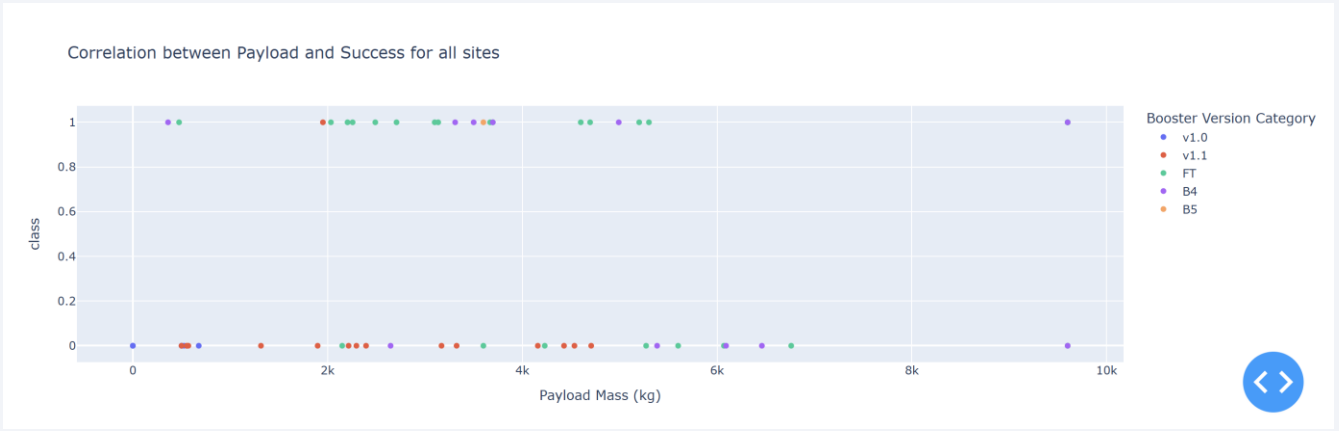
- We created map objects such as markers, clusters and distance lines to get a better understanding geographically of each launch site, including the distances to landmark locations. The clusters show us the success rates for each launch site.
- The completed notebook can be viewed here: [IBM-Capstone/lab_jupyter_launch_site_location.ipynb at Data-Analytics · newmetagetright/IBM-Capstone]



Build a Dashboard with Plotly Dash

- Using Plotly Dash, we created interactive dropdown, pie chart, range slider and scatterplot dashboard components. The visualizations make it easier to filter and comprehend the data pertaining to the relationships between launch site, payload mass, booster versions, and successful launches/landings.
- The completed python script for the dashboard can be viewed here: [IBM-Capstone/spacex_dash_app.py at Data-Analytics · newmetagetright/IBM-Capstone]





Predictive Analysis (Classification)

- We standardized the dataset to improve model performance and split it up into training and testing sets. We then specified a list of parameters so that GridSearchCV could find the combination that would result in the highest accuracy. We fit the training data into each model and compared their performances using confusion matrixes and scoring methods.
- The completed notebook can be viewed here: [IBM-Data-Science-Capstone/SpaceX_Machine Learning Prediction_Part_5.ipynb at Data-Analytics · newmetagetright/IBM-Data-Science-Capstone]

```
transform = preprocessing.StandardScaler()
X = transform.fit_transform(X)
X
```

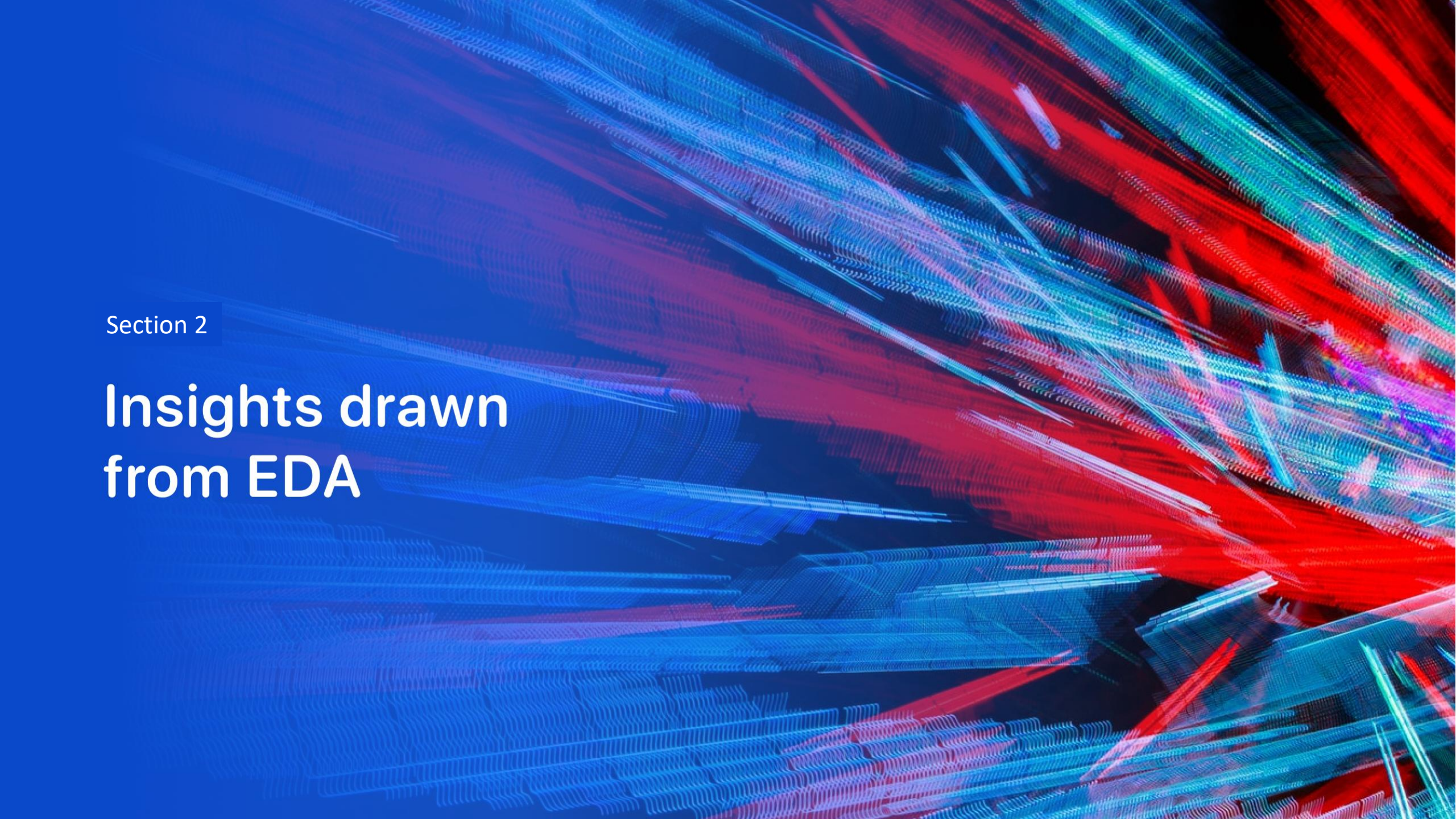
```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,test_size=0.2,random_state=2)
```

```
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge
lr=LogisticRegression()
logreg_cv= GridSearchCV(estimator=lr,param_grid=parameters,cv=10)
logreg_cv.fit(X_train,Y_train)
```

