IBM Data Science Capstone Project: Space X

Ross-Dean Jonker

2022-06-16

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

EXECUTIVE SUMMARY

- Methodologies:
  - Data Collection through API and Web scraping
  - Exploratory Data Analysis (EDA) - Visualization
  - Exploratory Data Analysis (EDA) - SQL
  - Interactive Visual Analytics
  - Interactive Dashboard (Dash App)
  - Predictive Analysis – Classification (Model)
- Summary:
  - Graphs and Results from EDA
  - Interactive Visuals and results
  - Predictive Analysis Results

INTRODUCTION

- **Context:**

In this capstone, we will predict if the Falcon 9 first stage will land successfully. 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. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- **Problems to be addressed:**

- What factors determine if rocket will land successfully?

- What are the relationships between the factors and outcome?

- What are the optimum conditions to ensure successful landing?

METHODOLOGY

Data Collection

- Space X Rest API

Space X Launch data was collected from the Space X Rest API.

- Web Scraping

Web Scraping was used to gather Falcon 9 launch data from the Space X Wikipedia webpage.

https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches

Exploratory Data Analysis (EDA)

- The relationship was explored between:
  - Flight Number and Launch Site
  - Payload and Launch Site
  - Success rate of each orbit type
  - Flight Number and Orbit type
  - Payload and Orbit type
  - Success Rate and Year

The relationship between Success Rate and year provided the most valuable insight, as it shows that Space X improved their success on landing the rockets from 2013 onwards.

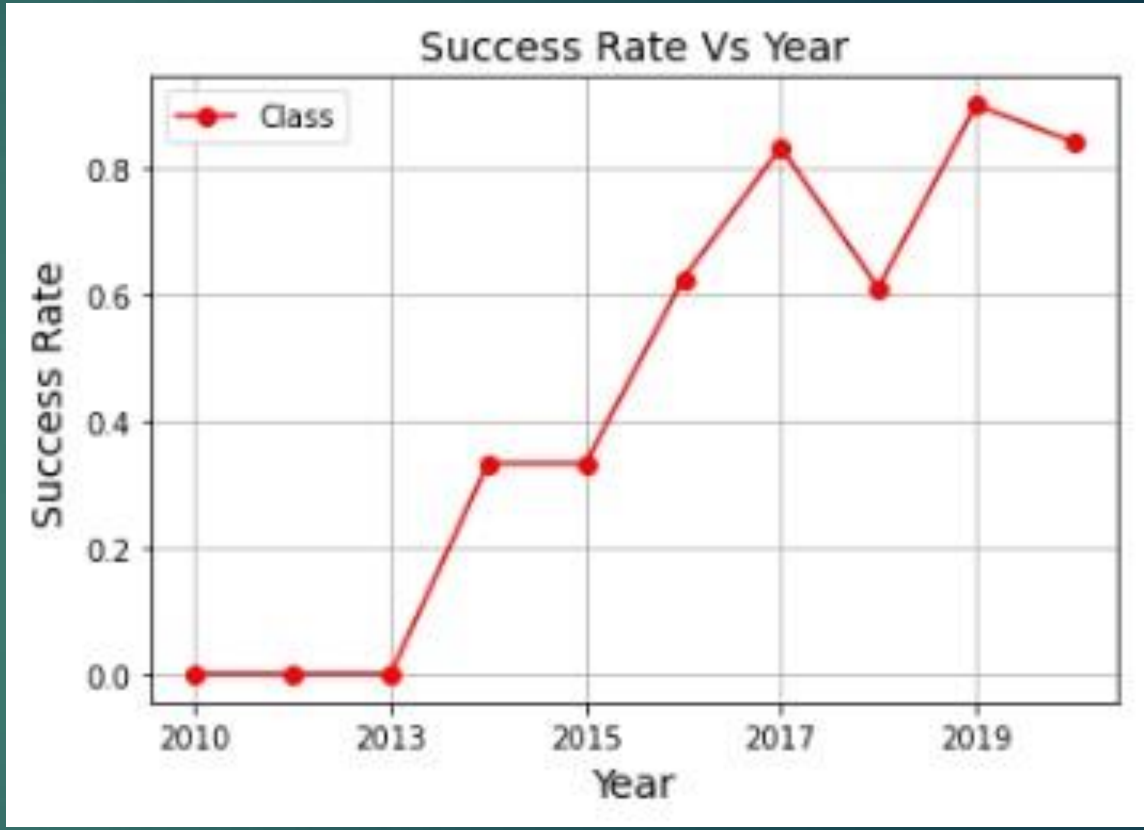


Figure 1: Showing the relationship between Success rate and Year

Exploratory Data Analysis (EDA)

- Using SQL queries we determined:
  - *The names of the unique launch sites in the space mission*
  - *The **total payload** mass carried by boosters launched by NASA was **107 010 kg***
  - *The **average payload mass** carried by booster version F9 was **2 534 kg***
  - *The **first successful landing** outcome in ground pad was **2015-12-22***
  - *The **maximum payload mass** was **15 600 kg***
  - *The **landing outcomes** between **2010-06-04 and 2017-03-20***

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

Figure 3: Showing Unique names of launch sites

landing__outcome	total_number
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

Figure 2: Showing results landing outcomes between 2010-06-04 and 2017-03-20

Interactive Visual Analytics

- Using Folium an interactive map was created:
  - Markers and Cluster were added to represent the launch sites and launches

