IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Alex Kumbar
2025-02-21

Outline

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

Executive Summary

- 4 Classification models were used to determine the likelihood of a stage 1 booster being recoverable
- All models performed similarly with Support Vector Machine being slightly more consistent.
- Using current feature set model can predict reusability with 83% accuracy
- More data would increase confidence in the model's ability to generalize.

Introduction

- Launching material into space is very expensive
  - Costing upward of $165 million per launch
- SpaceX Stage 1 reusability
  - SpaceX claims their launches only cost $62 million
  - They reuse the stage 1 booster, which greatly reduces cost
- Any business attempting to enter the space needs to predict costs
  - What factors correlate with recovering a stage 1 booster?
  - What is the likelihood of recovering a stage 1 booster?

Section 1

Methodology

Methodology

Overview

- Data collection methodology:
- Data wrangling
- Exploratory data analysis (EDA) using visualization and SQL
- Interactive visual analytics using Folium and Plotly Dash
- Predictive analysis using classification models

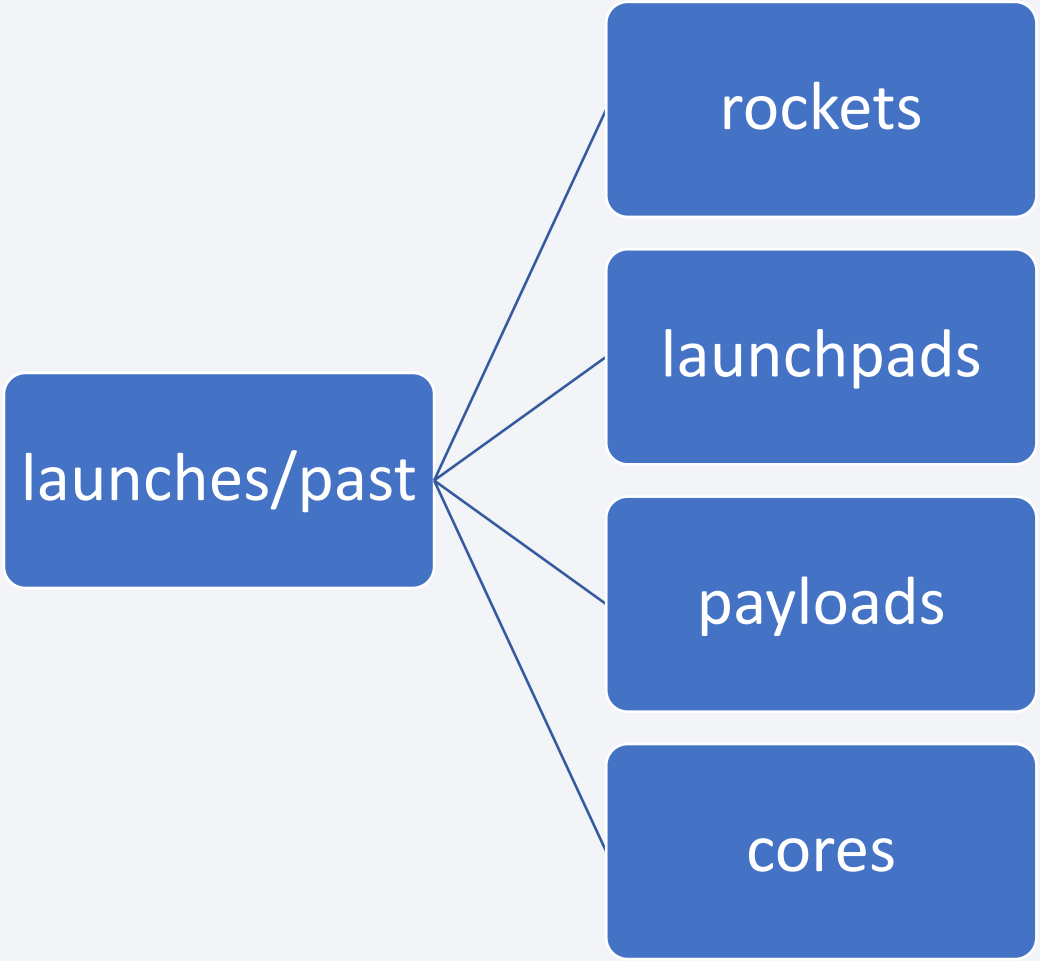
Data Collection

SpaceX API

Wikipedia Scraping

Data Collection – SpaceX API

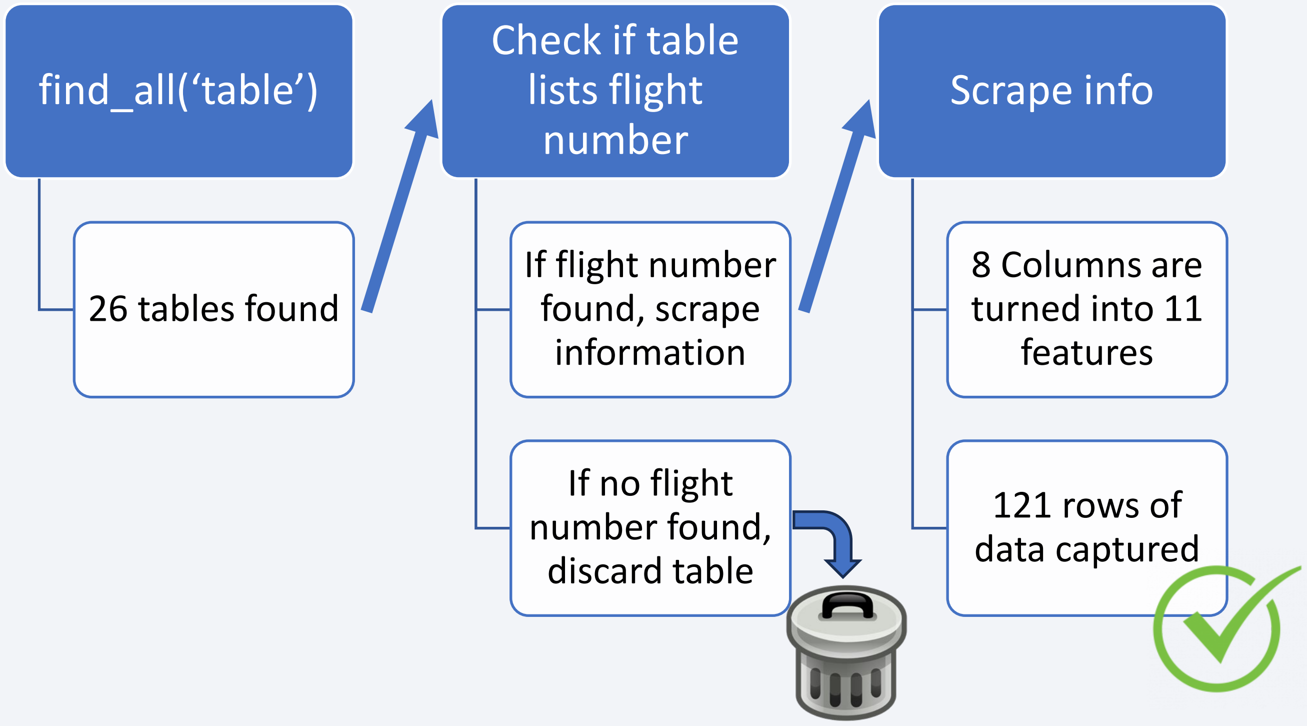
- SpaceX REST API can be reached at https://api.spacexdata.com/v4/
- First request was made to "launches/past" then based on the response info requests were made to the 4 other endpoints
- https://github.com/Rabmuk/Coursera-IBM-Data-Science-Capstone/blob/main/1%20jupyter-labs-spacex-data-collection-api-v2.ipynb



Data Collection - Scraping

- Web scraping from Wikipedia at https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches

