Winning Space Race with Data Science

Michael Doggrell
5/8/2023

Outline

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

Executive Summary

- Classification models for logistic regression, SVM (support vector machine), decision tree classifier, and KNN (K nearest neighbors) were tested
- All models were processed to determine best values of hyperparameters with GridSearchCV
- All four models accurately predicted the outcome of 15 of 18 launches, and inaccurately predicted 3 launches as successful that actually were not

Introduction

SpaceX has become one of the first successful commercial rocket launch companies. They have accomplished this largely by successfully creating a process that allows the main booster stage of the rockets to return to Earth safely and be re-used. By re-using the boosters, it allows the company to offer its services at a significantly lower cost compared with other commercial providers.

What factors contribute to the successful recovery of a rocket booster? In this study we will examine these factors so that the results can be perfected, and perhaps duplicated by another company, SpaceY.

Github Data Repository Links

All of the files referenced in this document can be found in my Github data repository:

https://github.com/michdoggr/data-science-capstone-project

I set all my Jupiter notebook hyperlinks in this document to open with nbviewer, because that seems to work the best, however they can still be viewed or downloaded from Github.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

  The data used here was collected from two sources

  - JSON files imported from the SpaceX API, such as Rockets, Launchpads, Payloads, and Cores
  - The launch info from the List of Falcon 9 and Falcon Heavy launches web page, using web scraping techniques from the Python BeautifulSoup package

- Perform data wrangling

  - The data was combined into a Pandas dataframe
  - Data for Falcon 1 launches was deleted, since we are only interested in Falcon 9 launches for this project

Methodology (cont.)

- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - The data was separated into training and testing datasets for analysis using several models for comparison

Data Collection

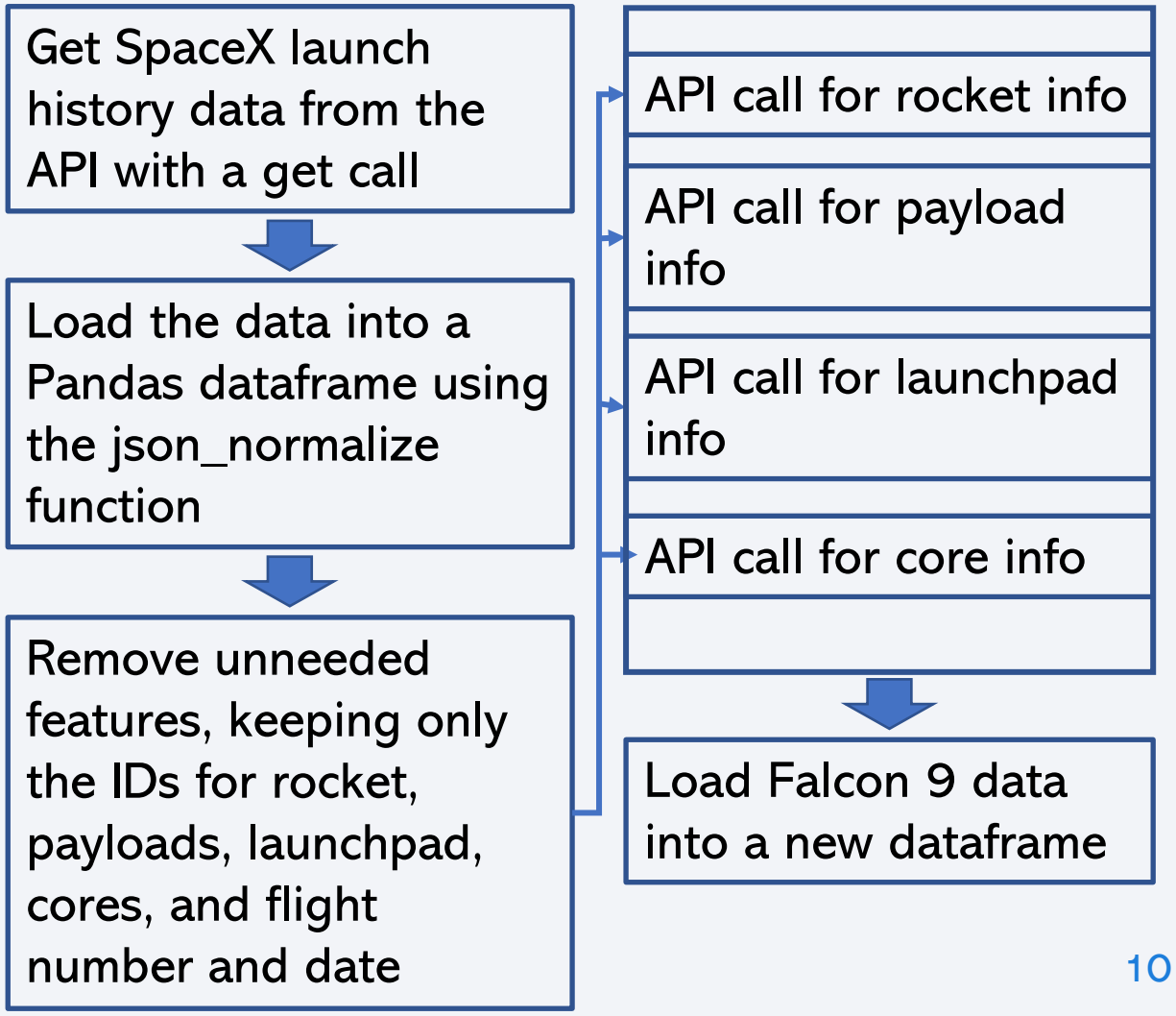
The data used here was collected from two sources

- JSON files imported from the SpaceX API, such as Rockets, Launchpads, Payloads, and Cores
- The launch info from the List of Falcon 9 and Falcon Heavy launches web page, using web scraping techniques from the Python BeautifulSoup package

A detailed description of the data collection process will be provided next.

