IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Diego Mendez
4/3/2023

Outline

Executive Summary

Introduction

Methodology

Results

Conclusion

Appendix

Executive Summary

- Data on SpaceX rocket launches was collected from an API and Webscraping, and cleaned in order to perform exploratory data analysis which provided insights on what variables would be most suitable for predicting successful rocket landing outcomes using machine learning models.
- Exploratory analysis showed that factors to consider when predicting successful rocket landing outcomes included: flight number, launch site, payload mass, and Orbit. Training four machine learning models with these variables resulted in equally accurate models for predicting successful rocket landing outcomes. Being able to predict successful landing outcomes will help our company, SpaceY, stay competitive in the space race.

Introduction

- The space race is heating up again and one company is dominating the industry: SpaceX. The main driving force for their dominance is their ability to successfully land the first stage of their rockets which dramatically reduces their costs. As a new rocket company, SpaceY must look to the leader of the industry if it wants to be competitive and win bids.
- How can SpaceY be competitive with the industry leader?
  - Determining the cost of SpaceX launches by predicting successful landing outcomes can help win bids.
  - Determining what factors play a role in successful landing outcomes can help keep our own costs down.
- Can we accurately predict if a rocket launch will have a successful landing of the first stage?
  - What variables should we consider for our own rocket launches?

Section 1

Methodology

METHODOLOGY

Executive Summary

Data collection methodology:

- Data requested from the SpaceX API
- Data collected by web scraping SpaceX's Wikipedia page

Data wrangling

- Missing values replaced with means
- One Hot encoding to transform categorical variables (i.e. Orbit)

Exploratory data analysis (EDA) using visualization and SQL

- 6 plots and 10 SQL queries to gather insights from the collected data

Interactive visual analytics using Folium and Plotly Dash

- Interactive map with launch sites marked as well as proximities to key places
- Interactive Dashboard with two plots

Predictive analysis using classification models

- Standardized data, split into test and train sets, and used GridSearchCV to find the best hyperparameters for 4 models: Logistic Regression, SVM, Decision Tree & KNN.

Data Collection

Data requested from the SpaceX API

1. Request Data from SpaceX API
2. Convert result to a DataFrame
3. Update DataFrame IDs to values calling SpaceX API again
4. Filter DataFrame to only include Falcon 9 launches

Data web scraped from the SpaceX Wikipedia web page

1. Request Falcon 9 HTML page and parse using BeautifulSoup
2. Locate table with pertinent information and store data in a dictionary
3. Convert dictionary into a DataFrame

Data Collection – SpaceX API

Request

Request Data from SpaceX API

- spacex_url="url"
- response=requests.get(spacex_url)
- response.status_code

Decode

Decode response content and turn into a DataFrame

- data=pd.json_normalize(response.json())

Replace

Replace ID data with actual values calling API again

- getBoosterVersion(data)
- getLaunchSite(data)
- getPayloadData(data)
- getCoreData(data)

Create

Create DataFrame from dictionary

- df=pd.DataFrame.from_dict(launch_dict)

GitHub URL: IBM_Applied_Data_Science_Capstone/1 - Data Collection API Lab.ipynb at main · Ketzaal/IBM_Applied_Data_Science_Capstone (github.com)

Data Collection - Scraping

Request

Request Falcon 9 Launch Wiki page from URL

- static_url="Wikipedia URL"
- response=requests.get(static_url).text

Create

Create BeautifulSoup object from response text

- soup=BeautifulSoup(response,'html.parser')

Find

Find appropriate table and create dictionary with its info

- first_launch_table=html_tables[2]
- launch_dict=dict.fromkeys(column_names)

Convert

Convert dictionary into DataFrame

- df=pd.DataFrame.from_dict(launch_dict,orient='index').transpose()

GitHub URL: IBM_Applied_Data_Science_Capstone/2 - Data Collection with Web Scraping Lab.ipynb at main · Ketzaal/IBM_Applied_Data_Science_Capstone (github.com)

Data Wrangling

Identify

Identify percentage of missing values in columns & replace with means

- df.isnull().sum()/df.count()*100
- avg_mass=data_falcon9['PayloadMass'].astype('float').mean(axis=0)
- data_falcon9['PayloadMass'].replace(np.nan,avg_mass,inplace=**True**)

Verify

Verify column data types are appropriate

- df.dtypes

Calculate

Calculate number of launches for each site and Orbits

- df['LaunchSite'].value_counts()
- df['Orbit'].value_counts()

Convert

Convert Landing Outcomes using One Hot Encoding into 2 categories: success(1), failure(0).

- landing_outcomes=df['Outcome'].value_counts()
- For i, outcome in enumerate(landing_outcomes.keys()):
- bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
- landing_class=[]for I in df['Outcome']: if I in bad_outcomes: landing_class.append(0) else: landing_class.append(1)landing_class

Save

Save one hot encoding result as new column: "Class"

- df['Class']=landing_class

GitHub URL: IBM_Applied_Data_Science_Capstone/3 - Data Wrangling Lab.ipynb at main · Ketzaal/IBM_Applied_Data_Science_Capstone (github.com)

EDA WITH DATA VISUALIZATION

A total of 6 charts were plotted to explore the data and answer the relevant questions

- Scatter plot of Flight Number vs. Launch Site
  - Is there a relationship between Flight Number/Launch Site and landing outcome?
- Scatter plot of Payload vs. Launch Site
  - Is there a relationship between Payload and Launch Site?
- Bar chart for the success rate of each orbit type
  - Is there a relationship between Orbit and Landing Outcome?
- Scatter plot of Flight number vs. Orbit type
  - Is there a relationship between Flight number/Orbit and Landing Outcome?
- Scatter plot of Payload vs. Orbit type
  - Is there a relationship between Payload/Orbit and Landing Outcome?
- Line chart of yearly average success rate
  - Has the success rate increased or decreased over time?

GitHub URL: IBM_Applied_Data_Science_Capstone/5 - EDA with Visualization Lab.ipynb at main · Ketzaal/IBM_Applied_Data_Science_Capstone (github.com)

EDA with SQL

A total of 10 SQL queries were made to achieve the following:

- Display names of unique Launch Sites
- Display 5 records where the launch site name begins with “CCA”
- Calculate total payload mass carried by boosters launched by NASA (CRS)
- Calculate average payload mass carried by booster version F9 v1.1
- Find date of first successful landing outcome in ground pad
- Find the name of boosters which have success in drone ship landings and have a payload mass between 4000 and 6000
- Calculate the total number of successful and failure mission outcomes
- Find the booster versions which have carried the maximum payload mass
- Find the booster version, launch site and month of launch for failed landing outcomes in drone ship in 2015
- Rank the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order

GitHub URL: IBM_Applied_Data_Science_Capstone/4 - EDA wtih SQL Lab.ipynb at main · Ketzaal/IBM_Applied_Data_Science_Capstone (github.com)

Interactive Map with Folium

- Circle and marker added to folium map for each launch site(4)
  - Circle/markers clearly indicate the location of launch sites on the map
- Markers and clusters added to indicate success and failure landing outcomes
  - Markers/clusters help visualize if geography of launch has an impact on successful landing
- Markers and lines added with distances to nearby railways, highways, coastline, and cities
  - Markers/lines help visualize the relationship between launch sites and their surroundings

GitHub: IBM_Applied_Data_Science_Capstone/6 - Interactive Visual Analytics with Folium Lab.ipynb at main · Ketzaal/IBM_Applied_Data_Science_Capstone (github.com)

Dashboard with Plotly Dash

- Dashboard consists of 2 plots
- Pie chart shows success/failure landing outcomes
  - Interactivity allows toggling between all launch sites or selected sites through a dropdown menu
  - Provides quick visual assessment and comparison of landing outcomes by launch site
- Scatter plot shows payload mass vs. landing outcome with colors distinguishing booster versions
  - Same interactivity as pie chart in addition to allowing the user to filter payload range data to view using a range slider
  - Provides quick visual assessment and comparison for different launch sites, payloads, and booster versions with respect to landing outcomes.