- https://github.com/Rabmuk/Coursera-IBM-Data-Science-Capstone/blob/main/2%20jupyter-labs-webscraping.ipynb



Data Wrangling

- Organized data by Launch Site, Orbit, and Landing Outcome
- Needed to simplify Landing Outcome Features
  - Originally 8 different classification options
  - Simplified to a binary “success” or “failure”

- LandingPad has 26 null values. These are left alone because when this feature is one-hot-encoded the null values will be 0’s

- https://github.com/Rabmuk/Coursera-IBM-Data-Science-Capstone/blob/main/3%20labs-jupyter-spacex-Data%20wrangling-v2.ipynb

EDA with Data Visualization

- Quickly see relationships between landing success and various features
  - Flight number, launch site, and success
  - Payload mass, launch site, and success
  - Success rate by target orbit
  - Flight number, target orbit, and success
  - Payload mass, target orbit, and success
  - Average success by year
- Charts will be provided in Section 2
- https://github.com/Rabmuk/Coursera-IBM-Data-Science-Capstone/blob/main/5%20jupyter-labs-eda-dataviz-v2.ipynb

EDA with SQL

- Analysis of Launch data
  - Launch location, summarized and specific
  - Customer
  - Payload mass vs booster type
  - First successful launch
  - Booster types used for largest payloads
  - Dates of failures
  - Outcomes during specific time period
- https://github.com/Rabmuk/Coursera-IBM-Data-Science-Capstone/blob/main/4%20jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Group launch data into the 4 launch site and display on map
  - Use markers to display the success or failure status of launches
- Map showing launches and their suroundings. Mapping distance to relevant landmarks like:
  - Roads
  - Railroads
  - Cities
  - Ocean

- https://github.com/Rabmuk/Coursera-IBM-Data-Science-Capstone/blob/main/6%20lab-jupyter-launch-site-location-v2.ipynb

Build a Dashboard with Plotly Dash

- Pie Chart for Summary of all locations successes or single location success and failure
- Success vs Failure scatterplot, filterable on location and payload weight
- This dashboard quickly gives information on how location and payload mass affects success of retrieving the stage 1 booster

- https://github.com/Rabmuk/Coursera-IBM-Data-Science-Capstone/blob/main/7%20spacex-dashboard.py

Predictive Analysis (Classification)

- Using 80 features (most coming from one-hot-encoding of categorical data) trained 4 different models
- Split data to have 20% testing then split training data into 10 folds for training and validation
- https://github.com/Rabmuk/Coursera-IBM-Data-Science-Capstone/blob/main/8%20SpaceX-Machine-Learning-Prediction-Part-5-v1.ipynb

Logistic Regression	Support Vector Machine
Decision Tree	K-Nearest Neighbors

Results

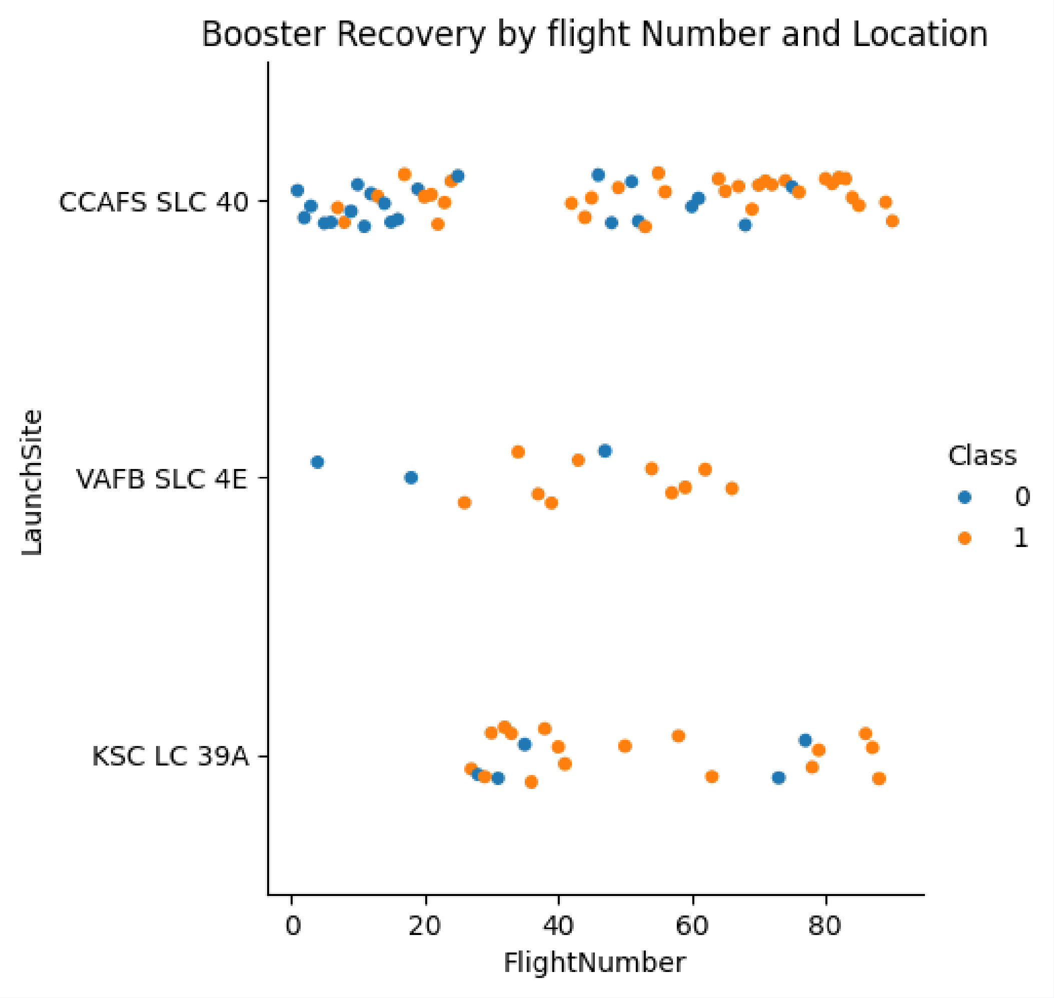
- SVM is the best model.
- Sometimes Decision Tree performs better than SVM, but depending on the random seed when it is trained the test accuracy for Decision Tree is sometimes less than SVM
    - Certain random seeds can cause Decision Tree to overfit for the training data
- Logistic Regression and KNN also have high training and testing accuracy, just slightly less than SVM

