IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Timothy Danielle C. Buitre
January 30, 2023

Outline

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

Executive Summary

The purpose of this research is to build the best model to determine or classify if the first stage of a Falcon 9 rocket will land so that we can determine the cost of a launch.

To achieve this we have gathered data from SpaceX REST API and web scraping of a Wikipedia page that contains data regarding the said project. The data that was collected from two sources were consolidated and cleaned. We have explored and analyzed the data using SQL and visualizations including folium to map out the locations of the different launch sites. A Plotly Dash App were also created to easily interact, navigate, and visualize the data.

After leaning the features that are relevant in classifying the success of a launch such as payload weight, launch site locations, orbit type, booster versions, etc., different machine learning models were tested to identify the best in classifying launches. The best models were LR, SVM, and KNN, all having 94% accuracy score.

Introduction

- ## Project background and context

The commercial space age is here, and the most successful company in this regard is founded by Elon Musk, SpaceX. One of the main reason for this is that their rocket launches are relatively inexpensive because they can 'reuse' the first stage. Therefore, as a team from the competing company SpaceY founded by Allon Musk, our job is to determine if the first stage of a Falcon 9 rocket will land so that we can determine the cost of a launch.

- ## Problems you want to find answers

The problem that needs to be solved now is how to determine if the first stage will land successfully for it to be reused. It is good to identify which features of a rocket launch will affect the success rate of the first stage landing successfully: Which launch site has the highest success rate? What are the F9 versions of the rocket that has the highest launch success rate? Does the weight of the payload affect the success rate of a launch? What are the other contributing factors affects the success of landing the first stage of a rocket launch?

Section 1

Methodology

Methodology

Executive Summary

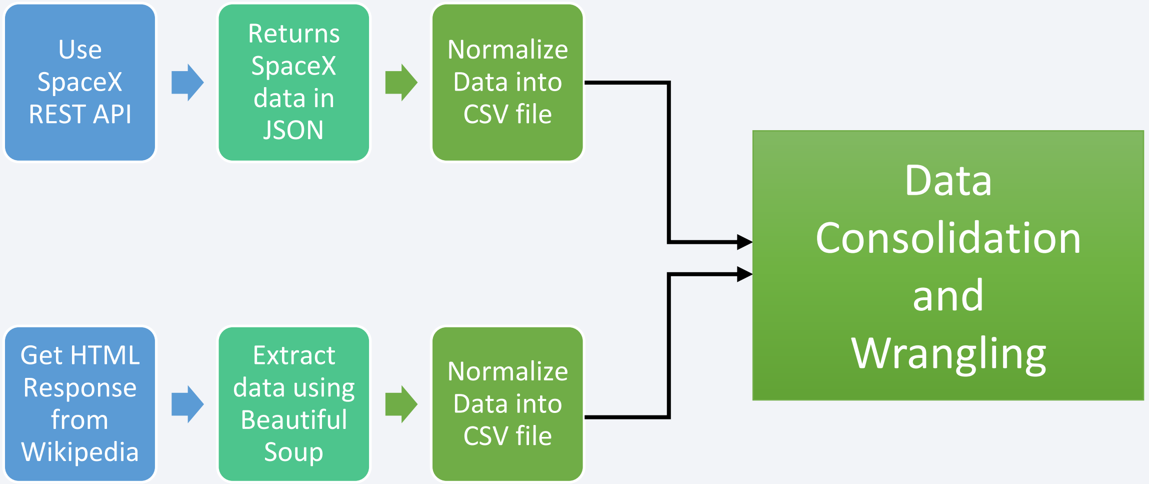
- Data collection methodology:
  - SpaceX REST API
  - Web Scraping from Wikipedia
- Perform data wrangling
  - One Hot Encoding data features for Machine Learning and Data cleaning of null values and irrelevant columns
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - LR, KNN, SVM, DT models have been trained and evaluated to identify the best classification model

Data Collection

In this research, we have gathered the data using both SpaceX REST API and web scraping.

We have worked on with the SpaceX launch data that is gathered from an API, specifically the SpaceX REST API. This API gave us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome. The main endpoint of the SpaceX REST API that we have worked on is api.spacexdata.com/v4/launches/past.

For additional data, we have web scraped the data of Falcon 9 historical launch records from a Wikipedia page titled “List of Falcon 9 and Falcon Heavy launches”. The table was parsed and converted into a Pandas data frame with the help of the BeautifulSoup package in Python.



Data Collection – SpaceX API

- Data Collection with SpaceX REST calls

'FLOW CHART'

- GitHub URL of the completed SpaceX API calls notebook https://github.com/danwithcode/Capstone-Project/blob/master/SpaceX%20API%20Data%20Collection.ipynb

1. Getting response from API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

2. Converting response to a JSON File

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

3. Apply custom functions to clean data

```
# Call getBoosterVersion
getBoosterVersion(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getCoreData
getCoreData(data)
```

4. Assign lists to a dictionary then convert it into a pandas dataframe

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

```
# Create a data from launch_dict
df= pd.DataFrame(launch_dict)
```

5. Filter the dateframe (only include Falcon 9)

```
data_falcon9 = df[df['BoosterVersion']!='Falcon 1']
data_falcon9.head()
```

6. Data Wrangling (deal with Null values)

```
# Calculate the mean value of PayloadMass column
PLMmean = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan,PLMmean,inplace=True)
data_falcon9.isnull().sum()
```

7. Save and export file into a CSV file

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

Data Collection - Scraping

- Web scraping process

‘Key Phrases and Flowcharts’

- GitHub URL of the completed web scraping notebook

[https://github.com/danwithcode/Capstone-Project/blob/master/Web scraping.ipynb](https://github.com/danwithcode/Capstone-Project/blob/master/Web%20scraping.ipynb)

1. Getting response from HTML

```
# use requests.get() method with the provided static_url
response = requests.get(static_url)
# assign the response to a object
html = response.text
```

2. Create a BeautifulSoup object from the HTML response

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

3. Finding the tables

```
html_tables= soup.find_all('table')
```

4. Finding the tables

```
column_names = []
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if name != None and len(name) > 0:
        column_names.append(name)
```

5. Create a dictionary

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6. Fill up the dictionary with launch records extracted from table rows. (For Full code refer to Code Input #13 in notebook)

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
```

7. Convert the dictionary into a dataframe

