IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Wenjuan Bian>
<07-12-2022>

Outline

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

Executive Summary

- Summary of methodologies

The records of SpaceX launches was collected from SpaceX REST API or Wikipedia (using webscraping). First, the raw records were parsed and converted into a pandas data frame for data analysis and features engineering. Matplotlib and Folium were used to draw insights. Second, some features were selected, and the corresponding data were standardized and split into training dataset and test dataset. The predictive models were built and tested. Lastly, a Plotly Dash application was built for users to perform interactive visual analytics on SpaceX launch data in real-time.

Executive Summary

- Summary of all results
- Different launch sites have different success rates. CCAFS LC-40 has a success rate of 26.9 %, while KSC LC-39A has a success rate of 76.9%.
- The more massive the payload, the less likely the first stage will return.
- The sucess rate since 2013 kept increasing till 2020.
- Three models were built to predict if the first stage will land successfully and the best hyperparameter for SVM, Classification Trees and Logistic Regression were found. For the test data set, the accuracies for Logistics Regression, Support Vector Machine and K nearsdt neighbors are 0.8334, while the accuracy for Decision tree method is 0.7778.
- A Plotly Dashboard was built to perform interactive visual analytics on SpaceX launch data in real-time. This dashboard contains input components such as a dropdown list and a range slider to interact with a pie chart and a scatter point chart.

Introduction

- Project background and context

  Falcon 9 rocket launches with a cost of 62 million dollars by reusing the first stage. However, other providers cost upward of 165 million dollars each launch. If we can determine if the first stage will land, we can determine the cost of a launch, which can be used if an alternate company wants to bid against SpaceX for a rocket launch. In this project, the data of SpaceX launches were collected from API and Wikipedia. The data were cleaned, reformatted and standardized to build predictive models. The models were trained and tested for predicting if the Falcon 9 first stage will land successfully. In addition, interactive visual analytics were performed to build a

- Problems you want to find answers

  - What factors can impact successful landing of Falcon 9 first stage?
  - From historical data of previous launches, can the landing outcome of future launches be predicted?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data were collected by get request from SpaceX API and WebScrapping from Wikipedia.
- Perform data wrangling
  - The outcomes of landing were converted into Training labels with 1 means the booster successfully landed and 0 means it was unsuccessful
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Classification models were built, trained and tested for predicting the outcome of first stage through Support Vector Machine, Decision Tree, Logistic Regression and K Nearest

Data Collection

- Data was sourced in two ways:
- 1. SpaceX REST API.

•The API provides data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome. The data can be requested from SpaceX API with an URL

•The SpaceX REST API URL is "https://api.spacexdata.com/v4/launches/past".

- 2. Web Scraping

•The Wikipedia page containing information about Falcon 9 launches. The information was extracted by using BeautifulSoup from its Wikipedia URL.

•The address is 1027686922

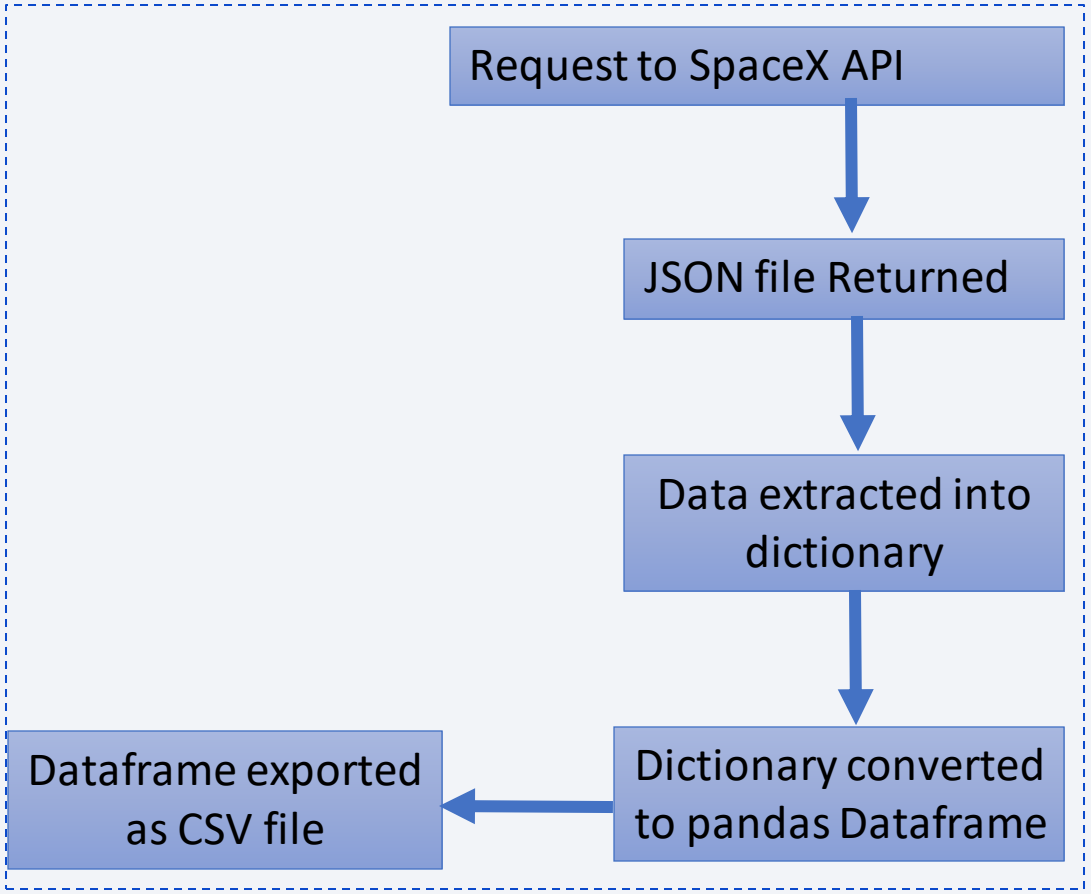
Data Collection – SpaceX API

•The SpaceX REST API call returned data in JSON format, the returned data was passed into a dictionary, and then the dictionary was converted to a Pandas dataframe, which was subsequently exported as a CSV flat file.

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

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

•The notebook prepared for this procedure can be accessed via this github Link: https://github.com/wbian2022/coursera_capstone_project/blob/d3adacc5fffd535ce5449a1f0ce3f13295b1e8dd/jupyter-labs-spacex-data-collection-api%20_Wen.ipynb

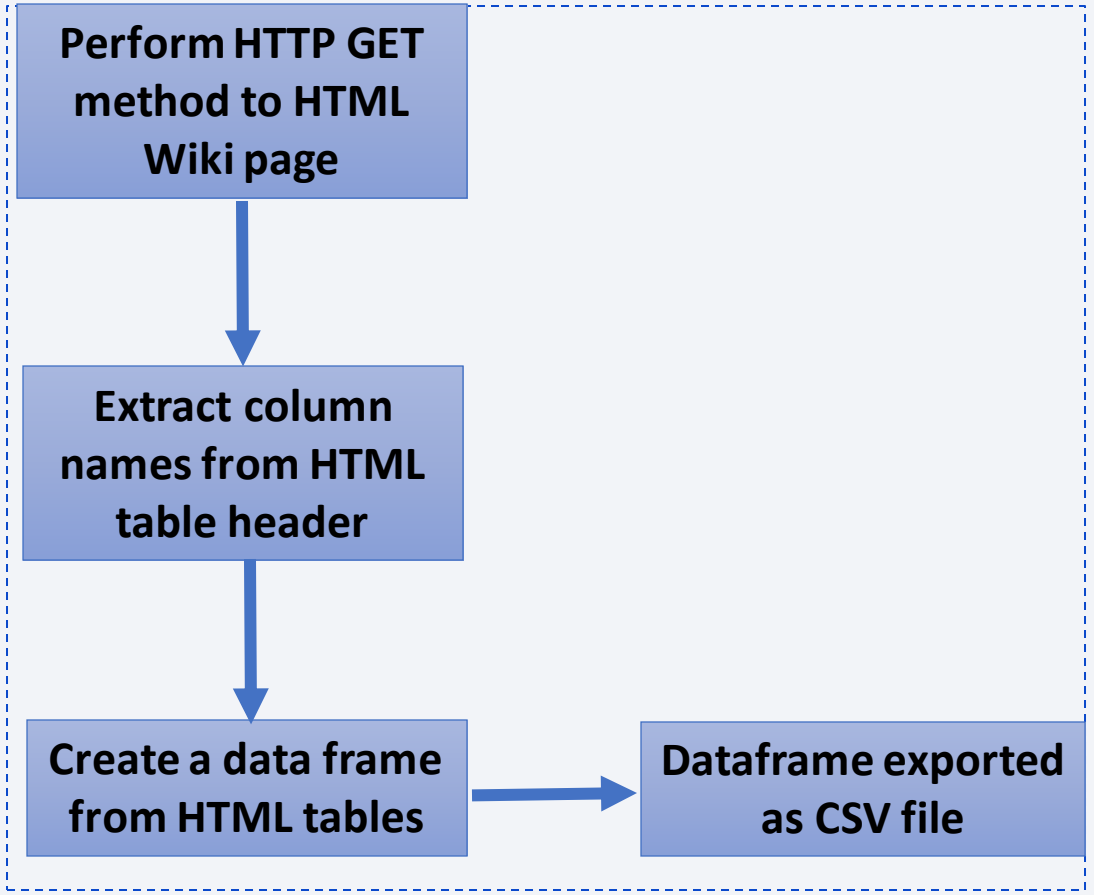


Data Collection - Scraping

- Create a Beautifulsoup object from HTML response, extract and collect all relevant columns from the HTML table header, and then create a data frame by parsing the launch HTML tables.

