IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Bill Louer
August 22, 2024

Outline

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

Executive Summary

- Summary of methodologies:
  - REST API and Webscraping were used for data acquisition
  - Basic data wrangling was performed
  - Exploratory data analysis was used
  - Interactive plotting using Folium maps and Plotly Dash was developed
  - Four (4) classification models were developed and evaluated based on a training and testing dataset.
- Summary of all results:
  - SpaceX launches from 4 locations (3 FL, 1 CA)
  - SpaceX has improved its landing success over time. KSC has the highest success rate.
  - Models were developed for KNN, Decision Tree, Support Vector Machines and Logistic Regression
  - All models demonstrated a validation accuracy of 83%
  - A logistic regression model was selected for its ability to predict classification probability

Introduction

- SpaceY: Private rocket launch company engaged in space travel; competing with SpaceX
- SpaceX's advantage: Re-use the Rocket's1st stage to keep costs low; $62MM vs. $165MM!!
- Prediction of a successful Falcon 1st stage landing will help SpaceY predict the cost of their competitor SpaceX's, next launch.
- Objective: Develop predictive tool for SpaceY to predict if SpaceX's Falcon 9 will land the 1st stage successfully

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX API
  - Wikipedia Web Scraping using Beautiful Soup
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Normalize data and build training and testing datasets
  - Compare logistical regression, support vector machines, decision tree regression, K nearest neighbor models types and evaluate parameters using GridSearchCV.
  - Evaluate scoring accuracy of test data set and select model.

Data Collection

- Downloaded the course json file
- Called SpaceX API to get SpaceX Rockets, Launches and Payloads data based on ID numbers.
- Screened data to Falcon 9 launches and replaced missing payload mass entries with mean payload
- Web-scraping of Wikipedia page for Falcon9 and Falcon Heavy Data (BeautifulSoup)

Data Collection – SpaceX API

- Data acquisition with SpaceX REST API calls.

- GitHub URL: https://github.com/wlouer/IBM_Final_Project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

1. Response from get request

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_
response = requests.get(static_json_url)
```

2. Convert to json and normalize to df

```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

3. Use functions to call API to acquire Booster, LaunchPad, Core and Payload data

4. Filter the data to acquire Falcon9 launches only

```
# Hint data['BoosterVersion']!='Falcon 1'
filter = data_falcon9['BoosterVersion'] == 'Falcon 9'
data_falcon9[filter]
```

5. Write data to csv file:

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

- Web-Scraping using Beautiful Soup

- GitHub URL: https://github.com/wlouer/IBM_Final_Project/blob/main/jupyter-labs-webscraping.ipynb

1. Request Falcon9 Wiki page from url and parse html with BeautifulSoup

```
# use requests.get() method with the provided static_url
response = requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, 'html.parser')
```

2. Get column names from parsed html

```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
headers_all = first_launch_table.find_all('th')
for header in headers_all:
    name = extract_column_from_header(header)
    if name != None and len(name) > 0:
        column_names.append(name)
```

3. Build dictionary of lists containing each columns data

4. Convert to Dataframe

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

5. Save to csv file

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling (Continue Here)

- Clean NaN data and evaluate data values and grouping.
- GitHub URL (Data Wrangling): https://github.com/wlouer/IBM_Final_Project/blob/main/jupyter-labs-webscraping.ipynb

1. Calculate each site's Launches:

```
In [5]:  # Apply value_counts() on column LaunchSite
         df.LaunchSite.value_counts()
```

2. Calculate the occurrence of each orbit

```
In [6]:  # Apply value_counts on Orbit column
         df.Orbit.value_counts()
```

3. Calculate the number of occurrence of mission outcomes.

```
In [8]:  landing_outcomes = df.Outcome.value_counts()
```

4. Convert landing outcomes to binary values.

```
In [12]: # landing_class = 0 if bad_outcome
         # landing_class = 1 otherwise
         landing_class = []
         for row in df['Outcome']:
             if row in bad_outcomes:
                 landing_class.append(0)
             else:
                 landing_class.append(1)

         print(landing_class)
```

EDA with Data Visualization

- Visualization Plots
  - Flight Number vs. Payload Mass: Has success/failure changed with flight number or mass?
  - Flight Number vs. Launch Site: Has success/failure changed with flight number vs. Launch Site?
  - Payload Mass and Launch Site: Has success/failure changed with flight number vs. Launch Site?
  - Success Rate vs. Orbit Type: How does Orbit influence landing success?
  - Flight Number vs Orbit type: For each orbit, does landing success increase?

EDA with Data Visualization (continued)

- Visualization Plots
  - Payload Mass and Orbit type: How does success/failure change with Orbit and Payload mass?
  - Launch success rate vs Year: Has launch success rate improved or lessened?

- GitHub URL of completed EDA with data visualization notebook:
  - https://github.com/wlouer/IBM_Final_Project/blob/main/edadataviz.ipynb

EDA with SQL

- SQL queries:
  - Unique Launch Sites
  - Launch sites that begin with CCA (Cape Canaveral)
  - Max payload mass when NASA was the customer
  - Average payload mass when the Booster version was V1.1
  - Date of first successful landing on a ground pad
  - Boosters resulting in successful landing on drone ship with payload (4,000 and 6,000 kg)
  - Number of successful and failed mission outcomes (not landing outcomes)

EDA with SQL (continued)

- SQL queries (Continued):
  - Name of booster version that carried max payload
  - Month names, failed landings on drone ship ,booster versions, launch_site for year 2015
  - Ranked count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order.
- GitHub URL of completed EDA with SQL notebook: https://github.com/wlouer/IBM_Final_Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Added circle marker and text for each launch site to see where, geographically the launch sites were.
- Marked each flight from each launch site with a marker cluster. Colors were added for success or failure to visualize:
  - The number of launches at each site
  - The success and failures at each site.
- Marked a line and measurement to the nearest ocean, railroad, highway and large city
  - Proximity to these features show how materials are transported to the sites
  - Proximity to cities and oceans show how safety is considered when conducting Rocket Launches
- GitHub URL: https://github.com/wlouer/IBM_Final_Project/blob/main/lab_jupyter_launch_site_location.html

Build a Dashboard with Plotly Dash

- Created pie-chart with dropdowns to:
  - Visualize ALL sites success rate
  - Visualize each site's successful and failed landings.
- Visualize successful and failed landings based on payload range for a dropdown selection of all sites or individual sites.
  - The range in payload was used as a slider
- GitHub URL of your completed Plotly Dash lab:

https://github.com/wlouer/IBM_Final_Project/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Normalized/Transformed the dataset using the Pre-processing Standard Scaler package in sci-kit learn
- Divided the dataset into a training set and a test set using 20% for testing.
- Created four (4) different models using GridSearch to find the best hyper-parameters:
  - Logistic Regression
  - Support Vector Machine
  - Decision Tree
  - K Nearest Neighbor
- Calculated the validation score on the test data set
- Plotted a confusion matrix to see the predictions vs. actual data
- GitHub URL of your completed predictive analysis lab:

https://github.com/wlouer/IBM_Final_Project/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Predictive Analysis (Classification)

- Model Development:
  - Logistic Regression
  - Support Vector Machine
  - Decision Tree
  - K Nearest Neighbor
- GitHub URL: https://github.com/wlouer/IBM_Final_Project/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

1. Normalized dataset using StandardScalar

```
# students get this
transform = preprocessing.StandardScaler()
X = transform.fit_transform(X)
```

2. Divided data into training and testing set

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state = 2)
```

3. Created and fit model

