Winning Space Race with Data Science

ROSHAN PAUDEL

16/11/2015

Outline

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

Executive Summary

- Data collected via SpaceX API and Web Scraping.
- Data wrangling performed to clean and preprocess launch data.
- EDA conducted using visualizations and SQL.
- Interactive Folium map and Plotly Dash dashboard developed.
- Classification models built to predict Falcon 9 landing success.

Introduction

- Project background and context

Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Problems you want to find answers

  - ✓ What factors determine if the rocket will land successfully?
  - ✓ The interaction amongst various features that determine the success rate of a successful landing.
  - ✓ What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

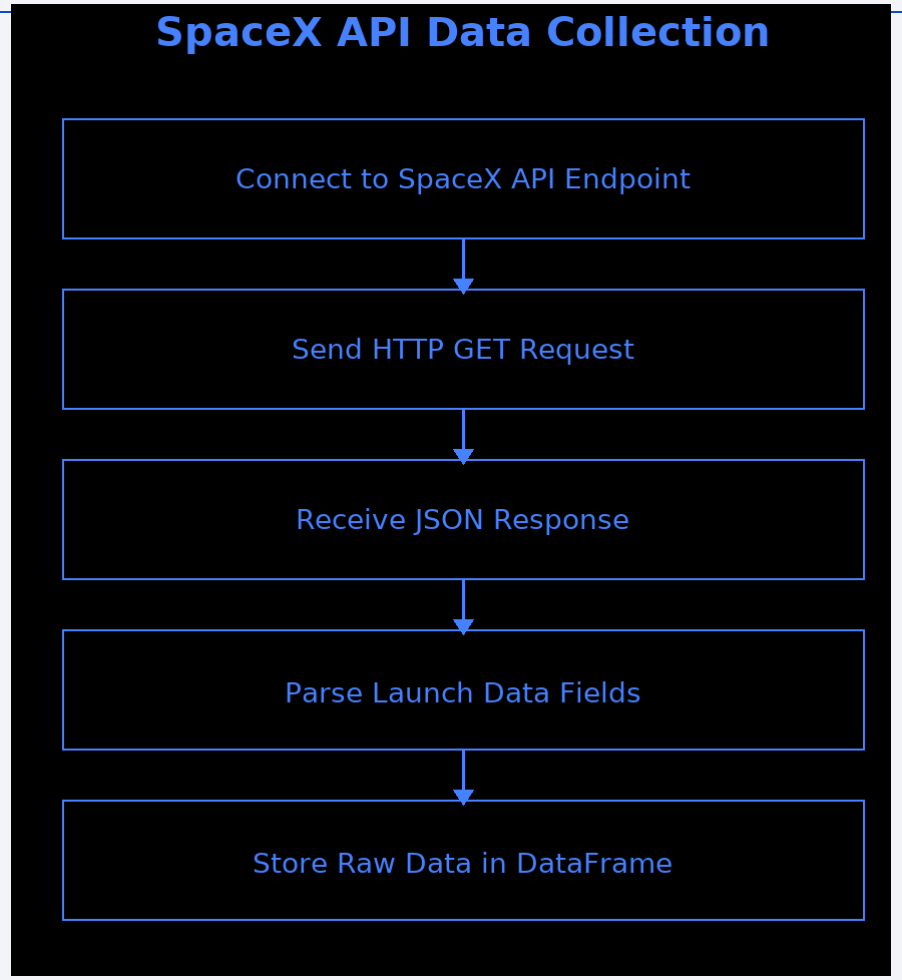
- Data collection methodology:
  - Perform data wranglingSpaceX REST API (Launch, booster, and landing data)
  - Web scraping from Wikipedia for payload and launch site details
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- The data was collected using various methods
  - ✓ Data collection was done using get request to the SpaceX API.
  - ✓ Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
  - ✓ We then cleaned the data, checked for missing values and fill in missing values where necessary.
  - ✓ In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
  - ✓ The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

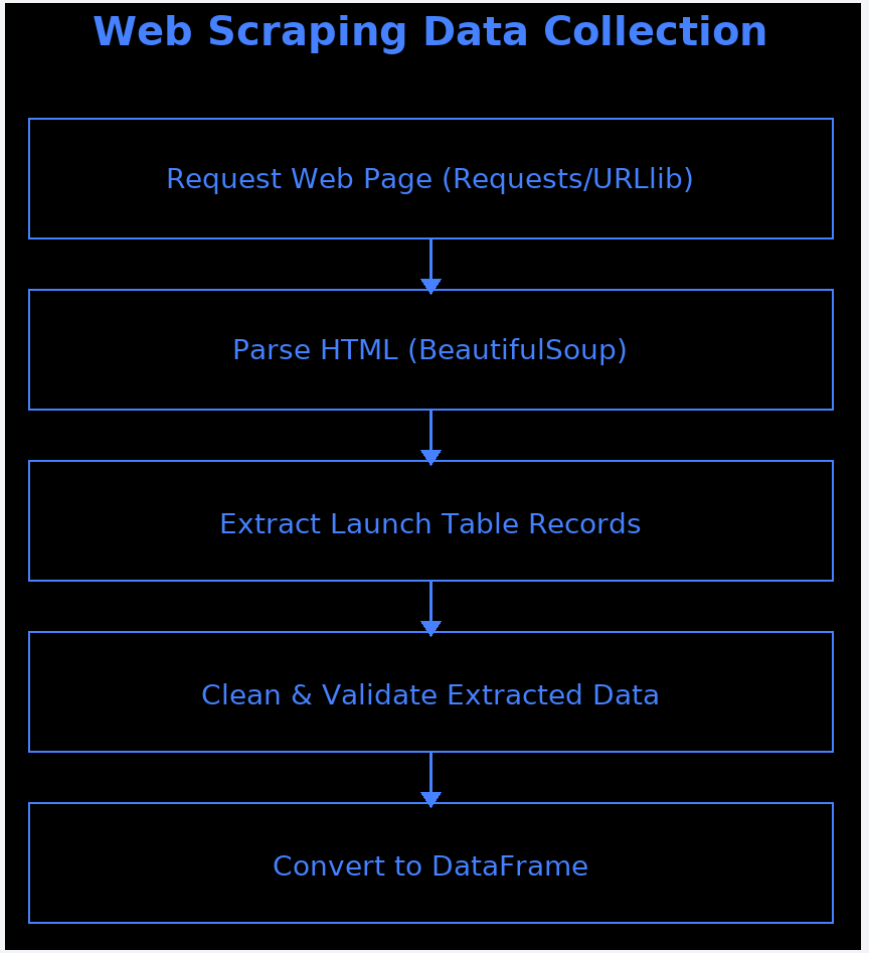
Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.
- The link to the notebook is Data Collection python file github link.



