IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

By,
Advait Krishna

Outline

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

Executive Summary

- Summary of methodologies

  The project involves a comprehensive data-driven journey, starting with the collection of data through API interactions, which allows for the retrieval of relevant information from various sources. Subsequently, web scraping is employed as another robust method for data collection, expanding the scope of available data. Following the data gathering phase, the focus shifts to data wrangling, where the collected raw data is organized and cleaned to ensure its suitability for analysis. The exploration begins with Exploratory Data Analysis (EDA) using SQL queries to extract meaningful insights from structured datasets. This is complemented by visual exploration techniques, utilizing data visualization tools to present patterns and trends effectively. Further enhancing the analytical depth, interactive visual analytics with Folium are employed, providing an immersive experience in geospatial data representation. The project then advances into the realm of machine learning prediction, leveraging the insights gained from previous stages to develop models for forecasting and decision-making. This comprehensive approach ensures a systematic and thorough exploration of the dataset, leading to valuable insights and informed outcomes.

- Summary of all results

  The success rates of launch sites exhibit a positive correlation with increasing flight numbers, indicating a trend where a higher flight number corresponds to a greater likelihood of success. Among specific orbital categories, including ES-L1, GEO, HEO, SSO, and VLEO, the success rates are notably high, suggesting the efficiency and reliability of launches into these orbits. Notably, missions with heavy payloads, particularly those destined for Polar, LEO, and ISS, demonstrate higher rates of successful landings. Analyzing the temporal aspect, the success rate has shown a consistent upward trajectory from 2013 to 2020, reflecting an overall improvement in launch outcomes. Geographically, all launch sites are strategically located near the equator and in close proximity to the American coast, specifically in Florida and California. Despite their coastal positioning, launch sites maintain a considerable distance from highways, railways, and urban areas. Furthermore, the success rates vary based on payload weight, with low-weighted payloads (0-4000kg) exhibiting a higher success rate compared to heavier payloads (4000-10000kg). Among the launch sites, KSC LC – 39A stands out with a remarkable success rate of 76.9% and a failure rate of 23.1%, marking it as the site with the highest success rate among the three considered launch locations.

Introduction

- Project background and context

  SpaceX promotes Falcon 9 rocket launches on its website at a price of 62 million dollars, significantly lower than the costs associated with other providers, which can reach upwards of 165 million dollars per launch. A substantial portion of this cost advantage stems from SpaceX's innovative ability to reuse the first stage of the rocket. Consequently, the success or failure of landing the first stage plays a crucial role in determining the overall launch cost. This information holds strategic significance for potential competitors seeking to bid against SpaceX for rocket launches. The primary objective of this project is to develop a machine learning pipeline capable of predicting the successful landing of the first stage, thereby providingvaluable insights into the cost estimation for rocket launches.

- Problems you want to find answers
  - What influences the successful landing of a rocket?
  - The interplay of different factors that contribute to the success rate of a rocket's landing.
  - What operational conditions must be established to guarantee the success of a landing program?

Section 1

Methodology

Methodology

Executive Summary

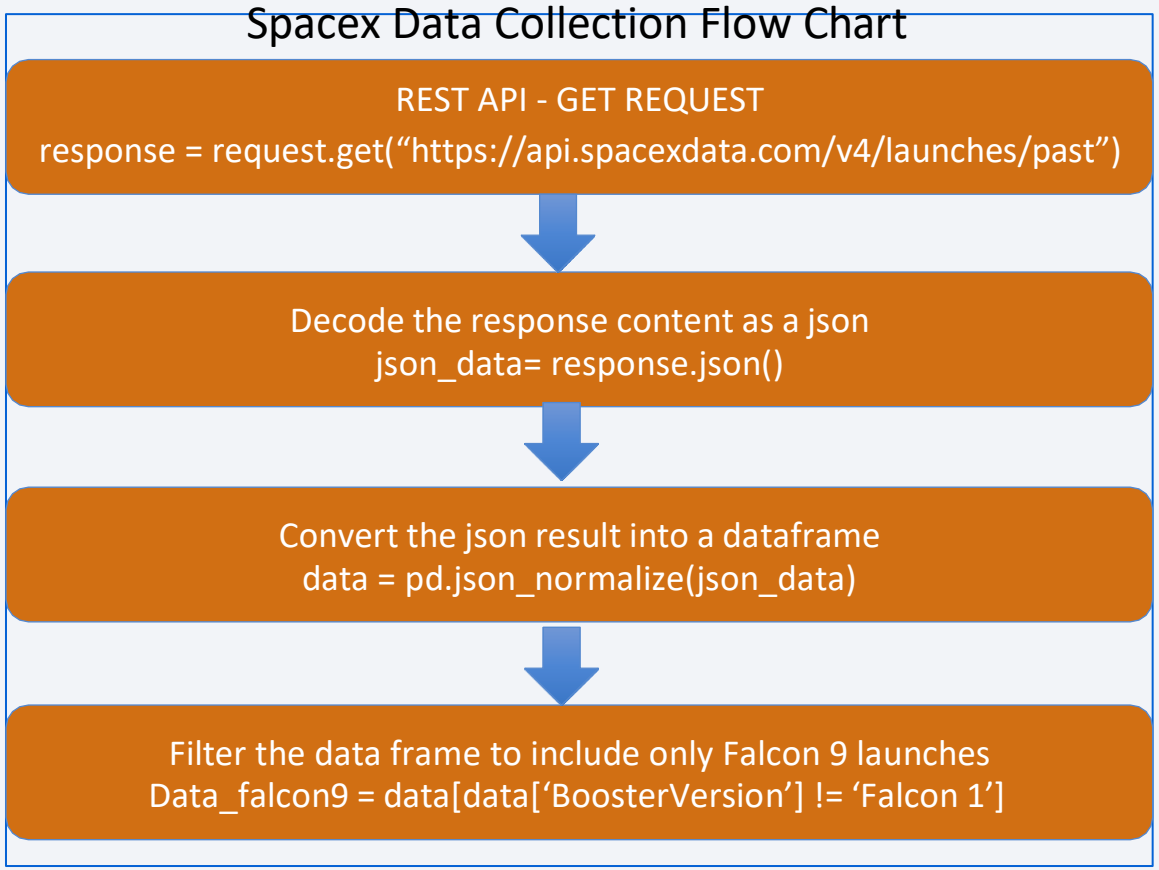
- Data was collection through the SpaceX REST API and Web Scrapping from Wikipedia website
- Perform data wrangling
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

- Data Collection Process and Flowcharts are described in following slides

Data Collection – SpaceX API

- First, we collected data from SpaceX API with get request method, then we clean the requested data as we needed and then perform some data wrangling and formatting on it.

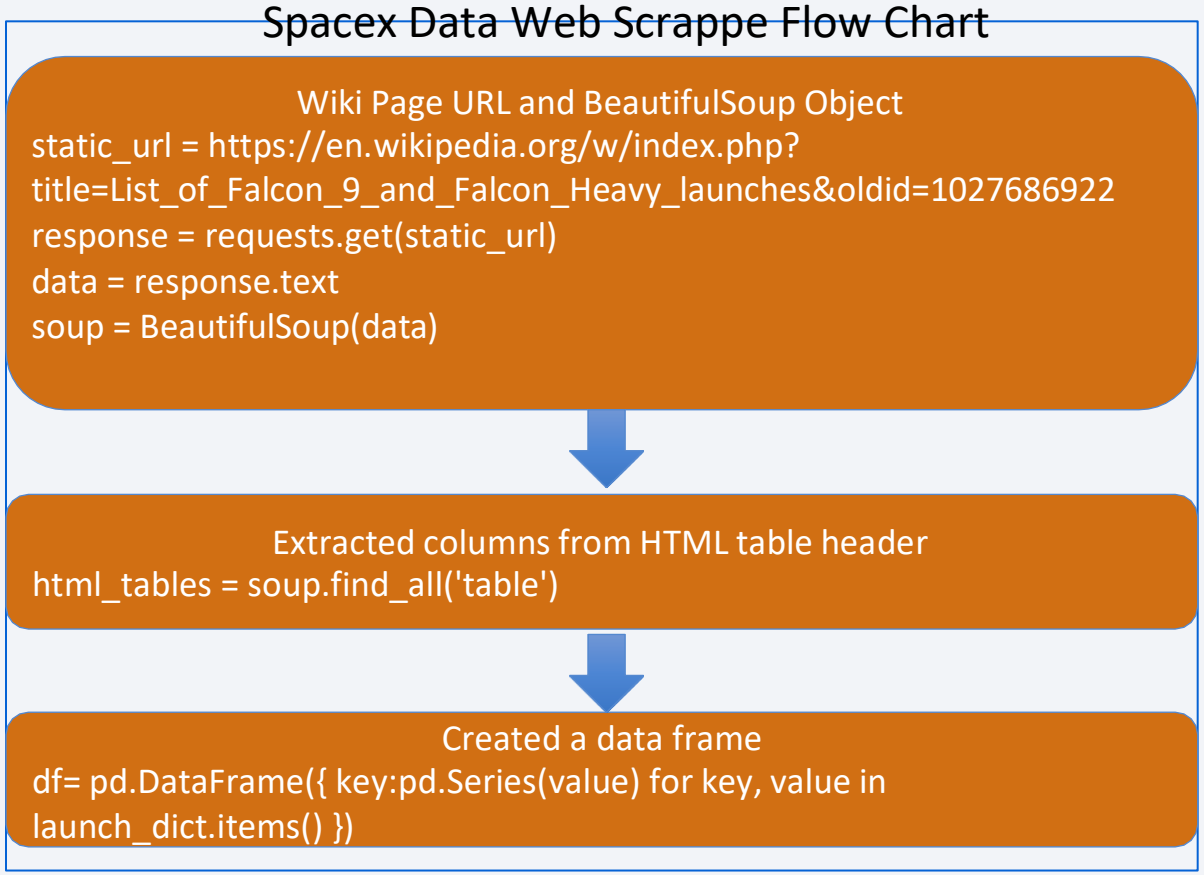


The GitHub URL of the SpaceX API calls notebook

https://github.com/sureshmanjoshi/testrepo/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- First, we request the Falcon9 launch Wiki page from its URL and create a BeautifulSoup object from the HTML response.
- Then extracted all column names from the HTML table header.
- Finally, created a data frame by parsing the launch HTML Table.