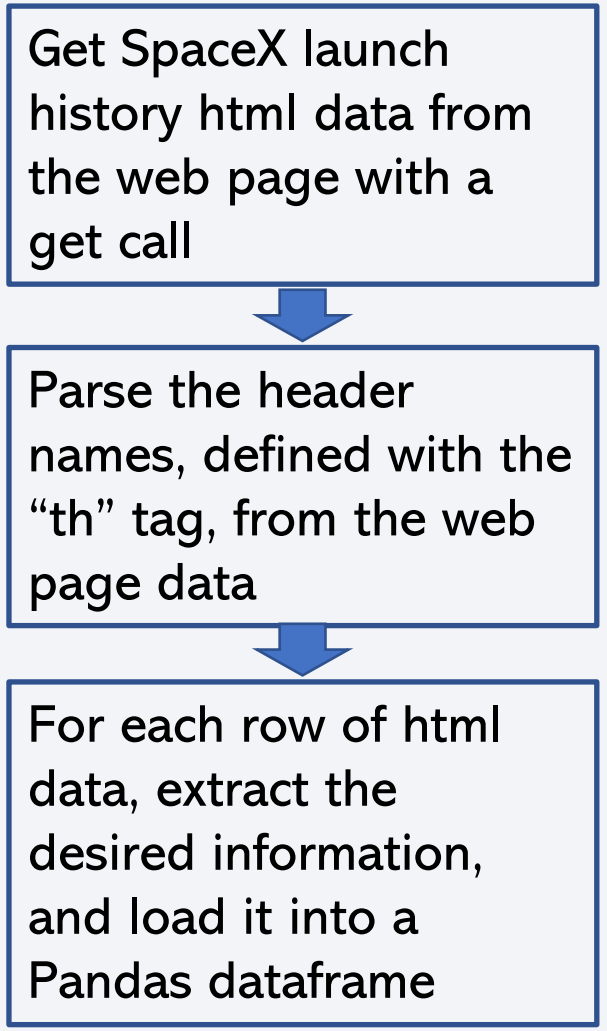
Data Collection – SpaceX API

- The basic process is to obtain the needed information from the SpaceX API, load it into a Pandas dataframe, and then begin the cleaning and formatting steps
- GitHub URL: https://github.com/michdoggr/data-science-capstone-project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- The process is to obtain the needed information from the SpaceX Wikipedia page, use BeautifulSoup to parse the header names out to be used for column names, then extract the needed data from each row of html, load it into a Pandas dataframe, and then begin the cleaning and formatting steps
- GitHub URL: https://github.com/michdoggr/data-science-capstone-project/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling

- First, only the data for Falcon 9 launches was retained in the data frame
- Some launches were missing the payload mass, so the mean payload mass was calculated, and then used to replace the null values
- The number of rows with missing values was determined, along with the data type of each column
- Totals were calculated for the number of launches at each site, the number of launches for the different orbit types, and the count of each landing outcome
- A new column called class was added, with a value of 1 for successful outcomes, and 0 for unsuccessful outcomes, and it was populated based on the value of the landing outcome. This is the target value of the prediction modeling.

Data Wrangling (cont.)

- After the exploratory data analysis, the following features were chosen for the predictive model: FlightNumber, PayloadMass, Orbit, LaunchSite, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial
- Most of the features are self-explanatory, but here some additional info on ones that may not be clear: grid fins and legs refer to whether those were used on the flight, reused is true if the core was reused, and reused count is the number of times it was reused, and block is used to differentiate versions of cores
- The following features were converted to categorical dummy variables using the Pandas get_dummies method and one hot encoding for use in the predictive modeling process: Orbit, LaunchSite, LandingPad, Serial
- All numeric variables were converted to float type
- GitHub URL: https://github.com/michdoggr/data-science-capstone-project/blob/main/labs-jupyter-spacex-data_wrangling.ipynb

EDA with Data Visualization

The following charts were created for the EDA with Data Visualization phase. Scatter plots indicate landing success with an orange dot, and failure with blue. We want to visualize these so clear patterns and relationships can be seen:

- Scatter plot of payload vs. flight number
- Scatter plot of launch site vs. flight number
- Scatter plot of launch site vs. payload
- Bar chart showing success rate for each orbit type
- Scatter plot of orbit vs. flight number
- Scatter plot of orbit vs. payload
- Line chart showing success rate trend by year

EDA with Data Visualization (cont.)

- GitHub URL: https://github.com/michdoggr/data-science-capstone-project/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

Queries were created to pull the following data:

- Names of launch sites, and some launch sites that begin with CCA
- Calculate the total payload carried for customer NASA (CRS)
- Calculate the average payload carried by booster F9 V1.1
- Date of first successful ground landing
- Successful drone ship landings with payload between 4,000 and 6,000
- Count of success and failure mission outcomes
- Boosters that carried the maximum payload
- Failed drone ship landings in 2015

EDA with SQL (cont.)

- Booster version and launch site for successful landings between 6/4/2010 and 3/20/2017

- GitHub URL: https://github.com/michdoggr/data-science-capstone-project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- This map displays each SpaceX launch site marked by a circle
- There are clickable cluster markers which will display green or red markers indicating the success or failure of missions launched from that site
- For VAFB, a few geographical features were selected, and a line from the launch site, and the distance of the feature are displayed
- These were added to provide a bird's eye view of where the launch sites are located, and especially their proximity to the ocean
- GitHub URL: https://github.com/michdoggr/data-science-capstone-project/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- The Plotly dashboard displays two interactive charts, a pie chart showing launch sites and their success rate, and a scatter plot showing success or failure by payload range.
- A drop down selection is provided to select a launch site, or all launch sites to be displayed on each of the charts, and there is also a slider selection bar used by the payload chart.
- These were added to allow the user to focus on just the information they are interested in at a given time
- GitHub URL: https://github.com/michdoggr/data-science-capstone-project/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- First the target variable of class (landing outcome) was separated from the other features to be used in the modeling process
- As class is a categorical variable with 1 for success and 0 for failure, models intended for predicting a categorical variable (as opposed to a numeric value with a continuous range of values) were used
- The data was normalized using the sklearn StandardScaler method
- The train_test_split method was used to split the available data into train and test sets, with 80% used for training, and 20% used for testing
- GitHub URL: https://github.com/michdoggr/data-science-capstone-project/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.ipynb

Results

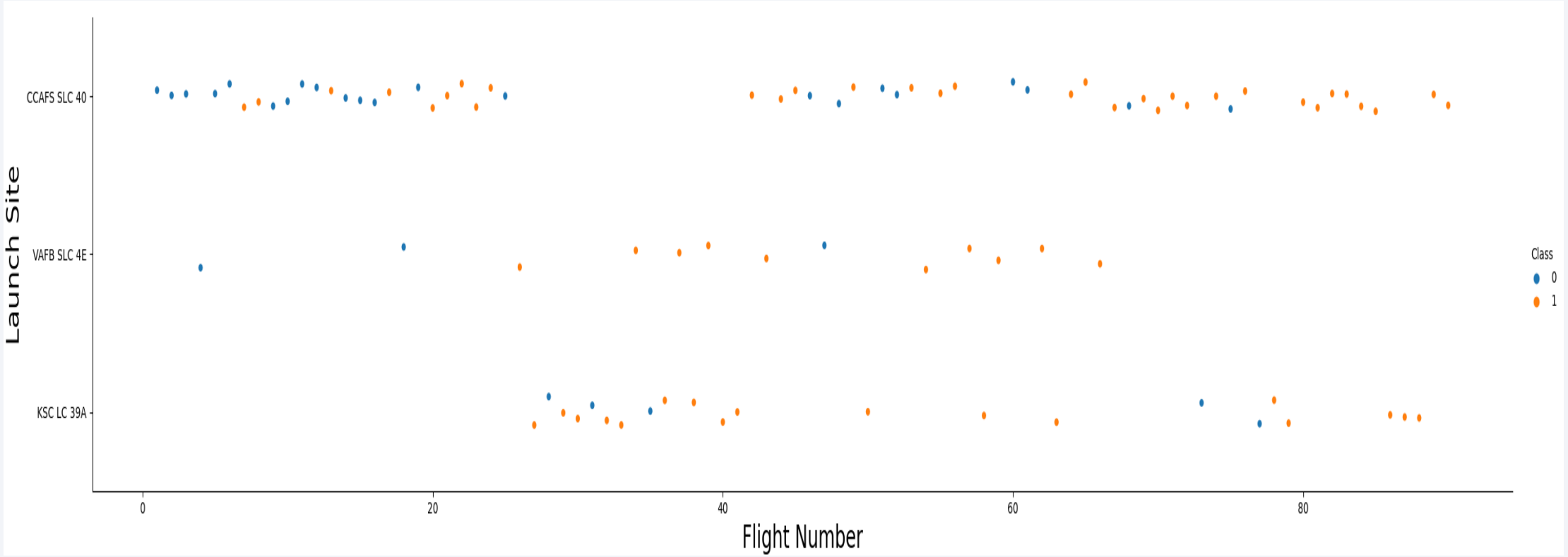
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

