Winning Space Race with Data Science

Felix Marinus Weißenfeld
13.09.2023

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

Summary of Methodologies

- The research attempts to identify the factors for a successful rocket landing. To make this determination, the following methodologies where used:
- **Collect** data using SpaceX REST API and web scraping techniques
- **Wrangle** data to create success/fail outcome variable
- **Explore** data with data visualization techniques, considering the following factors: payload, launch site, flight number and yearly trend
- **Analyze** the data with SQL, calculating the following statistics: total payload, payload range for successful launches, and total # of successful and failed outcomes
- **Explore** launch site success rates and proximity to geographical markers
- **Visualize** the launch sites with the most success and successful payload ranges
- **Build Models** to predict landing outcomes using logistic regression, support vector machine (SVM), decision tree and K-nearest neighbor (KNN)

Exploratory Data Analysis:
Launch success has improved over time
KSC LC-39A has the highest success rate among landing sites
Orbits ES-L1, GEO, HEO, and SSO have a 100% success rate

Visualization/Analytics:
Most launch sites are near the equator, and all are close to the coast

Predictive Analytics:
All models performed similarly on the test set. The decision tree model

Introduction

Background

- SpaceX, a leader in the space industry, strives to make space travel affordable for everyone. Its accomplishments include sending spacecraft to the international space station, launching a satellite constellation that provides internet access and sending manned missions to space. SpaceX can do this because the rocket launches are relatively inexpensive ($62 million per launch) due to its novel reuse of the first stage of its Falcon 9 rocket. Other providers, which are not able to reuse the first stage, cost upwards of $165 million each. By determining if the first stage will land, we can determine the price of the launch. To do this, we can use public data and machine learning models to predict whether SpaceX – or a competing company – can reuse the first stage.

Explore

- How payload mass, launch site, number of flights, and orbits affect first-stage landing success
- Rate of successful landings over time
- Best predictive model for successful landing (binary classification)

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data Collection trough SpaceX API and web scraping from Wikipedia
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Predict landing outcomes

Data Collection – SpaceX API

Using the SpaceX API

1. Request data from SpaceX API
2. Decode response using .json() and converting the data into a pandas dataframe with .json_normalize()
3. Create a subset of our dataframe keeping only the neccessary features
4. Request information using custom functions
5. Construct the dataset (using the defined global variables) and then create a pandas dataframe
6. Filter the dataframe to include only Falcon 9 launches
7. Replace np.nan values in the data by the .mean() of PayloadMass
8. Export the data into a csv

https://github.com/marinusman/IBM_SpaceX_Capstone-Project_FMW/blob/27975748979f897596551f96e1fbcf2c5e39f7e6/1_jupyter-labs-spacex-data-collection-api_FMW.ipynb

Data Collection – Web Scraping

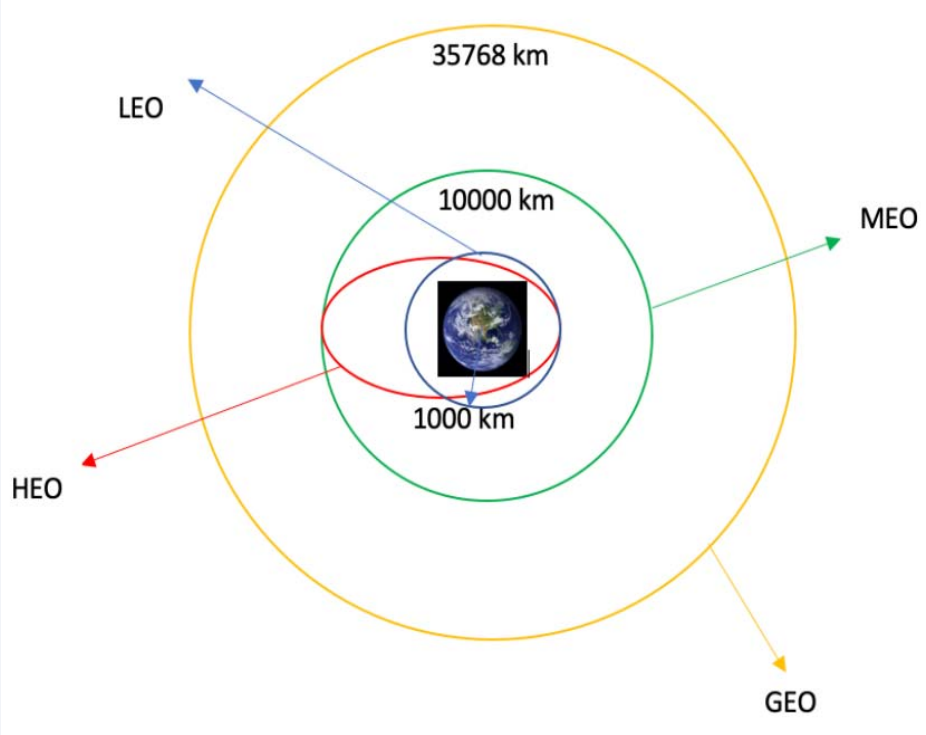
web scraping Wikipedia

1. **Request** Falcon 9 data from Wikipedia
2. Create a **BeautifulSoup object** from the HTML response
3. **Extract** column names from HTML table header
4. **Collect** data from parsing HTML tables
5. **Create** a dictionary and a dataframe
6. Export the data into a csv

https://github.com/marinusman/IBM_SpaceX_Capstone-Project_FMW/blob/27975748979f897596551f96e1fbcf2c5e39f7e6/2_jupyter-labs-webscraping_FMW.ipynb

Data Wrangling

1. **Perform exploratory data analysis** and determined the training labels.
2. **Calculate**
   a. the number of launches on each site
   b. determine the number and occurrence of each orbit
3. Create a **landing outcome** label from outcome column and
4. exported the results to csv.



https://github.com/marinusman/IBM_SpaceX_Capstone-Project_FMW/blob/27975748979f897596551f96e1fbcf2c5e39f7e6/3_labs-jupyter-spacex-Data%20wrangling_FMW.ipynb

EDA with Data Visualization

The data was visually explored by showing the relationship between

- Flight Number and Payload
- Flight Number and Launch Site
- Payload Mass and Launch Site
- Payload Mass and Orbit type

➢ **Scatter Plots** were used to show the relationships between the variables

➢ Comparisons among discrete categries were visualized using **bar charts**

httpgithub.com/marinusman/IBM_SpaceX_Capstone-Project_FMW/blob/27975748979f897596551f96e1fbcf2c5e39f7e6/5_jupyter-labs-eda-dataviz_FMW.ipynbs://

EDA with SQL

- We loaded the SpaceX dataset into a SQL database
- We applied EDA with SQL to get insight from the data.
- Used queries to find out for instance:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.

https://github.com/marinusman/IBM_SpaceX_Capstone-Project_FMW/blob/27975748979f897596551f96e1fbcf2c5e39f7e6/4_jupyter-labs-eda-sql-edx_sqllite_FMW.ipynb

Build an Interactive Map with Folium

Markers Indicating Launch Sites

- Added **blue circle** at **NASA Johnson Space Center's coordinate** with a **popup label** showing its name using its latitude and longitude coordinates
- Added **red circles** at **all launch sites coordinates** with a **popup label** showing its name using its name using its latitude and longitude coordinates

Map with Folium

- **Colored Markers of Launch Outcomes**
- Added **colored markers** of **successful**(**green**) and **unsuccessful**(**red**) **launches** at each launch site to show which launch sites have high success rates

Distances Between a Launch Site to Proximities

- Added **colored lines** to **show distance between** launch site **CCAFS SLC-40 and** its proximity to the **nearest coastline, railway, highway, and city**

https://github.com/marinusman/IBM_SpaceX_Capstone-Project_FMW/blob/27975748979f897596551f96e1fbcf2c5e39f7e6/6_lab_jupyter_launch_site_location.jupyterlite_FMW.ipynb

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.

