Winning Space Race with Data Science

Steven Machonis
5/19/2022

Outline

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

Executive Summary

I used launch data from multiple sources including SpaceX's API and Wikipedia articles on launch information.

I ran multiple visualizations looking for variables that appeared to be related to each other and determined that Booster Version, Payload Mass, and Orbit are the most related to the success of the Booster recovery.

I used 4 different methods of Classification Predictors and determined that while all 4 returned an accuracy score of around 83%, that the Classification Tree predictor is what I would recommend. It had the higher accuracy of the parameter selection using GridsearchCV.

I also recommend revisiting the data as time goes on and as more recent data becomes available to see if one of the other methods ends up standing out in the future.

Introduction

- With the private industry space race just starting, more companies are starting to get into the industry and are looking to current companies for guidance.
- One of the ways that SpaceX is able to make this endeavor more affordable is by recovering and re-using the first stage of their rockets.
- Our goal is to determine if the first stage of our rocket will be recoverable. We will use multiple data points from SpaceX's successful and unsuccessful recoveries to determine the best factors that increase the chance we can recover the first stage.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - We will mainly be using two sets of data, one from the SpaceX API and one from Wikipedia data.
- Perform data wrangling
  - Data sets were modified to remove nulls, extra columns, and create new columns for analyzing the set.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Training and Testing sets were used to determine the best parameters for analysis and models were tuned based on them and evaluated based on each estimator's score method.

Data Collection

- Data Set 1: SpaceX API
- Data Set 2: Wikipedia Web Scraping

Data Collection – SpaceX API

Get the core data from SpaceX on their rocket launches

Makes 3 requests to the SpaceX API to get the following information to determine the values of indexed data in the core data pull:

Booster Version, Launch Sites, and Payload Data

Those data fields are then loaded into a new DataFrame containing all the pertinent information.

We then filtered out the Falcon 1 launches so our file only contains the Falcon 9 launches.

That data is saved to a CSV file.

Github Source:

https://github.com/smachonis/IBM-Data-Science-Capstone/blob/master/Data%20Collection%20Notebook.ipynb

Data Collection - Scraping

First we extract the Falcon 9 rocket launches HTML table from Wikipedia using Beautiful Soup.

Next, we create a new Data Frame by parsing the data found in the HTML table and building the headers and rows.

That information is saved into a CSV file.

Github Source:

https://github.com/smachonis/IBM-Data-Science-Capstone/blob/master/Data%20Collection%20Notebook%20with%20Webscraping.ipynb

Data Wrangling

I pulled the data into a new Data Frame from the CSV file

I reviewed for Null values and found that Landing Pad was the only field that contained nulls, due to rockets not always landing on an actual Landing Pad

I reviewed the distribution of launches related to the Launch Site and the Expected Orbit type

I reviewed the Outcomes column which was a complex column with two values describing whether the rocket successfully landed and where is was attempted such as True Ocean, False FTLS, etc.

The Successful True or False was then turned into a new Column with a 1 or 0 if the booster was successfully landed.

We determined that the landing was successful about 66% of the time.

Github Source:

https://github.com/smachonis/IBM-Data-Science-Capstone/blob/master/Data%20Collection%20Notebook.ipynb

EDA with Data Visualization

The following charts were created and reviewed

1. Flight Number vs Payload with Success Rate

The success rate increased as the flight number increased. In addition, payload appears to make a difference in the success rate as well. (Higher = Better)

2. Flight Number vs Launch Site with Success Rate

The success rate is different for each Lauch Site. The launch site has changed over time.

3. Payload Mass vs Launch Site with Success Rate

Higher Mass appears to be better. 1 site only sent rockets with Mass less than 10,000kg

4. Orbit Success Rate

ES-L1, GEO, HEO, and SSO have the highest success rate. Most others are in the 50 to 60% range, except SO which is 0%.

5. Flight Number vs Orbit with Success Rate

Orbit types have changed over time with newer orbits being more successful.

6. Payload vs Orbit with Success Rate

Heavy payloads are higher for some orbits more than others. Others like GTO fall somewhere in the middle.

7. Success rate over time

The rate has increased since 2013

Github Source: https://github.com/smachonis/IBM-Data-Science-Capstone/blob/master/EDA%20with%20Data%20Visualization.ipynb

EDA with SQL

Ran the following queries:

Display the names of the unique launch sites in the space mission

Display 5 records where launch sites begin with the string 'CCA’

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

Display average payload mass carried by booster version F9 v1.1

List the date when the first successful landing outcome in ground pad was achieved.

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

List the total number of successful and failure mission outcomes

List the names of the booster_versions which have carried the maximum payload mass.

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

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

Github Source:
https://github.com/smachonis/IBM-Data-Science-Capstone/blob/master/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

First, all launch sites were added to the map.

Next, I marked on the map which launches were successful/failed for each site. Mark Clusters were used to give summary information of total launches on the map as well.

After that, I started adding lines and distance calculators between the launch sites and close landmarks such as coastlines, highways, railways, and cities. Most launch sites appear to be close to the coastline, but further away from cities.

Github Source:
https://github.com/smachonis/IBM-Data-Science-Capstone/blob/master/Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Build a Dashboard with Plotly Dash

I included two charts with a dropdown for Launch Sites and a slider for Payload Mass to more easily view the relationship between those two variables. The first pie chart looks at the success rate of those launch sites, either a single on or all sites together.

The second charter is a scatter plot that looks at success rate versus the range of payload mass at either a single site or at all sites, showing categorized by the Booster version that was used.

Github Source:
https://github.com/smachonis/IBM-Data-Science-Capstone/blob/master/spacex_dash_app_final.py

Predictive Analysis (Classification)

I separated the data set into a training and a testing set with a 80/20 split. Using the testing set, I determined the best hyperparameters for each of the classification methodologies that I would be using.

I used GridsearchCV on K-Next Neighbor, Classification Tree, Logistical Regression, and Simple Vector Machines to determine the parameters and then ran the score functions for each.

A decision matrix was created for each one and evaluated.

The final conclusion is that all 4 models produced about the same level of accuracy with optimized parameters.

Github Source:
https://github.com/smachonis/IBM-Data-Science-Capstone/blob/master/Machine%20Learning%20Prediction.ipynb

Results

Each of the Machine Learning methodologies produced the same accuracy percentage, which was about 83.3%. The biggest issue to look out for was the number of false positives in the predictor, which was also consistent in all methods.

I feel that Booster Version, Payload Mass, and Orbit are the best determinants of a successful launch based on the EDA process.