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

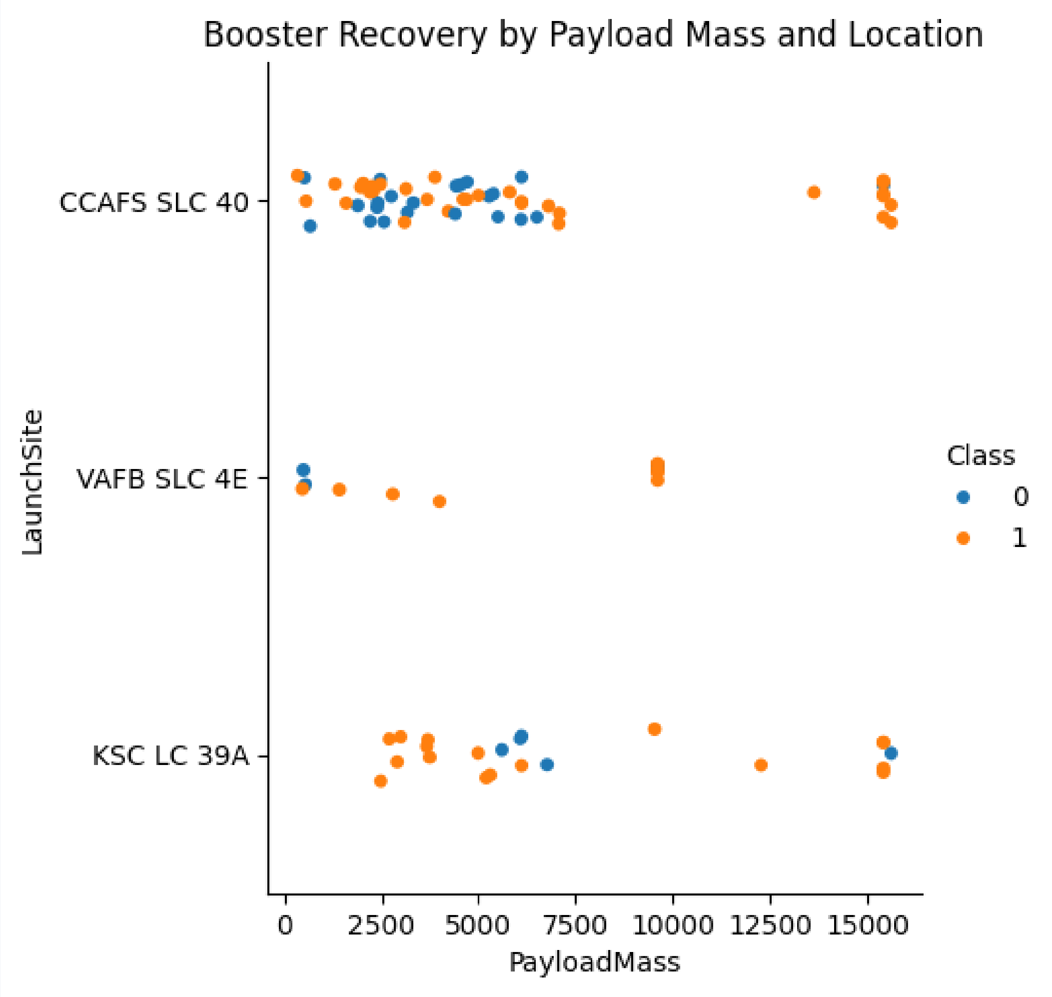
- Not all Launch Site operating continuously.
- Overtime Booster Recovery improved
- CCAFS SLC 40 had the most amount of failed recoveries



Payload vs. Launch Site

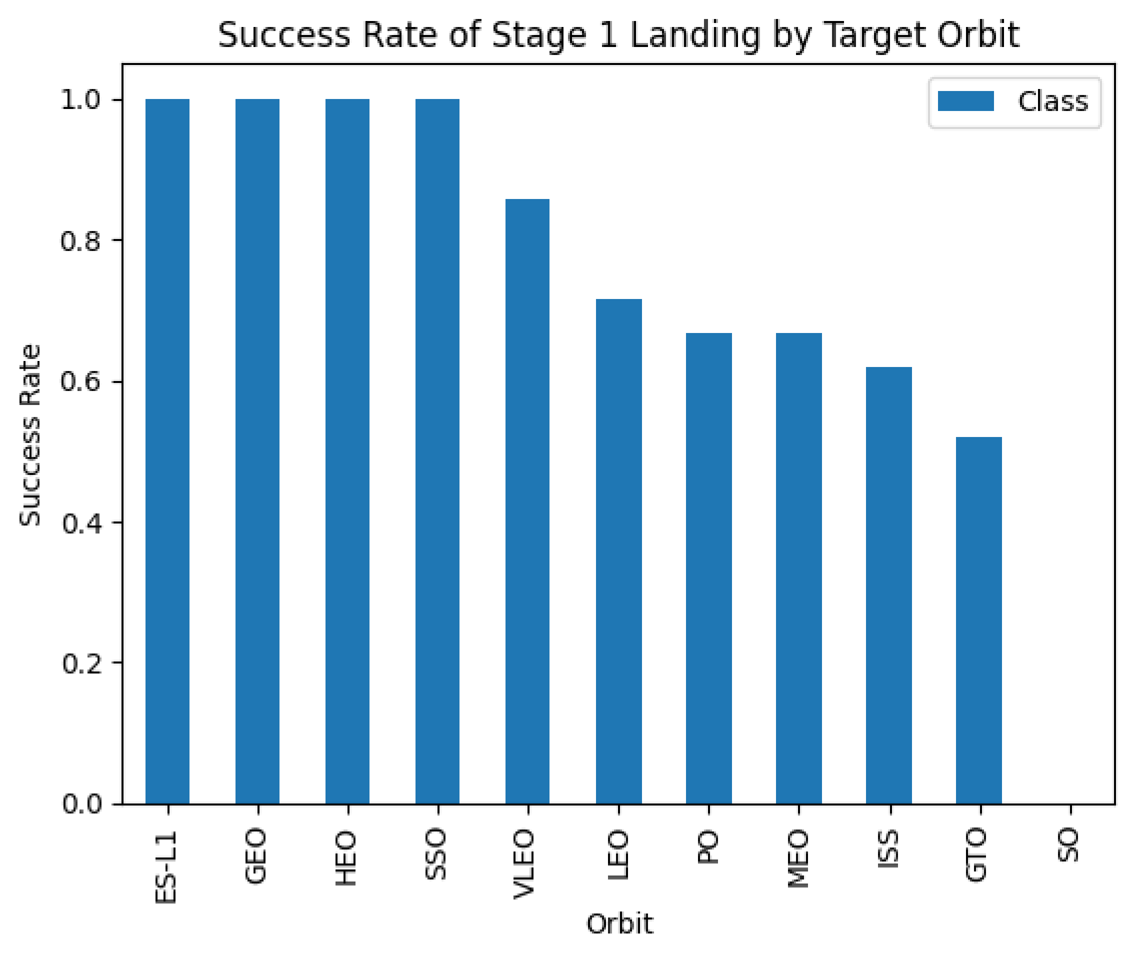
- High payload launches have high success rate

- VAFB SLC 4E has no launches with very heavy payloads



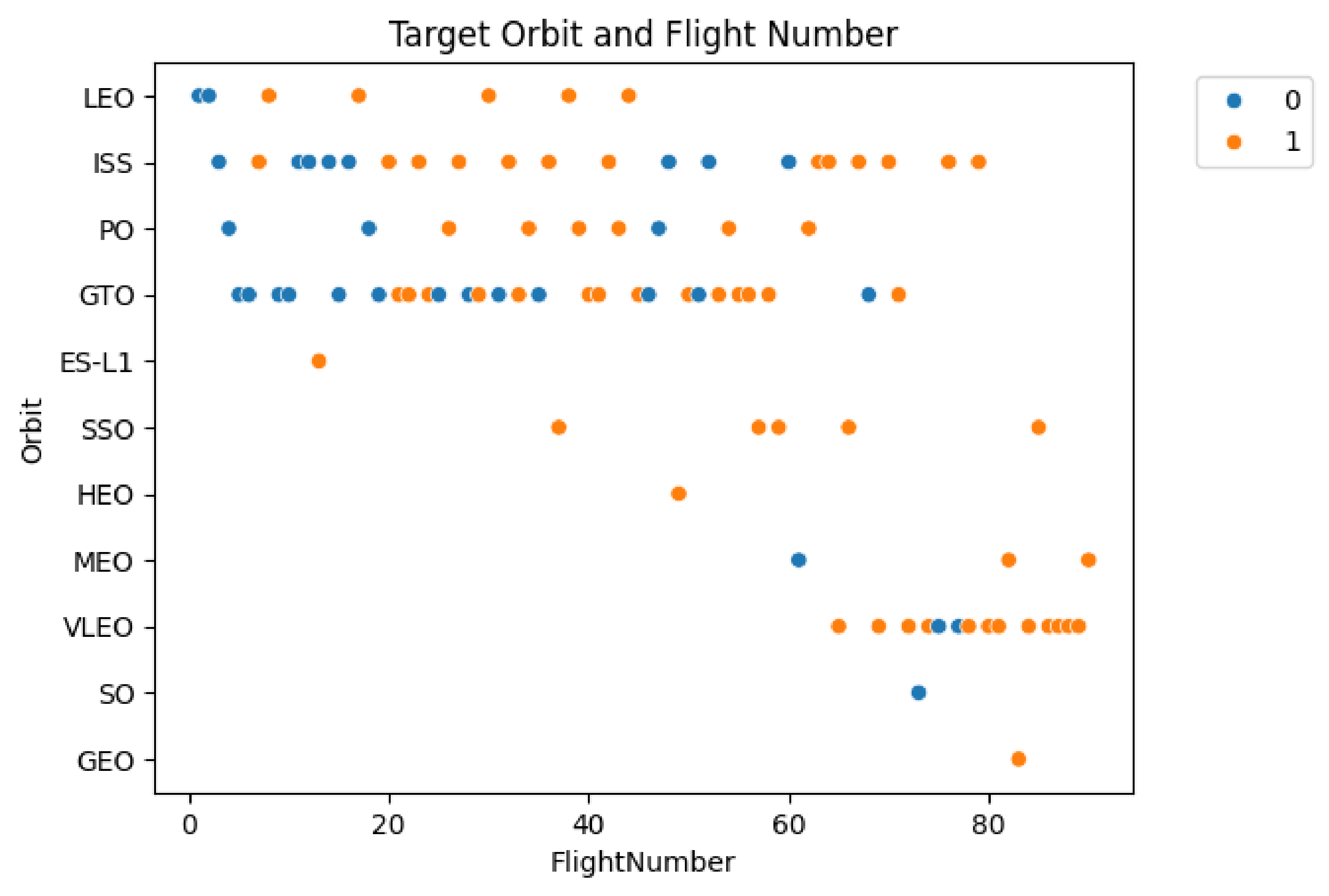
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SO each only had 1 data point
- All others had 3+ data points
- SSO is impressive with 5 launches and all succesful