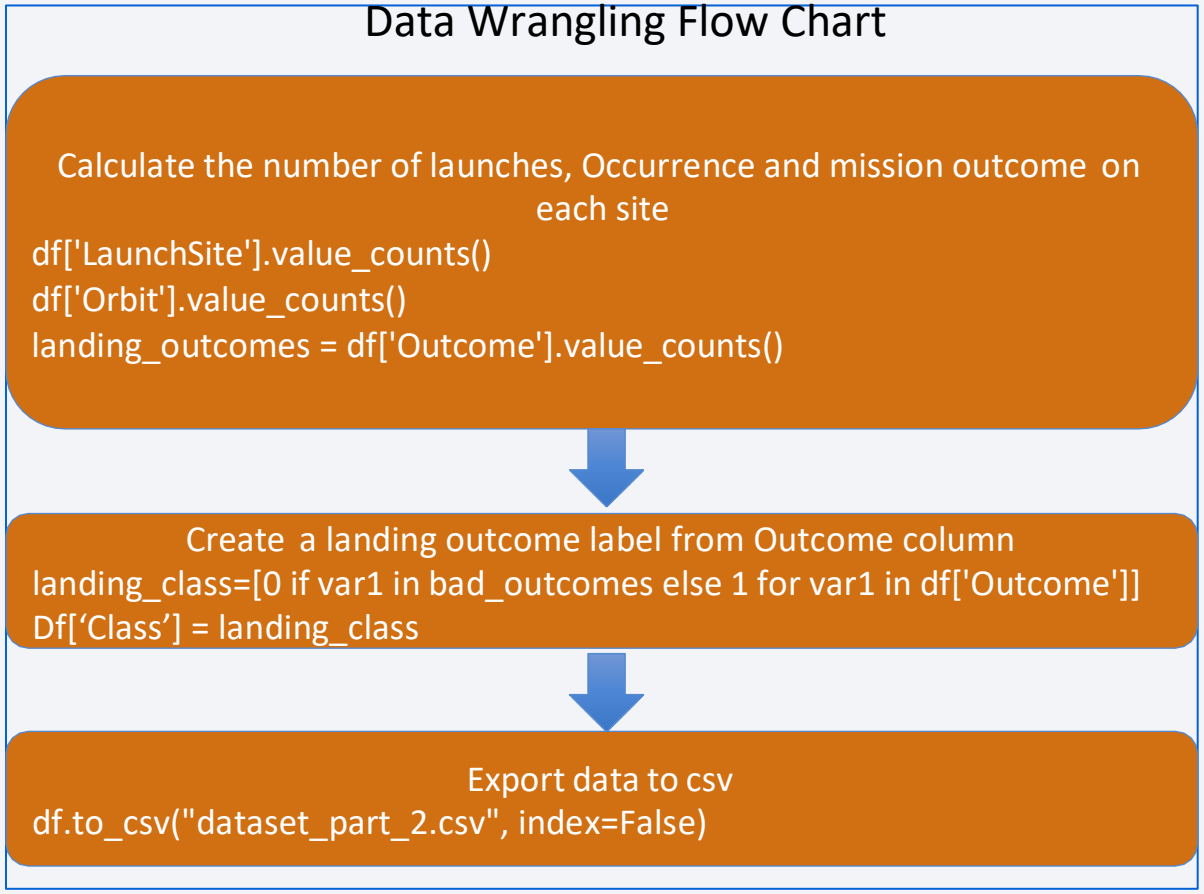


GitHub URL link of the web scraping notebook:

https://github.com/sureshmanjoshi/testrepo/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

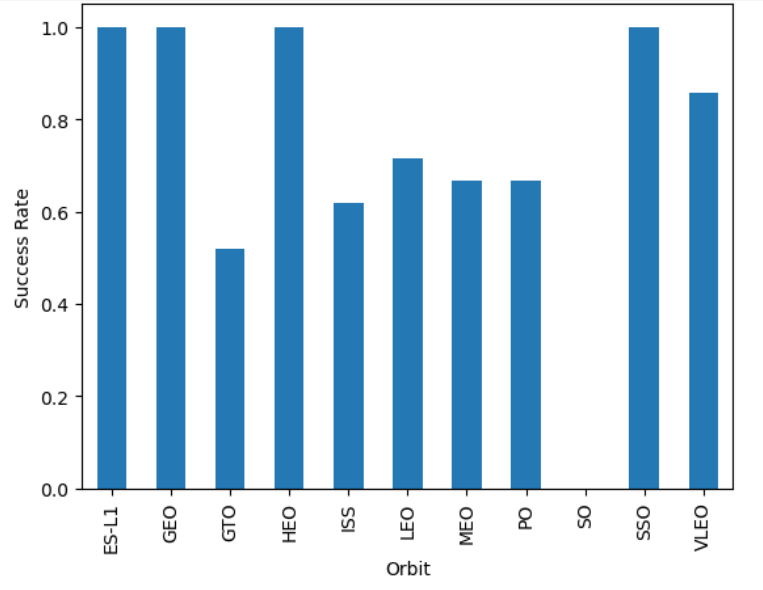
- To find data pattern and training label, we perform Exploratory Data Analysis(EDA).
- We calculated each site's number of launch, number and occurrence and mission outcome of each orbit.
- Finally, created a landing outcome label from Outcome columns and exported the csv data.

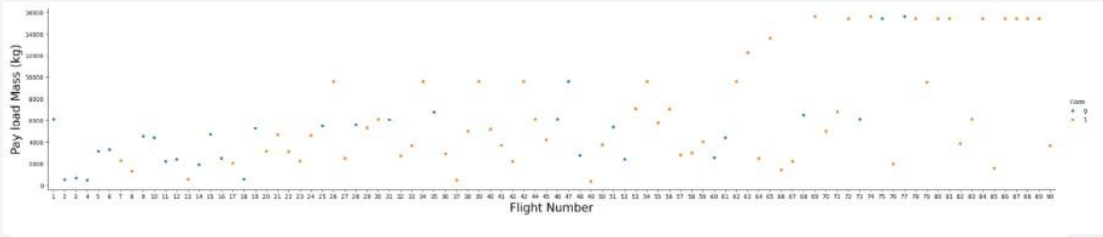


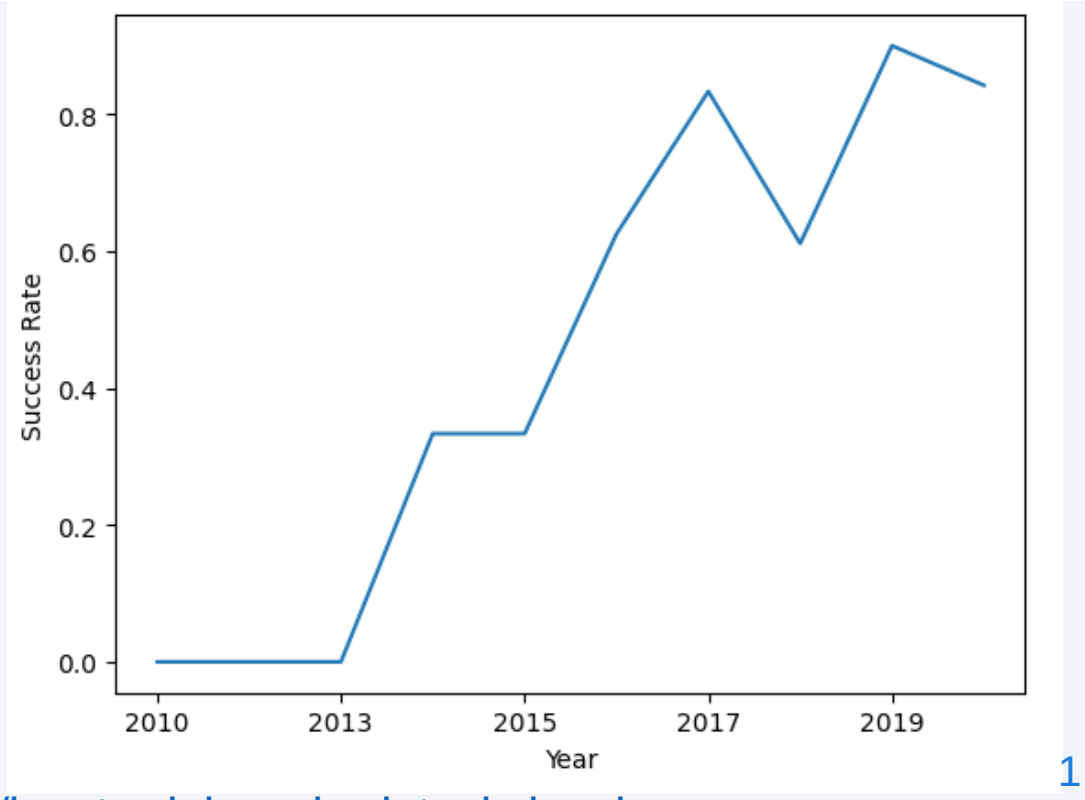
GitHub URL of data wrangling related notebooks:
https://github.com/sureshmanjoshi/testrepo/blob/main/labs-jupyter-spacex-Data wrangling.ipynb

EDA with Data Visualization

- The data exploration phase involved visualizing relationship between Flight Number and Launch sites, Payload Mass and Launch Site , success rate of each orbit type, Flight Number and Orbit type and Yearly Launch Success.