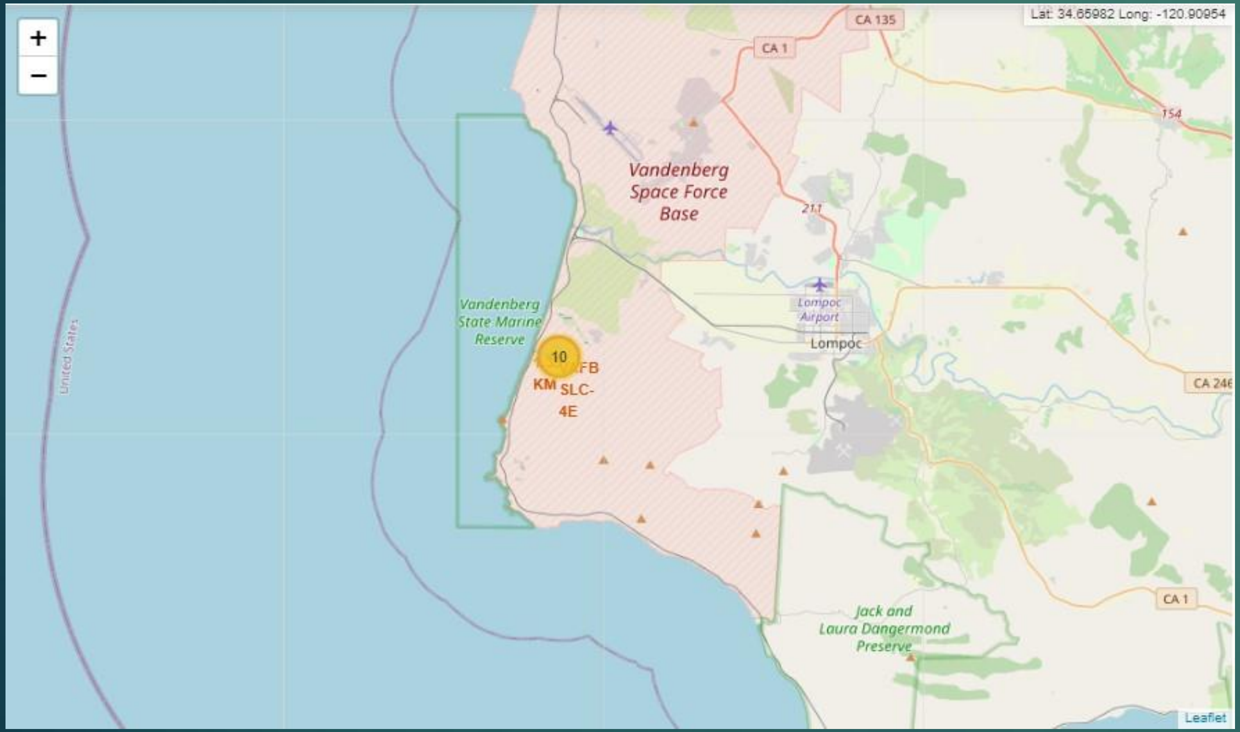


Figure 4: Showing a launch site on interactive map

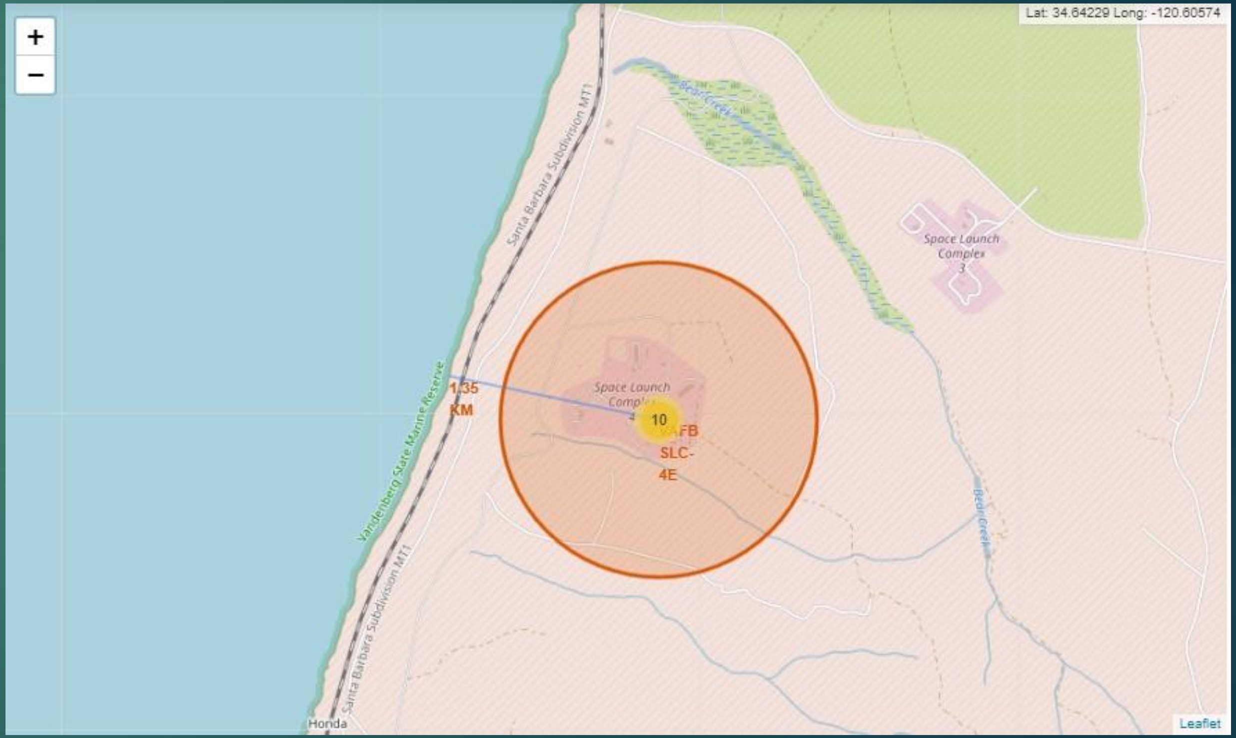


Figure 5: Showing a launch site on interactive map (Zoomed in)

Interactive Dashboard

- Using Plotly Dash, an interactive dashboard was created:
  - It displays pie charts and plots based on user inputs.

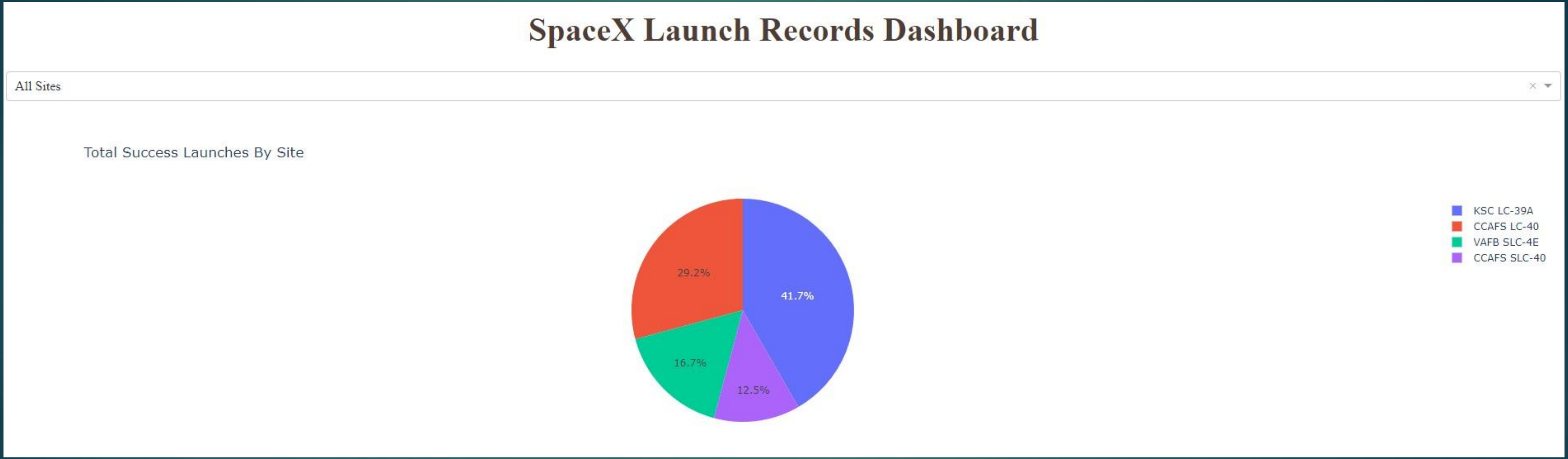


Figure 6: Showing Pie Chart using Dash App.