GitHub: IBM_Applied_Data_Science_Capstone/7 - Interactive Dashboard with Plotly Dash Lab.py at main · Ketzaal/IBM_Applied_Data_Science_Capstone (github.com)

Predictive Analysis (Classification)

Standardize

Standardize data
- transform=preprocessing.StandardScaler()
- X=transform.fit(X).transform(X)

Split

Split data into training and test data
- X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.2,random_state=2)

Create

Create model object and parameters dictionary
- lr=LogisticRegression()
- parameters={"C":[0.01,0.1,1],'penalty':['l2'],'solver':['lbfgs']}# l1 lasso l2 ridgelr=LogisticRegression()

Optimize

Optimize hyperparameters using GridSearchCV
- logreg_cv=GridSearchCV(lr,parameters,cv=10)

Train

Train the model
- logreg_cv.fit(X_train,Y_train)

Evaluate

Evaluate the accuracy of the model
- lr_score=lr_optimized.score(X_test,Y_test)

- Four Machine Learning classification models constructed using this process to predict landing outcome
1. Logistic Regression
2. Support Vector Machine (SVM)
3. Decision Tree
4. K-Nearest Neighbor (KNN)
- Standardized data to allow models such as KNN to be effective
- Split 20% of data into a test data set to evaluate model performance

GitHub: IBM_Applied_Data_Science_Capstone/8 - Machine Learning Prediction Lab.ipynb at main · Ketzaal/IBM_Applied_Data_Science_Capstone (github.com)

Results

EXPLORATORY DATA ANALYSIS RESULTS FROM VISUALIZATIONS AND SQL QUERIES

INTERACTIVE ANALYTICS DEMO IN SCREENSHOTS FROM PLOTLY DASH DASHBOARD

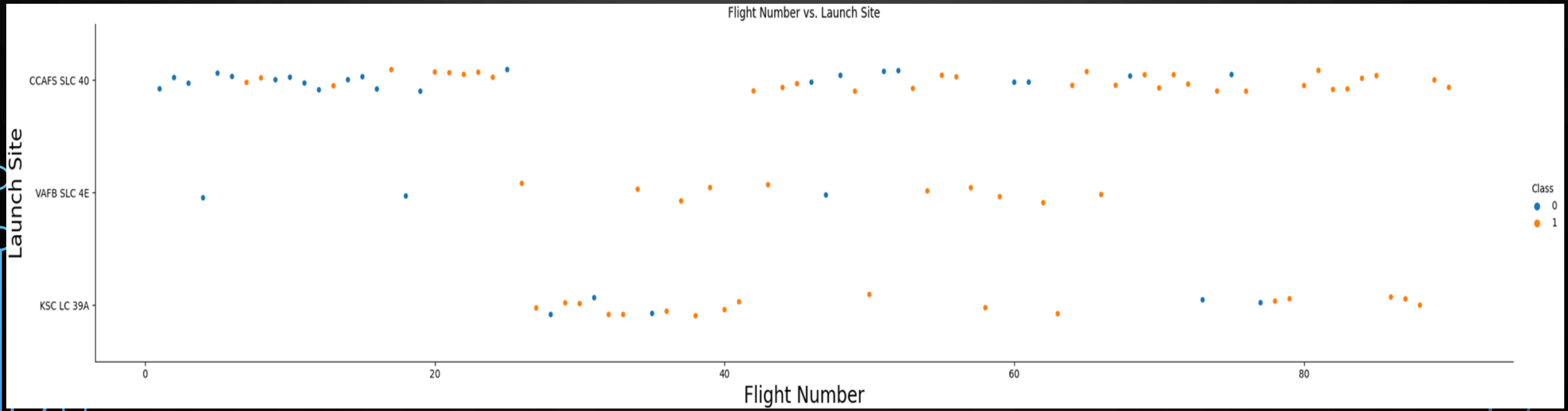
PREDICTIVE ANALYSIS RESULTS

Section 2

Insights drawn from EDA

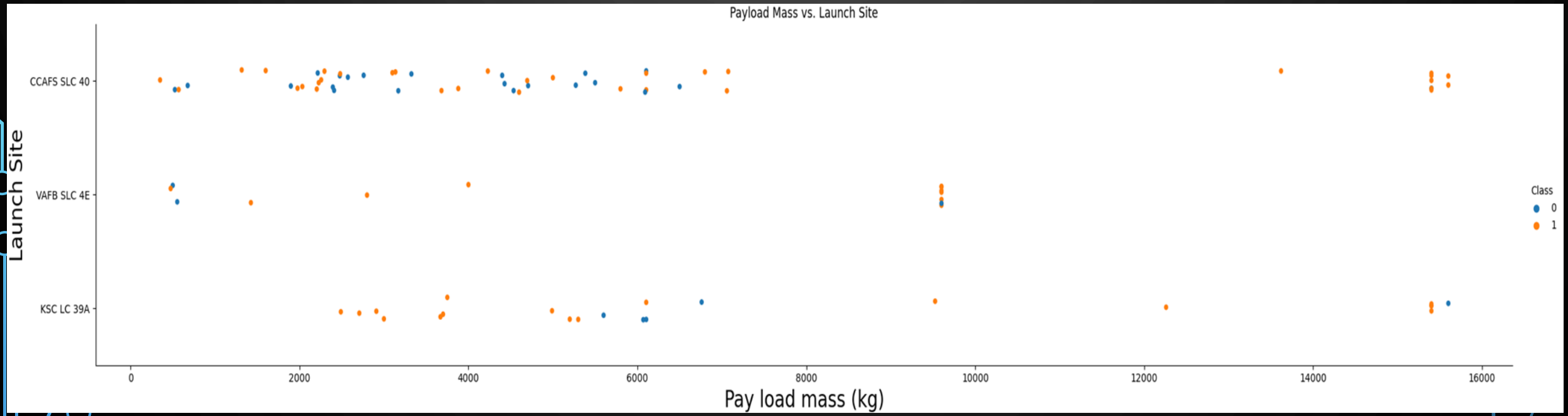
Flight Number vs. Launch Site

- Is there a relationship between Flight Number/Launch Site and landing outcome?
  - Most flights are coming from CCAFS SLC 40
  - VAFB SLC 4E and KSC LC 39A have relatively few landing failures



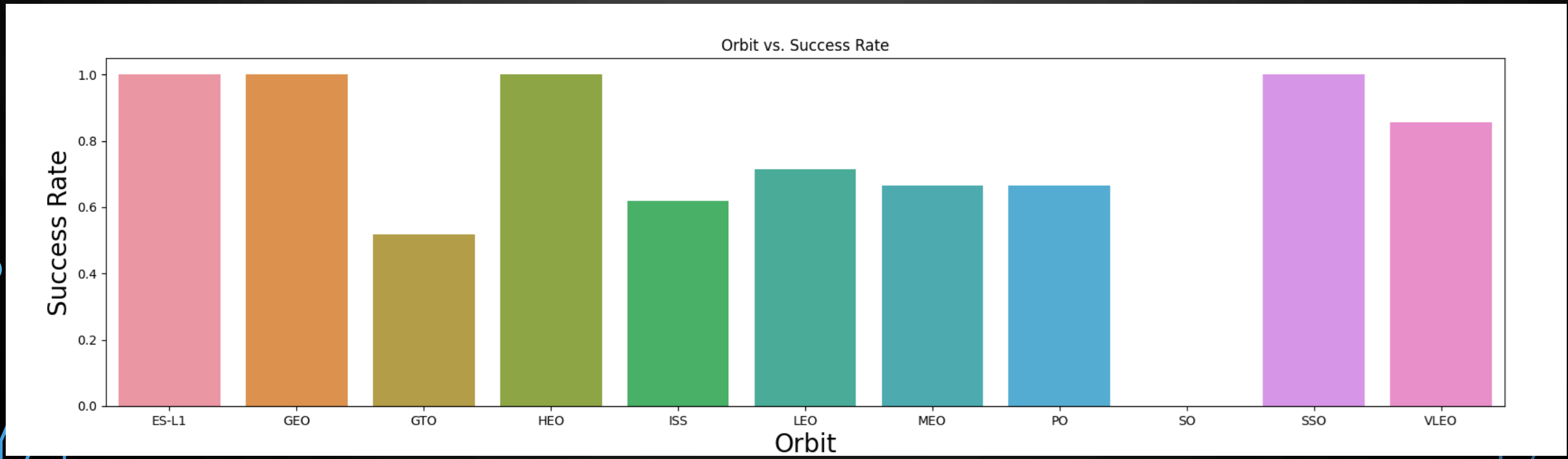
Payload vs. Launch Site

- Is there a relationship between Payload and Launch Site?
  - VAFB SLC 4E has no rocket launches with payloads above 10,000kg