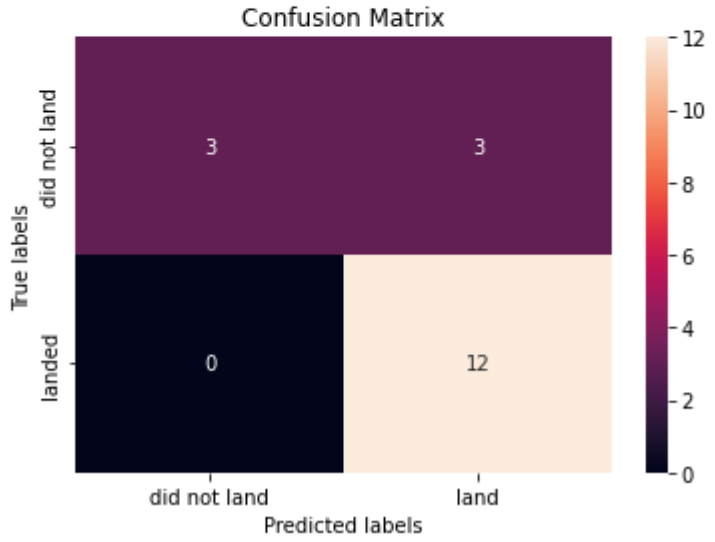
As with most things in technology, results tend to be better over time, so revisiting this process with newer data will help to make the predictor as accurate as possible moving forward.

At this time, I would recommend the Tree Classification Predictor since the accuracy of the parameter selection was slightly higher than the rest while maintaining the same over-all accuracy score.

```
In [23]: tree_cv.score(X_test,Y_test)
Out[23]: 0.8333333333333334
```

We can plot the confusion matrix

```
In [24]: yhat = tree_cv.predict(X_test)
         plot_confusion_matrix(Y_test,yhat)
```



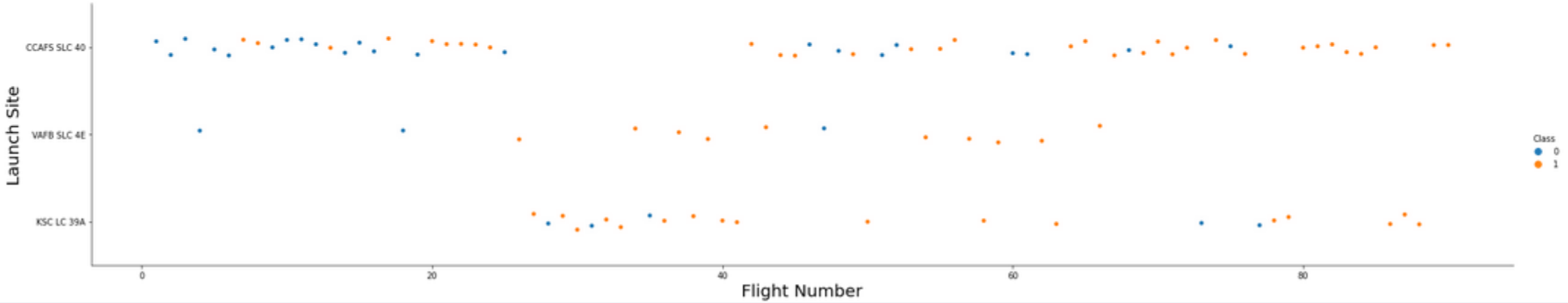
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

In [4]:

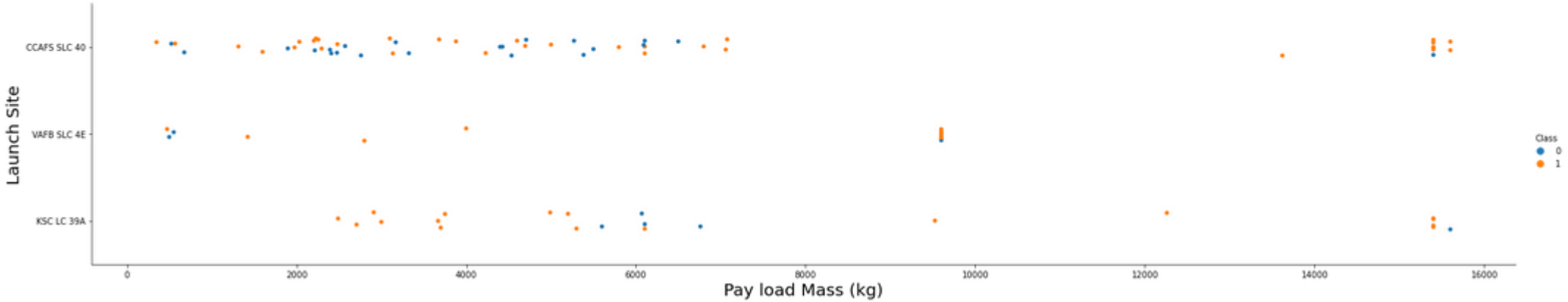
```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



As the flight number increased, in other words, as time went on, flights became more successful at all locations. VAFB SLC 4E has not been used recently, but was experiencing some success.

Payload vs. Launch Site

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Pay load Mass (kg)",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



This chart shows that the launches with higher payload (> 10000kg) have a much better success rate. This also shows that VAFB doesn't appear to be able to use a payload that high. That may also explain the VAFB not recently being used for launches as SpaceX finds sites that can handle the large payload weight due to the success of launches.

Success Rate vs. Orbit Type

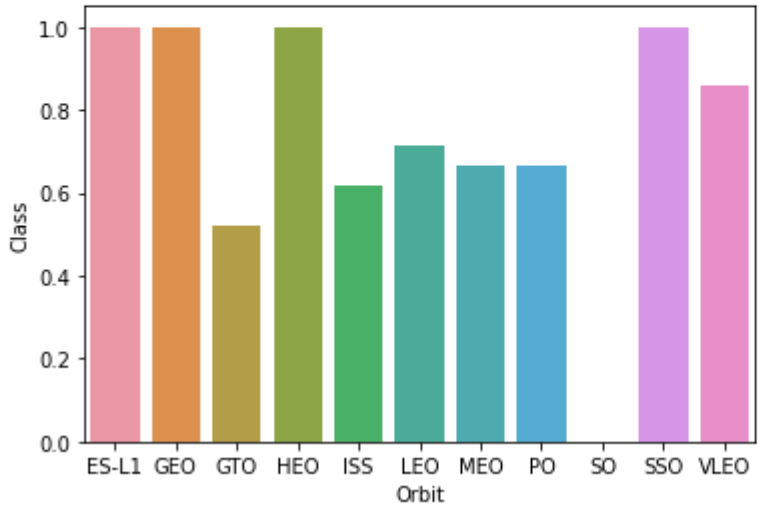
Certain orbit types definitely have a higher success rate than others.

ES-L1, GEO, HEO, and SSO have a success rate near 100%, while most others rate between 50 and 60%.

SO appears to be the only one that has an almost 0% chance.

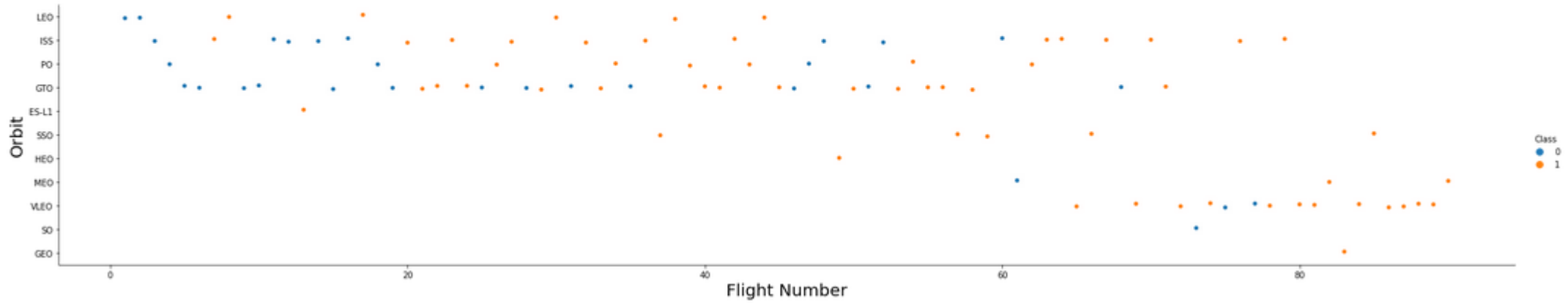
In [6]:
```
# HINT use groupby method on Orbit column and get the mean of Class column
orbit_group = df[['Orbit','Class']]
grouped_orbit = orbit_group.groupby(['Orbit'],as_index=False).mean()
sns.barplot(x='Orbit',y='Class', data=grouped_orbit)
```

Out[6]:
```
<AxesSubplot:xlabel='Orbit', ylabel='Class'>
```



Flight Number vs. Orbit Type

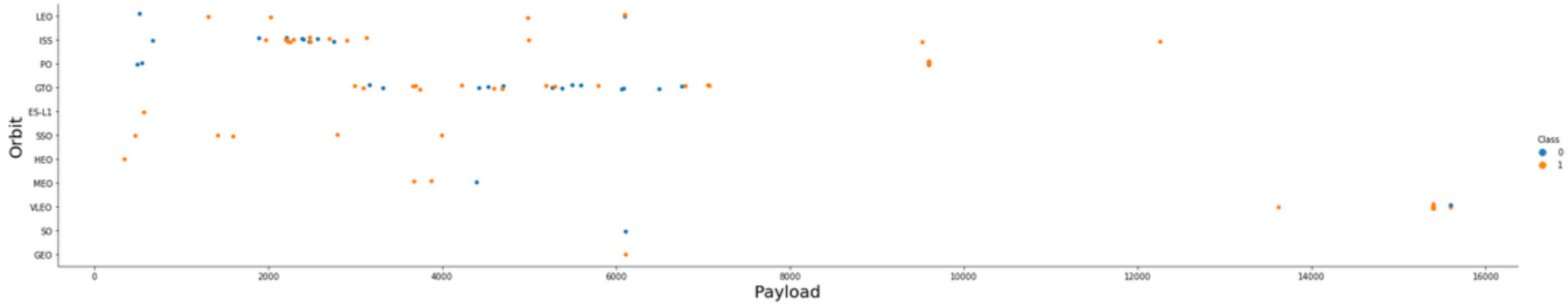
In [7]:
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Other than the higher success rate as time goes on for all Orbits, the VLEO appears to have the biggest relationship with the flight number, while others like GTO seem to have no relationship.

Payload vs. Orbit Type

In [8]:
```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Payload",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



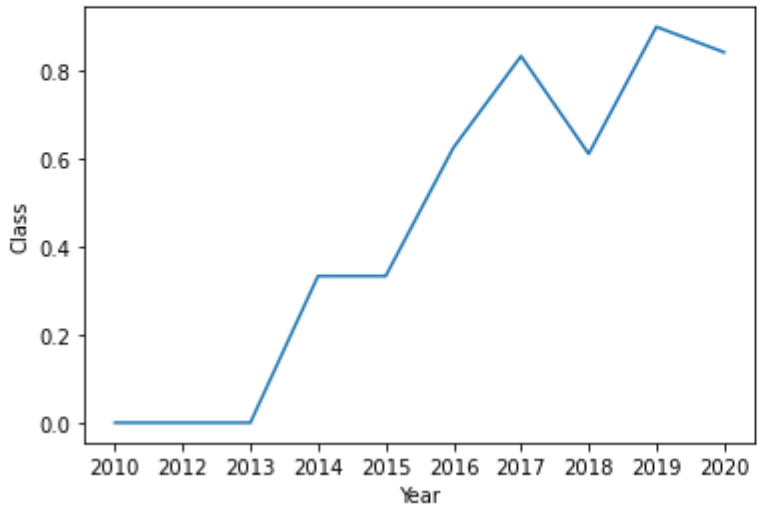
Again, there seems to be a relationship between Payload and Orbit type for some orbit types, such as VLEO, Polar, and ISS. Other orbit types like GTO again seem to have no relationship. These relationships are similar to the relationship between Orbit and Flight Number as well.

Launch Success Yearly Trend

As you can see, the success rate has continued to generally increase since 2013.

In [10]:
```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
sns.lineplot(x='Year',y='Class', data=year_group)
```

Out[10]:
```
<AxesSubplot:xlabel='Year', ylabel='Class'>
```



All Launch Site Names

These are the unique names of the launch sites for the Falcon 9 rocket.

```
In [9]: %sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL
 * ibm_db_sa://ldw69913:***@ea286ace-86c7-4d5b-858
Done.
```

Out[9]:

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

Launch Site Names Begin with 'CCA'

In [10]:
```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://ldw69913:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90108kqb1od8lcg.databases.appdomain.cloud:31505/bludb
Done.
```

Out[10]:

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

Here are the first 5 records of launches that happened from either of the 2 CCA launch sites.