Dropdown List with Launch Sites

- Allow user to select all launch sites or a certain launch site

Dashboard with Plotly Dash

- **Slider of Payload Mass Range**
- Allow user to select payload mass range

Pie Chart Showing Successful Launches

- Allow user to see successful and unsuccessful launches as a percent of the total

Scatter Chart Showing Payload Mass vs. Success Rate by Booster Version

- Allow user to see the correlation between Payload and Launch Success

https://github.com/marinusman/IBM_SpaceX_Capstone-Project_FMW/blob/27975748979f897596551f96e1fbcf2c5e39f7e6/7_spacex_dash_app_FMW.py

Predictive Analysis (Classification)

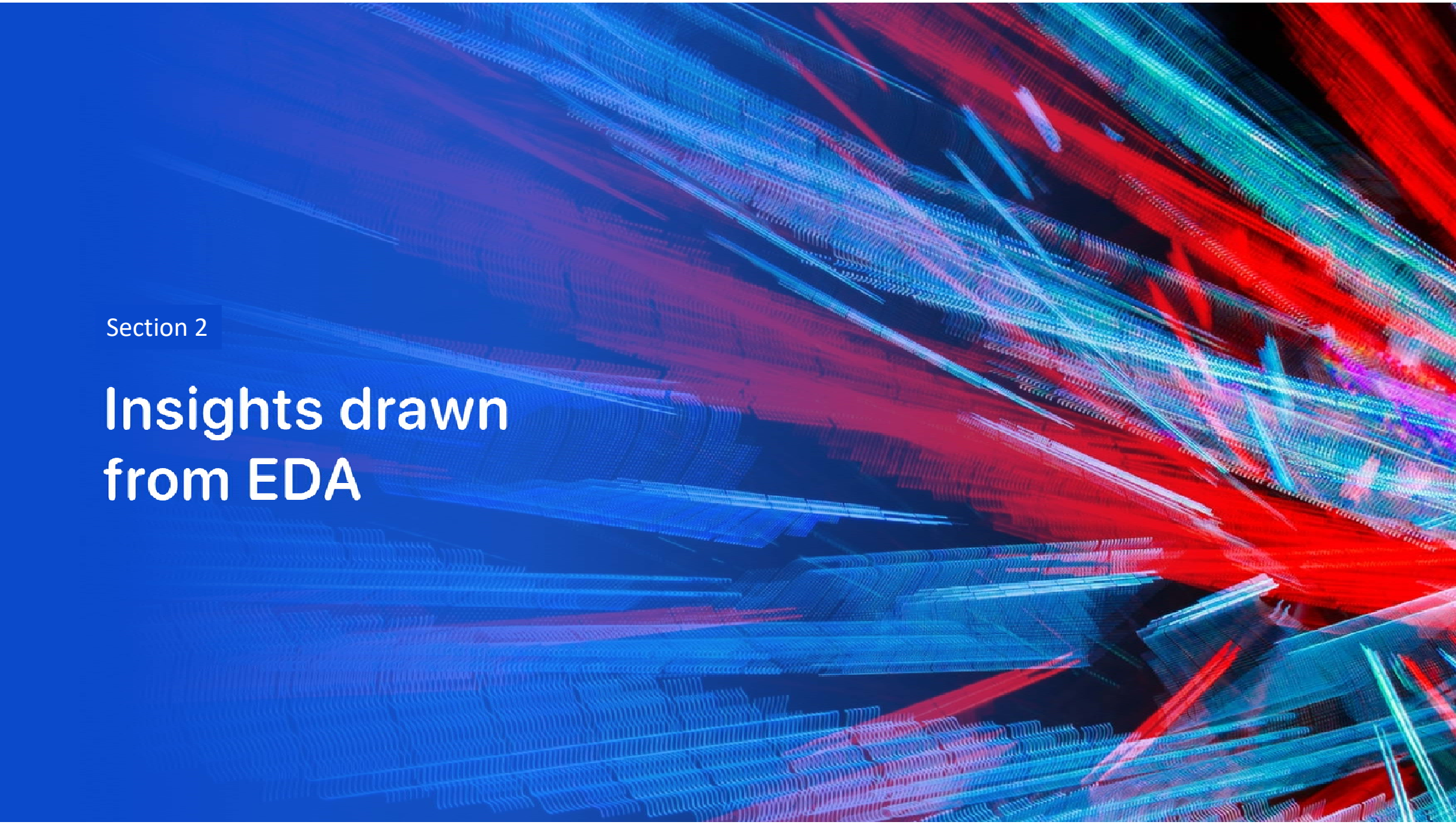
- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.

1. **Create** NumPy array from the Class column
2. **Standardize** the data with StandardScaler. Fit and transform the data.
3. **Split** the data using train_test_split
4. **Create** a GridSearchCV object with cv=10 for parameter optimization
5. **Apply** GridSearchCV on different algorithms: logistic regression (LogisticRegression()), support vector machine (SVC()), decision tree (DecisionTreeClassifier()), K-Nearest Neighbor (KNeighborsClassifier())
6. **Calculate** accuracy on the test data using .score() for all models
7. **Assess** the confusion matrix for all models
8. **Identify** the best model using Jaccard_Score, F1_Score and Accuracy

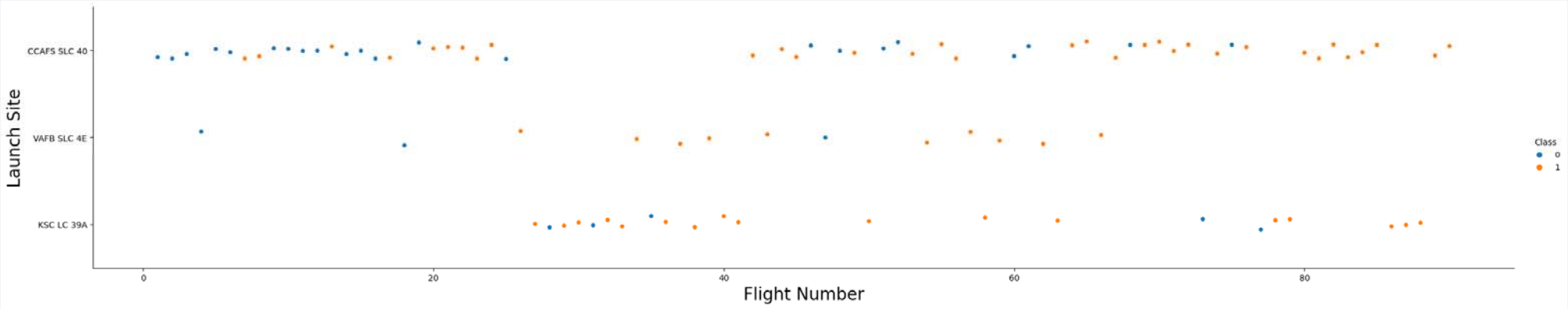
https://github.com/marinusman/IBM_SpaceX_Capstone-Project_FMW/blob/27975748979f897596551f96e1fbcf2c5e39f7e6/8_SpaceX_Machine_Learning_Prediction_Part_5_FMW.jupyterlite.ipynb

