IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

K Fitz

May 24, 2024

Outline

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

Executive Summary

This analysis methodology was undertaken through data collection (API and web scraping), data wrangling, exploratory data analysis (SQL and data visualization), Folium interaction visual analytics and machine learning prediction.

A summary of results is provided through Interactive and predictive analytics.

Introduction

SpaceX has a significant competitive advantage through its Falcon 9 first stage reuse strategy, driving down the typical provider launch cost by over 100 million US dollars. Our goal of this data analytics project is to generate a machine learning pipeline to predict first stage landing success by determining:

- Landing success factors;
- Interaction of rocket features and landing performance; and
- Optimal operating conditions

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - API collection and web-scraping through SpaceX REST API and wiki pages.
- Perform data wrangling
  - Data digested from JSON objects and html tables into pandas dataframe for visualization and analysis. One hot encoding applied to categorical features for regression analysis.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Machine learning / Falcon 9 first stage performance

Data Collection

- SpaceX Rest API

SpaceX Rest API through python request

Json decoding of response, normalize into pandas dataframe

Data hygiene, clean missing values and set data types

- Wiki of historic rocketry performance

Falcon 9 Performance pages on Wiki

Python download and Beatifulsoup to parse html table information

Conversion into pandas dataframe

Data Collection – SpaceX API

SpaceX Rest API through python get request

Collect JSON values: Booster, Payload, Launch and Landing sites

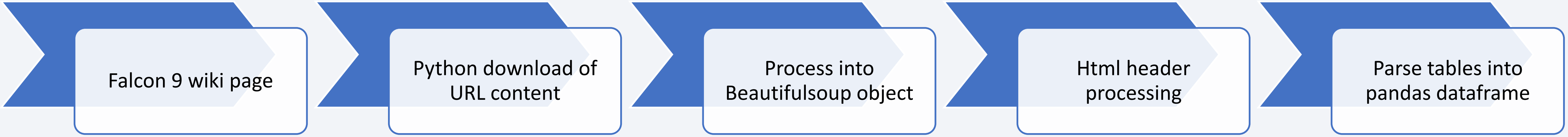
Convert json to pandas dataframe

Isolate only Falcon 9 launch information

Update missing values

- GitHub URL of the completed SpaceX API calls notebook

Data Collection - Scraping



- GitHub URL of the completed web scraping notebook

Data Wrangling

Identify data labels

Interim calculations (site launches, orbit occurrences, mission success per orbit type)

Form dependent variable (landing outcome)

Csv file creation for further processing

- [GitHub URL of the completed data wrangling notebook]

EDA with Data Visualization

Charts:

- Launch site vs Flight number
- Payload (Mass in kg) vs
  - Flight number
  - Launch site
  - Orbit type

Scatter plots used to show relationships and trends;

Bar charts to compare categorical data and quantities.

- GitHub URL of the completed EDA with data visualization notebook

EDA with SQL

- SQL Queries performed:
    - Names of unique launch sites
    - Records with launch sites beginning ‘CCA’
    - Total mass payload (kg) of boosters launched by NASA (CRS)
    - F9 v1.1 average mass payload (kg)
    - Date of first successful ground pad landing
    - Name boosters with successful drone ship landings and mass between 4000 and 6000 kg
    - In 2015, all booster versions, launch sites with failed drone ship landings by month
    - Count of landing outcomes between June 4, 2010 and March 20, 2017 (descending)
- GitHub URL of the completed EDA with SQL notebook

Build an Interactive Map with Folium

- Launch site markers
  - Blue circle at NASA Johnson space centre to show name, lat/long coordinates
  - Red circle at all launch sites to show name, lat/long coordinates
- Launch outcomes
  - Successful (green) and failed (red) launches at each launch site to observe patterns
- Launch site distance to key geographic and urban features
  - Coloured lines to show CCAFS SLC-40 proximity to coastline, railway, city, highway

- GitHub URL of your completed interactive map with Folium map

Build a Dashboard with Plotly Dash

- Dashboard plots/graphs and interactions
  - Dropdown list with launch sites to observe success rates (pie chart)
  - Slide bar to view success rates by launch site with varying payload sizes (kg) (scatter plot)

- GitHub URL of your completed Plotly Dash lab
- Working version URL shared in google colab (click play icon)

Predictive Analysis (Classification)

- Steps:

NumPy array for class column (success fail)

Standarscalar to standardize the data

train_test_split to create training array

GridSearchCV object for parameter optimization (cv=10)

GridSearchCV algos: LogisticRegression(), SVC(), DecisionTreeClassifier(), KNeighborsClassifier()

.score for all models to calc accuracy

Confusion matrix review

Jaccard_Score, F1_Score, Accuracy

= Best model determination

- [GitHub URL of the completed predictive analysis lab]()

Results

- Exploratory Data Analysis
  - Improvement in launch success can be observed over time
  - Landing at the LSC LC-39A site has greatest success rate at 76.9%
  - Launches into Orbits ESL1, GEO ,HEO, SSO continue to have 100% success
- Visual Analytics
  - Launches occur on the coasts, near the equator
  - Launch sites have good proximity for logistics (people, resources), while maintaining good distance from sensitive areas (airports, highways, urban centres)
- Predictive Analysis
  - Decision tree model found to be the best predictive model

