Winning Space Race with Data Science

Vignesh G
13-01-2022

Outline

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

Executive Summary

Methodology

- Data Collection – SpaceX rest API
- Data Wrangling – Python-Pandas
- Exploratory Analysis – SQL, Pandas
- Perform interactive visual analytics using Folium and Plotly Dash
- Predicative Analysis- Sklearn-Classification (SVM, Decision Tree, Knn, Logistic regression)

Summary of result

- Interactive Dashboard made with Plotly-Dash that helps with easy visual analysis
- ML Model selection done to prediction if the first stage will land with 83.3% accuracy

Introduction

Project background and context

SpaceX advertises Falcon 9 rocket cost of 62 million dollars while other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

Problems find answers

Predict if the reusable first stage of the rocket will land or not.

Section 1

Methodology

Methodology

- Data collection methodology:
  - Data was collected by using various SpaceX REST API endpoints (https://api.spacexdata.com/v4/)
  - Another source of data was obtained by web-scrapping Wikipedia (https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches) using beautiful soup
- Perform data wrangling
  - Data wrangling was done using python to take care of missing values, performing one hot encoding and feature addition was done by merging results from multiple APIs
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - The cleaned data was split into train and test data, different classification ML models like SVM, Decision Tree, Knn, Logistic regression were trained using the training data, GridSearchCV was used to find the optimum hyperparameter for each of the model and the accuracy score was computed for each of the model on the test data. The model with the highest test accuracy score is selected for the final prediction.

Data Collection

- **Rest API**

Data was collected by using various SpaceX REST API endpoints (https://api.spacexdata.com/v4/)

- **Web Scrapping**

Another source of data was obtained by web-scrapping Wikipedia (https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches) using beautiful soup

Data Collection – SpaceX API

- The main data for the launch was collected using https://api.spacexdata.com/v4/launches/past

- Feature addition to the main data was done requesting data from different APIs

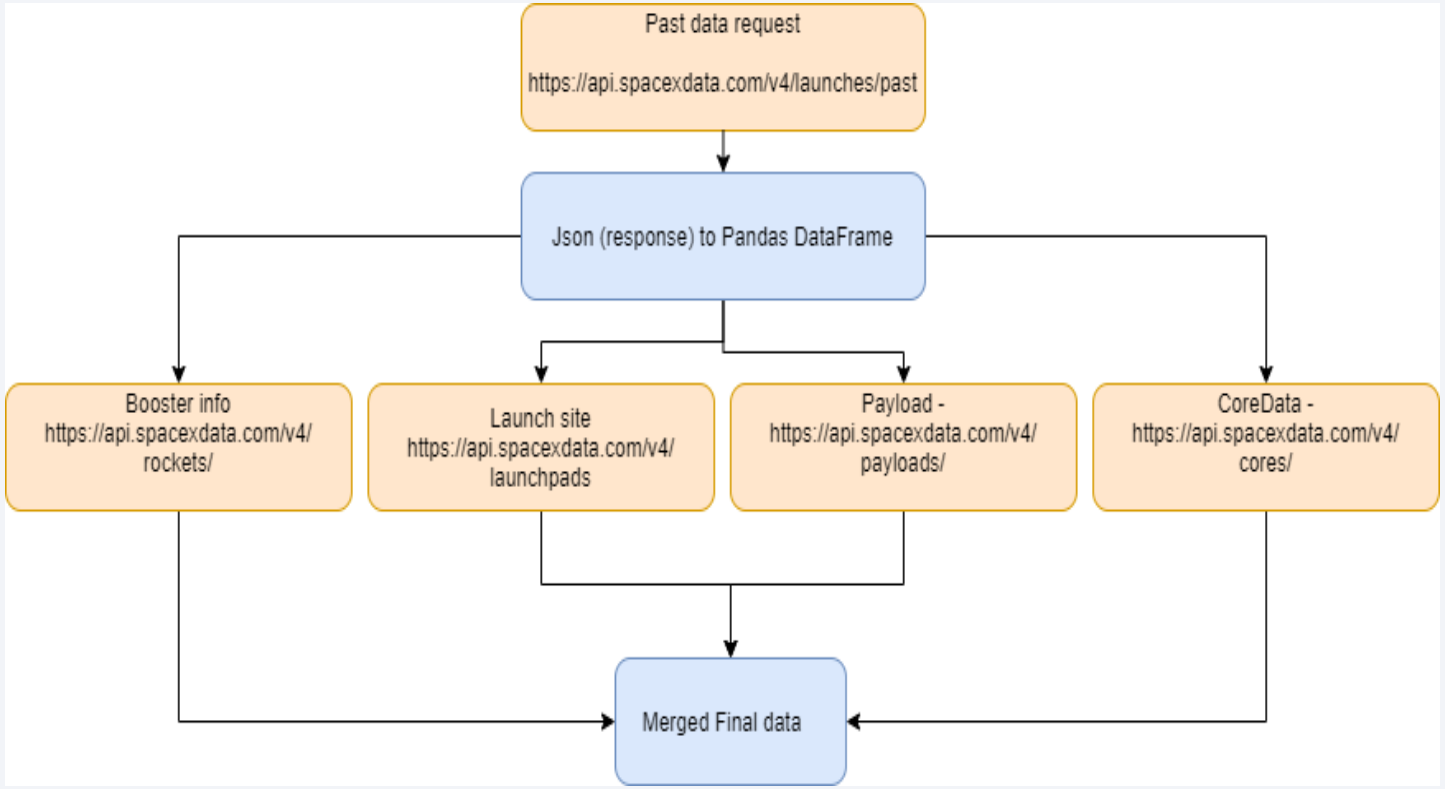
  Booster info https://api.spacexdata.com/v4/rockets/

  Launch site - https://api.spacexdata.com/v4/launchpads/

  Payload - https://api.spacexdata.com/v4/payloads/

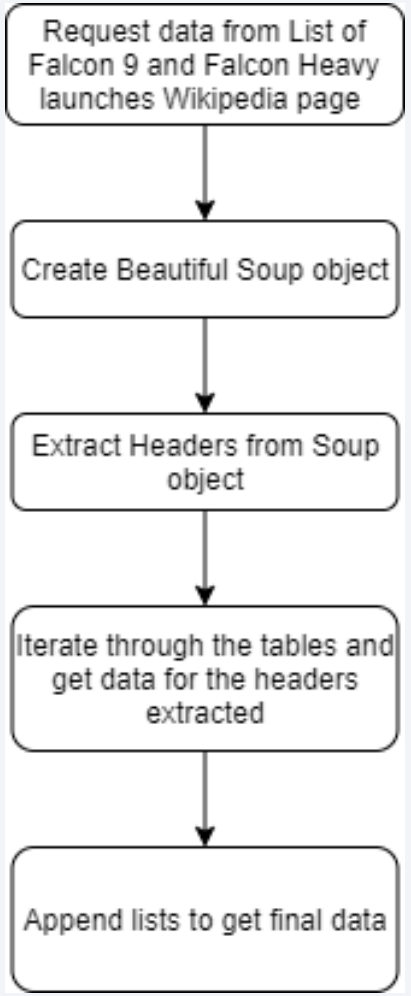
  CoreData - https://api.spacexdata.com/v4/cores/

- Notebook link – Click here



Data Collection - Scraping

- Extract a Falcon 9 launch records HTML table from Wikipedia
- Parse the table and convert it into a Pandas data frame
- Notebook link – Click here



Data Wrangling

- The data from multiple SpaceX APIs were merged to get the final dataset
- The data was sliced for only the columns required for the analysis
- Filtering was done for the following:
  - No of cores=1
  - No of payload=1
  - Booster version=Falcon1
- Missing vales :
  - Payload mass – Missing values replaced with mean
  - Landing pad – Retain as None