- The notebook for this procedure can be accessed via the github Link:

https://github.com/wbian2022/coursera_capstone_project/blob/master/jupyter-labs-webscraping_wen.ipynb



Data Collection – Scraping

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

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

```
html_tables = soup.find_all("table")
print(html_tables)
```

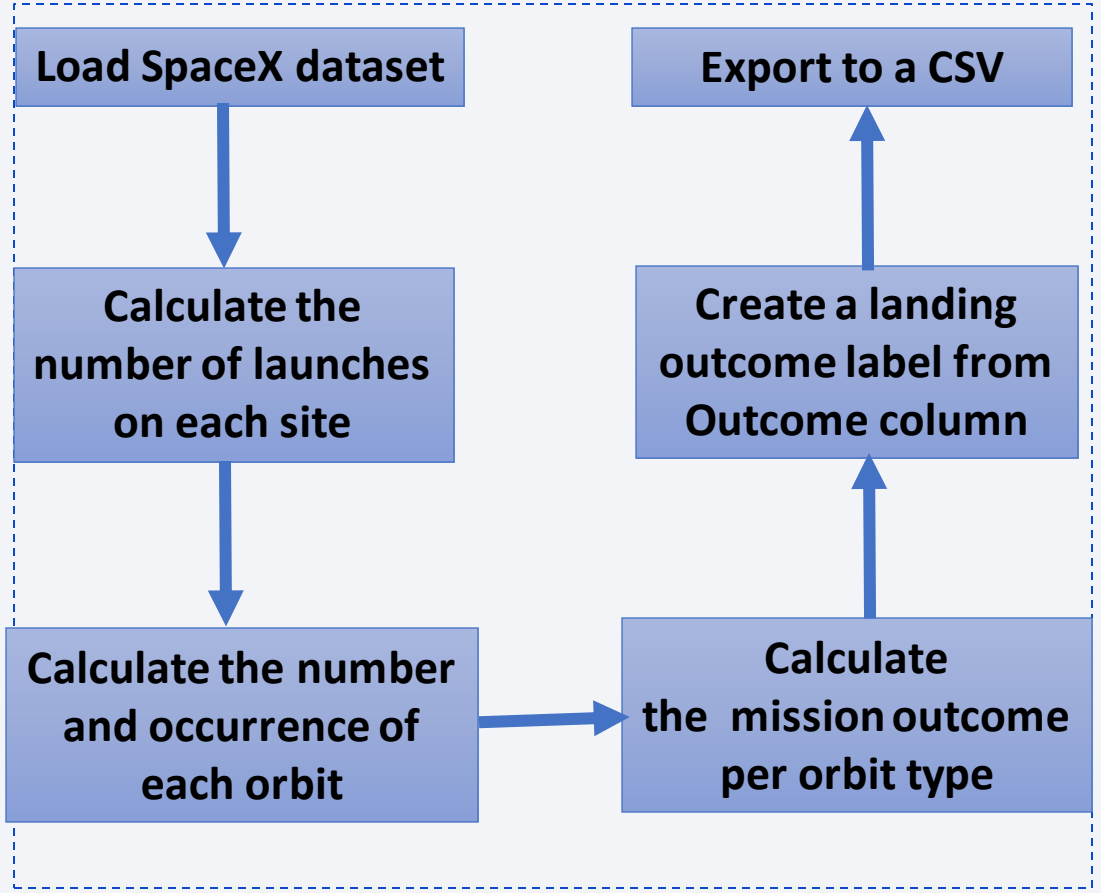
```
column_names = []

# Apply find_all() function with `th` element on
# Iterate each th element and apply the provided
# Append the Non-empty column name (`if name is n

temp = soup.find_all('th')
for x in range(len(temp)):
    try:
        name = extract_column_from_header(temp[x])
        if (name is not None and len(name) > 0):
            column_names.append(name)
    except:
        pass
```

Data Wrangling

- The data was loaded from the data collection section.
- Some calculations were performed by method value_counts () .
- Landing outcome label was created
- The data was exported and stored as a CSV file
- The GitHub URL is https://github.com/wbian2022/coursera_capstone_project/blob/e1791385fbea5c9e44dfbcb0b24dccc17cf523c2/labs-jupyter-spacex-Data%20wrangling%20-%20Wen.ipynb



Data Wrangling - Code

```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = df['Outcome'].replace({'False Ocean': 0, 'False ASDS': 0, 'None Nor
df['Outcome'] = df['Outcome'].astype(int)
df.info()
```

EDA with Data Visualization

Visualized charts were plotted out to see the relationships between variables and how different variables would affect the launch outcome

- FlightNumber vs. PayloadMass and the outcome of launch
- FlightNumber vs. LaunchSite and the outcome of launch
- FlightNumber vs. Orbit type and the outcome of launch (Bar chart was created to see the relationship between success rate of each orbit type)
- PayloadMass vs. Orbit type and the outcome of launch
- Launch success yearly trend

- The GitHub URL of the EDA is:

https://github.com/wbian2022/coursera_capstone_project/blob/ca2f32c418dc4bf8f60441a28081e4bb15c3519f/jupyter-labs-eda-dataviz_wen.ipynb

EDA with SQL

SQL queries performed:

- The names of the unique launch sites in the space mission
- Five records where launch sites begin with the string 'CCA'
- The total payload mass carried by boosters launched by NASA (CRS)
- The average payload mass carried by booster version F9 v1.1
- The first successful landing outcome in ground pad was achieved
- The names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- The names of the booster_versions with maximum payload mass
- The month of failure landing of drone ship in the year 2015
- The count of successful landing between 04-06-2010 and 20-03-2017 was ranked

EDA with SQL

- The GitHub URL of the EDA with SQL notebook is :

  https://github.com/wbian2022/coursera_capstone_project/blob/master/jupyter-labs-eda-sql-coursera_sqllite_Wen.ipynb

Build an Interactive Map with Folium

- The launch success rate may depend on the location and proximities of a launch site. In this section, all the launch sites were marked and the success/failed launches for each site were marked by Folium
  - All the launch sites were marked with circles by folium.Circle, and labeled with the site name by folium.Marker,
  - The success/failed launches for each site were marked by MarkCluster
  - The distance between coast line point and a launch site was calculated marked by folium.Marker and the closest coastline was marked
  - Lines were drew between launch site to its closest city, railway, highway and the distances were calculated and marked in the map by folium.Marker.
- The GitHub URL is:

  https://github.com/wbian2022/coursera_capstone_project/blob/master/lab_jupyter_launch_site_location_Wen1.ipynb

Build a Dashboard with Plotly Dash

- The plotly dash includes
  - ➢ Launch site drop-Down input

    There are four different launch sites and we would like to first see which one has the largest success count. Then, we would like to select one specific site and check its detailed success rate. Dropdown menu let us select different launch sites
  - ➢ Callback function to render success-pie-chart based on selected site dropdown

    The callback function is to get the selected launch site from site-dropdown and render a pie chart visualizing launch success counts.
  - ➢ Range slider to select payload

    Range slider can help us to find if variable payload is correlated to mission outcome
  - ➢ Call back function to render success-payload-scatter-chart scatter plot

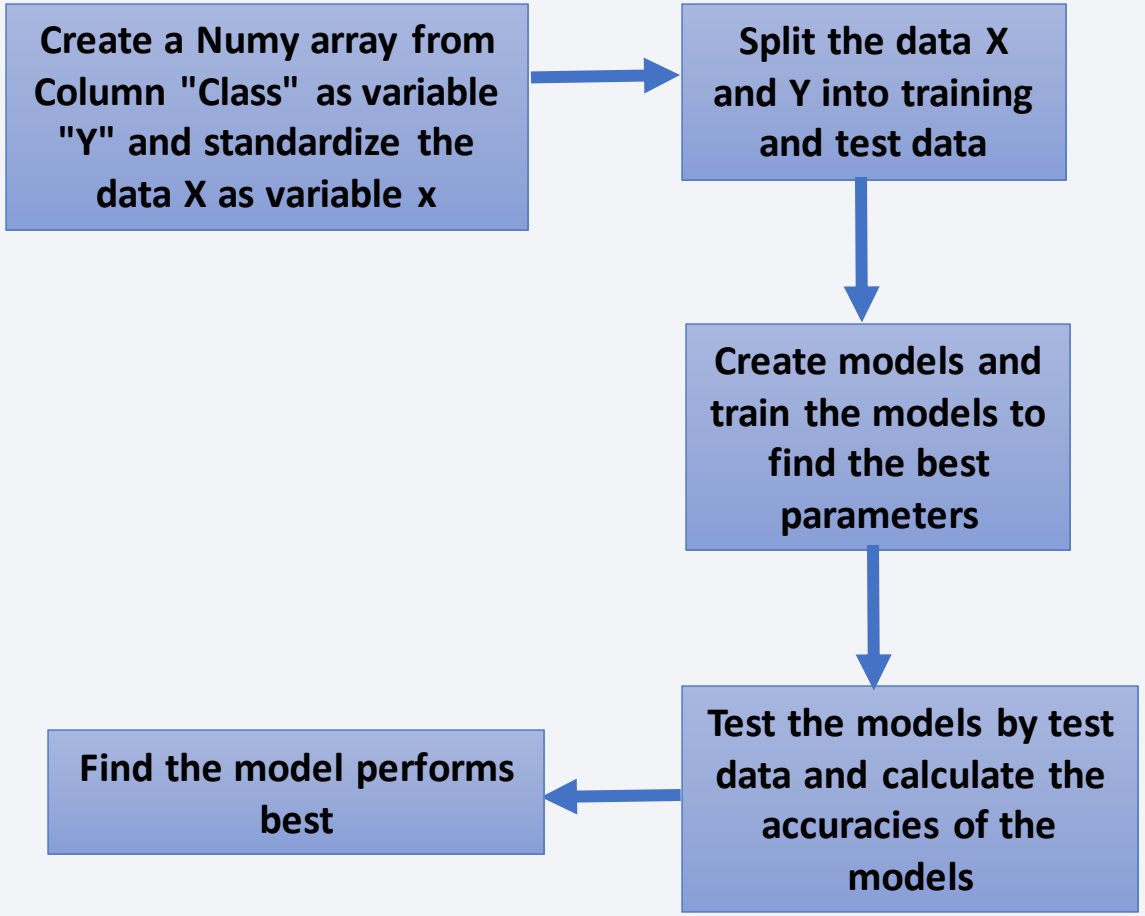
    By this function, we can visually observe how payload may be correlated with mission outcomes for selected site(s)
- The GitHub URL of the Plotly Dash lab is:

  https://github.com/wbian2022/coursera_capstone_project/blob/master/spacex_dash_app_wen.py