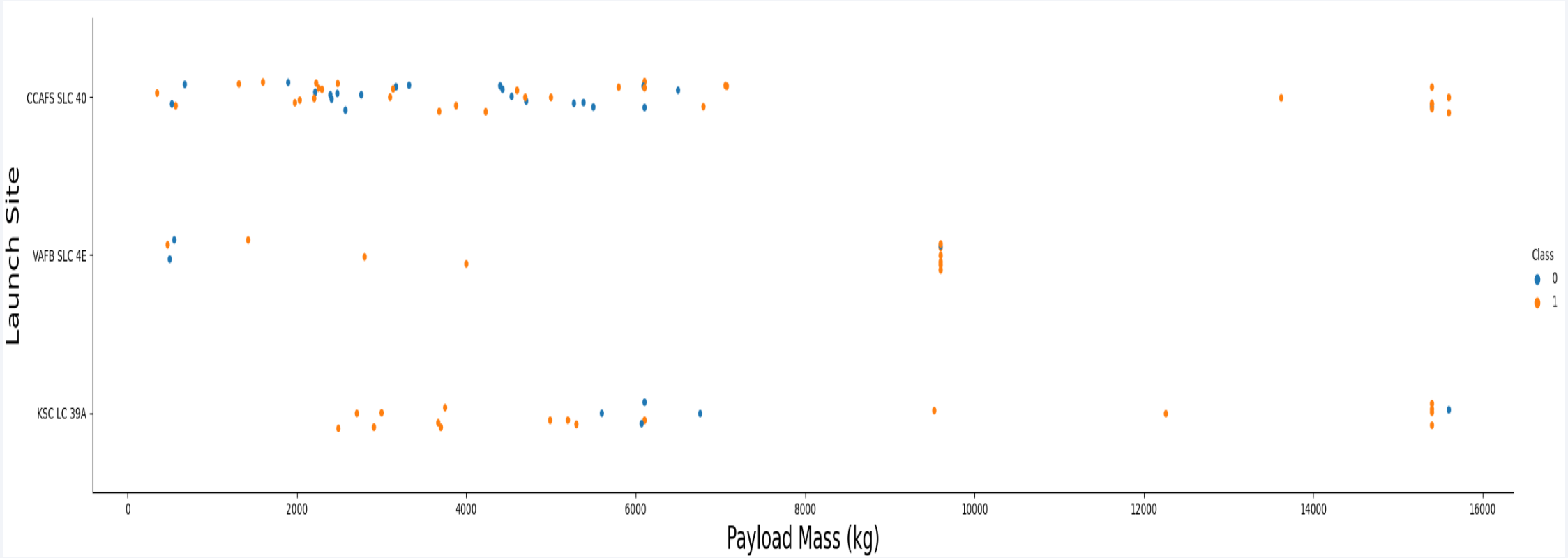
Flight Number vs. Launch Site

- Flights from each launch site with success or failure indicated



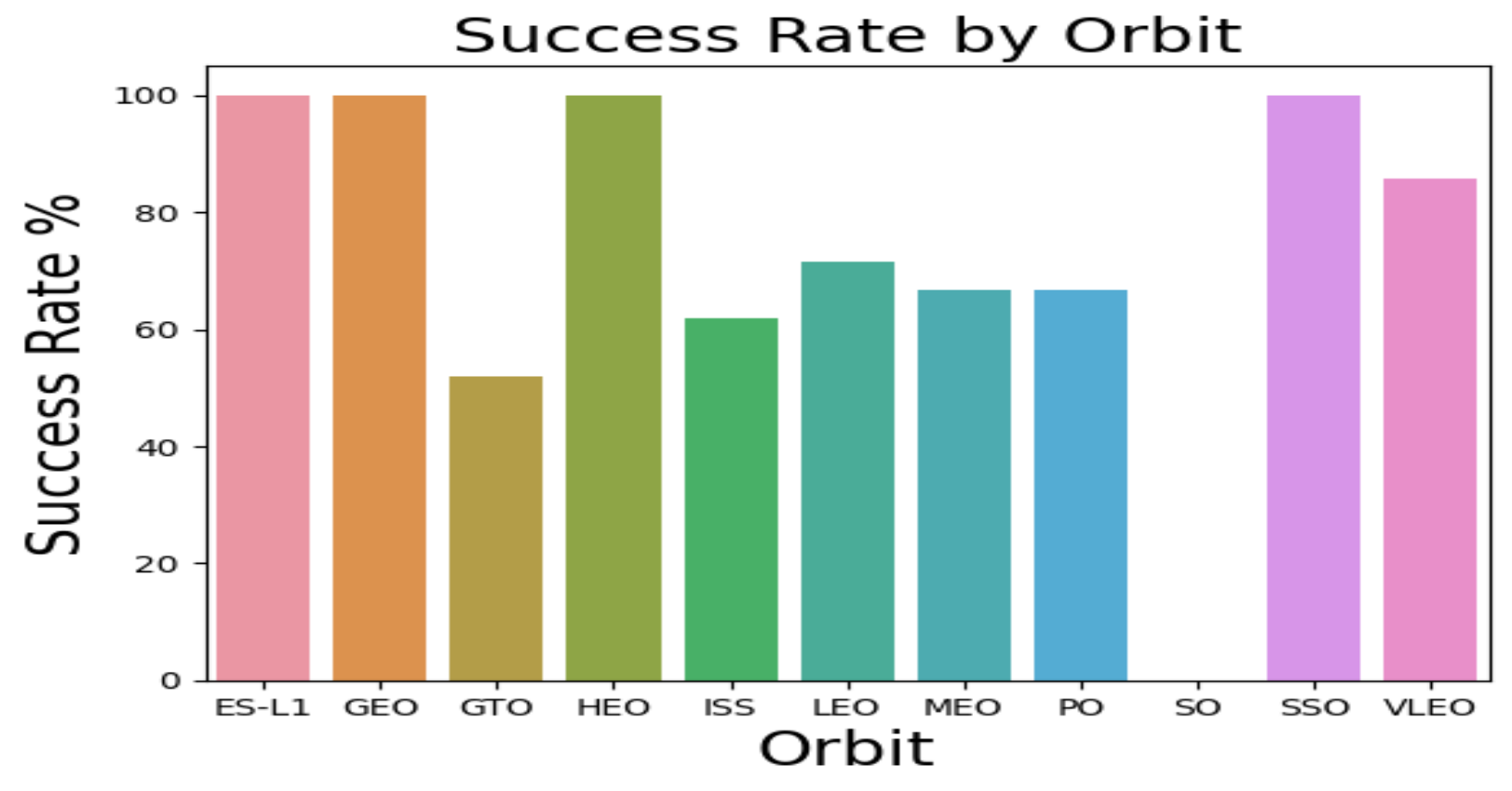
Payload vs. Launch Site

- Payload mass for launch sites with success or failure indicated



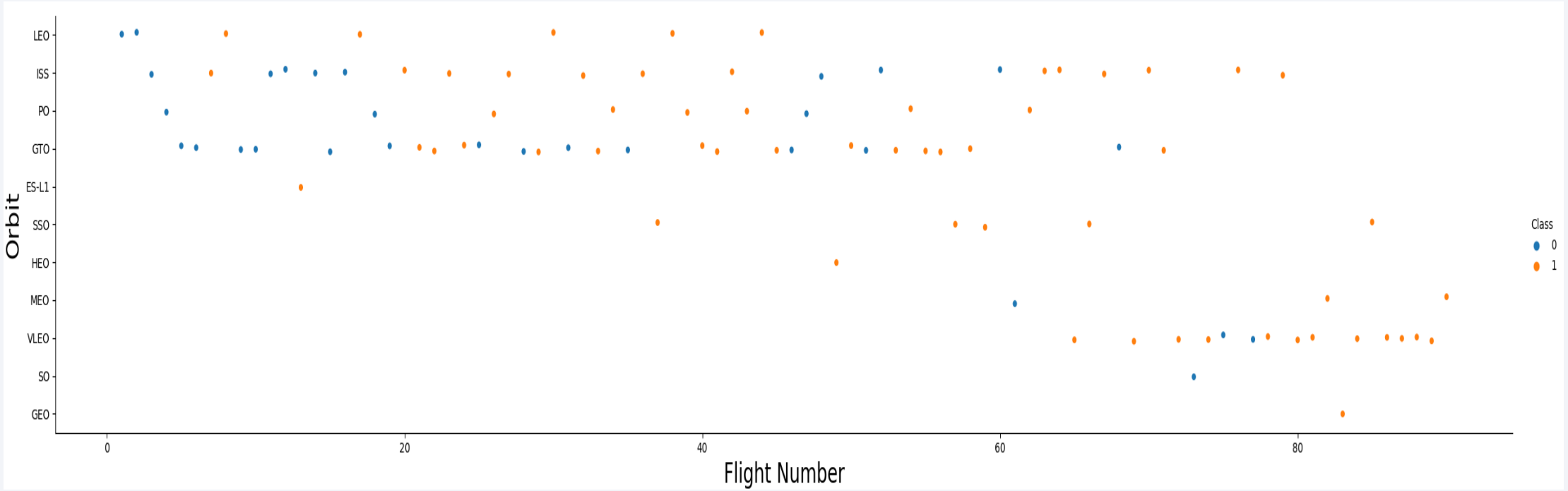
Success Rate vs. Orbit Type

- Success rate by orbit type



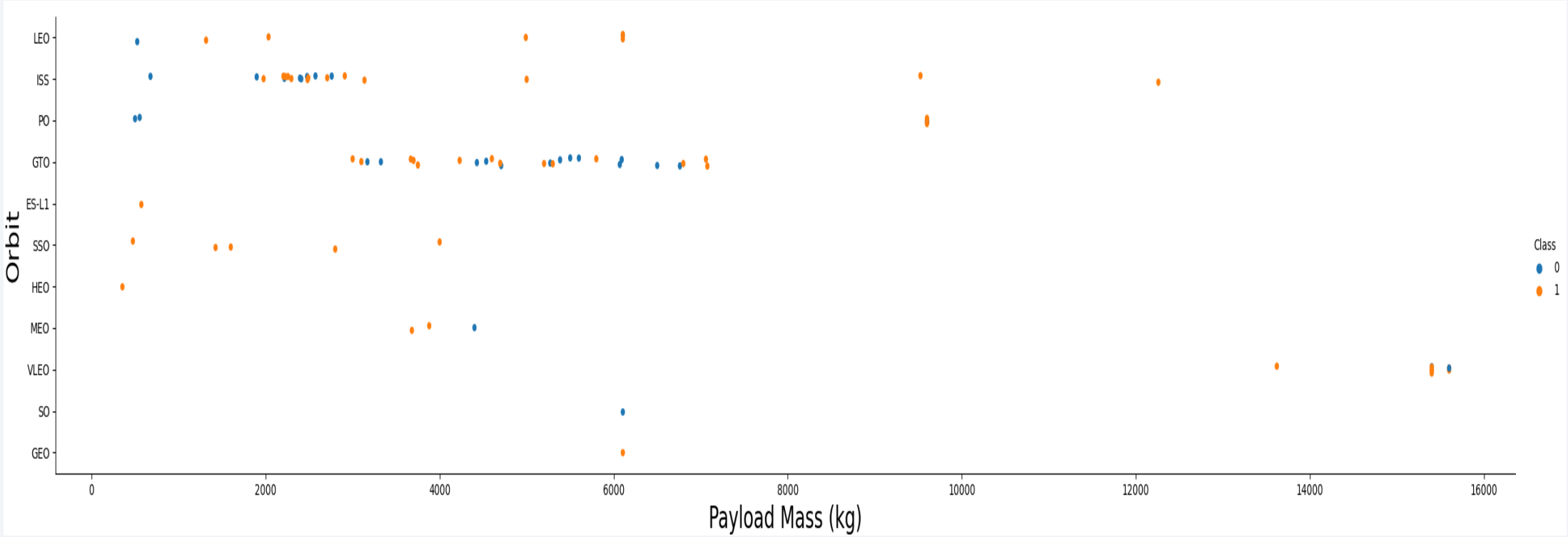
Flight Number vs. Orbit Type

- Orbit types for each flight, with success or failure indicated



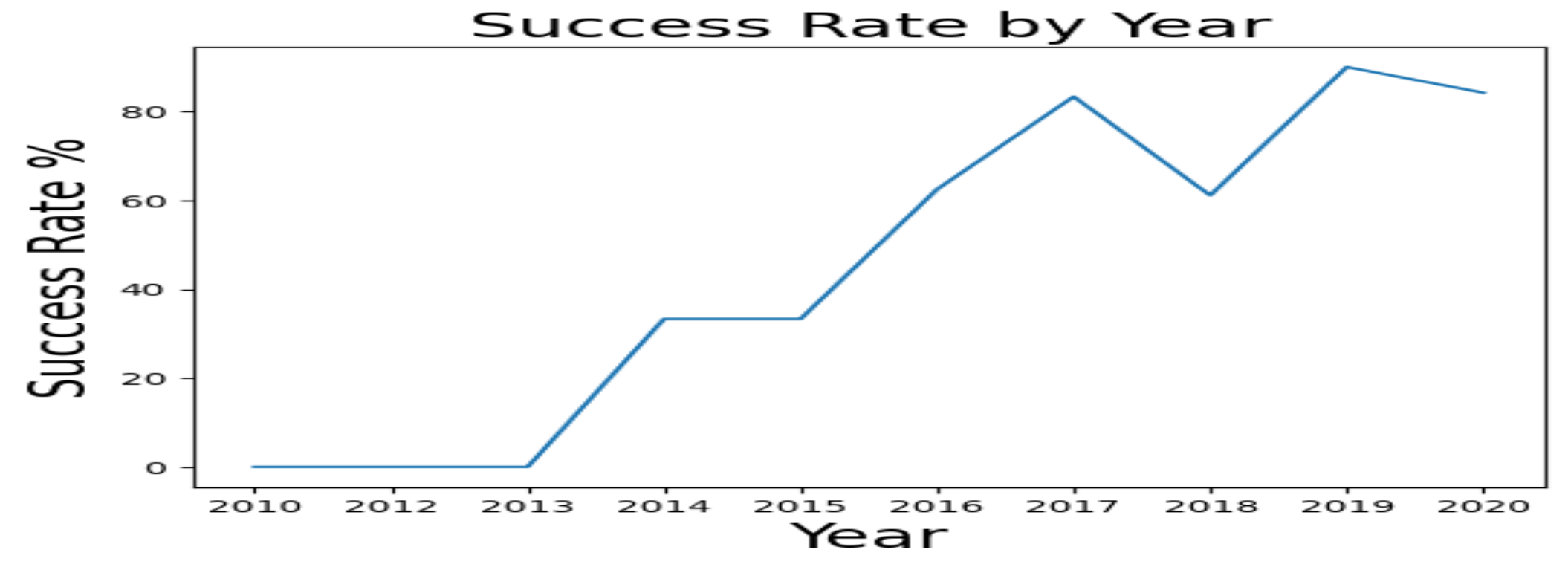
Payload vs. Orbit Type

- Orbit type and payload mass, with success or failure indicated



Launch Success Yearly Trend

- The success rate was zero for the first three years of the program, then it mainly increased after that, with a couple of dips



All Launch Site Names

- Note: The SQL used for the query results that follow are shown in the appendix

- Launch sites used by SpaceX

Results:

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

Launch Site Names Begin with 'CCA'