- GitHub URL of EDA with data visualization notebook:

https://github.com/sureshmanjoshi/testrepo/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- We used SQlite database to load Spacex data
- We used SQL queries to get following information from the data:
  - Finding the names of unique launch sites in the space mission
  - Find out the total payload mass carried by boosters launched by NASA
  - Find out average payload mass carried by booster version F9 v1.1
  - Find the first successful landing outcome in ground pad was achieved]
  - Find the total number of successful and failure mission outcomes
  - Ranked the landing outcomes between 2010-06-04 to 2017-03-20
- GitHub URL of EDA with SQL notebook: https://github.com/sureshmanjoshi/testrepo/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- We added map objects such as markers, circles, lines to a folium map.
- We added circle for locate all launch sites in the map.
- We used marker color green for success (1) and red for failure (0) to a folium map.
- We used lines to calculate distance between launch site and selected points such as coastline, railway, highway, city.

- GitHub URL of interactive map with Folium map: https://github.com/sureshmanjoshi/testrepo/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- We created interactive Spacex Launch Record Dashboard.
- We plotted interactive pie chart to display success rate of launch sites. If selected all sites, it displays all sites success rate. If we select specific site, then it will plot the success and failure rate of the launch.
- We have also plotted scatter plot to display the correlation between payload and success of the launch site.
- GitHub URL of Plotly Dash lab: https://github.com/sureshmanjoshi/testrepo/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- We used Pandas to load data into data frame. Then we manipulate some data using Numpy for required format. Then we standardized the data using StandardScalar and transform methods.
- We split the data into training and testing data. Then the models are trained and hyperparameters are selected using the GridSearchCV function.
- We used accuracy as the metric in our model. Then we used confusion matrix to evaluate the performance of our model.
- We found the classification model as best performance model.
- GitHub URL predictive analysis lab: https://github.com/sureshmanjoshi/testrepo/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.ipynb

Predictive Analysis (Classification) - Flow chart

```
from js import fetch
import io

URL1 = "https://cf-courses-data.s3.us.cloud-object-stora
resp1 = await fetch(URL1)
text1 = io.BytesIO((await resp1.arrayBuffer()).to_py())
data = pd.read_csv(text1)
```

```
URL2 = 'https://cf-courses-data.s3.us.cloud-object-storag
resp2 = await fetch(URL2)
text2 = io.BytesIO((await resp2.arrayBuffer()).to_py())
X = pd.read_csv(text2)
```

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

```
# students get this
transform = preprocessing.StandardScaler()
scaler = preprocessing.StandardScaler().fit(X)
X = scaler.transform(X)
```

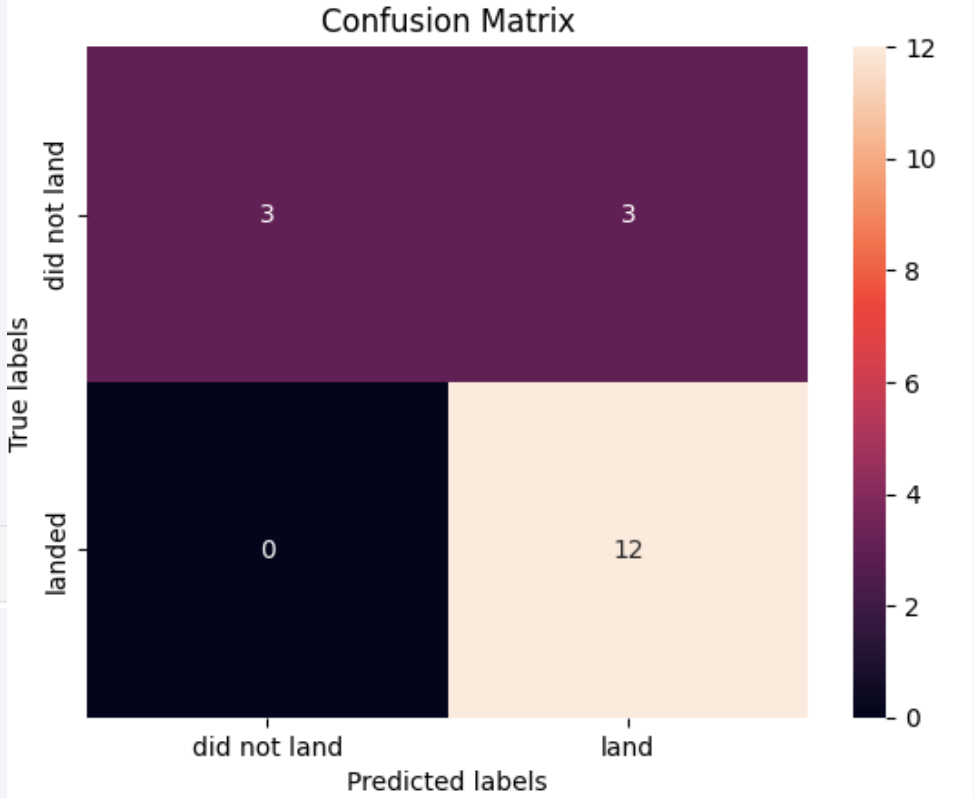
```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)
```

```
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}#
lr=LogisticRegression()
gscv = GridSearchCV(lr, parameters, scoring='accuracy', cv=10)
logreg_cv = gscv.fit(X_train, Y_train)
```

```
logreg_cv.score(X_test, Y_test)
```

```
0.8333333333333334
```

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



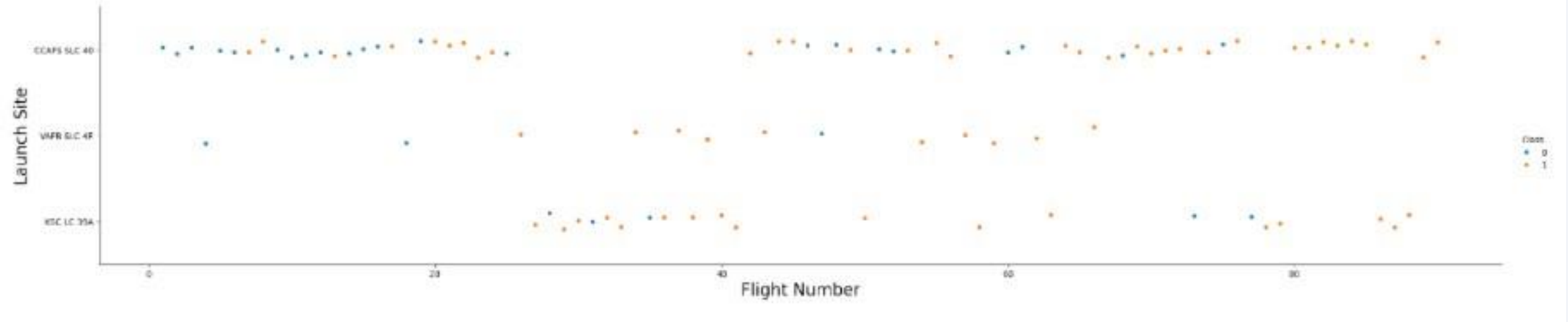
Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