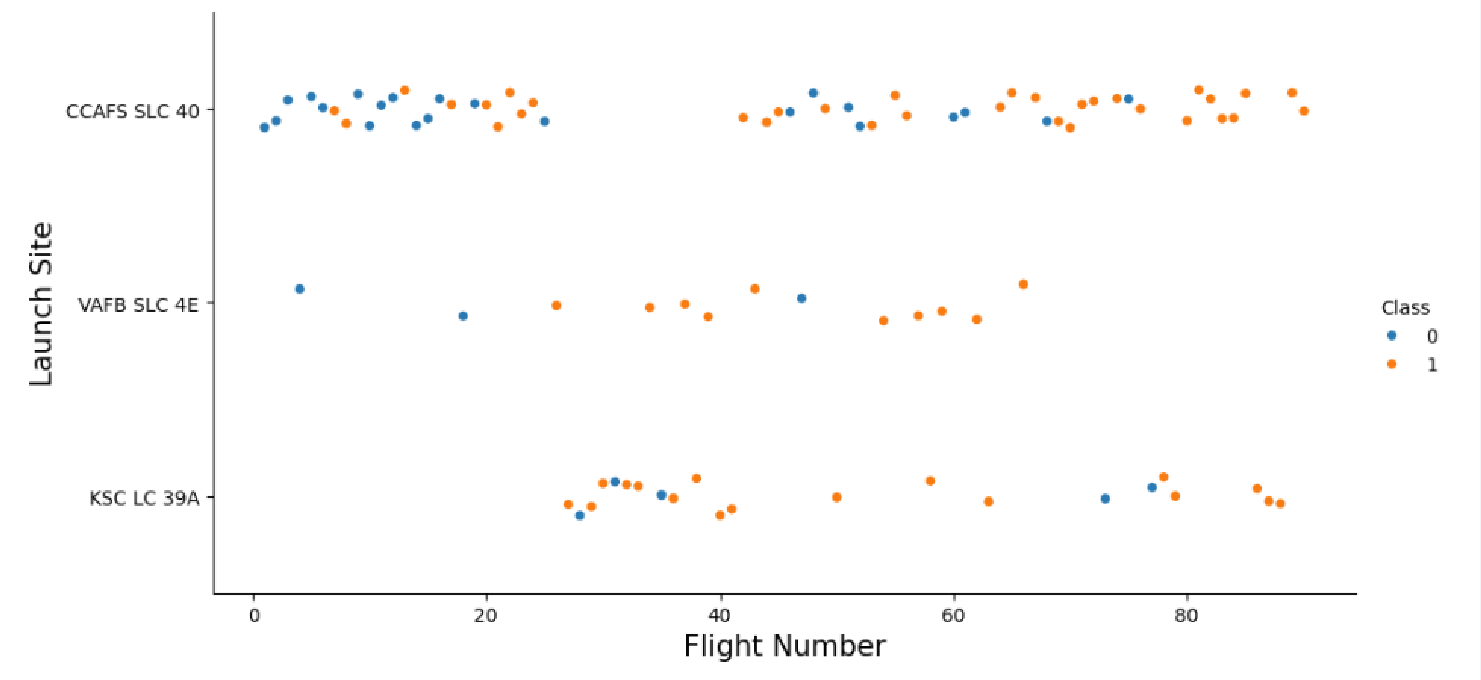
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

Observations:

- Recent flights (higher flight number) have greater success
- ‘40 has far majority of launches
- ‘4E and ‘39A have higher number of successful launches



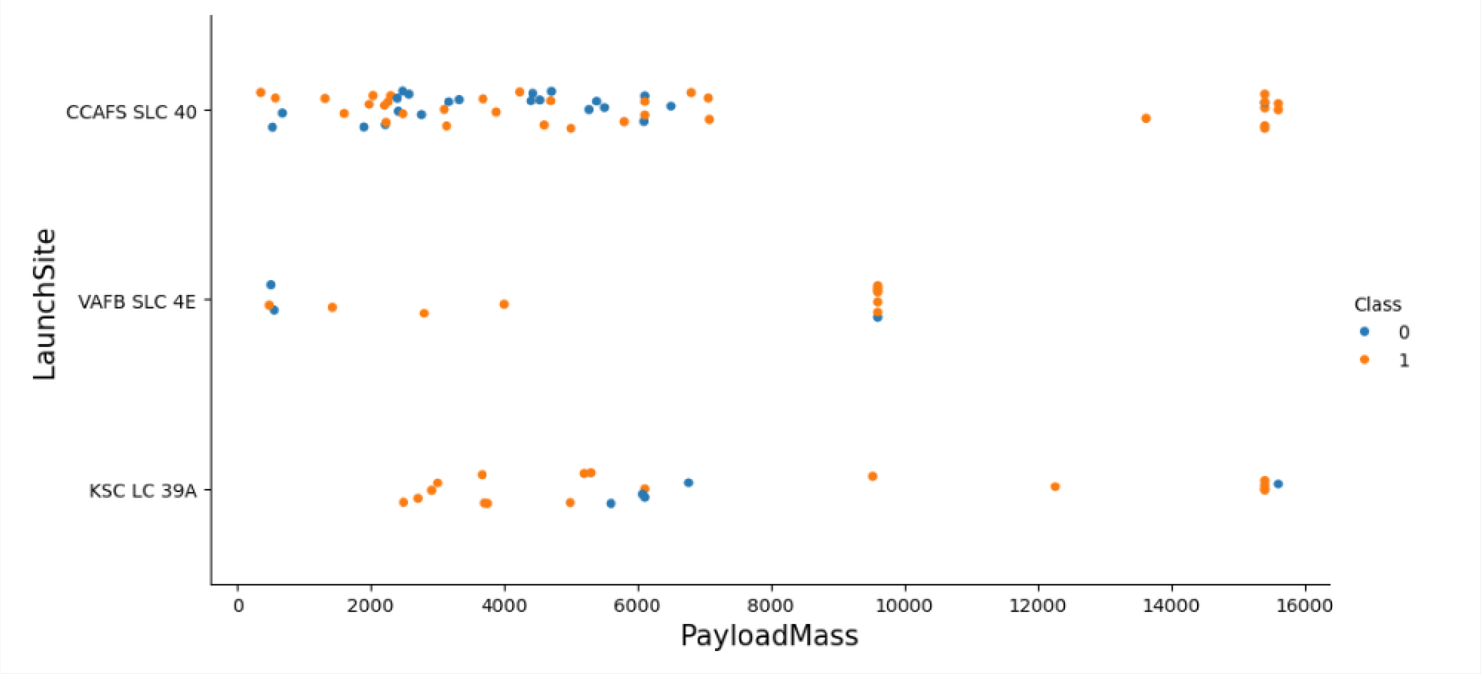
1 Success

0 Failure

Payload vs. Launch Site

Observations:

- Success increases with Payload size
- ‘4E has no payloads over 10000 kg
- 100% success rate with site ’40 at higher Payloads (>12000 kg)



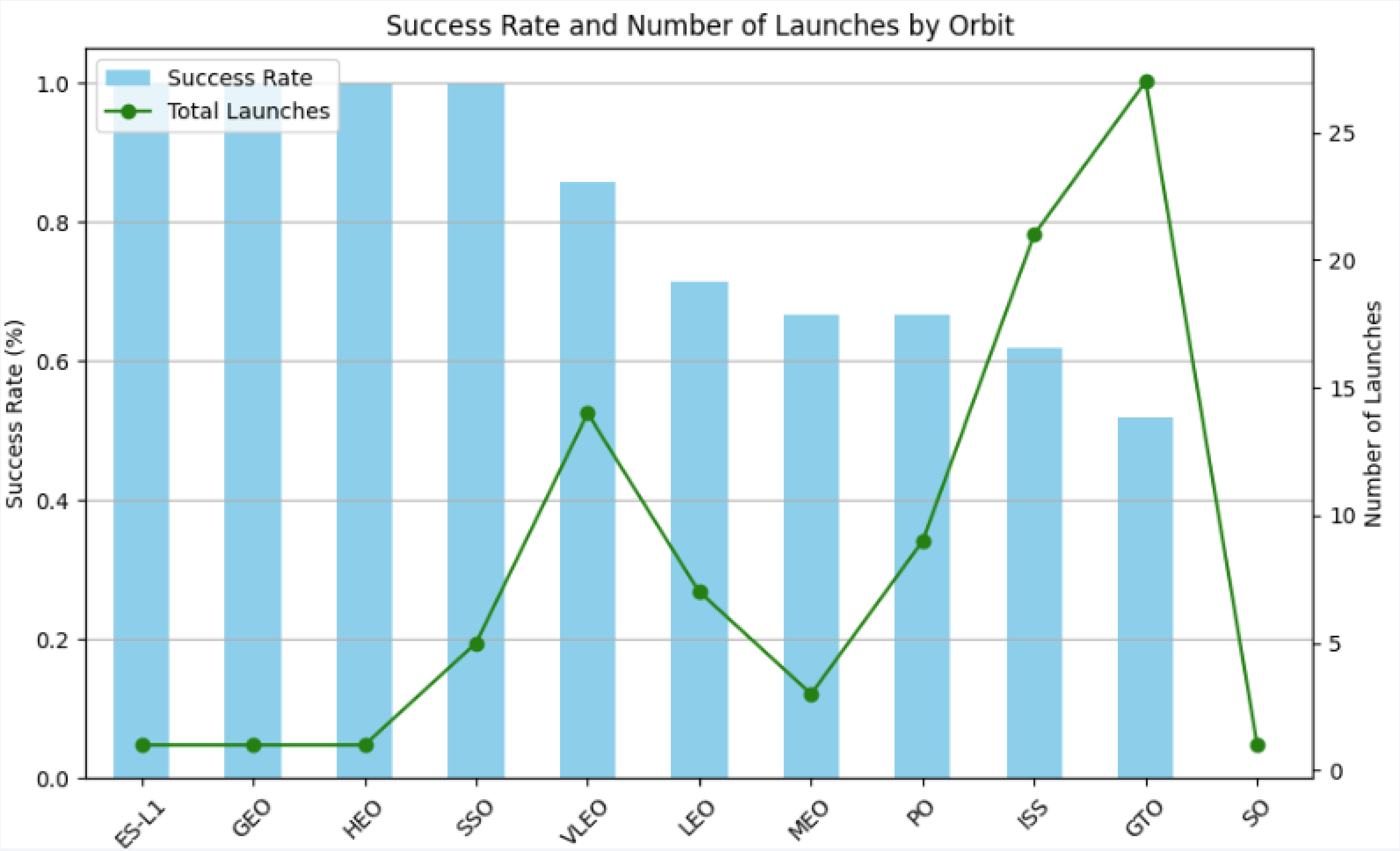
1 Success

0 Failure

Success Rate vs. Orbit Type

Observations:

- ES-L1, GEO, HEO, SSO have perfect launch records (note only SSO has more than 1 launch)
- SO has zero successes (note only 1 launch)