- Five records where launch sites begin with `CCA`

Result:

Date	Time (UTC)	BoosterVersion	LaunchSite	Payload	PAYLOADMASSKG	Orbit	Customer	MissionOutcome	LandingOutcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Total payload carried for customer NASA (CRS)

Results:

sum(payloadmasskg)
45596

Average Payload Mass by F9 v1.1

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

Results:

avg(payloadmasskg)
2928.4

First Successful Ground Landing Date

- First successful landing outcome on a ground pad was on this date

Results:

Date
20151222

Successful Drone Ship Landing with Payload between 4000 and 6000

- Boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Results:

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

Total Number of Successful and Failure Mission Outcomes

- Count of missions where the overall mission outcome was a success of failure

Results:

Successful Outcomes	Failure Outcomes
100	1

Boosters Carried Maximum Payload

- These are the boosters which have been used on the launches carrying maximum payload (15,600 kg)

Results:

BoosterVersion
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

- Failed landing outcomes using a drone ship, with booster version and launch site, for 2015

Results:

Month	BoosterVersion	LaunchSite	LandingOutcome
01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- Successful landings between the date 2010-06-04 and 2017-03-20

Results:

Date	BoosterVersion	LaunchSite	LandingOutcome
20170219	F9 FT B1031.1	KSC LC-39A	Success (ground pad)
20170114	F9 FT B1029.1	VAFB SLC-4E	Success (drone ship)
20160814	F9 FT B1026	CCAFS LC-40	Success (drone ship)
20160718	F9 FT B1025.1	CCAFS LC-40	Success (ground pad)
20160527	F9 FT B1023.1	CCAFS LC-40	Success (drone ship)
20160506	F9 FT B1022	CCAFS LC-40	Success (drone ship)
20160408	F9 FT B1021.1	CCAFS LC-40	Success (drone ship)
20151222	F9 FT B1019	CCAFS LC-40	Success (ground pad)


Section 3
Launch Sites
Proximities Analysis

SpaceX Launch Sites

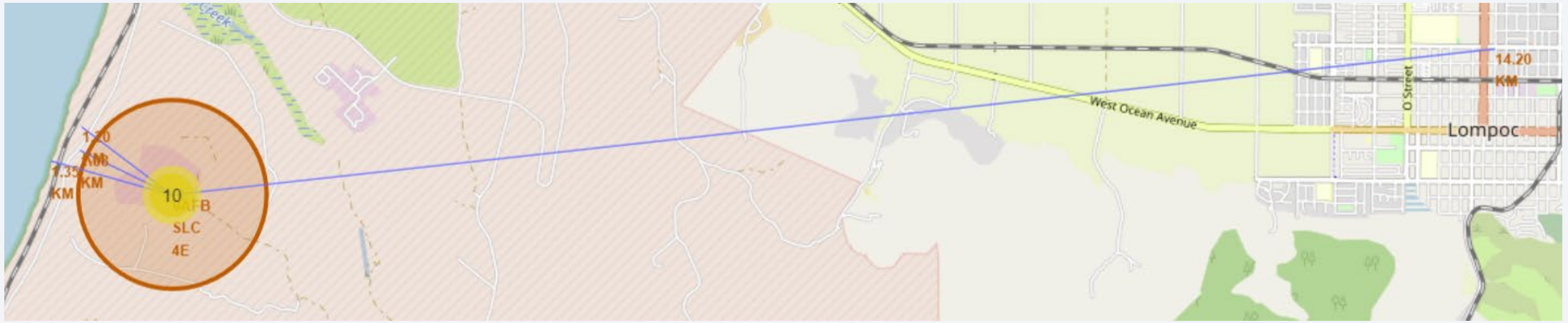
- The launch locations are near both the east and west coasts, at previously established rocket launch facilities