Predictive Analysis – Classification

- Different predictive models were created:
  - logistic regression
  - support vector machine
  - decision tree classifier
  - k nearest neighbors (KNN)

- Accuracies for models:
  - logistic regression - 84.64%
  - support vector machine - 84.82%
  - **decision tree classifier - 87.5%**
  - k nearest neighbors (KNN) - 84.82%

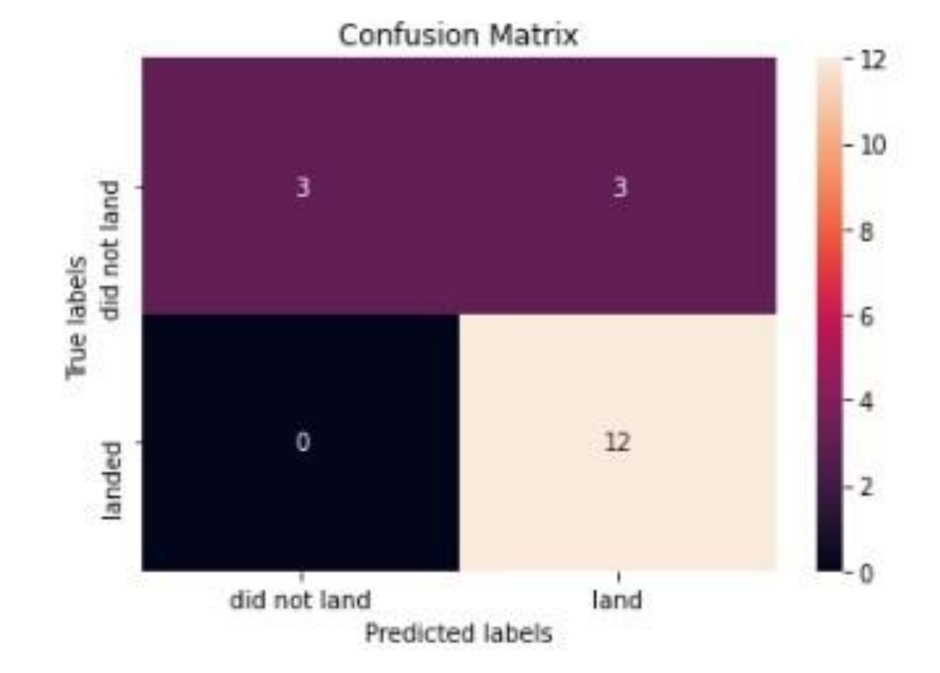
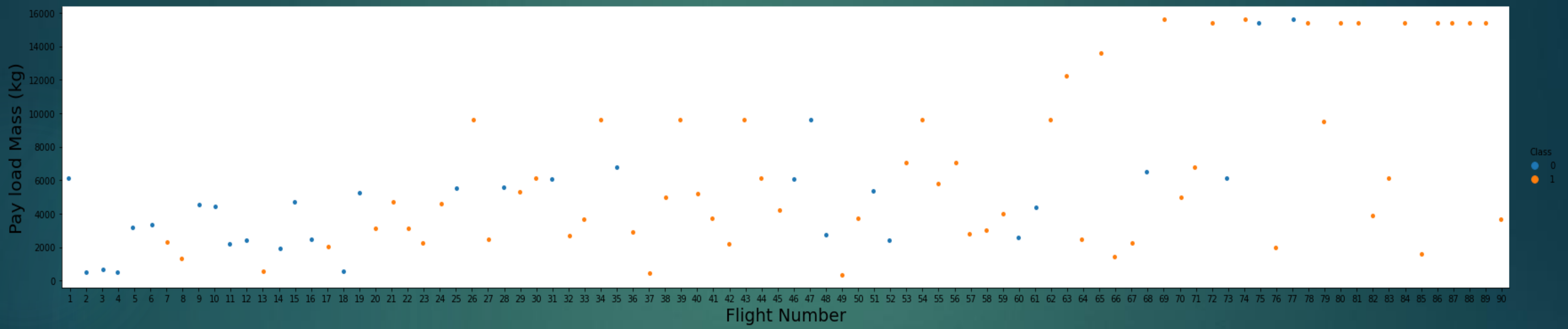


Figure 7: Showing confusion matrix for Decision Tree Model

RESULTS

Exploratory Data Analysis (EDA)

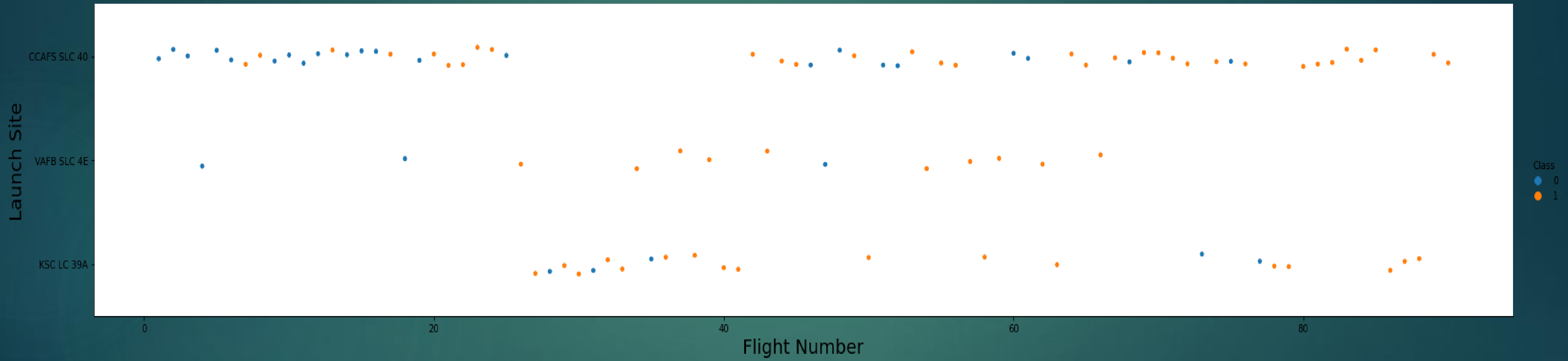
Flight Number vs Payload Mass



From the above figure we can see that the Payload mass has an influence on the success of landing the rocket. As Payload increase the success rate seems to increase.