Results

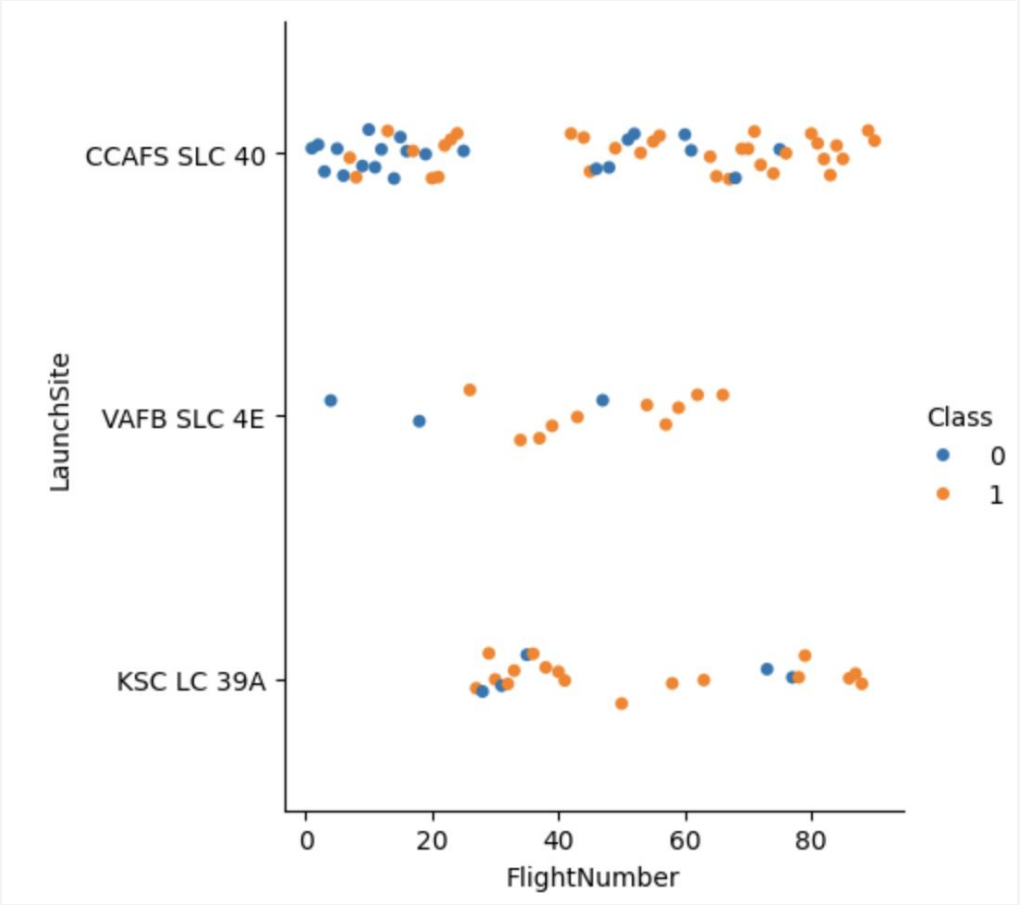
- Through our exploratory analysis, we found that there were several variables that contributed to the success or failure of a SpaceX landing. This primarily being Payload Mass, Booster Version, Flight Number, and Launch Site.
- By training models on the preprocessed data, we were able to create a model that achieved a performance of 87.7% accuracy when it came to predicting the outcome of a launch. This model was the decision tree classifier with the following hyperparameters:

```
tuned hyperparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 6, 'max_features': 'sqrt', 'min
_samples_leaf': 1, 'min_samples_split': 5, 'splitter': 'random'}
accuracy : 0.8767857142857143
```


Section 2
Insights drawn from EDA

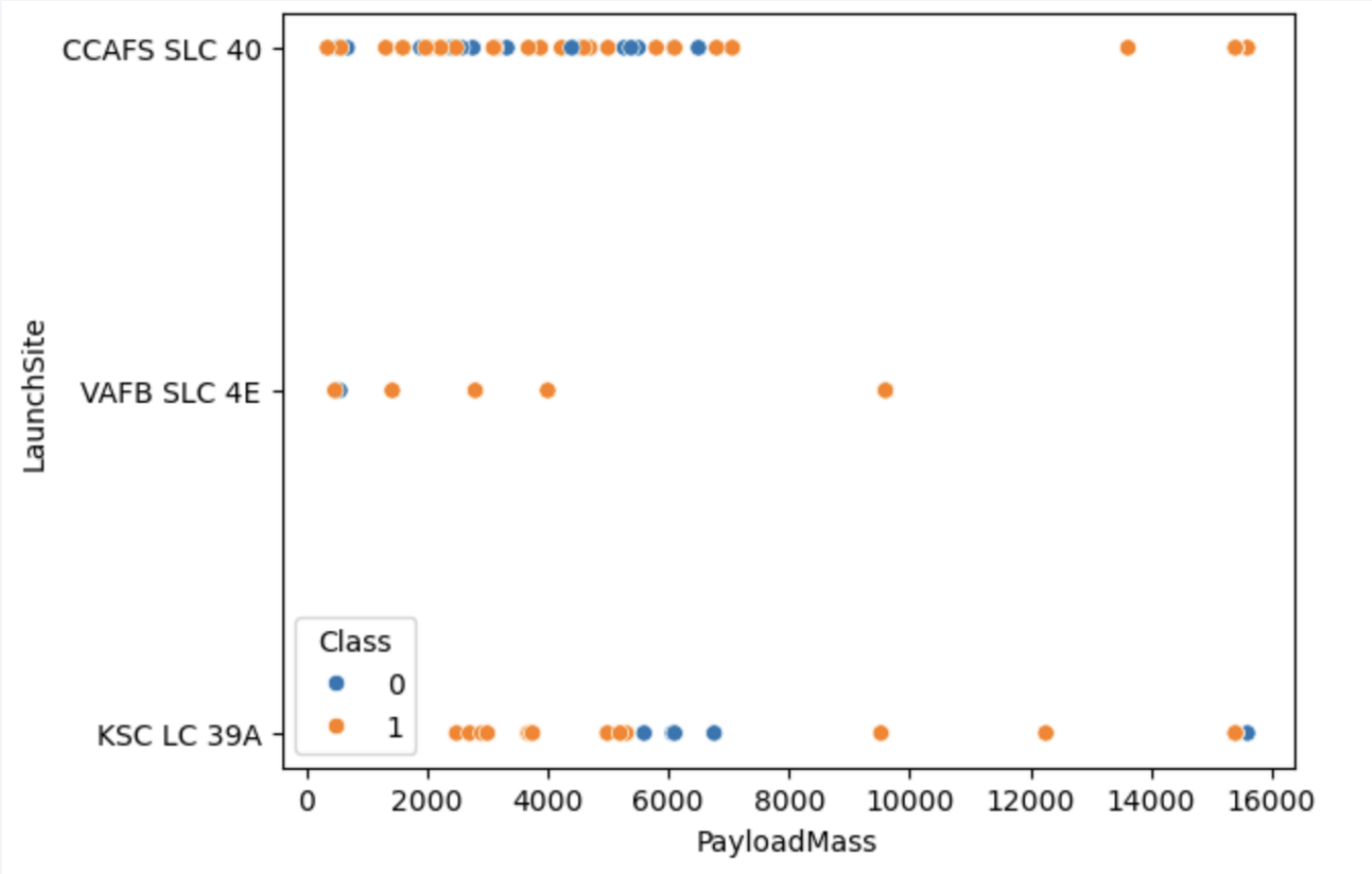
Flight Number vs. Launch Site

- In general, as the number of flights increased, the landing success rate increased as well for each launch site.