```
df=pd.DataFrame(launch_dict)
```

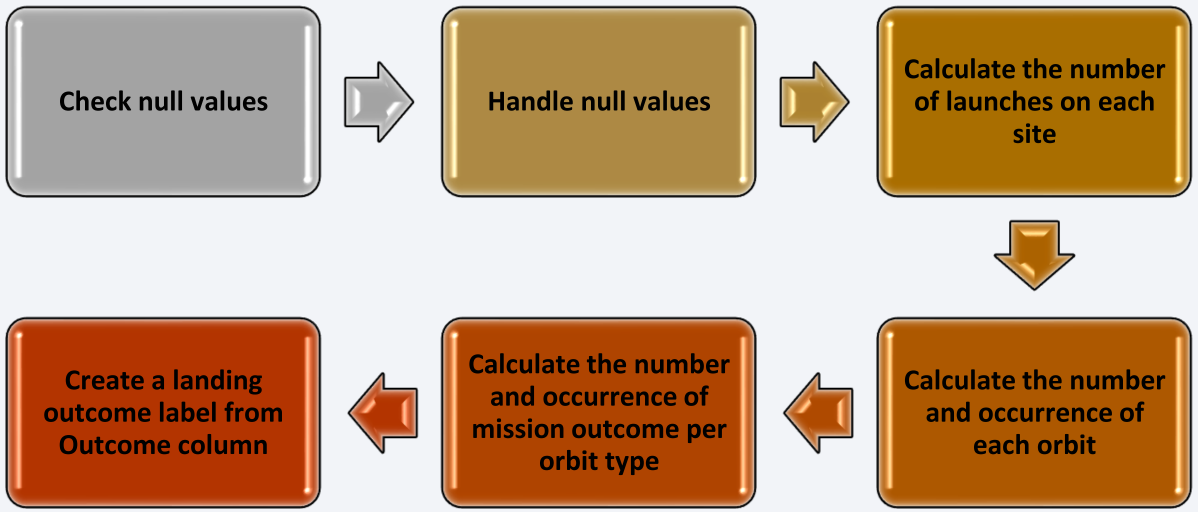
8. Save and export file into a CSV file

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

Data Wrangling

The consolidated data were evaluated and scanned to identify missing values and deal with them.

Each samples were also classified into either success or failure regarding the landing of the first stage after a launch.



LINK-

https://github.com/danwithcode/Capstone-Project/blob/master/Data%20Wrangling.ipynb

EDA with Data Visualization

The charts and graphs that were used to visualized the data for it to be easily analyzed are <u>scatter plots</u>, <u>bar chart</u>, and a <u>line plot</u>.

Scatter plot is great for visualizing the relationship between two variables that is why it is the most used tool in this section.

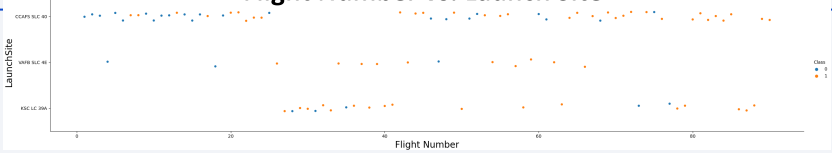
Bar chart is great for visualizing values in a category or group. It is used to visualize the success rate of each orbit type.

Lastly, a line plot is great for visualizing trend in a timeline. It was used here to visualize the launch success yearly trend.

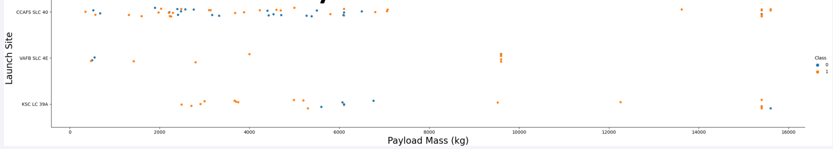
Each of this charts and graphs will have their own slide and an in-depth discussion in section 2 (Insights Drawn from EDA).

GitHub Link - https://github.com/danwithcode/Capstone-Project/blob/master/EDA%20Visualization.ipynb

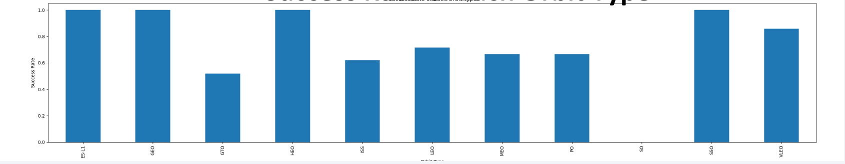
Flight Number vs. Launch Site



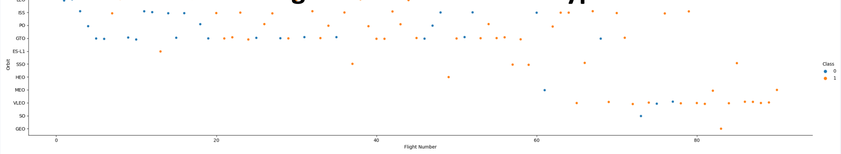
Payload vs. Launch Site



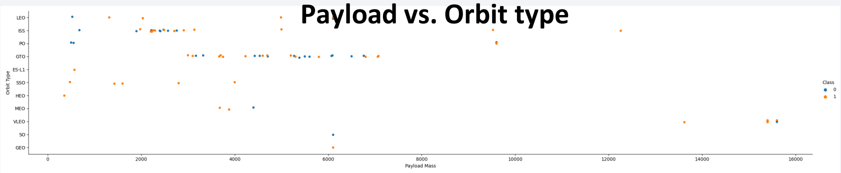
Success Rate of Each Orbit Type



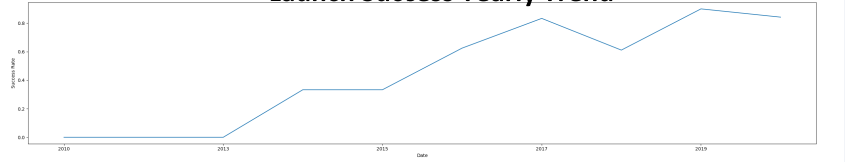
Flight Number vs. Orbit type



Payload vs. Orbit type



Launch Success Yearly Trend



EDA with SQL

SQL Queries Performed Includes:

- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string 'CCA’
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the date when the first successful landing outcome in ground pad was achieved.
- Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster versions which have carried the maximum payload mass. Use a subquery
- Listing the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Ranking 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 URL -https://github.com/danwithcode/Capstone-Project/blob/master/EDA%20with%20SQL.ipynb

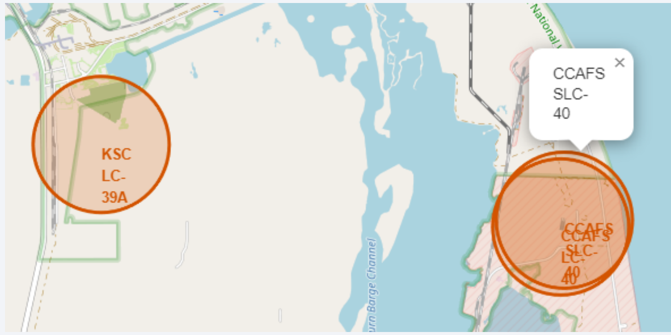
Build an Interactive Map with Folium

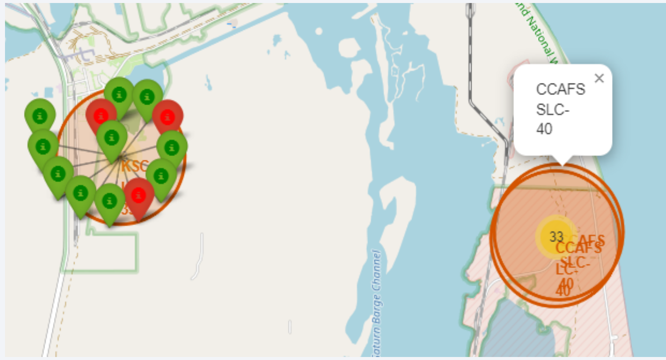
The map objects that have been added to the map were circles, pop-up markers, marker clusters, and line objects.

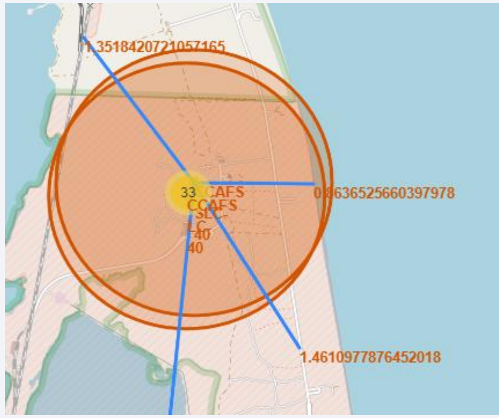
Through this interactive folium map we've learned that the **optimal location** for a launch site has certain distances from its proximities.

GitHub URL - https://nbviewer.org/github/danwithcode/Capstone-Project/blob/master/Visual%20Analytics%20and%20Dashboard.ipynb

*Note: The Jupyter notebook in GitHub were bind to a Jupyter nbviewer. GitHub doesn't render large Jupyter Notebooks.







*__Circles__ were added to mark all launch sites on a map

*__Pop-up markers__ were also added to easily select launch sites with close proximity to each other

*__Marker Clusters__ were added to mark the success/failed launches for each site on the map

*__Line objects__ were added to calculate the distances between a launch site to its proximities such as: railways, highways, coastline and cities.

Build a Dashboard with Plotly Dash

The plots/ graphs that were added to the Dashboard built on Plotly Dash were:

“Success Pie Chart”
and
“Success Payload Scatter Chart.”

GitHub URL - https://github.com/danwithcode/Capstone-Project/blob/master/DashApp.py

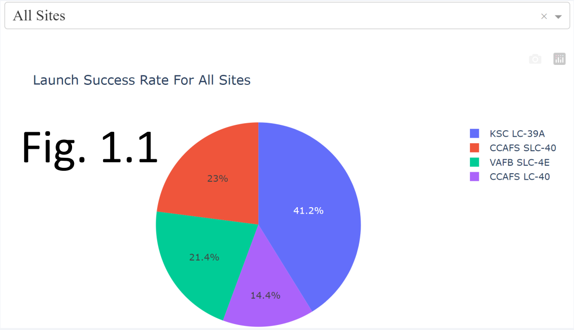


Fig. 1.1

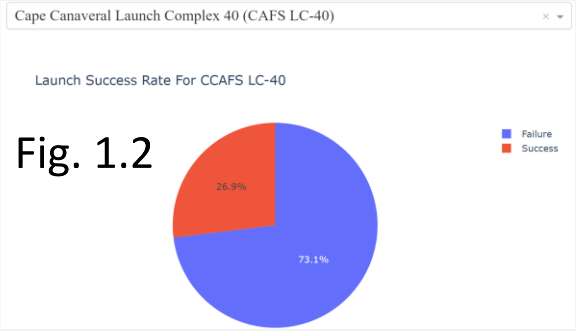


Fig. 1.2

Fig. 2

Success Pie Chart: This chart renders the success rate of all sites by default (Fig 1.1). It also visualizes the success/failure pie chart of individual launch sites through the drop down options (Fig 1.2). This makes it easier to identify which launch site has the most successful launches and vice versa.

Success Payload Scatter Chart: This chart renders the class (success/fail) of each samples in their respective payload range, color coded according to their booster versions. By default it shows all sites. Different sites are also rendered with the dropdown option. The range of the payload can also be toggled using the slider. This makes it easier to identify which payload range and booster version has the most success for each launch sites. (Fig. 2)

Predictive Analysis (Classification)

The data was standardized using the preprocessing function in the scikitlearn package. The target and predictive variables were then split into testing and training set using the train_test_split function.

The data were then fitted into the different models (LR, SVM KNN, and DT). GridSearchCV was used to test different parameters and find the most optimal of them.

To find the best performing model, each model was then visualized into a confusion matrix and their scores were calculated using R^2.

GitHub URL - https://github.com/danwithcode/Capstone-Project/blob/master/Predictive%20Analysis.ipynb

Model Development Process Flow Chart

1. Create a Pandas Series from a NumPy array created from the column Class of the data