```
parameters = {'kernel':('linear', 'rbf','poly','rbf', 'sigmoid'),
              'C': np.logspace(-3, 3, 5),
              'gamma':np.logspace(-3, 3, 5)}
svm = SVC()
```

```
svm_cv = GridSearchCV(estimator=svm, param_grid=parameters, scoring='accuracy', cv=10)
svm_cv.fit(X_train, Y_train)
```

4. Measure test accuracy with testing data set

```
accuracy_svm_cv = svm_cv.score(X_test, Y_test)
accuracy_svm_cv
```

5. Plot confusion matrix and compare model accuracy

```
yhat=svm_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```

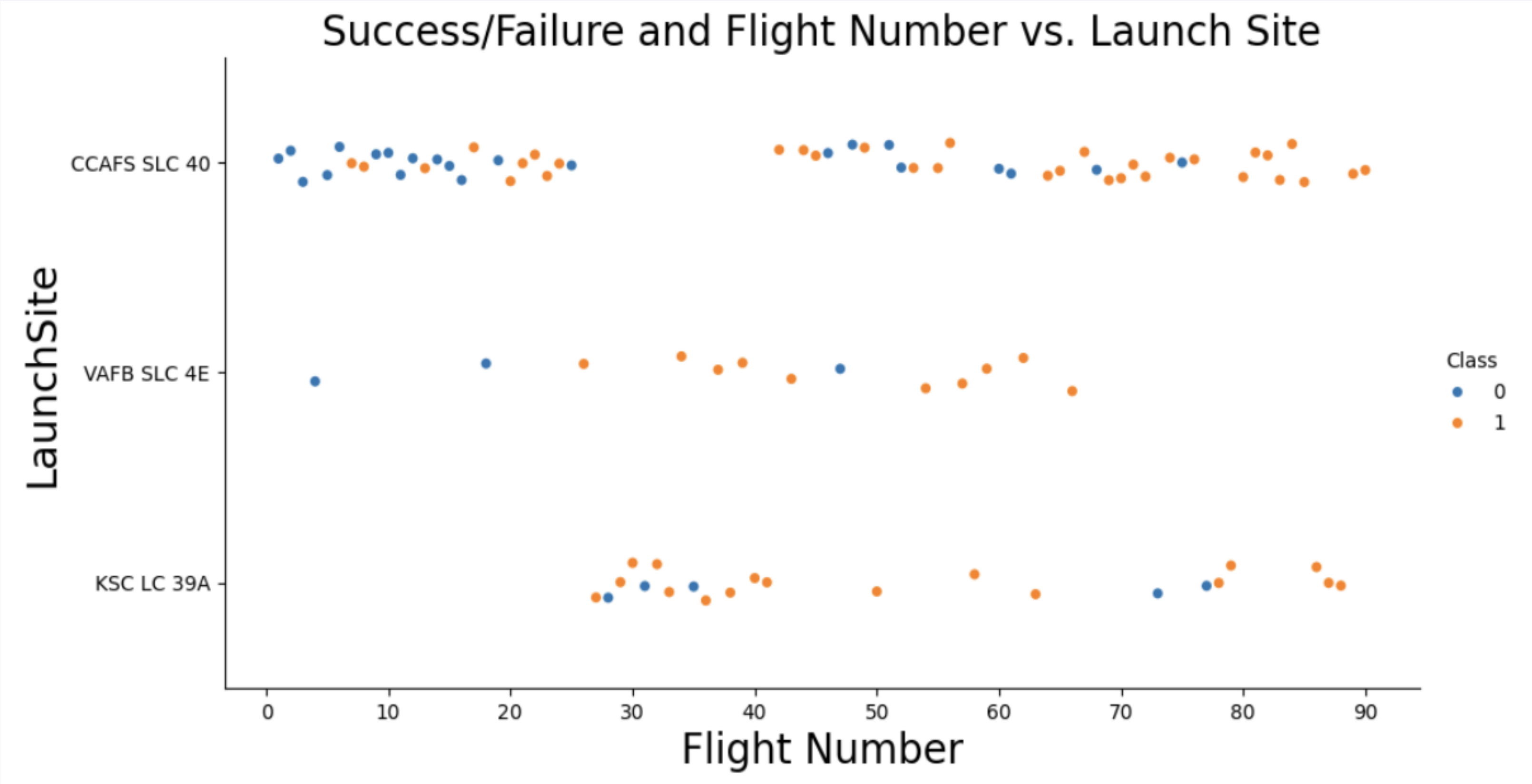
Results