Notebook link – Click here

EDA with Data Visualization

- Scatter plots used:
  - Payload vs Flight no – As flight no increases success rate increases
  - Launch site vs Flight no –
  - Launch site vs Payload - VAFB-SLC has a limit of 10,000 kg
  - Orbit vs payload – SSO has 100% success, Higher payloads of PO have 100% success
- Bar Chart used:
  - Orbit vs Success mean – Success is highly dependent on type of orbit
- Line chart used:
  - Year vs Success – Success increased with time, meaning newer launches are more successful

Notebook link – Click here

EDA with SQL

- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string 'KSC'
- 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 where the successful landing outcome in drone ship was achieved.
- Listing the names of the boosters which have success in ground pad 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.
- Listing the records which will display the month names, successful landing outcomes in ground pad ,booster versions, launch site for the months in year 2017
- Ranking the count of successful landing outcomes between the date 2010-06-04 and 2017-03-20 in descending order

Notebook link – Click here

Build an Interactive Map with Folium

- Mark all launch sites on a map using folium.map.Marker
- Mark the success/failed launches for each site on the map – MarkerCluster()
- Calculate distance between two points using haversine distance
- Display the distance between two co-ordinates and mark the distance between them using a folium.polyline

Link to notebook – Check here

Build a Dashboard with Plotly Dash

- **Pie Chart**
  - Pie Chart showing the total launches by a certain site/all sites
  - Display relative proportions of multiple classes of data.
  - Drop Down provided to select individual launch sit
- **Scatter Plot**
  - Showing the relationship with Outcome and Payload Mass (Kg) for the different Booster Versions
  - A Slider is provided to alter payload

Predictive Analysis (Classification)

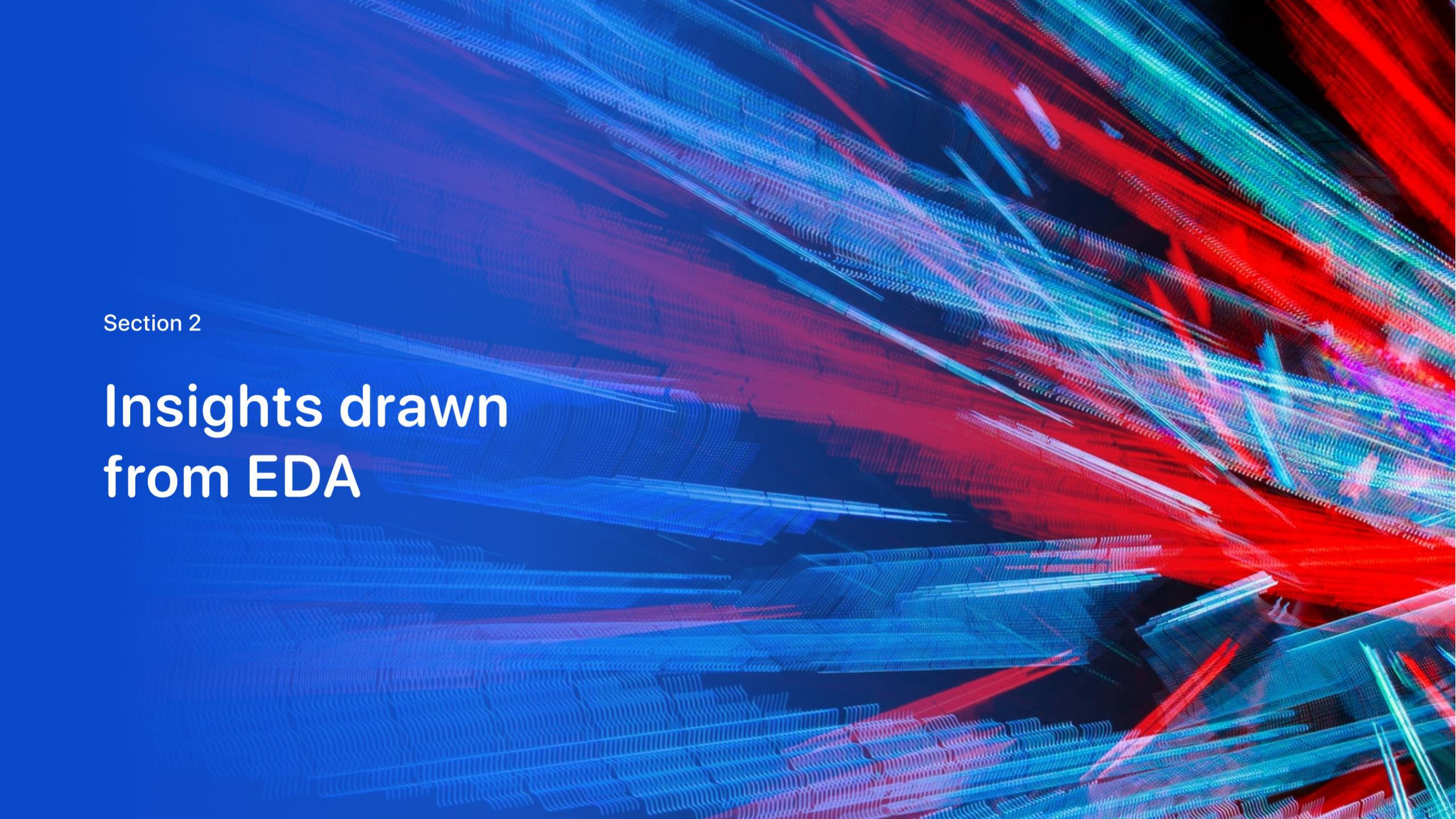
- **BUILDING MODEL**
  - Load our dataset into NumPy and Pandas
  - Transform Data and Split our data into training and test data sets
  - Train different ML classification models on training data
  - Hyper parameter optimization is done using GridSearchCV applied on each model
- **EVALUATING MODEL**
  - Prediction are made using the test data for each model
  - Accuracy score is calculated based on prediction and actual data of test data
  - Confusion matrix is plotted for each model
- The model with the best accuracy score is selected
- Link to notebook – Click here

Results

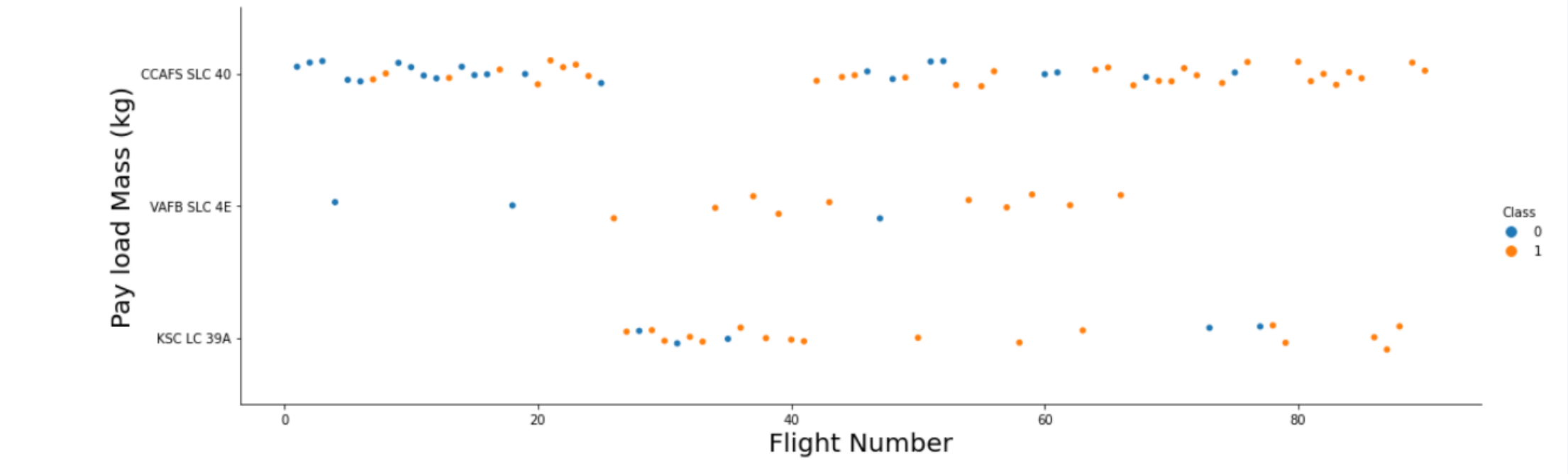
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