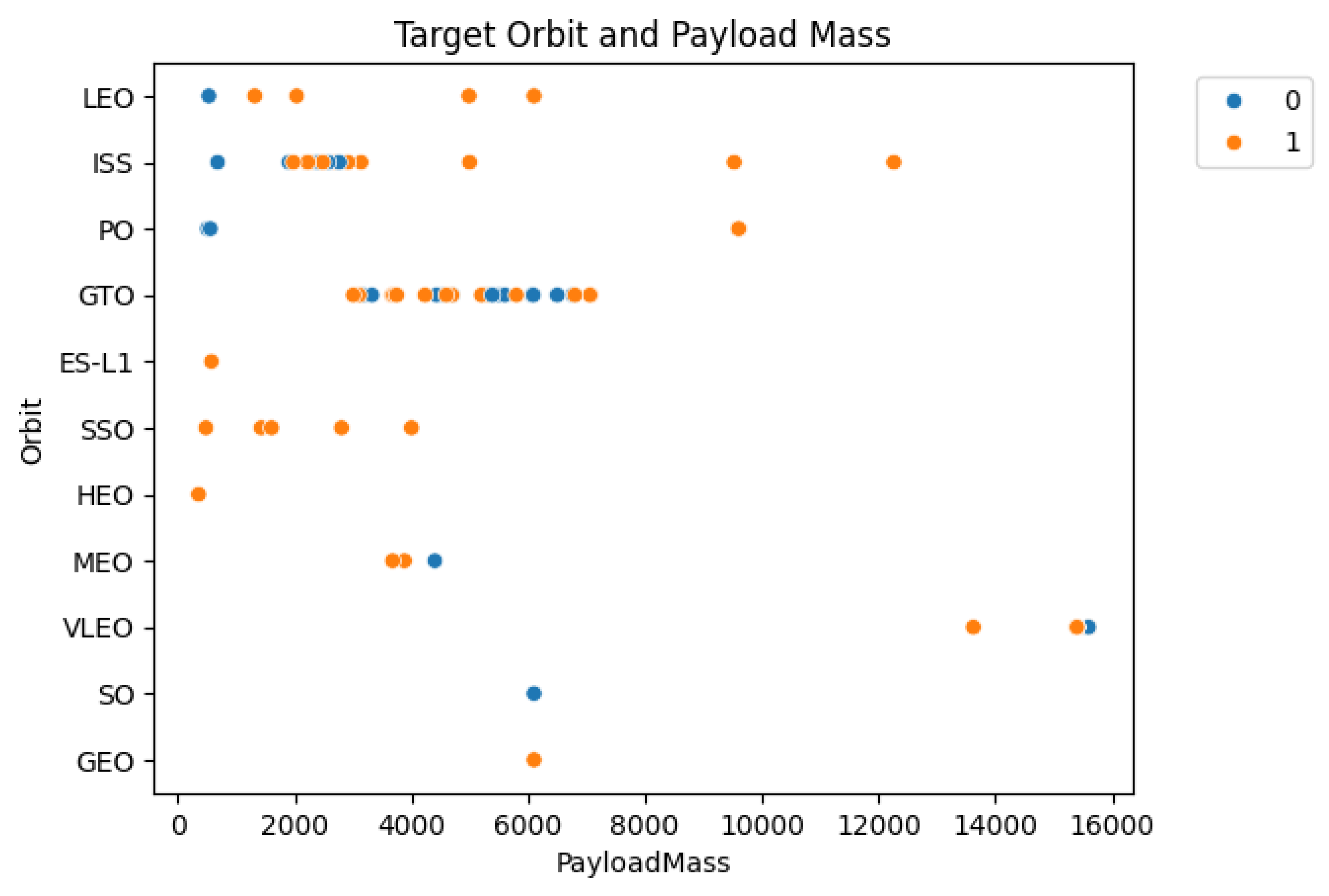
Flight Number vs. Orbit Type

- Over time LEO, PO, and GTO became less popular
- SSO and VLEO became much more common later on