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

Section 2

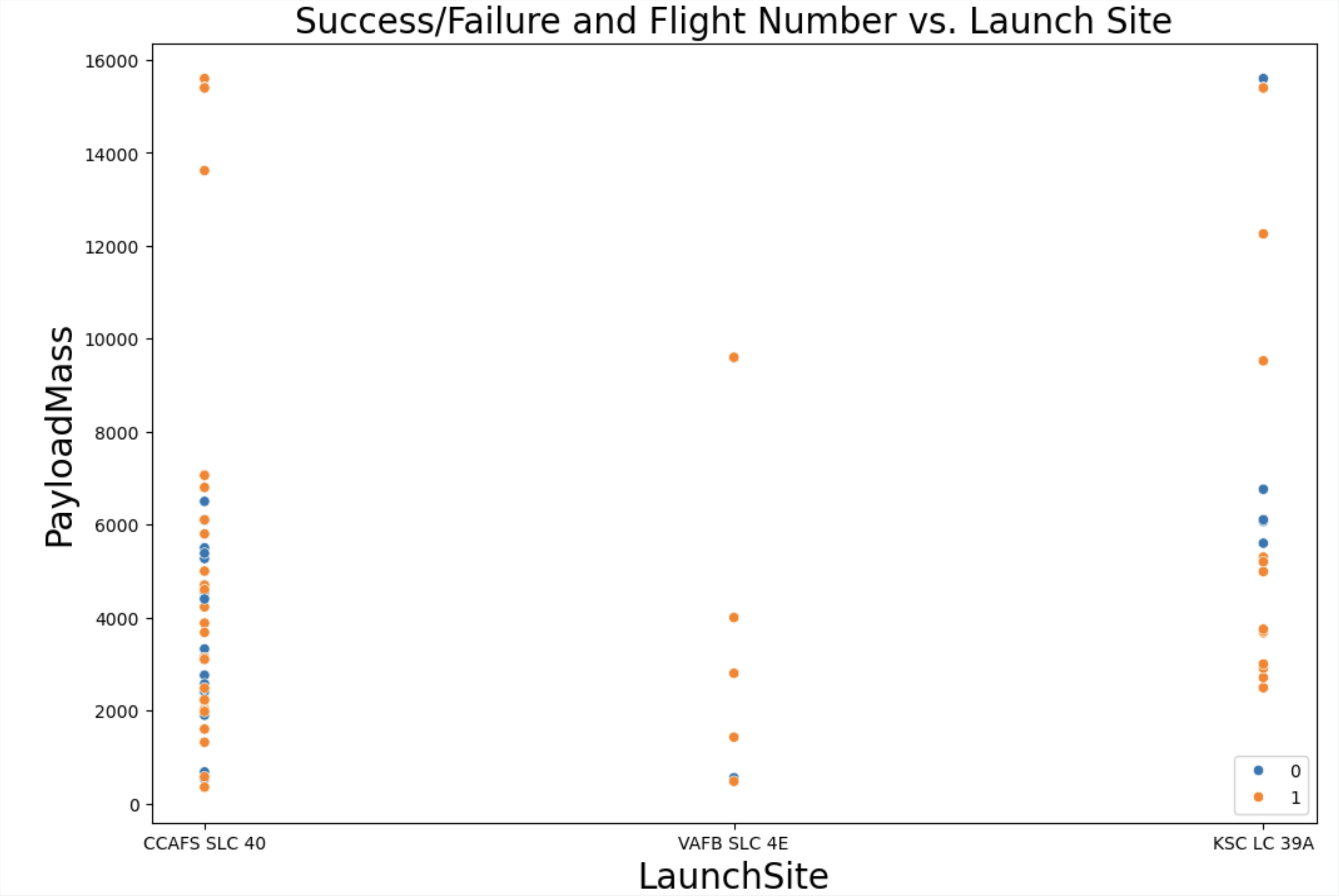
Insights drawn from EDA

Flight Number vs. Launch Site



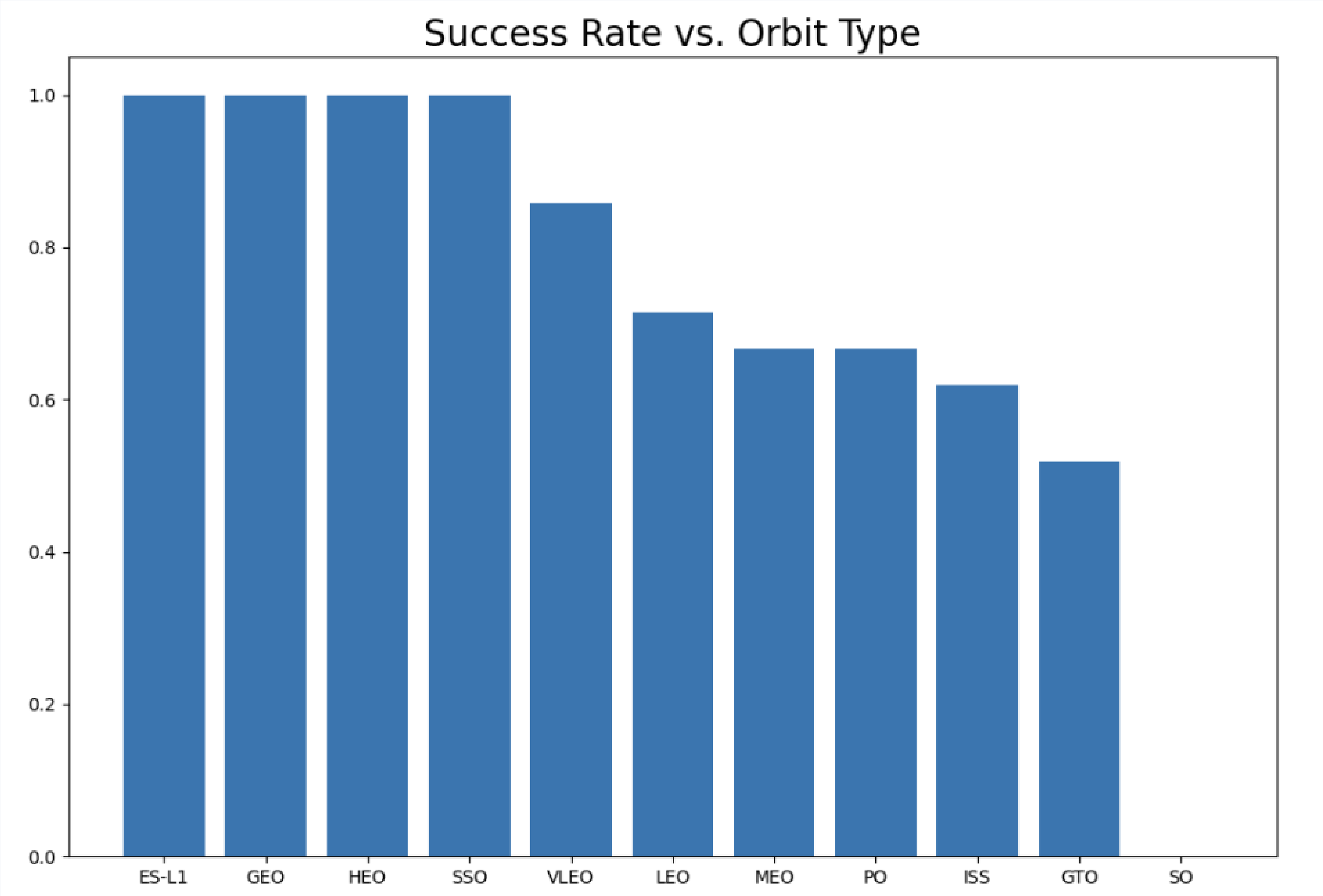
- Success Rate improved significantly with additional flights at each site
- Flights at Kennedy Space Center commenced just before flight number 30

Payload vs. Launch Site



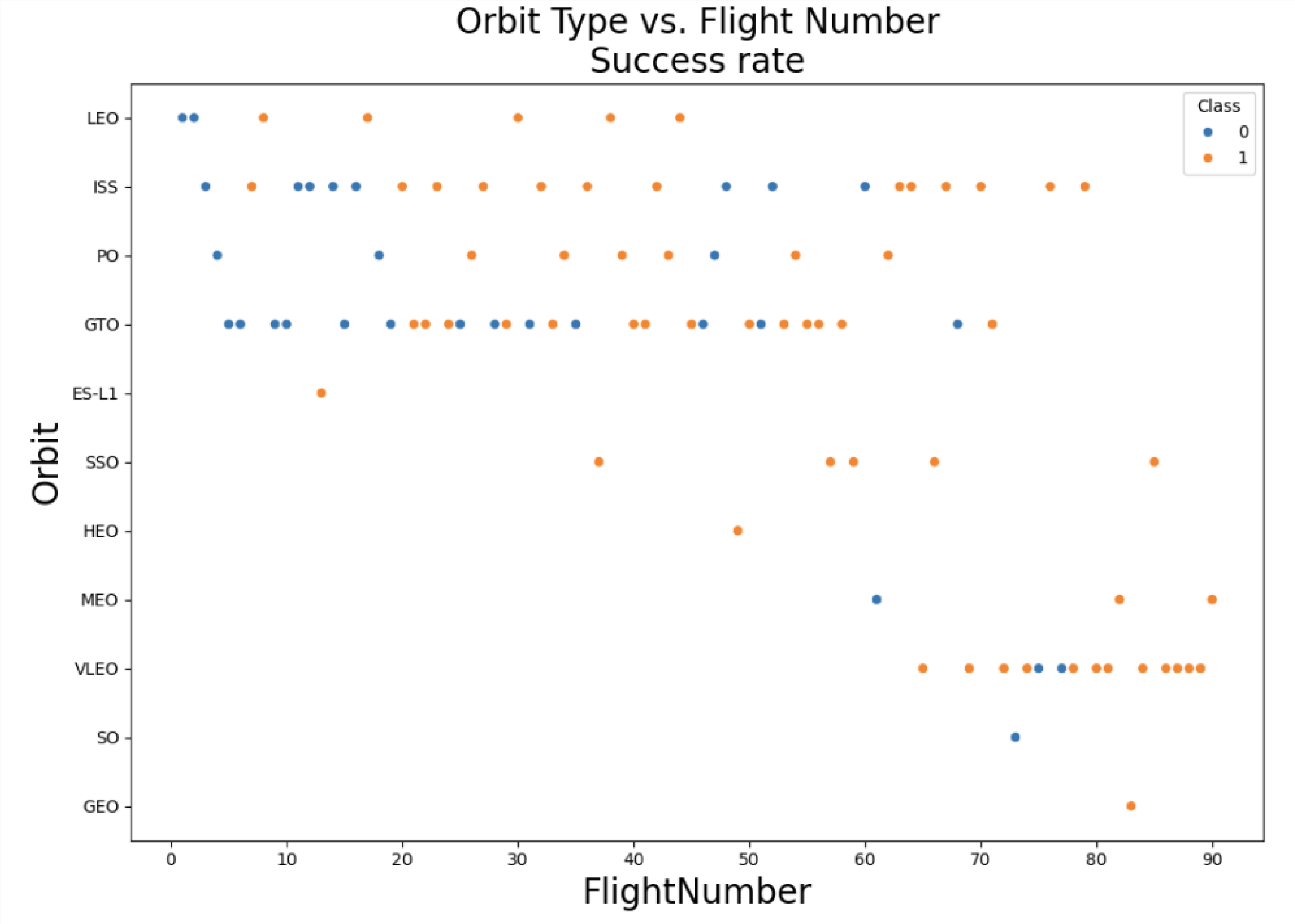
- CCAFS SLC-40 and KSC LC-39A sites have been used to successfully launch and land rockets with payloads greater than 10,000 kg
- VAFB SLC-4E successful but limited to 10,000kg payload.