Results

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

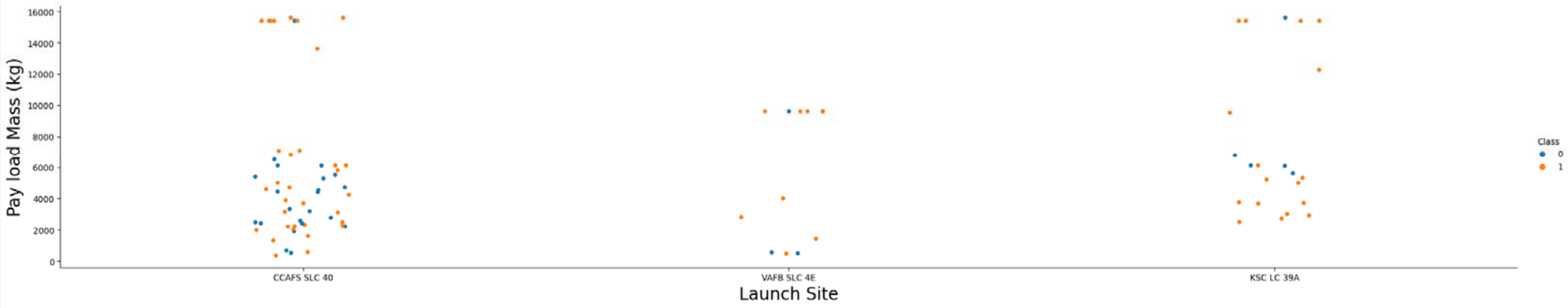

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



- **Earlier flights** had a **lower success rate** (**blue = fail**)
- **Later flights** had a **higher success rate** (**orange = success**)
- Around half of launches were from CCAFS SLC 40 launch site
- VAFB SLC 4E and KSC LC 39A have higher success rates
- We can infer that new launches have a higher success rate