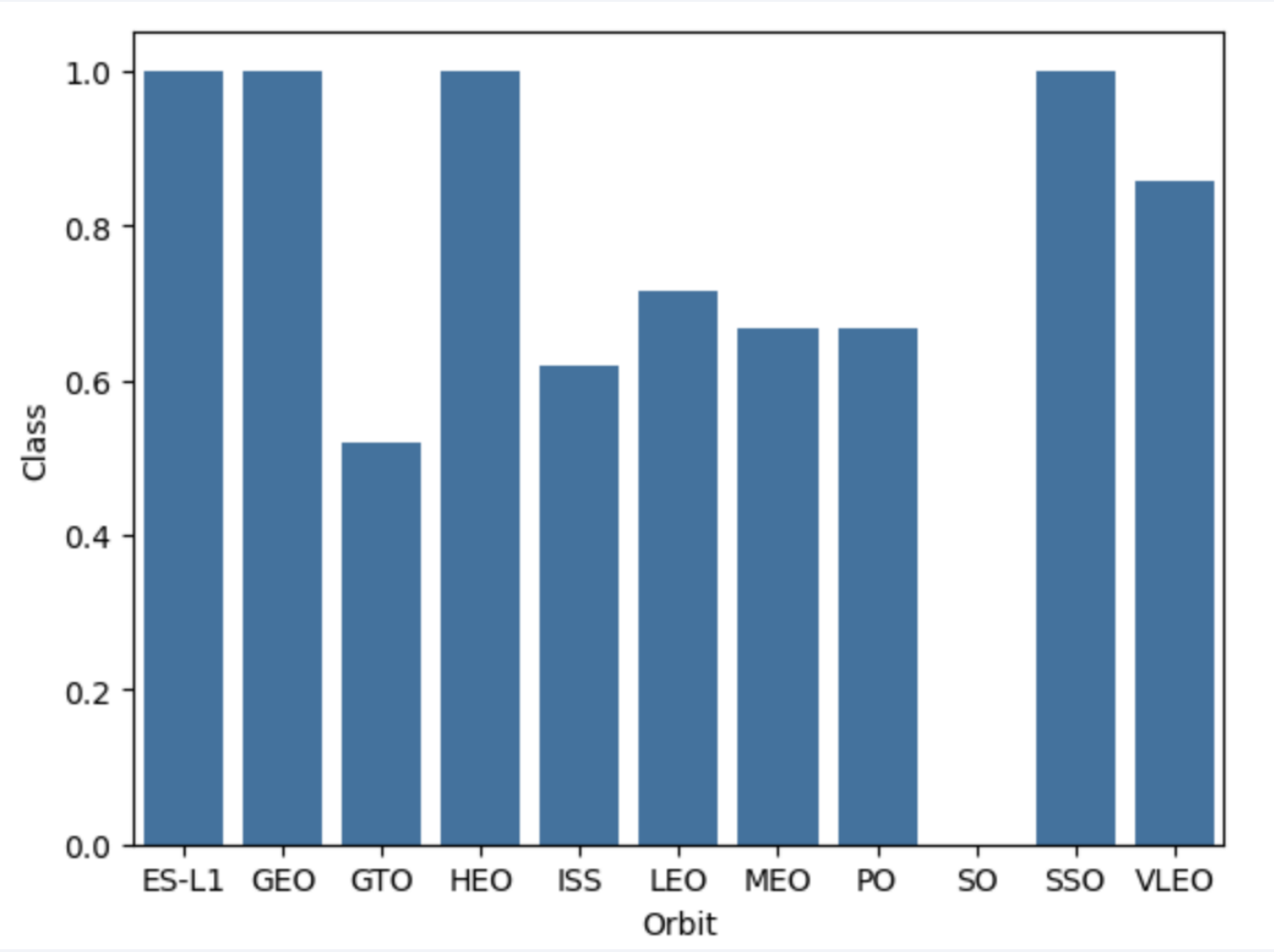
Payload vs. Launch Site

- We see that most of the launches carried a payload between 1000kg and 6000kg and that those above 8000kg had higher success rates.



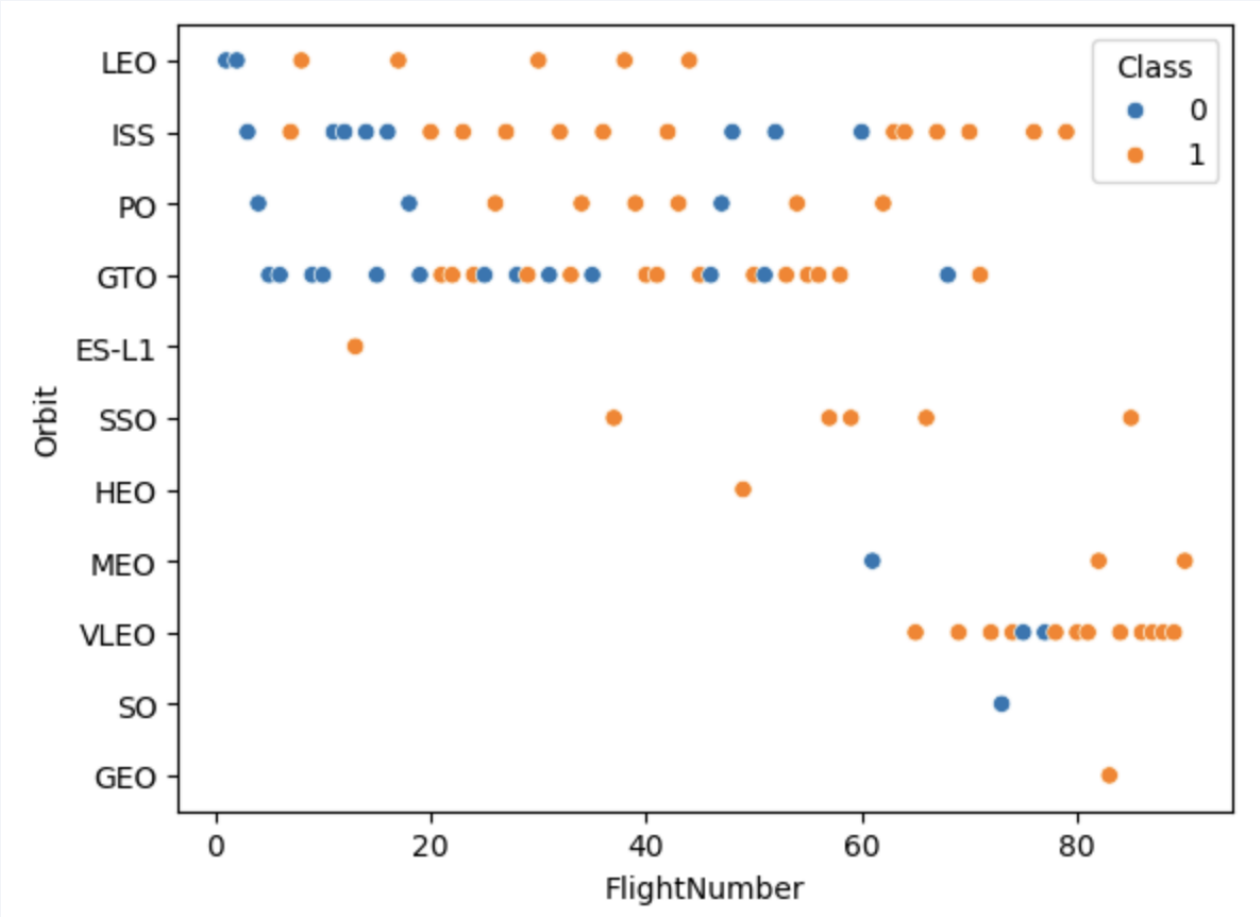
Success Rate vs. Orbit Type

- As shown, four orbit types command a success rate of 100%, while the average success rate falls closer to 60%. The SO orbit type being the outlier at a 0% success rate.