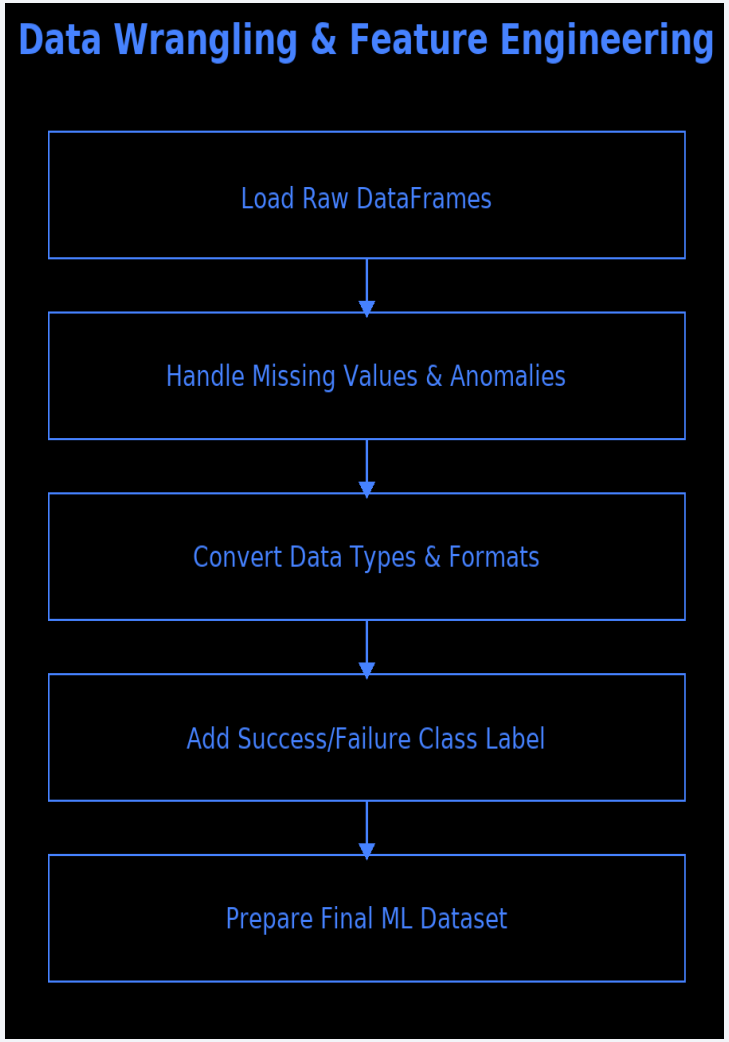
Data Collection - Scraping

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- The link to the notebook is Data Scraping



Data Wrangling

- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits.
- We created landing outcome label from outcome column and exported the results to csv.
- The link to the notebook is Data Weangling