Total Payload Mass

This is the total Payload Mass in Kilograms for all boosters carried by NASA. 107,010kg

```
In [12]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER LIKE '%NASA%'
 * ibm_db_sa://ldw69913:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90108kqb1o
Done.
Out[12]:    1
            107010
```

Average Payload Mass by F9 v1.1

This is the total Payload Mass in Kilograms for the F9 v1.1 booster. 2,534kg

```
In [13]: %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION LIKE 'F9 v1.1%'
 * ibm_db_sa://ldw69913:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od81cg.data
Done.
Out[13]: 1
2534
```

First Successful Ground Landing Date

The first successful ground pad landing occurred on 12/22/2015.

```
In [14]: %sql SELECT MIN(DATE) FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (ground pad)'
 * ibm_db_sa://ldw69913:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90108kqb1od8lcg.dat
Done.
Out[14]:        1
        2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

Below are the 4 types of boosters that were able to successfully land on a drone ship where the Payload Mass was between 4000kg and 6000kg.

In [15]:
```
%sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BET
WEEN 4000 and 6000
```

```
 * ibm_db_sa://ldw69913:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb
Done.
```

Out[15]:

booster_version
F9 FT B1021.2
F9 FT B1031.2
F9 FT B1022
F9 FT B1026

Total Number of Successful and Failure Mission Outcomes

Below is the total number of Successful and Failed missions.

```
In [17]: %sql SELECT MISSION_OUTCOME, COUNT(*) FROM SPACEXTBL GROUP BY MISSION_OUTCOME
 * ibm_db_sa://ldw69913:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb
Done.
```

Out[17]:

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

Here are the boosters that were used to carrier the maximum payload.

```
In [19]: %sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
 * ibm_db_sa://ldw69913:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90108kqb1od8lcg.databases.appdomain.cloud:31505/bludb
Done.
```

Out[19]:

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

Below are the failed landings on a drone ship, along with their booster versions, and launch site name, that occurred in 2015.

```
In [25]: %sql SELECT DATE,LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone shi
         p)' AND DATE_PART('year',DATE) = '2015'
```

```
 * ibm_db_sa://ldw69913:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb
Done.
```

Out[25]:

DATE	landing__outcome	booster_version	launch_site
2015-01-10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

Here are the number of flights by landing outcomes that took place between 6/4/2010 and 3/20/2017.