Predictive Analysis (Classification)

- Predictive models were built by Logistics Regression, Support Vector Machine, Decision Tree and K Nearest Neighbors
- The models were trained to find the best parameters, and were evaluated by test data.
- The GitHub URL for the predictive analysis is:

  https://github.com/wbian2022/coursera_capstone_project/blob/33ff2043a9c305bb875a2b8b1658c9d46316db63/SpaceX_Machine%20Learning%20Prediction_Part_5_Wen.ipynb

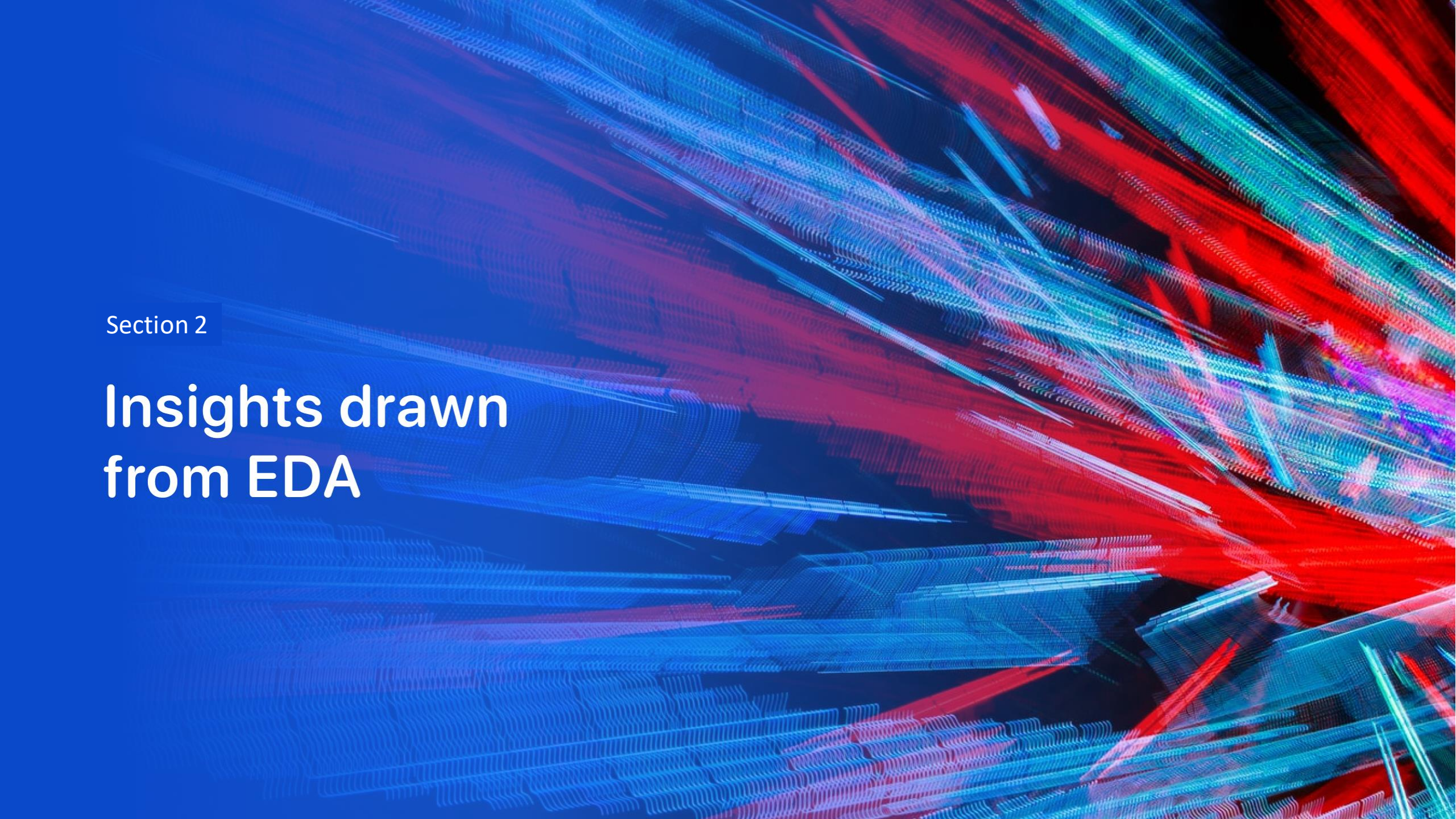


Results

- Exploratory data analysis results:
  - Different launch sites have different success rates. CCAFS LC-40 has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.
  - The more massive the payload, the less likely the first stage will return.
  - The sucess rate since 2013 kept increasing till 2020.
- Predictive analysis results
  - The accuracy for Logistics Regression, Support Vector Machine and K Nearest Neighbors is 0.8334, with accuracy for decision tree is 0.7778