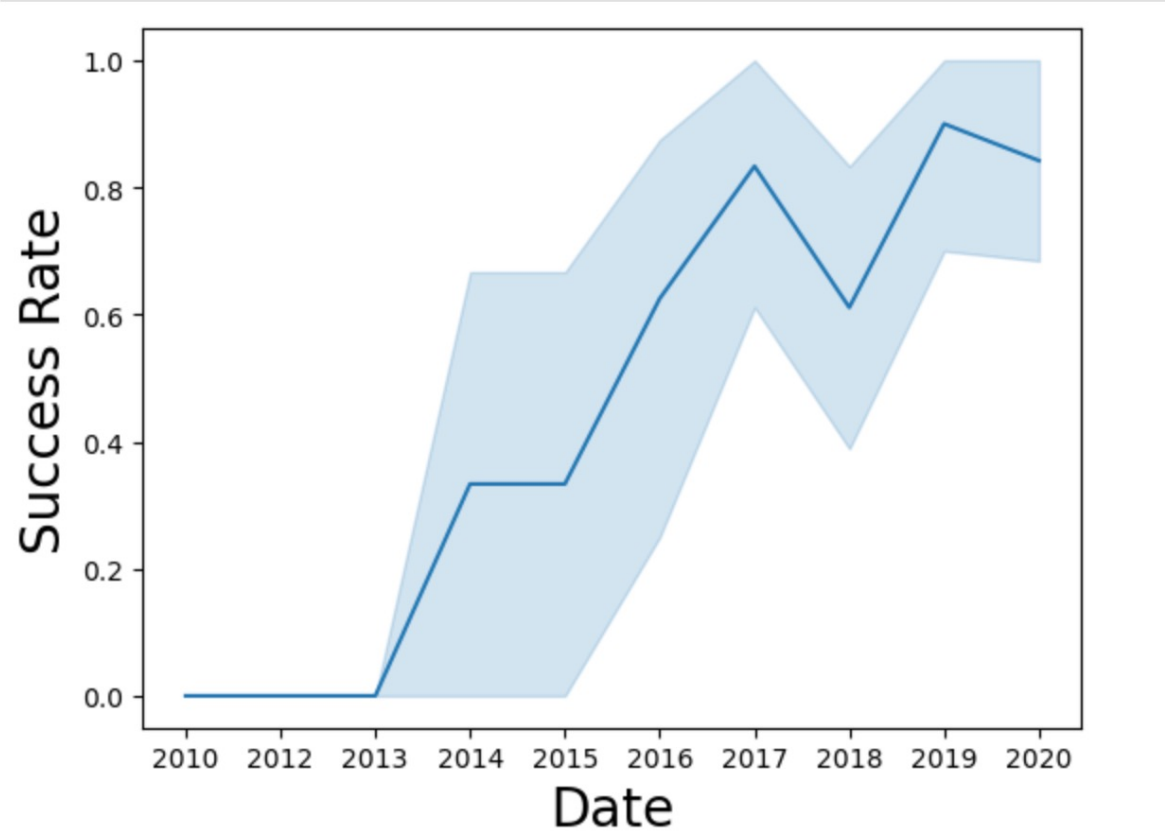


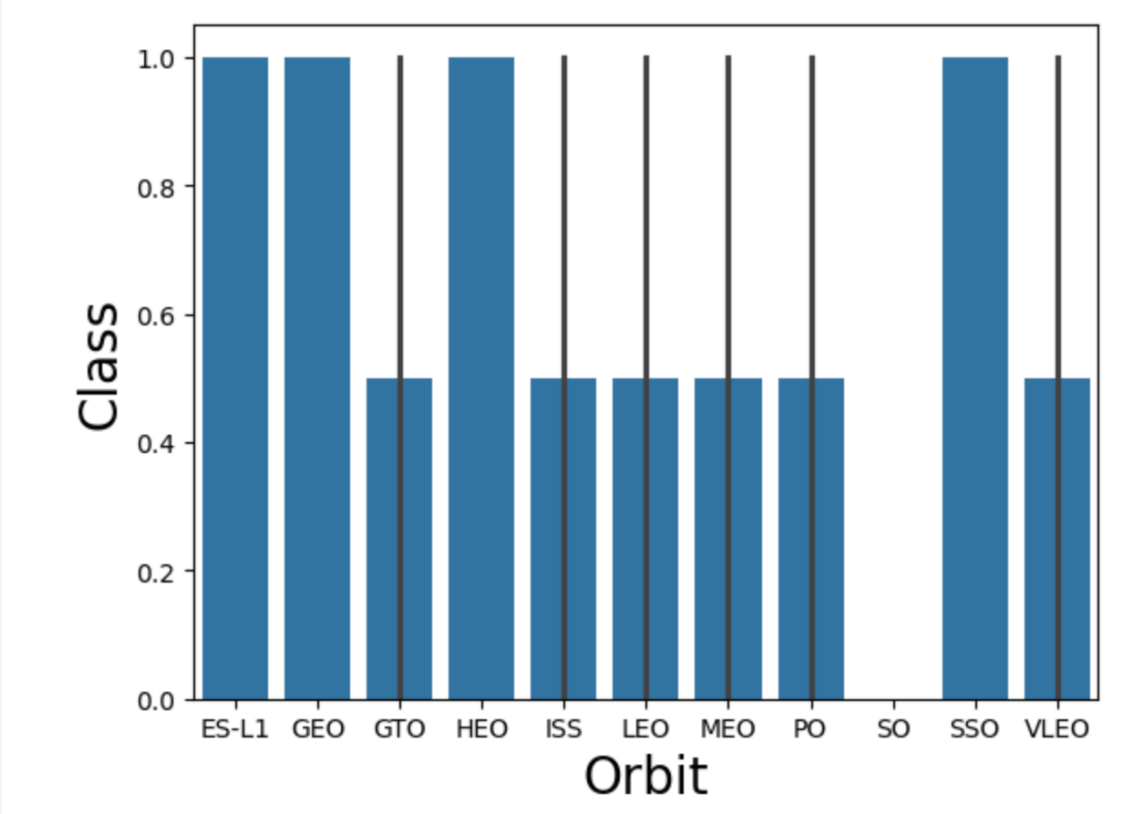
EDA with Data Visualization

- Data Visualization is most necessary as a part of Exploratory Data Analysis. This helps in analyzing the data in an easier way using graphs and charts.
- Scatter plots were created between many attributes to obtain their relationship.
- Bar Charts, Line Charts and Cat plots were also created to analyze the data.

- The link to the notebook is EDA with Visualization

EDA with Data Visualization





EDA with SQL

- We will be adding the data to a database where we can query the reuired results using SQL.
- We have displayed the names of unique launch sites, total payload mass carried by boosters launched by NASA (CRS) and many more,
- We can SQL and extract the data for analysis using Pattern Matching, Buil-in Mathematical Functions and many more
- The link to the notebook is EDA with SQL

EDA with SQL EXAMPLES

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

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

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

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

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

Build an Interactive Map with Folium

- We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distances between a launch site to its proximities.
- The link to the notebook is Visual analytics of folium
- We answered some question for instance:
  - ✓ Are launch sites near railways, highways and coastlines.
  - ✓ Do launch sites keep certain distance away from cities.

Build a Dashboard with Plotly Dash

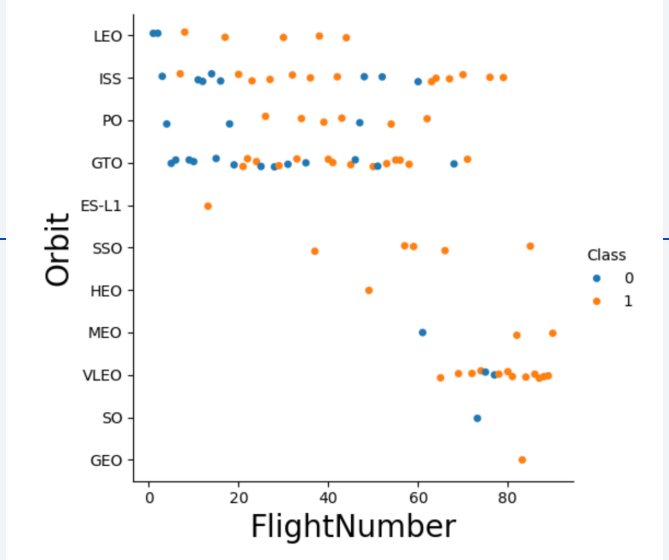
- We have created a real-time visual dashboard using Plotly which provides better view of Graphs and Charts.
- The plots were added to the dashboard because it provides real-time experience to the users with a great view of data.
- Added a dropdown to the dashboard to select launch sites and find the EDA of that particular launch site.
- The link to the notebook is Plotly Dash

Predictive Analysis (Classification)

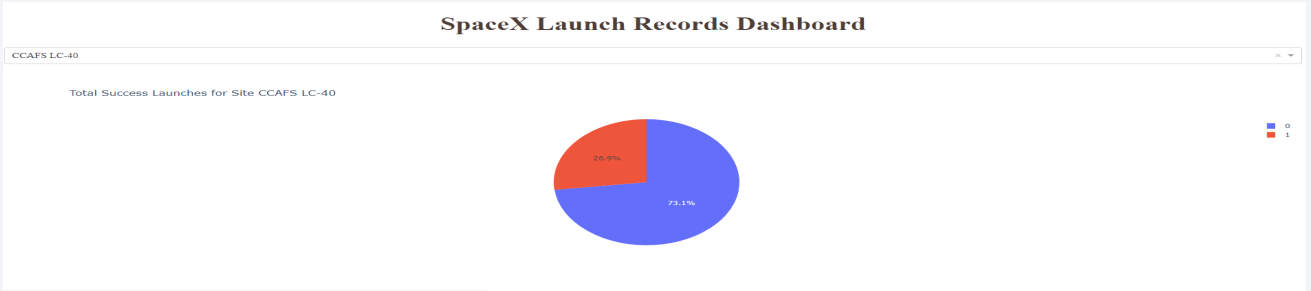
- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- The link to the notebook is Modelling

Results

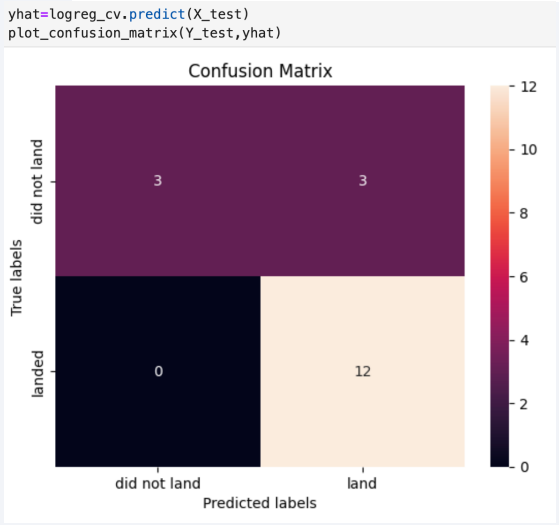
- Exploratory data analysis results :