```
In [27]: %sql SELECT LANDING__OUTCOME, COUNT(*) FROM SPACEXTBL WHERE DATE BETWEEN DATE('2010-06-04') AND DATE('2017-03-20') GROUP BY L
         ANDING__OUTCOME ORDER BY COUNT(*) DESC
          * ibm_db_sa://ldw69913:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb
         Done.
```

Out[27]:

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

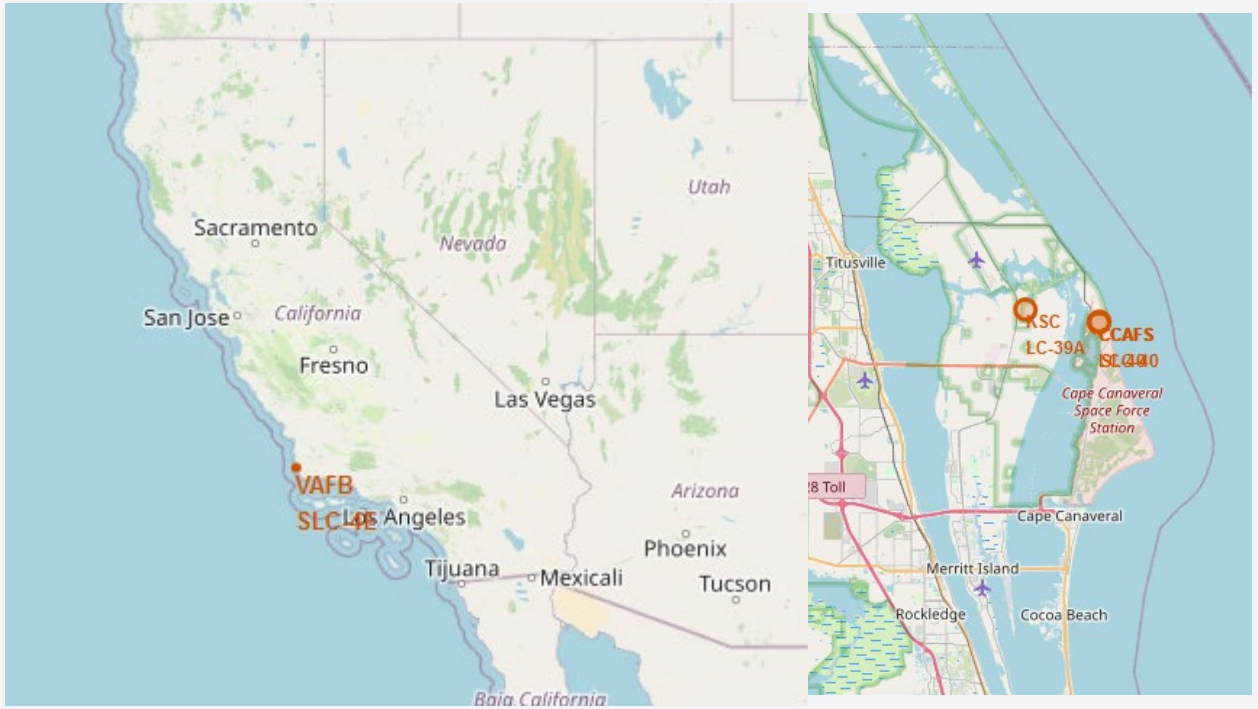
Section 3

Launch Sites Proximities Analysis

All Launch Sites

All of the launch sites are in the United States.

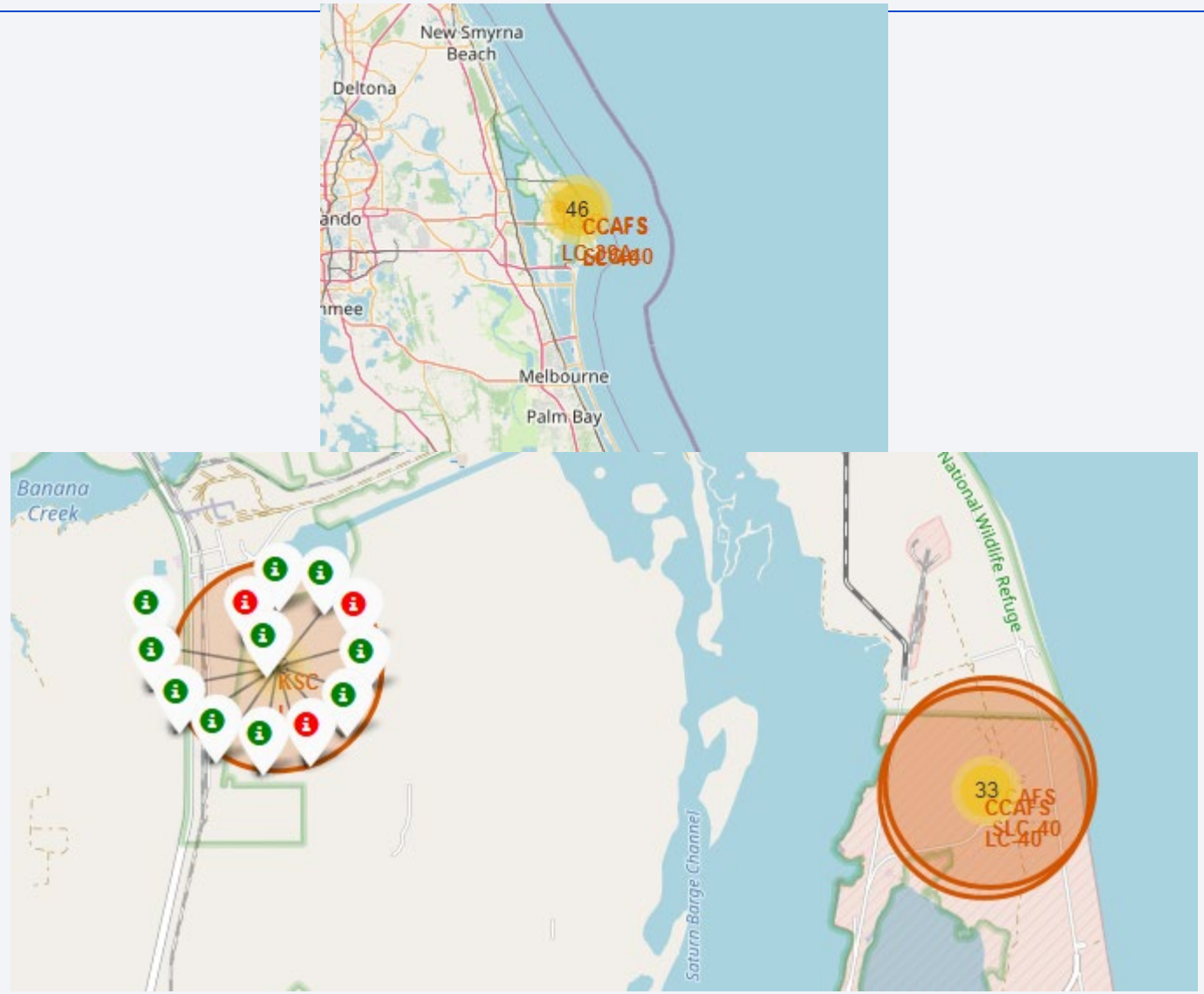
They are also all on the Coastline of California and Florida.



All Launch Sites Outcomes

Each site is labled with the name and the number of launches.

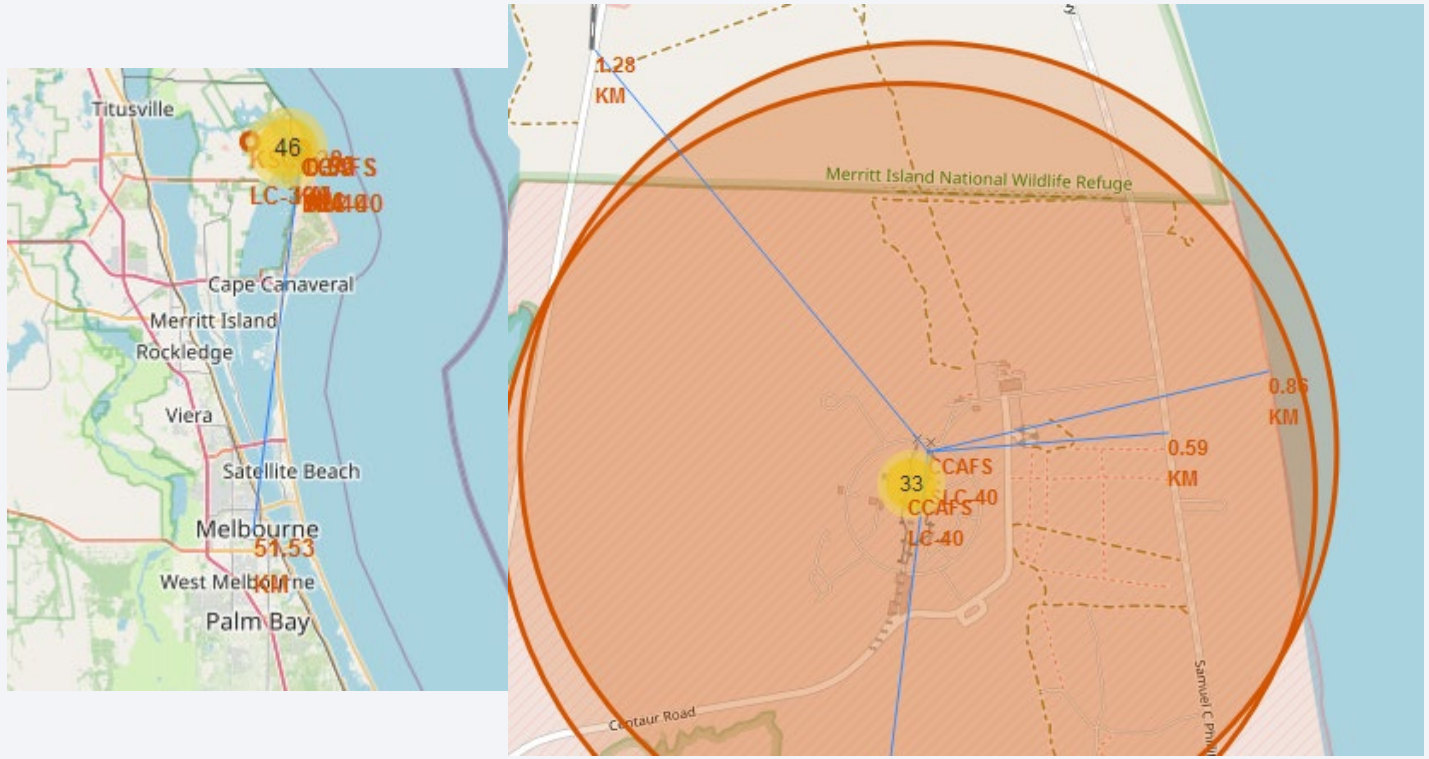
Each launch site has it's own cluster of markers showing the number of successful and failed launches

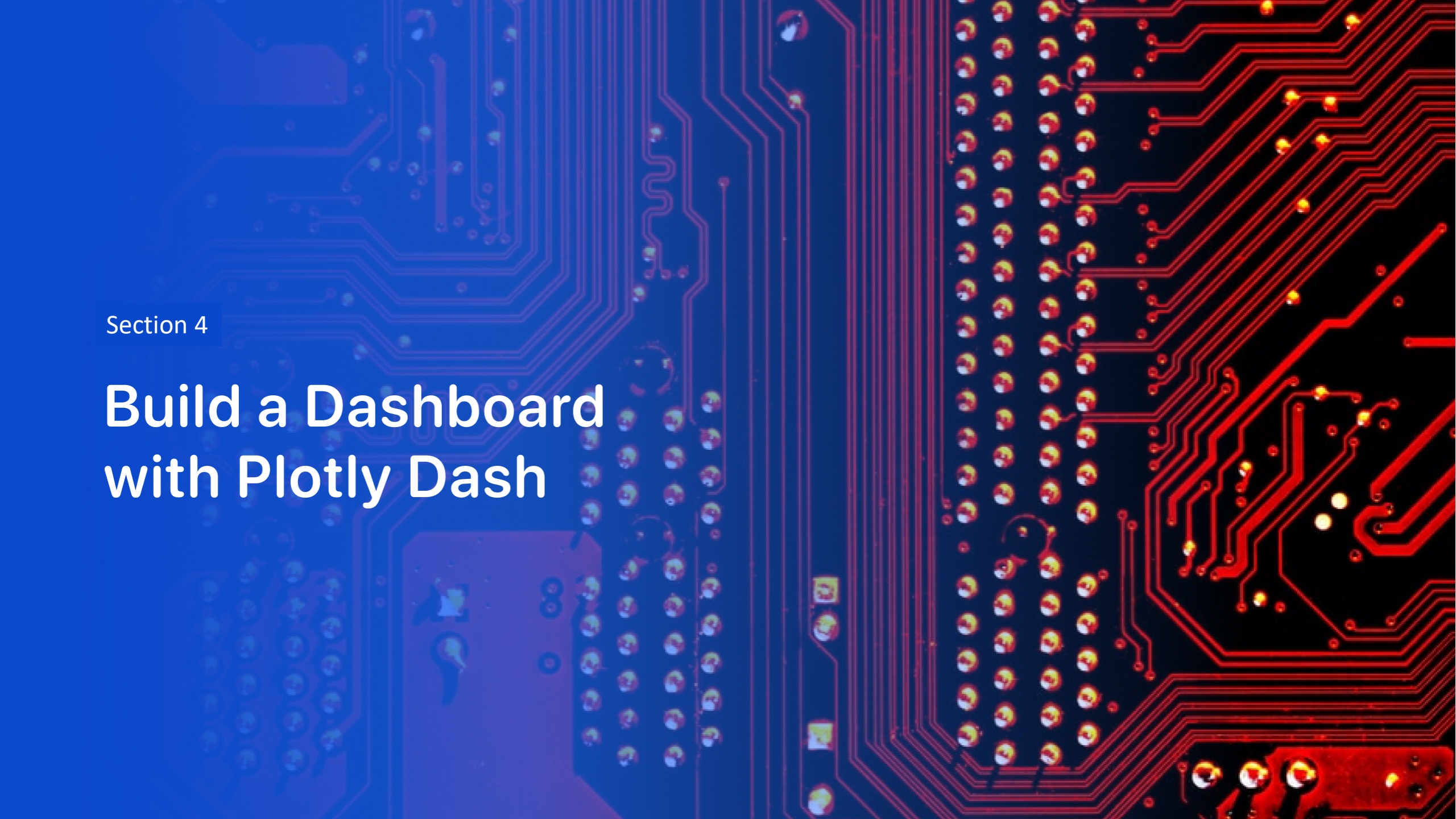


Launch Site Distances to Landmarks

The maps shows the distance in KM between the launch site and various landmarks including the Coastline, Highways, Railways, and Cities.

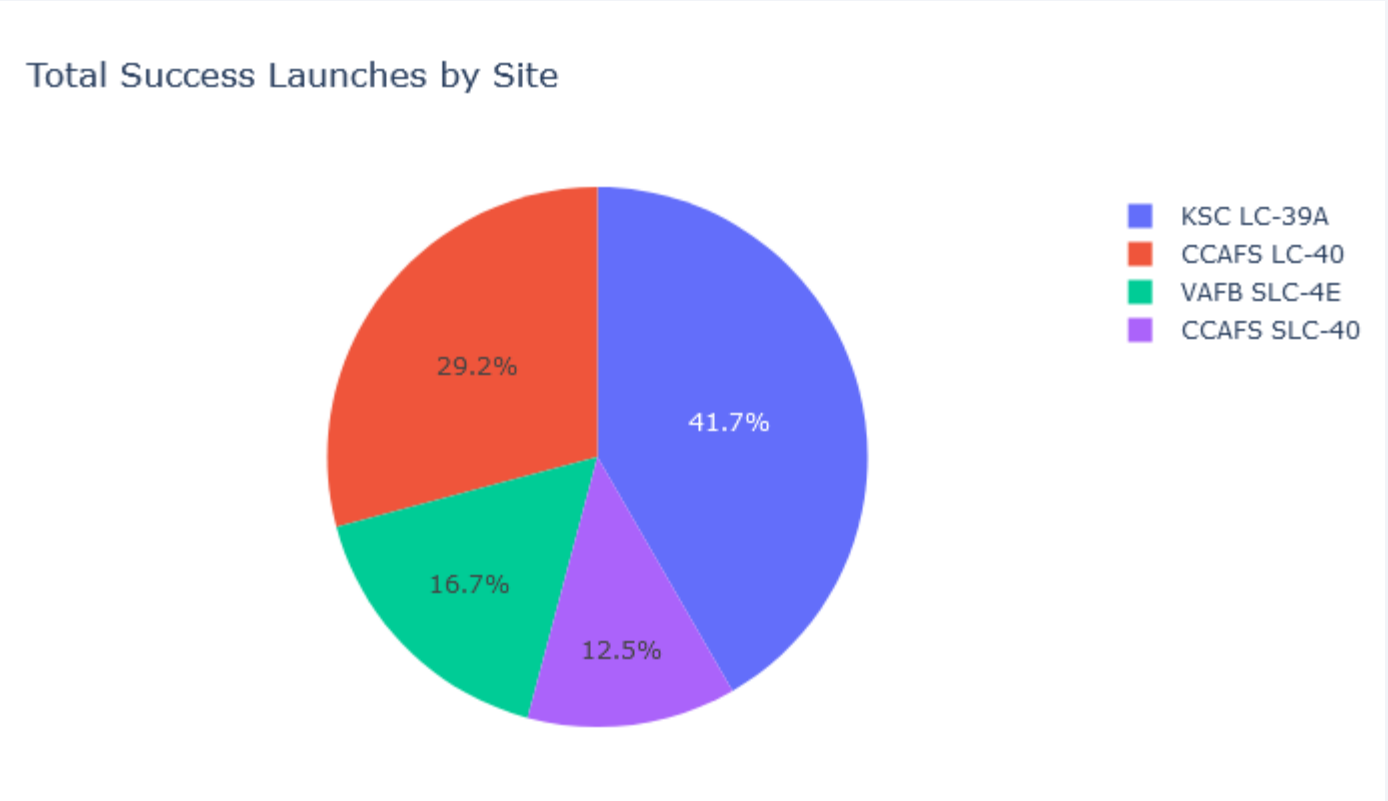
Most launch sites are near a coastline and a railway, but tend to be further away from large cities.