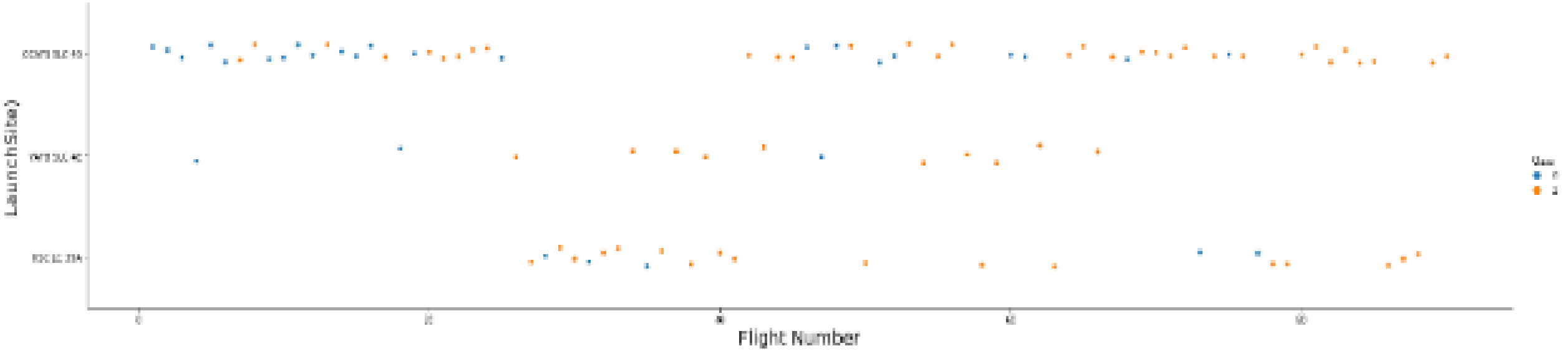
Section 2

Insights drawn from EDA



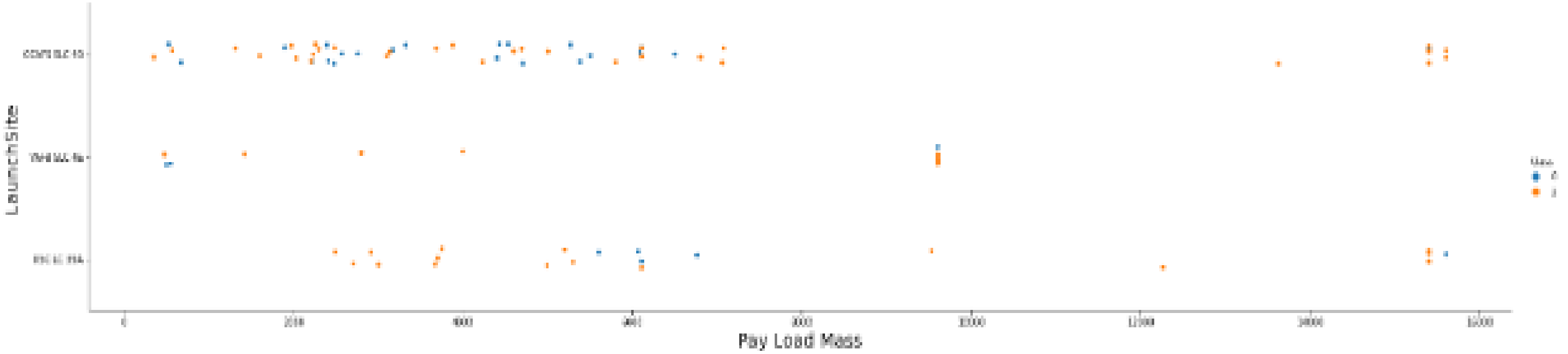
Flight Number vs. Launch Site

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



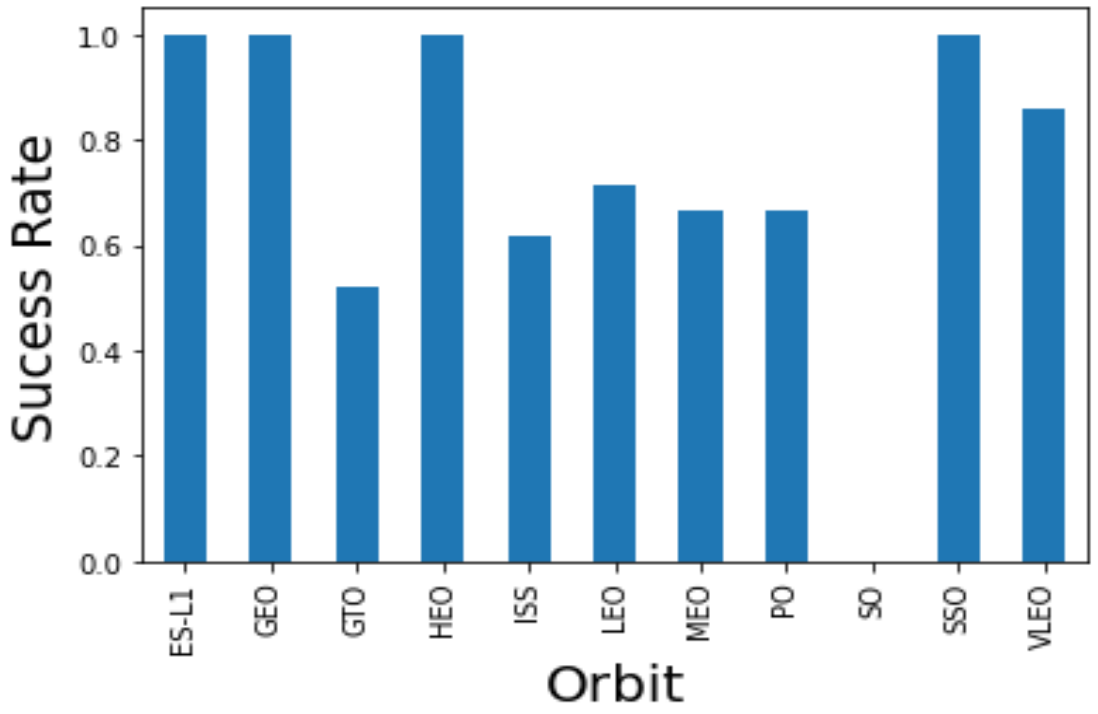
Payload vs. Launch Site

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



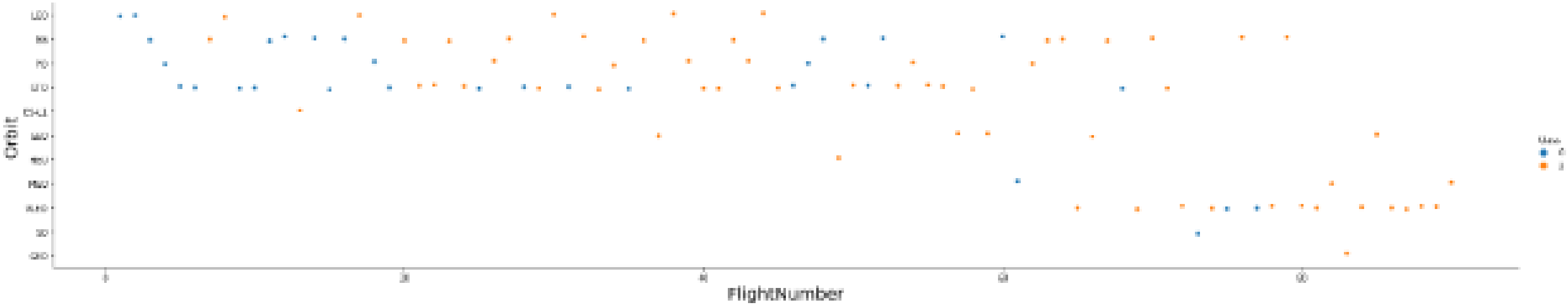
Success Rate vs. Orbit Type

```
# HINT use groupby method on Orbit column and get the mean of Class column
df.groupby(['Orbit']).mean()['Class'].plot(kind='bar')
plt.xlabel("Orbit",fontsize=20)
plt.ylabel("Sucess Rate",fontsize=20)
plt.show()
```



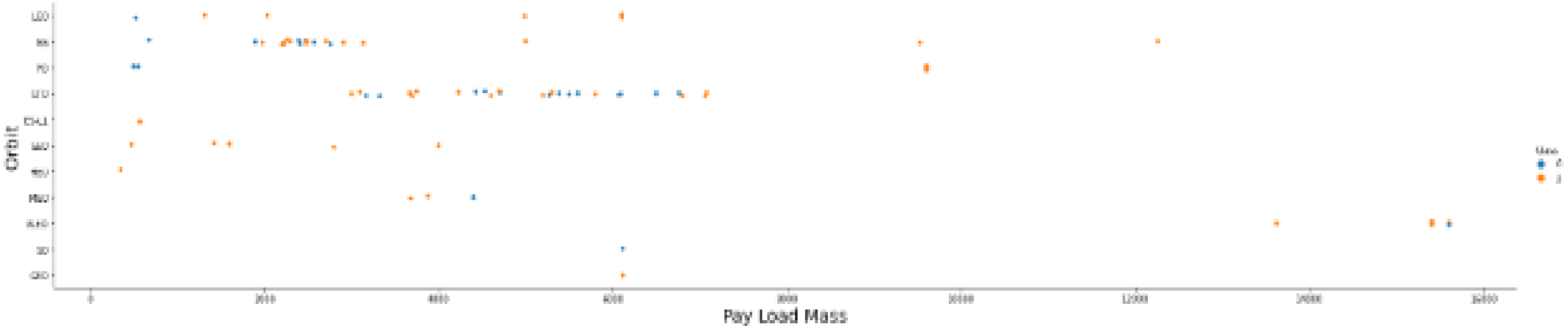
Flight Number vs. Orbit Type

```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbi
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

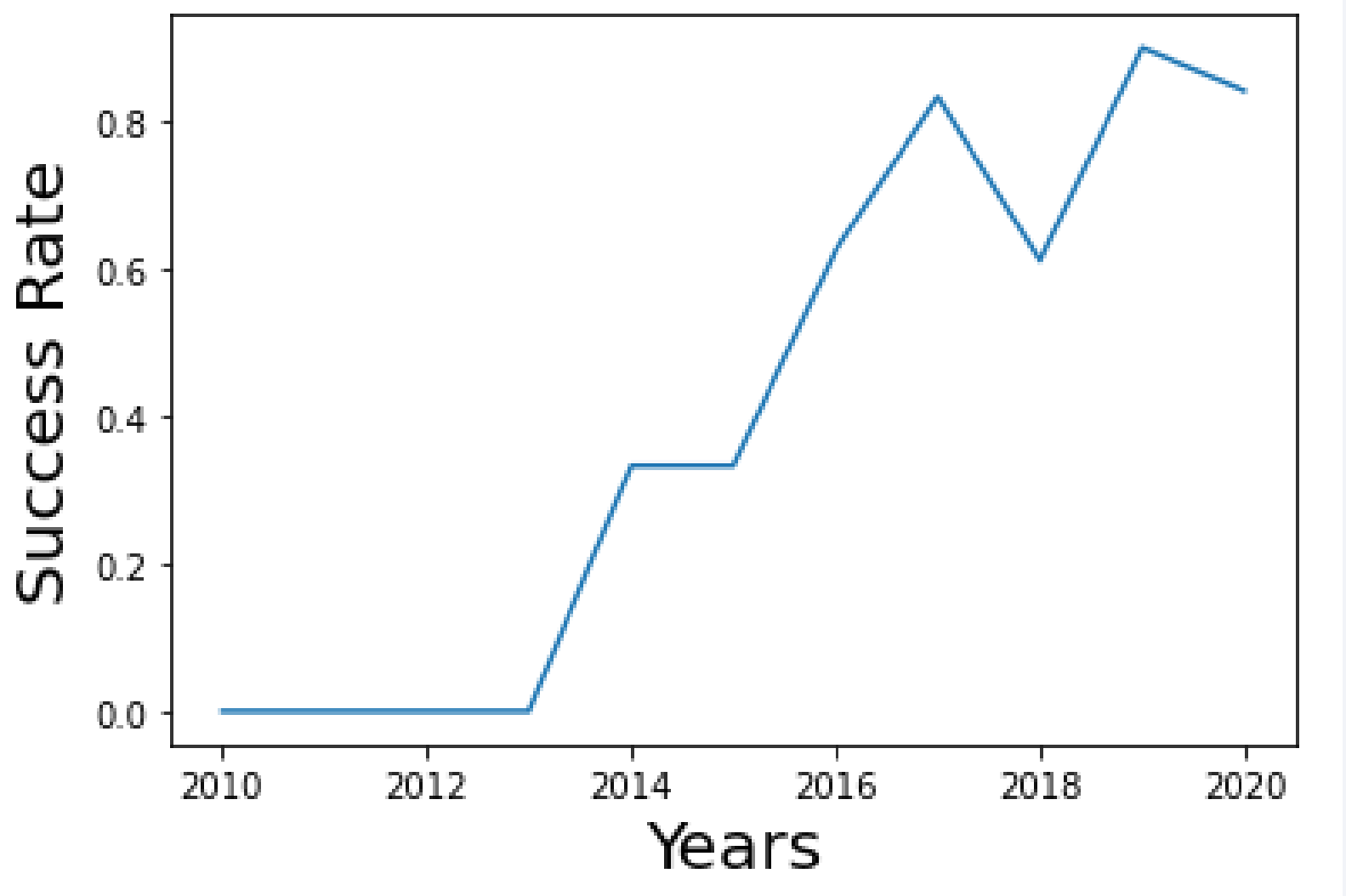


Payload vs. Orbit Type

```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Pay Load Mass",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Launch Success Yearly Trend



Launch Success Yearly Trend

```
# A function to Extract years from the date
year=[]
def Extract_year(date):
    for i in df["Date"]:
        year.append(i.split("-")[0])
    return year
```

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success
df['Year'] = pd.DataFrame(Extract_year(df['Date'])).astype('int')
sns.lineplot(x = df['Year'].unique() , y = df.groupby(['Year'])['Class'].mean())
plt.xlabel("Years",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```

All Launch Site Names

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

```
%sql select distinct(LAUNCH_SITE) from SPACEXTBL
```

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

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

Launch Site Names Begin with 'CCA'

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

```
%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

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

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

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

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

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

avg(PAYLOAD_MASS__KG_)
2928.4

The average payload mass by F9 v1.1 is 2928.40

First Successful Ground Landing Date

```
%sql select min(Date) from SPACEXTBL where "Landing _Outcome" = "Success (ground pad)"
```

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

min(Date)
01-05-2017

The first successful ground landing is Jan 5th, 2017

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select BOOSTER_VERSION from SPACEXTBL
where  "Landing_Outcome" = 'Success (drone ship)'
and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
```