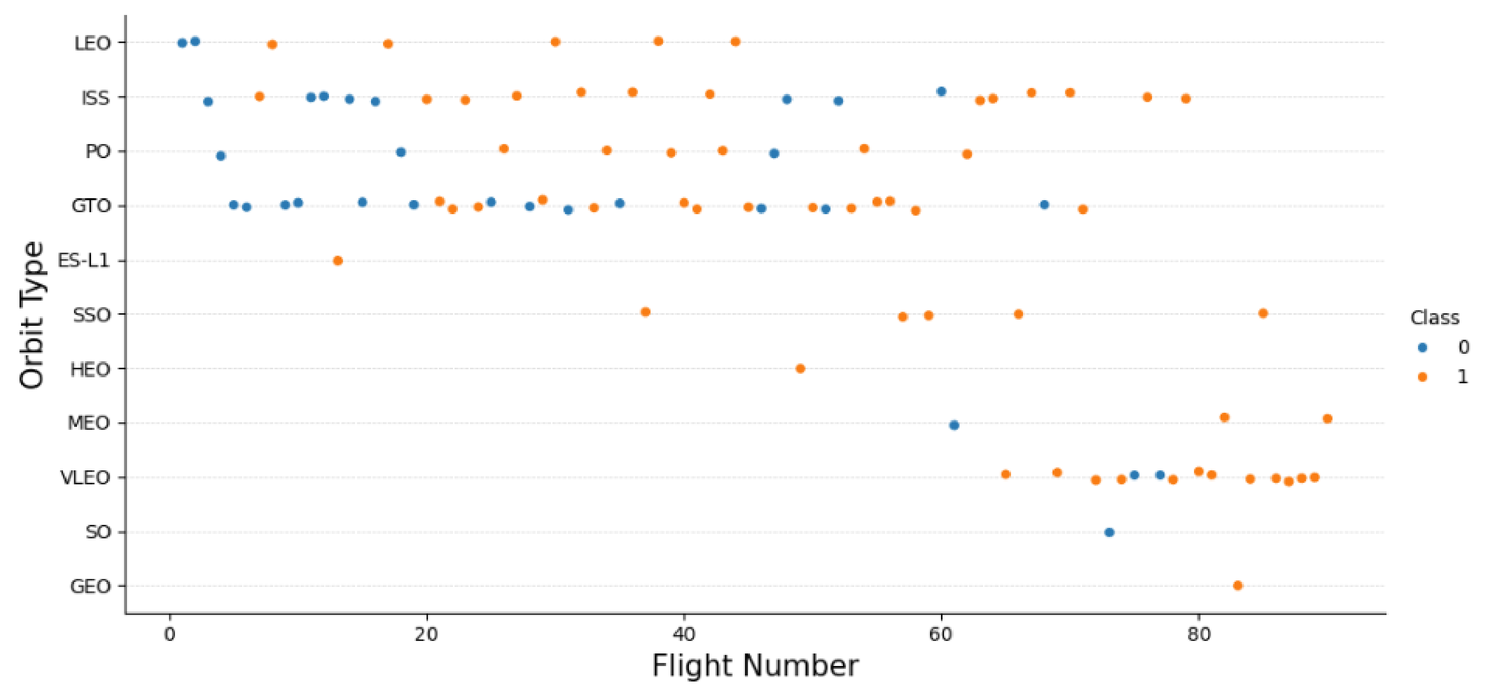
Note:

Added secondary y-axis to improve level of info

Flight Number vs. Orbit Type

Observations:

- Successful launches tend to increase with number of flights, except GTO does not have this pattern
- As observed in prior slide, sites with fewer launches have higher success



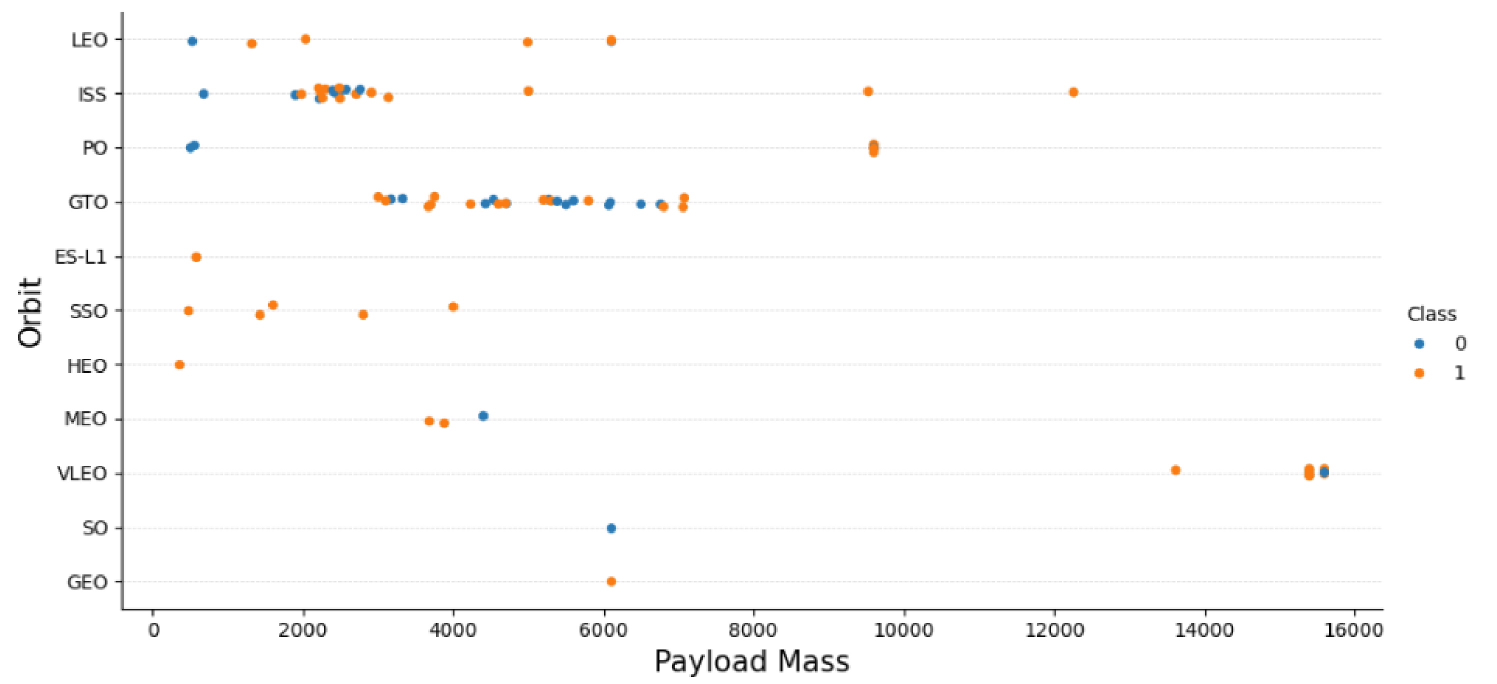
1 Success

0 Failure

Payload vs. Orbit Type

Observations:

- Heavy payload launches have greater success with LEO, ISS, and PO orbits
- Success does not seem to vary with different payloads in GTO orbit