Exploratory Data Analysis (EDA)

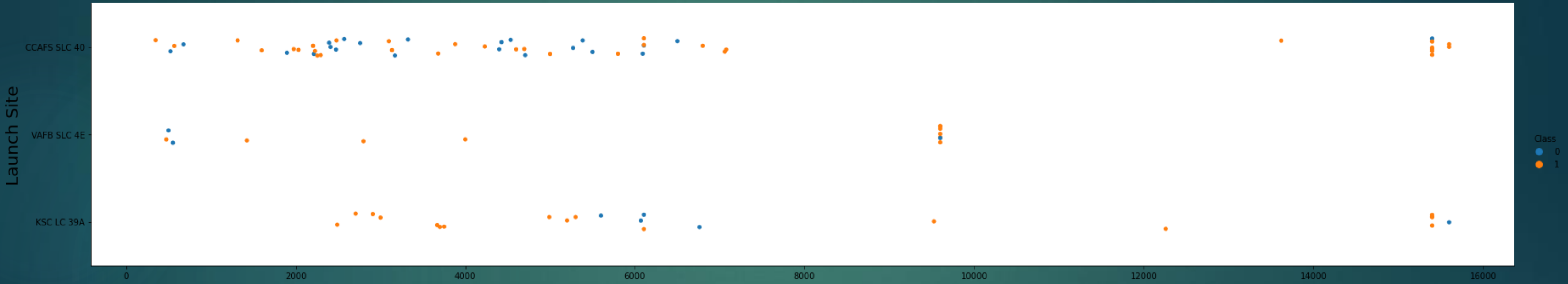
Flight Number vs Launch Site



From the above figure we can see that the Launch Site has an influence on the success of landing the rocket for the later flight number. From this we can see that KSC LC-39A has best success rate.

Exploratory Data Analysis (EDA)

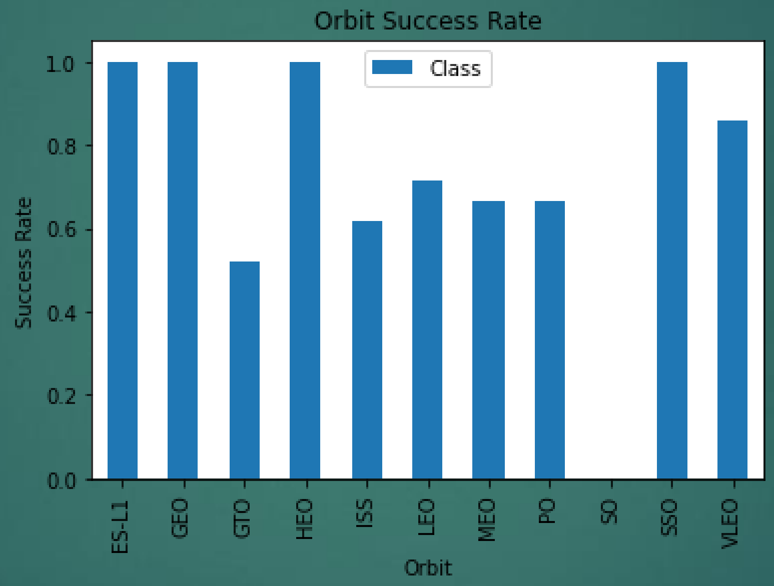
Launch Site vs Payload Mass



From the above figure we can see that the Payload mass and Launch Site have an influence on the success of landing the rocket. For KSC LC-39A the lower the payload mass the higher success rate.

Exploratory Data Analysis (EDA)

Orbit vs Success Rate



From the above figure we can see the success rate for different orbital missions. We can see orbits ES-L1, GEO, HEO and SSO are the most successful.

Exploratory Data Analysis (EDA)

- Using SQL queries we determined:
  - *The names of the unique launch sites in the space mission*
  - *The **total payload** mass carried by boosters launched by NASA was **107 010 kg***
  - *The **average payload mass** carried by booster version F9 was **2 534 kg***
  - *The **first successful landing** outcome in ground pad was **2015-12-22***
  - *The **maximum payload mass** was **15 600 kg***
  - *The **landing outcomes** between **2010-06-04 and 2017-03-20***

landing__outcome	total_number
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

Figure 2: Showing results landing outcomes between 2010-06-04 and 2017-03-20

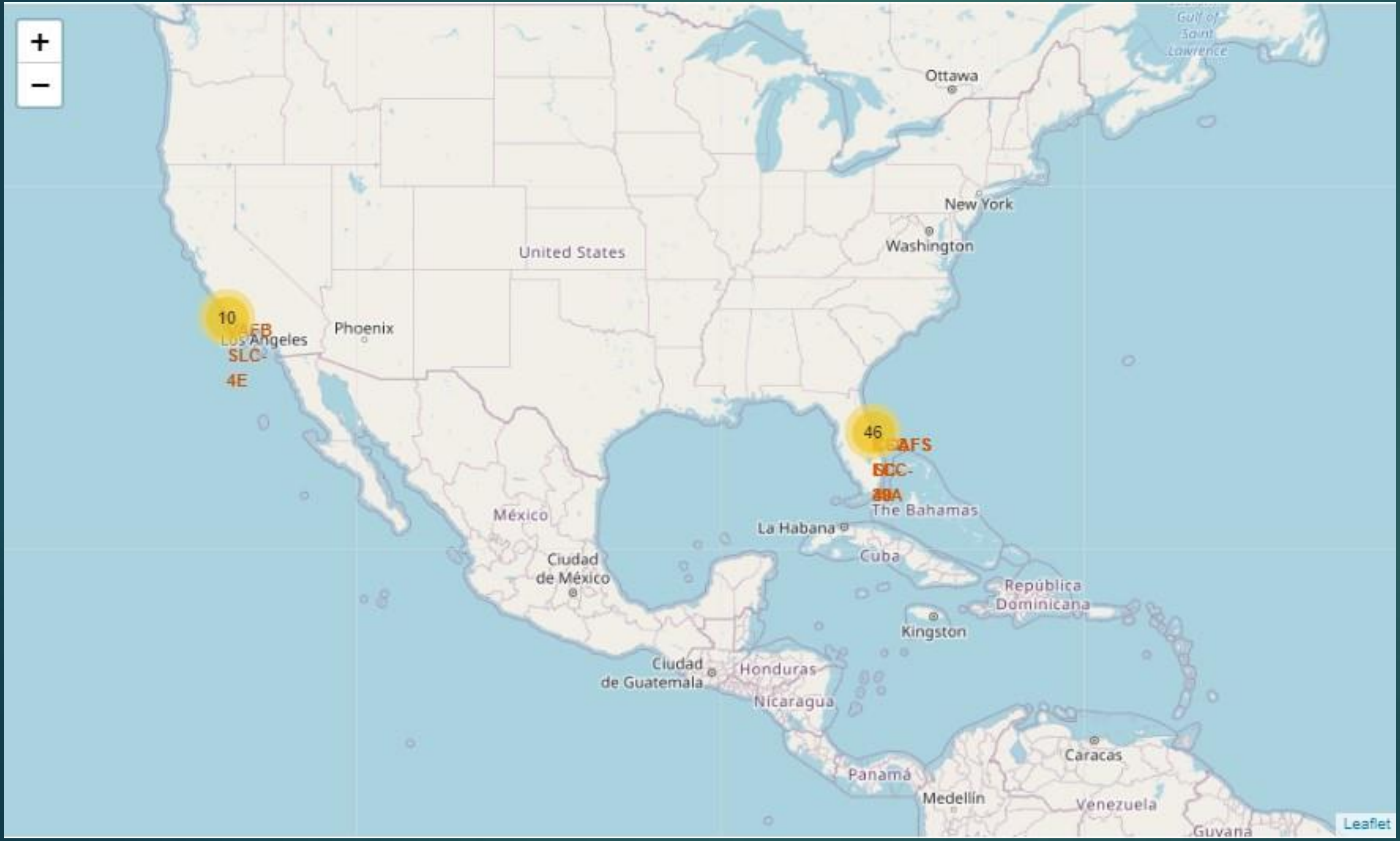
launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Figure 3: Showing Unique names of launch sites