```
Y = data['Class'].to_numpy()
Y = pd.Series(Y)
type(Y)
```

2. Standardize the data

```
X = preprocessing.StandardScaler().fit_transform(X)
X
```

3. train_test_split the data

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

4. Create an object for each model and perform GridSearchCV to find the best parameter for each of them

LR

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

```
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}
lr=LogisticRegression()

logreg_cv = GridSearchCV(lr, parameters, cv=10)
logreg_cv.fit(X, Y)
logreg_cv.best_estimator_
```

SVM

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

```
svm_cv = GridSearchCV(svm, parameters, cv=10)
svm_cv.fit(X, Y)
svm_cv.best_estimator_
```

DT

```
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()
```

```
tree_cv = GridSearchCV(tree, parameters, cv=10)
tree_cv.fit(X, Y)
tree_cv.best_estimator_
```

KNN

```
parameters = {'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
              'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
              'p': [1,2]}

KNN = KNeighborsClassifier()
```

```
knn_cv = GridSearchCV(KNN, parameters, cv=10)
knn_cv.fit(X, Y)
knn_cv.best_estimator_
```

5. Calculate accuracy of each model

LR

```
print('score on train data: ', logreg_cv.score(X_train, Y_train))
print('score on test data : ', logreg_cv.score(X_test, Y_test))

score on train data:  0.875
score on test data :  0.9444444444444444
```

SVM

```
print('score on train data: ', svm_cv.score(X_train, Y_train))
print('score on test data : ', svm_cv.score(X_test, Y_test))

score on train data:  0.8611111111111112
score on test data :  0.9444444444444444
```

DT

```
print('score on train data: ', tree_cv.score(X_train, Y_train))
print('score on test data : ', tree_cv.score(X_test, Y_test))

score on train data:  0.8611111111111112
score on test data :  0.8333333333333334
```

KNN

```
print('score on train data: ', knn_cv.score(X_train, Y_train))
print('score on test data : ', knn_cv.score(X_test, Y_test))