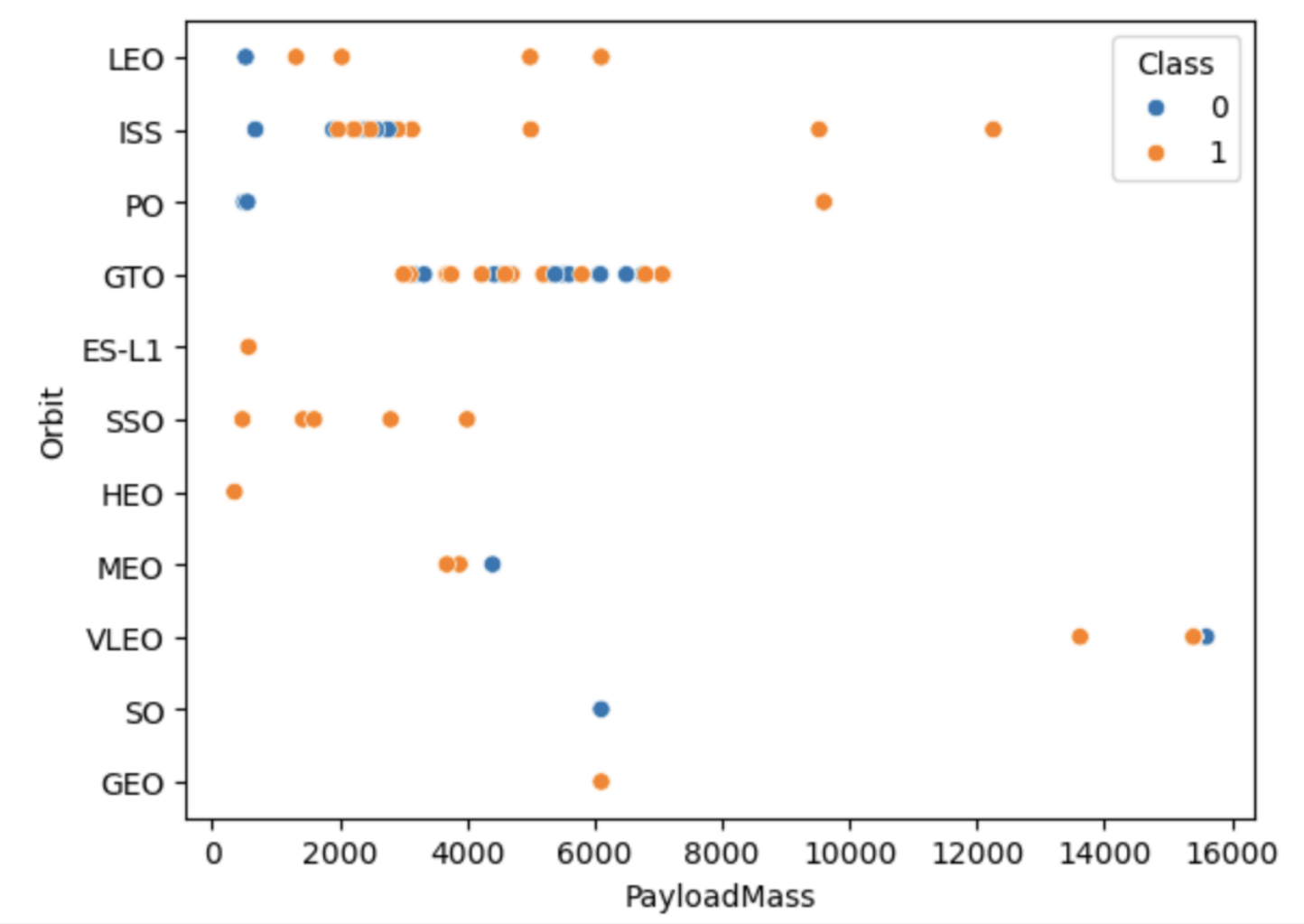
Flight Number vs. Orbit Type

- Interestingly, the bottom half of orbit launches (SSO and below) did not take place until around the 40th flight. We can see that generally the success rate increased over the number of flights per orbit type.



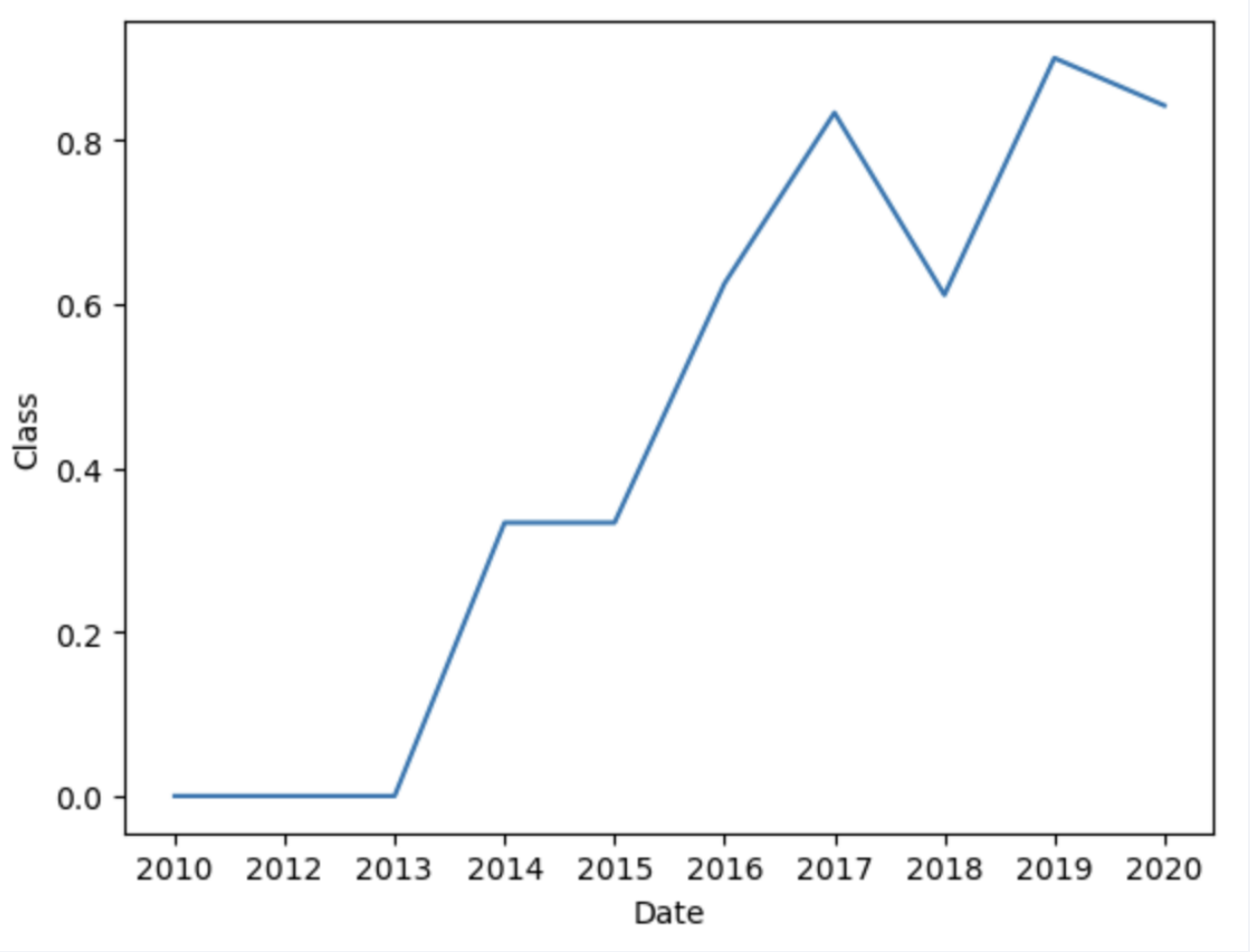
Payload vs. Orbit Type

- We see that the bulk of orbit launches consist of a payload mass of 6000kg and under except for VLEO orbit type, whose payloads are each over 13000kg.