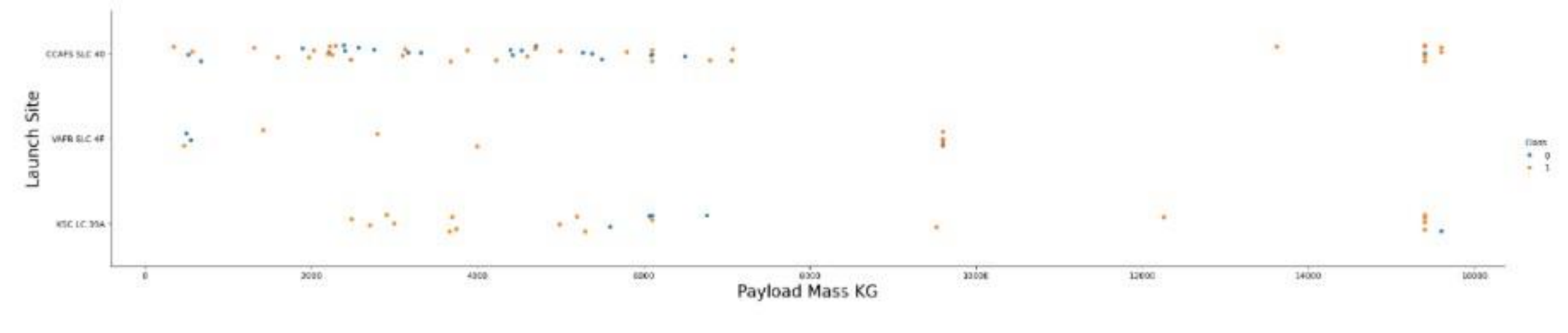
Insights drawn from EDA

Flight Number vs. Launch Site



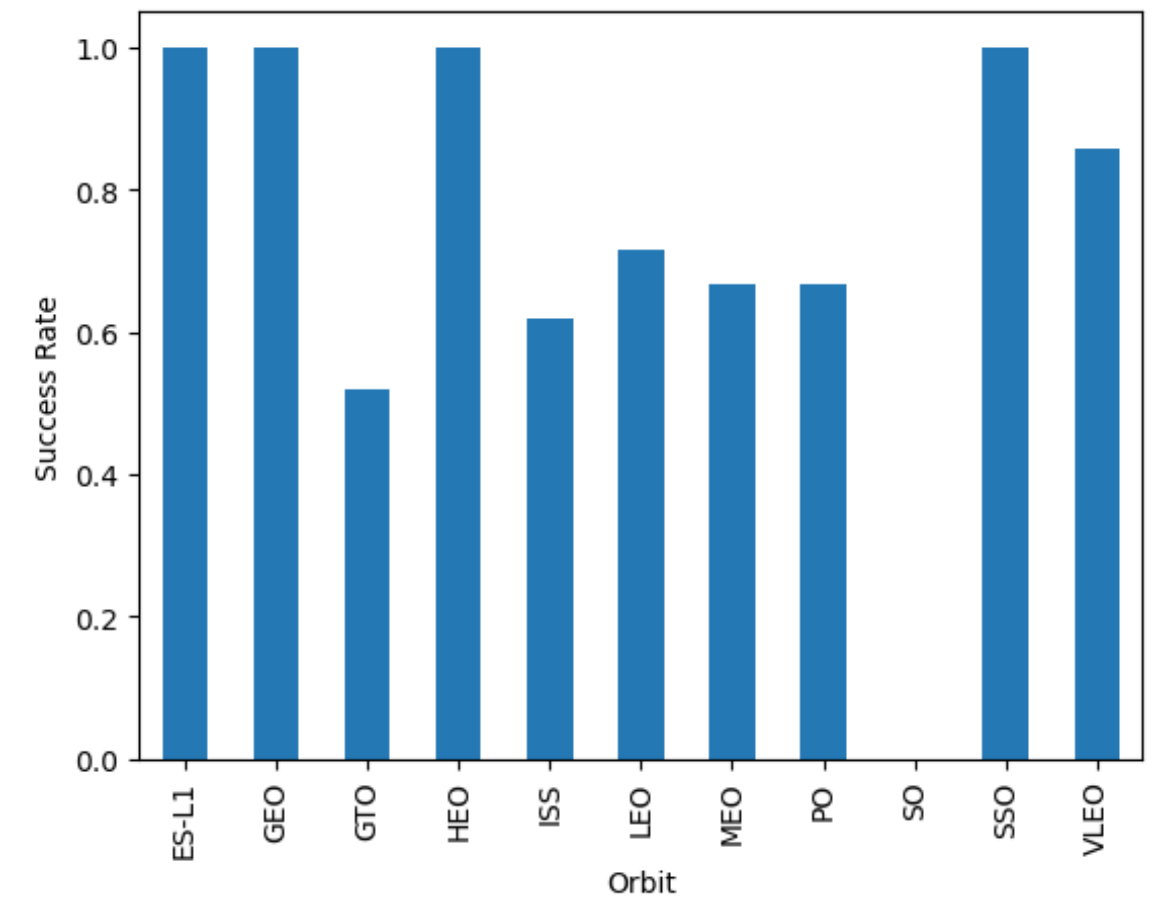
- From the above plot, we see that the greater the flight number greater the success rate of launch site.

Payload vs. Launch Site



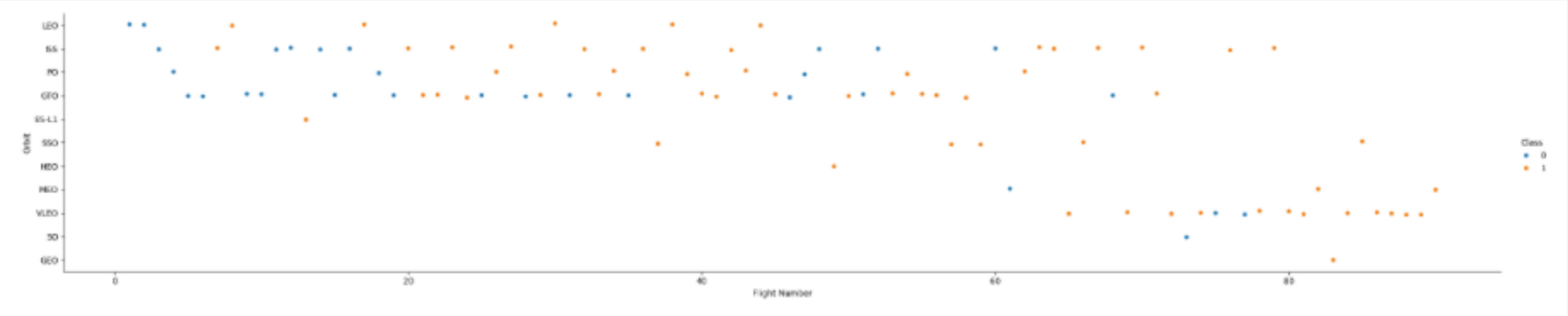
- From the above plot, we will find that for VAFB-SLC launch site there are no rockets launched for heavypayload mass > 10000

Success Rate vs. Orbit Type



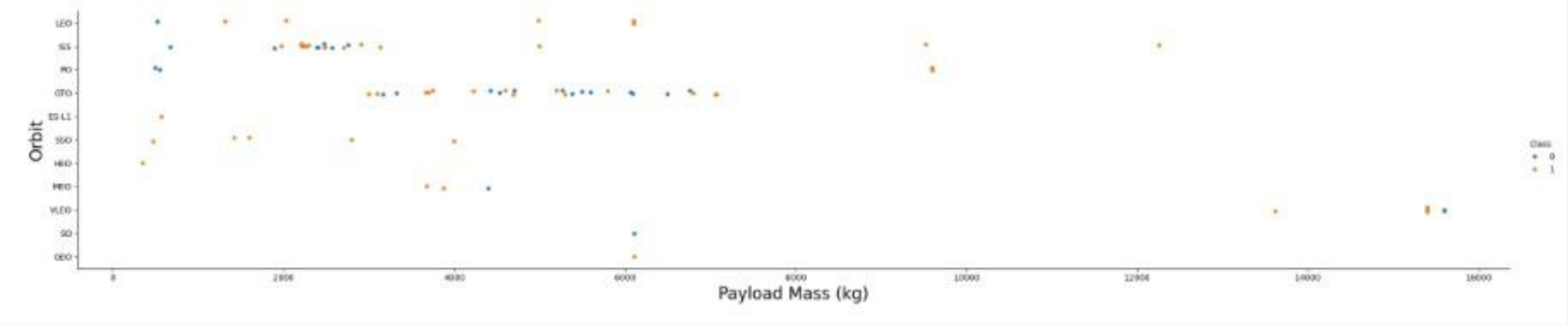
- From the Bar Plot, it is clear that ES-L1, GEO, HEO, SSO and VLEO had the most success rate.

Flight Number vs. Orbit Type



- We can 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.

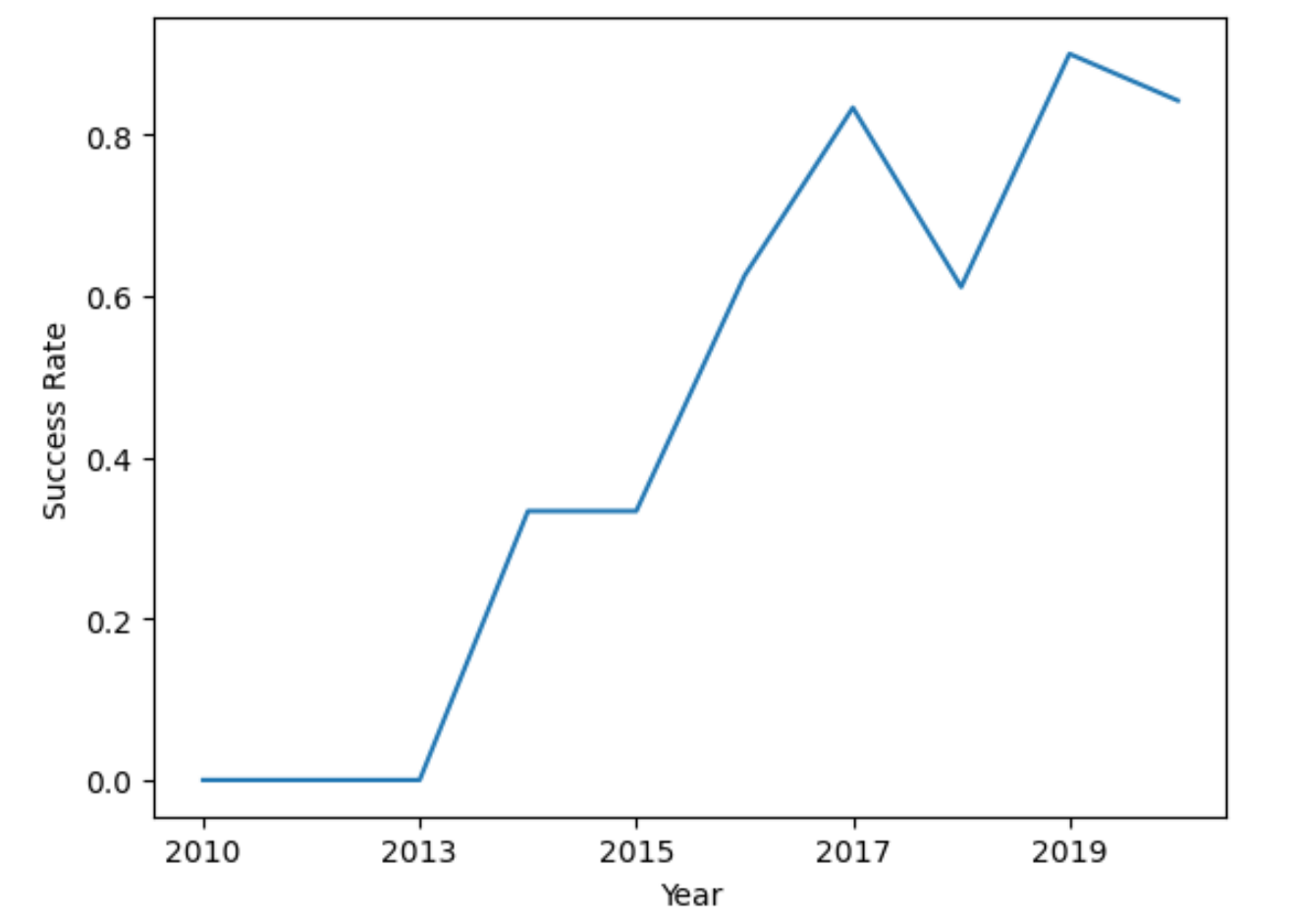
Payload vs. Orbit Type



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

Launch Success Yearly Trend

- We can observe that the success rate since 2013 kept increasing till 2020.