score on train data:  0.875
score on test data :  0.9444444444444444
```

6. Plot the confusion matrix of each model

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

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

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

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

Results

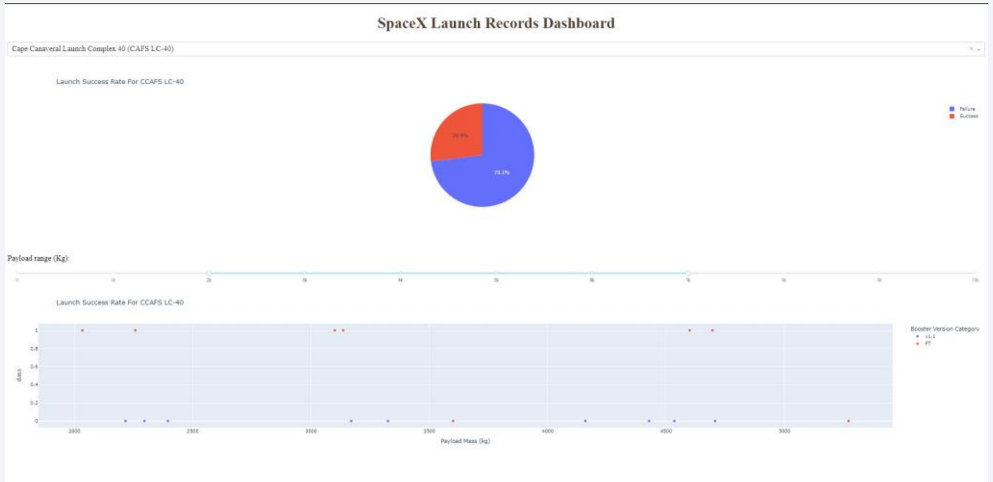
Exploratory Data Analysis Results

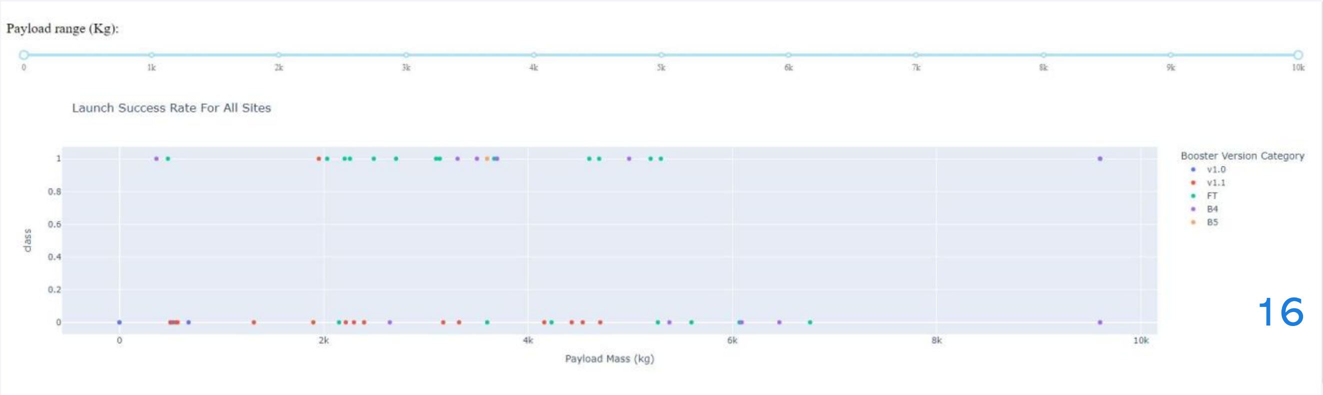
- Low weighted payload performs better than the heavier payload
- KSC LC 39A had the most successful launches from all site
- We can observe that the success rate launches of SpaceX since 2013 kept increasing till 2020. It means that the success rate of SpaceX launches are directly proportional to time.
- Orbit GEO, HEO, SSO, and ES L1 are the orbit types of a launch with the highest rate

Predictive Analysis Results

- The Logistic Regression, SVM, and KNN models are the best in terms of accuracy for this data set.
- Decision Tree model performed the worst with just .83 accuracy compare to the other models who got a .94 accuracy score.
- All of the models made at least one false positive prediction.

Interactive Analytics Demo in Screenshots

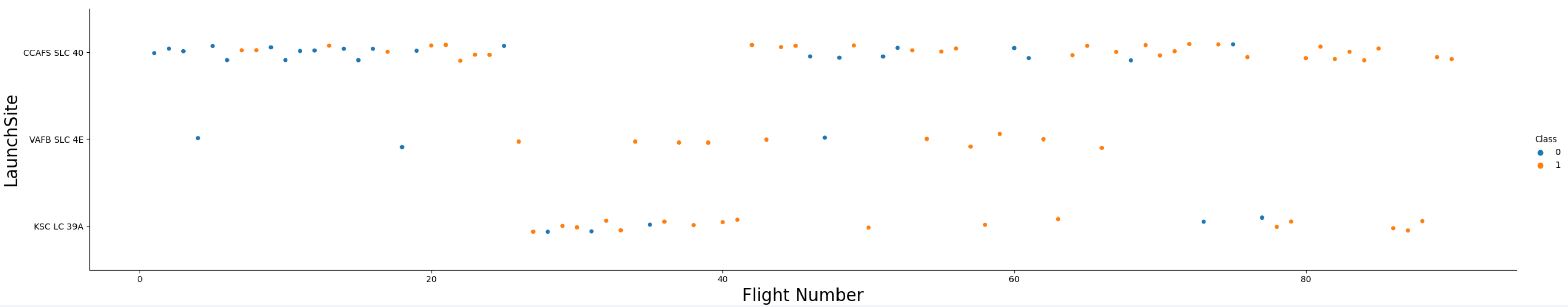




Section 2

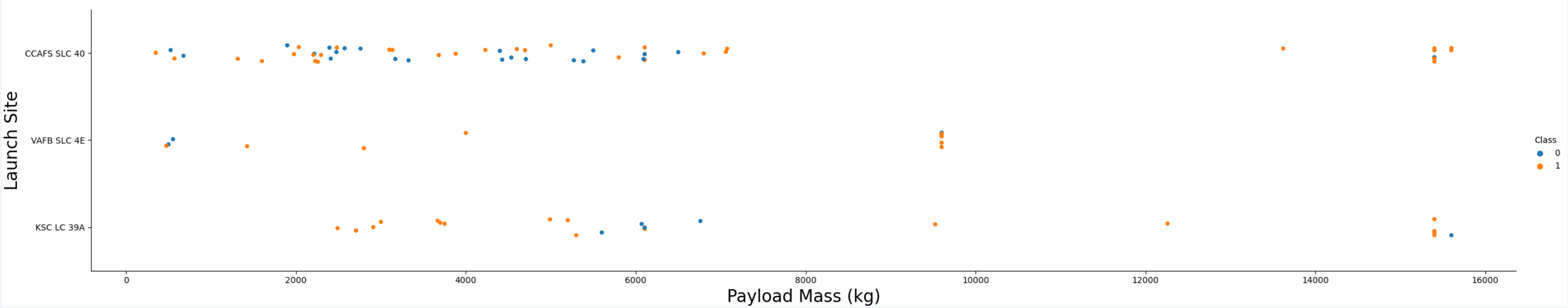
Insights drawn from EDA

Flight Number vs. Launch Site



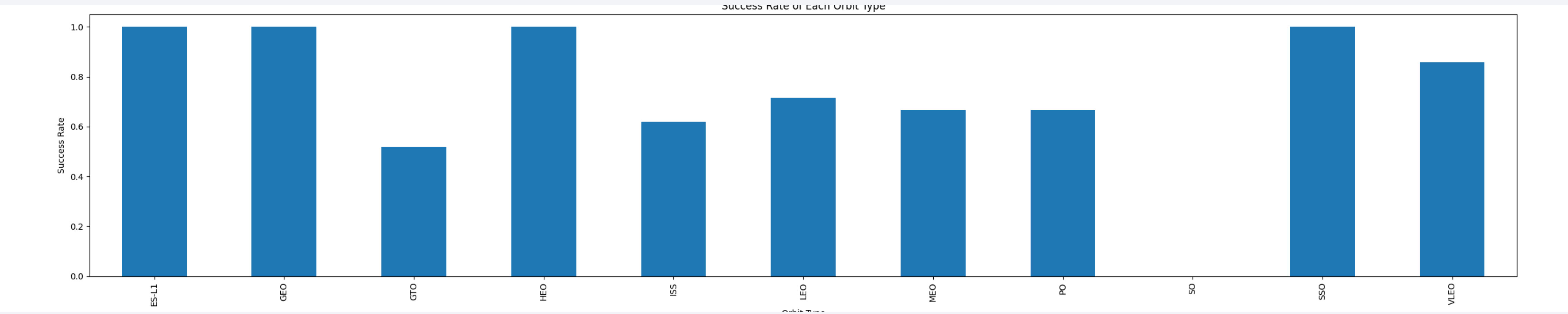
- CCAFS SLC 40 launch site has significantly higher launch count than the other launch sites
- KSC LC 39A had the most successful launches from all site

Payload vs. Launch Site



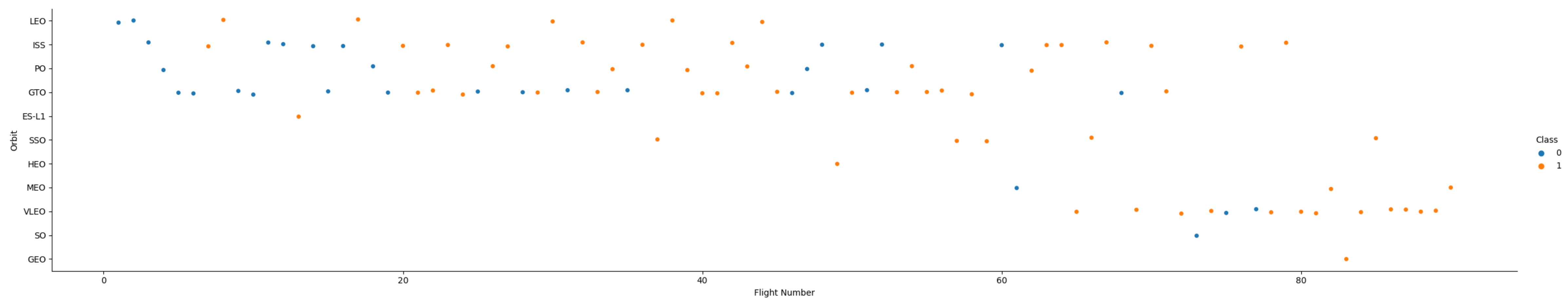
- The majority of payloads with lower mass has been launched from CCAFS SLC 40
- VAFB SLC 4E launch site does not launch rockets with payload higher than 10,000 kg

Success Rate vs. Orbit Type



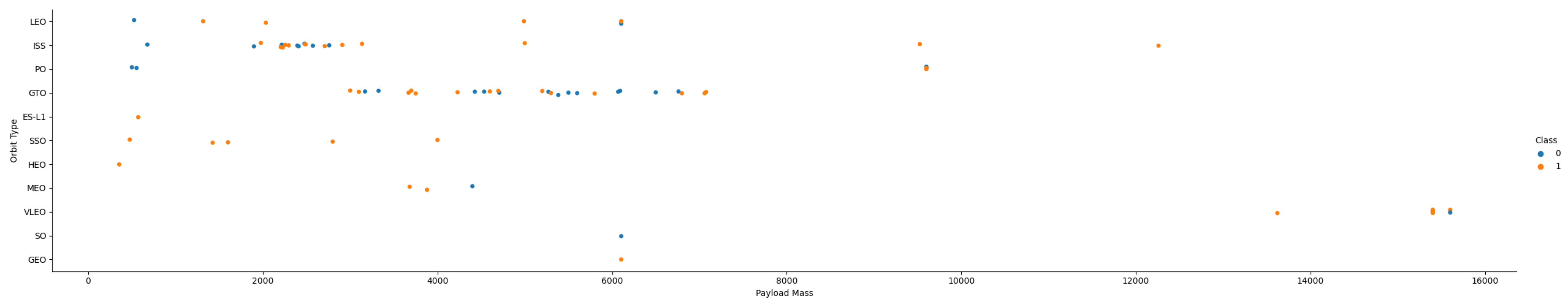
- Orbit GEO, HEO, SSO, and ES L1 are the orbit types of a launch with the highest rate
- SO orbit type launch did not make any successful attempt at landing a rocket

Flight Number vs. Orbit Type



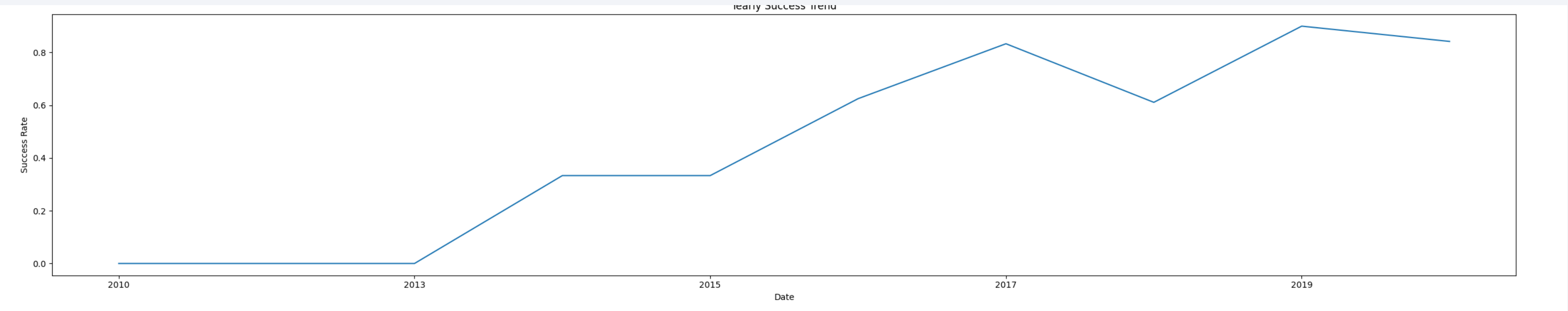
- We should see that in the LEO orbit the success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.
- A trend can be observed of shifting to VLEO launches in recent years

Payload vs. Orbit Type



- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there.

Launch Success Yearly Trend



- We can observe that the success rate since 2013 kept increasing till 2020. It means that the success rate of SpaceX launches are directly proportional to time. This means as time progresses the more the SpaceX company has perfected the landing of the first stage of a launch.

All Launch Site Names

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

```
%%sql
SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;
```

```
 * ibm_db_sa://bgy27221:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

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

- There are only 4 unique launch sites for SpaceX launches of F9 Boosters

Launch Site Names Begin with 'CCA'

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

```
 * ibm_db_sa://bgy27221:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

Out[5]:

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

- 5 records where launch sites begin with the string 'CCA'

Total Payload Mass

```
[9]: %%sql
SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer='NASA (CRS)';

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

[9]:

SUM(PAYLOAD_MASS__KG_)
45596

- The total payload mass carried by boosters launched by NASA (CRS)

Average Payload Mass by F9 v1.1

In [7]:
```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE booster_version LIKE 'F9 v1.1%';
```

```
 * ibm_db_sa://bgy27221:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

Out[7]:

1
2534

- The average payload mass carried by booster version F9 v1.1 in kilogram

First Successful Ground Landing Date

```
In [8]: %sql SELECT MIN(DATE) from SPACEXTBL WHERE LANDING__OUTCOME = 'Success (ground pad)';

 * ibm_db_sa://bgy27221:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

Out[8]:

1
2015-12-22

- The first successful ground landing was in December 22, 2015.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
In [9]: %sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000;

 * ibm_db_sa://bgy27221:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

Out[9]:

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

- This is the list of names of boosters which have successfully landed on drone ship that had payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

In [10]:
```
%sql SELECT MISSION_OUTCOME, COUNT(*) FROM SPACEXTBL GROUP BY MISSION_OUTCOME;
```

```
 * ibm_db_sa://bgy27221:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

Out[10]:

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

- The total number of successful mission outcomes is 100 with one having a unclear status on its payload.
- There is one mission failure during flight.

Boosters Carried Maximum Payload

In [11]:
```
%%sql SELECT BOOSTER_VERSION FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

```
 * ibm_db_sa://bgy27221:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

Out[11]:

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

- This are the list of names of the booster which have carried the maximum payload mass

2015 Launch Records

```
In [12]: %%sql SELECT landing__outcome, booster_version, launch_site, date FROM SPACEXTBL
         WHERE landing__outcome = 'Failure (drone ship)' AND
         YEAR(DATE) = '2015';
```

```
 * ibm_db_sa://bgy27221:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

Out[12]:

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

- The list of failed landing outcomes in drone ship with their booster versions and launch site names in the year 2015.