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

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

Total Number of Successful and Failure Mission Outcomes

```
sql select count(MISSION_OUTCOME) from SPACEXTBL
where MISSION_OUTCOME = 'Success'
or MISSION_OUTCOME = 'Failure (in flight)'
```

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

count(MISSION_OUTCOME)
99

Boosters Carried Maximum Payload

```
%sql select BOOSTER_VERSION from SPACEXTBL where
PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

2015 Launch Records

```
%sql SELECT substr(Date, 4, 2) from SPACEXTBL where
"Landing_Outcome" = 'Failure (drone ship)' and substr(Date,7,4)='2015'
```

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

substr(Date, 4, 2)
01
04

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

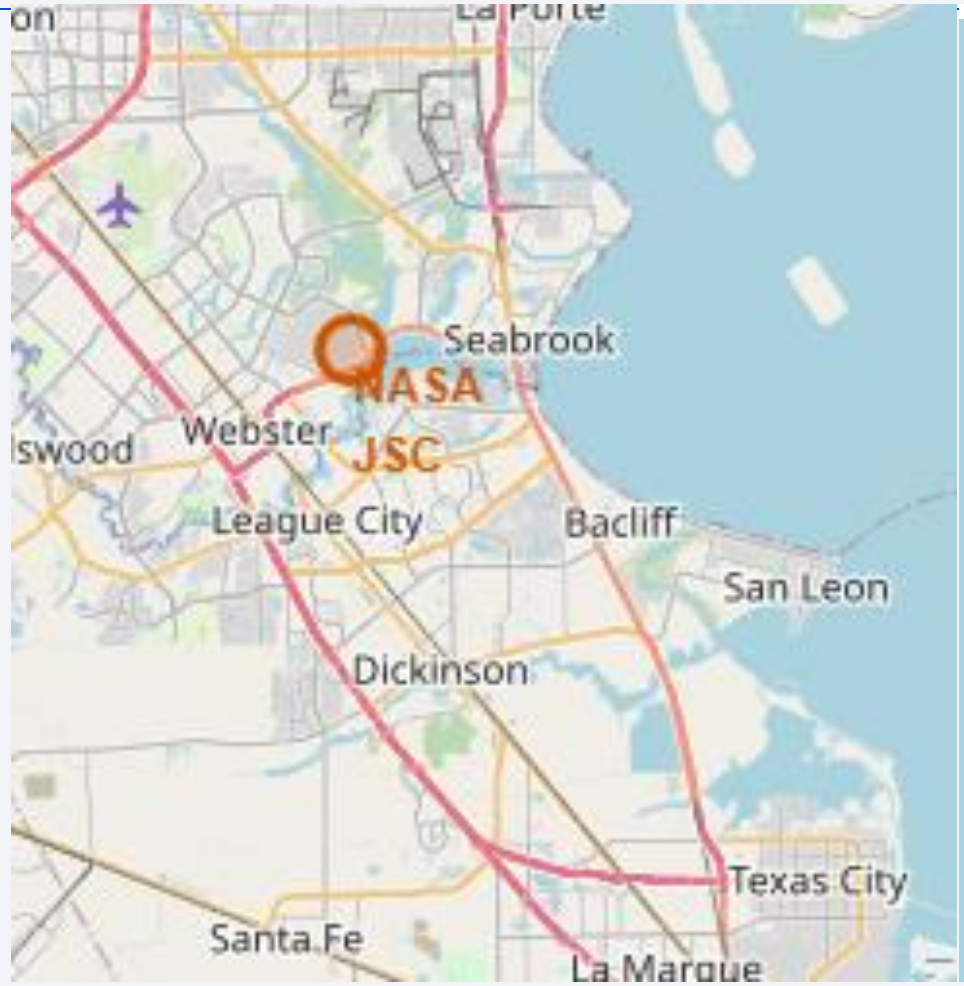
```
%sql select * from SPACEXTBL where "Landing _Outcome" like "Success%" and DATE between '04-06-2010'and'20-03-2017'
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit
08-04-2016	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)
06-05-2016	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO
18-07-2016	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)
14-08-2016	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO
14-01-2017	17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO
19-02-2017	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)
14-08-2017	16:31:00	F9 B4 B1039.1	KSC LC-39A	SpaceX CRS-12	3310	LEO (ISS)
07-09-2017	14:00:00	F9 B4 B1040.1	KSC LC-39A	Boeing X-37B OTV-5	4990	LEO
09-10-2017	12:37:00	F9 B4 B1041.1	VAFB SLC-4E	Iridium NEXT 3	9600	Polar LEO
11-10-2017	22:53:00	F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105	5200	GTO
15-12-2017	15:36:00	F9 FT B1035.2	CCAFS SLC-40	SpaceX CRS-13	2205	LEO (ISS)

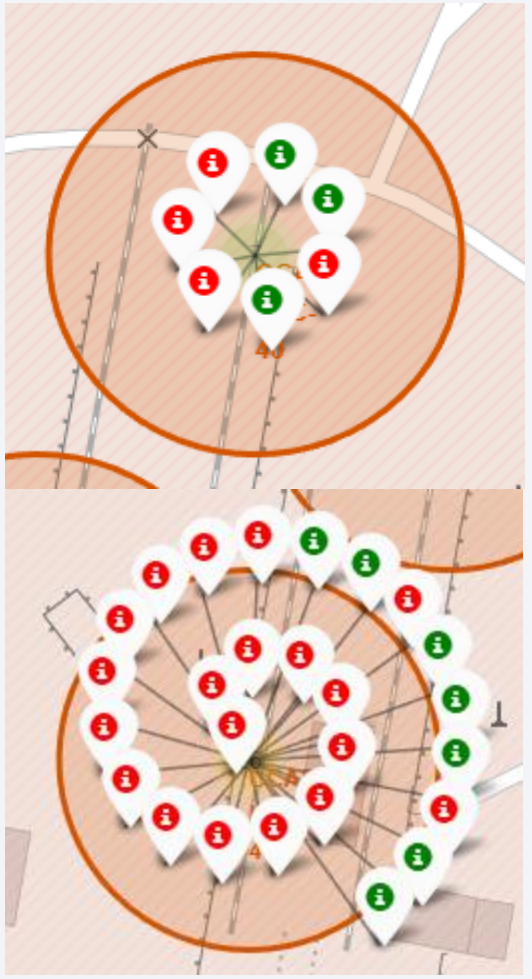

Section 3
Launch Sites
Proximities Analysis

Launch sites markers



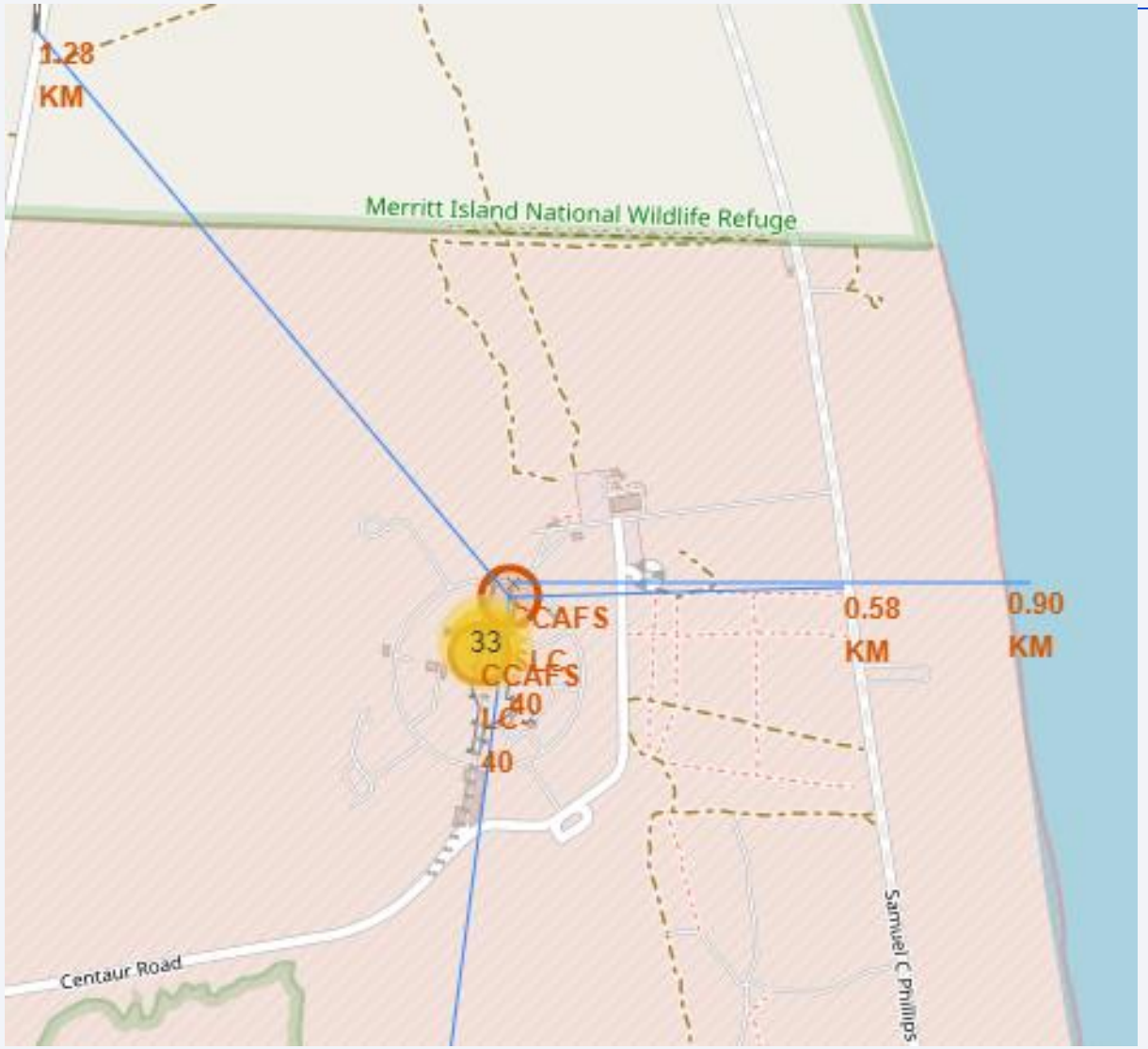
- Launch sites are marked with circles and labels