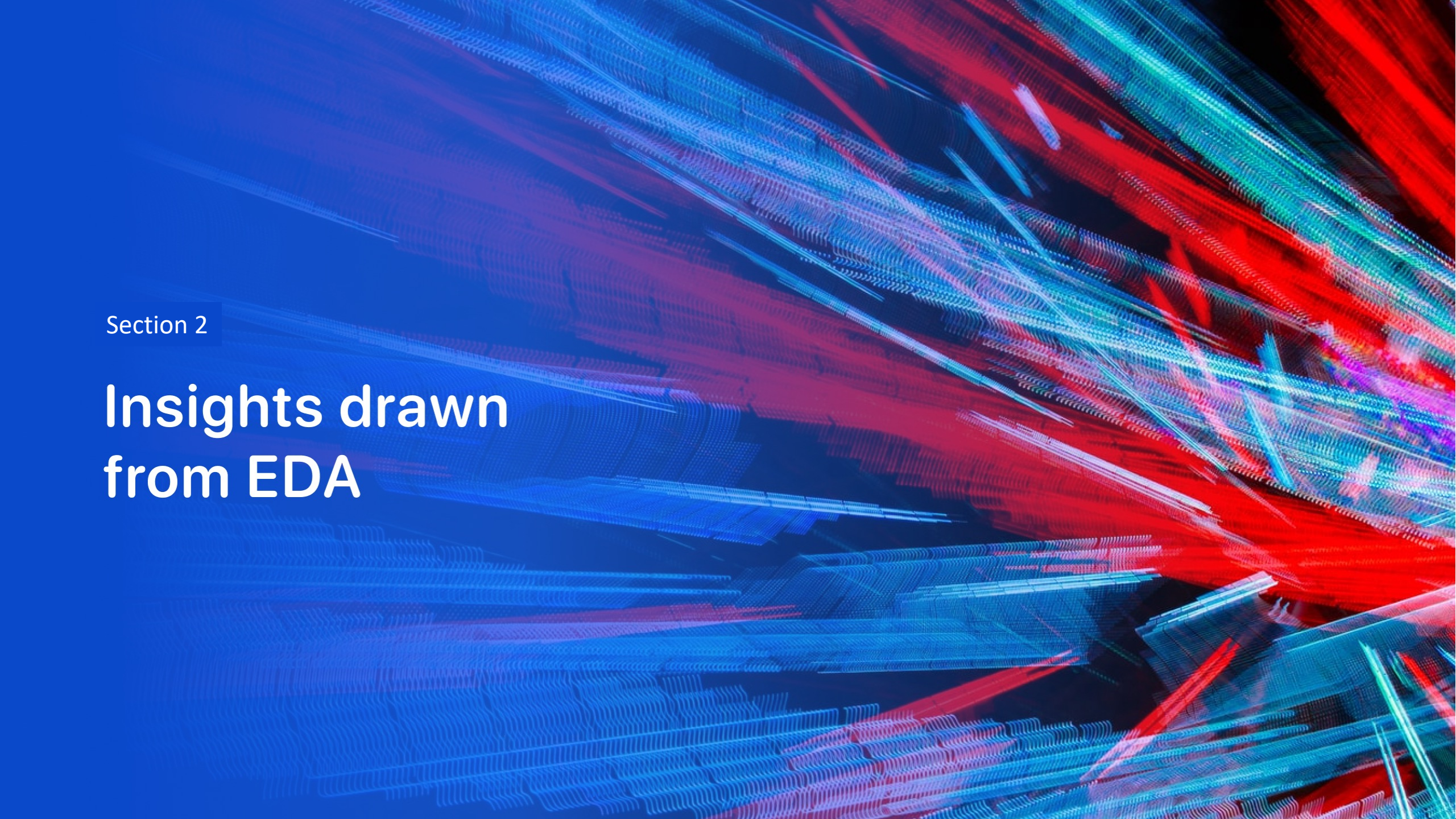


- Interactive analytics samples :



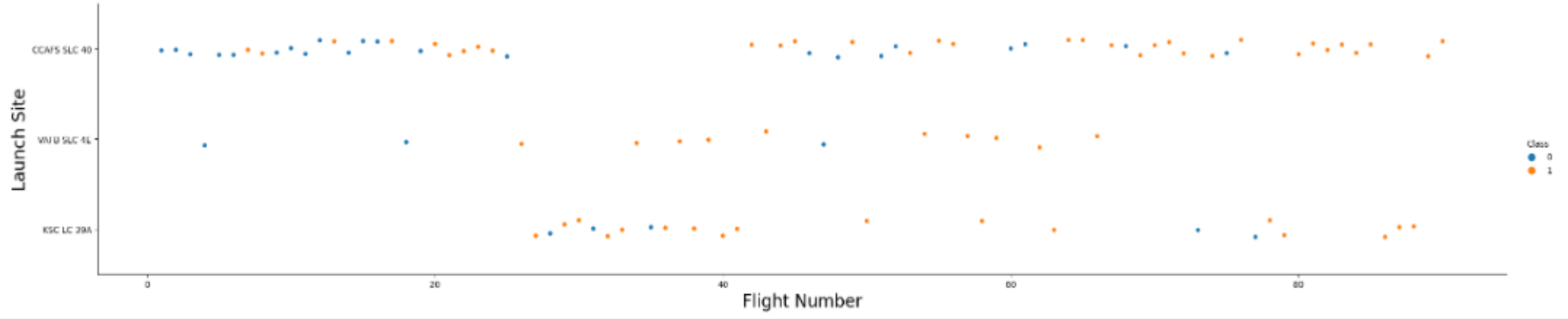
- Predictive analysis results :