Interactive Visual Analytics

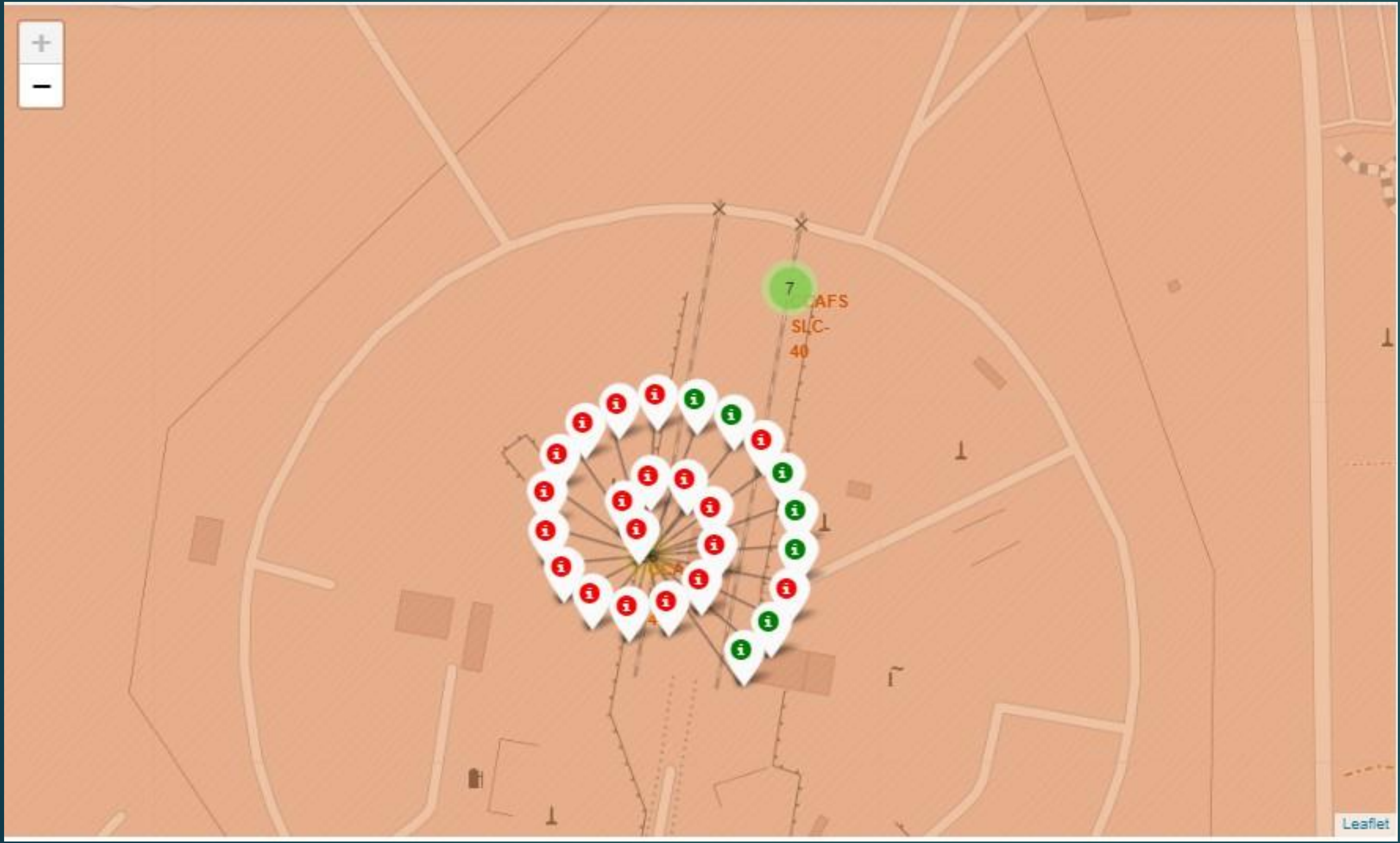
Interactive Map with markers for launch sites