Launch outcomes

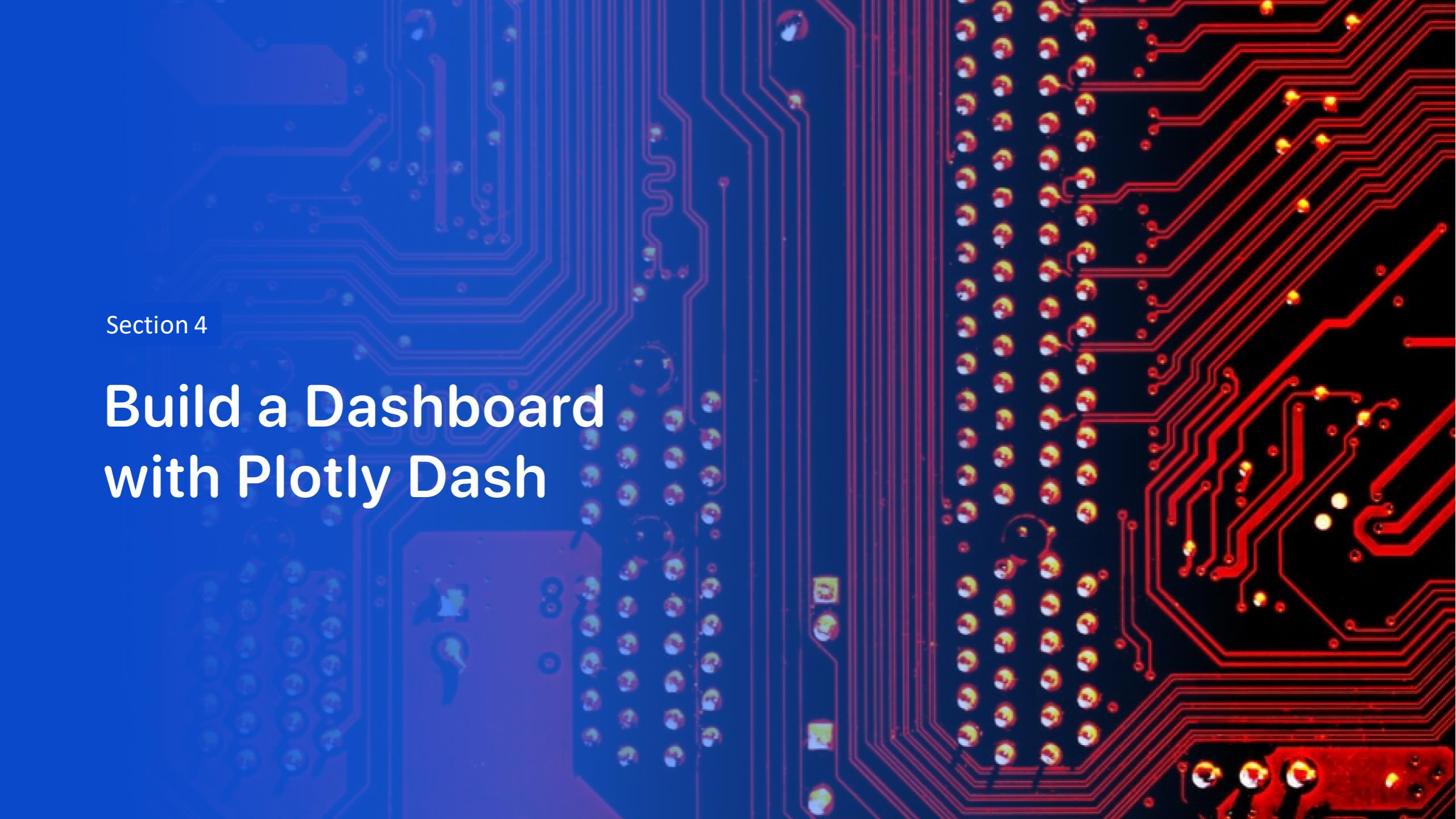


- Launch outcomes for each site are calculated and marked
- Green Markers are successful launches and Red Markers are failure launches

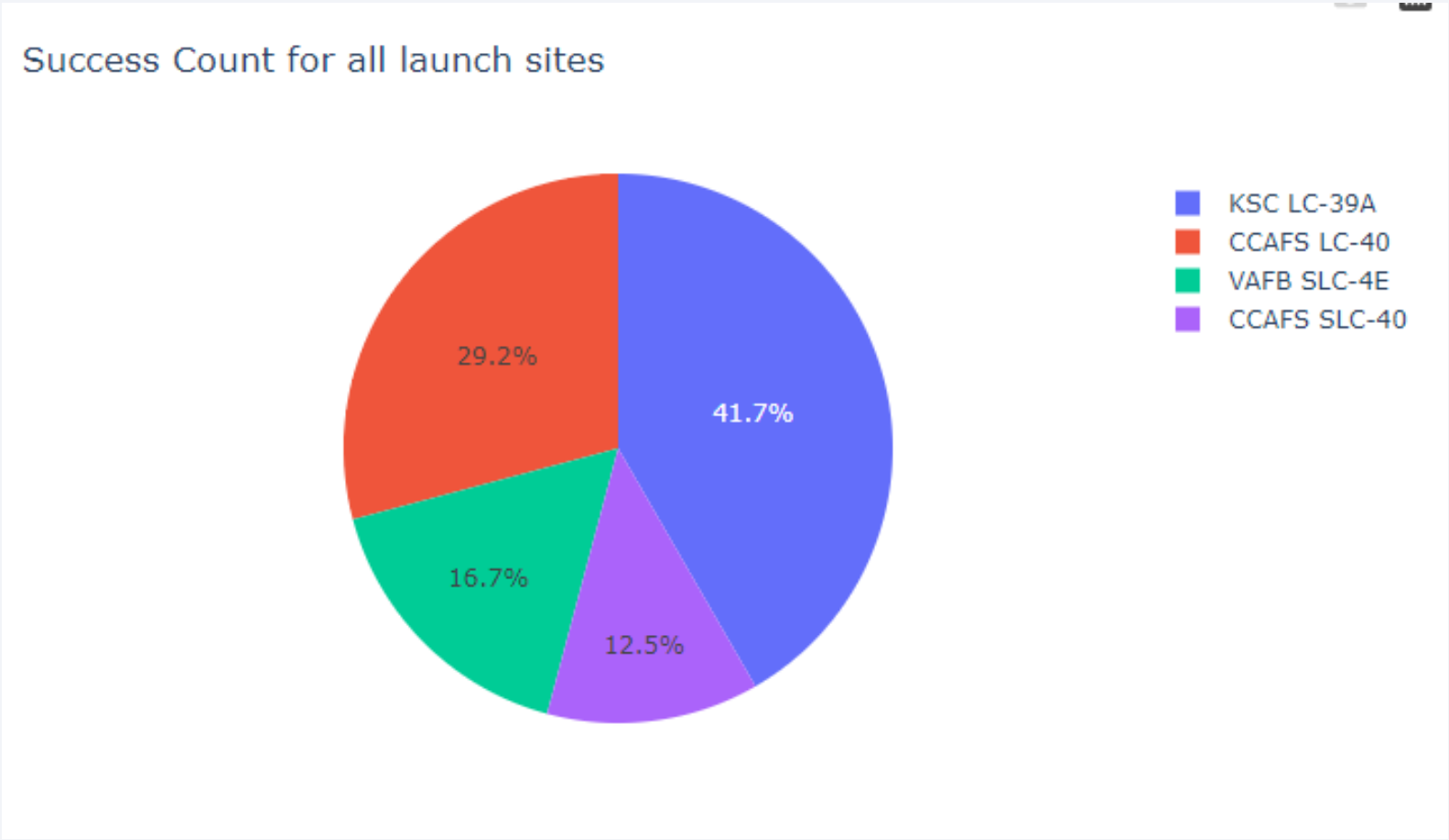
The distances between a launch site to its proximities



- Are launch sites in close proximity to railways? Yes
- Are launch sites in close proximity to highways? Yes
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