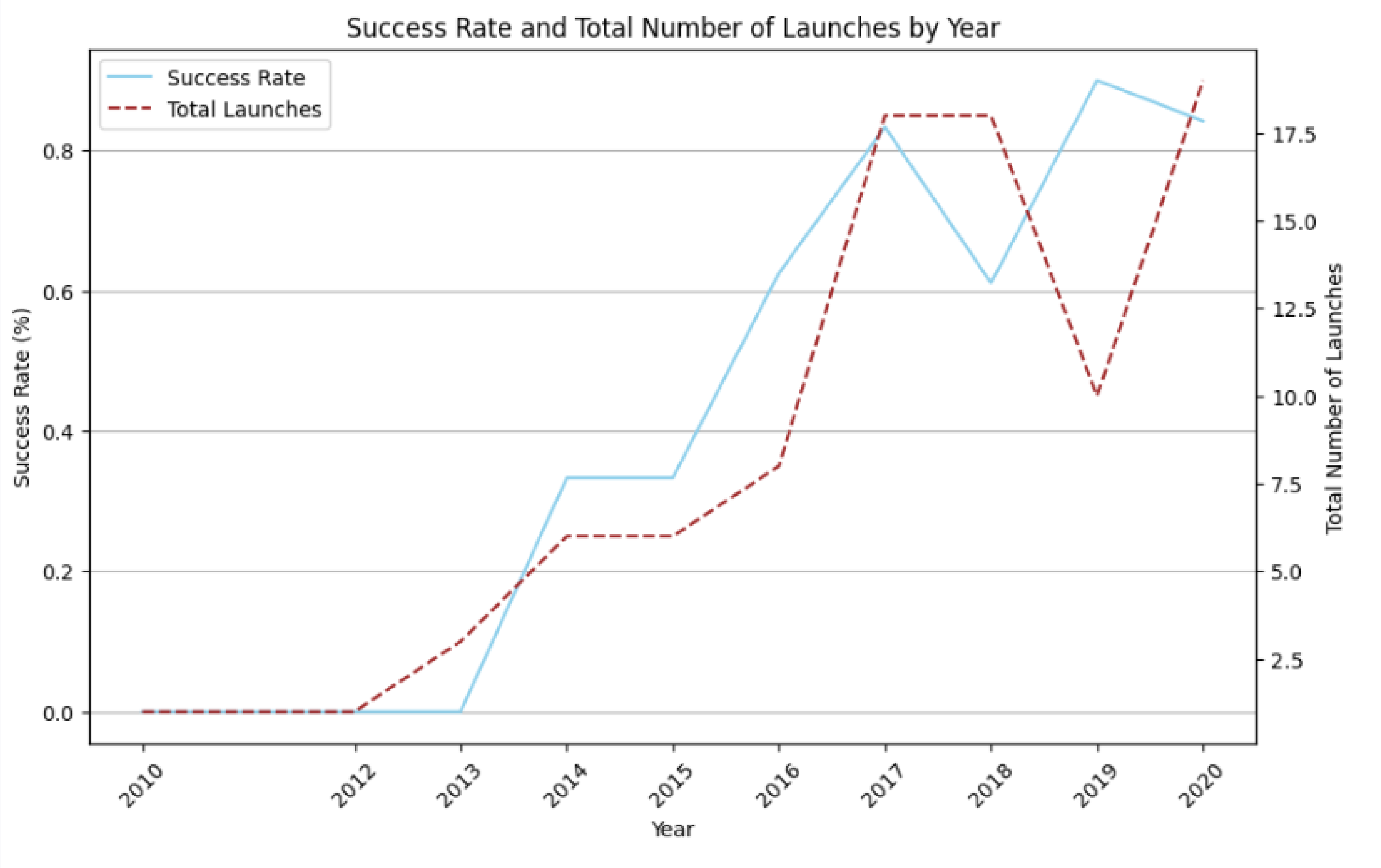
1 Success

0 Failure

Launch Success Yearly Trend

Observations:

- Average Launch Success increases over time, with exception of 2018
- A reduction in the number of launches in 2019 follows the poor launch success in 2018



Note:

Added # of launches to secondary y-axis for additional information

All Launch Site Names

- Names of the unique launch sites

```
[13]: %sql select distinct (Launch_Site) from SPACEXTABLE
```

```
 * sqlite:///my_data1.db
Done.
```

[13]:

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`

```
[12]: %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5
 * sqlite:///my_data1.db
Done.
```

[12]:

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 carried by boosters from NASA

Display the total payload mass carried by boosters launched by NASA (CRS)

[13]:
```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer = "NASA (CRS)"
```

```
 * sqlite:///my_data1.db
Done.
```

[13]:

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

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

```
[14]: %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version like "F9 v1.1%"

 * sqlite:///my_data1.db
Done.
```

[14]:

avg(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

- Date of the first successful landing outcome on ground pad

```
[15]: %sql select min(Date) from SPACEXTABLE where Mission_Outcome = "Success" and Landing_Outcome like "%(ground pad)"

 * sqlite:///my_data1.db
Done.
```

[15]:

min(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
[16]: %sql select Booster_version from SPACEXTABLE where Mission_Outcome = "Success" and Landing_Outcome like "%(ground pad)" and (PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000)

 * sqlite:///my_data1.db
Done.
```

[16]:

Booster_Version
F9 FT B1032.1
F9 B4 B1040.1

Total Number of Successful and Failure Mission Outcomes

- Total number of successful and failure mission outcomes

```
[17]: # Use %sql magic to execute SQL queries and assign results to variables
success = %sql SELECT COUNT(*) FROM SPACEXTABLE WHERE Mission_Outcome = "Success"
failure = %sql SELECT COUNT(*) FROM SPACEXTABLE WHERE Mission_Outcome != "Success"
print(f"Successes {success} \nand Failures {failure}")
```

```
 * sqlite:///my_data1.db
Done.
 * sqlite:///my_data1.db
Done.
Successes +----------+
| COUNT(*) |
+----------+
|    98    |
+----------+
and Failures +----------+
| COUNT(*) |
+----------+
|    3     |
+----------+
```

Boosters Carried Maximum Payload

- Names of the booster which have carried the maximum payload mass