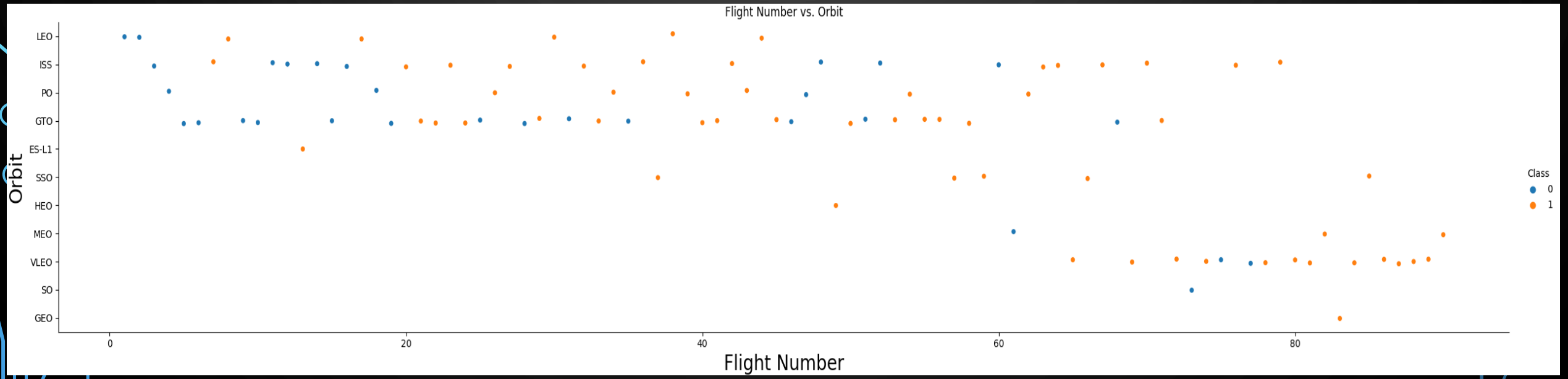
Success Rate vs. Orbit Type

- Is there a relationship between Orbit and Landing Outcome?
  - ES-L1, GEO HEO, SSO, VLEO have the highest success rate



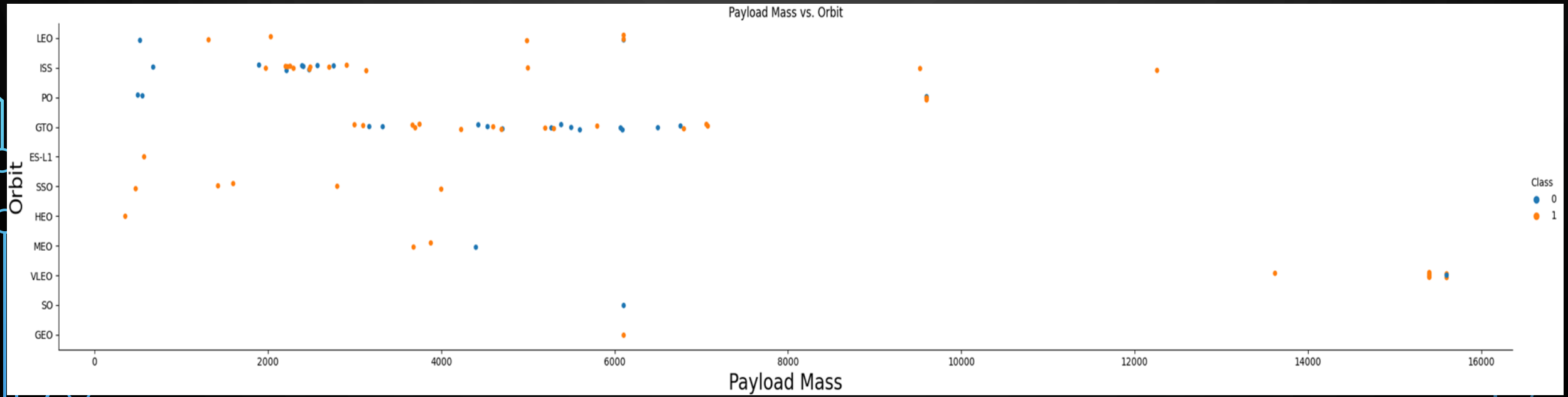
Flight Number vs. Orbit Type

- Is there a relationship between Flight Number/Orbit and Landing Outcome?
  - LEO orbit success rate increases as flight number increases
  - SSO, MEO & MEO orbits have very few total flights



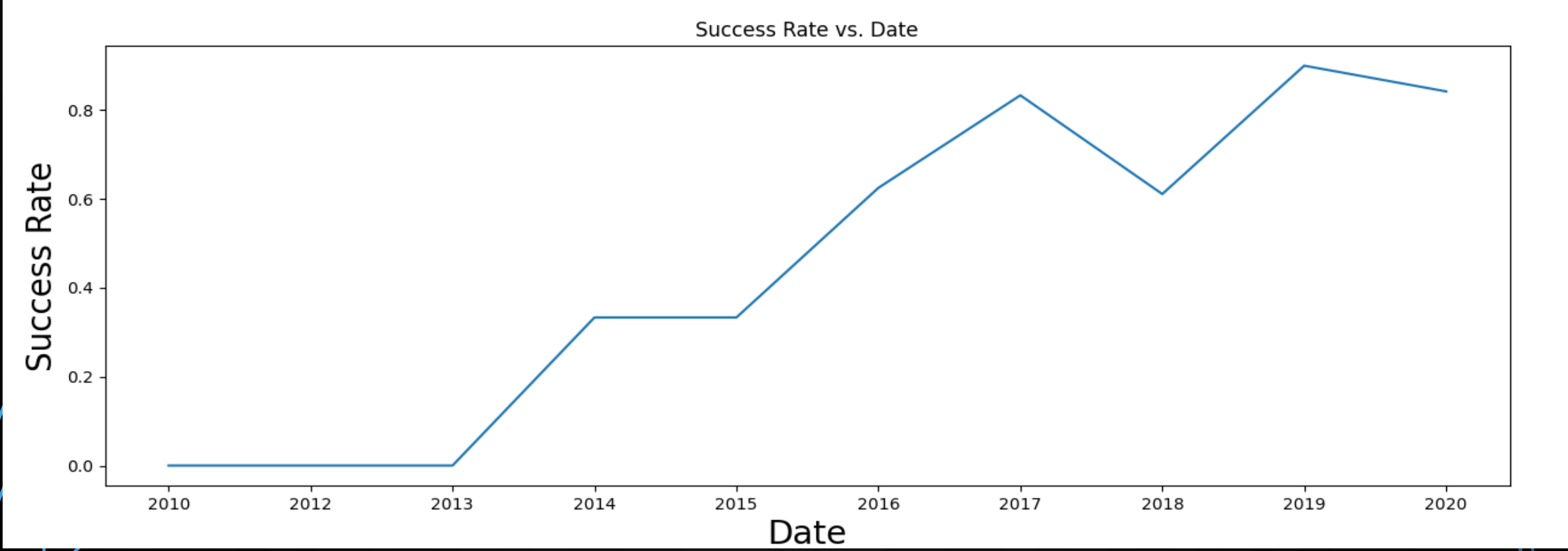
Payload vs. Orbit Type

- Is there a relationship between Payload/Orbit and Landing Outcome?
  - For heavy payloads Polar, LEO and ISS orbits have higher success rater
  - SSO has no payloads higher than 5,000kg and only successful landing outcomes



Launch Success Yearly Trend

- Is there a trend in Success Rate over time?
  - Success rate increased from 2013 to 2020 with a slight dip in 2018



All Launch Site Names

- Find the names of the unique launch sites
  - 4 unique launch sites found in the dataset using distinct function

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
  - 5 records shown for Launch Site CCAFS LC-40 using statement like 'CCA' and Limit=5.
  - 2 Failures and 3 No attempts found

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
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

- Calculate the total payload carried by boosters from NASA
  - Payload mass calculated by filtering data for NASA as a customer and using sum function to add the result displayed in kg.