Interactive Visual Analytics



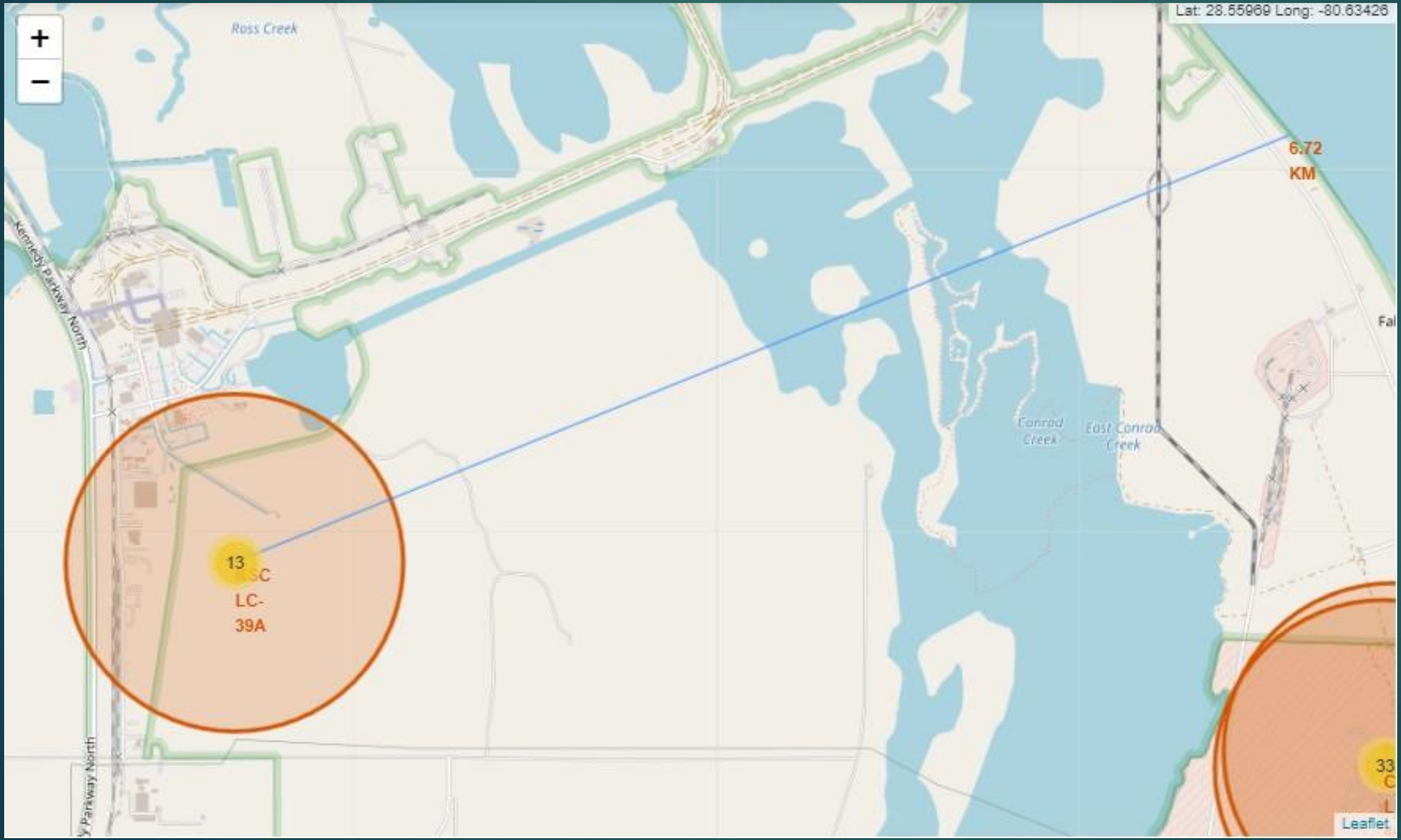
Interactive Map with markers for launches per launch site, clustered.

Interactive Visual Analytics



Interactive Map with icons for launches per launch site.

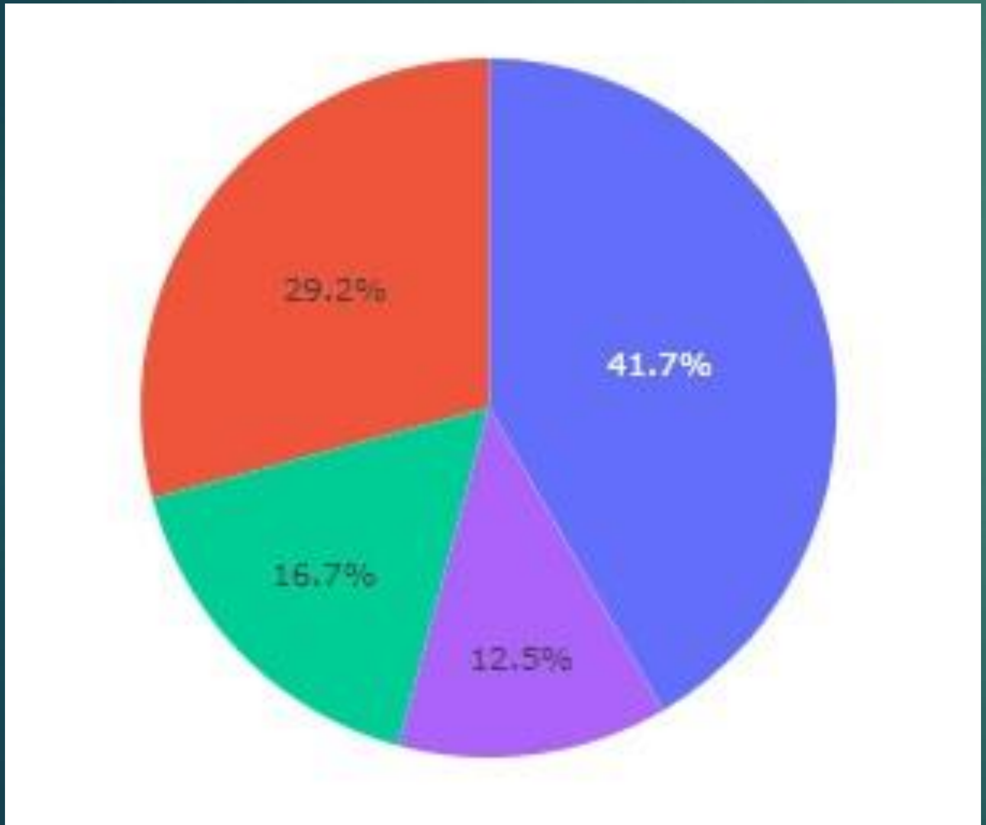
Interactive Visual Analytics

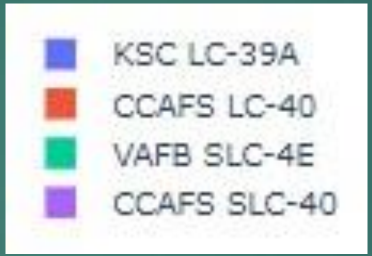


Interactive Map with distance lines from launch site to coastline.