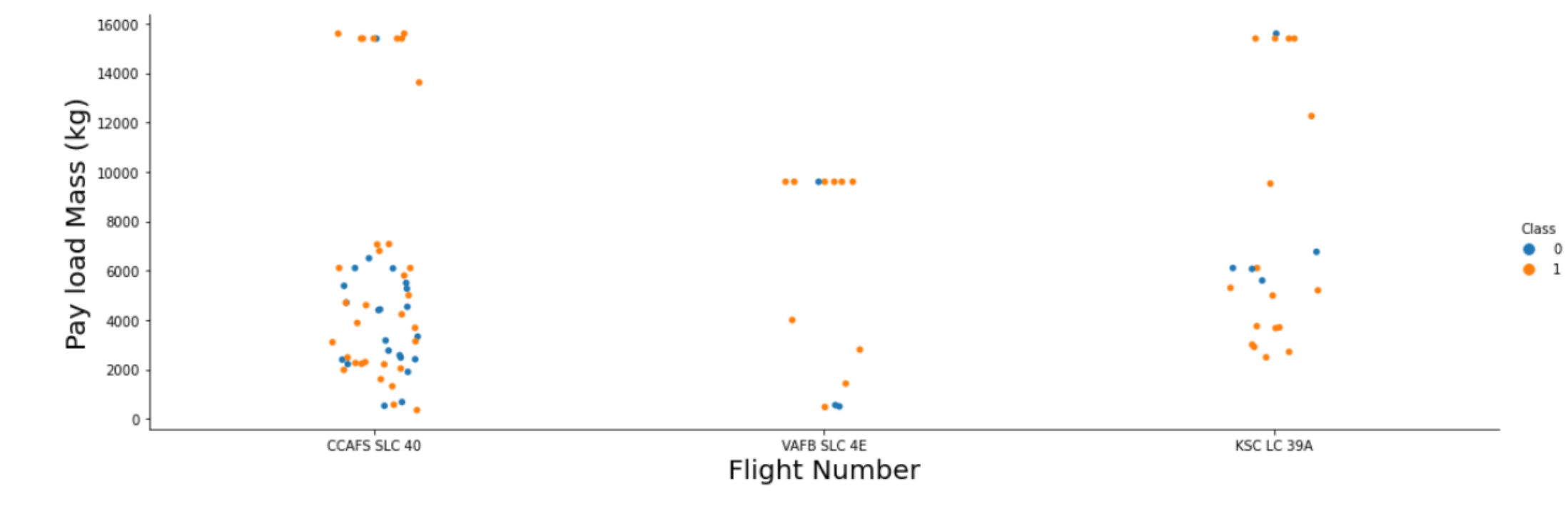


Flight Number vs. Launch Site



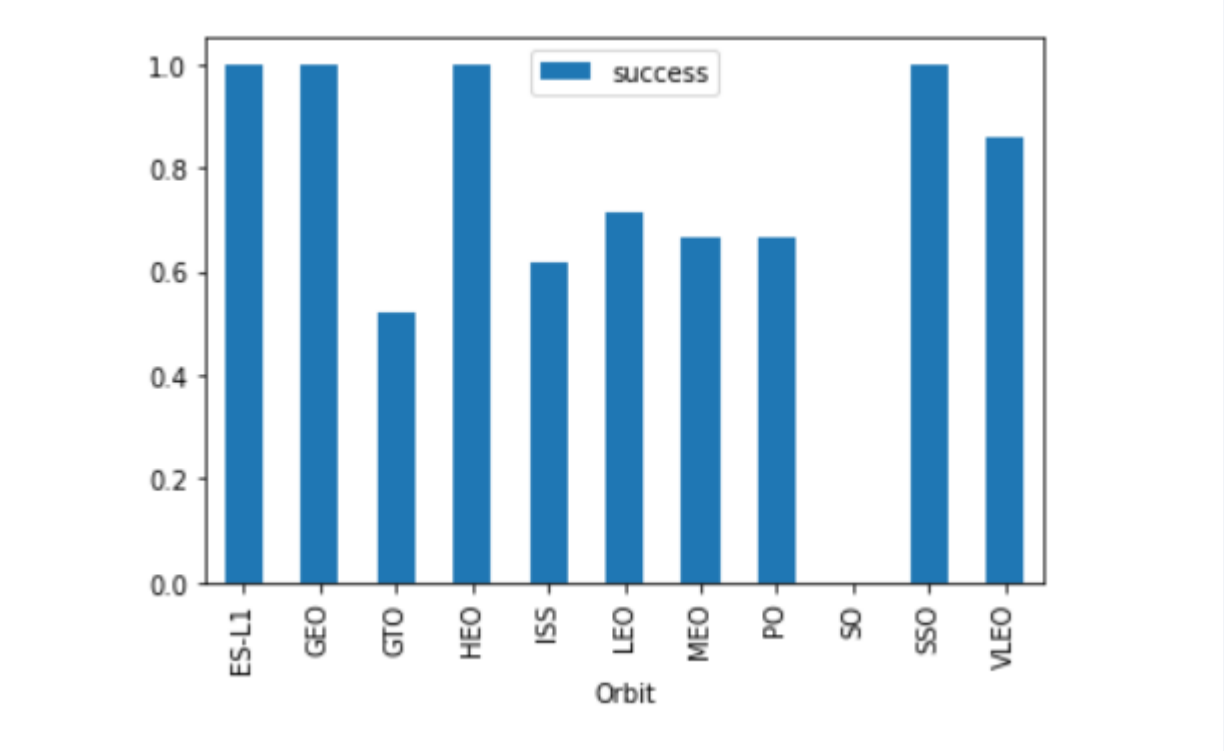
- As the Flight number increased the success rate increases for all launch sites

Payload vs. Launch Site



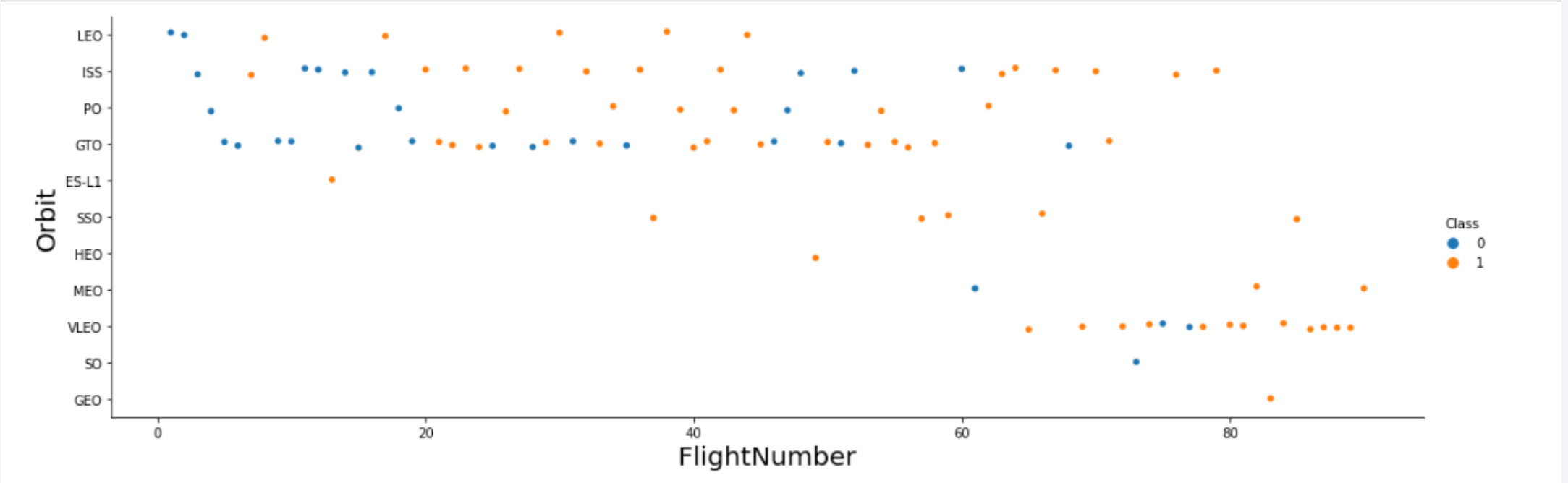
- Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).
- Heavy payloads also tend to have a higher success rate