All Launch Site Names

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

```
[8]: %%sql

     select DISTINCT(Launch_Site) from spacextable
```

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

[8]:

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

- We used DISTINCT keyword in SQL query to show unique launch sites of SpaceX data set.

Launch Site Names Begin with 'CCA'

- We used like and limit keywords to display following result

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

```
[13]: %sql select * from spacextable where Launch_Site like "CCA%" limit 5
 * sqlite:///my_data1.db
Done.
```

[13]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- We used sum() method in SQL Query to calculate total payload mass.

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

```
[14]: %sql select sum(PAYLOAD_MASS__KG_) as "Total Payload Mass" from spacextable where Customer = "NASA (CRS)"
```

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

[14]:

Total Payload Mass
45596

Average Payload Mass by F9 v1.1

- We used average() function to calculate average payload mass.

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

```
[16]: %sql select avg(PAYLOAD_MASS__KG_) as "Average Payload Mass" from spacextable where Booster_Version = "F9 v1.1"
```

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

[16]:

Average Payload Mass
2928.4

First Successful Ground Landing Date

- We used min() function and applied group by keyword on customer.

```
[24]: %sql select Customer, min(Date) as Date from spacextable where Landing_Outcome = "Success (ground pad)" group by Customer
 * sqlite:///my_data1.db
Done.
```

[24]:

Customer	Date
NASA (CRS)	2016-07-18
NRO	2017-05-01
Northrop Grumman	2018-01-08
Orbcomm	2015-12-22
U.S. Air Force	2017-09-07

Successful Drone Ship Landing with Payload between 4000 and 6000

- We used and condition to determine payload mass between 4000 to 6000 and used where clause to filter successful drone ship landing.

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

```
[28]: %sql select Booster_Version, PAYLOAD_MASS__KG_ from spacextable where Landing_Outcome = "Success (drone ship)" and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
 * sqlite:///my_data1.db
Done.
```

[28]:

Booster_Version	PAYLOAD_MASS__KG_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

Total Number of Successful and Failure Mission Outcomes

- We applied count() function on Mission_Outcome and group by keyword .

List the total number of successful and failure mission outcomes

[31]:
```
%sql select Mission_Outcome, count(Mission_Outcome) as "Total Number" from spacextable group by Mission_Outcome
```

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

[31]:

Mission_Outcome	Total Number
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- We use subquery and max() function to determine Boosters carried maximum payload.

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

[10]:
```
%%sql
select Booster_Version, PAYLOAD_MASS__KG_ as "Payload_Mass" from spacextable where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from spacextable)
```

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

[10]:

Booster_Version	Payload_Mass
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

- Sqlite does not support month names, so we used substr() function on date field to split month and date and where condition for filter.

```
[41] %sql select substr(Date,6,2) as Month, substr(Date, 0,5) as Year, Booster_Version, Launch_Site, Landing_Outcome from spacextable where substr(Date,0,5) = '2015' and Landing_Outcome = "Failure (drone ship)"
 * sqlite:///my_data1.db
Done.
```

[41]

Month	Year	Booster_Version	Launch_Site	Landing_Outcome
01	2015	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- We used count() function and group by keyword to determine rank and used where clause with between and and conditions to filter data.

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.

```
[44]: %sql select Date, Landing_Outcome, count(Landing_Outcome) as "Landing Outcome" from spacextable where Date between "2010-06-04" and "2017-03-20" group by Landing_Outcome order by count(Landing_Outcome) desc
 * sqlite:///my_data1.db
Done.
```

[44]:

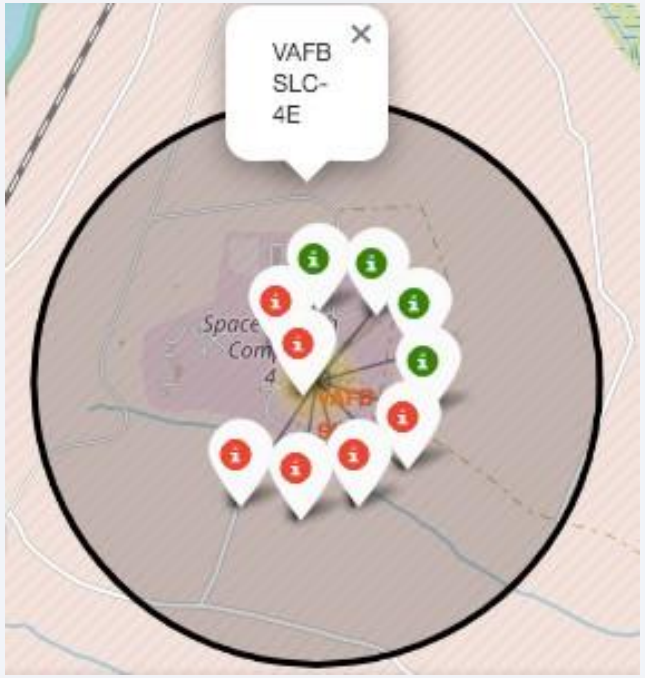
Date	Landing_Outcome	Landing Outcome
2012-05-22	No attempt	10
2016-04-08	Success (drone ship)	5
2015-01-10	Failure (drone ship)	5
2015-12-22	Success (ground pad)	3
2014-04-18	Controlled (ocean)	3
2013-09-29	Uncontrolled (ocean)	2
2010-06-04	Failure (parachute)	2
2015-06-28	Precluded (drone ship)	1

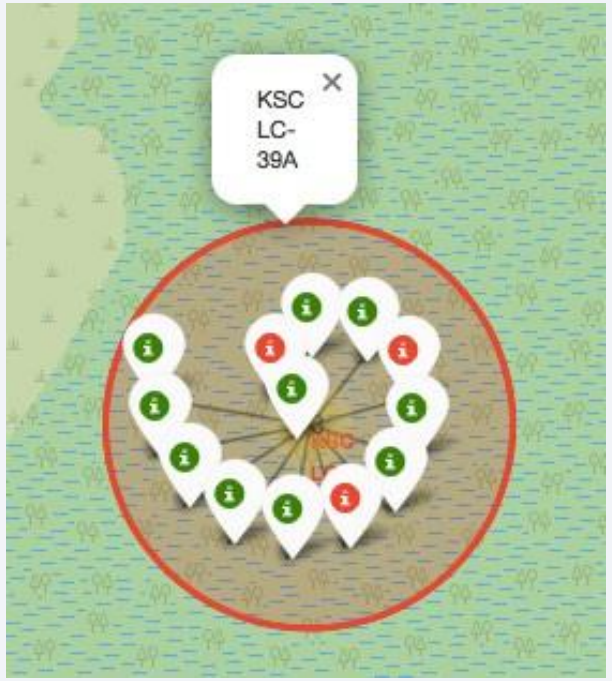

Section 3
Launch Sites
Proximities Analysis

Launch Sites in global map

- We can see that all launch sites are in proximity to the equator line and all launch sites are very close proximity to the American coast, Florida and California.