TOTAL_PAYLOAD_MASS
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
  - Average payload mass calculated by filtering data for booster version F9 v1.1 with like statement and applying mean function

AVERAGE_PAYLOAD_MASS
2534.6666666666665

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
  - Date of first successful landing outcome found by applying min function to date and limit 1

Date	Landing _Outcome
22-12-2015	Success (ground pad)

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 versions found by filtering data by landing outcome and payload mass using statement between 4000 and 6000

Booster_Version	PAYLOAD_MASS__KG_	Landing _Outcome
F9 FT B1022	4696	Success (drone ship)
F9 FT B1026	4600	Success (drone ship)
F9 FT B1021.2	5300	Success (drone ship)
F9 FT B1031.2	5200	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
  - Total success and failure calculated using sum function on case when landing outcome is like ‘Success’ and like ‘Failure’ respectively

TOTAL_SUCCESS	TOTAL_FAILURE
61	10

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
  - Booster versions found by using distinct function and where statement filtering for max payload using a subquery and max function.

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 year 2015
  - Records found by using substr and filtering data for year 2015 and landing outcome as ‘Failure (drone ship)’.

MONTH	Booster_Version	Launch_Site	Landing _Outcome
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

- 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
  - Ranked by using count function and filtering data using selected dates with between statement, group by landing outcome and order by landing outcome using desc for descending

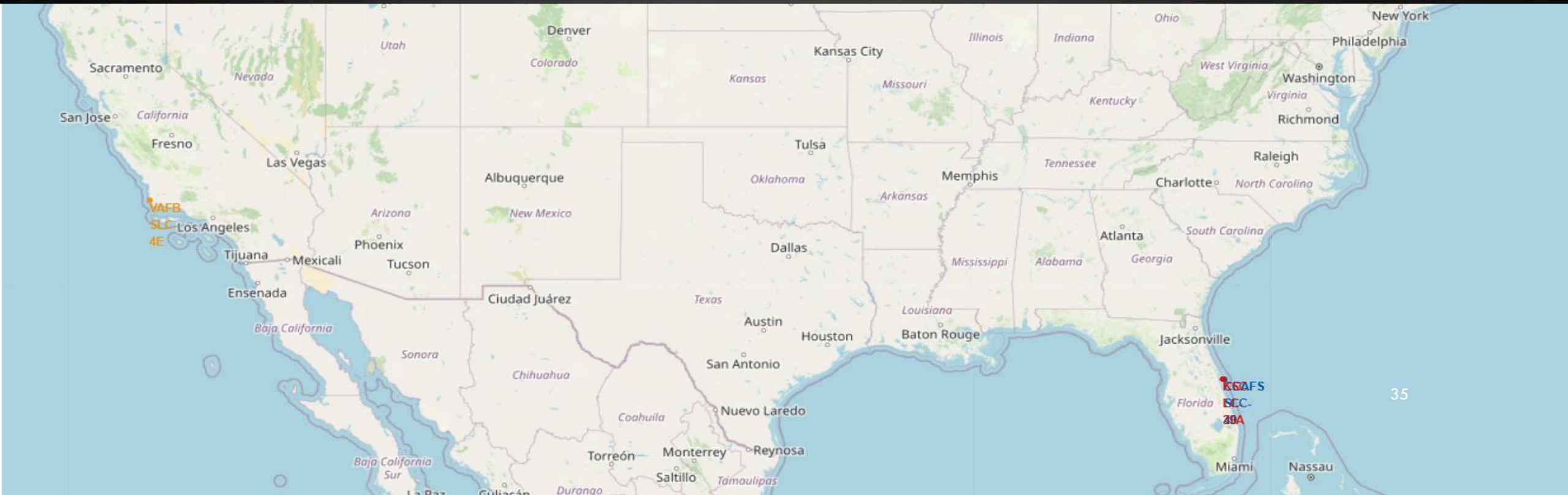
FREQUENCY	Landing _Outcome
20	Success
10	No attempt
8	Success (drone ship)
6	Success (ground pad)
4	Failure (drone ship)
3	Failure
3	Controlled (ocean)
2	Failure (parachute)
1	No attempt

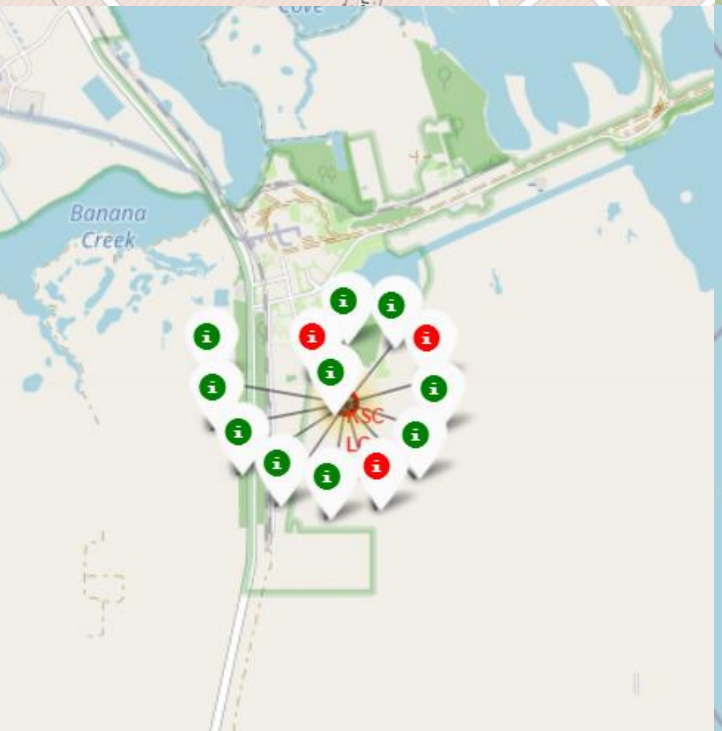
Section 3

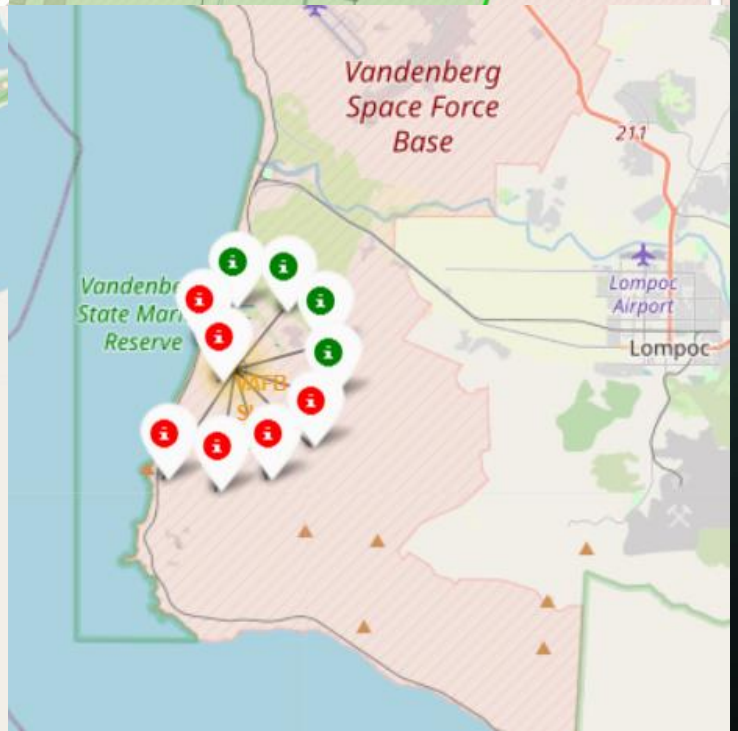
Launch Sites Proximities Analysis

FOLIUM MAP: ALL LAUNCH SITES

- Four launch sites all located in the US.
- 3 launch sites in Florida and 1 in California
- VAFB SLC-4E furthest from the equator