Launch Success Yearly Trend

- Notably, there has been a long-term positive trend in successful launch outcomes since 2013, heeding the stagnation in 2014-2015 and the decline in years 2017-2018 and 2019-2020. Overall, over an 80% increase in success rates from 2010 to 2020.



All Launch Site Names

- Queried the launch site column within the database table to include the names of unique launch sites.

```
%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

Launch Site Names Begin with 'CCA'

- Selected 5 records in which the launch site name began 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
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success
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
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success

Total Payload Mass

- Developed a query to calculate the total payload mass originating from launches conducted by NASA (CRS)

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass" FROM SPACEXTABLE WHERE "Customer" LIKE "%NASA (CRS)%"
```

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

Total Payload Mass
48213

Average Payload Mass by F9 v1.1

- Calculated the average payload mass for launches equipped with booster version 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

First Successful Ground Landing Date

- Found the date of the first successful ground landing

```
%sql SELECT MIN("Date") FROM SPACEXTABLE 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

- Queried the database to select booster versions that had successful drone ship landings and carried a payload mass between 4000kg and 6000kg

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

Total Number of Successful and Failure Mission Outcomes

- Grouped the total number of flights by mission outcomes

```
%sql SELECT "Mission_Outcome", COUNT(*) AS "Total" FROM SPACEXTABLE 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

- Filtered and selected for booster versions that have carried the maximum payload amount

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

- Selected the month, booster versions, and launch sites that had failure drone ship landing outcomes for the year 2015.

Month	Landing_Outcome	Booster_Version	Launch_Site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Selected the count of landing outcomes between the dates 2010-06-04 and 2017-03-20 in descending order

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

Section 3

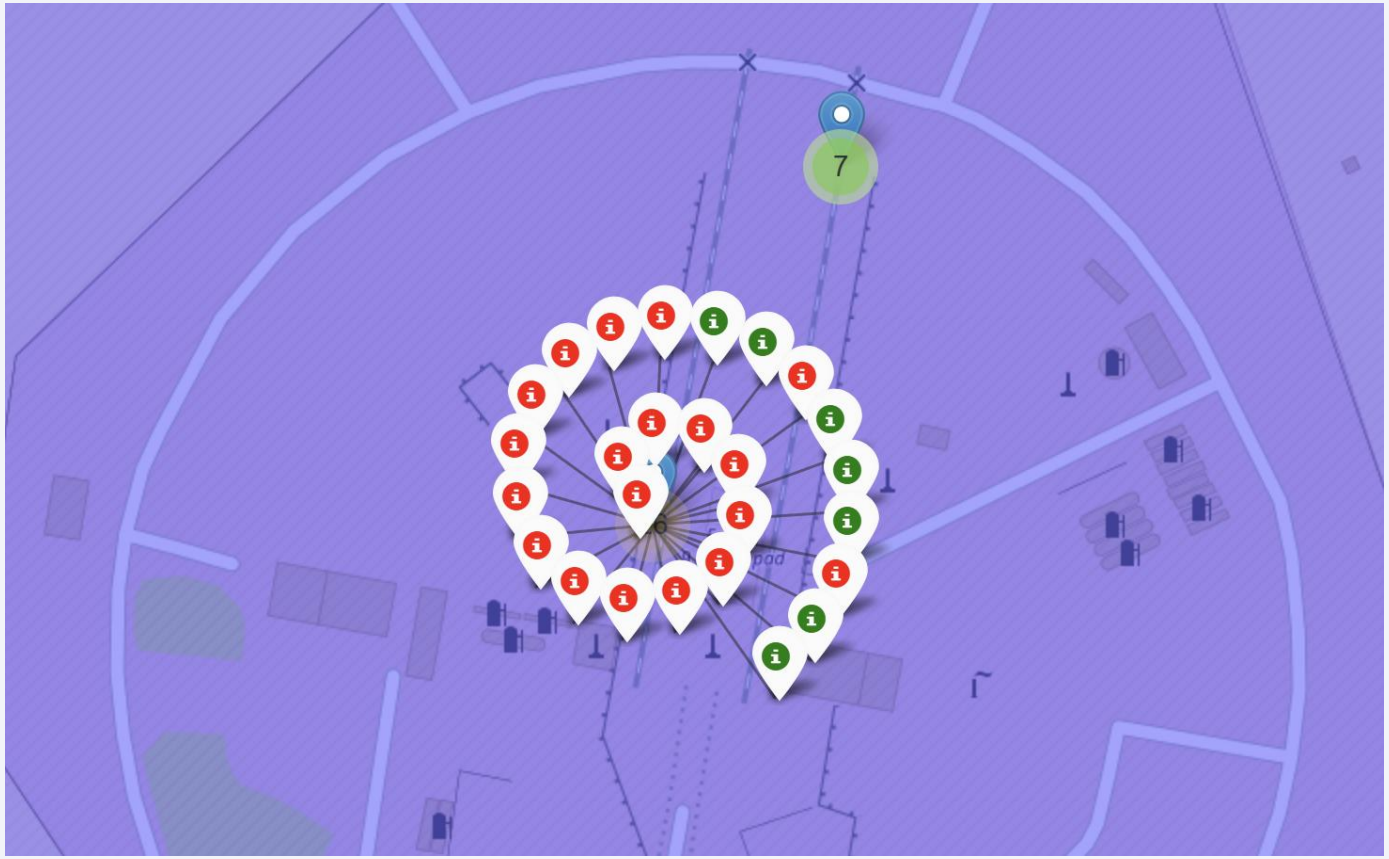
Launch Sites Proximities Analysis

Generated Map with Markers of Each Launch Site

- The map shows the geographic location markers and marker popups of each Launch Site within the dataset.