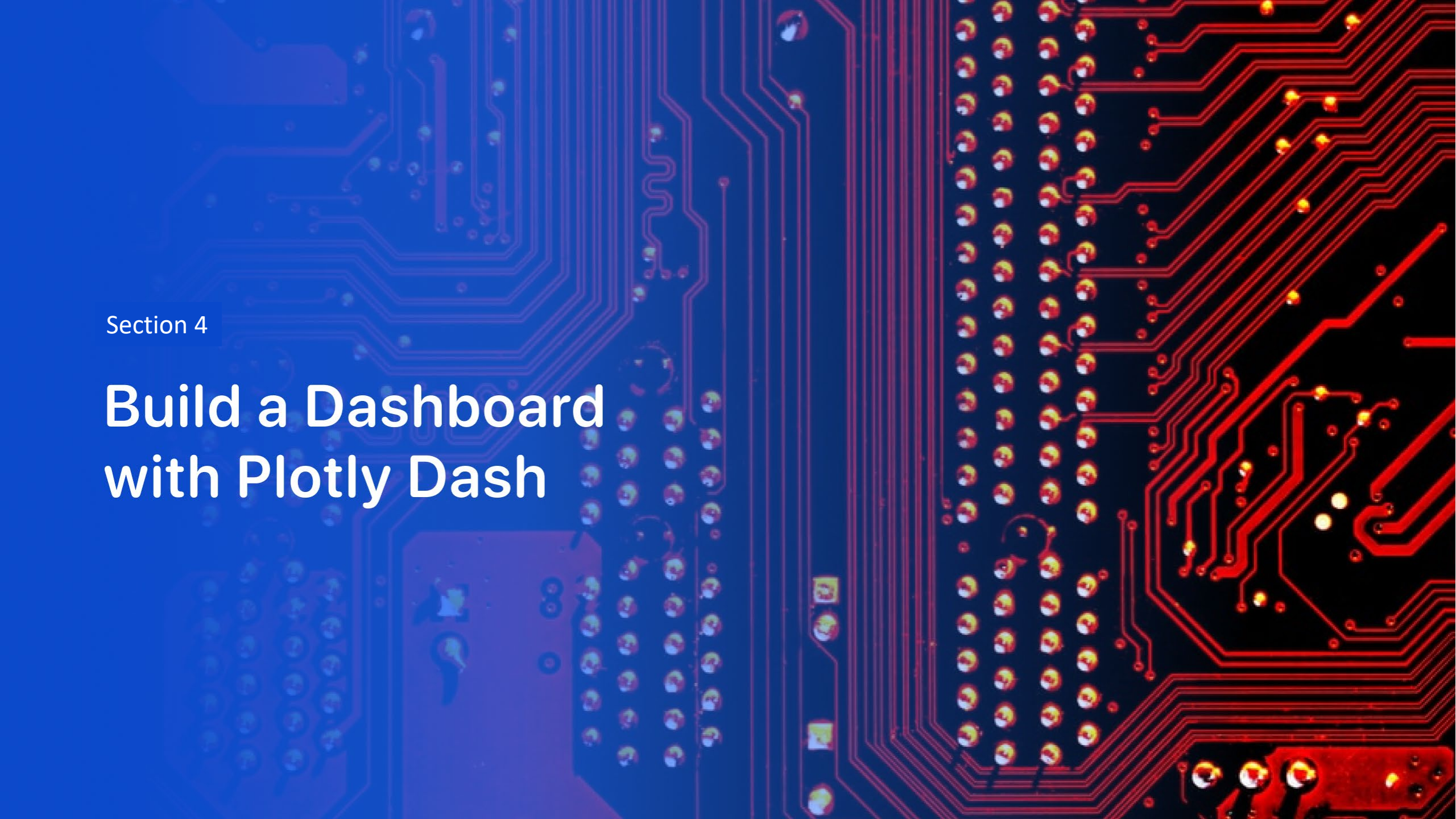
Mission Outcomes at Launch Sites

- The green (successful) and red (unsuccessful) markers provide a quick overview of the success rate at the different launch sites.

Proximity of Launch Site to Geographic Features

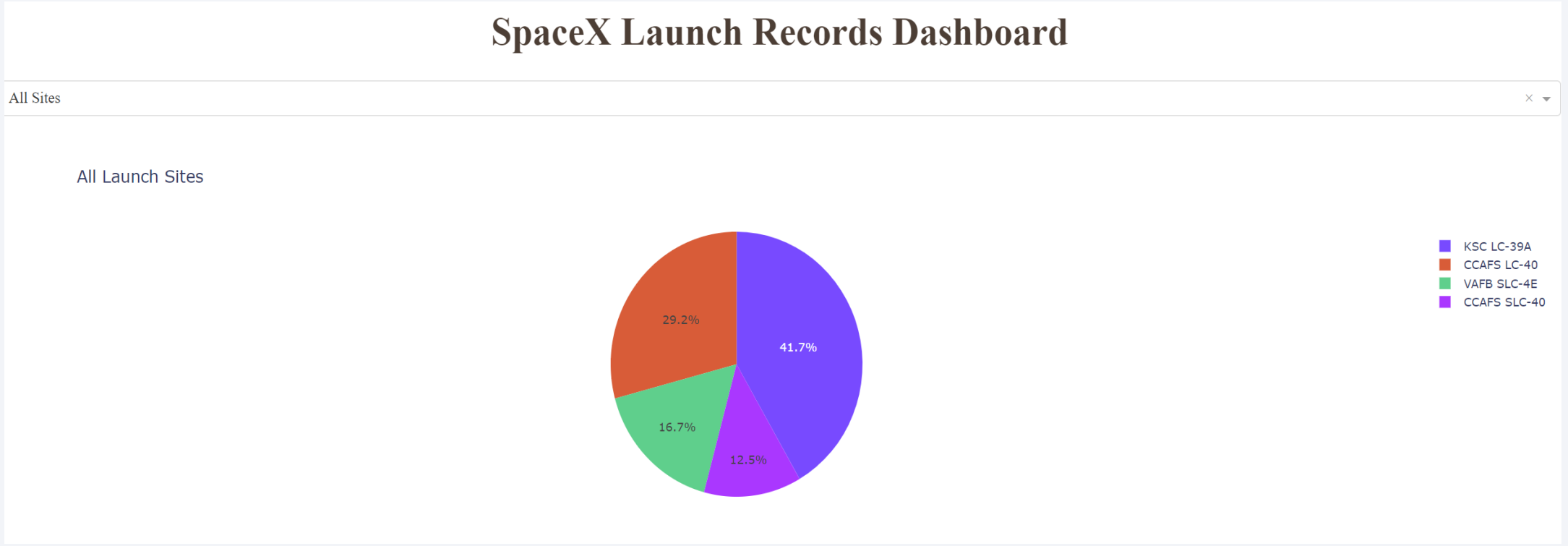
- The launch sites are location near the coast, and away from towns and cultural features where a mishap could cause damage or injury to other people and property




Section 4
Build a Dashboard with Plotly Dash

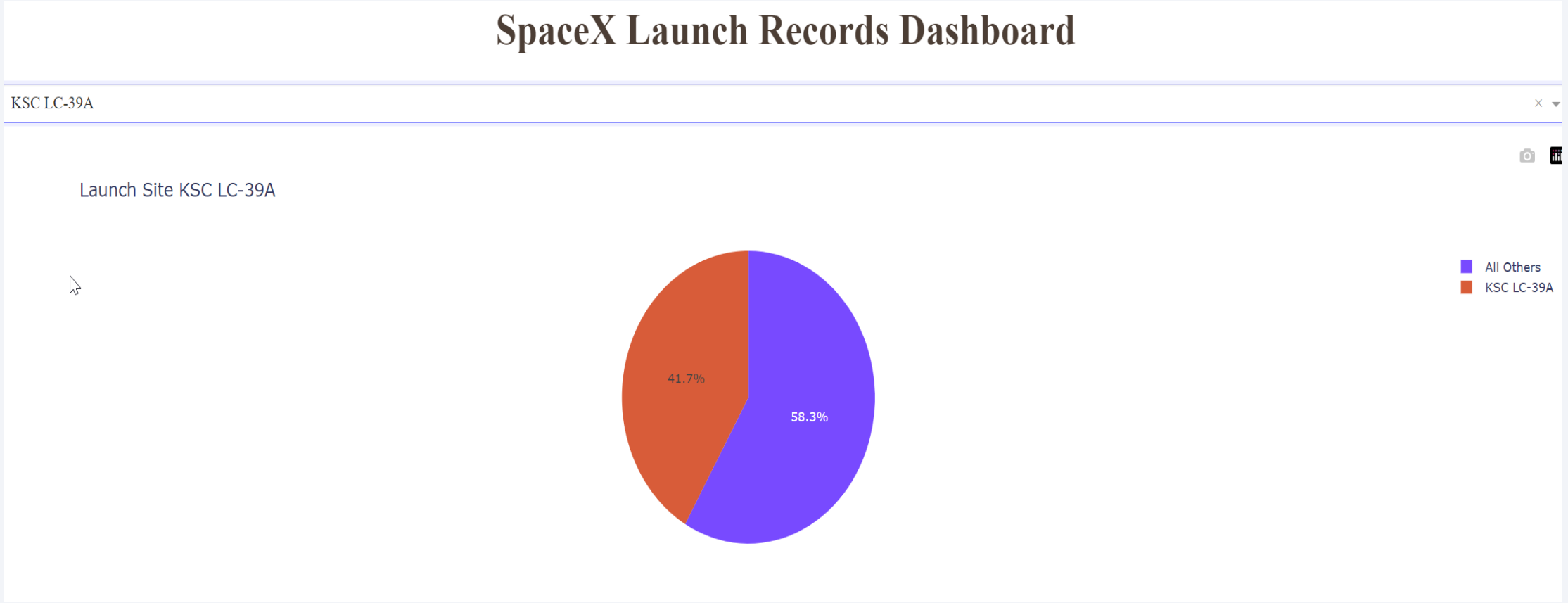
SpaceX Launch Success Rate by Launch Site