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

In [13]:

```
%%sql
SELECT LANDING__OUTCOME, COUNT(*) AS qty FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING__OUTCOME
ORDER BY qty DESC;
```

```
 * ibm_db_sa://bgy27221:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

Out[13]:

landing__outcome	qty
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

- The rank of landing outcomes between the date 2010-06-04 and 2017-03-20, based on count and in descending order

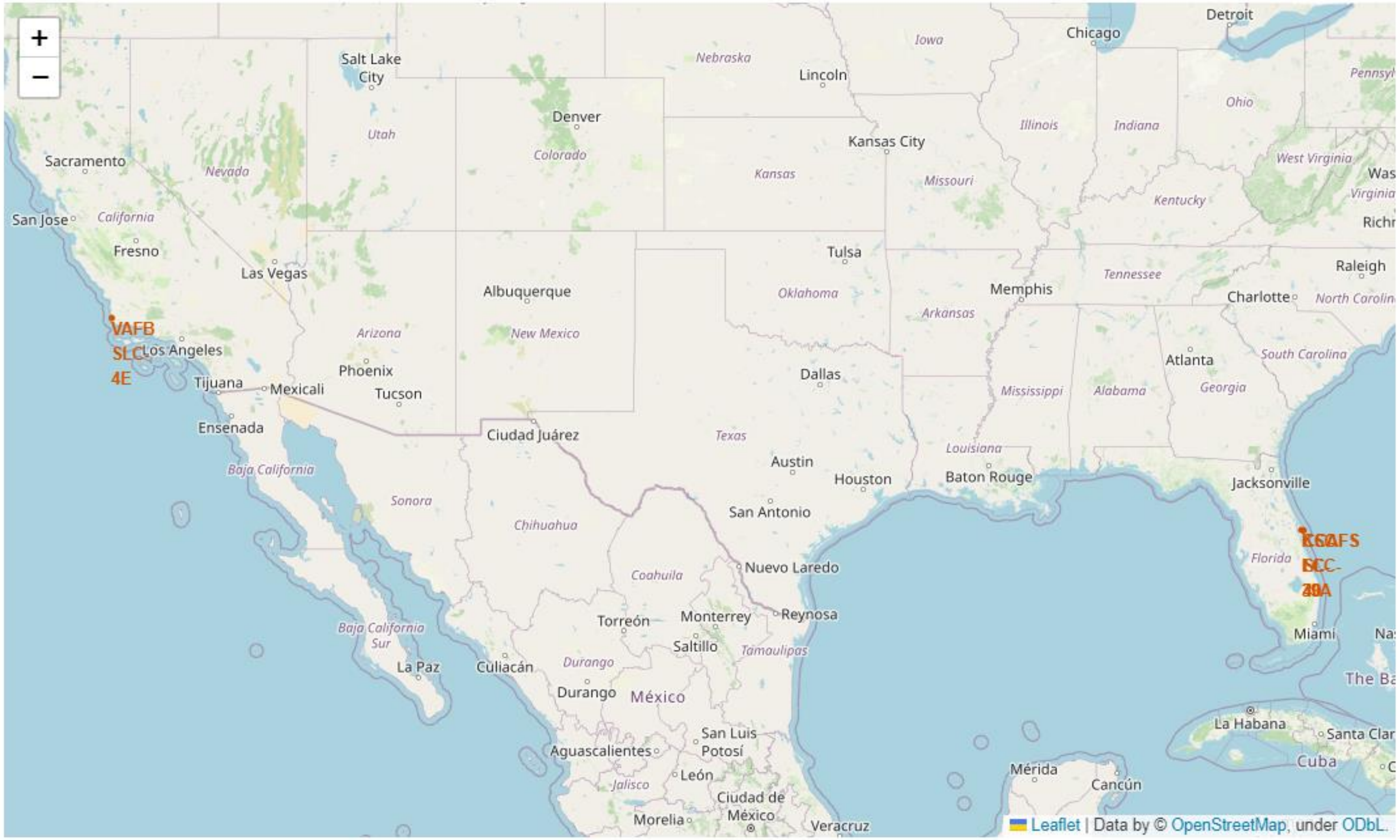
Section 3

Launch Sites Proximities Analysis

Launch Sites Markers

In this screenshot of the map, with the help of putting circles and markers, we can easily navigate that 3 out of 4 launch sites used by SpaceX for their F9 Boosters are located around Florida.

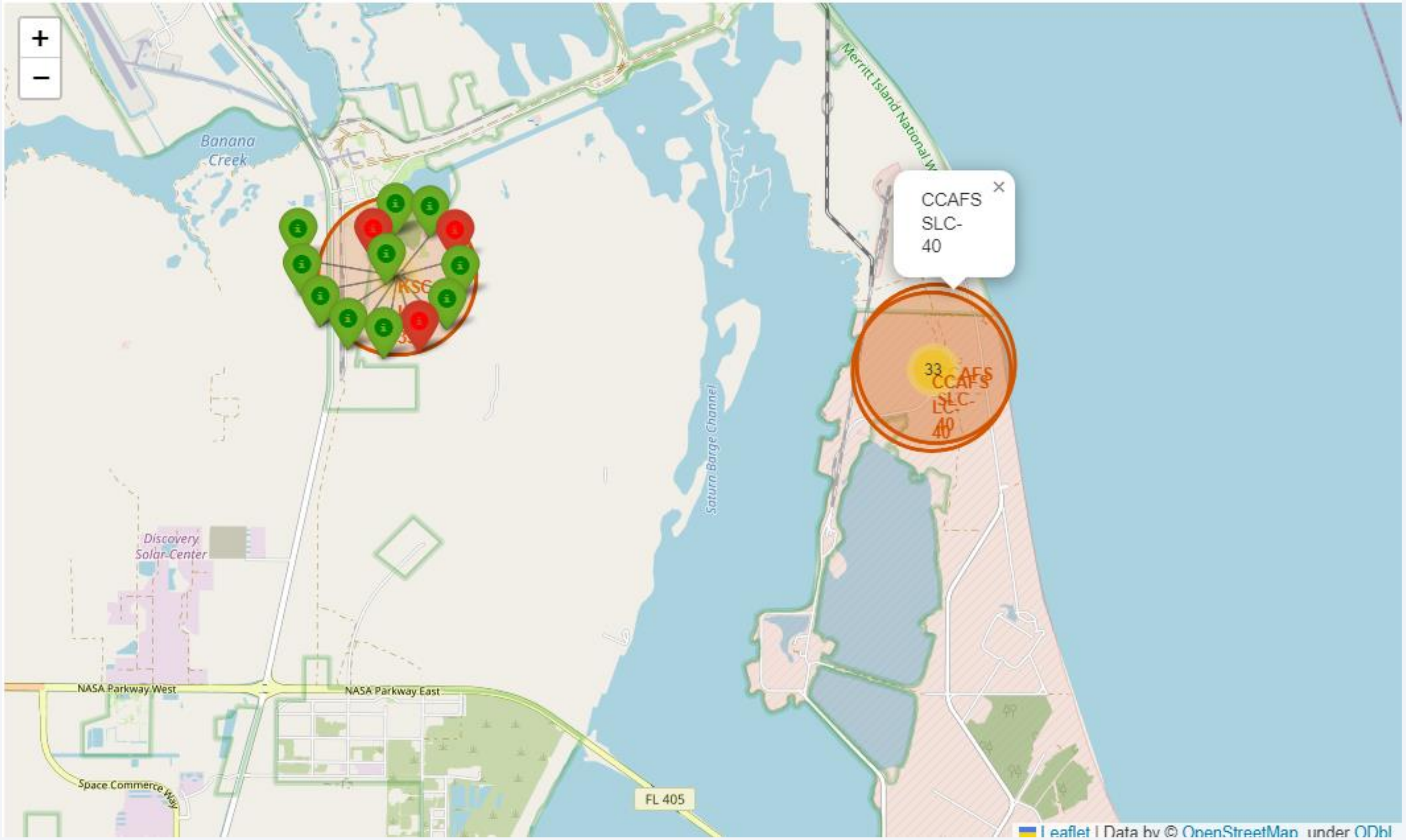
On the other hand VAFB SLC 4E alone is located around California.



Color-labeled Launch Outcomes

If we zoom in on our folium map , we can see the marker clusters that were added to mark the success/failed launches for each site on the map.

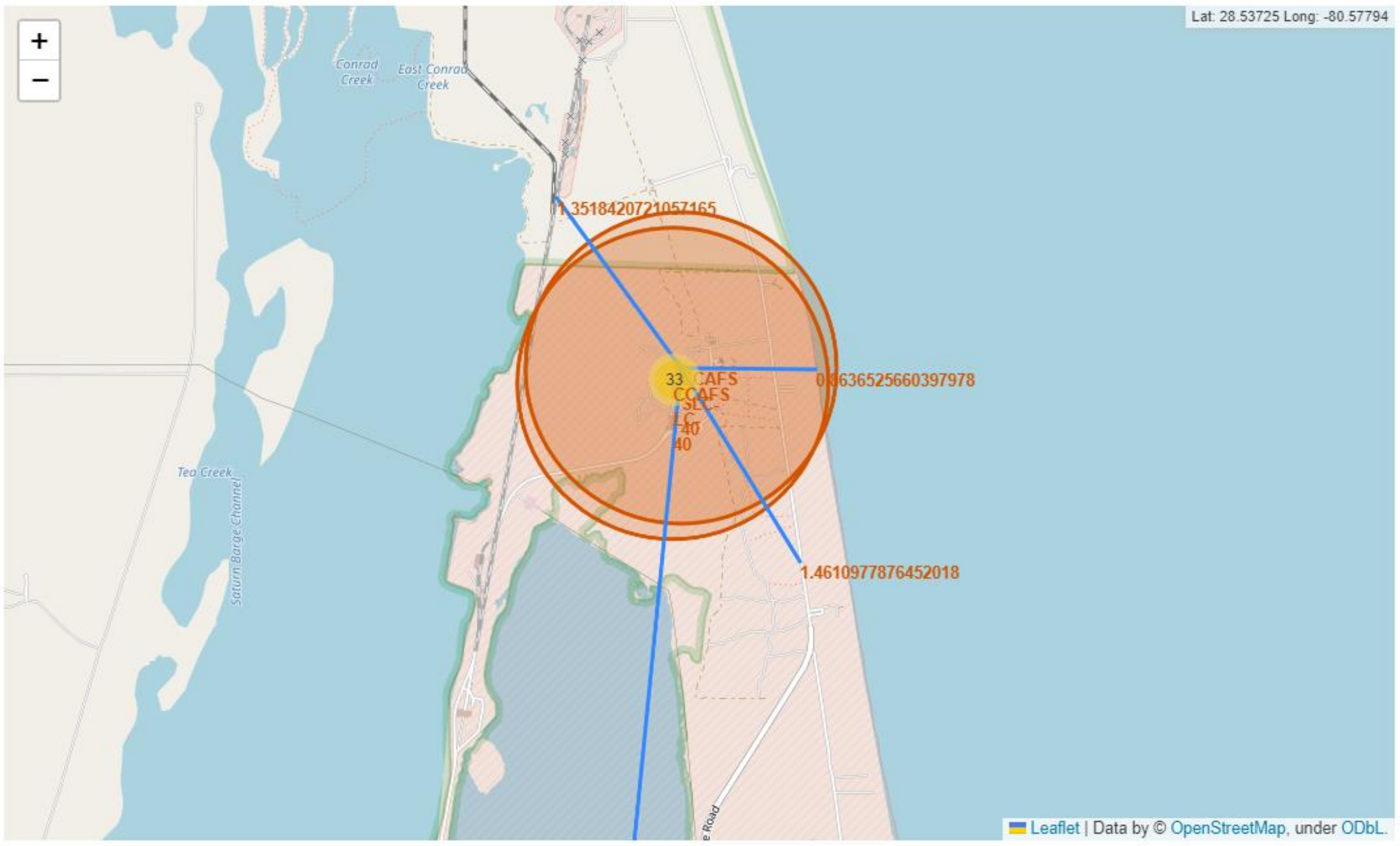
We can also see the pop-up markers that were added to easily select launch sites with close proximity to each other.