Launch sites with color markers



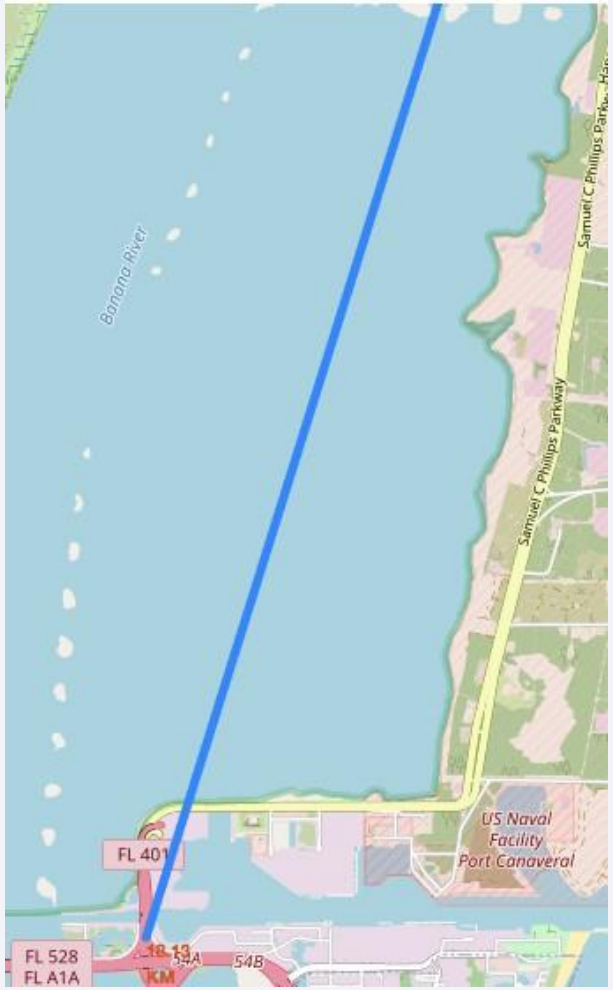


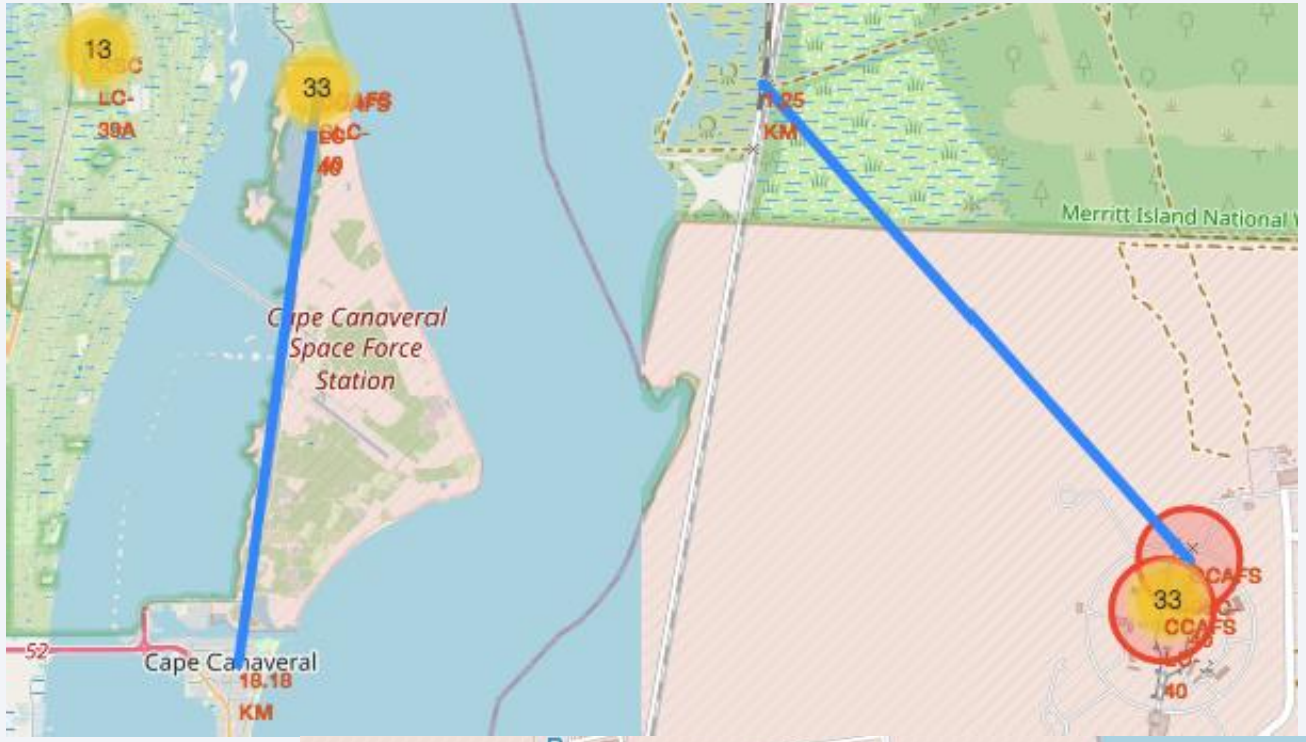
California

Florida

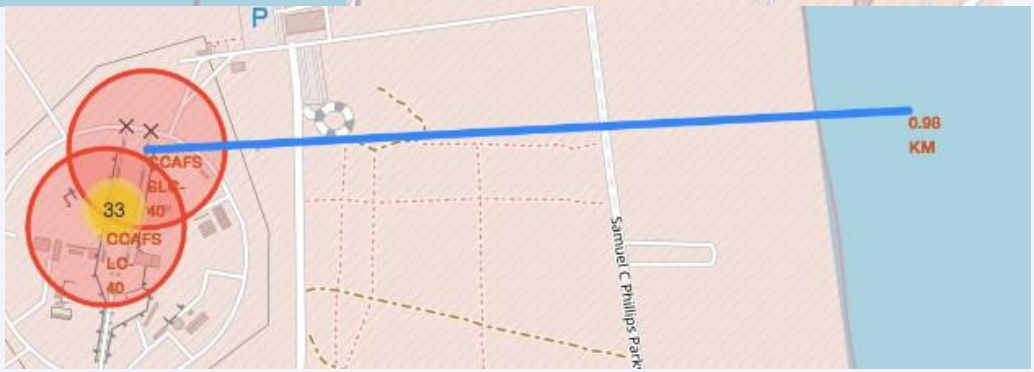
Green Marker indicates Successful launch and Red Marker indicates Failure launch.

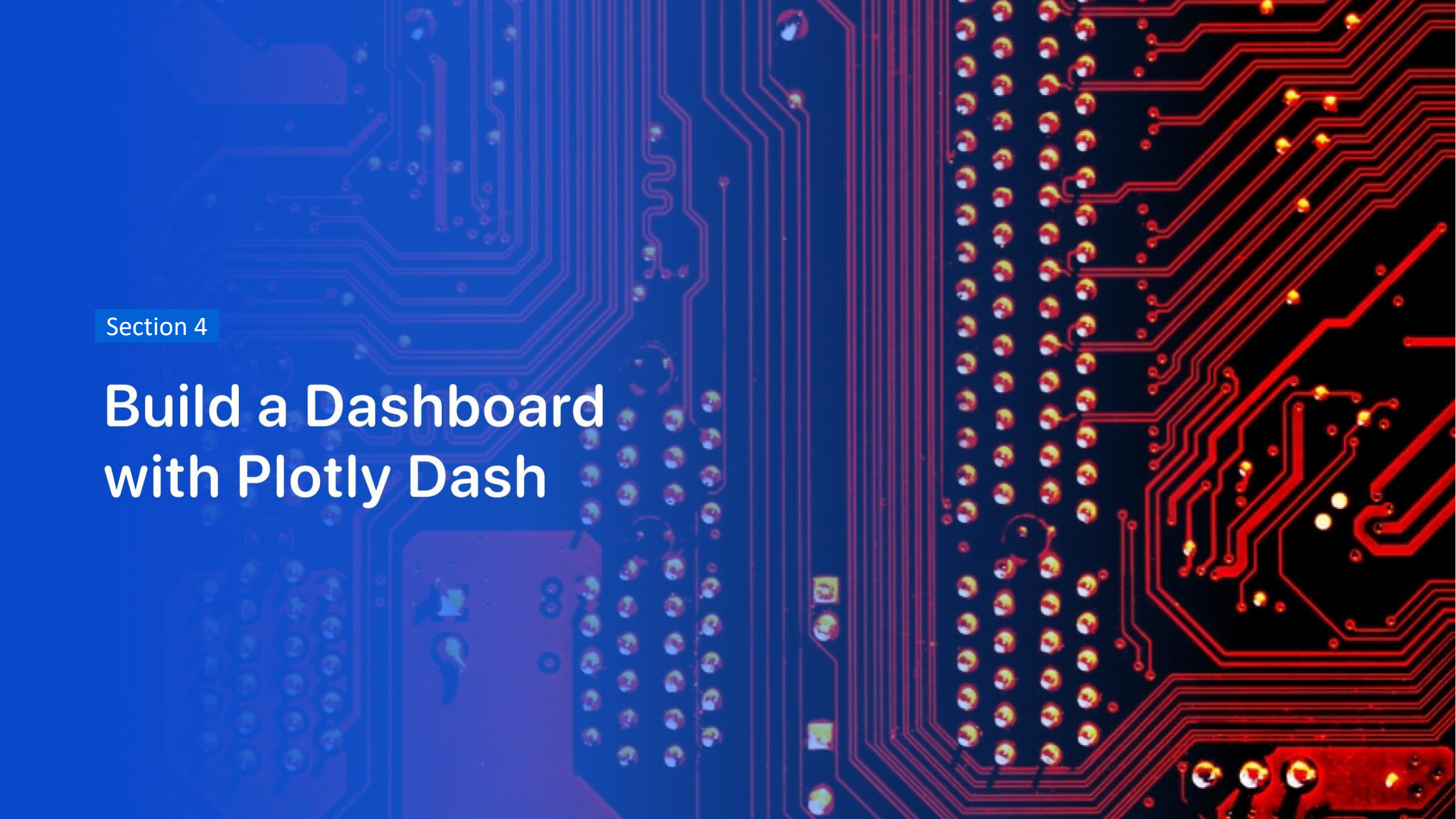
Launch site distance from Coast, Railway, Highway and City