The most successful launch sites, KSC LC-39A, and CCAFS LC-40 are in the Cape Canaveral area, which has a long history as a center of rocket launches



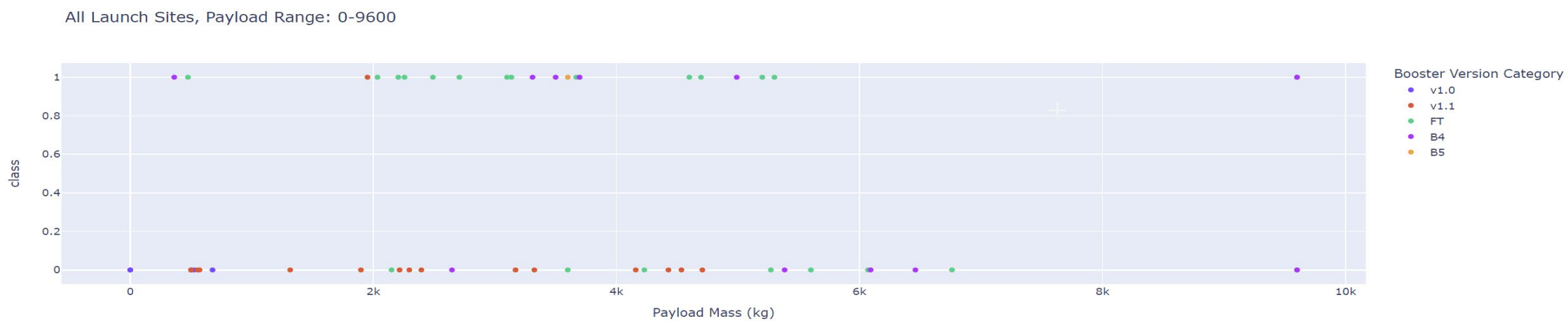
KSC LC-39A Success Rate and All Others

KSC stands for Kennedy Space Center, enough said!



Launch Success vs. Payload Size w/Booster Cat

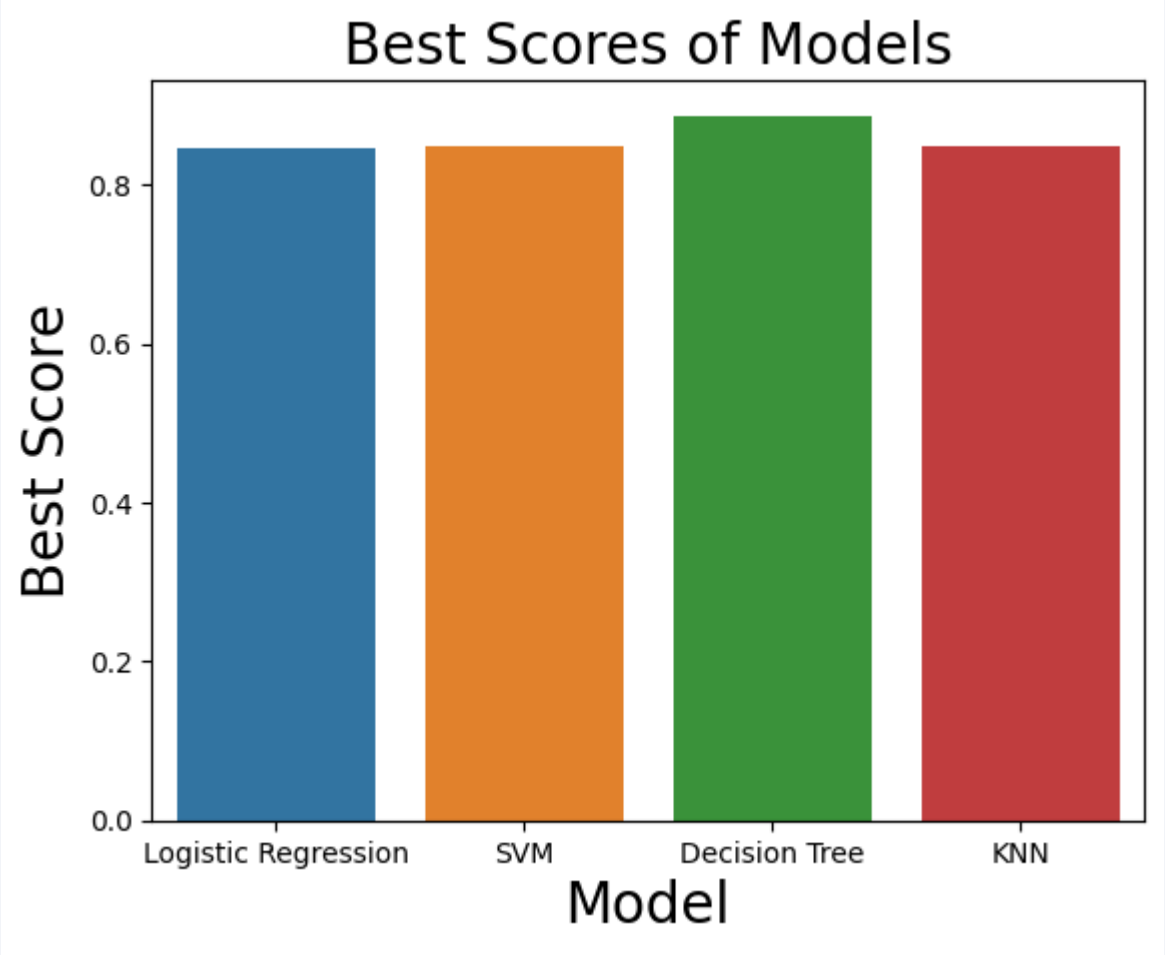
The FT and B4 booster categories account for almost all of the successes



Section 5

Predictive Analysis (Classification)