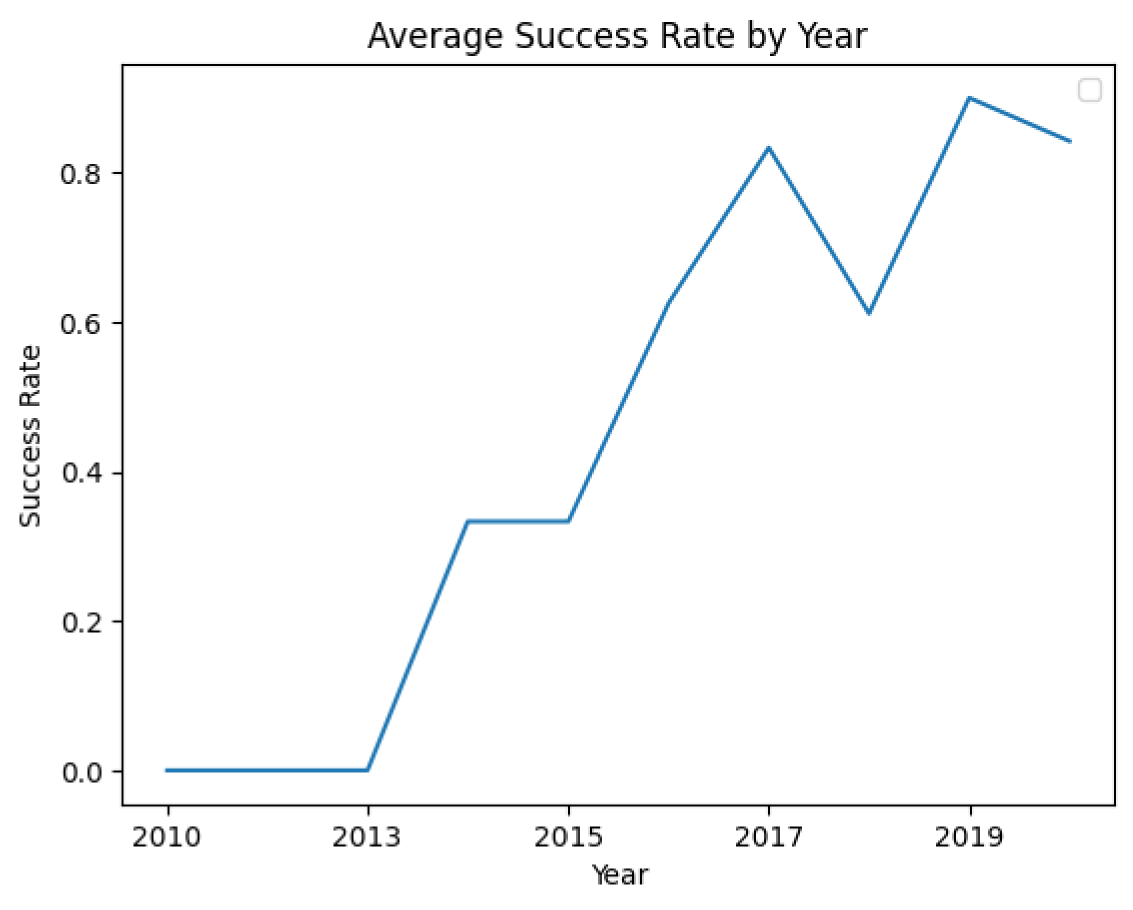
Payload vs. Orbit Type

- Payload mass for most orbit types fits into a certain range.
  - GTO is very grouped up
- ISS and PO have the most variation in payload mass



Launch Success Yearly Trend

- General Trent of increasing success rate
- Setback in 2018, but recovered strong for 2019



All Launch Site Names

- SQL query to find Distinct Launch Site names

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

Launch Site Names Begin with 'CCA'

- Using “like” comparison and the % wildcard

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
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	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- Using the Sum function and checking where Customer is ‘NASA (CRS)’

- 45596 kg was total for that customer

Average Payload Mass by F9 v1.1

- Using the Avg function and check when Booster version was “like” “F9 v1.1%”

- 2534.66 kg is the average weight launched by the F9 v1.1

First Successful Ground Landing Date

- Using the min function and filtering outcome by success

- 2015-12-22 was the first successful stage 1 rocket booster landing

Successful Drone Ship Landing with Payload between 4000 and 6000

- Using 3 conditions for a where clause to filter by success, upper, and lower payload mass

Booster_Version	PAYLOAD_MASS__KG_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

Total Number of Successful and Failure Mission Outcomes

- Group by was used to quickly count the occurrences of different mission outcomes

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

Boosters Carried Maximum Payload

- Where clause comparing payload mass to a subquery that finds the max of payload mass for the table

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

- Use substr function to extract month and year information from the data column

Date	Month	Landing_Outcome	Booster_Version	Launch_Site
2015-01-10	01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Use Group by and Order by with a where clause to gather relevant data

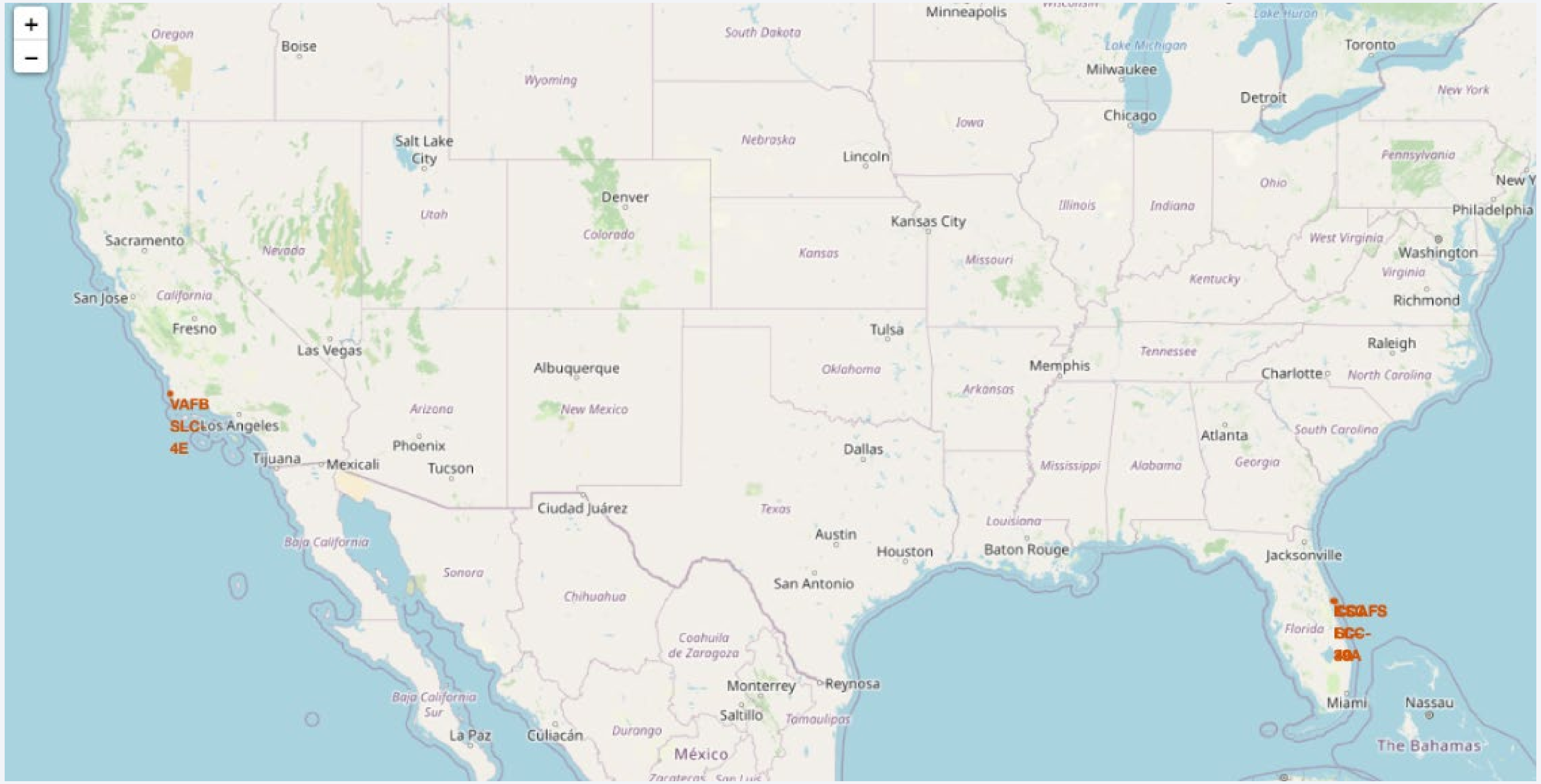
Landing_Outcome	Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