Success Rate vs. Orbit Type



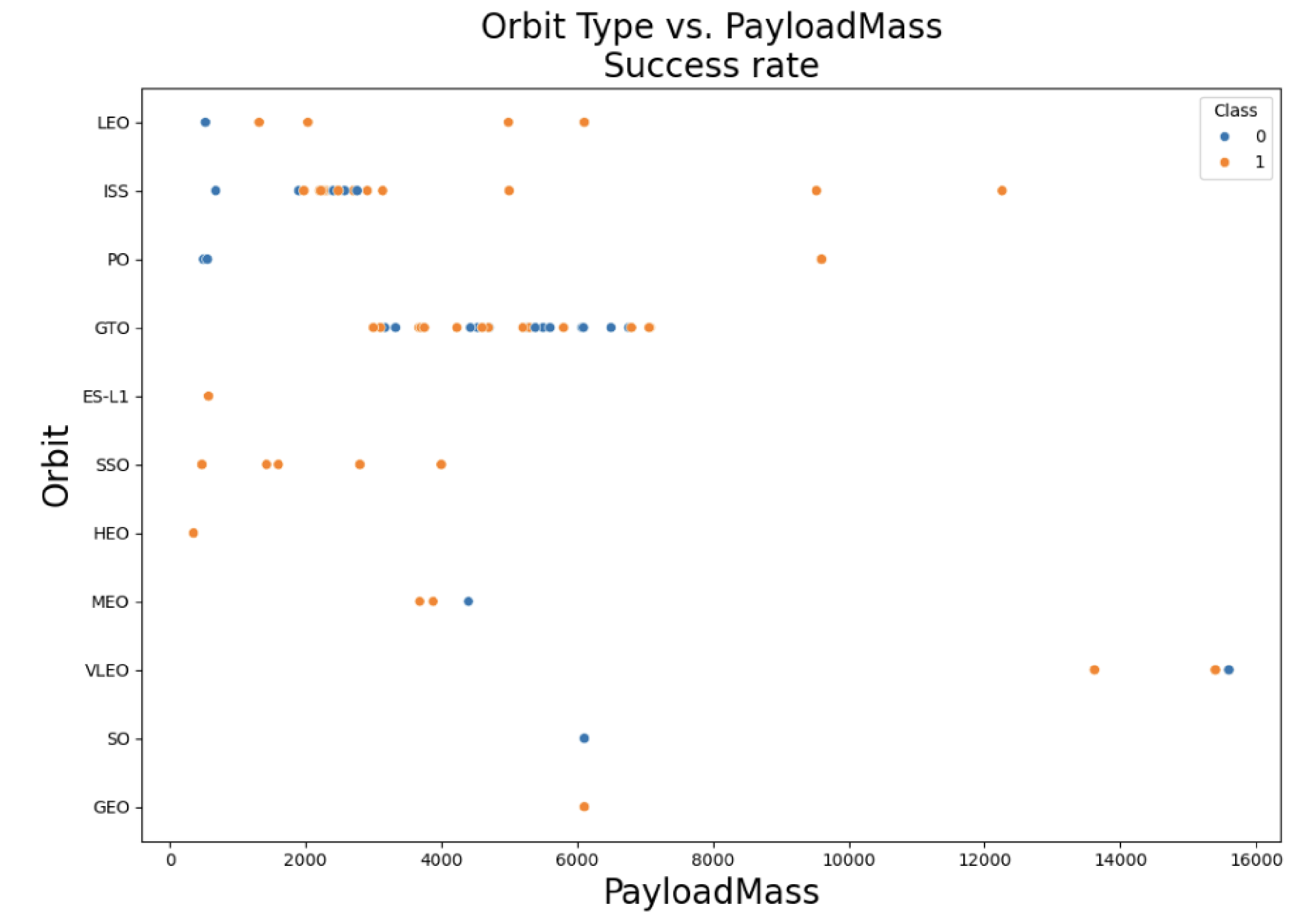
- ES-L1, GEO, HEO and SSO orbits have a perfect landing success rate.
- GTO orbits have the lowest landing success rate of about 50%.

Flight Number vs. Orbit Type



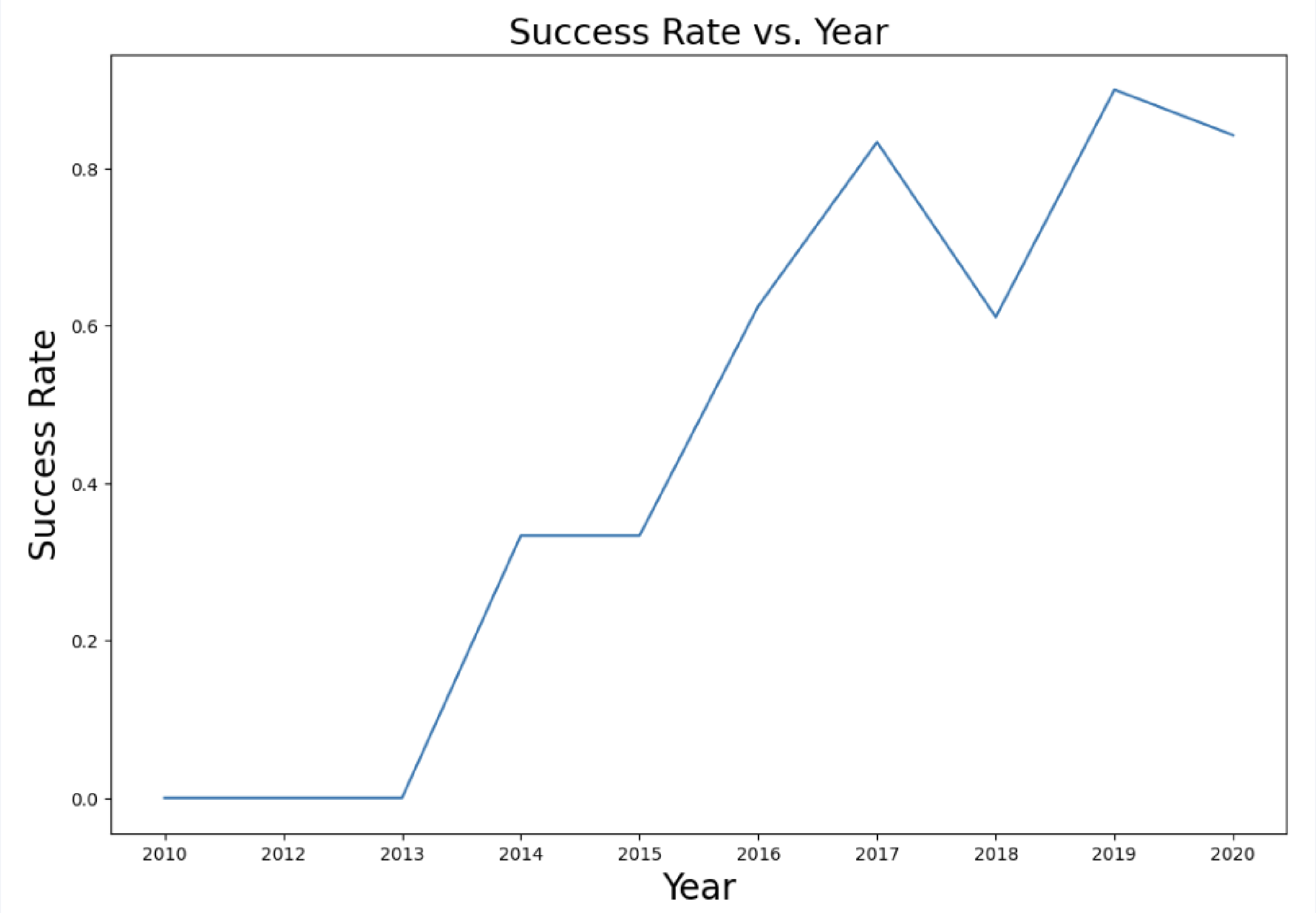
- Initial orbit types were LEO, ISS, PO, GTO, and ES L1.
- Other orbits were attempted after flight number 35.
- Only one(1) GEO orbit and one(1) SO orbit was attempted.