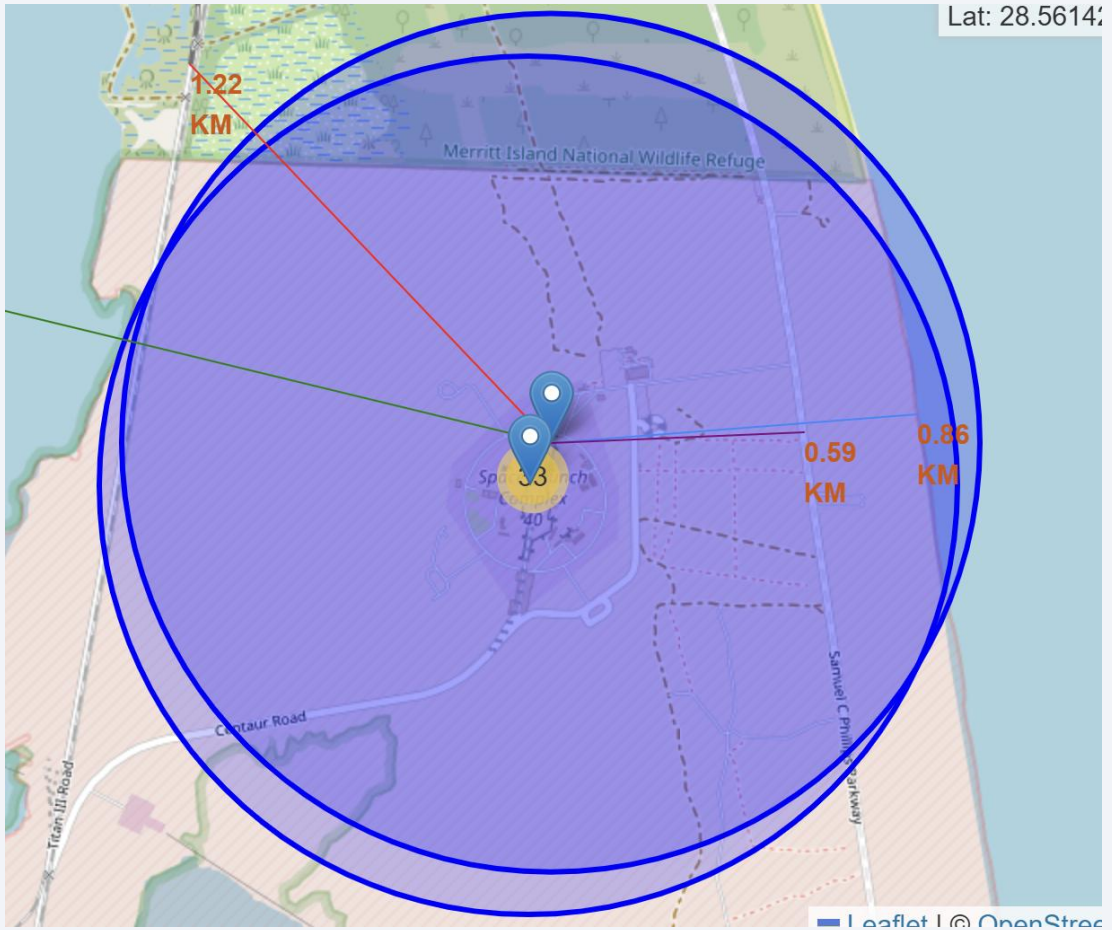
CCAFS LC-40 Launch Site Cluster Markers

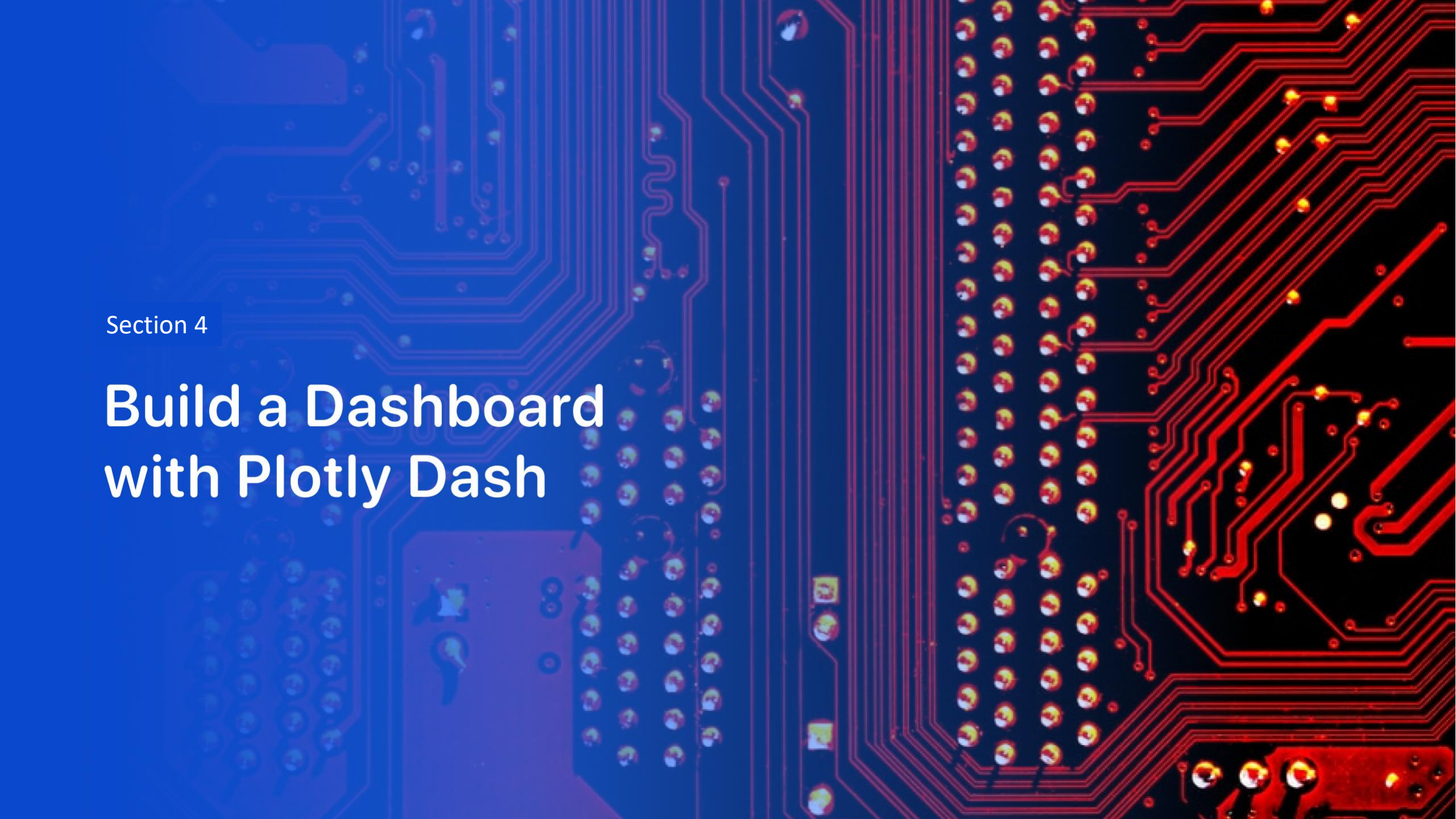
- A map that consists of cluster markers showing the landing outcomes of launches based on color on the CCAFS LC-40 launch site. Red indicates failure and green indicates success.



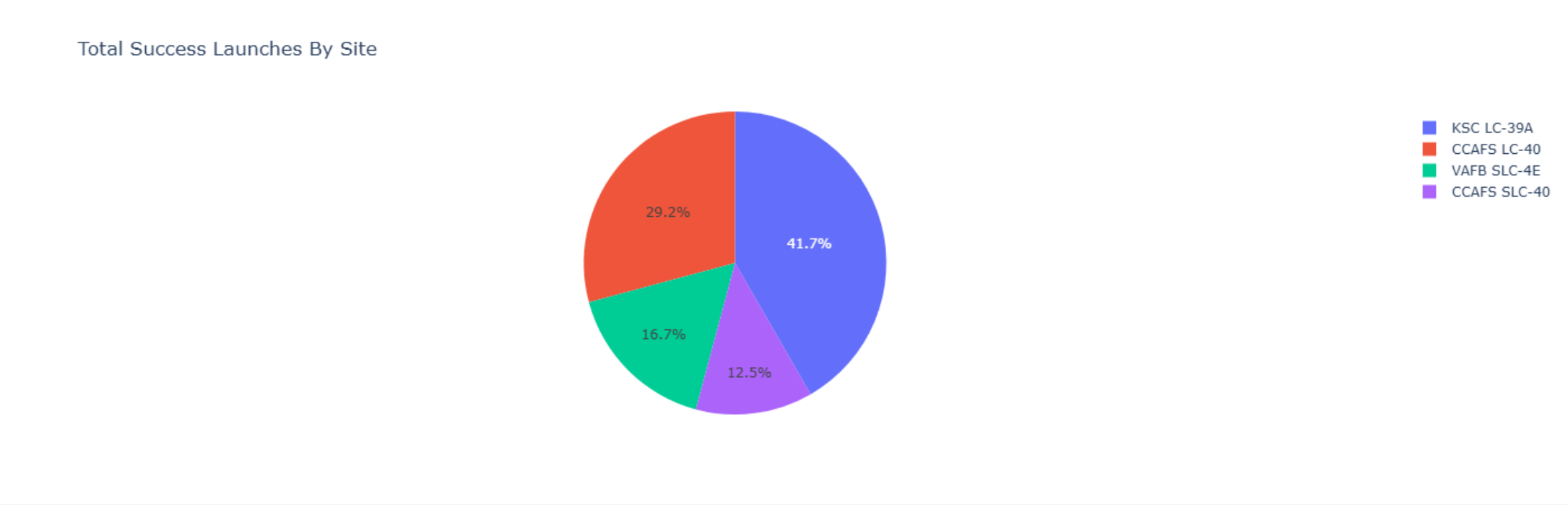
Landmark Distance Lines from CCAFS SLC-40 Launch Site