Section 3

Launch Sites Proximities Analysis

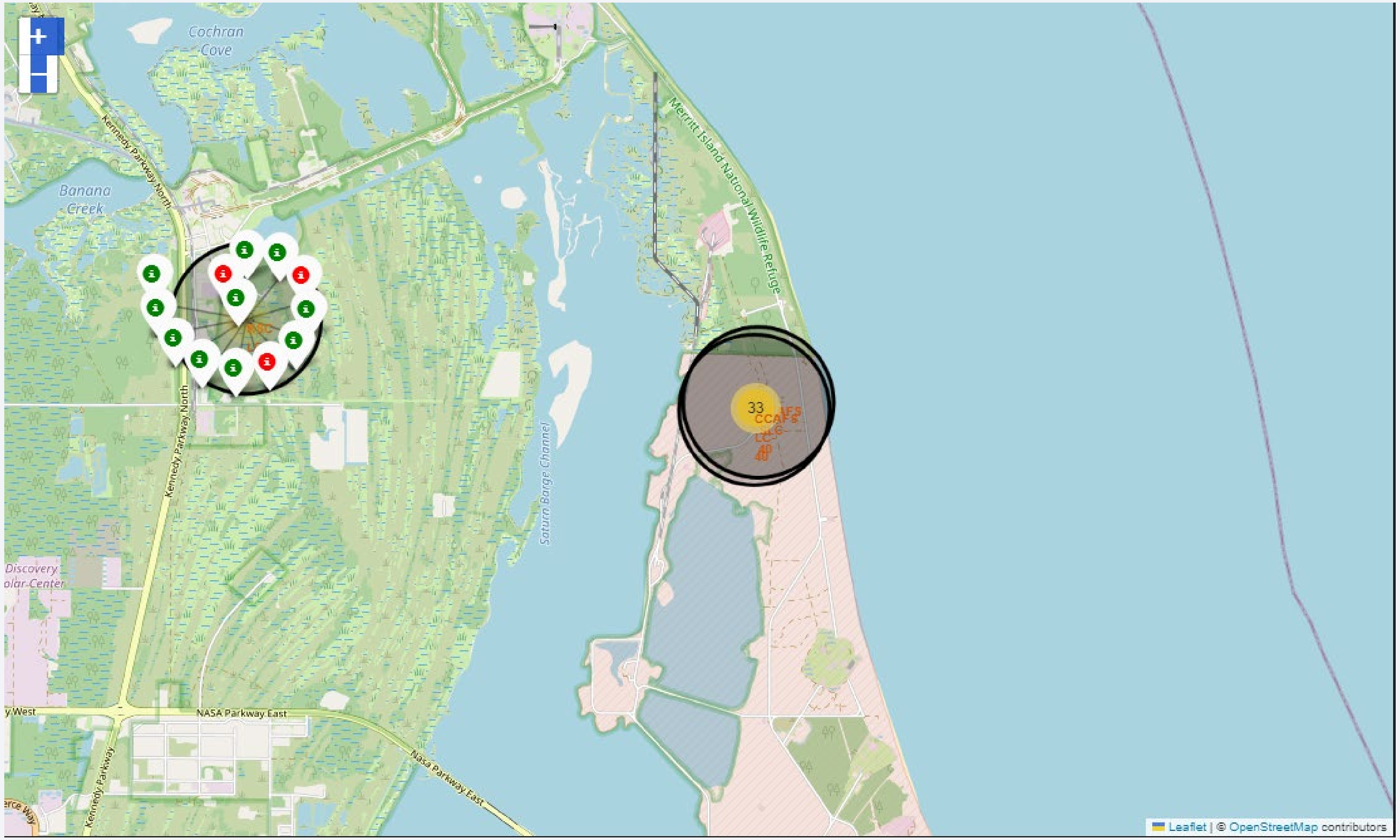
SpaceX Launch Sites

- One Launch Site in California
- Three Launch sites in Florida



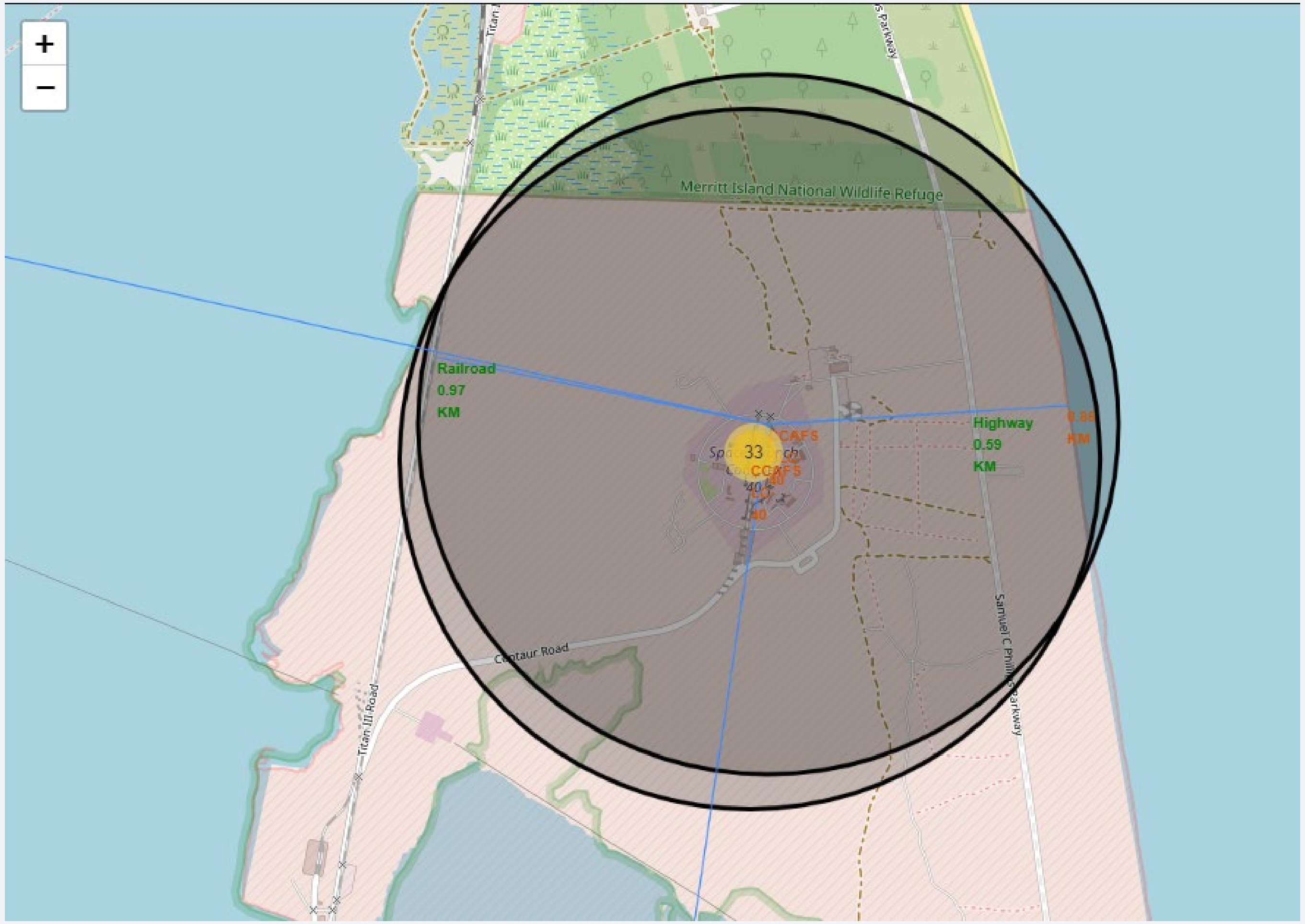
Markers Grouped and Details

- Marker groups will summarize how many launches occurred at a site
- Clicking on a site will display info markers with Green for success and Red for failure



Distance to nearby features

- Looking at CCAFS site with lines drawn to nearest coast, highway, railroad, and two near cities.
- Distances are labeled (city distances are off screen)