Interactive Dashboard

Total Success Rate for all sites



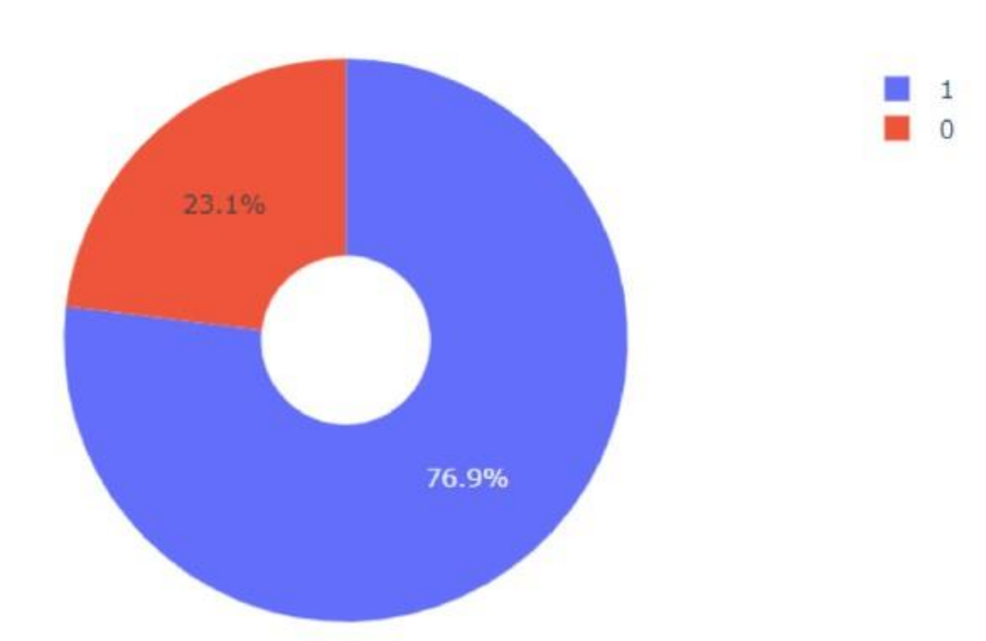


Legend for sites in pie chart

We can see that KSC LC-39A was the most successful launch site.

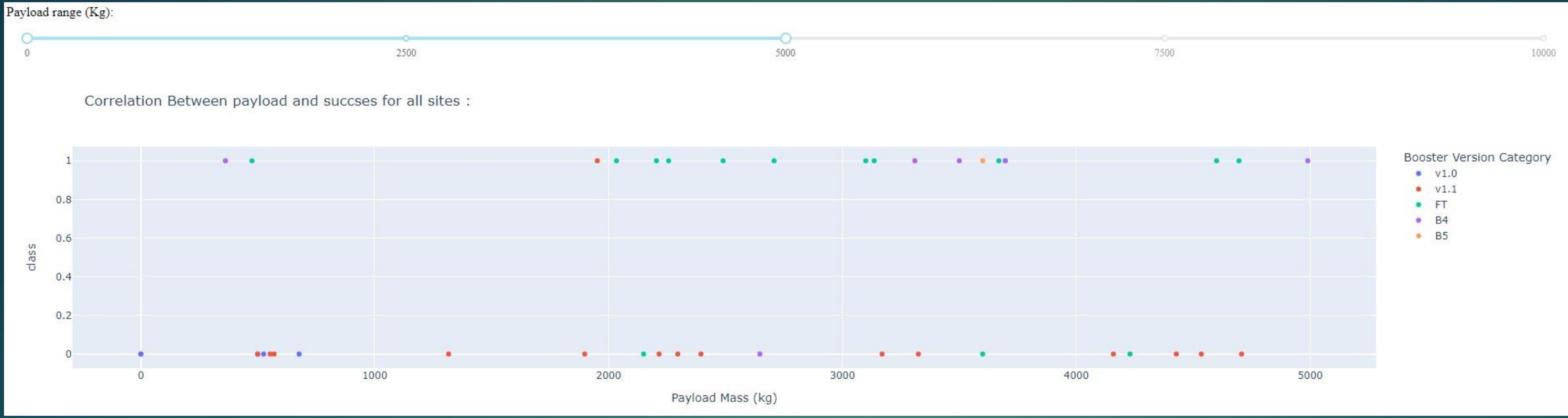
Interactive Dashboard

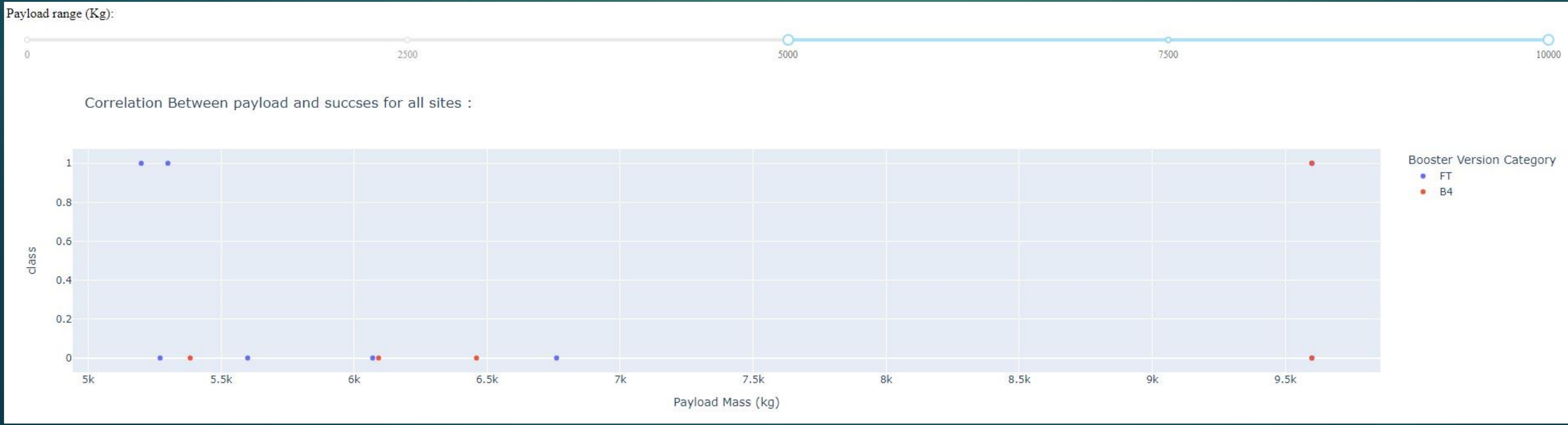
Success Rate for KSC LC-39A



We can see that KSC LC-39A has a 76.9% success rate for land rockets and 23.1% failure rate.

Interactive Dashboard





We can see that lower weighted payload were more successful across the launch sites.