```
[18]: %sql select distinct (Booster_version) from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE) order by Booster_Version

 * sqlite:///my_data1.db
Done.
```

[18]:

Booster_Version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

- Failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

[19]:
```
%sql SELECT CASE WHEN strftime('%m', Date) = '01' THEN 'January' WHEN \
strftime('%m', Date) = '02' THEN 'February' WHEN strftime('%m', Date) = '03' \
THEN 'March' WHEN strftime('%m', Date) = '04' THEN 'April' WHEN strftime('%m', Date) = '05' THEN 'May' WHEN strftime('%m', Date) = '06' THEN 'June' \
WHEN strftime('%m', Date) = '07' THEN 'July' WHEN strftime('%m', Date) = '08' THEN 'August' WHEN strftime('%m', Date) = '09' THEN 'September' \
WHEN strftime('%m', Date) = '10' THEN 'October' WHEN strftime('%m', Date) = '11' THEN 'November' WHEN strftime('%m', Date) = '12' THEN 'December' \
END AS Month, Booster_Version, Launch_Site, Landing_Outcome FROM SPACEXTABLE WHERE Date LIKE '2015%' ORDER BY Date, Landing_Outcome
```

```
 * sqlite:///my_data1.db
Done.
```

[19]:

Month	Booster_Version	Launch_Site	Landing_Outcome
January	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
February	F9 v1.1 B1013	CCAFS LC-40	Controlled (ocean)
March	F9 v1.1 B1014	CCAFS LC-40	No attempt
April	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)
April	F9 v1.1 B1016	CCAFS LC-40	No attempt
June	F9 v1.1 B1018	CCAFS LC-40	Precluded (drone ship)
December	F9 FT B1019	CCAFS LC-40	Success (ground pad)

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

- Ranked 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

```
[20]: %sql select count(landing_outcome) as "Number", landing_outcome from SPACEXTABLE where Date >= '2010-06-04' and Date <= '2017-03-20' group by landing_outcome order by "Number" desc