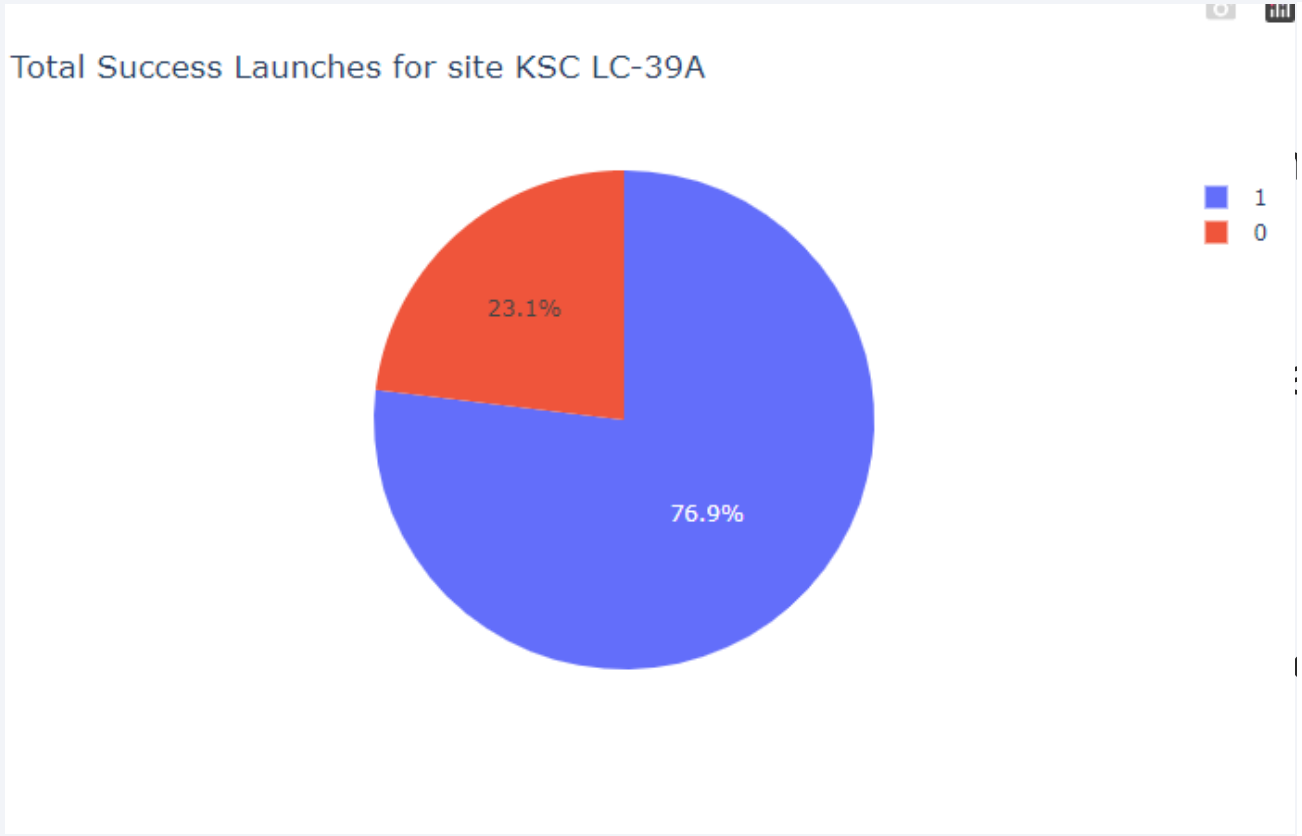

Section 4
Build a Dashboard with Plotly Dash

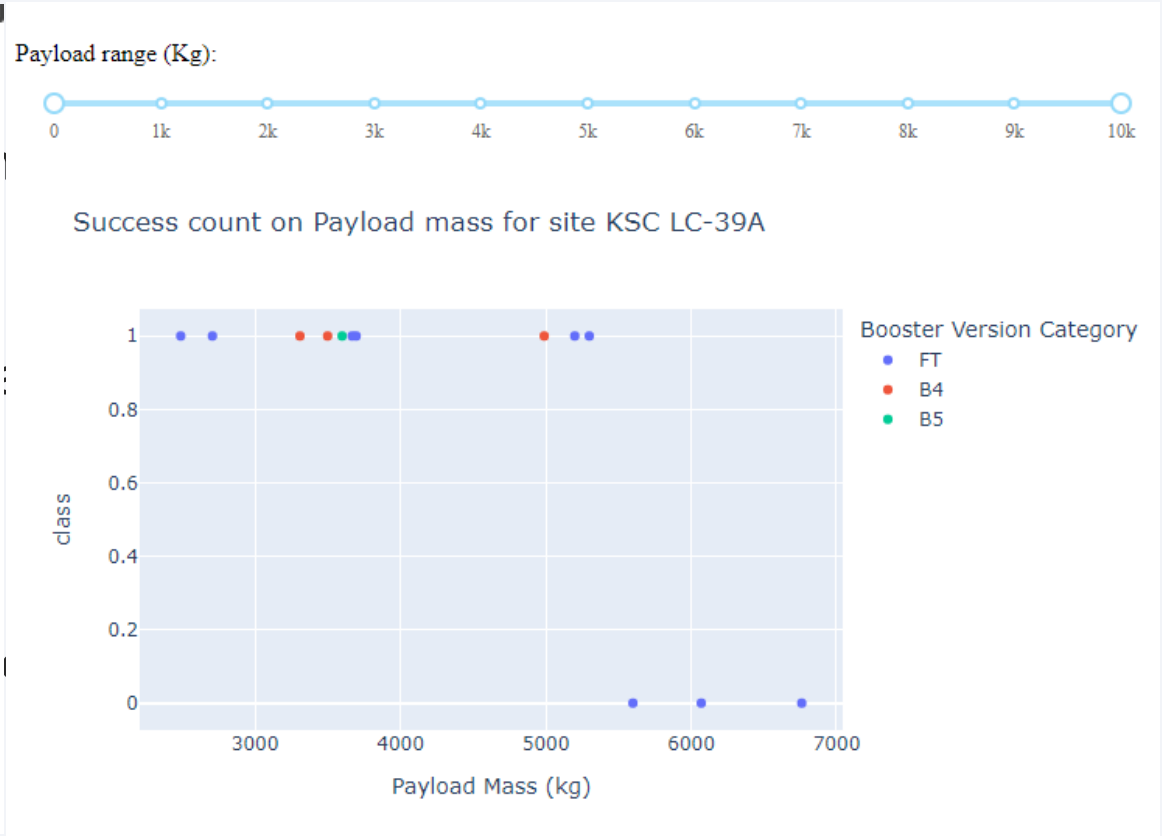
Launch outcome of each launch site



- KSC LC-39A has the most successful launches
- CCAFS LC-40 has the fewest successful launches

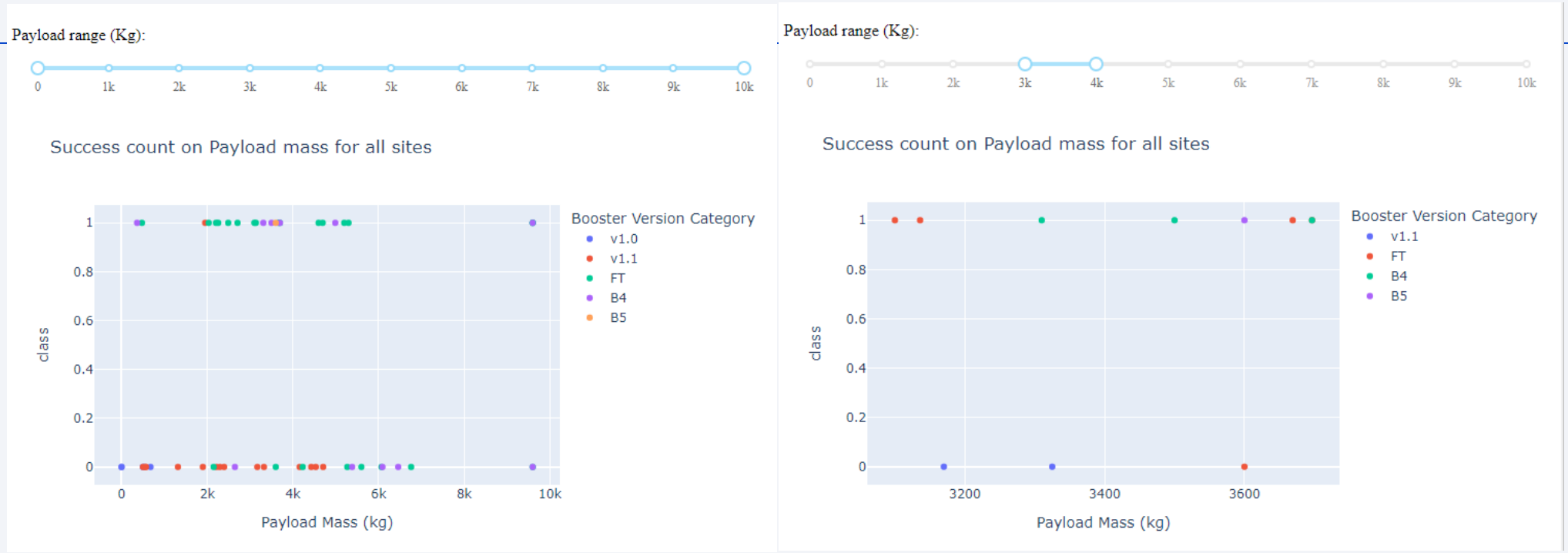
Success launches for KSC LC-39A





- Three launches with payload mass higher than 5500 were failure
- All the launches with payload less than 5500 were all successful

Scatter plot of payload vs launch outcome with range slider



- KSC LC-39A has the largest successful launches and highest launch success rate
- Payload between 3000-4000 has the highest launch success rate.
- Payload higher than 6000 has the lowest launch success rate.
- Which F9 Booster version FT has the highest launch success rate

Section 5

Predictive Analysis (Classification)