- This map shows the distance lines with measured labels from the CCAFS SLC-40 launch site to the nearest landmarks, which includes the nearest city, railroad, highway and coastline.



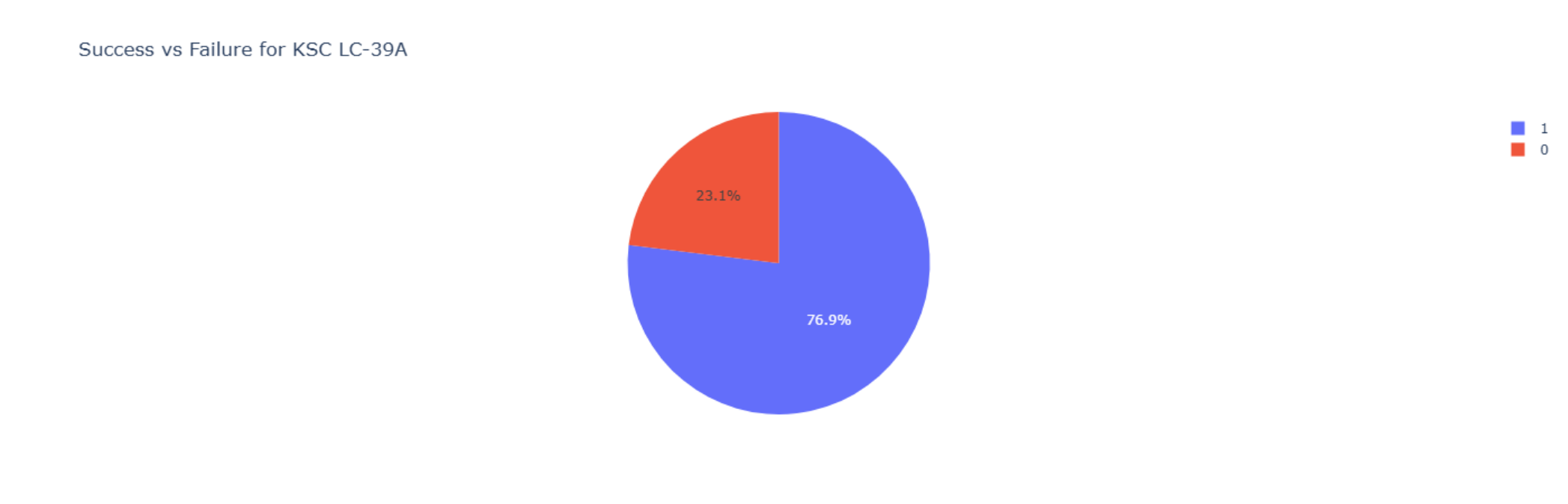

Section 4
Build a Dashboard with Plotly Dash

Total Successful Launches By Site



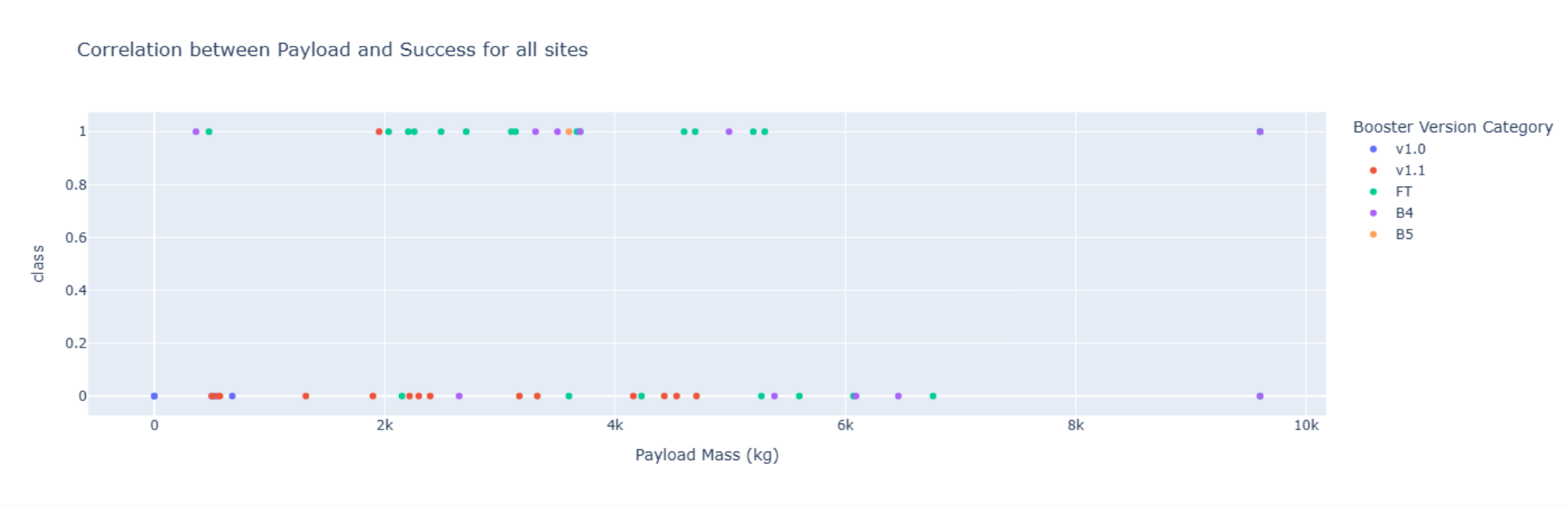
- The pie chart shows the overall distribution of successful launches per launch site, with KSC LC-39A having the most successful launches, and CCAFS SLC-40 having the least.

Landing Outcomes for KSC LC-39A Launch Site



- This pie chart shows the portions of launch outcomes existing for the KSC LC-39A launch site. As demonstrated by the chart, 76.9% launch outcomes were successful while 23.1% were failures.