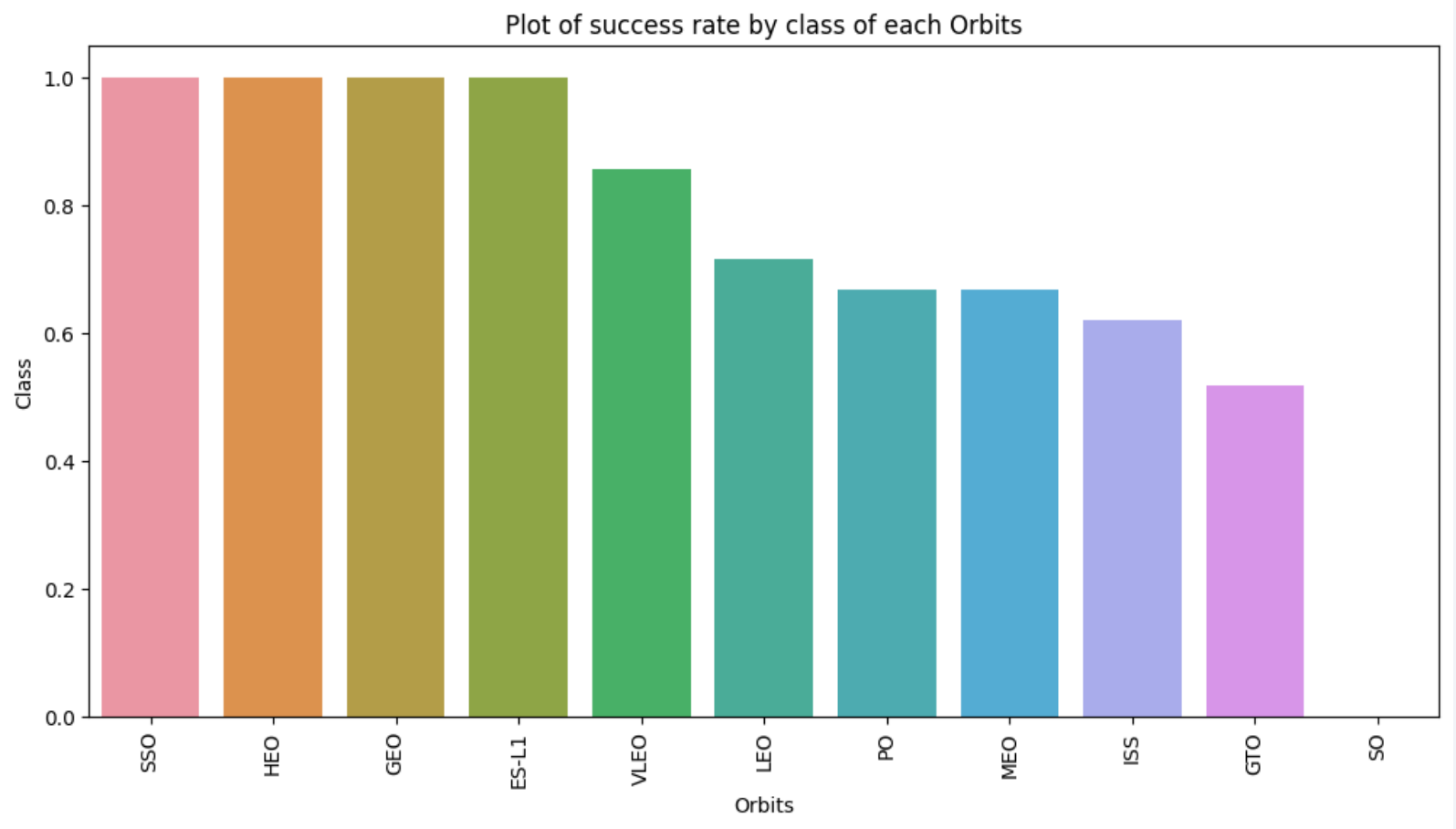
Payload vs. Launch Site



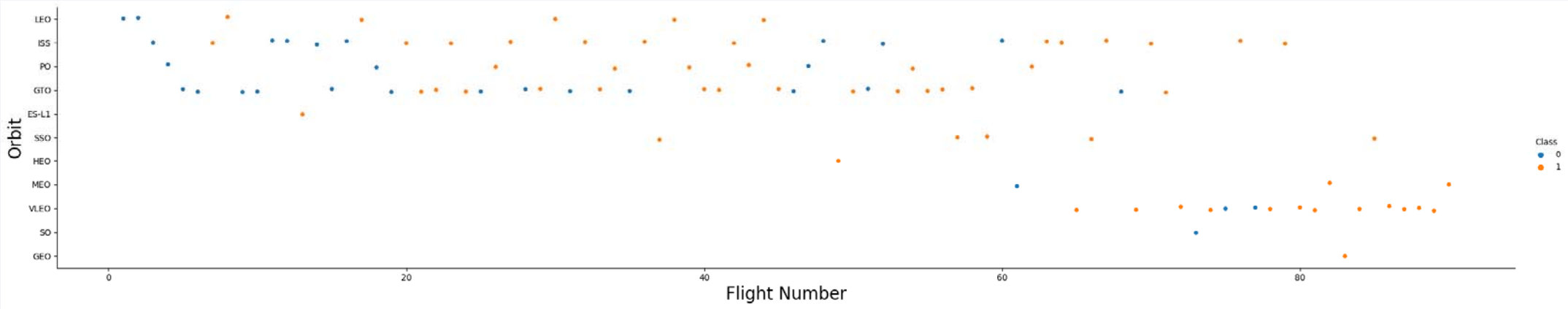
- Typically, the **higher** the **payload mass** (kg), the **higher** the **success rate**
- Most launces with a payload greater than 7,000 kg were successful
- KSC LC 39A has a 100% success rate for launches less than 5,500 kg
- VAFB SKC 4E has not launched anything greater than ~10,000 kg

Success Rate vs. Orbit Type

- SSO, HEO, GEO, ES-L1, has a **success rate of 100%**

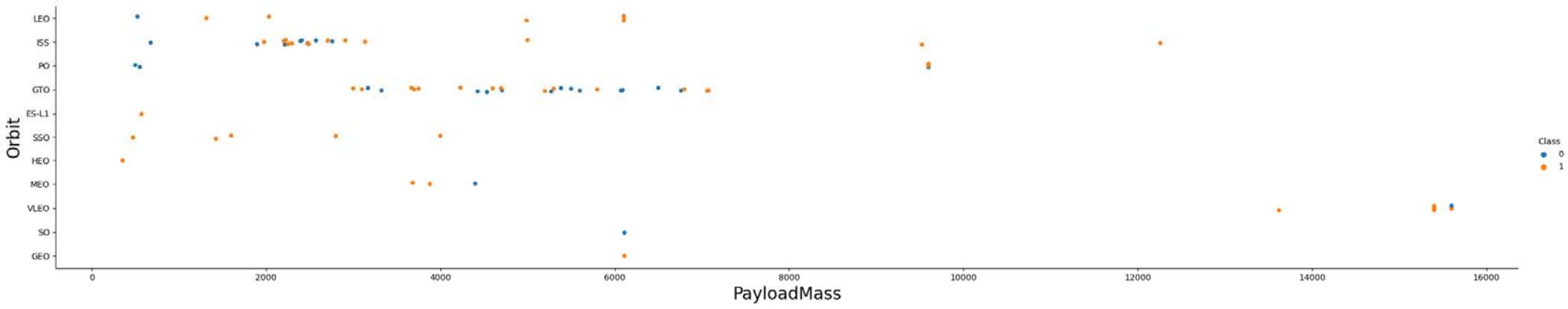


Flight Number vs. Orbit Type



- The success rate typically increases with the number of flights for each orbit
- This relationship is highly apparent for the LEO orbit
- The GTO orbit, however, does not follow this trend

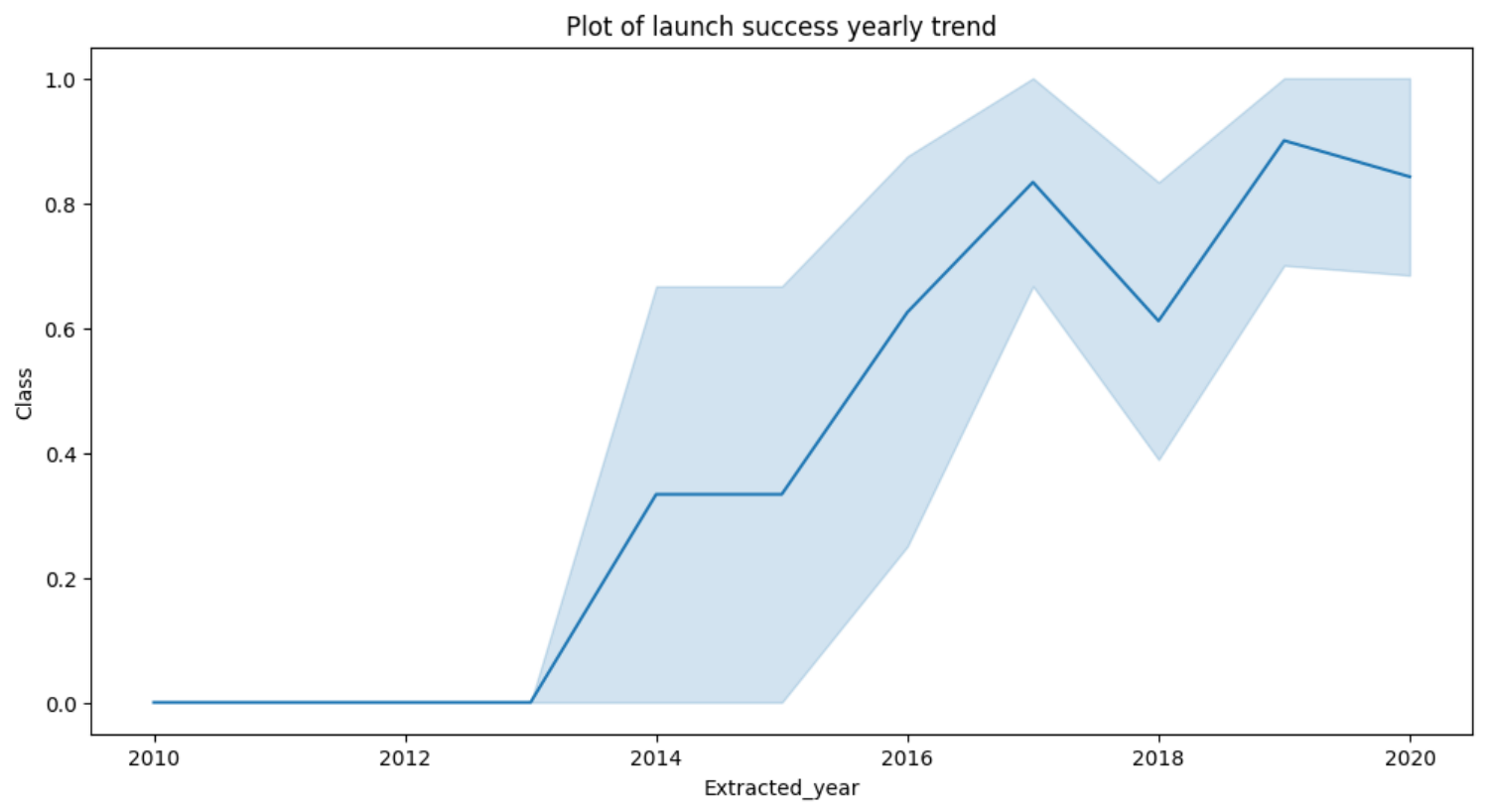
Payload vs. Orbit Type



- Heavy payloads are better with LEO, ISS and PO orbits
- The GTO orbit has mixed success with heavier payloads