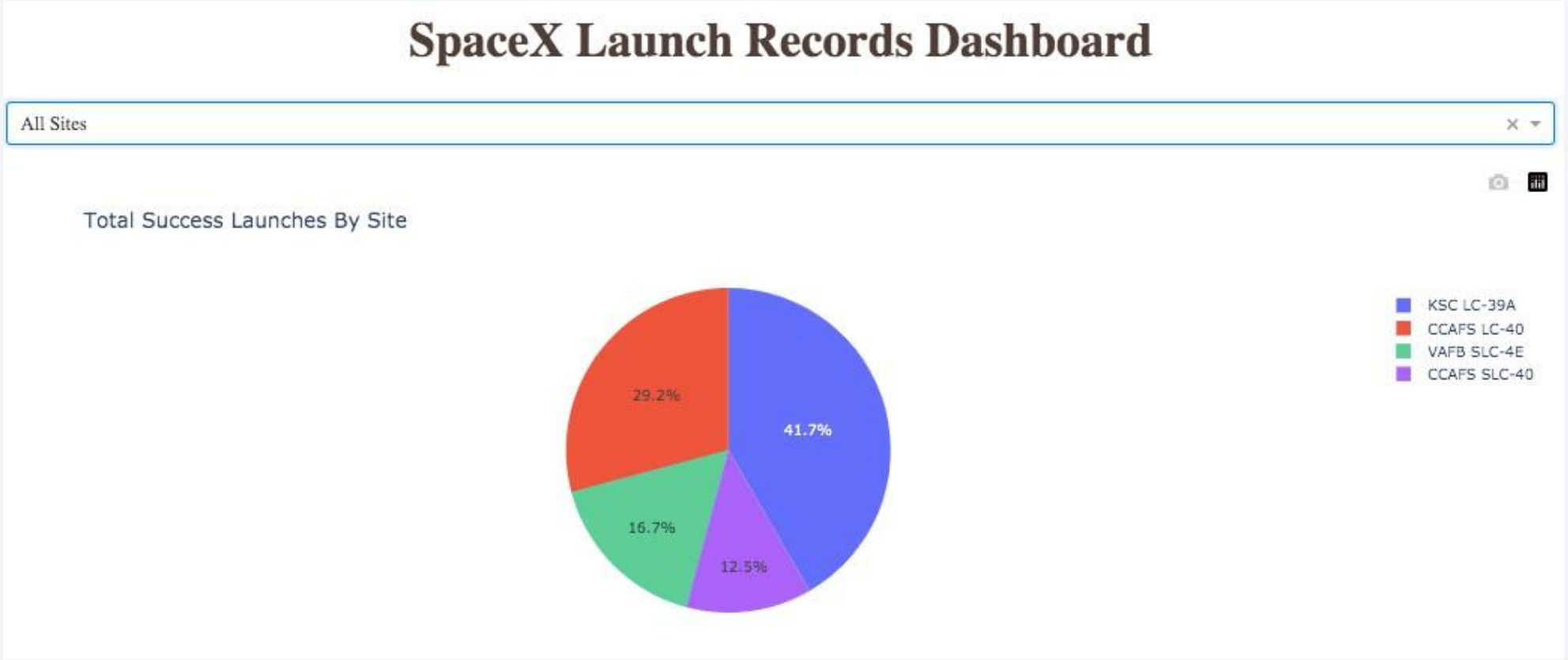


Launch sites are close proximity to the coast but distance away from Highway, Railway and City



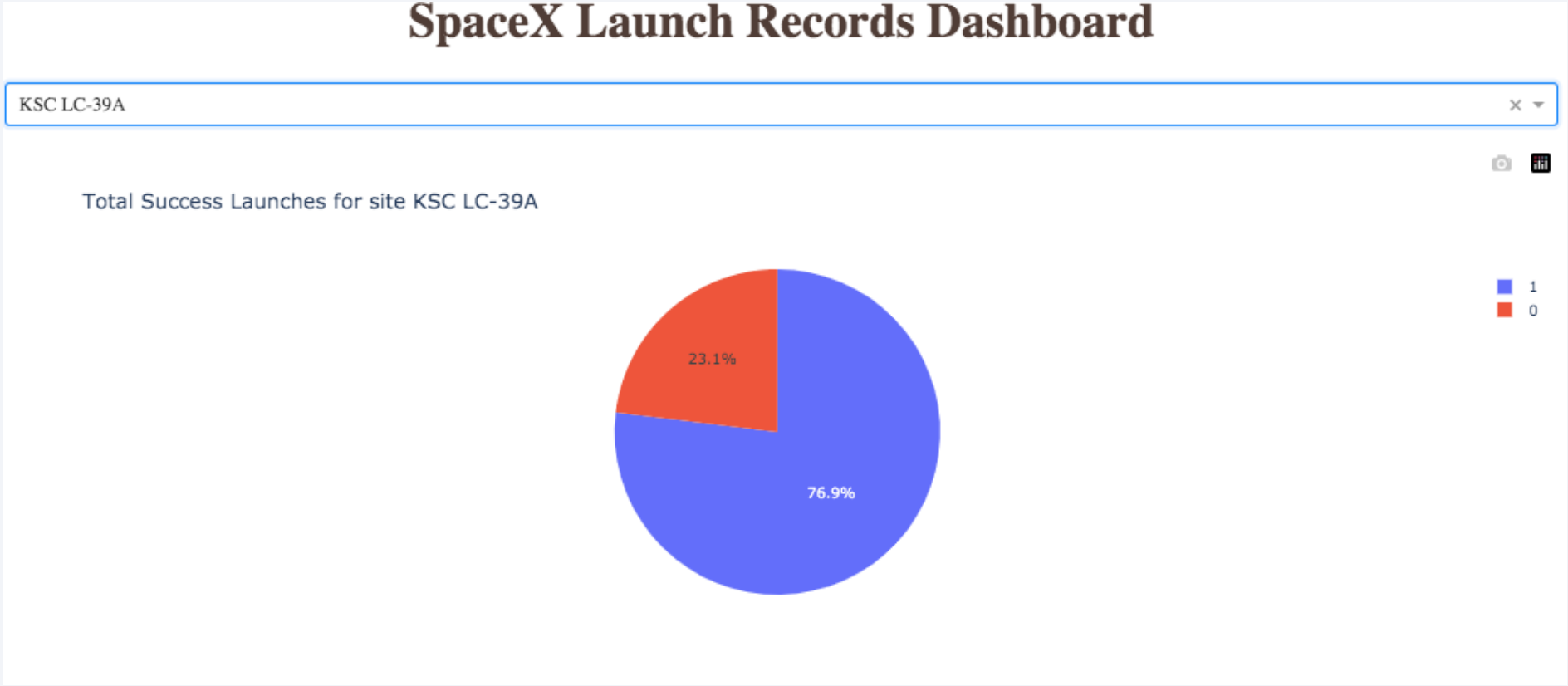

Section 4
Build a Dashboard with Plotly Dash

Success percentage of each launch site with Pie chart



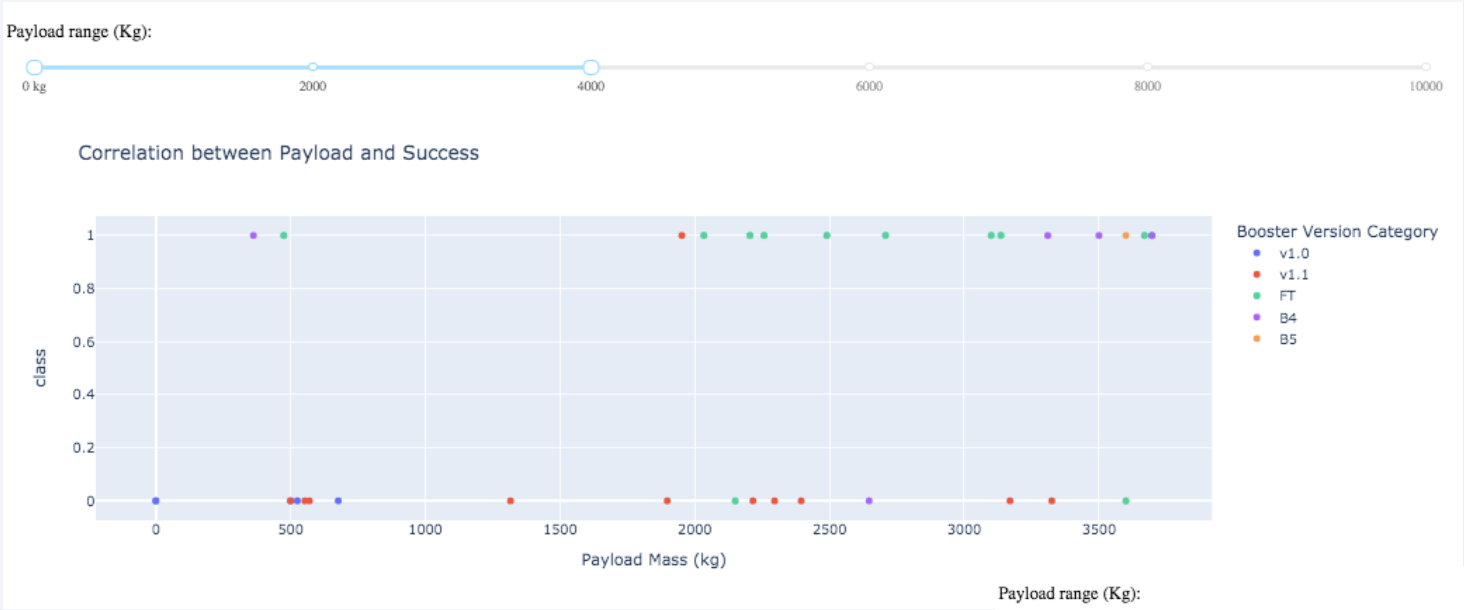
- KSC LC－39A has most success rate of 41.7% among all other launch site

Launch site with highest launch success rate with Pie chart



- KSC LC - 39A has 76.9% success and 23.1% failure rate, which is highest success rate among other three launch sites.