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 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()
```

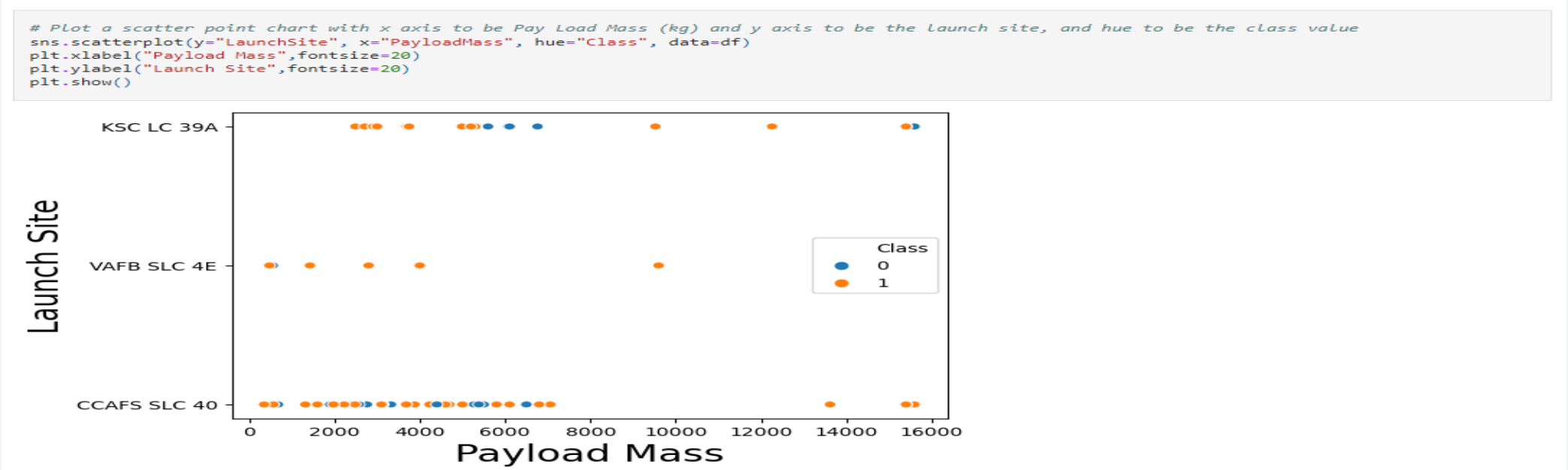


- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.

Payload vs. Launch Site

- We can find that heavier Payload Mass rockets are launched in KSC LC 39A and CCAFS SLC 40 Launch Sites.

```
# 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.scatterplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df)
plt.xlabel("Payload Mass",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```

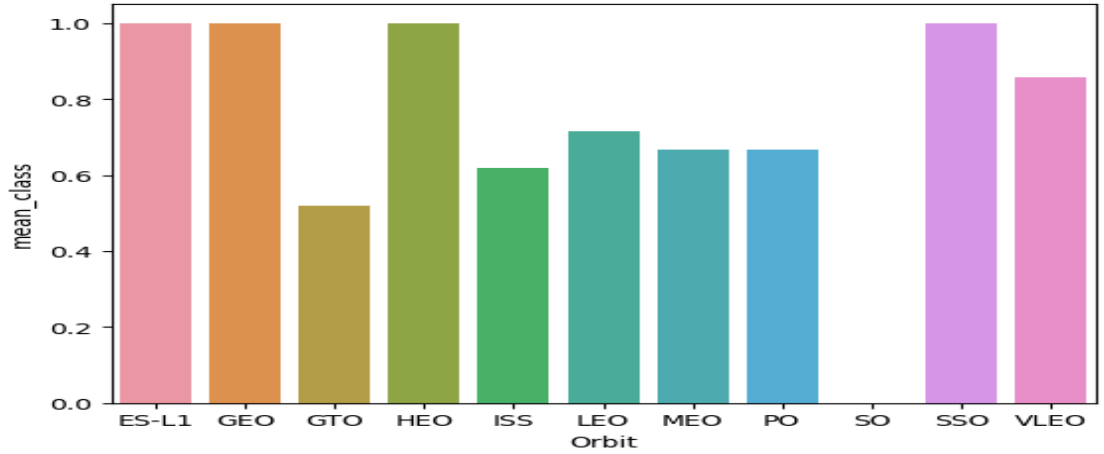


Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate

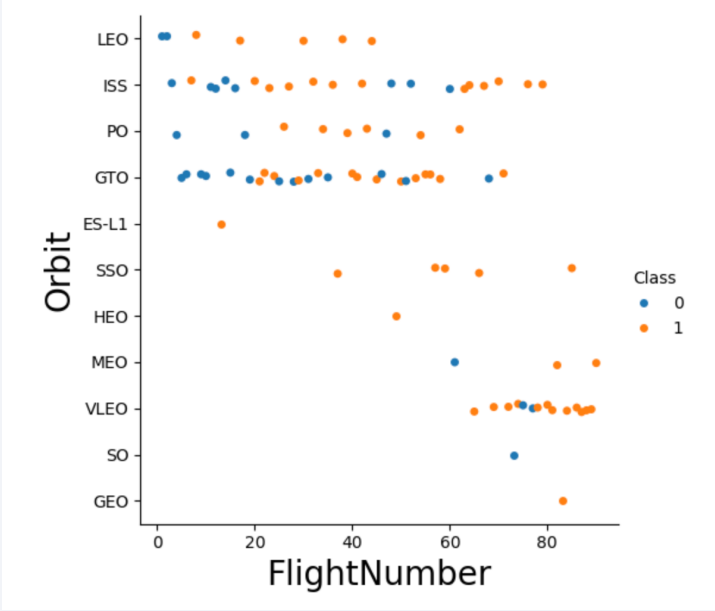
```
# HINT use groupby method on Orbit column and get the mean of Class column
A = df.groupby(['Orbit']).agg(mean_class=("Class", 'mean'))
A = A.reset_index()

# plot barplot
sns.barplot(x="Orbit",y="mean_class",data=A)
```

```
<AxesSubplot:xlabel='Orbit', ylabel='mean_class'>
```