Scatter Chart of Payload Mass and Landing Outcomes By Booster Version



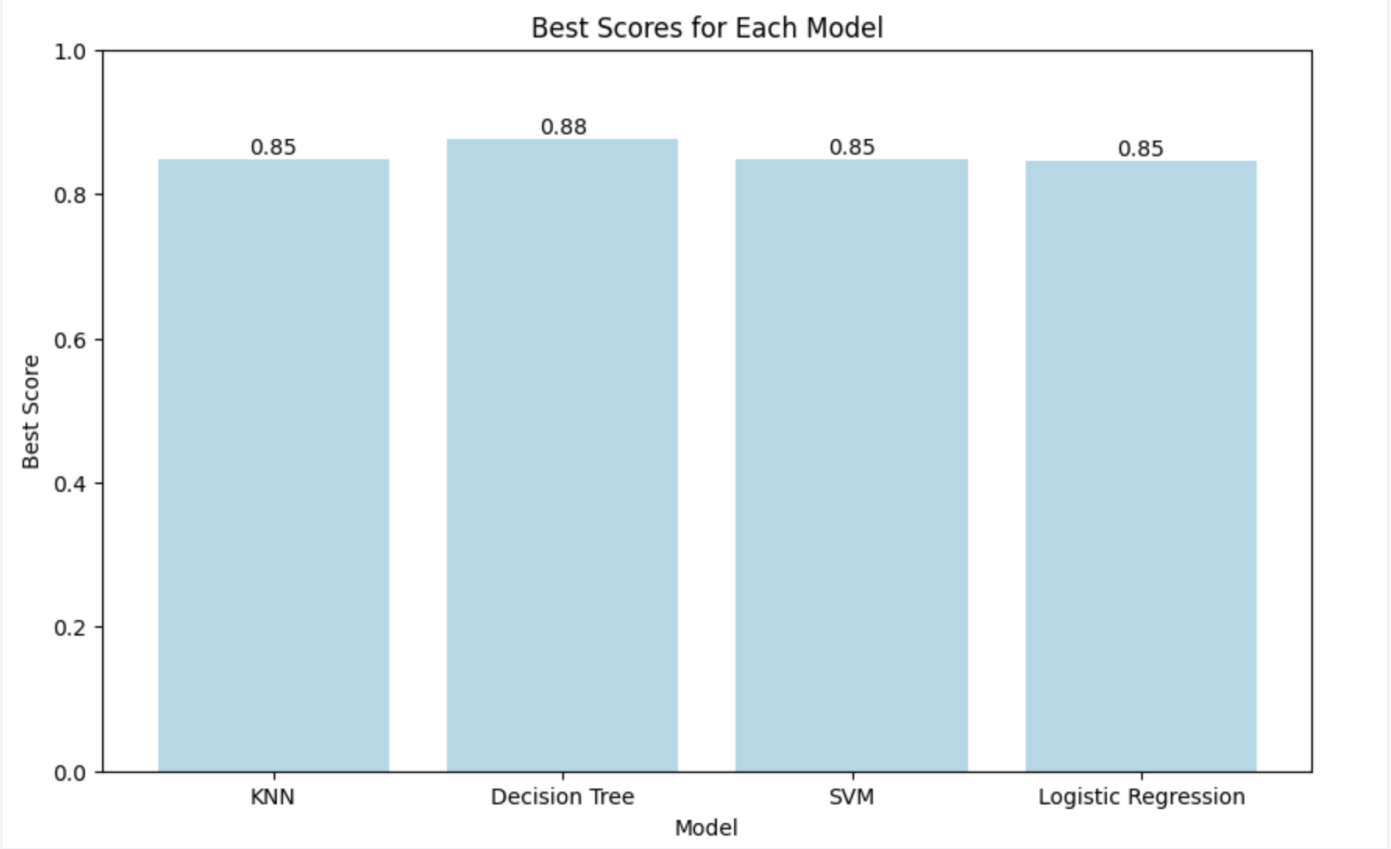
- The scatterplot shows the landing outcomes by payload mass for each booster version. The booster version FT had the highest success rates among all booster versions.

Section 5

Predictive Analysis (Classification)

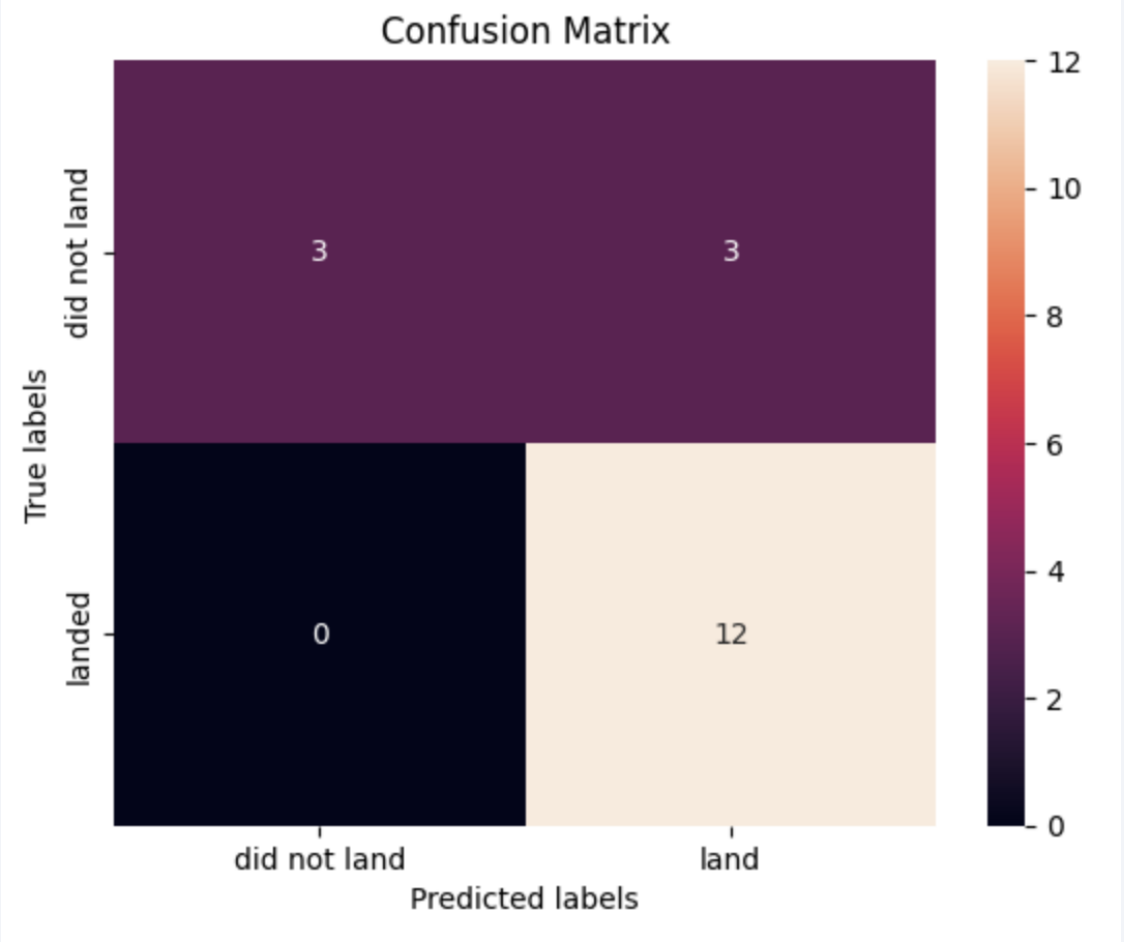
Classification Accuracy

- As depicted in the bar chart, the Decision Tree Classifier model had the highest accuracy among the four that were trained and tested, with a best score of 88% accuracy.



Confusion Matrix

- The confusion matrix displays the count and categories of the true and predicted labels of landing outcomes forecasted by the Decision Tree Classifier. The model predicted with 100% accuracy when the landing outcome was successful, and with 50% accuracy when the landing outcome was a failure.



Conclusions

- From the SpaceX launch dataset, we were able to identify key features correlated with landing outcomes and use these variables to closely examine and visualize the relationships that were present among them.
- Furthermore, we developed four machine learning models using tuned hyperparameters to train and test on the data set. These four were: K-Nearest Neighbors, Logistic Regression, Support Vector Machine, and Decision Tree Classifier models.
- Of the four models, the Decision Tree Classifier was able to predict launch outcomes with the highest accuracy at 87.6%.

Thank you!