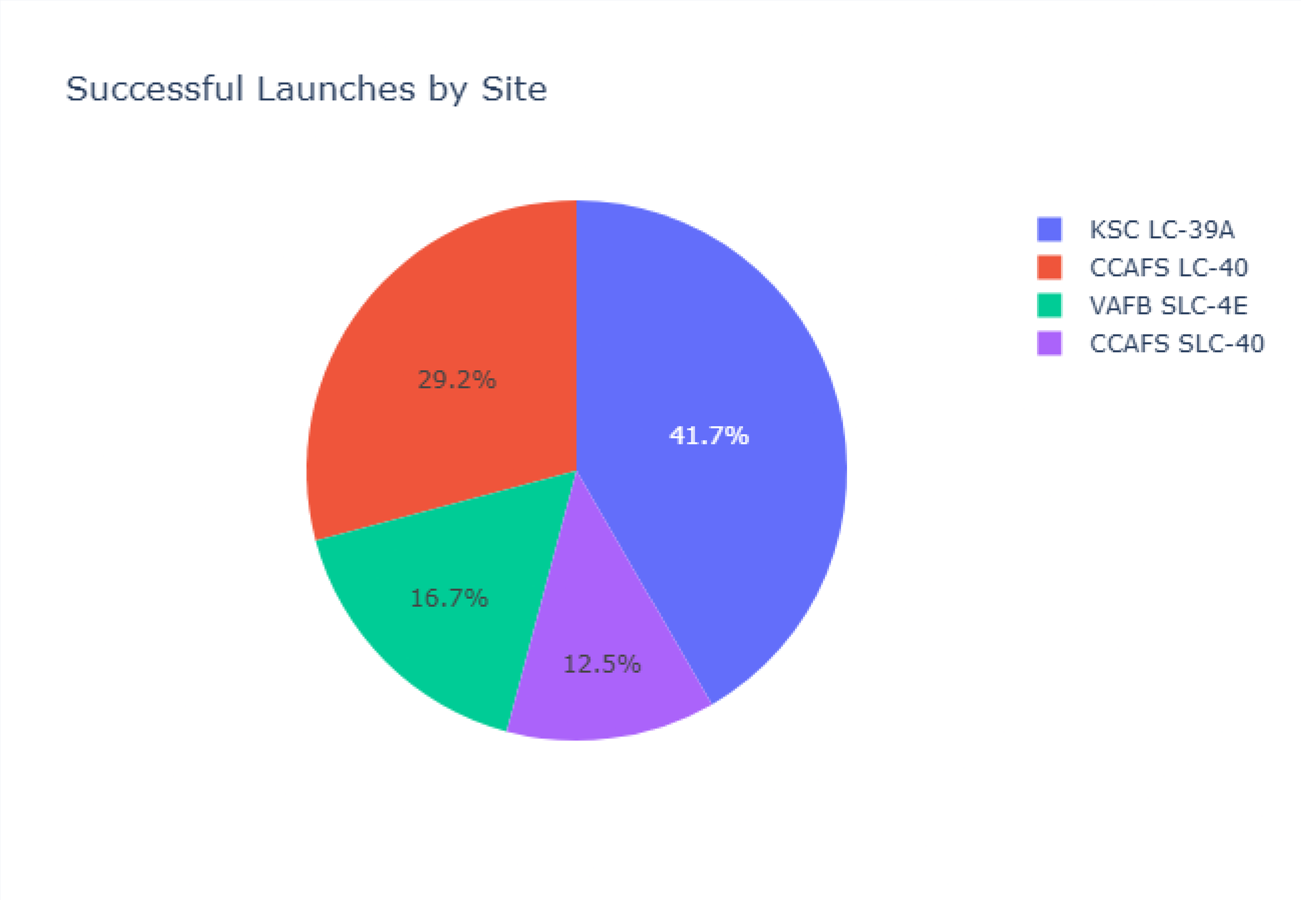


Section 4

Build a Dashboard with Plotly Dash

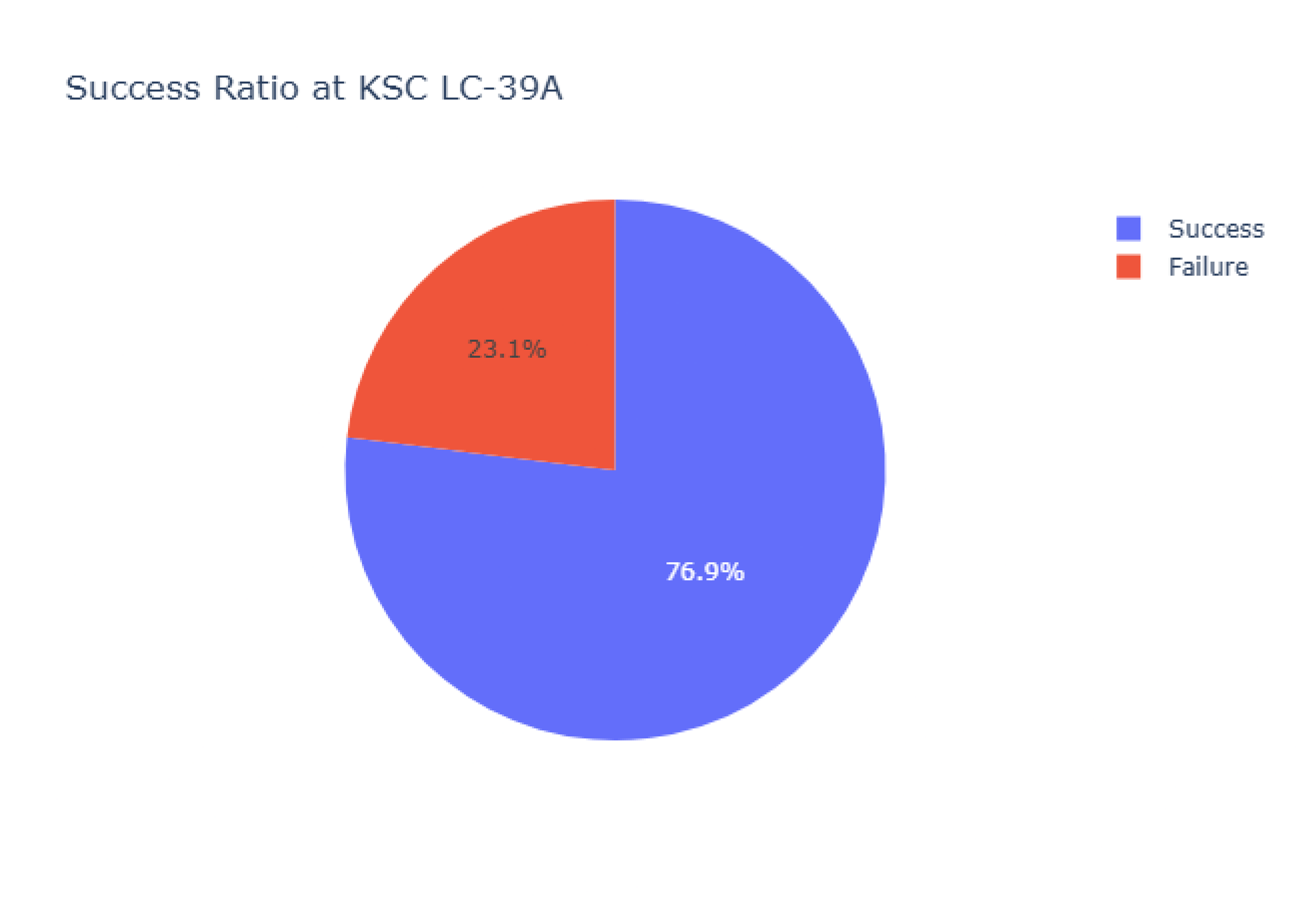
Success by location

- Plotly Pie chart for successful launches by location
- Quickly get an overview of which locations have the most successes
- With the option to drill down into each location to see success rate