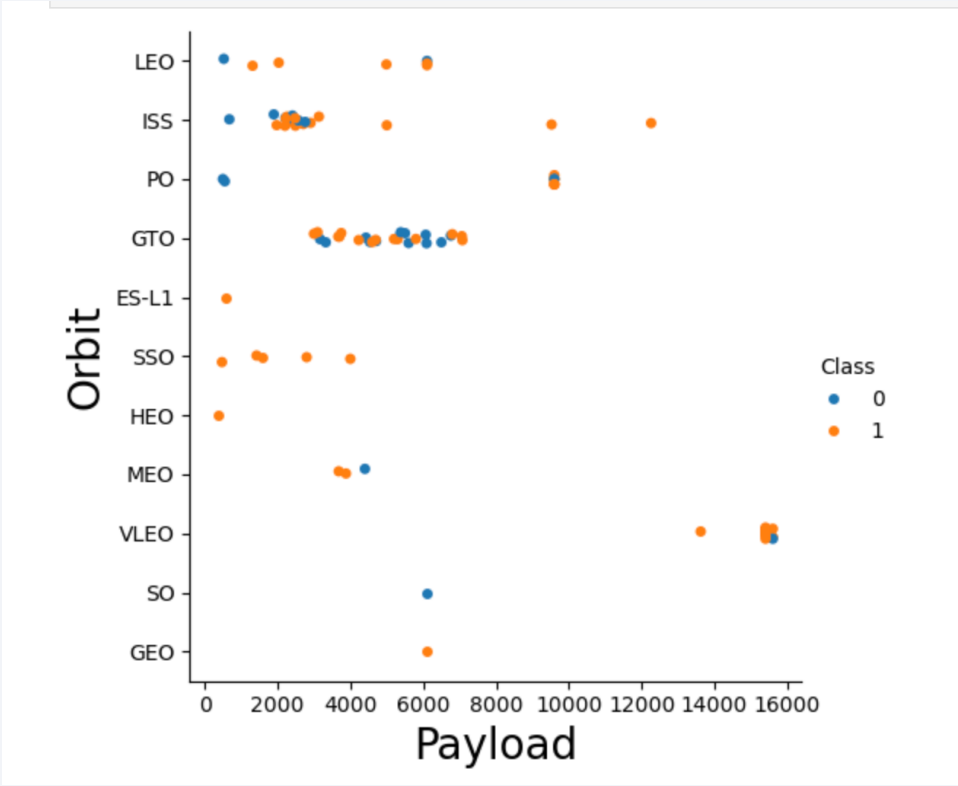
Flight Number vs. Orbit Type

- The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



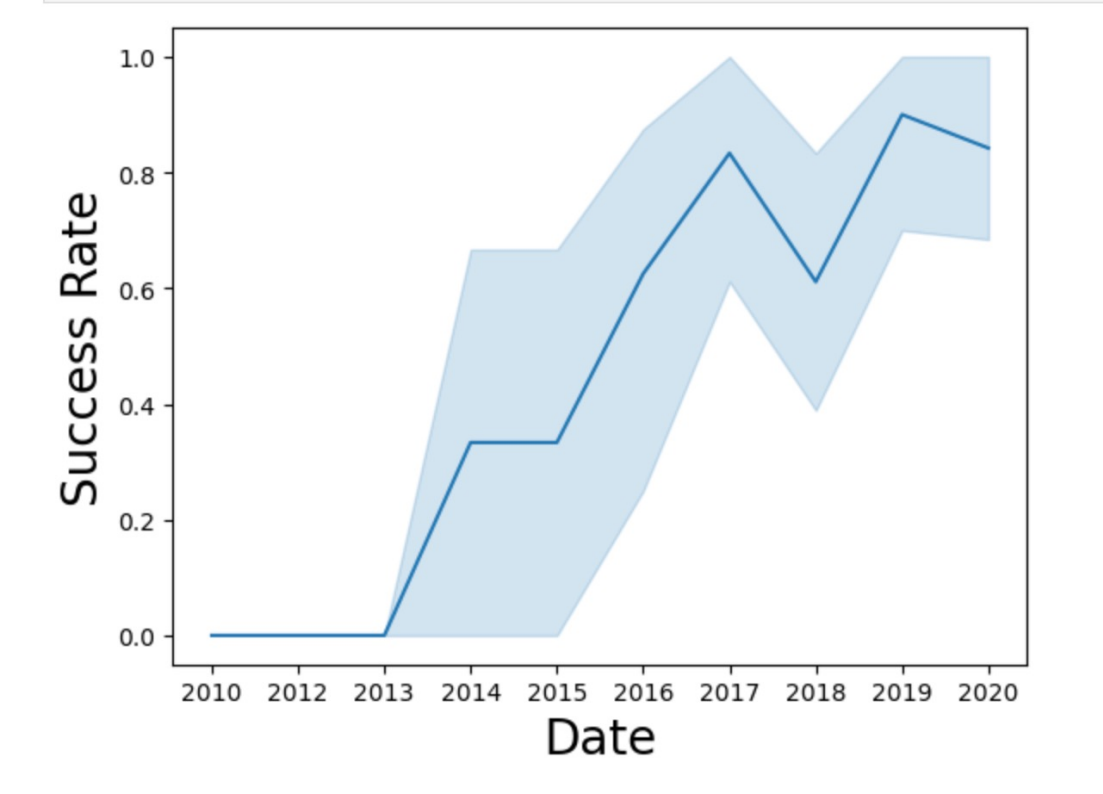
Payload vs. Orbit Type

- We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



Launch Success Yearly Trend

- From the plot, we can observe that success rate since 2013 kept on increasing till 2020.



All Launch Site Names

- We used the key word DISTINCT to show only unique launch sites from the SpaceX data.

```
%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'

- We used the query below to display 5 records where launch sites begin with `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	Lan
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Fai
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	Fai
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	

Total Payload Mass

- We calculated the total payload carried by boosters from NASA as 45596 using the query below

```
%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

- We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

```
%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

First Successful Ground Landing Date

We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015

```
%sql select min(DATE) from SPACEXTBL where Landing_Outcome = 'Success (ground pad)'
```

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

min(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

We used the WHERE clause to filter for boosters which have successfully landed on drone ship and applied the AND condition to determine successful landing with payload mass greater than 4000 but less than 6000

```
%sql select Booster_Version from SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__K
```

```
 * 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

- We used wildcard like ‘%’ to filter for WHERE MissionOutcome was a success or a failure.

```
%sql select DISTINCT mission_outcome, count(*) as count from XMG48161.SPACEX group by mission_outcome;
```

```
 * ibm_db_sa://xmg48161:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32328/bludb
Done.
```

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

Boosters Carried Maximum Payload

- We determined the booster that have carried the maximum payload using a subquery in the WHERE clause and the MAX() function.

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

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

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

- We used a combinations of the WHERE clause, LIKE, AND, and BETWEEN conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

```
%sql SELECT SUBSTR(Date,6,2) AS Month, Booster_Version, Launch_site FROM SPACEXTBL WHERE Landing_Outcome LIKE
```

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

Month	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

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

- We selected Landing outcomes and the COUNT of landing outcomes from the data and used the WHERE clause to filter for landing outcomes BETWEEN 2010-06-04 to 2010-03-20.
- We applied the GROUP BY clause to group the landing outcomes and the ORDER BY clause to order the grouped landing outcome in descending order.

```
%sql select distinct landing__outcome, count(*) as landing_outcome_count from XMG48161.SPACEX where date between '2010-06-04' and '2017-03-20' group b
 * ibm_db_sa://xmg48161:***@2d46b6b4-cbf6-40eb-bbce-6251e6ba0300.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32328/bludb
Done.
```

landing__outcome	landing_outcome_count
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