Section 4
Build a Dashboard with Plotly Dash

Total Successful Launches on All Sites

The chart shows the percentage of successful launches of each site. It shows that the most successful one is KSC LC-39A with 41.7% of the successful launches. Hovering over the pie piece, I can see that it has had 10 successful Launches.

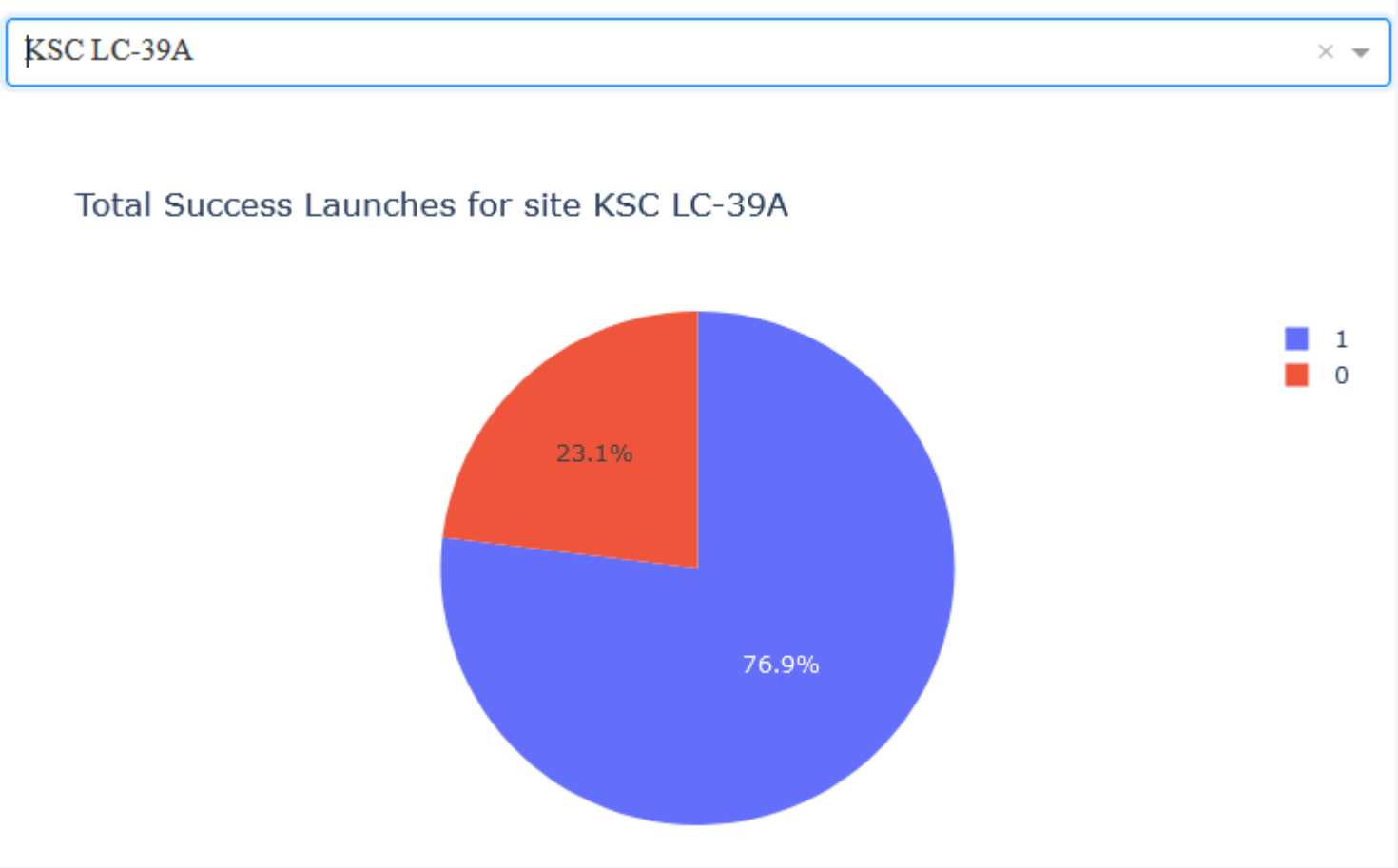


Most Successful Site

The most successful site is KSC LC-39A. They succeeded almost 79% of the time.

The had 10 successes and only 3 failures.

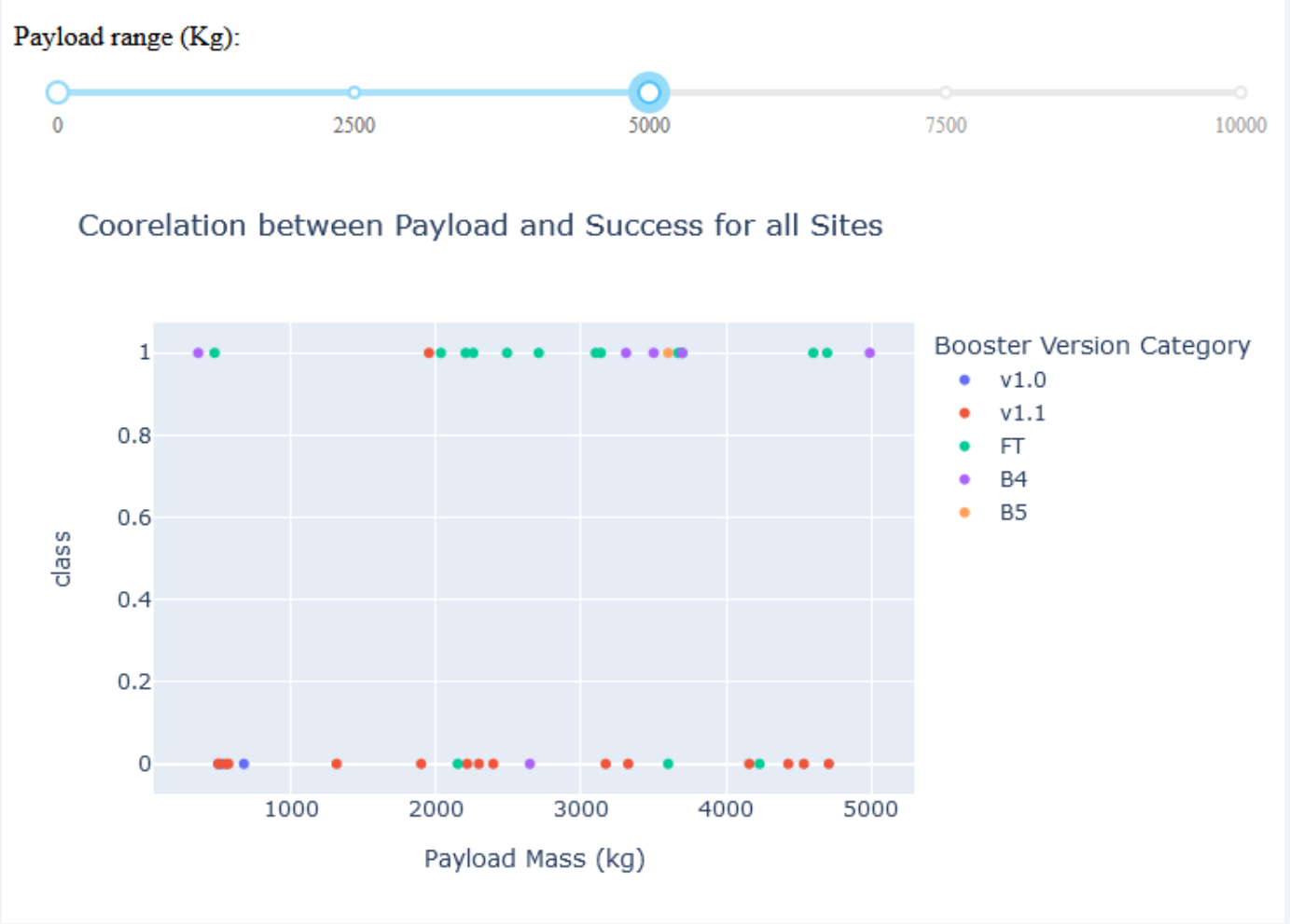
The next closest site is CCAFS LC-40 with a success rate of 73.1%



Success Rate for Payloads less than 5000kg

All sites are selected and the payload range is limited to less than 5000kg.

At this lower amount of Payload, certain Booster versions tend to be more successful, like FT and B4. v1.0 and v1.1 appear to be very unsuccessful at these low payload amounts.



Section 5

Predictive Analysis (Classification)

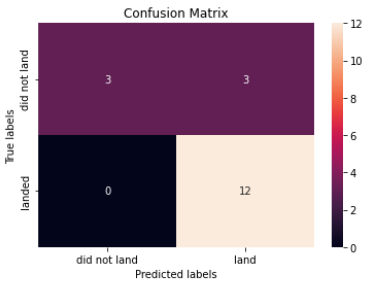
Classification Accuracy

After optimizing the methodology for the best parameters, all 4 methods came out with the same accuracy score of 83.3%.