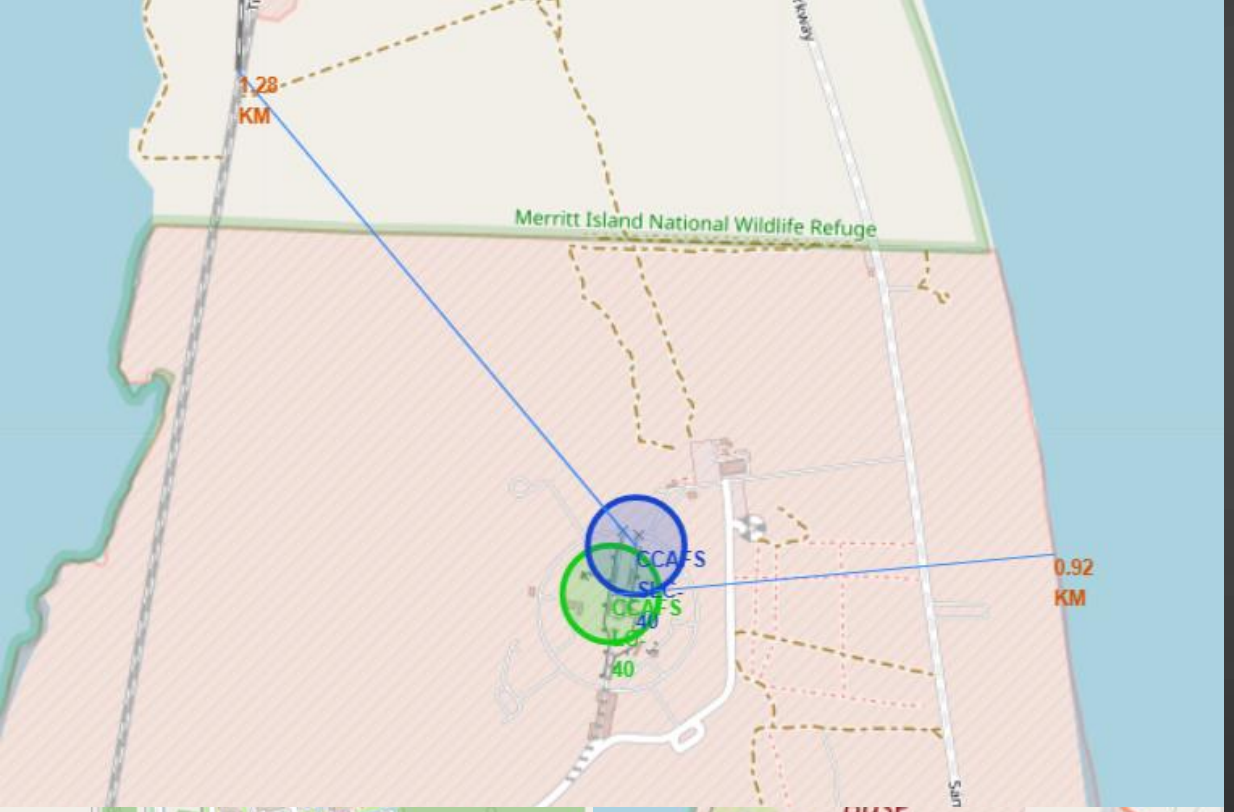


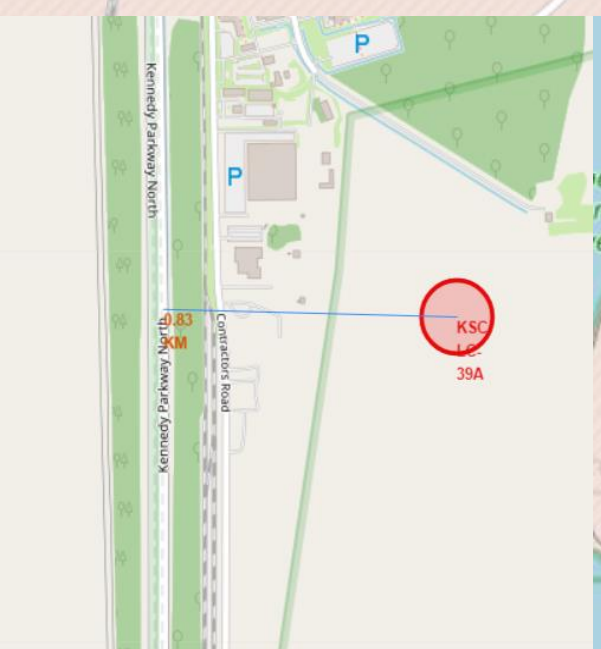




FOLIUM MAP: OUTCOMES

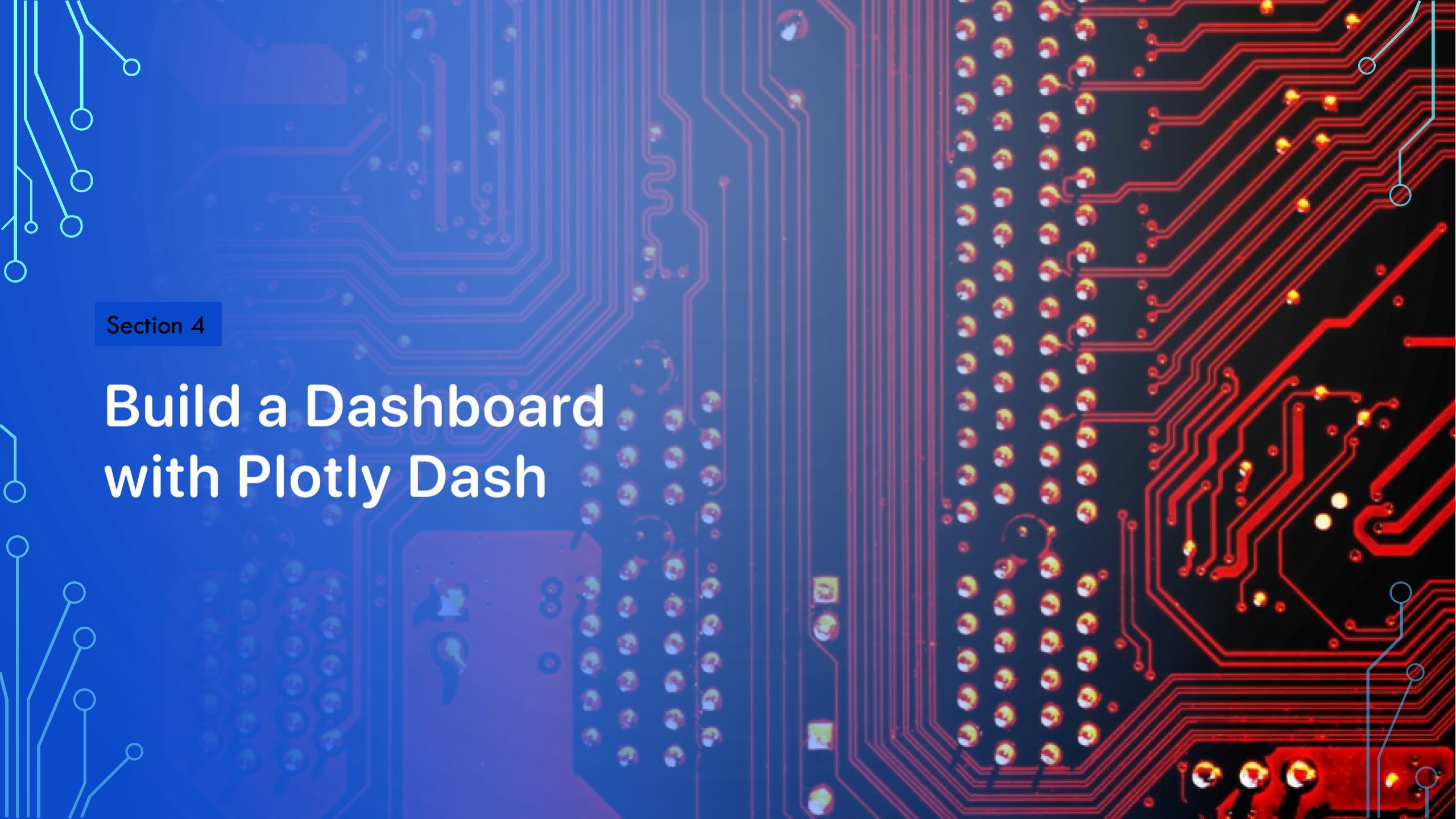
- Landing outcomes for CCAFS LC-40 (Top Left), CCAFS SLC-40 (Top Right), KSC LC-39A (Bottom Left), VAFB SLC-4E (Bottom Right)
- Successful landings visualized as green markers and failures as red markers