Success Rate vs. Orbit Type



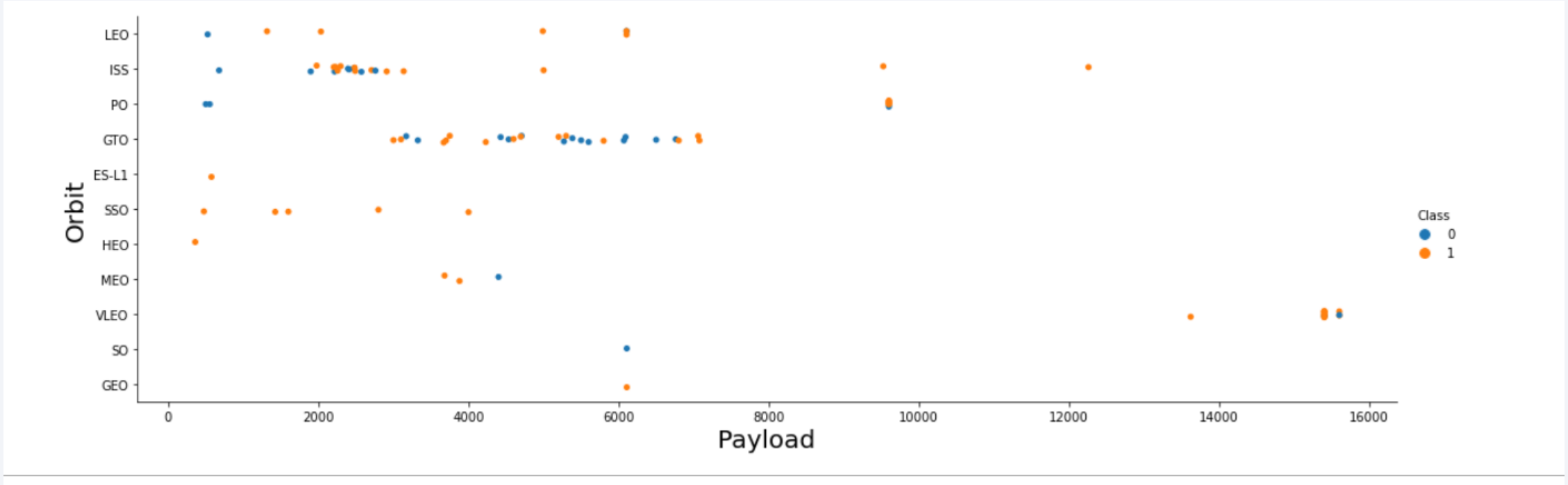
- Some orbits have 100% success rate

Flight Number vs. Orbit Type



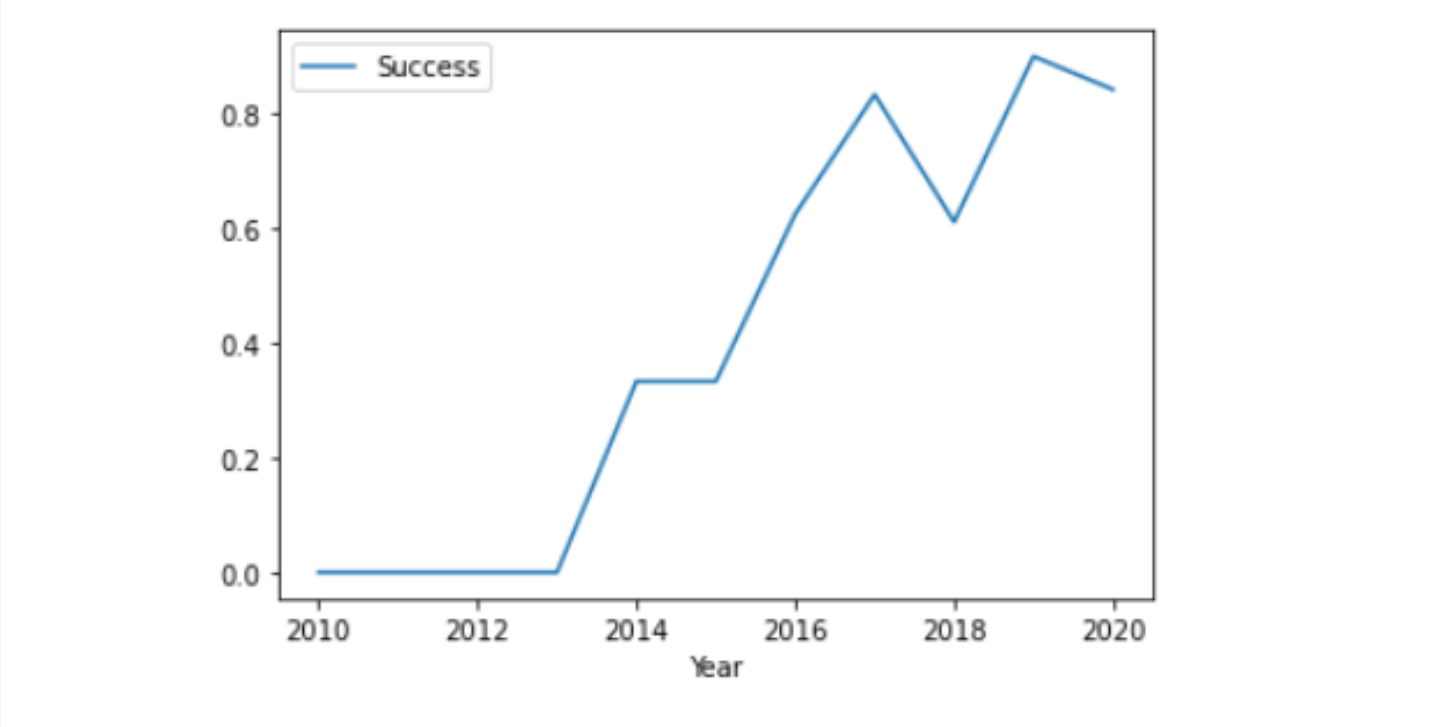
- You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



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

Launch Success Yearly Trend



- There is a increasing trend in the success rate with time

All Launch Site Names

- Find the names of the unique launch sites

In [7]:
```
%%sql
select distinct(launch_site) from SPACEXDATASET
```

```
 * ibm_db_sa://mml64344:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[7]:

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

```
In [33]: %%sql
select * from SPACEXDATASET where upper(launch_site) like '%CCA%' limit 5
```

```
 * ibm_db_sa://mml64344:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[33]:

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

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

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

In [41]:
```
%%sql
select customer,sum(payload_mass__kg_) as sum_of_payload from SPACEXDATASET group by customer having customer ='NASA (CRS)'
```

```
 * ibm_db_sa://mml64344:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[41]:

customer	sum_of_payload
NASA (CRS)	45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

In [46]:
```
%%sql
select booster_version,avg(payload_mass__kg_) as avg_payload from SPACEXDATASET group by booster_version having booster_version
= 'F9 v1.1'
```