Launch Success Yearly Trend

- The success rate improved from 2013-2017 and 2018-2019
- The success rate decreased from 2017-2018 and from 2019



All Launch Site Names

%sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL;

Launch_Site

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

Launch Site Names Begin with 'KSC'

- %sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE'KSC%' LIMIT 5

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-03-16	06:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-01-05	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
2017-05-15	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

Total Payload Mass

- %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = 'NASA (CRS)';

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1';

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- %sql SELECT MIN(DATE) FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (ground pad)';

MIN(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- %sql SELECT PAYLOAD FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;

Payload
JCSAT-14
JCSAT-16
SES-10
SES-11 / EchoStar 105

Total Number of Successful and Failure Mission Outcomes

- %sql SELECT MISSION_OUTCOME, COUNT(*) as total_number FROM SPACEXTBL GROUP BY MISSION_OUTCOME;

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

Boosters Carried Maximum Payload

- %sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);

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

- %sql SELECT Booster_Version, Launch_Site, Landing_Outcome FROM SPACEXTBL WHERE Landing_Outcome LIKE 'Failure (drone ship)' AND Date BETWEEN '2015-01-01' AND '2015-12-31'

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

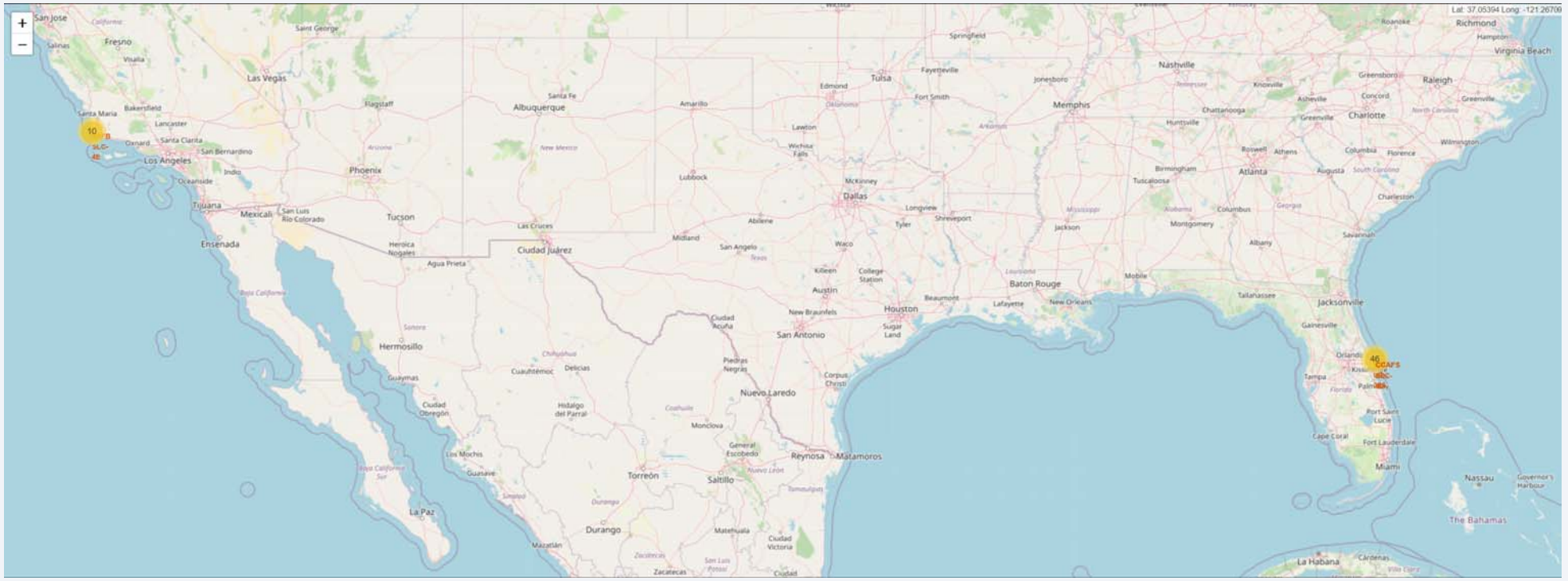
Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- %sql SELECT Landing_Outcome, COUNT(Landing_Outcome) FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY COUNT(Landing_Outcome) DESC

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


Section 3
Launch Sites
Proximities Analysis

Launch Sites are located near the equator



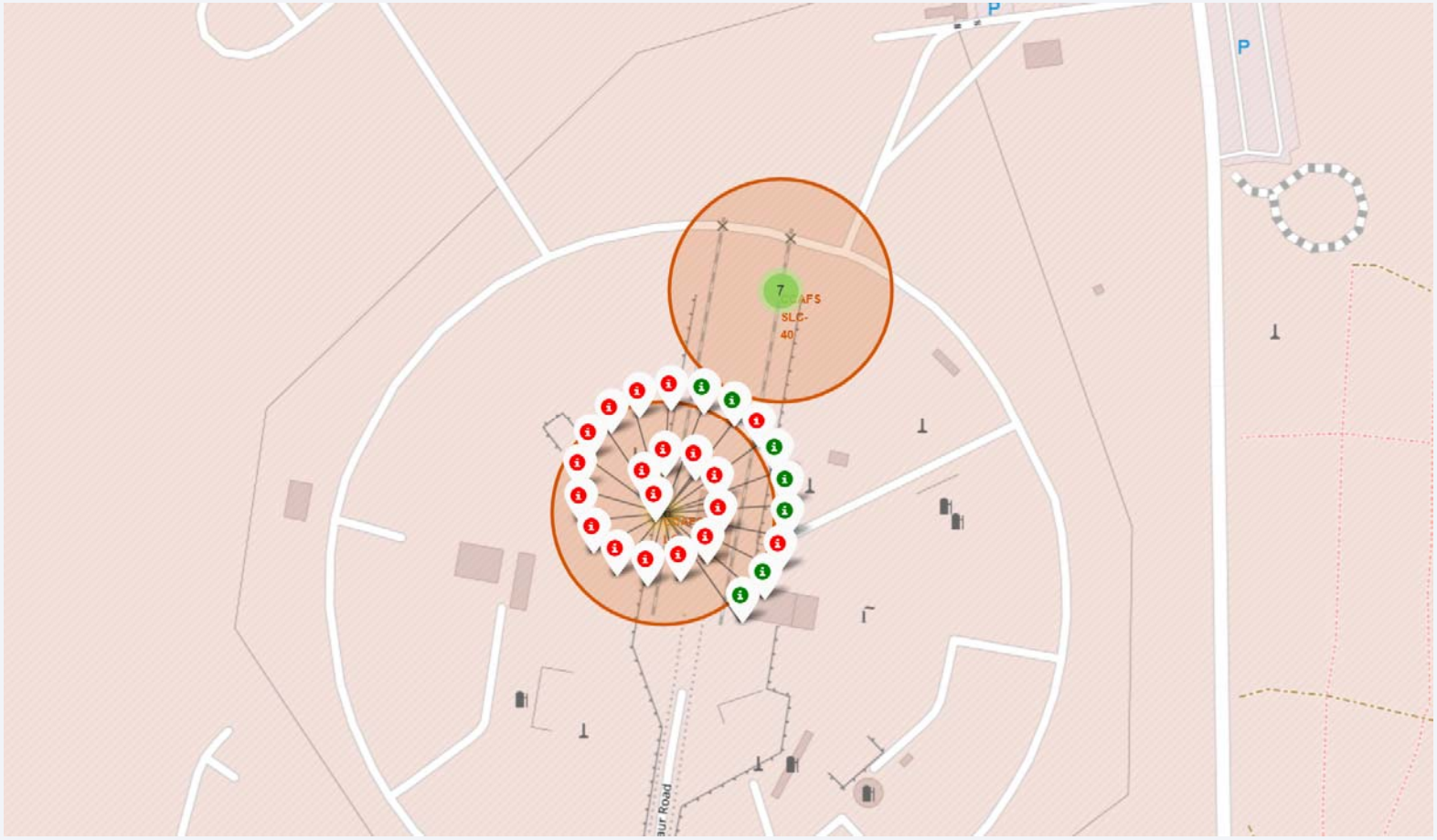
- **Near Equator**: the closer the launch site to the equator, the **easier**it is **to launch**to equatorial orbit, and the more help you get from Earth's rotation for a prograde orbit. Rockets launched from sites near the equator get an **additional natural boost**-due to the rotational speed of earth