 * sqlite:///my_data1.db
Done.
```

[20]:

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

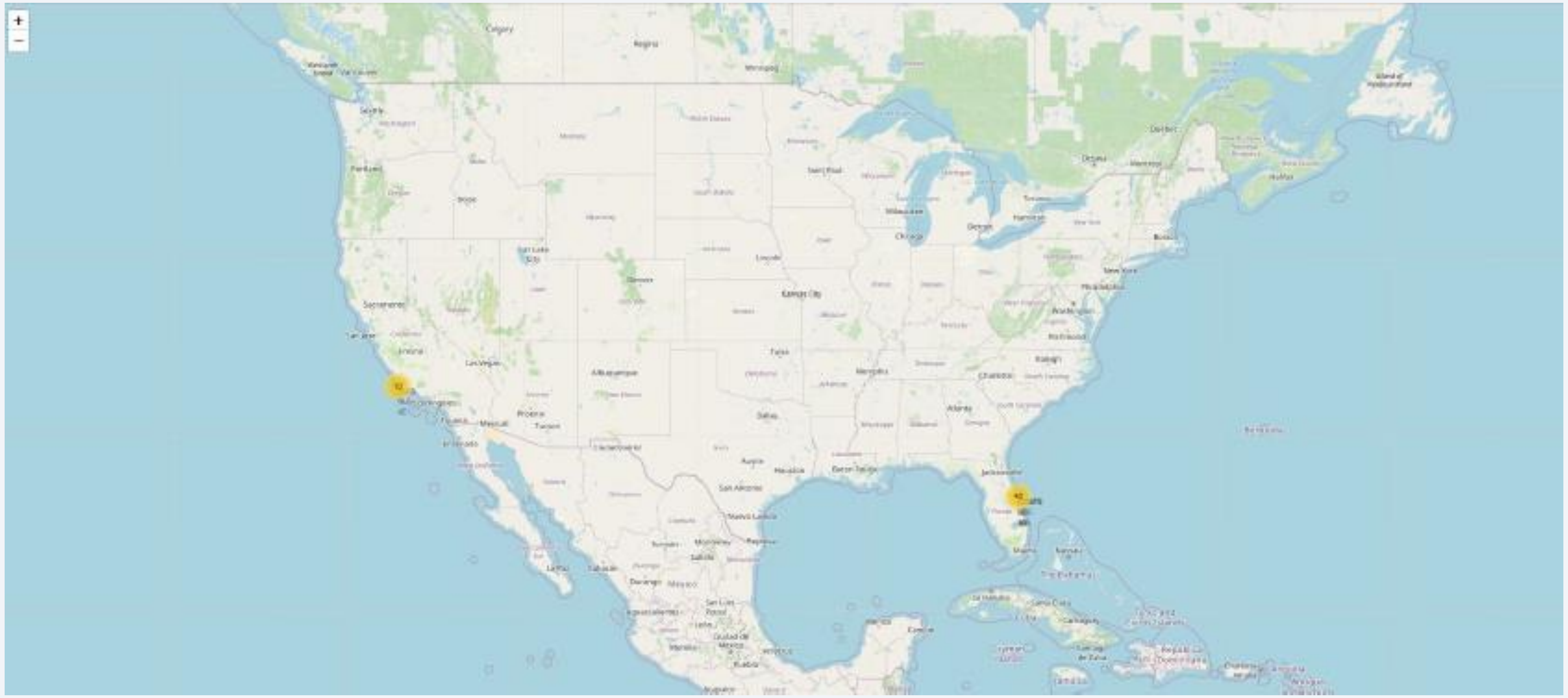
Section 3

Launch Sites Proximities Analysis

Launch Sites, Coastal near Equator

- Launch sites near the equator have an acceleration benefit from the rotation of the earth, adding ‘free’ velocity

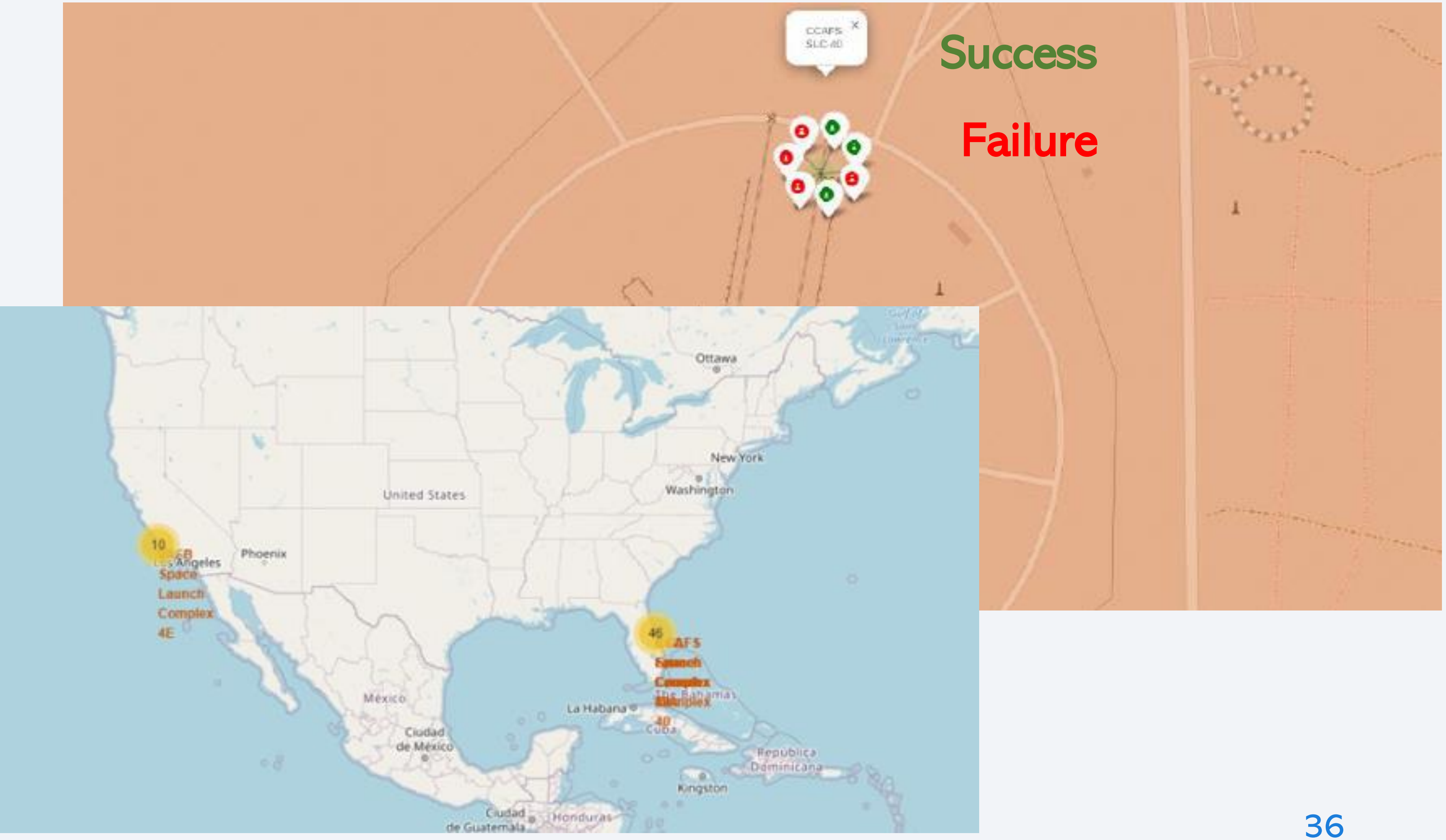
- Launch sites near coast in restricted areas to reduce risk to population ground areas and other flight activity



Site Launch Locations and Performance

- 3/7 success performance for CCAFS-SLC-40 site

- Yellow circle represents cluster of launch sites



Proximity to landmarks and features

- Launch sites on coast, at safe Distances from sensitive areas such as airports, urban centres, railways

Section 4

Build a Dashboard with Plotly Dash

Relative launch success

SpaceX Launch Records Dashboard

Select Launch Site:

All Sites

Success Counts by Launch Site

Findings:

- ‘39A (41.7%) has the highest share of success across the 4 noted sites

Highest launch success ratio

SpaceX Launch Records Dashboard

Select Launch Site:

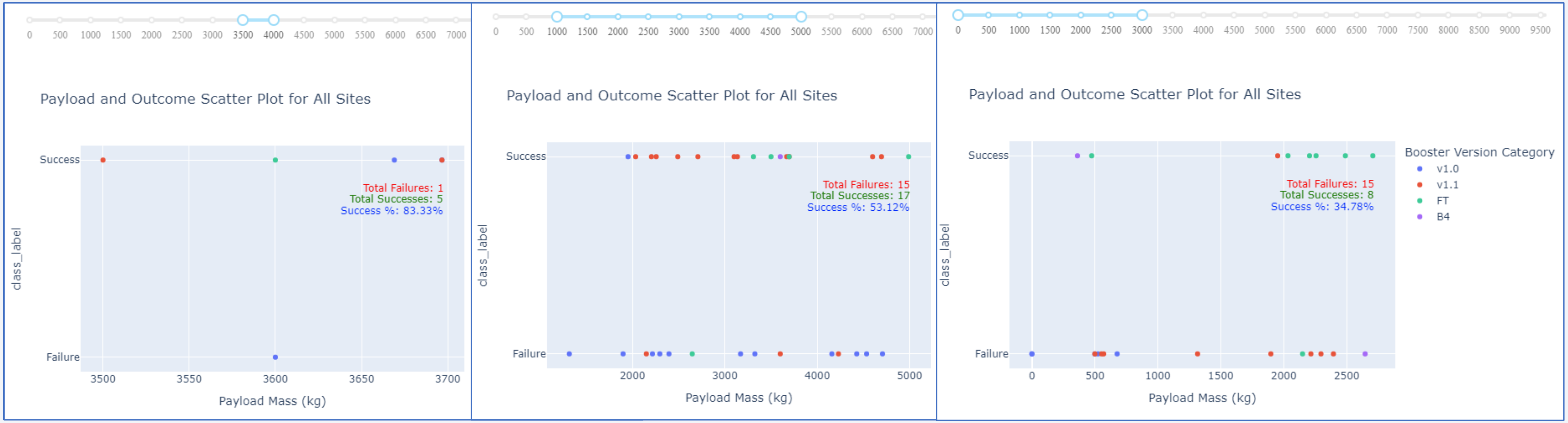
KSC LC-39A

Success and Failure Rates for KSC LC-39A

Findings:

- ‘39A (76.9%) has the highest launch success performance

Payload vs Launch Booster



Findings:

- Payloads between 3500 and 4000 kg have the highest success rate (83.33%)
- Payloads between 1000 and 5000 kg have a greater than 50% success rate (53.12%)
- Lighter Payloads to 3000 kg have the poorest rate (34.78%)

Section 5

Predictive Analysis (Classification)

Classification Accuracy