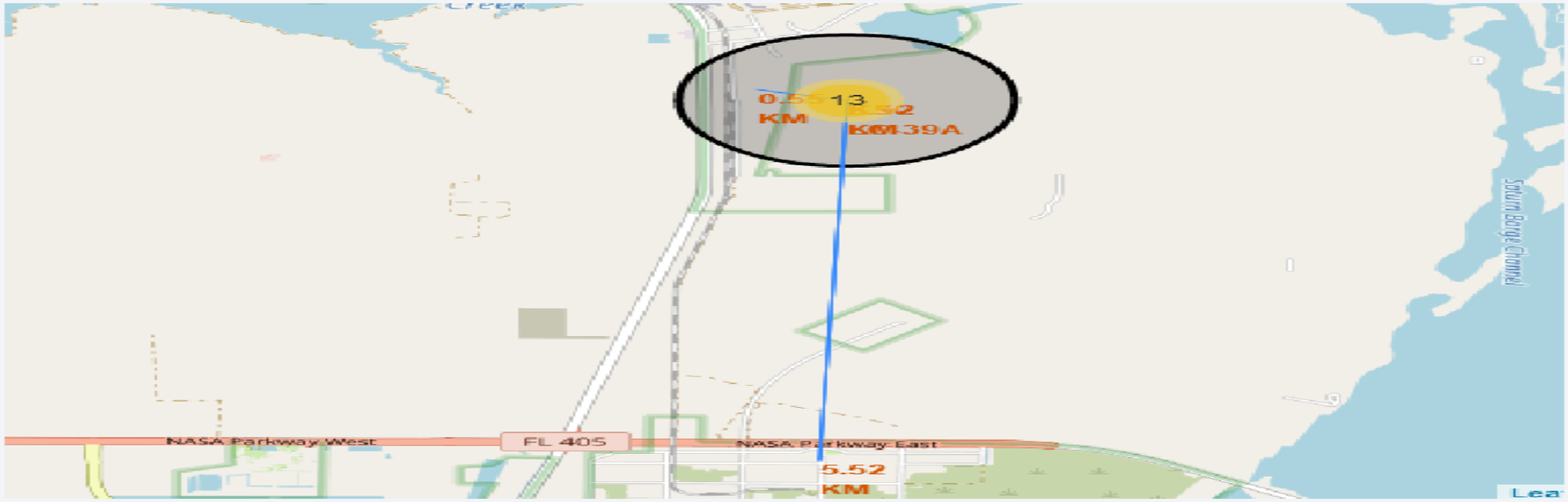
All Launch Sites

- All the launch sites are near to the sea.
- The launch should be taken with utmost safety as there is some areas of land nearby.

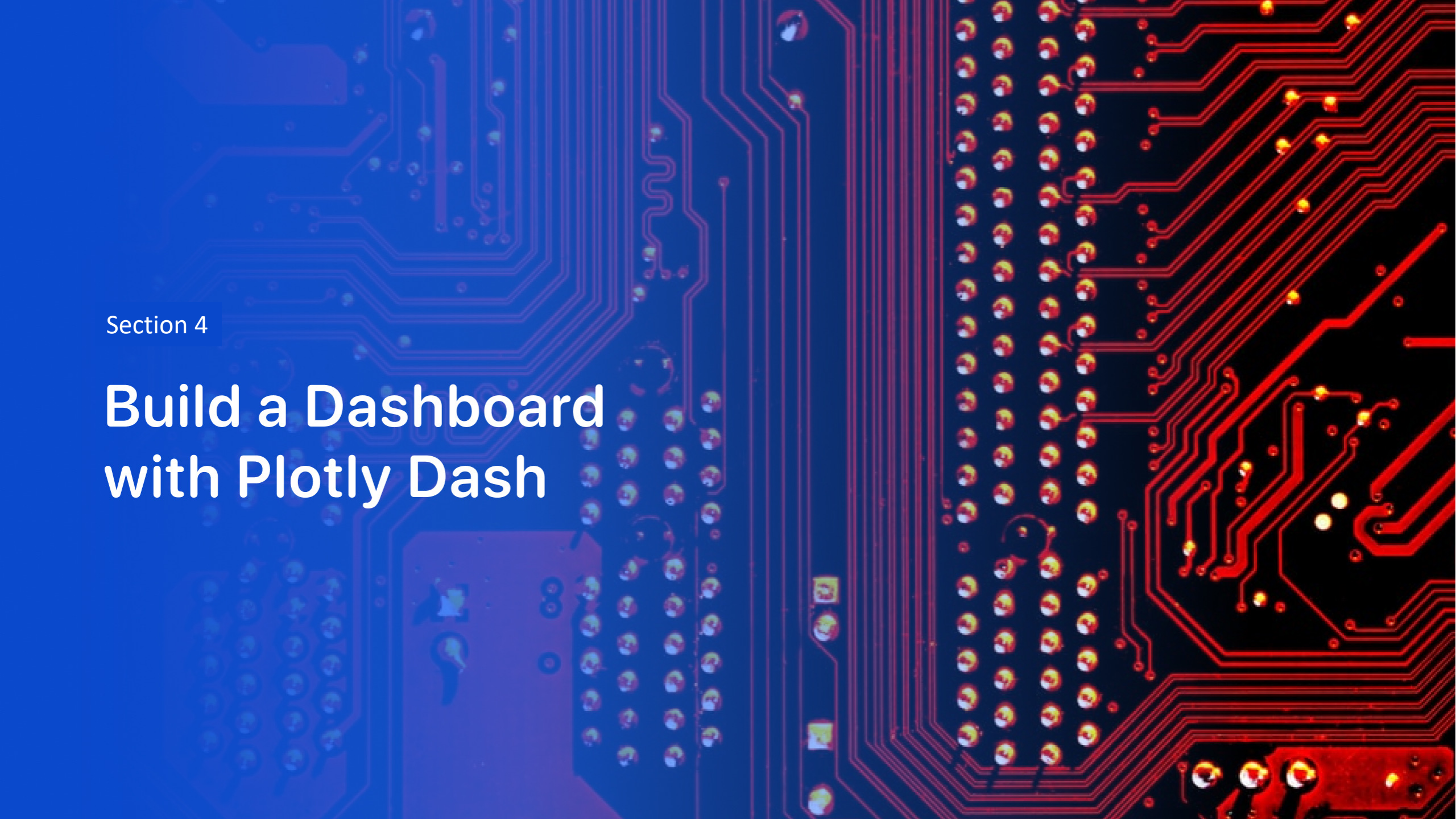
Launch Outcomes based on Sites

- The green markers in the map gives out successful launch outcomes and red marker gives out failed launch outcomes