```
 * ibm_db_sa://mml64344:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[46]:

booster_version	avg_payload
F9 v1.1	2928

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

In [53]:
```
%%sql
select MIN(DATE) as min_date,landing__outcome from SPACEXDATASET group by landing__outcome having landing__outcome='Success (gro
und pad)'
```

```
 * ibm_db_sa://mml64344:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[53]:

min_date	landing__outcome
2015-12-22	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

In [59]:
```
%%sql
select BOOSTER_VERSION,count(BOOSTER_VERSION) AS COUNT from SPACEXDATASET WHERE LANDING__OUTCOME='Success (drone ship)' and payl
oad_mass__kg_ between 4000 and 6000 group by BOOSTER_VERSION
```

```
 * ibm_db_sa://mml64344:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[59]:

booster_version	COUNT
F9 FT B1021.2	1
F9 FT B1031.2	1
F9 FT B1022	1
F9 FT B1026	1

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

In [91]:
```
%%sql
SELECT COUNT(MISSION_OUTCOME) as count,'Failure' as Status FROM SPACEXDATASET where mission_outcome like 'Failure%' union
SELECT COUNT(MISSION_OUTCOME) as count,'Success' as Status FROM SPACEXDATASET where mission_outcome like 'Success%'
```

```
 * ibm_db_sa://mml64344:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[91]:

COUNT	status
1	Failure
100	Success

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

```
In [14]: %%sql
SELECT DISTINCT(booster_version),payload_mass__kg_ FROM SPACEXDATASET WHERE payload_mass__kg_=(SELECT MAX(payload_mass__kg_) FROM SPACEXDATASET)
```

```
 * ibm_db_sa://mml64344:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[14]:

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

2015 Launch Records

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

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

In [15]:
```
%%sql
select DATE,landing__outcome,booster_version,launch_site from SPACEXDATASET WHERE DATE  BETWEEN '2015-01-01' AND '2015-12-31' AND landing__outcome = 'Failure (drone ship)'
```

```
 * ibm_db_sa://mml64344:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[15]:

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

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

- 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

In [16]:
```
%%sql
select landing__outcome,count(landing__outcome) as count from SPACEXDATASET WHERE DATE  BETWEEN '2015-06-04' AND '2017-03-20' group by landing__outcome order by COUNT desc
```

```
 * ibm_db_sa://mml64344:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

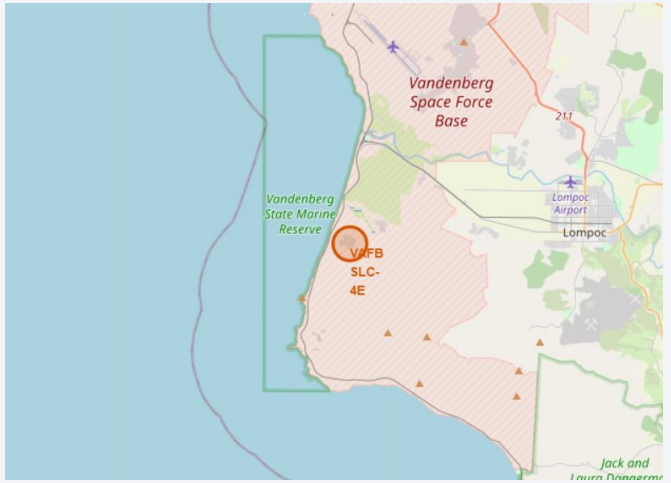
Out[16]:

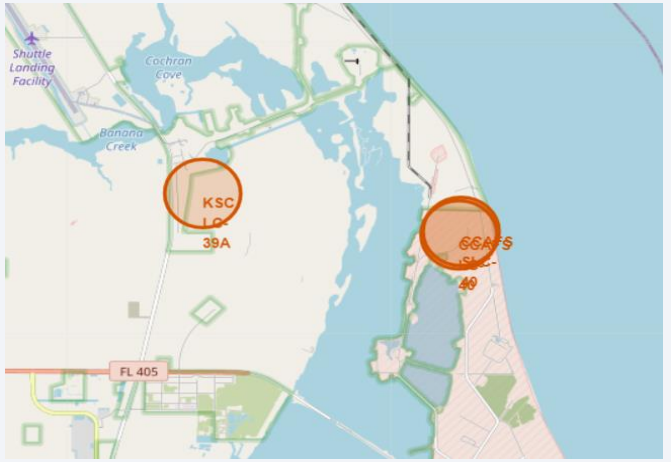
landing__outcome	COUNT
Success (drone ship)	5
Failure (drone ship)	3
Success (ground pad)	3
No attempt	1
Precluded (drone ship)	1