Payload vs. Orbit Type



- Payload mass greater than 8,000 kg were only used for:
  - International Space Station
  - Polar Orbit
  - Very Low Earch Orbits.
- These had high landing success rates.

Launch Success Yearly Trend



- Landing success rate has increased from 2000 to 2020 significantly.
- There was a dip in landing success rate in 2018.

All Launch Site Names

- Unique launch sites
- Florida:
  - Cape Canaveral CCAFS LC-40,
  - Cape Canaveral CCAFS SLC-40,
  - Kennedy Space Center KSC LC-39A
- California:
  - Vandenberg Air Force Base; VAFB SLC-4E
- There are 3 launch sites located near Cape Canaveral, Florida and 1 located North of Las Angeles.

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

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`

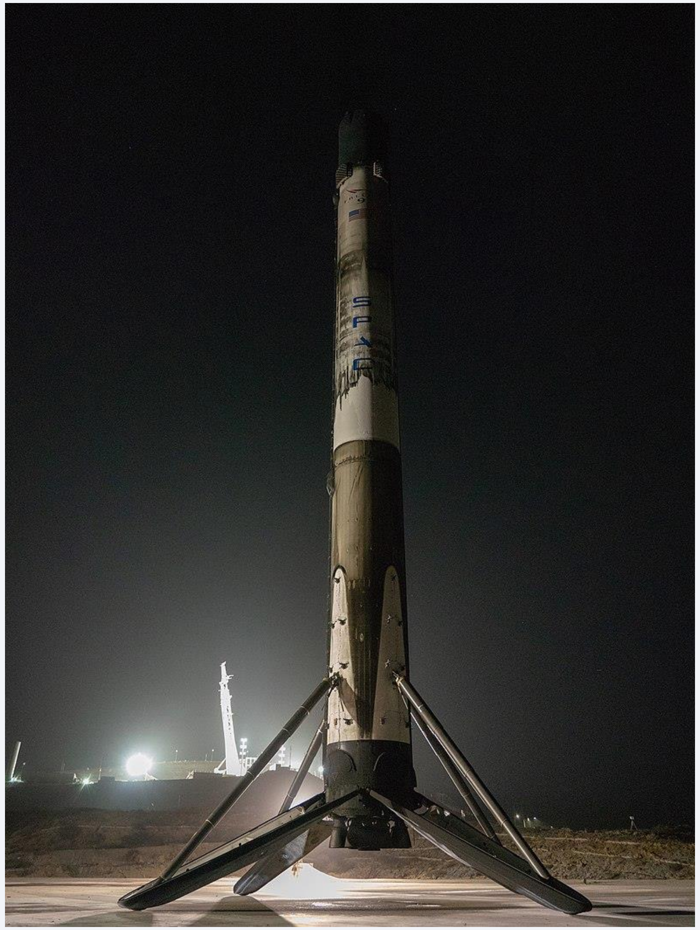
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight 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

- Total payload mass for customer NASA (CRS): 48,213 kg
  - SpaceX has transported the total weight equivalent of a fully loaded semi-truck!

Boosters Carried Maximum Payload

- Booster Version which carried the max payload:
  - F9 B5 B1048.4



Booster Version B1048
Courtesy: Wikipedia

Average Payload Mass by F9 v1.1

- Average payload mass carried by booster version F9 V1.1: 2,535 kg
  - Mass equivalent of a Tesla Model X.

First Successful Ground Landing Date

- First successful landing on a ground pad: Dec. 22, 2015

Successful Drone Ship Landing with Payload between 4000 and 6000

- Boosters landed on a drone ship with payload between 4,000 and 6,000 kg:
  - F9 FT B1020
  - F9 FT B1022
  - F9 FT B1026
  - F9 FT B1021.2
  - F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Total successful and failed missions (not landings):
  - 99 Successful,
  - 1 failed,
  - 1 payload status unclear

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

2015 Launch Records

- Failed Drone Ship Landings in 2015:
  - 2 Launched from Cape Canaveral CCAFS LC-40 with v1.1 Boosters

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

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

- Ranking the count of landing outcomes.
- Landing Outcomes between 2010-06-04 and 2017-03-20:

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

Section 3

Launch Sites Proximities Analysis

Falcon 9 Launch Sites

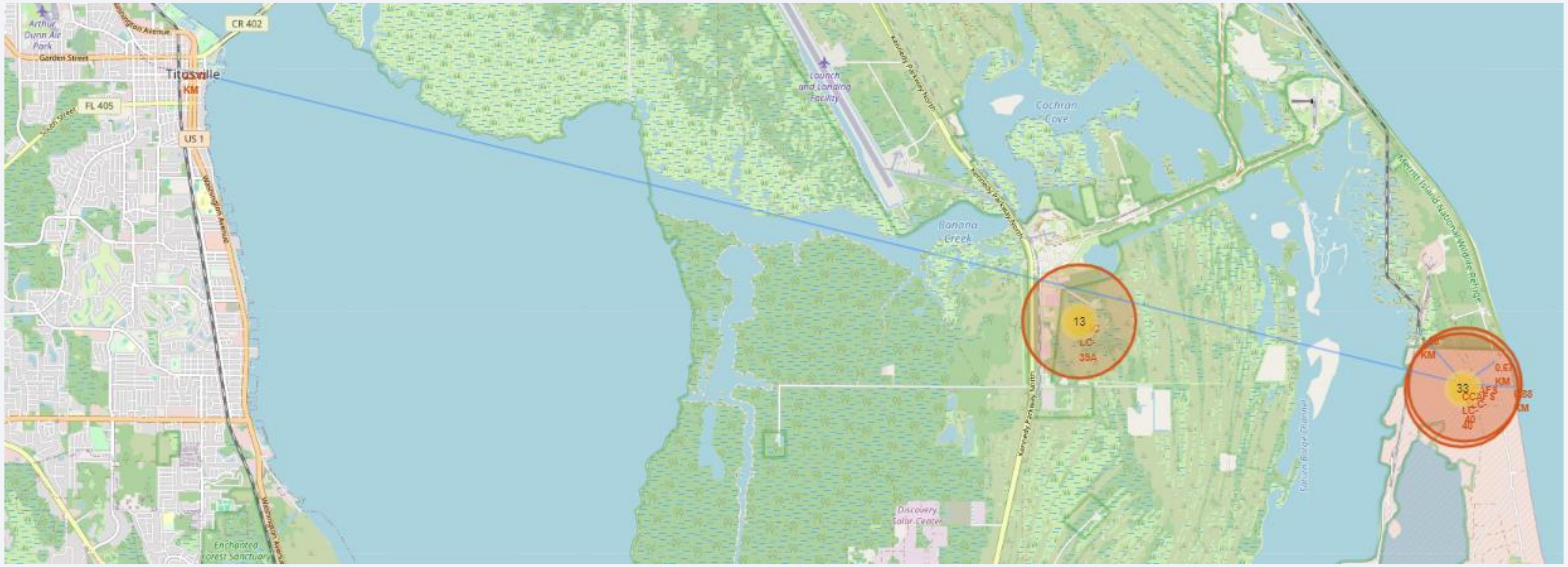
Falcon 9 Launch Sites (Vandenberg AFS, Cape Canaveral and Kennedy Space Center

Florida Launch Sites (Zoomed):
Kennedy Space Center KSC LC 35A,
Cape Canaveral CCAFS SLC-40, LC-40

Landing Successes and Failures per Site

- Launch landing successes are shown in green; failures in red.

Launch Sites, Distance to Highway, Rail and City



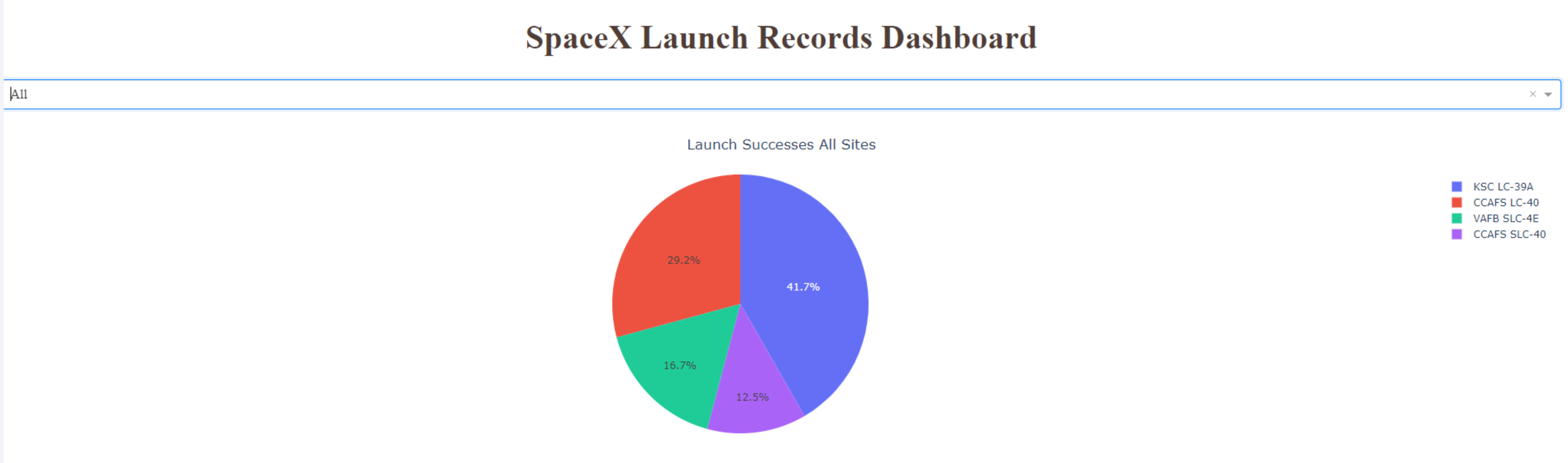
Distance to:

- Coastline: 0.9 km
- Rail: 1.2 km
- Highway: 0.67 km
- City: 23.4 km

Section 4

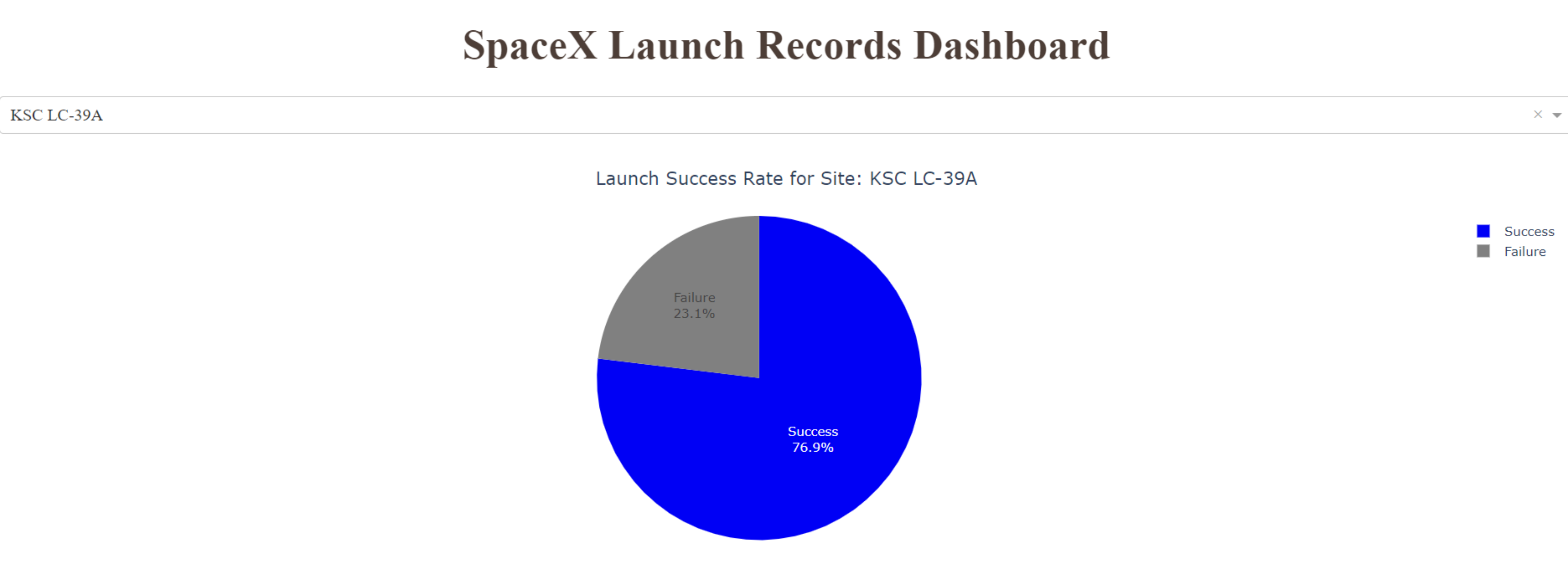
Build a Dashboard with Plotly Dash

Launch Successes: All Sites



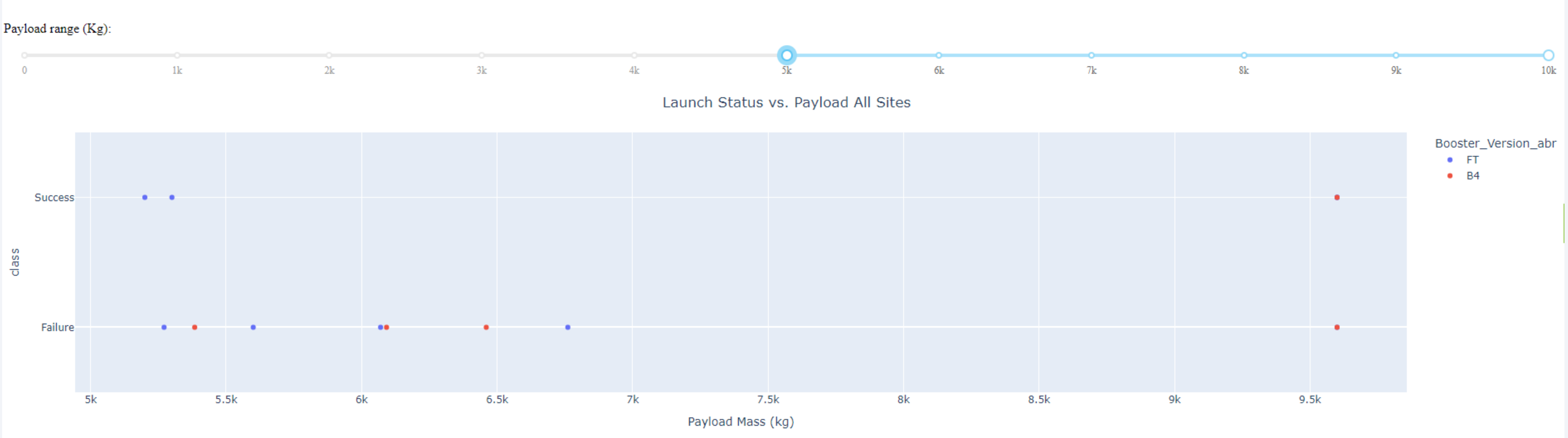
- Kennedy Space Center LC-49 has the highest number of launch successes (10), followed by CCAFS LC-40 (7), VAFB SLC-4E (4) and then CCAFS SLC-40 (3).

Launch Site with Highest Success Rate



- Kennedy Space Center has 10 successful landings, 3 failed landings

Launch Status vs. Payload All Sites (5,000 – 10,000 kg)

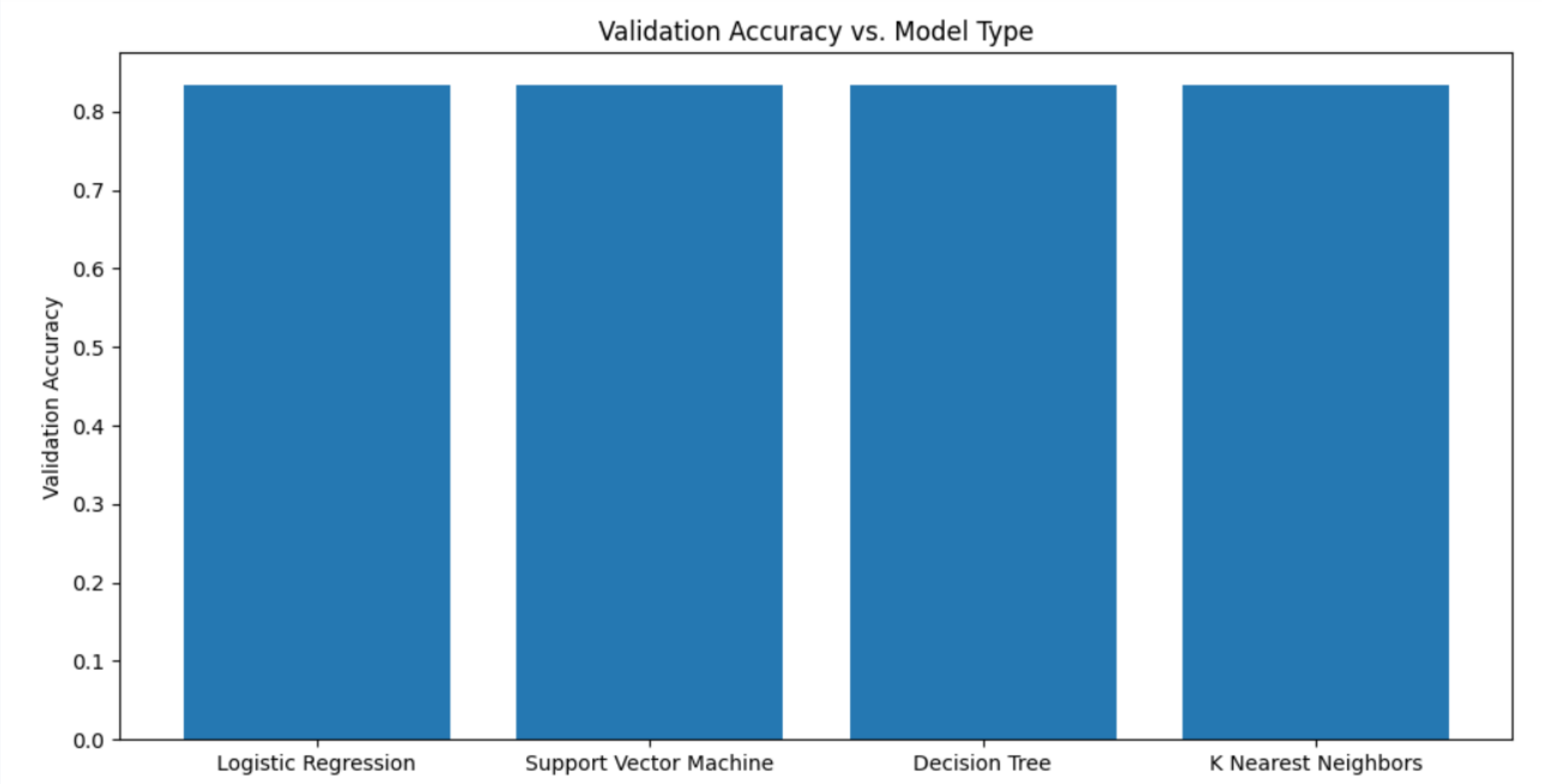


- Payload Range: 5,000 – 10,000 kg
- Booster with the most successful landings: FT
- Booster with the most failed landings: v1.1