```
scores_test = pd.DataFrame(np.array([jaccard_scores, f1_scores, accuracy]), index=['Jaccard_Score', 'F1_Score', 'Accuracy'] , columns=['LogReg', 'SVM', 'Tree', 'KNN'])
scores_test
```

	LogReg	SVM	Tree	KNN
Jaccard_Score	0.800000	0.800000	0.800000	0.800000
F1_Score	0.888889	0.888889	0.888889	0.888889
Accuracy	0.833333	0.833333	0.833333	0.833333

- Classification models accuracy

Note: a bar chart seemed superfluous

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

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

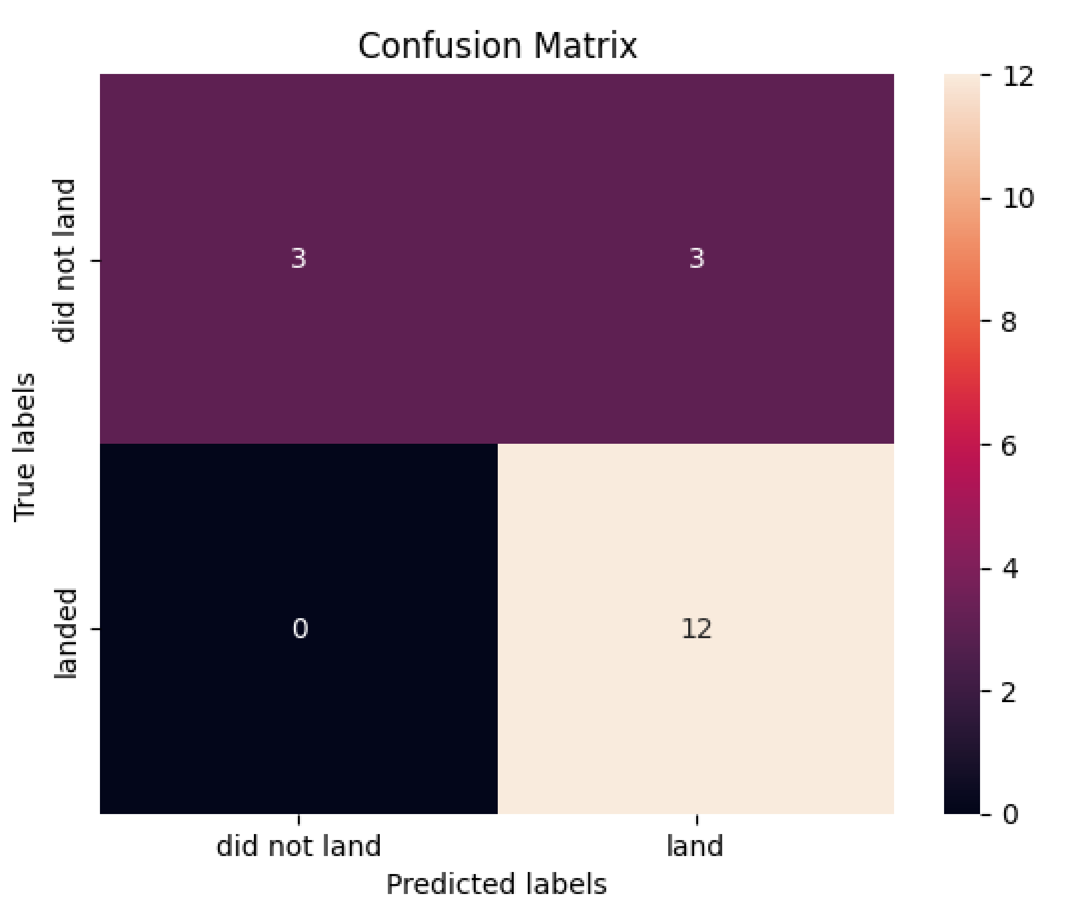
```
Best model is DecisionTree with a score of 0.8732142857142857
Best params is : {'criterion': 'entropy', 'max_depth': 18, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 5, 'splitter': 'random'}
```

- Best Model is the decision tree

Confusion Matrix

Findings:

- Confusion tree matrices are identical across models
- False positives (Type 1 errors) persist
- Precision 80%
- Recall 100%
- F1 Score 89%
- Accuracy 83.3%



Conclusions

Landing success factors

- Higher Payloads (kg)
- Orbiting in ESL1, GEO ,HEO, and especially SSO
- Improves over time
- Site LSC LC-39A has great landing success potential

Optimal operating conditions

- Proximity to the equator (likely due to free boost from earth rotation)
- Proximity to coast and distance from sensitive geographic features (populated areas, highways, airports)

Appendix

- Working Dash in google colab..

https://colab.research.google.com/drive/16pDXG2JMf_dmPgsJW6rwFQr16ib0XMbB?usp=sharing

Thank you!