```
In [23]: tree_cv.score(X_test,Y_test)
Out[23]: 0.8333333333333334
```

We can plot the confusion matrix

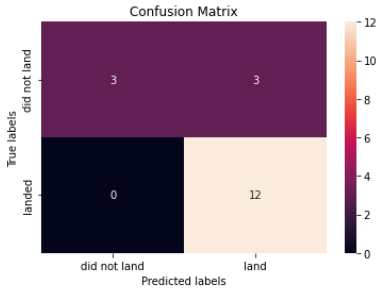
```
In [24]: yhat = tree_cv.predict(X_test)
         plot_confusion_matrix(Y_test,yhat)
```




```
In [18]: svm_cv.score(X_test,Y_test)
Out[18]: 0.8333333333333334
```

We can plot the confusion matrix

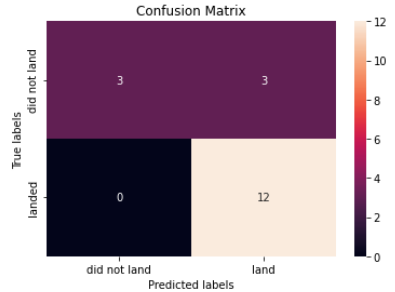
```
In [19]: yhat=svm_cv.predict(X_test)
         plot_confusion_matrix(Y_test,yhat)
```