Classification Accuracy

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

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

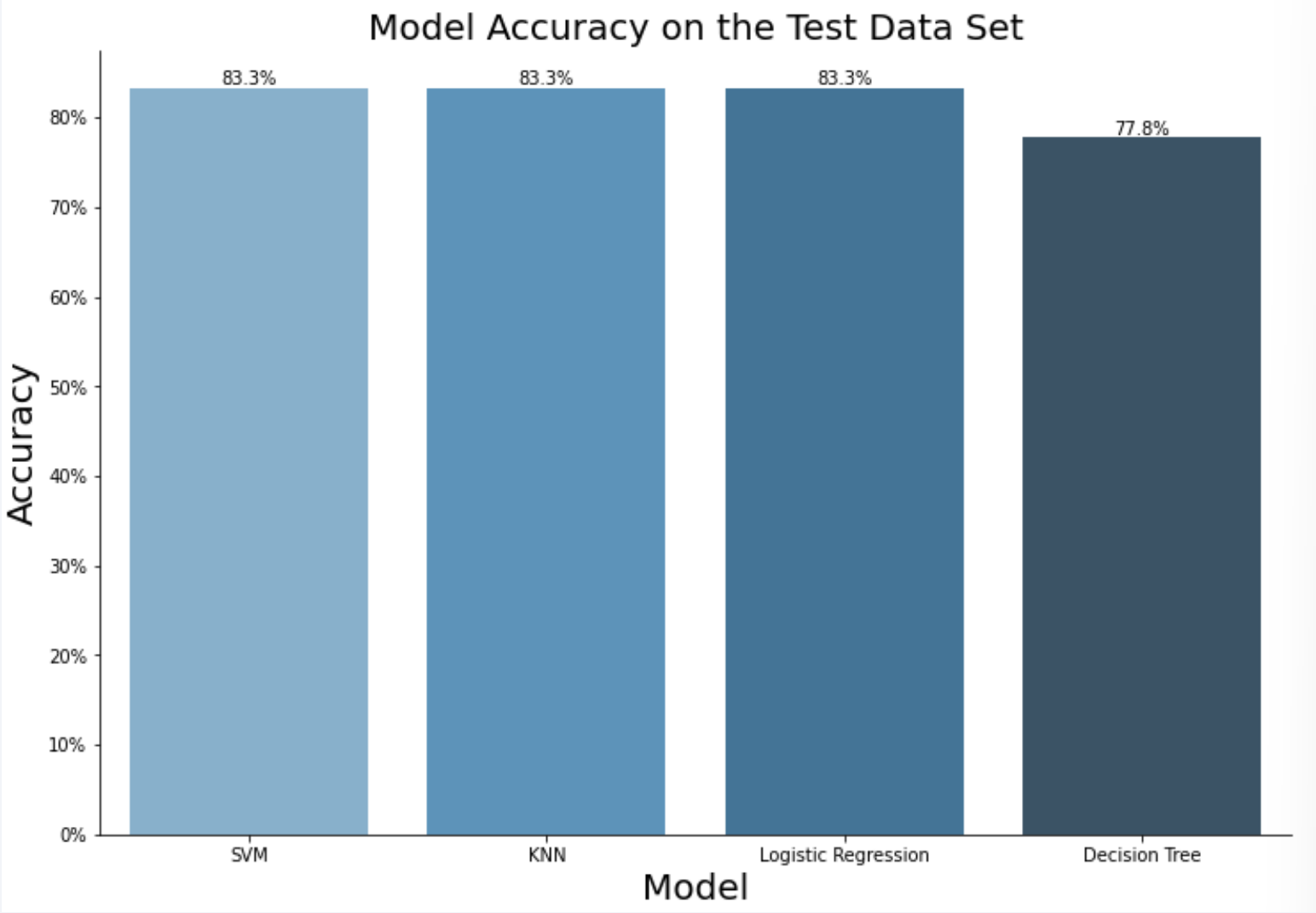
[38] ✓ 0.3s Python

```
Best model is DecisionTree with a score of 0.8875000000000002
Best params is : {'criterion': 'entropy', 'max_depth': 18, 'max_features': 'sqrt', 'min_samples_leaf':
2, 'min_samples_split': 2, 'splitter': 'best'}
```

- Decision Tree has the highest classification accuracy of 0.8875

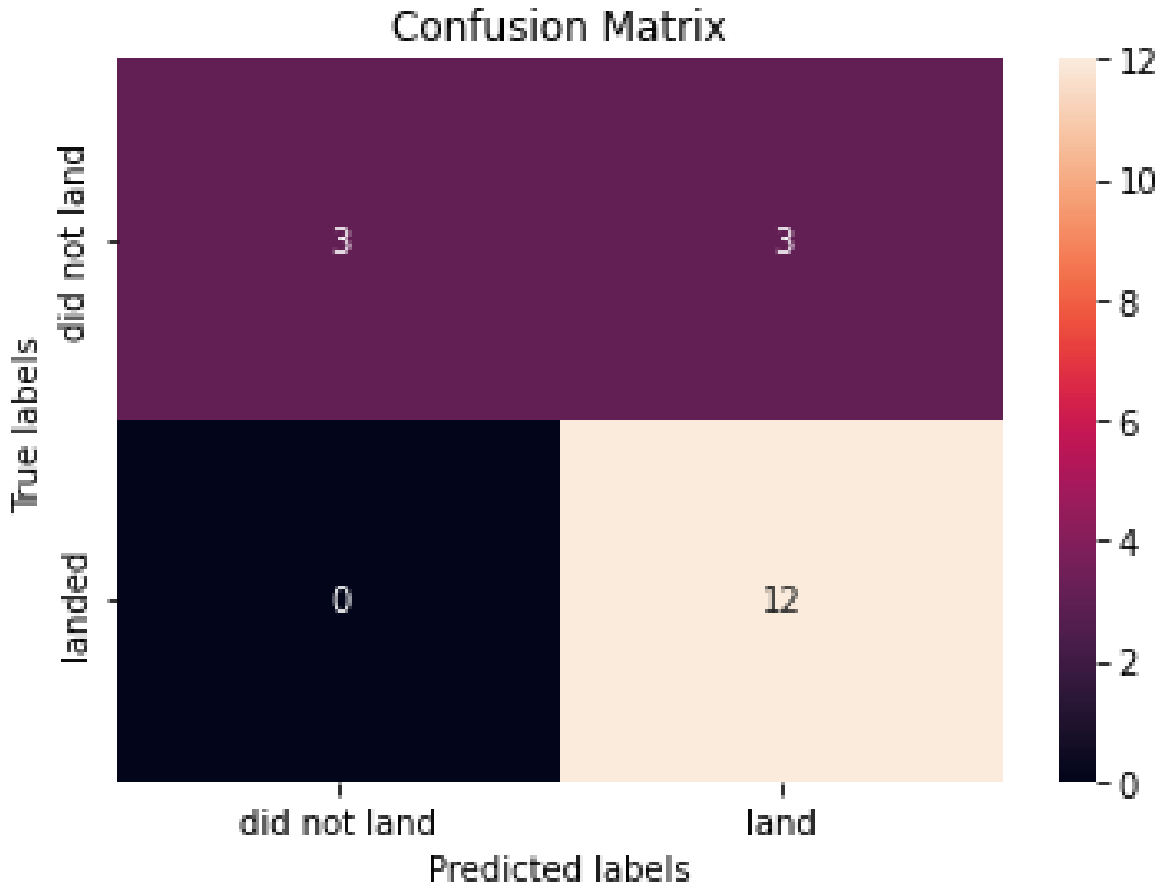
Classification Accuracy

- For the test data set, decision tree has the lowest accuracy



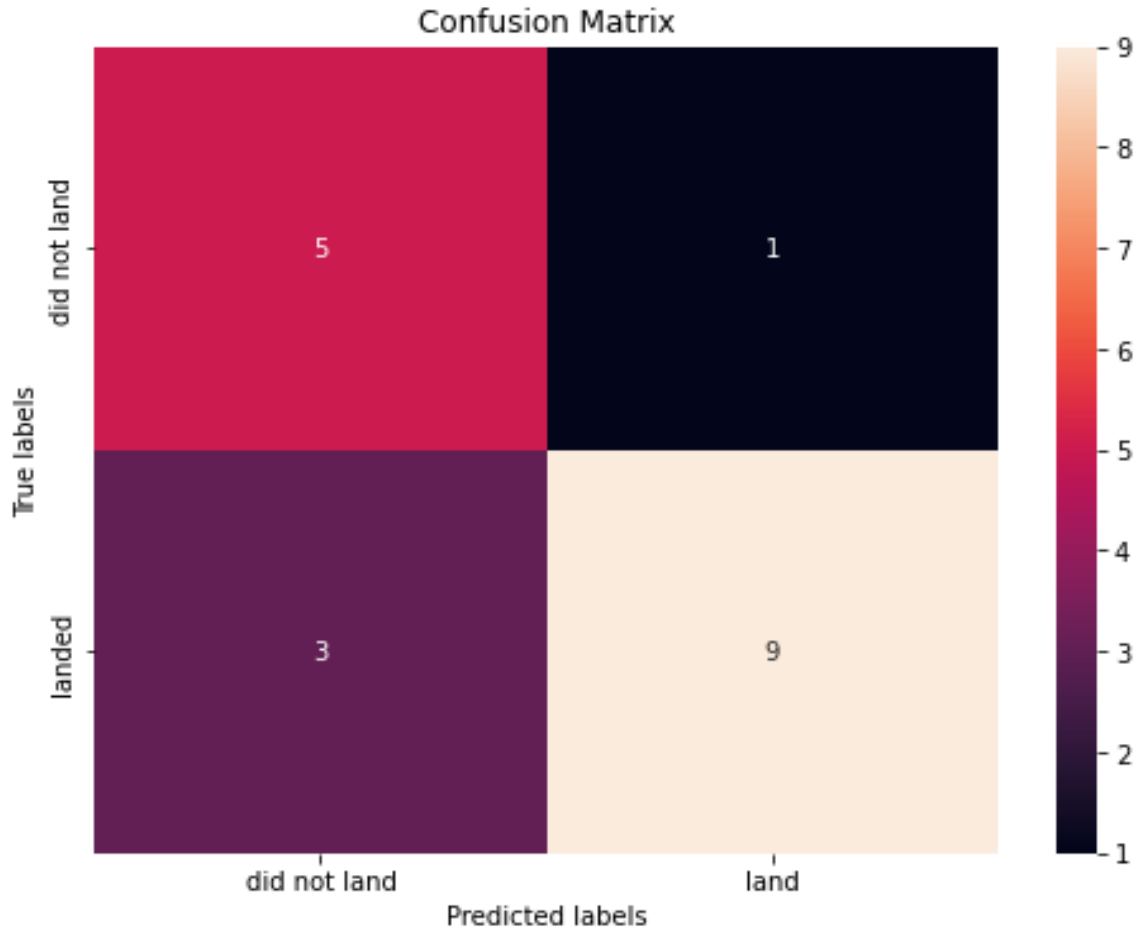
Confusion Matrix

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




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

✓ 0.1s



Conclusions

- For all the launch sites, KSC LC-39A has the largest successful launches and highest success rate.
- The more massive the payload, the less likely the first stage will return, and payload between 3000-4000 has the highest launch success rate.
- F9 Booster version FT has the highest launch success rate
- The sucess rate since 2013 kept increasing till 2020.
- Four models were built to predict if the first stage will land successfully and the Decision Tree has the highest classification accuracy of 0.8875. However, for the test data set, decision tree has the lowest accuracy of 0.7778 while the other three classifier have the accuracy of 0.8334

Thank you!