Launch outcomes at CCAFS SLC-40

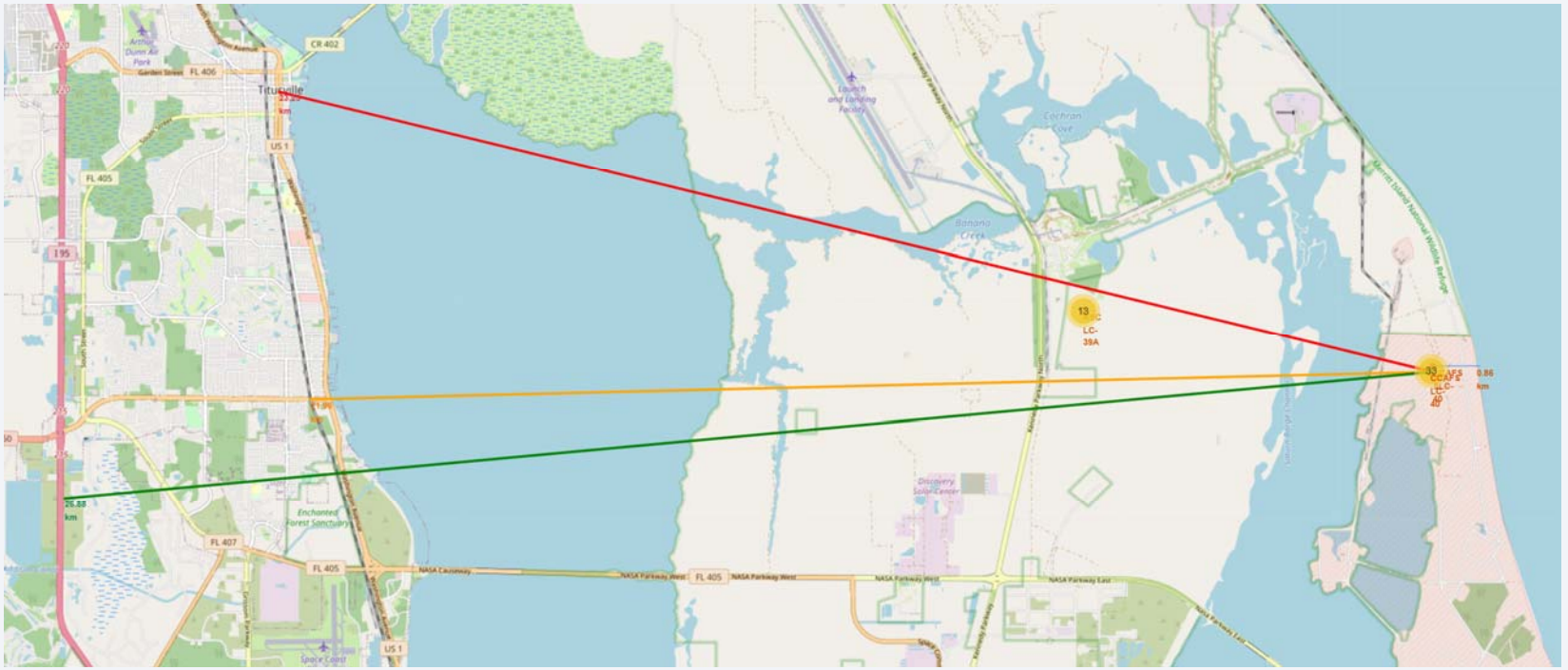
- **Green** markers for successful launches
- **Red** markers for unsuccessful launches

- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map

- Explain the important elements and findings on the screenshot

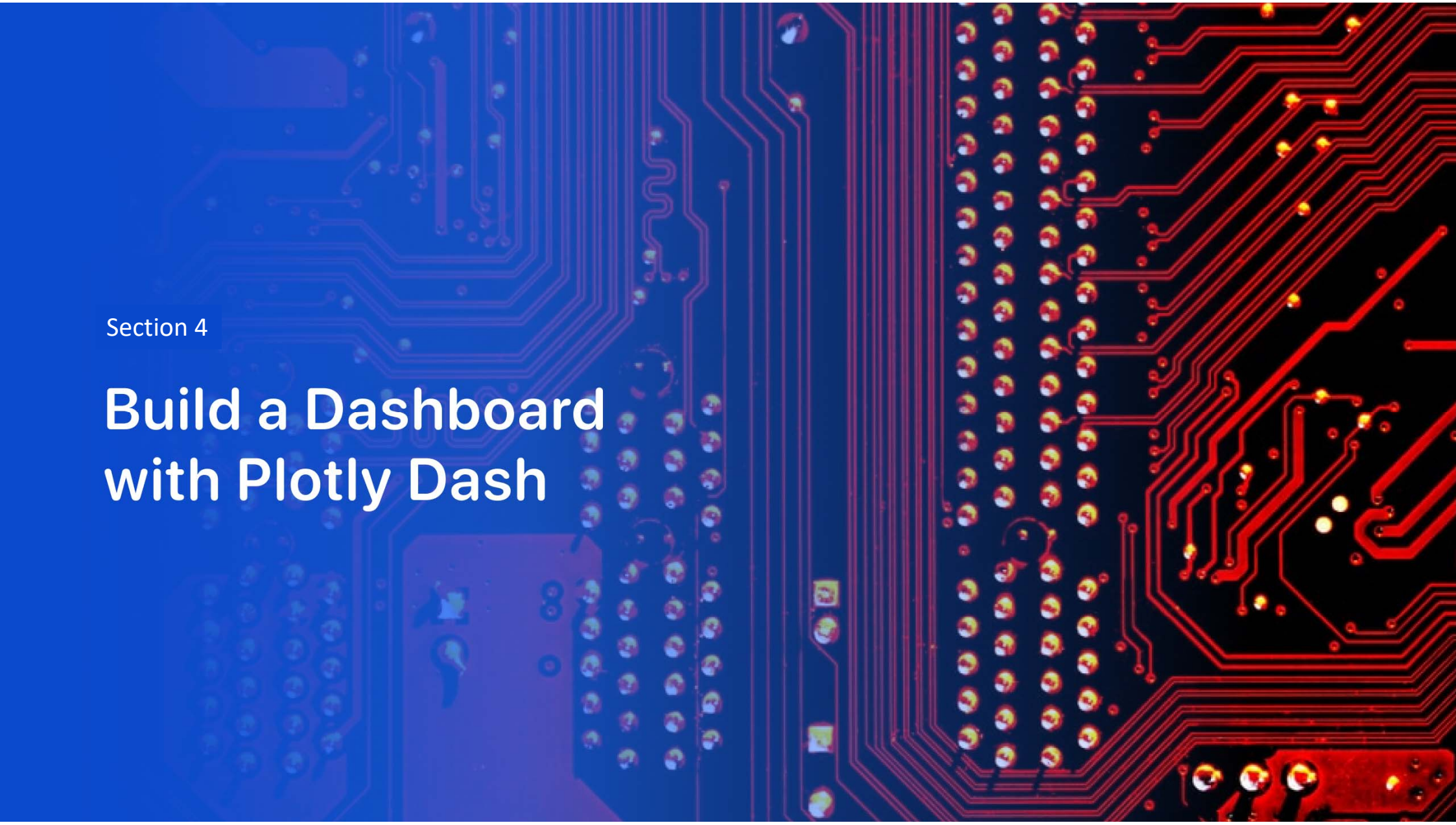


Launch Site is close to coastline

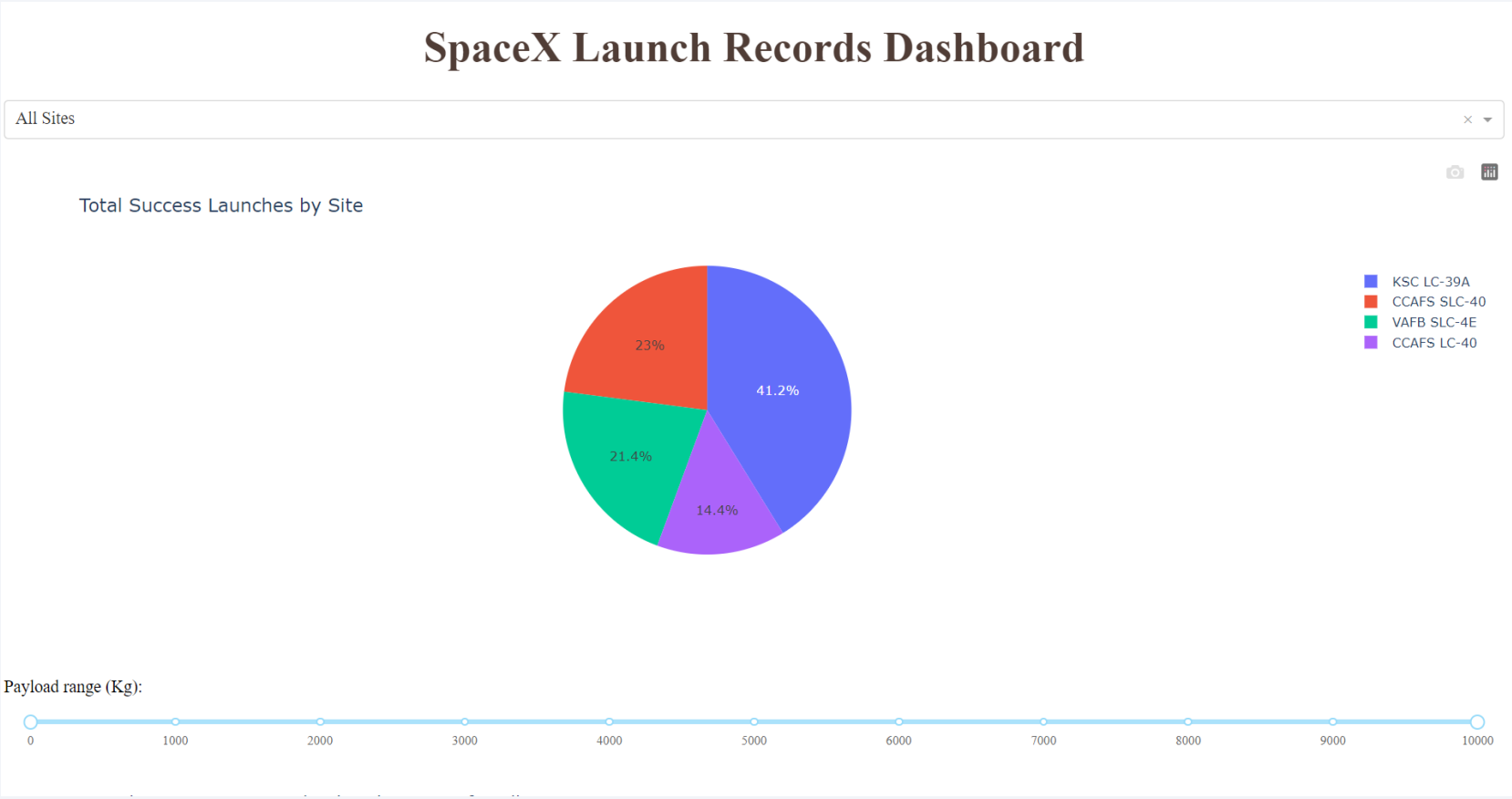


- City Distance (km) 23.234752126023245
- Railway Distance (km) 21.961465676043673 **CCAFS SLC-40**
- Highway Distance (km) 26.88038569681492
- **Coastline Distance (km) 0.8627671182499878**

- Coasts: help ensure that spent stages dropped along the launch path or failed launches don't fall on people or property.
- Safety / Security:needs to be an exclusion zone around the launch site to keep unauthorized people away and keep people safe.
- Transportation/Infrastructure and Cities: need to be away from anything a failed launch can damage, but still close enough to roads/rails/docks to be able to bring people and material to or from it in support of launch activities.