Section 5

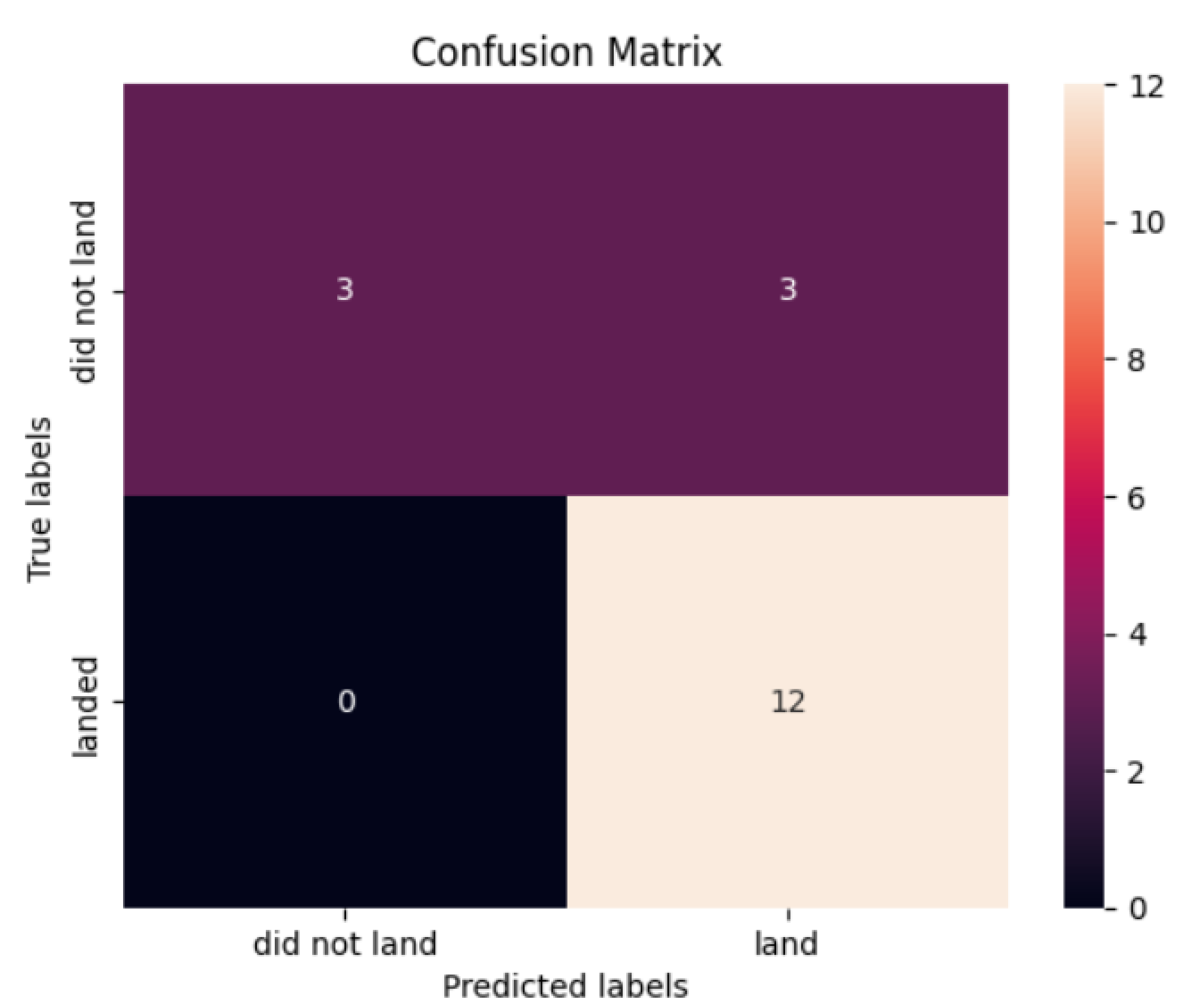
Predictive Analysis (Classification)

Classification Accuracy



- Each of the models produced a similar validation accuracy of 0.83
- However, the Logistic Regression model was selected because it has the ability to predict probability associated with a classification.

Confusion Matrix (Logistic Regression)



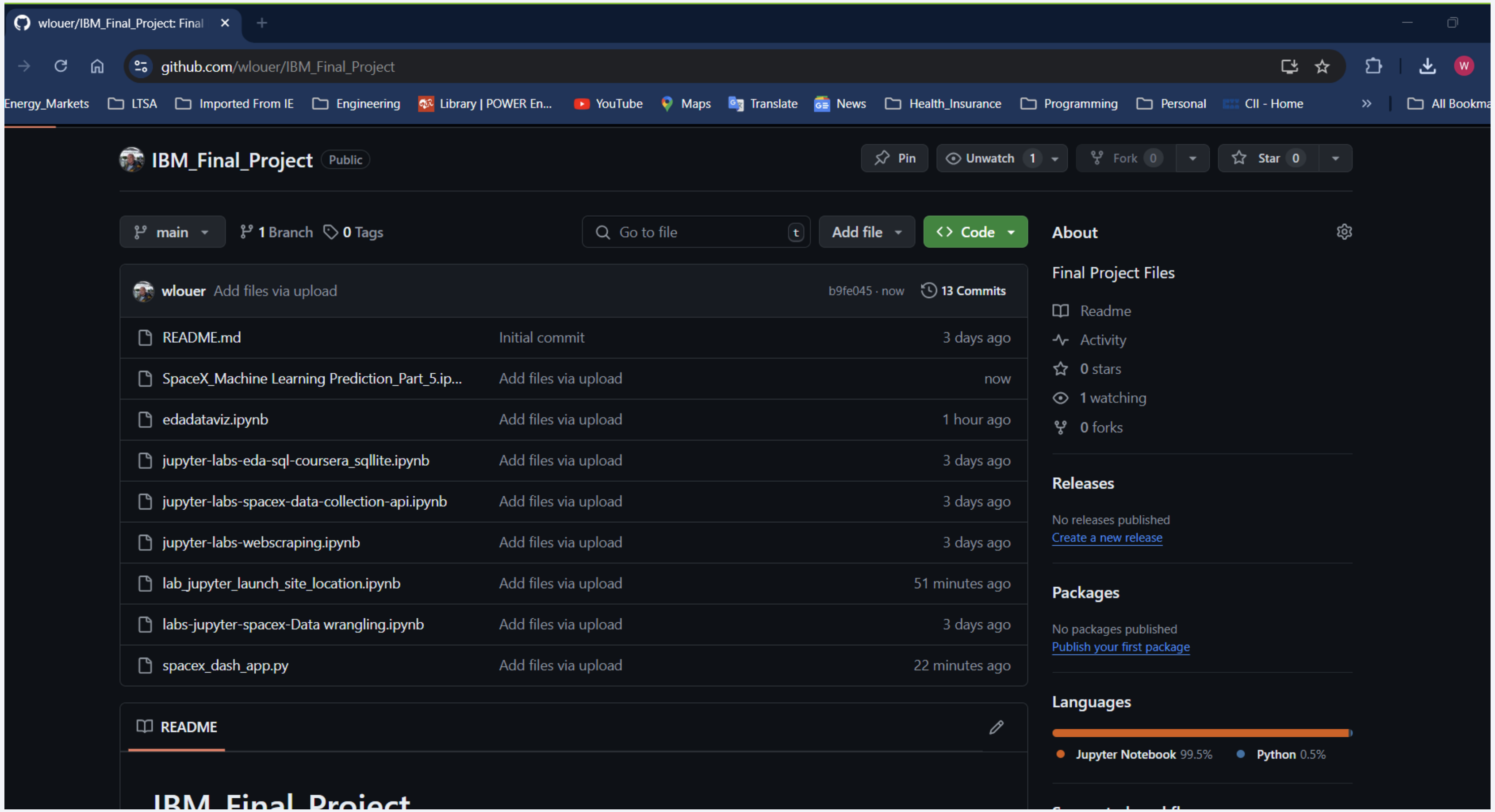
- Each of the models has the same confusion matrix:
- Model predicted all 12 successful landings.
- Model predicted 3 failures correctly.
- Model predicted 3 false positives.

Conclusions

- SpaceX launches from 4 locations (3 FL, 1 CA)
- Each location was selected for proximity to highway, railway, the ocean and away from large cities.
- SpaceX has improved its landing success over time. KSC has the highest success rate.
- Models were developed for KNN, Decision Tree, Support Vector Machines and Logistic Regression
- All models demonstrated a validation accuracy of 83%
- A logistic regression model was selected for its ability to predict classification probability

Appendix

References: GitHub Repository (https://github.com/wlouer/IBM_Final_Project)



Thank you!