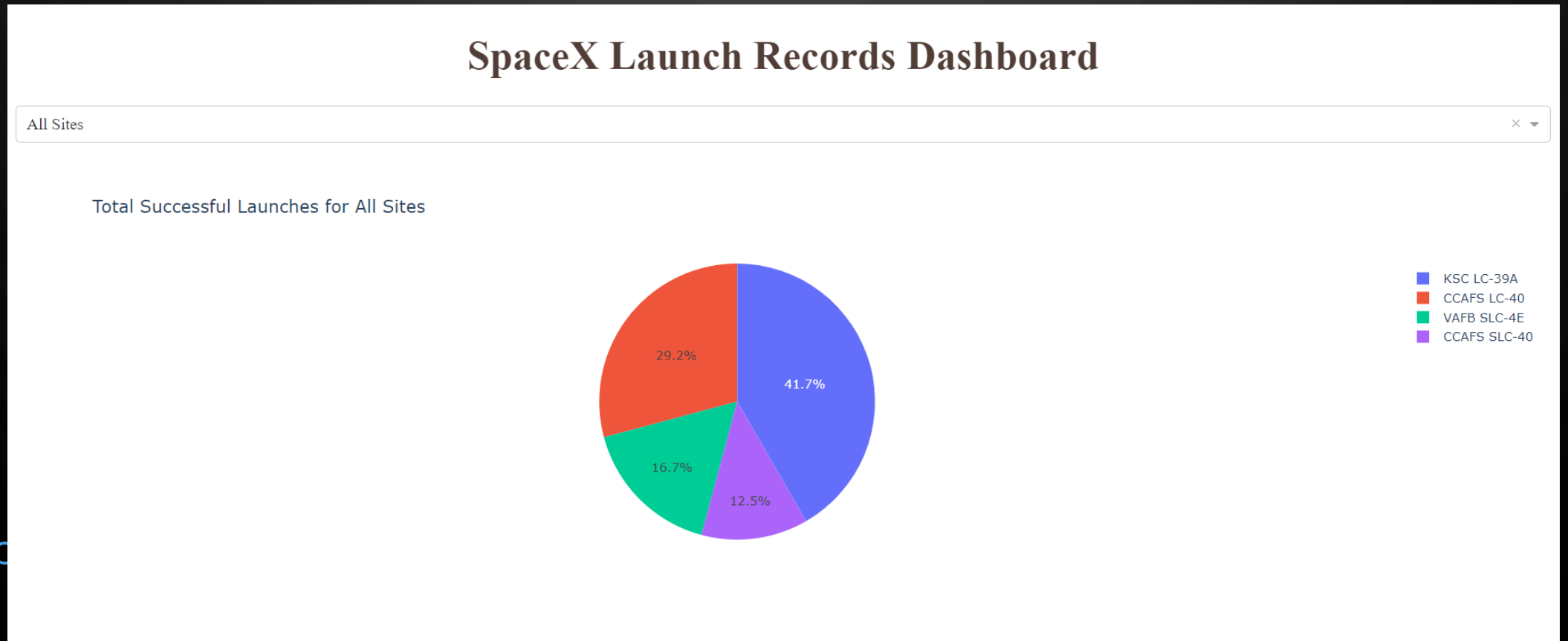
FOLIUM MAP: PROXIMITY TO KEY SITES

- Distance from CCAFS LC-40 to coastline and CCAFS SLC-40 to nearest railroad(Top), from KSC LC-39A to nearest highway(Bottom Left), and from VAFB SLC-4E to nearest city(Bottom Right)
- All sites are within 1 km of a coastline
- All sites are at least 10 km from nearest city


Section 4
Build a Dashboard with Plotly Dash

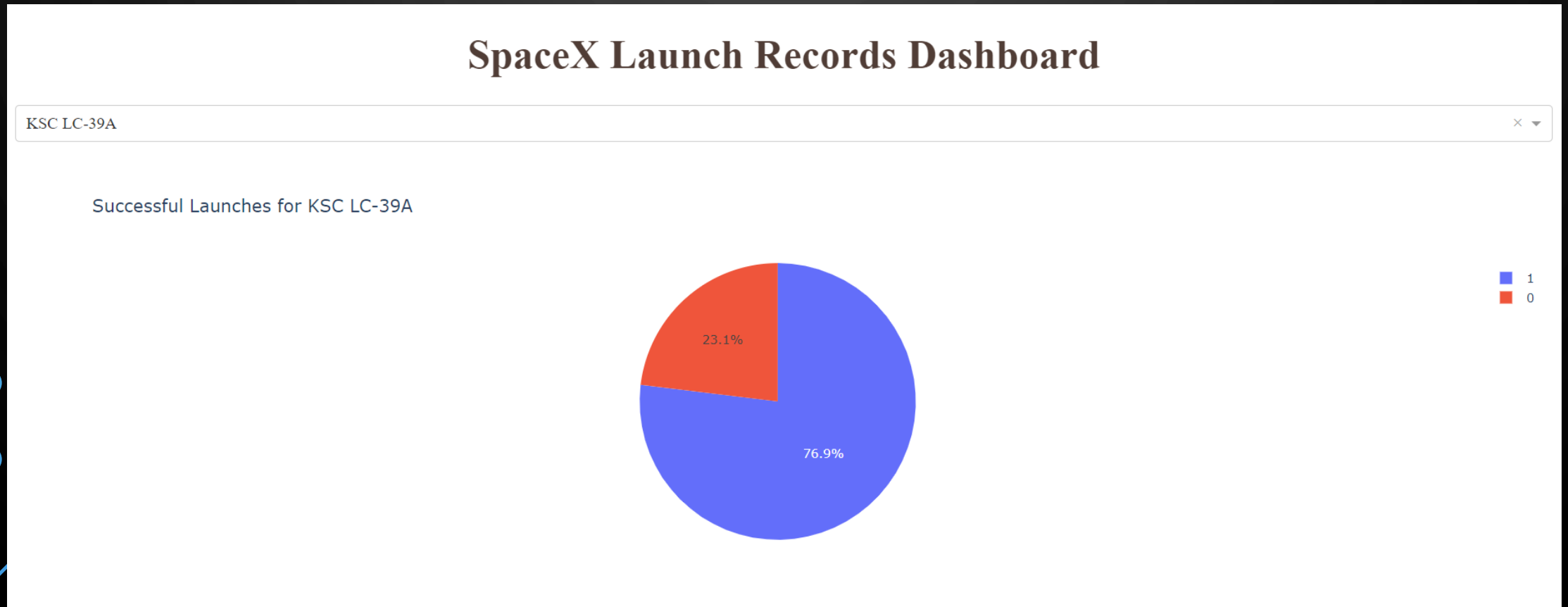
DASHBOARD: SUCCESS PIE CHART (ALL SITES)