Launch Site Proximities

Certain proximities such as coastline, highways, railroads, and cities from a launch site was also calculated. A line object was added and they are marked with their respective calculated distance.

Optimal location for a launch site has certain distances from its proximities.

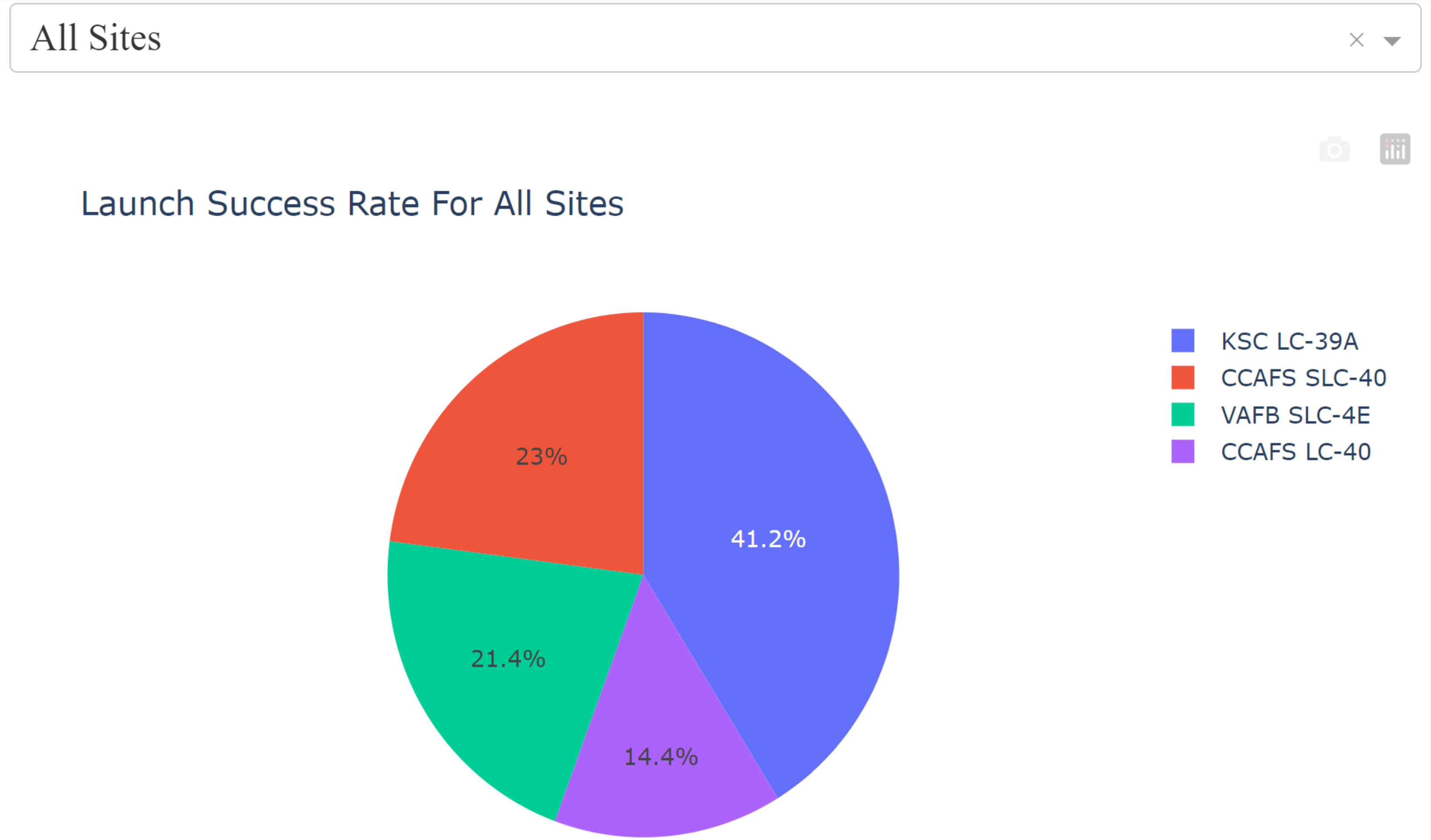


Section 4

Build a Dashboard with Plotly Dash

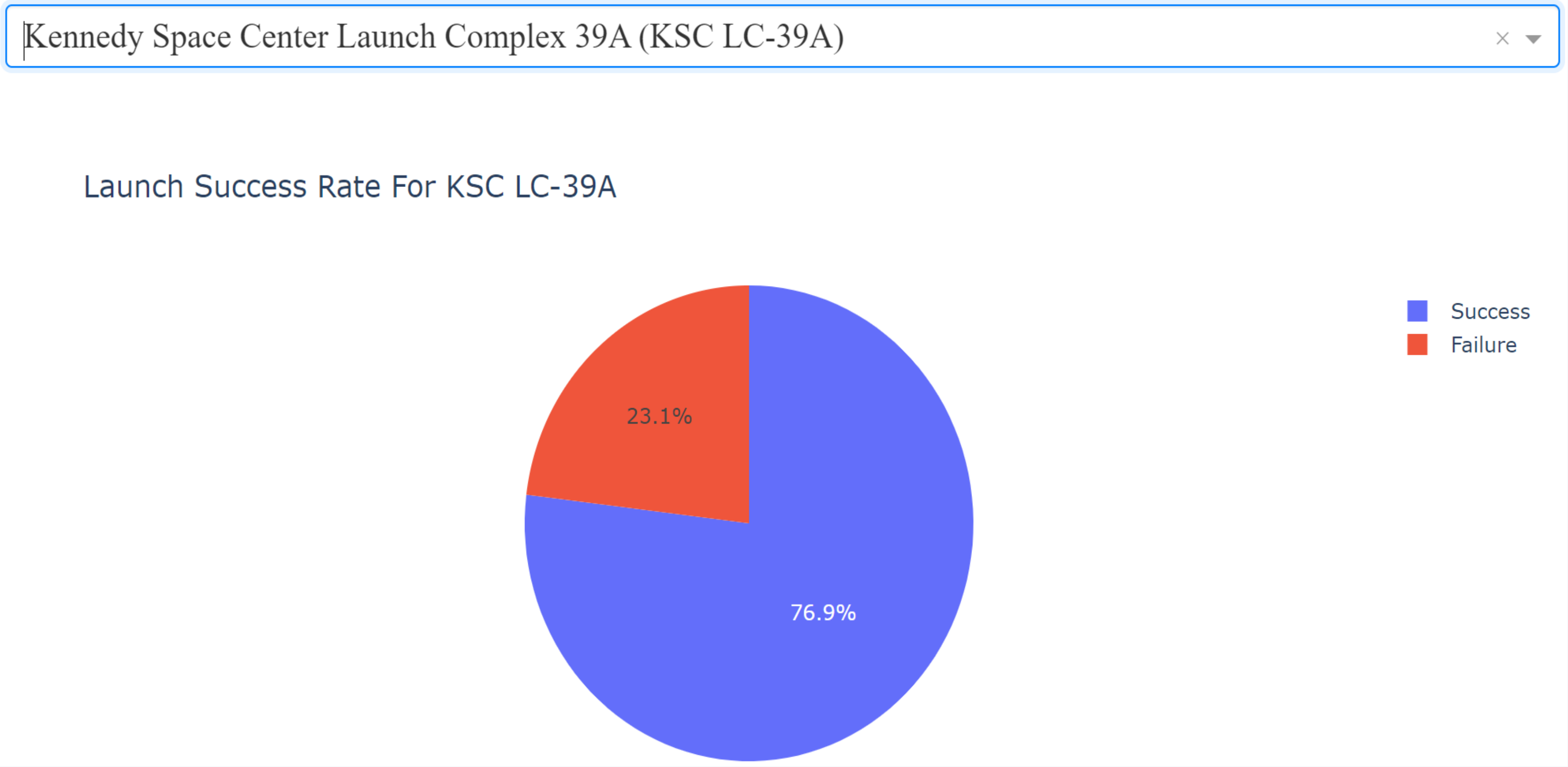
Pie Chart - Launch Success Rate For All Sites

We can easily derive through the chart that KSC LC-39A had the most successful launches among all sites. While on the other hand, CCAFS LC-40 has the lowest success rate with a measly 14.4%.



Pie Chart - Success Rate by Site

Kennedy Space Center Launch Complex 39A had the highest launch success rate ratio with 76.9% success over only 23.1% failure of its launches.



Scatter Plot - Payload vs. Launch Outcome

In figure 1, in the range of 0 to 5k payload, we will notice that the booster version of FT has the highest success rate over the other versions. While on the other hand v1.1 only had 1 success launch even with many attempts.