```
In [28]: knn_cv.score(X_test,Y_test)
Out[28]: 0.8333333333333334
```

We can plot the confusion matrix

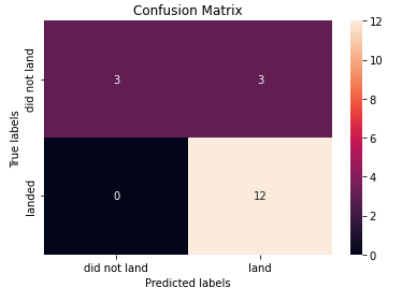
```
In [29]: yhat = knn_cv.predict(X_test)
         plot_confusion_matrix(Y_test,yhat)
```




```
In [13]: logreg_cv.score(X_test,Y_test)
Out[13]: 0.8333333333333334
```

Lets look at the confusion matrix:

```
In [14]: yhat=logreg_cv.predict(X_test)
         plot_confusion_matrix(Y_test,yhat)
```



Confusion Matrix

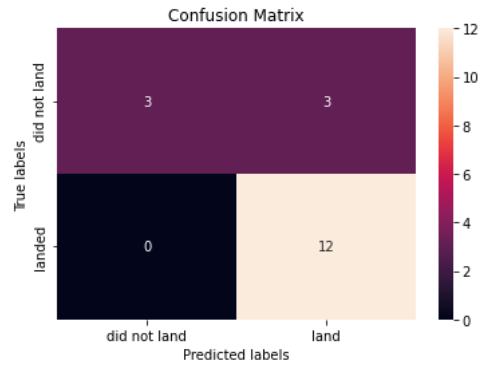
The Tree Classification is the best method. While the overall accuracy score was the same, the parameter optimization was more accurate than the others.

In [23]:
```
tree_cv.score(X_test,Y_test)
```

Out[23]:
```
0.8333333333333334
```

We can plot the confusion matrix

In [24]:
```
yhat = tree_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```



In [20]:
```
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
```

In [21]:
```
tree_cv = GridSearchCV(tree,parameters,cv=10)
tree_cv.fit(X_train,Y_train)
```

Out[21]:
```
GridSearchCV(cv=10, estimator=DecisionTreeClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [2, 4, 6, 8, 10, 12, 14, 16, 18],
                         'max_features': ['auto', 'sqrt'],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [2, 5, 10],
                         'splitter': ['best', 'random']})
```

In [22]:
```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 10, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_s
amples_split': 10, 'splitter': 'random'}
accuracy : 0.8892857142857142
```

Conclusions

- The Classification Tree is the best method based on the training and test site I used.
- Booster Version, Payload Mass, and Orbit are the best determinants of a successful launch based on the EDA process.
- I recommend launch sites near the coastline.

Appendix

- Github Project Repo: https://github.com/smachonis/IBM-Data-Science-Capstone

Thank you!