Predictive Analysis – Classification

Classification Model	Training Accuracy	Test Accuracy
Logistic Regression	0.8464	0.8333
Support Vector Machine	0.8482	0.8333
Decision Tree Classifier	0.875	0.7777
K Nearest-Neighbor (KNN)	0.8482	0.8333

The most accurate model on training data was the Decision Tree Classifier. The most accurate models for the test data are Logistic Regression, SVM and KNN.

Conclusions

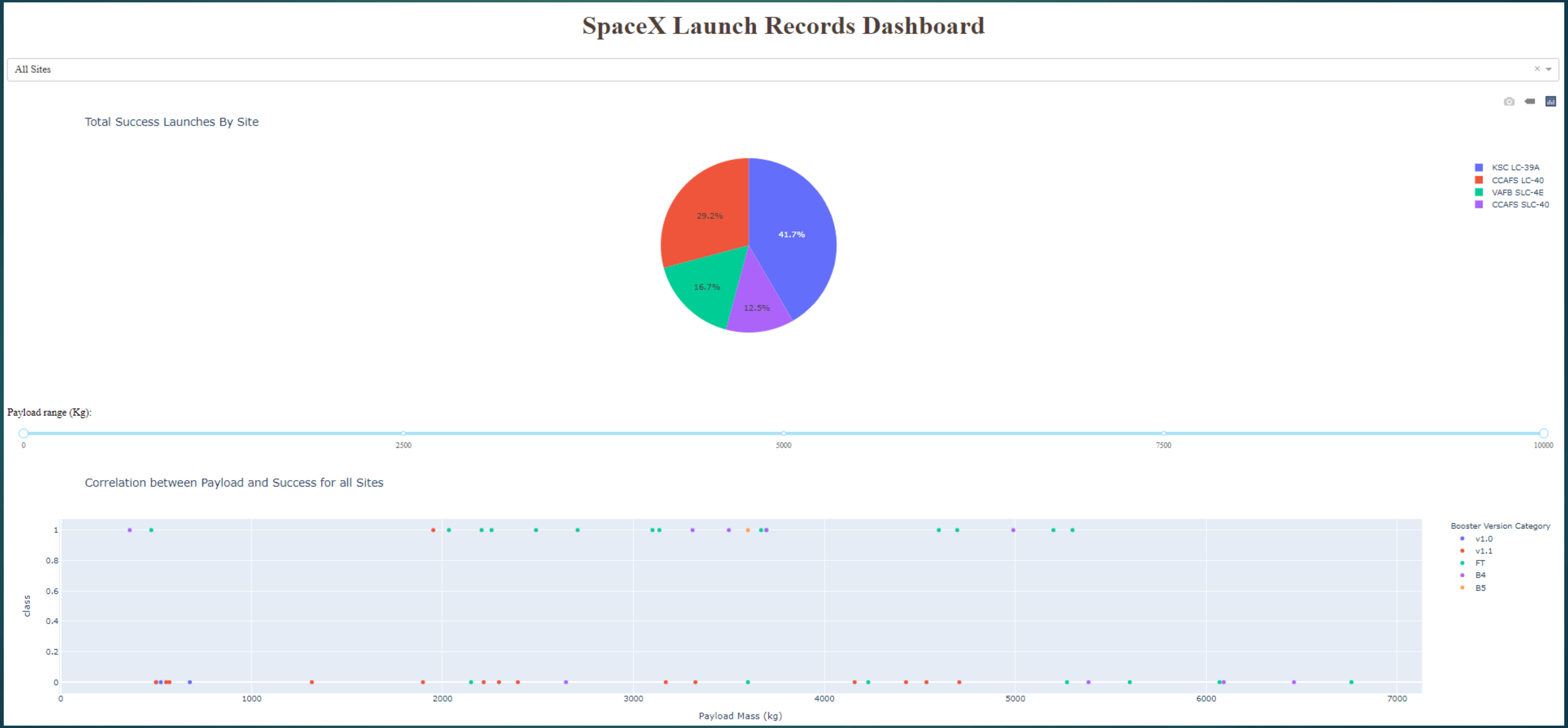
- Flight Number, Payload mass and launch site are indicators for a successful or failed launch prediction.
- Orbits ES-LI, GEO, HEO and SSO have the highest success rates.
- The success rate increased over time from 2013 onwards.
- KSC LC-39A was the most successful launch site.
- Lower weighted payloads are more successful than higher weighted payloads.
- The Decision tree model performed best on training data, but the other model were more accurate on test data.
- The cost therefore will be depended on whether or not the rocket will land successfully or not. If successful the cost will be lower than when not successful. From the aspects analyzed, one can determine which configuration will lead to the highest chance for a successful landing.
- **Thus a payload of lower than 5000 kg, launching from KSC LC-39A, into either orbit mentioned above would yield the greatest chance of having a successful landing.**

REFERENCES

- Python 3
- Jupyter Notebooks
- Plotly Dash
- Folium
- Pandas
- Numpy
- SKLearn
- Seaborn
- MatplotLib

APPENDIX

Plotly Dash App



APPENDIX

jupyter SpaceX_Machine Learning Prediction_Part_5 Last Checkpoint: 3 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

Import Libraries and Define Auxiliary Functions

We will import the following libraries for the lab

In [1]:

```
# Pandas is a software library written for the Python programming language for data manipulation and analysis.
import pandas as pd
# NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along
import numpy as np
# Matplotlib is a plotting library for python and pyplot gives us a MatLab like plotting framework. We will use this in our ploti
import matplotlib.pyplot as plt
#Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive ar
import seaborn as sns
# Preprocessing allows us to standarsize our data
from sklearn import preprocessing
# Allows us to split our data into training and testing data
from sklearn.model_selection import train_test_split
# Allows us to test parameters of classification algorithms and find the best one
from sklearn.model_selection import GridSearchCV
# Logistic Regression classification algorithm
from sklearn.linear_model import LogisticRegression
# Support Vector Machine classification algorithm
from sklearn.svm import SVC
# Decision Tree classification algorithm
from sklearn.tree import DecisionTreeClassifier
# K Nearest Neighbors classification algorithm
from sklearn.neighbors import KNeighborsClassifier
```

This function is to plot the confusion matrix.

In [2]:

```
def plot_confusion_matrix(y,y_predict):
    "this function plots the confusion matrix"
    from sklearn.metrics import confusion_matrix

    cm = confusion_matrix(y, y_predict)
    ax= plt.subplot()
    sns.heatmap(cm, annot=True, ax = ax); #annot=True to annotate cells
    ax.set_xlabel('Predicted labels')
    ax.set_ylabel('True labels')
    ax.set_title('Confusion Matrix');
    ax.xaxis.set_ticklabels(['did not land', 'land']); ax.yaxis.set_ticklabels(['did not land', 'landed'])
```

Jupyter Notebooks
Python
Libraries