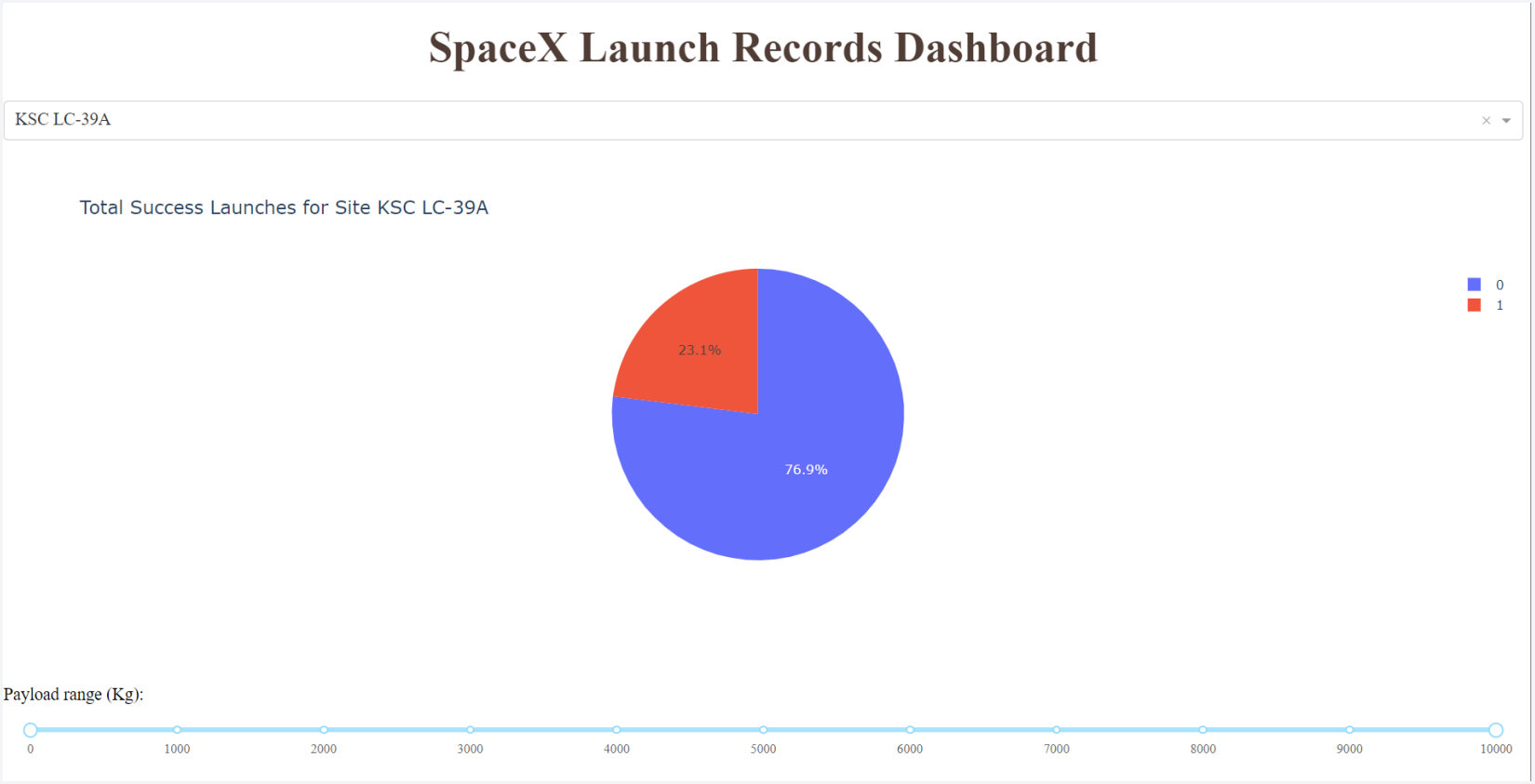

Section 4
Build a Dashboard with Plotly Dash

Launch Success by Site



- KSC LC-39A has the **most successful launches** (41.2%)

Launch Success KSC LC 39A



- **KSC LC-39A** has the **highest success rate** amongst launch sites (**76.9%**)

Payload Mass and Success

By Booster Version

- **Payloads between2,000 kg** and **5,000 kg** have the **highest success rate**
- 1 indicating successful outcome and 0 indicating an unsuccessful outcome

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- **All** the **models** performed at about the same level and had the **same scores** and **accuracy**. This is likely due to the **small dataset**. The **Decision Tree model slightly outperformed**
- Best model is DecisionTree with a score of 0.8732142857142856
- Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}

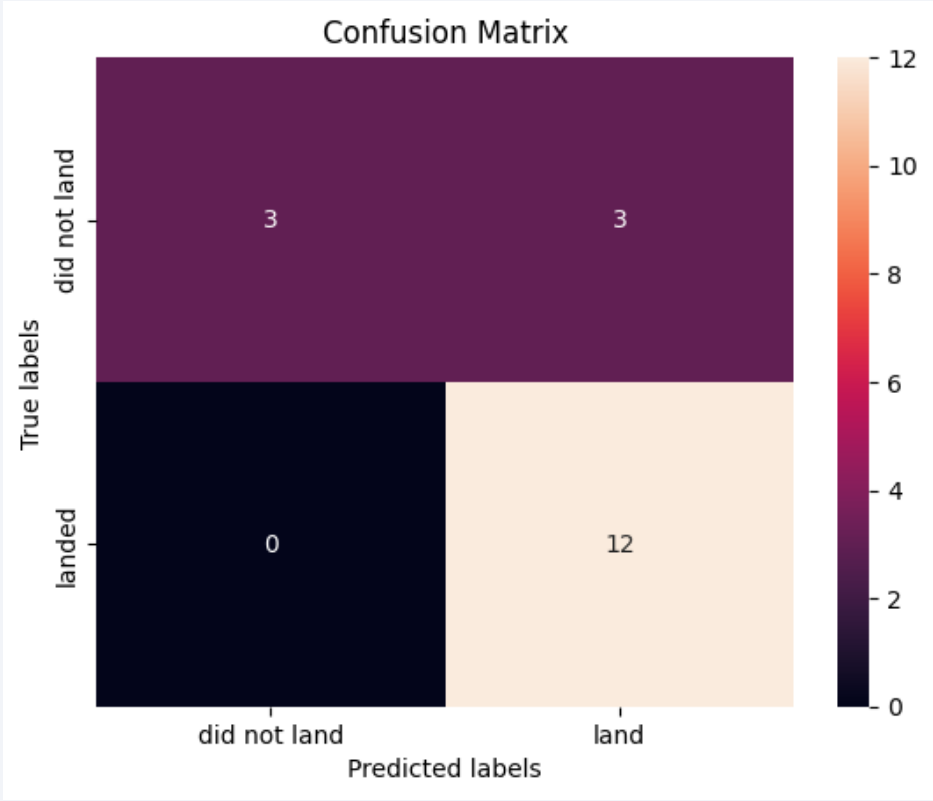
```
models = {'KNeighbors':knn_cv.best_score_,
          'DecisionTree':tree_cv.best_score_,
          'LogisticRegression':logreg_cv.best_score_,
          'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

Confusion Matrix

Performance Summary

- A confusion matrix summarizes the performance of a classification algorithm
- All the confusion matrices were identical
- The fact that there are false positives (Type 1 error) is not good
  - Confusion Matrix Outputs:12 True positive
  - 3 True negative
  - **3 False positive**
  - 0 False Negative



Conclusions

- **Model Performance**: The models performed similarly on the test set with the decision tree model slightly outperforming
- **Equator**: Most of the launch sites are near the equator for an additional natural boost -due to the rotational speed of earth -whichhelps save the cost of putting in extra fuel and boosters
- **Coast**: All the launch sites are close to the coast
- **Launch Success**: Increases over time
- **KSC LC-39A**: Has the highest success rate among launch sites. Has a 100% success rate for launches less than 5,500 kg
- **Orbits**: ES-L1, GEO, HEO, and SSO have a 100% success rate
- **Payload Mass**: Across all launch sites, the higher the payload mass (kg), the higher the success rate

Thank you!