Section 4

Launch Sites Proximities Analysis

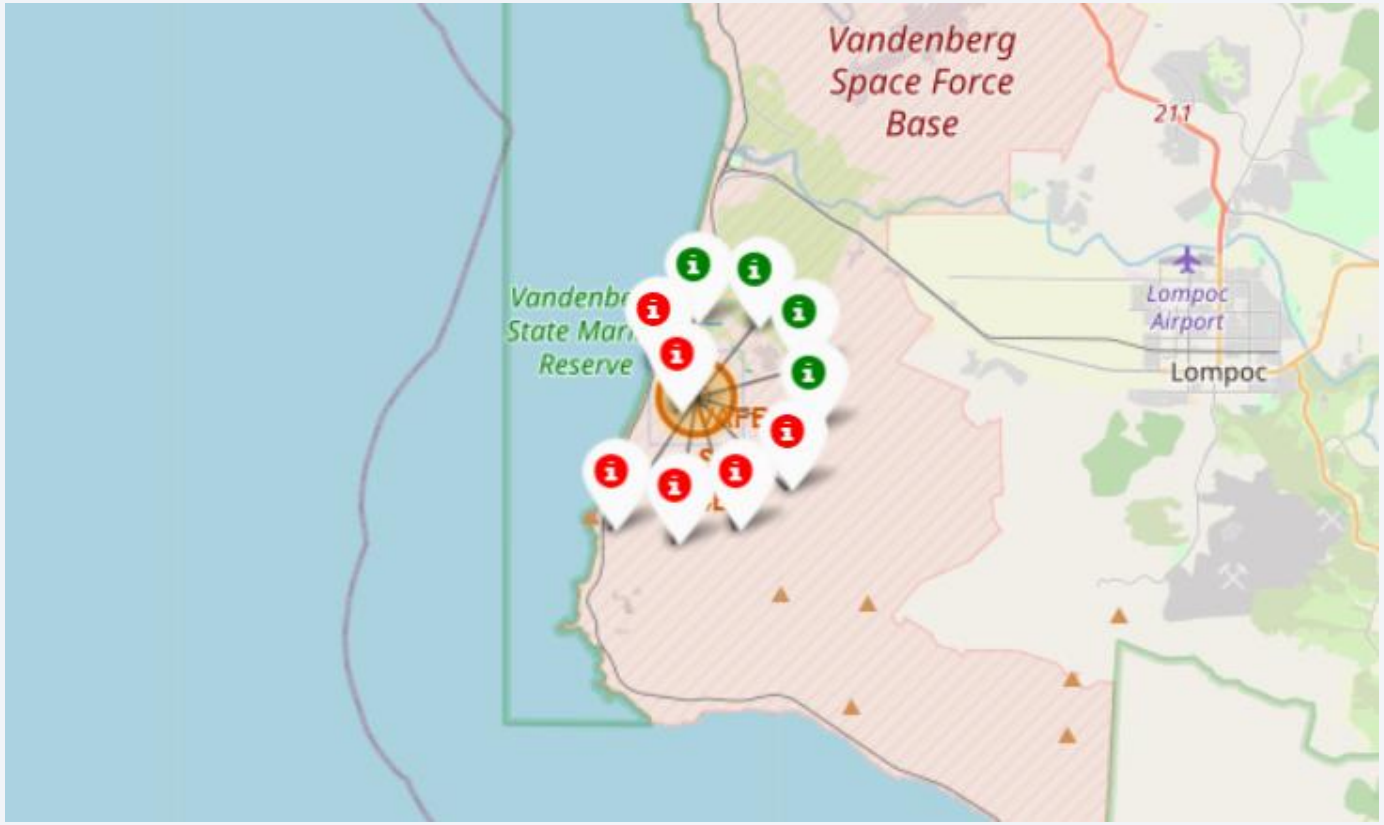
All launch sites



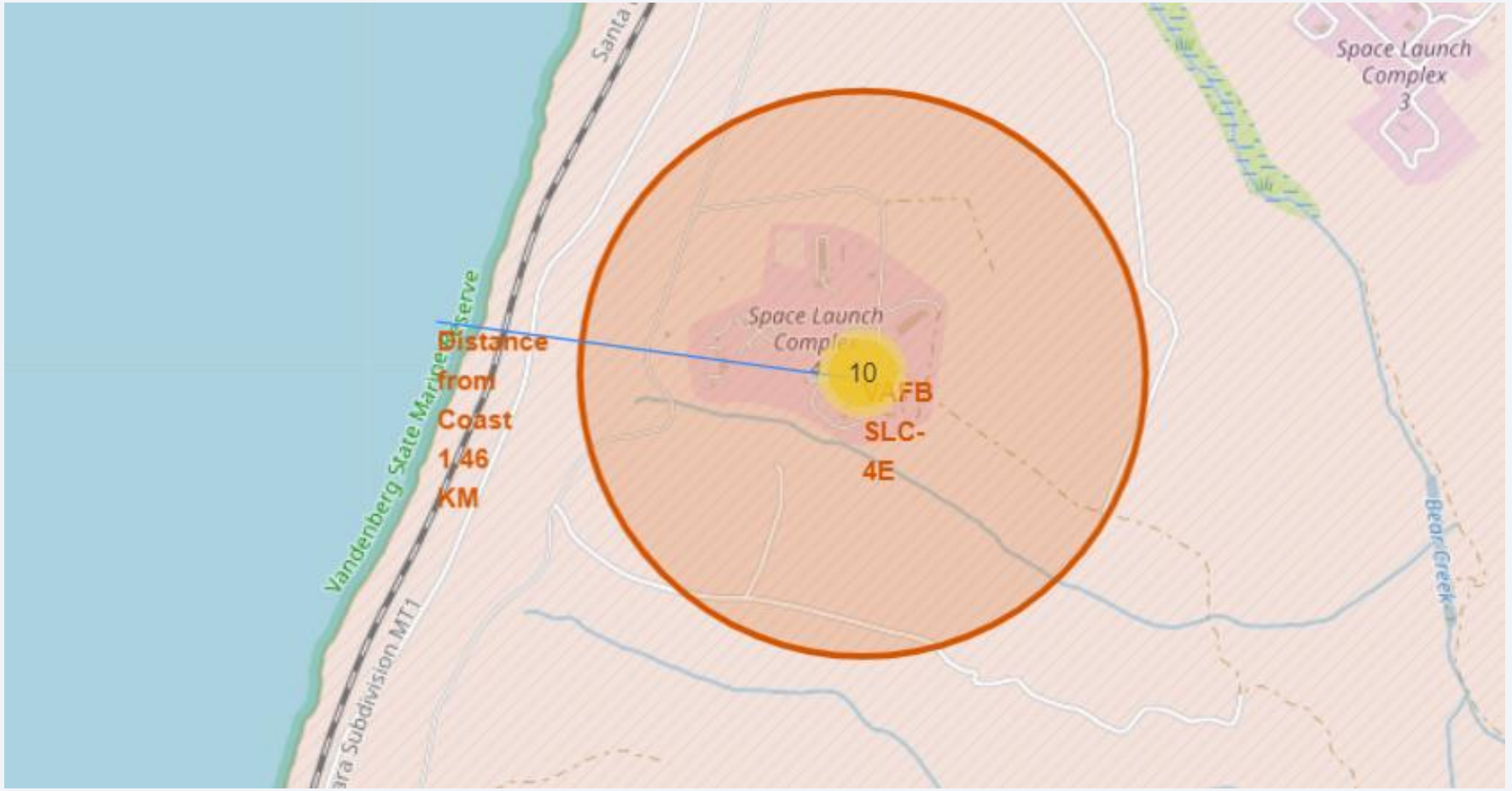


All launch sites are located near the coast

Outcome of launch

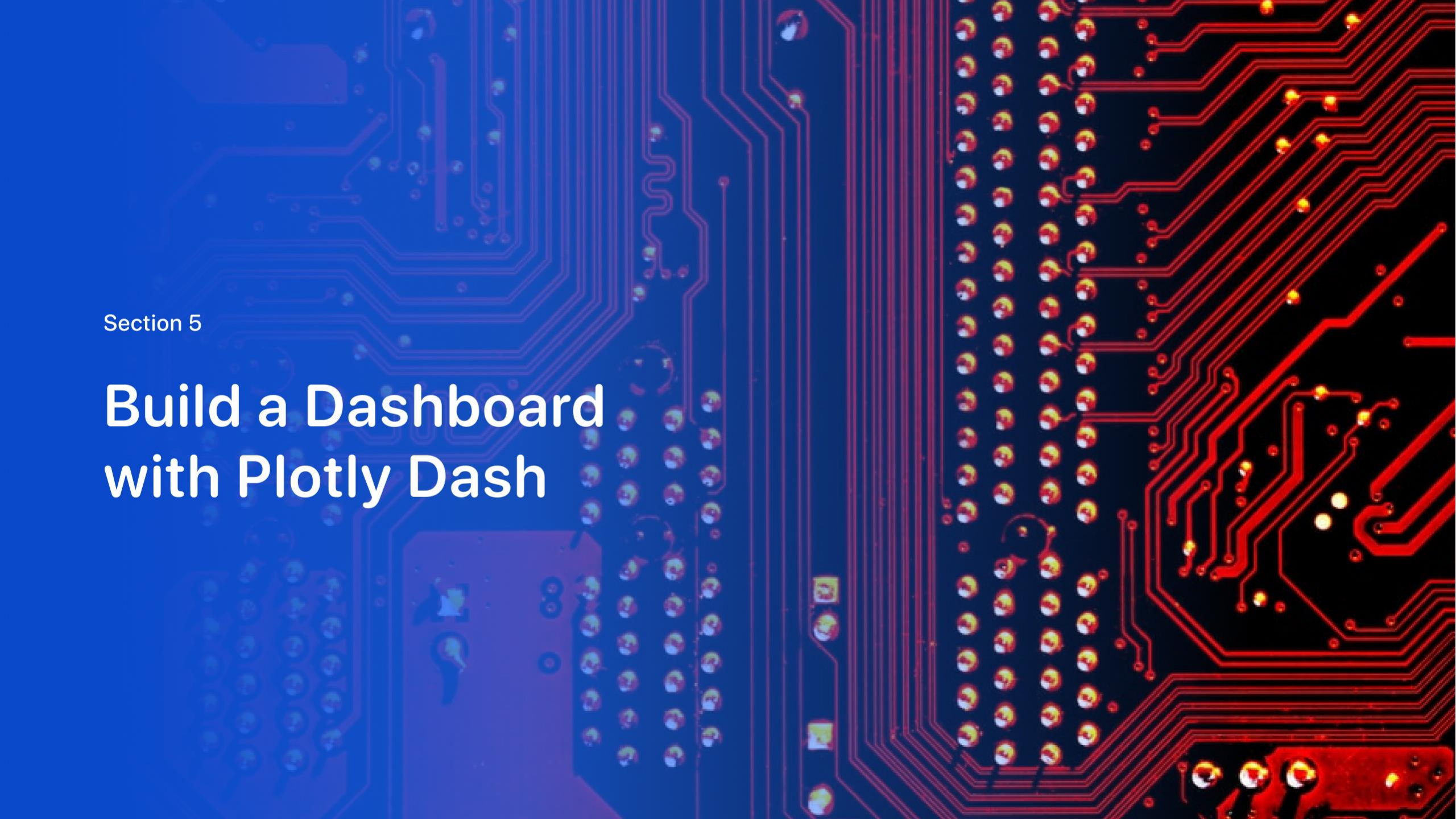


Proximity to coast

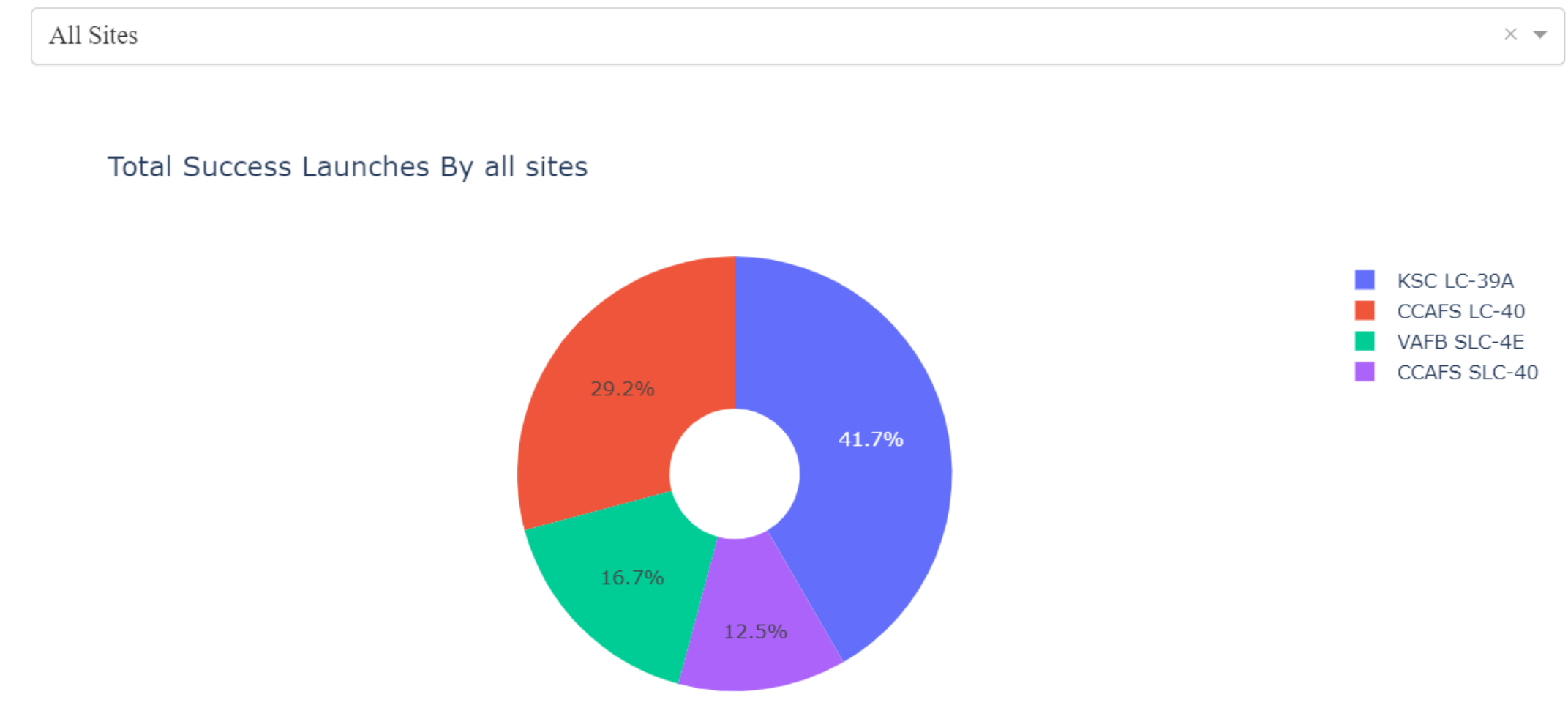


Section 5

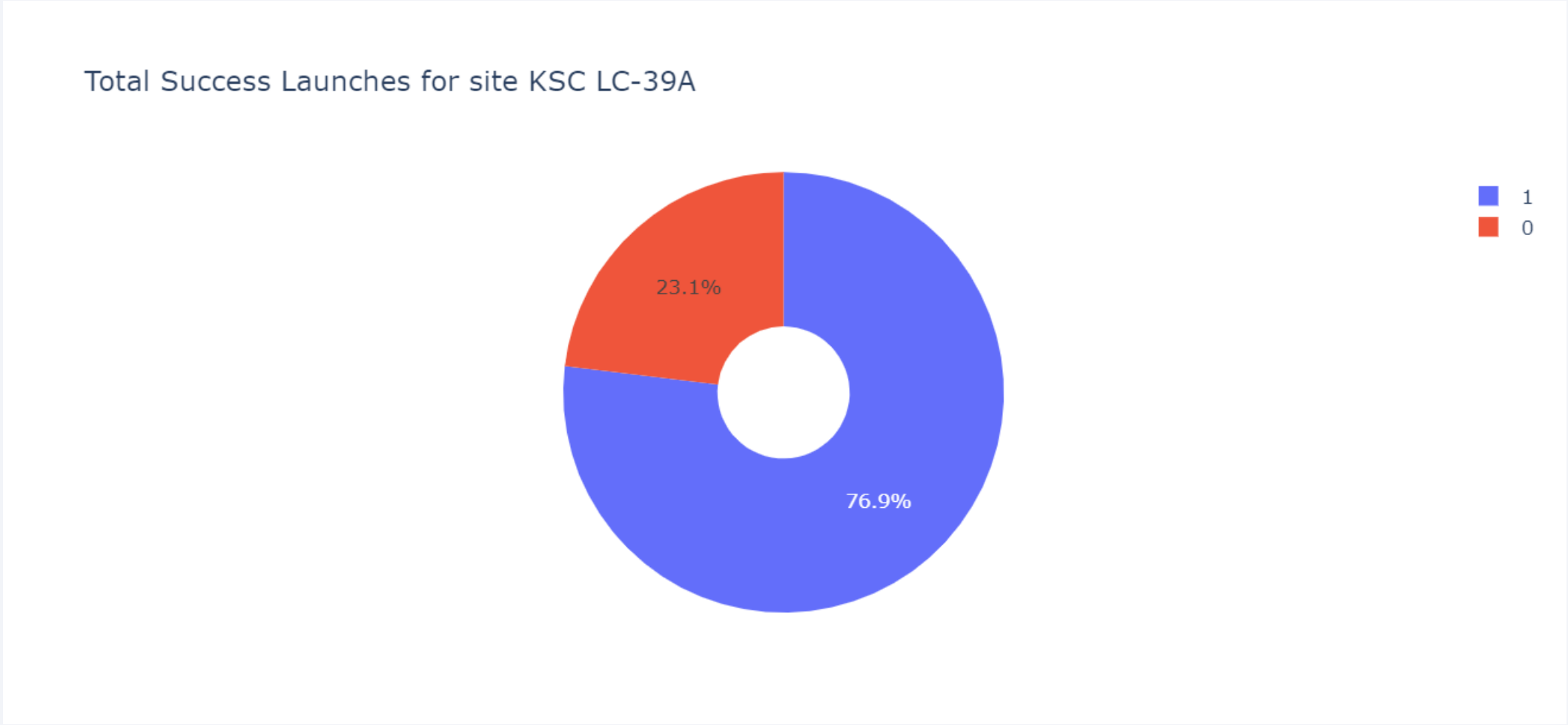
Build a Dashboard with Plotly Dash



Pie chart showing the success percentage achieved by each launch site



Pie chart for the launch site with highest launch success ratio



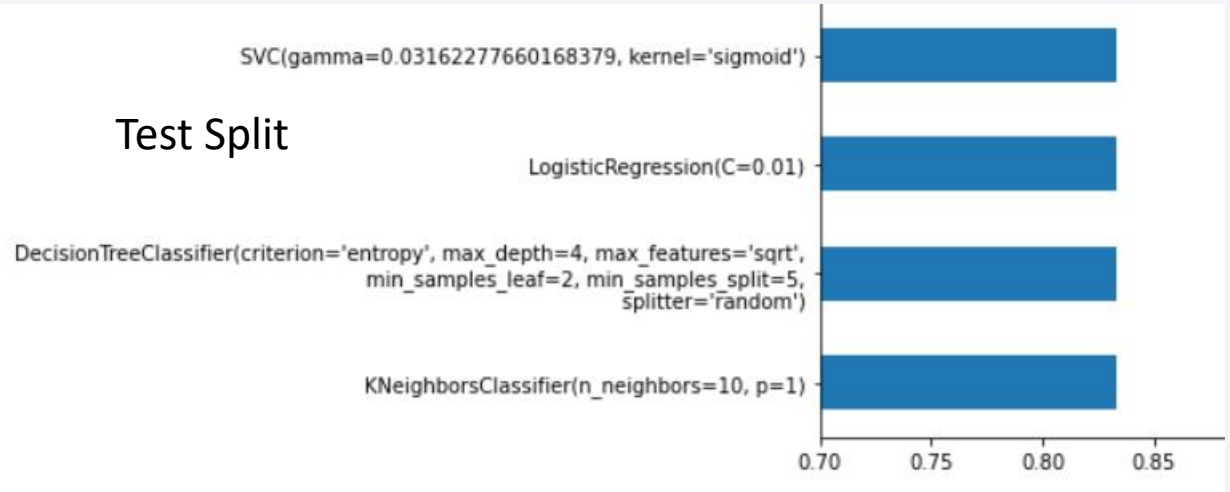
Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

Section 6

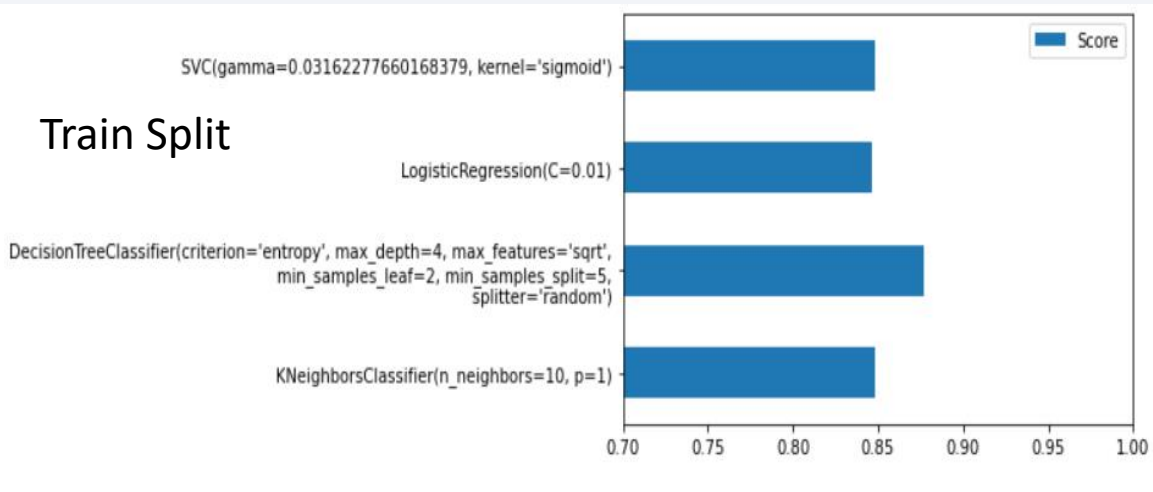
Predictive Analysis (Classification)

Classification Accuracy

Test Split



Train Split

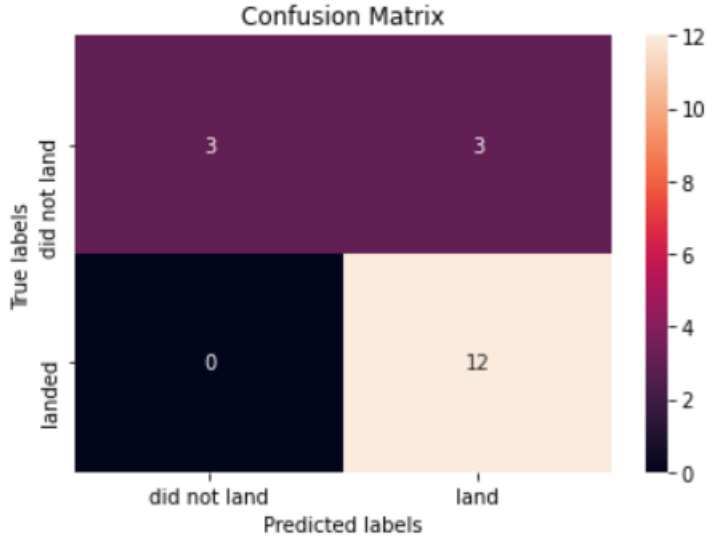