In figure 2, we can see that in the payload range 5k to 10k there is only two F9 booster versions used: FT and B4. It is also good to notice that most of the launches with heavy payload results to a failure to land

Fig. 1

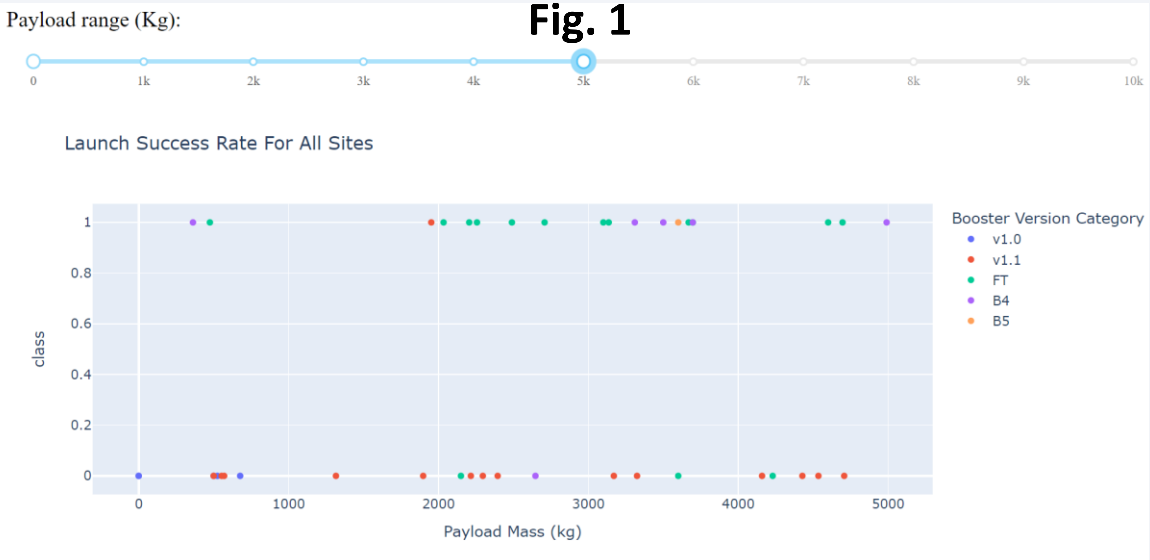
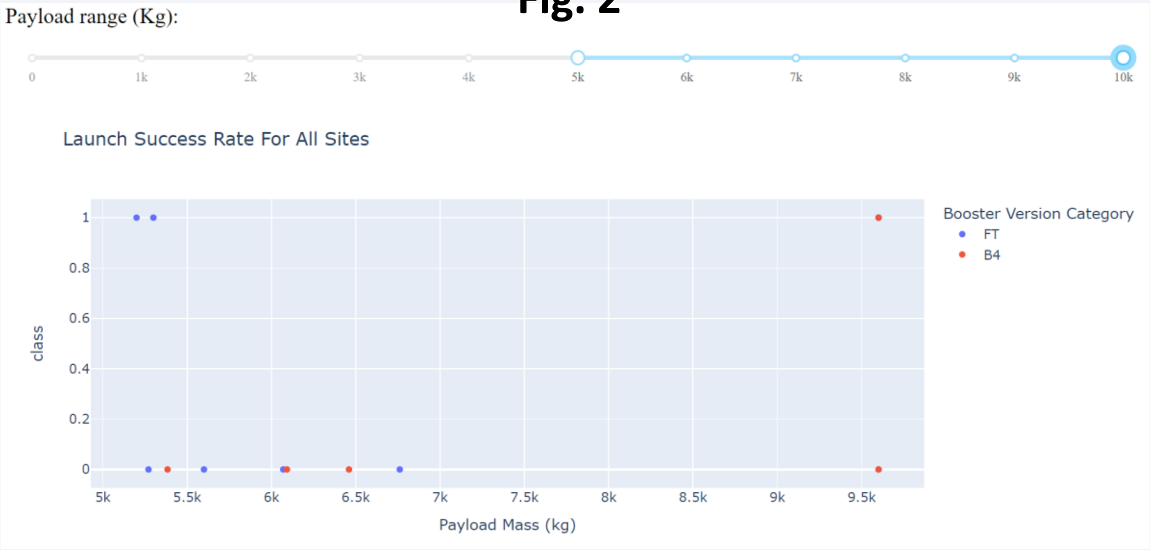


Fig. 2



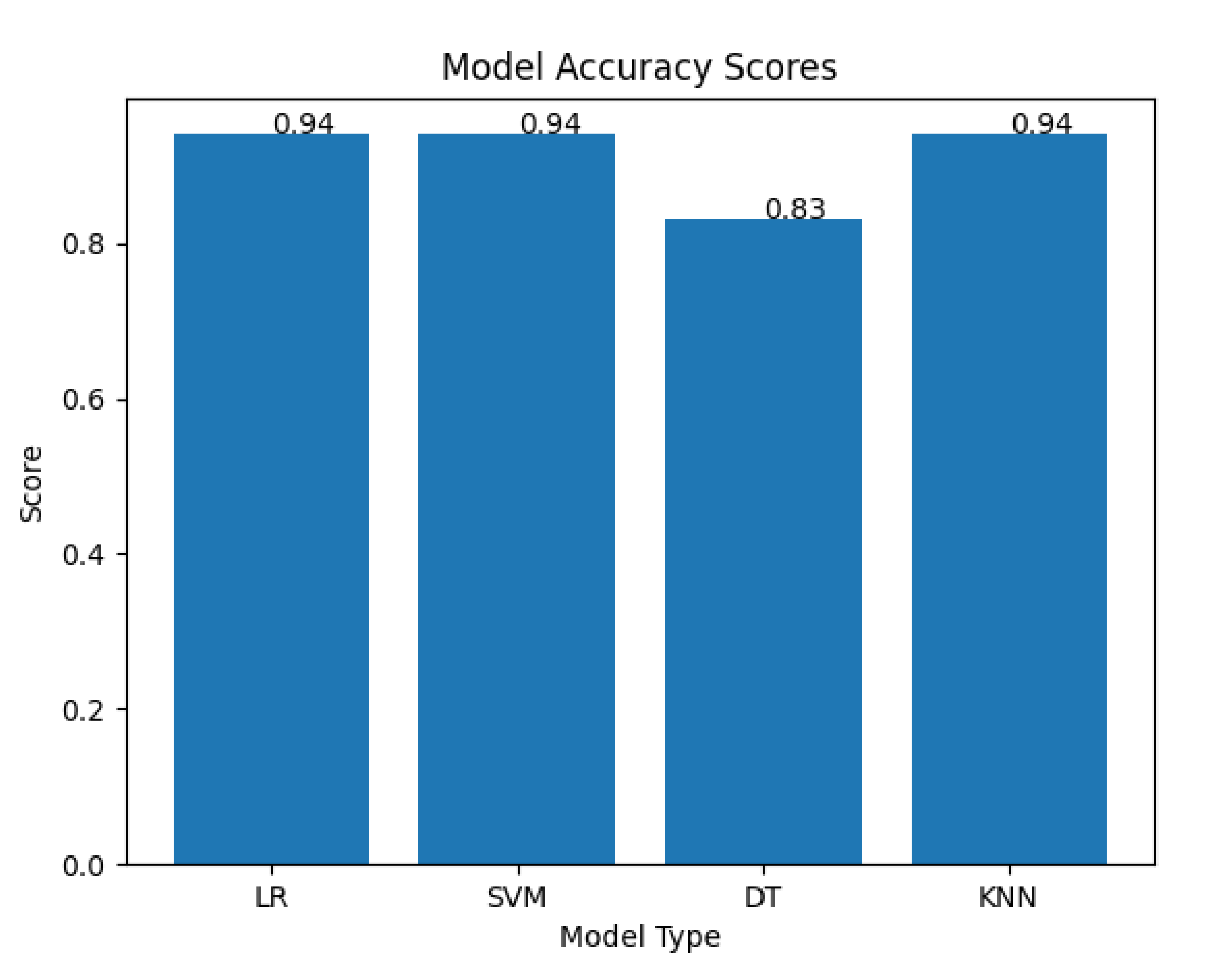
Section 5

Predictive Analysis (Classification)

Classification Accuracy

Logistic Regression, SVM, and KNN models are the best in terms of accuracy for this data set.

Decision Tree model performed the worst with just .83 accuracy compare to the other models who got a .94 accuracy score.

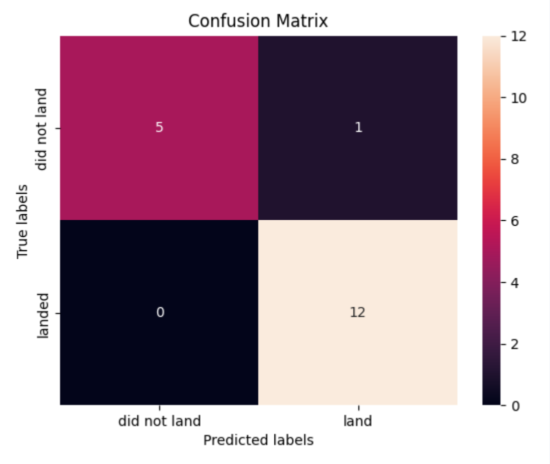


Confusion Matrix

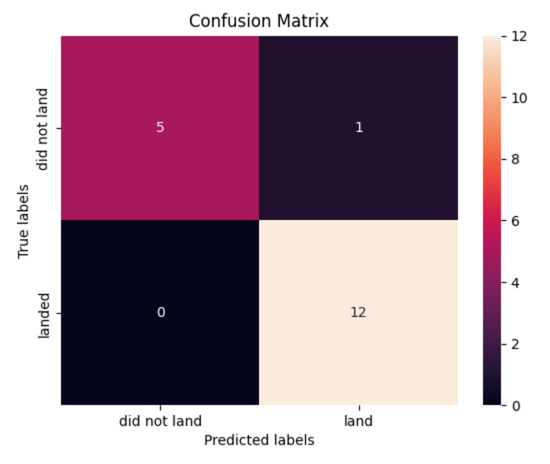
All of the models made at least one false positive prediction. Decision Tree have the worst prediction of 3 false positive affecting its accuracy.

The best models are LR, SVM, and KNN that made almost a 100% correct prediction.

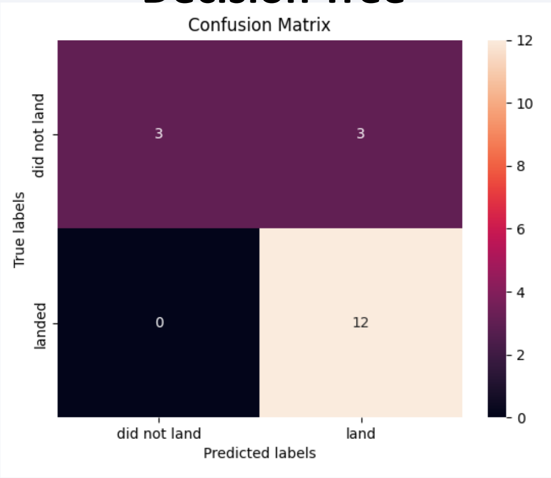
Logistic Regression



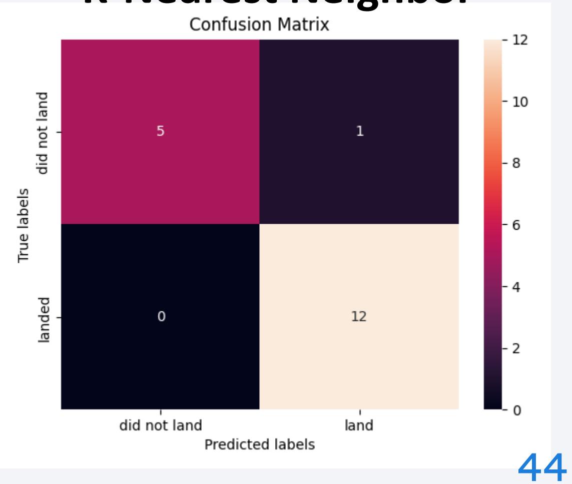
Support Vector Machine



Decision Tree



K-Nearest Neighbor



Conclusions

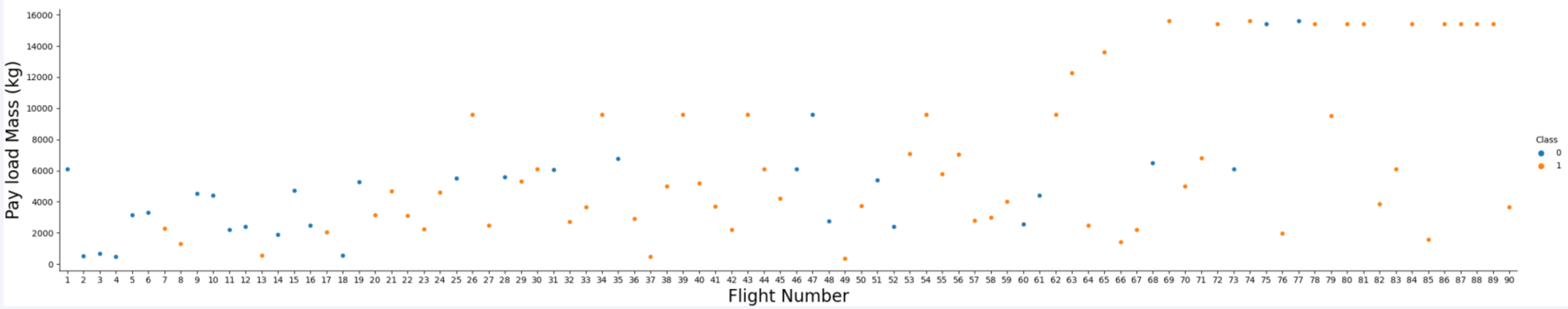
- We can observe that the success rate of SpaceX launches since 2013 kept increasing till 2020. It means that the success rate of SpaceX launches are directly proportional to time.
- Launches from Kennedy Space Center Launch Complex 39A had the highest launch success rate ratio with 76.9% success over only 23.1% failure of its launches.
- Orbit GEO, HEO, SSO, and ES L1 are the orbit types of a launch with the highest rate
- Low weighted payload performs better than the heavier payload.
- The Logistic Regression, Support Vector Machine, and K-Nearest Neighbor models are the best in terms of accuracy for this data set.

Appendix

- Payload mass null values were replaced with the mean of all Payload Mass values.

```
# Calculate the mean value of PayloadMass column
PLMmean = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan,PLMmean,inplace=True)
data_falcon9.isnull().sum()
```

- Flight Number vs. Payload Mass - We can notice on this graph that in recent time, heavy payload has also increased its success rate in general.



- GitHub Link for my whole Capstone Project – https://github.com/danwithcode/Capstone-Project/blob/master/Predictive%20Analysis.ipynb

Thank you!