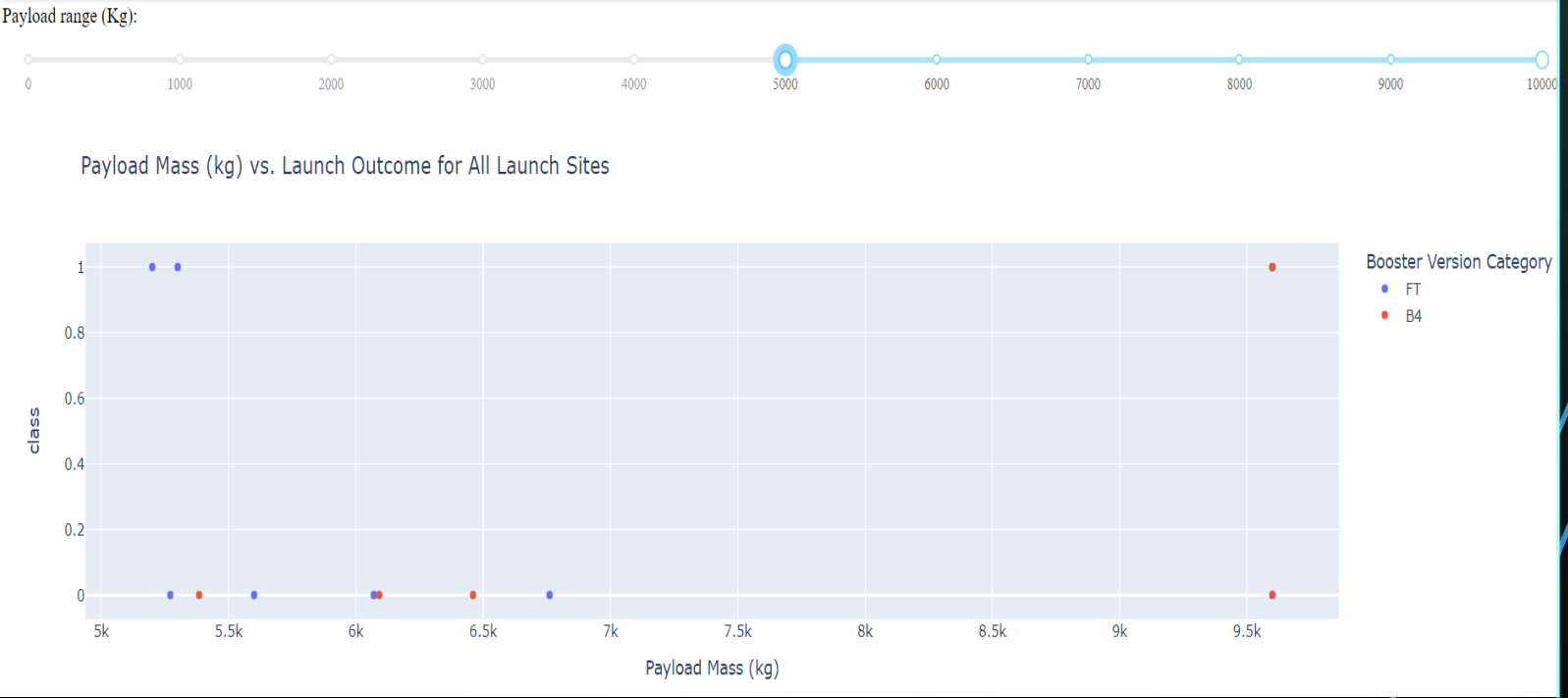
- KSC LC-39A accounts for the most successes
- CCAFS SLC-40 accounts for the least successes



DASHBOARD: SUCCESS PIE CHART (KSC LC-39A)

- Pie chart for site with highest success rate is shown
- KSC LC-39A has a success rate of 76.9%





DASHBOARD: PAYLOAD VS. OUTCOME SCATTER PLOT

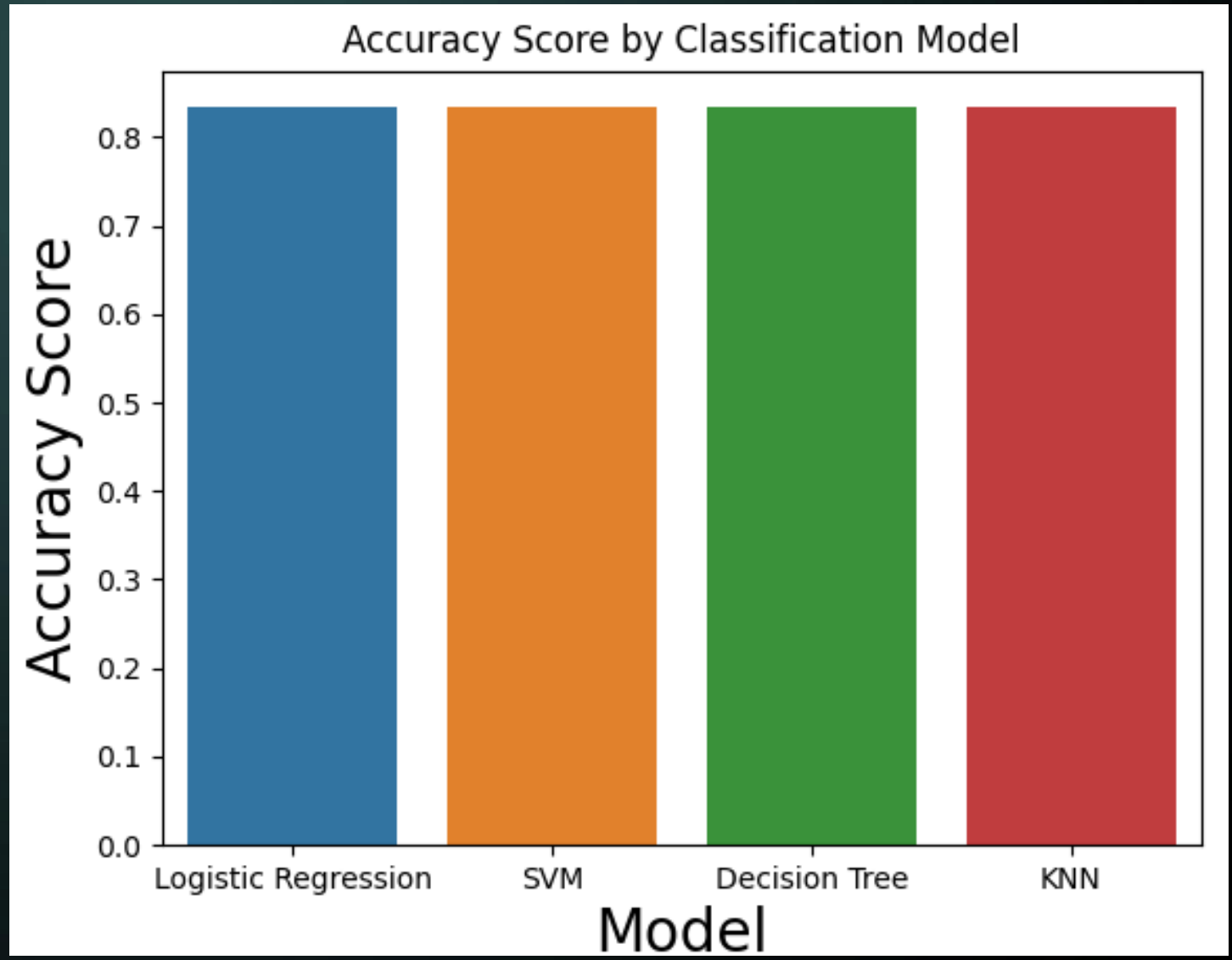
- Top chart shows outcome for payloads 0-5000kg and bottom chart shows outcomes for payloads 5000-10000kg.
- Only two boosters were used for heavy payloads: FT and B4.