- All the models under consideration gave the same accuracy score on the testing dataset
- Decision tree was selected as the final model due to slightly better performance on the training data (Decision tress is prone to overfitting yet for the sake of selection we will go ahead with it)

Confusion Matrix

- Decision Tree confusion matrix

```
In [21]: yhat = tree_cv.predict(X_test)
         plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- The Tree Classifier Algorithm is the best for Machine Learning for this dataset
- The success rates for SpaceX launches increases proportionally with time suggesting that advances in technology and learning from past is key in increasing success rate
- We can see that KSC LC-39A had the most successful launches from all the sites
- Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate

Appendix

- Effect of Payload mass on success

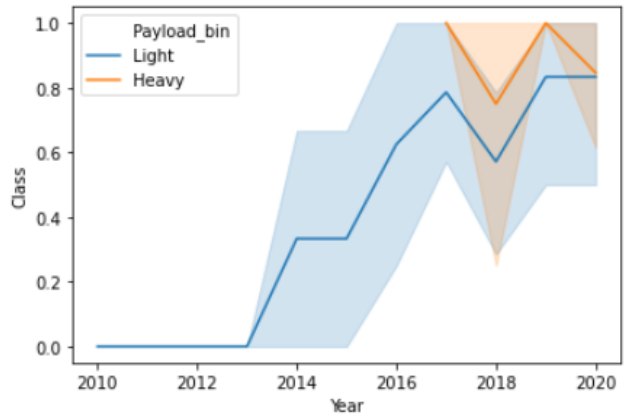
```
In [103]: df['Payload_bin']=pd.cut(x=df['PayloadMass'],bins=[0,7_000,df['PayloadMass'].max()],labels=['Light','Heavy'])
          df.groupby(['Payload_bin']).agg(MeanSuccess=('Class',np.mean),Count=('Class','count'))
```

Out[103]:

Payload_bin	MeanSuccess	Count
Light	0.584615	65
Heavy	0.880000	25

```
In [104]: sns.lineplot(x='Year',y='Class',hue='Payload_bin',data=df)
```

```
Out[104]: <matplotlib.axes._subplots.AxesSubplot at 0x7f8a1396edd0>
```



Payload when considered alone suggests that heavier payloads(>7000kg) perform better but,
When we combine payload and year of launch the trend is not as convincing as higher payload launches started only on 2017 where the success rate of even lighter payload is almost the same during that time period .

Thank you!