Classification Accuracy



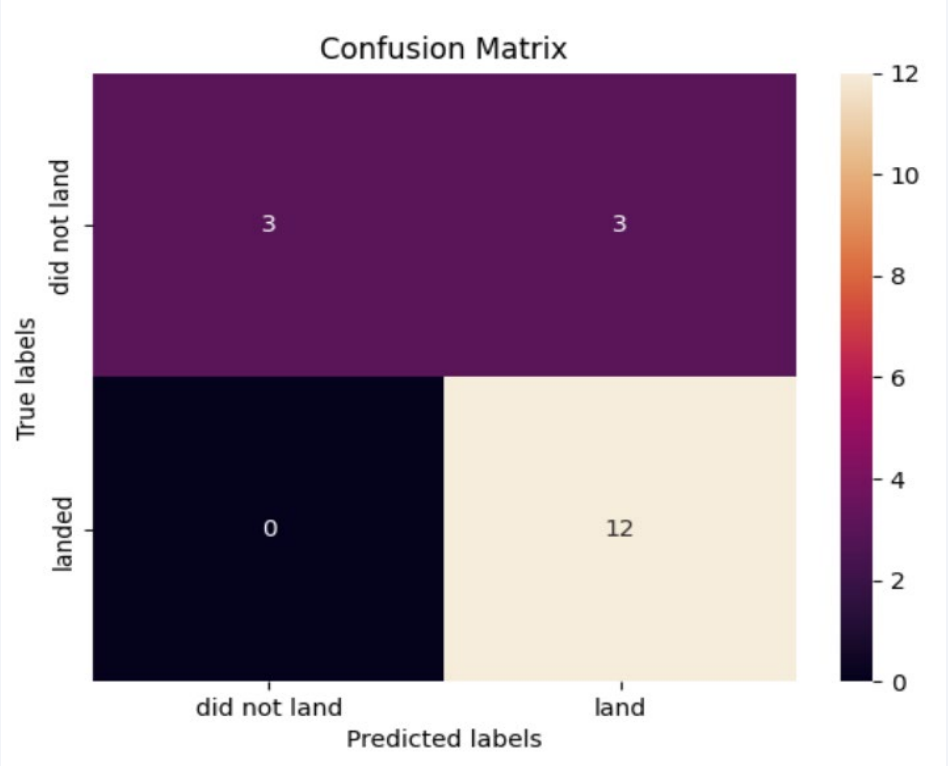
Classification Accuracy

All of the models had the same accuracy score of 0.8333 using the “score” method, but the GridSearchCV best scores were close, except for the decision tree, which had a higher score:

Logistic Regression	0.8464
SVM	0.8482
Decision Tree	0.8875
KNN	0.8482

Confusion Matrix

- While the decision tree model had the best GridSearchCV accuracy score of 0.8875, all of the models had the same confusion matrix, showing 12 accurately predicted landings, 3 accurate failed landings, and 3 inaccurately predicted failed landing.



Conclusions

A dataset showing the test values and predicted values can be found here:

Test Values and Predicted Values With Features

- The three incorrectly predicted landings were flights number 75, 14, and 12
- All of the incorrectly predicted landings were launched from site CCAFS SLC 40
- All of the incorrectly predicted landings used both legs and grid fins
- The payload mass on the three incorrect flight predictions were 15,400, 1,898, and 2,395, so the incorrect predictions don't appear correlated with payload
- All of the models performed about the same and gave a pretty good prediction of the landing outcome at 83% correct

Model Parameters

```
Train set: (72, 83) (72,)
Test set: (18, 83) (18,)

GridSearchCV(cv=10, estimator=LogisticRegression(),
             param_grid={'C': [0.01, 0.1, 1], 'penalty': ['l2'],
                         'solver': ['lbfgs']})

tuned hpyerparameters :(best parameters)  {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
accuracy : 0.8464285714285713

GridSearchCV(cv=10, estimator=SVC(),
             param_grid={'C': array([1.00000000e-03, 3.16227766e-02, 1.00000000e+00, 3.16227766e+01,
       1.00000000e+03]),
                         'gamma': array([1.00000000e-03, 3.16227766e-02, 1.00000000e+00, 3.16227766e+01,
       1.00000000e+03]),
                         'kernel': ('linear', 'rbf', 'poly', 'rbf', 'sigmoid')})

tuned hpyerparameters :(best parameters)  {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
accuracy : 0.8482142857142856

GridSearchCV(cv=10, estimator=DecisionTreeClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [2, 4, 6, 8, 10, 12, 14, 16, 18],
                         'max_features': ['auto', 'sqrt'],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [2, 5, 10],
                         'splitter': ['best', 'random']})

GridSearchCV(cv=10, estimator=KNeighborsClassifier(),
             param_grid={'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
                         'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],

tuned hpyerparameters :(best parameters)  {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
accuracy : 0.8482142857142858                         'p': [1, 2]})
```

Appendix

- API endpoints used for collecting JSON data:

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

  https://api.spacexdata.com/v4/rockets/

  https://api.spacexdata.com/v4/launchpads/

  https://api.spacexdata.com/v4/payloads/

  https://api.spacexdata.com/v4/cores/

- Test datasets: Test dataset with features, Test dataset after OneHotEncoding
- SpaceX Wikipedia page: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- Queries used:

  All Launch Site Names: select distinct(LaunchSite) from SPACEXTBL

  Launch Site Names Begin with 'CCA': select * from SPACEXTBL where LaunchSite like 'CCA' limit 5

Appendix

Total Payload Mass: select sum(payloadmasskg) from SPACEXTBL where Customer = 'NASA (CRS)'

Average Payload Mass by F9 v1.1: select avg(payloadmasskg) from SPACEXTBL where BoosterVersion = 'F9 v1.1'

First Successful Ground Landing Datey: select min(substr(Date,7,4)||substr(Date,4,2)||substr(Date,1,2)) as Date from SPACEXTBL where LandingOutcome = 'Success (ground pad)'

Successful Drone Ship Landing with Payload between 4000 and 6000: select distinct(BoosterVersion) from SPACEXTBL where LandingOutcome = 'Success (drone ship)'and PAYLOADMASSKG between 4000 and 6000

Total Number of Successful and Failure Mission Outcomesy: select * from (select count(*) as 'Successful Outcomes' from SPACEXTBL where MissionOutcome like 'Success') as S, (select count(*) as 'Failure Outcomes' from SPACEXTBL where MissionOutcome not like 'Success')

Boosters Carried Maximum Payload: select distinct(BoosterVersion) from SPACEXTBL where PAYLOADMASSKG = (select max(PAYLOADMASSKG) FROM SPACEXTBL)

Appendix

2015 Launch Records: select substr(Date,4,2) as Month, BoosterVersion, LaunchSite, LandingOutcome from SPACEXTBL where LandingOutcome = 'Failure (drone ship)' and substr(Date,7,4) = '2015'

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20: select substr(Date,7,4)||substr(Date,4,2)||substr(Date,1,2) as Date, BoosterVersion, LaunchSite, LandingOutcome from SPACEXTBL where LandingOutcome like 'Success' and substr(Date,7,4)||substr(Date,4,2)||substr(Date,1,2) between '20100604' and '20170320' order by substr(Date,7,4)||substr(Date,4,2)||substr(Date,1,2) descResults:

Thank you!