Section 5

Predictive Analysis (Classification)

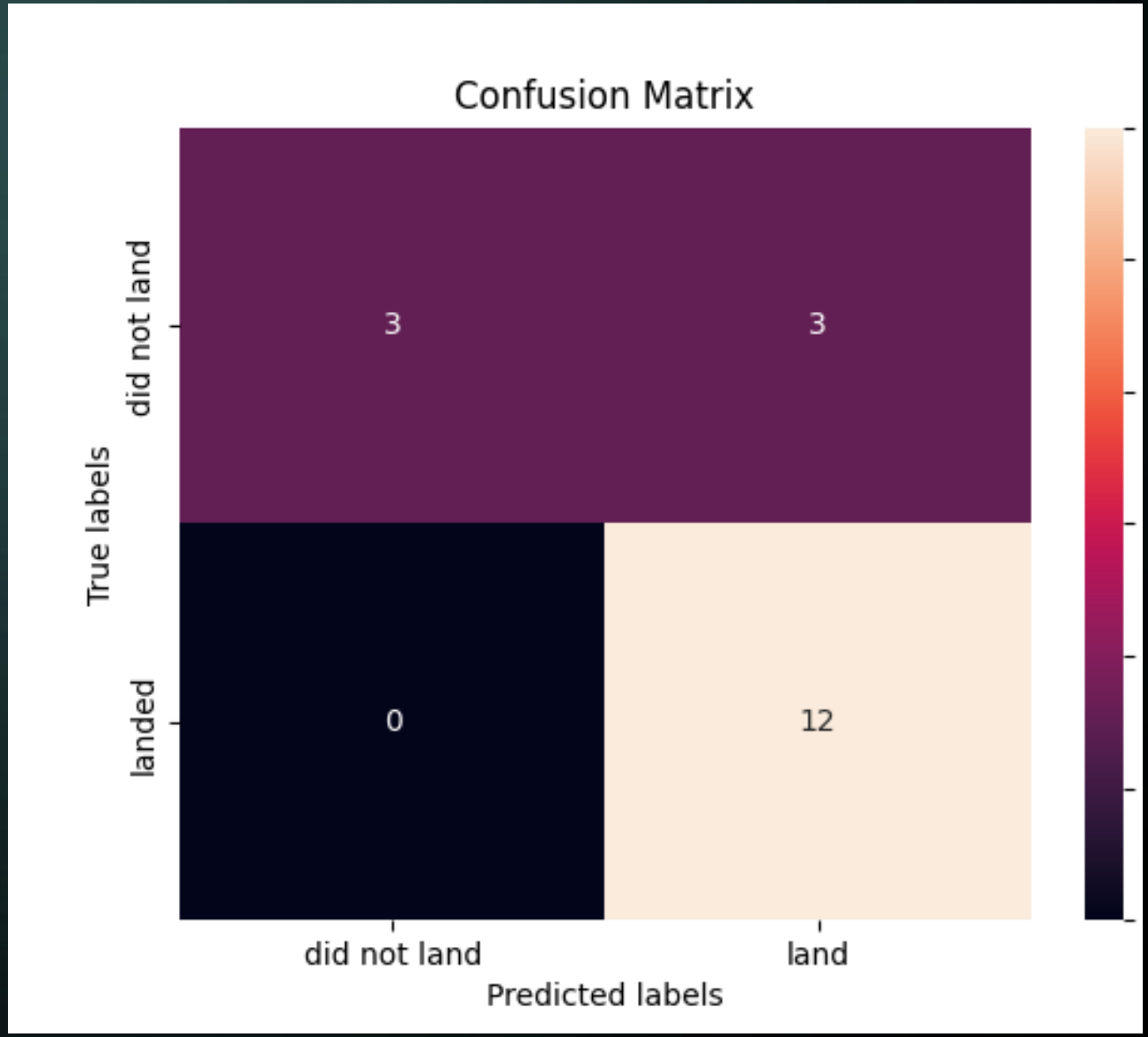
CLASSIFICATION ACCURACY

- All models performed equally well.
- Predicting landing outcomes with an accuracy score of 0.8334



CONFUSION MATRIX

- All four models had equal accuracy scores and the same confusion matrix
- 18 outcomes predicted and the models had 3 false positives
- Models predicted 3 flights to land (Top Right Square) when the actual rocket failed to land.



CONCLUSIONS

- Landing outcome of the first stage rocket can be predicted with an accuracy of 83.34%
- Variables influencing landing outcome
  - Launch Number – More launches yield more experience and better success rate. SpaceY should not become discouraged if initial success rate is low. Consistency and rapid iterations will prove key to SpaceY's success.
  - Launch Site – Geography plays a role. SpaceY should select its launch site carefully, as close to the equator as possible. Within 1km of a coastline.
  - Payload – Heavier payloads have lower success rates. SpaceY should focus on missions with low payloads (>5,000kg).
  - Orbit – Different orbits have markedly different success rates. SpaceY should focus on missions to orbits with high success rates: ES-L1, GEO HEO, SSO, VLEO.
- Considering these variables will allow SpaceY to win bids and stay competitive with SpaceX in the space race

APPENDIX

CODE FOR INTERACTIVE PIE CHART

```
# Add a callback function for `site-dropdown` as input, `success-pie-chart` as output
@app.callback(Output(component_id='success-pie-chart', component_property='figure'),
              Input(component_id='site-dropdown', component_property='value')
              )
def get_pie_chart(entered_site):
    filtered_df = spacex_df
    if entered_site == 'ALL':
        fig = px.pie(spacex_df,
                     values='class',
                     names='Launch Site',
                     title='Total Successful Launches for All Sites'
                     )
        return fig
    else:
        filtered_df = spacex_df[spacex_df['Launch Site']==entered_site]
        entered_site_df = filtered_df.groupby('class', as_index=False)['Flight Number'].count()
        fig = px.pie(entered_site_df,
                     values='Flight Number',
                     names='class',
                     title='Successful Launches for {}' .format(entered_site)
                     )
        return fig
        # return the outcomes piechart for a selected site
```

CODE FOR INTERACTIVE SCATTER PLOT

```
# Add a callback function for `site-dropdown` and `payload-slider` as inputs, `success-payload-scatter-chart` as output
@app.callback(Output(component_id='success-payload-scatter-chart', component_property='figure'),
                [Input(component_id='site-dropdown', component_property='value'),
                Input(component_id='payload-slider', component_property='value')
                ]
              )
def get_scatter_chart(entered_site, payload_range):
    filtered_df = spacex_df
    if entered_site == 'ALL':
        filtered_df = spacex_df[(spacex_df['Payload Mass (kg)'] > int(payload_range[0])) & (spacex_df['Payload Mass (kg)'] < int(payload_range[1]))]
        fig = px.scatter(filtered_df,
                        x='Payload Mass (kg)',
                        y='class',
                        color='Booster Version Category',
                        title='Payload Mass (kg) vs. Launch Outcome for All Launch Sites'
                        )
        return fig
    else:
        filtered_df = spacex_df[(spacex_df['Payload Mass (kg)'] > int(payload_range[0])) & (spacex_df['Payload Mass (kg)'] < int(payload_range[1]))]
        entered_site_df = filtered_df[filtered_df['Launch Site']==entered_site]
        fig = px.scatter(entered_site_df,
                        x='Payload Mass (kg)',
                        y='class',
                        color='Booster Version Category',
                        title='Payload Mass (kg) vs. Launch Outcome for {}' .format(entered_site)
                        )
        return fig
```

APPENDIX

- GitHub Link to main repository: Ketzaal/IBM_Applied_Data_Science_Capstone: All Notebooks and python files for the IBM Applied Data Science Capstone Course (github.com)

Thank you!