Payload vs. Launch Outcome scatter plot



0kg to 4000 kg payload

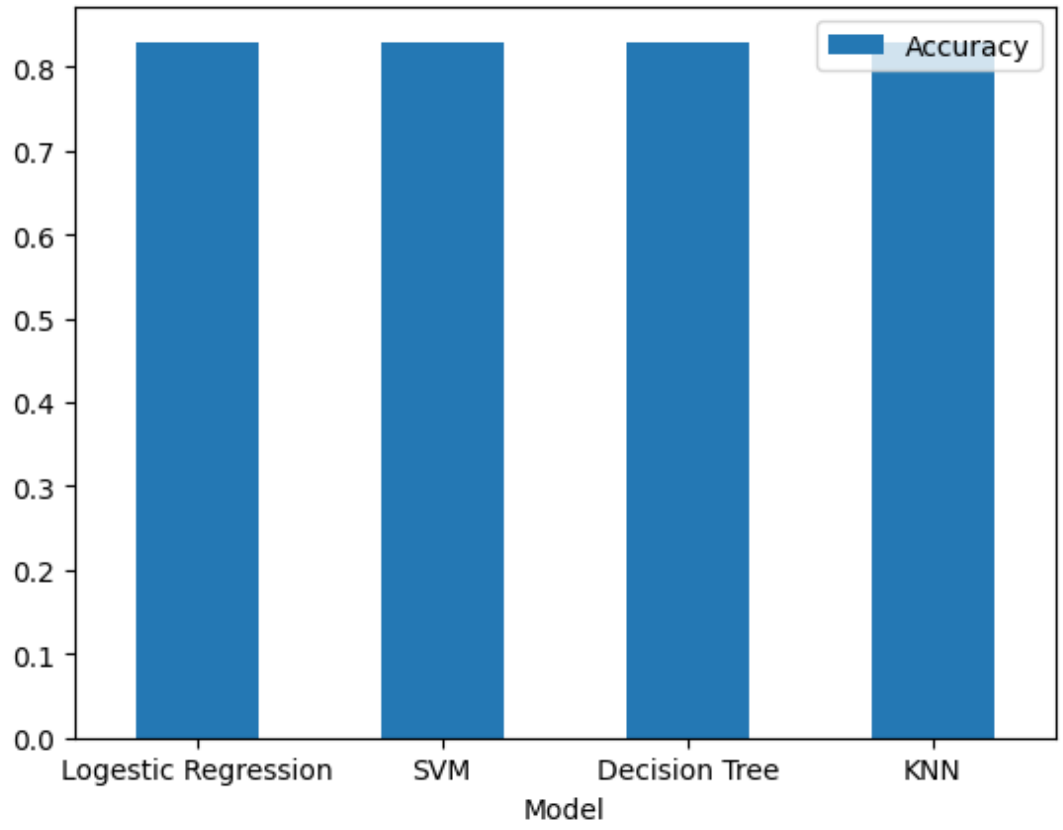
4000 kg to 10000 kg payload

- It is clearly seen that low weighted payload(0-4000kg) has most success rate than heavy weighted(4000-10000kg) payload
- FT Booster Version has the highest launch success rate.

Section 5

Predictive Analysis (Classification)

Classification Accuracy



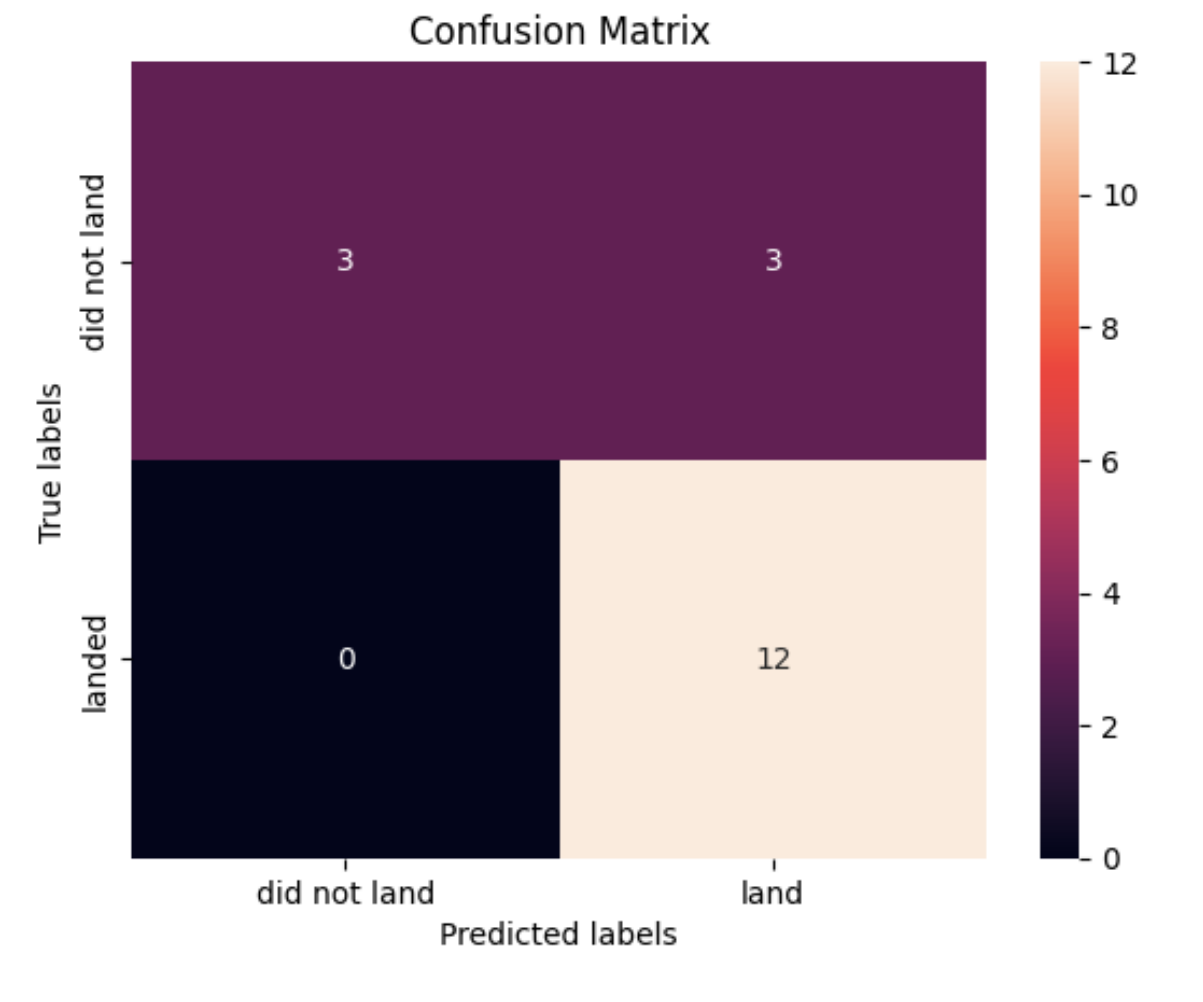
Find the method performs best:

```
print("Logestic Regression Score: ", logreg_cv.score(X_test, Y_test))
print("Support Vector Machine Score: ",svm_cv.score(X_test, Y_test))
print("Decision Tree Score: ",tree_cv.score(X_test, Y_test))
print("K Nearest Neighbors Score: ",knn_cv.score(X_test, Y_test))
```

```
Logestic Regression Score:  0.8333333333333334
Support Vector Machine Score:  0.8333333333333334
Decision Tree Score:  0.8333333333333334
K Nearest Neighbors Score:  0.8333333333333334
```

- They all perform practically the same.

Confusion Matrix



Examining the confusion matrix, we see that models can distinguish between the different classes. We see that the major problem is false positives.

Conclusions

- The greater the flight number greater the success rate of launch site.
- ES-L1, GEO, HEO, SSO and VLEO had the most success rate.
- With heavy payloads the successful landing rate are more for Polar, LEO and ISS.
- The success rate since 2013 kept increasing till 2020.
- All launch sites are in proximity to the equator line and all launch sites are very close proximity to the American coast, Florida and California
- Launch sites are close proximity to the coast but distance away from Highway, Railway and City
- Low weighted payload (0-4000kg) has most success rate than heavy weighted (4000-10000kg) payload
- KSC LC – 39A has 76.9% success and 23.1% failure rate, which is highest success rate among other three launch sites.

Appendix

- The GitHub URL of the SpaceX API calls notebook:
  https://github.com/sureshmanjoshi/testrepo/blob/main/jupyter-labs-spacex-data-collection-api.ipynb
- GitHub URL of the web scraping notebook:
  https://github.com/sureshmanjoshi/testrepo/blob/main/jupyter-labs-webscraping.ipynb
- GitHub URL of data wrangling related notebooks:
  https://github.com/sureshmanjoshi/testrepo/blob/main/labs-jupyter-spacex-Data wrangling.ipynb
- GitHub URL of EDA with data visualization notebook:
  https://github.com/sureshmanjoshi/testrepo/blob/main/jupyter-labs-eda-dataviz.ipynb
- GitHub URL of EDA with SQL notebook:
  https://github.com/sureshmanjoshi/testrepo/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb
- GitHub URL of interactive map with Folium map:
  https://github.com/sureshmanjoshi/testrepo/blob/main/lab_jupyter_launch_site_location.ipynb
- GitHub URL of Plotly Dash lab:https://github.com/sureshmanjoshi/testrepo/blob/main/spacex_dash_app.py
- GitHub URL predictive analysis lab:
  https://github.com/sureshmanjoshi/testrepo/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.ipynb

Thank you!