Site with best success ratio

- KSC LC-39A has the higher ratio of successful launches



Success by Location and Payload Mass

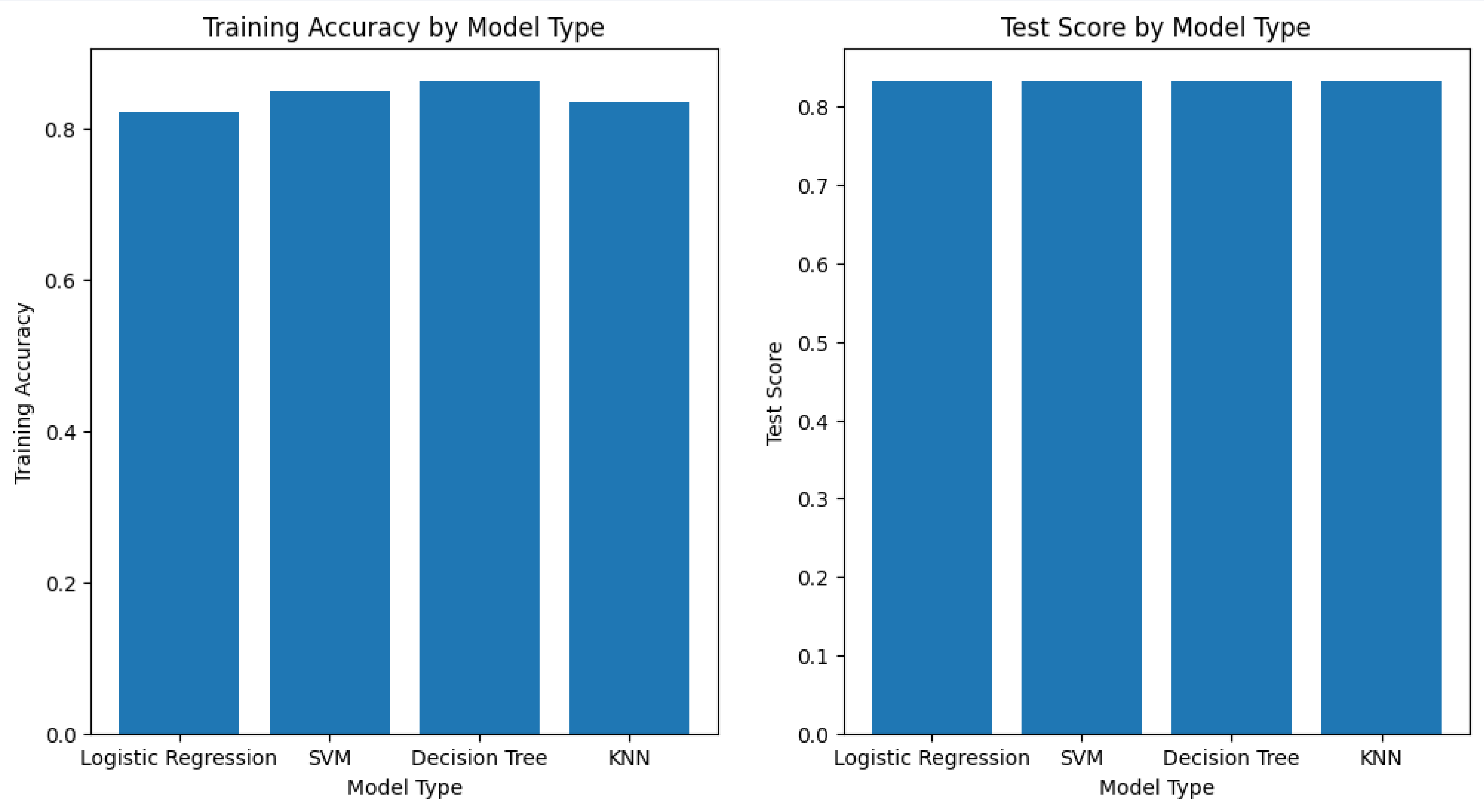
- Ability to use a two pointer slider to set lower and upper limit for payload mass. Can also be filtered by location

Section 5

Predictive Analysis (Classification)

Classification Accuracy

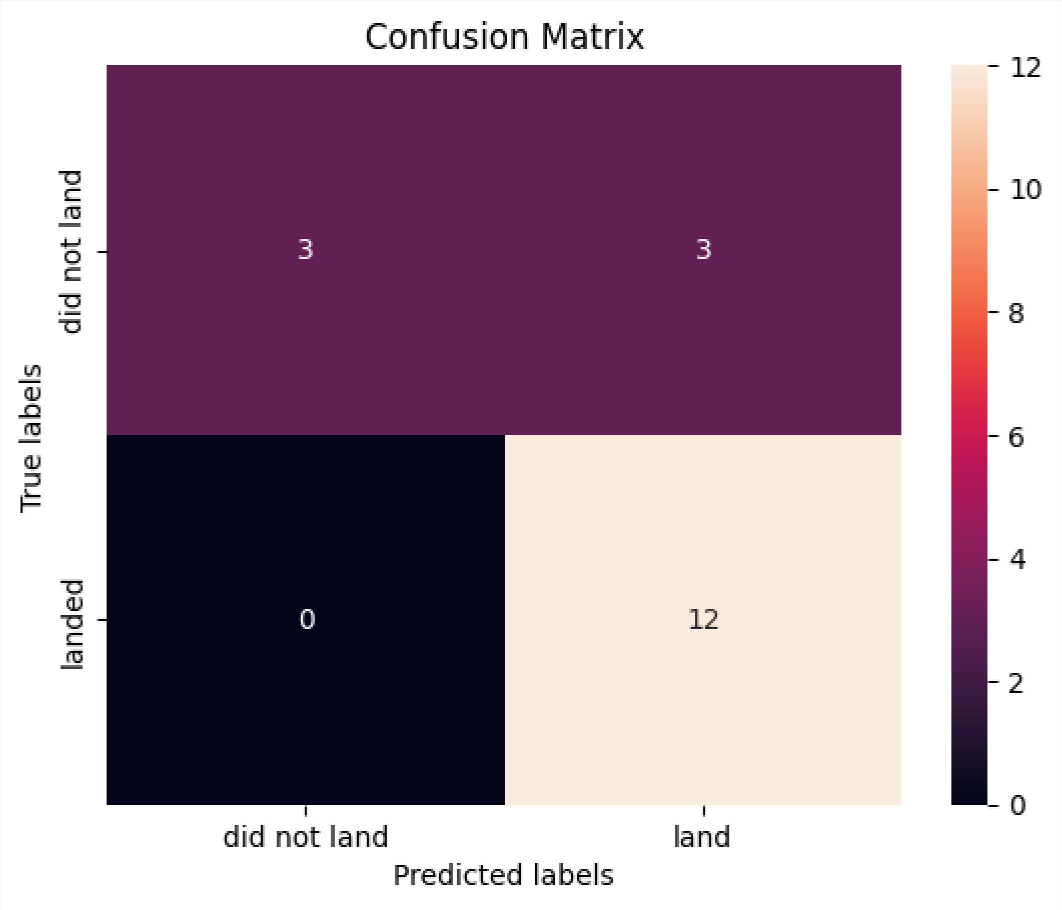
- All models had the same test accuracy when first running the notebook. If the Decision Tree was run again, it sometimes has a lower Test accuracy.
- SVM has best Training accuracy while never dropping in Test accuracy



Confusion Matrix

- When asked to predict the test data SVM had 3 false positives.

- The SVM matrix shows there were no false negatives



Conclusions

- The information gathered can be useful for any space rocketry company looking to control costs
- By understanding the factors that maximize the likelihood of reusing the stage 1 booster, a company can avoid the huge expense of building a new booster for each launch
- The analysis was a bit limited by the amount of data, with only 18 samples in the test set, it was hard to differentiate which models was truly best
- Continual data collection is very important. A pipeline for each model should be created to see if test accuracy diverges as more data becomes available.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Appendix: Pipeline

- As more data is collected a pipeline can be used to regularly retest model accuracy. Here is an example for Logistic Regression. Pipelines should be built for all 4 models

```
from sklearn.pipeline import Pipeline

from sklearn.preprocessing import StandardScaler

parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}

model_cv = GridSearchCV(

    LogisticRegression(),

    param_grid=parameters,

    cv=10,

)

pipe = Pipeline([

    ('scaler',StandardScaler()),

    ('cv',model_cv),

])
```

```
pipe.fit(X_train,Y_train)

pipe.score(X_test,Y_test)
```

Thank you!