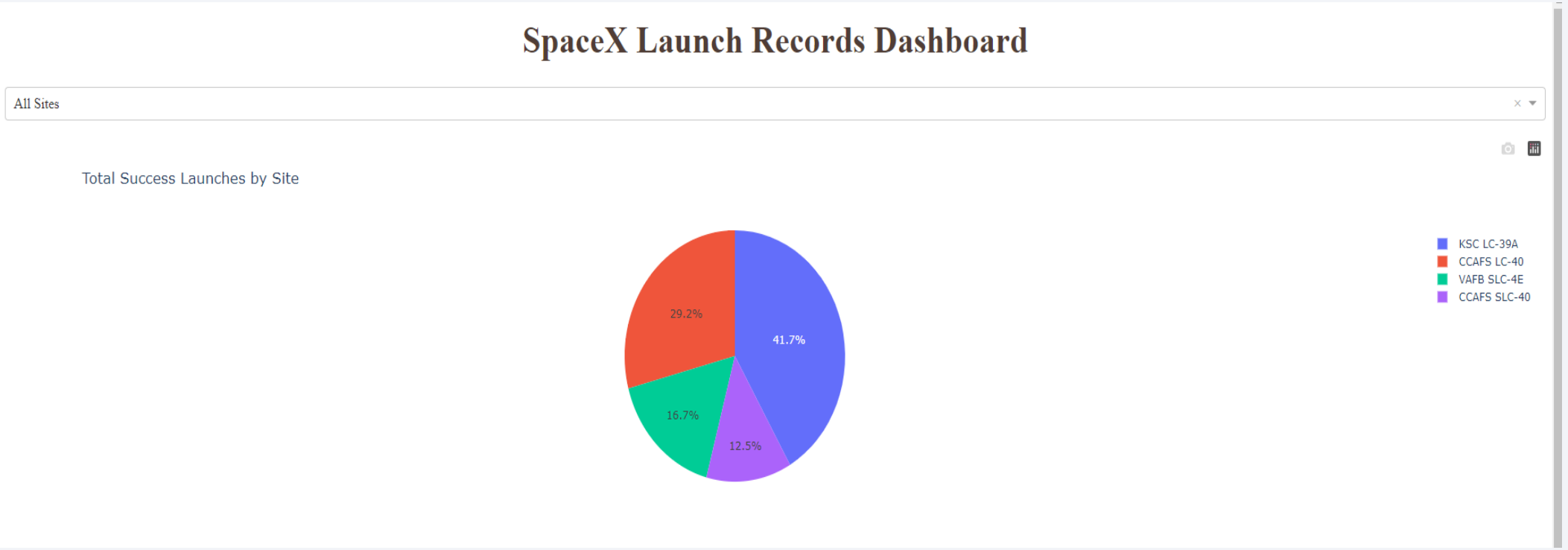
Launch Site distance to landmarks



- A line is drawn between the launch site and its closest Highway.

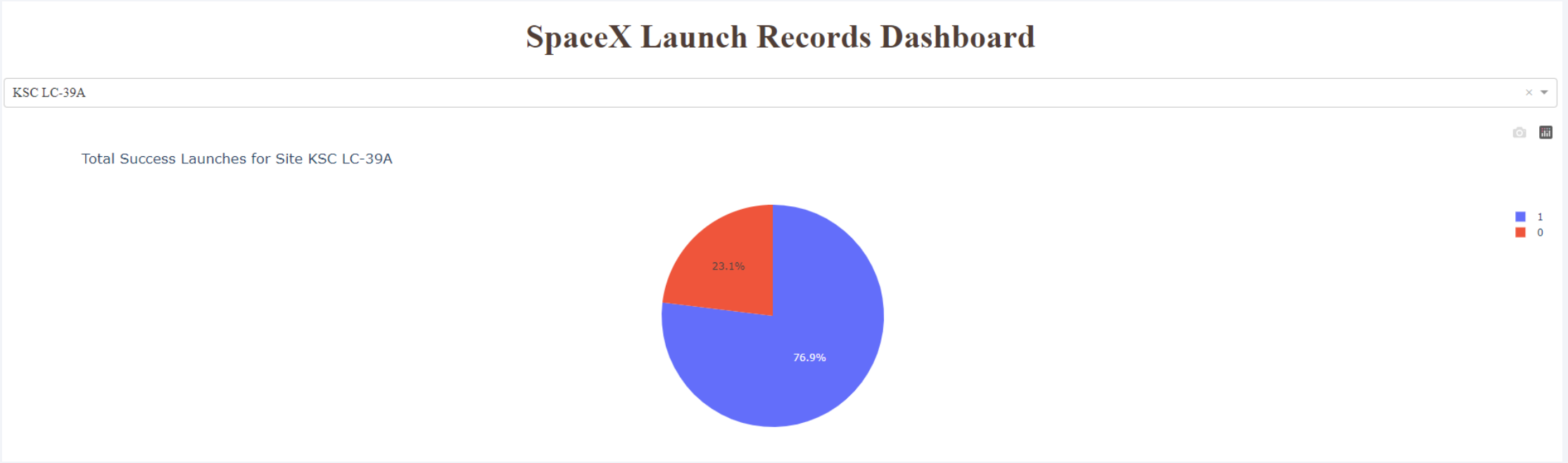

Section 4
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



- Pie chart with the success rate of rocket launched based on sites.

Pie chart showing the Launch site with the highest launch success ratio



- The pie chart which has the highest successful launches based on Launch Site

Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider

This Scatter Plot shows the success rate based on Payload Mass of rockets between 2000 and 6000 KG

Section 5

Predictive Analysis (Classification)

Classification Accuracy

```
print('Accuracy Score : ')
print('Logistic Regression Accuracy Score :',logreg_cv.best_score_)
print('Support Vector Machine Accuracy Score :',svm_cv.best_score_)
print('Decision Tree Accuracy Score :',tree_cv.best_score_)
print('KNN Accuracy Score :',knn_cv.best_score_)
```

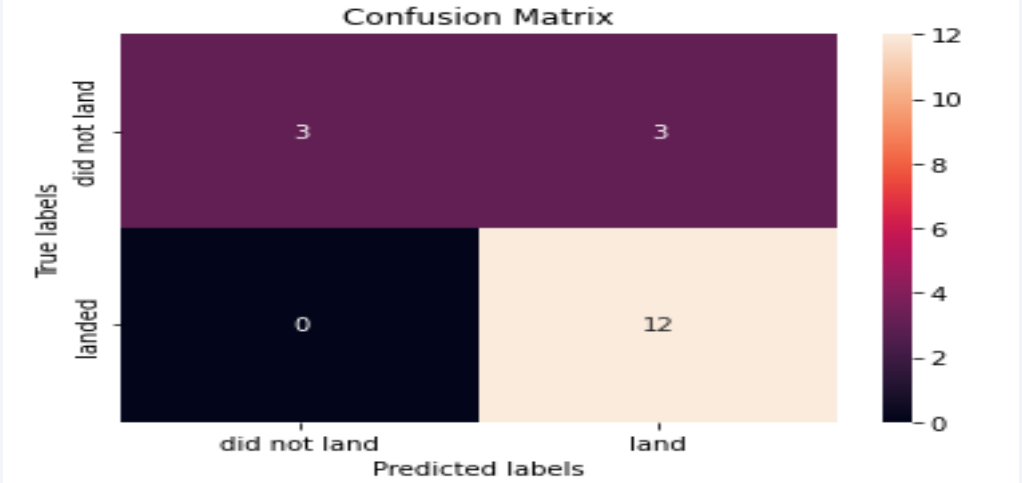
```
Accuracy Score :
Logistic Regression Accuracy Score : 0.8464285714285713
Support Vector Machine Accuracy Score : 0.8482142857142856
Decision Tree Accuracy Score : 0.8892857142857145
KNN Accuracy Score : 0.8482142857142858
```

- Decision tree has highest accuracy.

Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.

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



Conclusions

We can conclude that

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!