Winning Space Race with Data Science

Okechukwu C. Kene

2021-10-17

Outline

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

Executive Summary

- **Summary of methodologies**
- Space race of recent is fast becoming a commercial traveling with SpaceX dominating the space and working to make it less expensive. If the first stage of the rocket launches can be used over and over, it will go long way in reducing the cost. Various machine learning methodologies were applied to predict whether the first stage of SpaceX rocket launch will land successful which will enable re-use and cost reduction.

- **Summary of all results**
- Results culled from various models depicts an over 90% pointing that models provide excellent predictions if landing successes will be achieved.

- Furthermore, data exploration shows that launches from specific sites are more likely to land successfully.

Introduction

- Project background and context
  - As much as it may not be cheap space travelling is fast becoming a commercial business.
  - Concerns are high as to how to make commercial space travel less expensive. To be able to achieve this, cost of travel should include the reduction in cost for rocket launches.
  - SpaceX Falcon 9 rocket launches has been relatively cheap costing about of $62 million; when compared to others costing between $165 million dollars to $200 million. SpaceX is able to provide much of the savings because SpaceX can reuse the first stage.
- Problems you want to find answers
  - This project aims to determine and predict at first stage, will land using existing information from previous launches. If we can determine if the first stage will land, we can determine the cost of a launch.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data collection was achieved using SpaceX REST API to retrieve data on landing type, number of flights, landing pads and so on.
  - Python Beautiful Soup library was applied to get data from Wikipedia.
  - Data collected were normalized and converted into a python data frame for further processing and exploration
- Perform data wrangling
  - Data collected were processed by replacing missing data such as PayloadMass with the average payload mass from the dataset

Methodology - cont..

- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - After the exploratory analysis of the data we standardized the data , splitted them in training and testing dataset for training and testing our machine learning models.

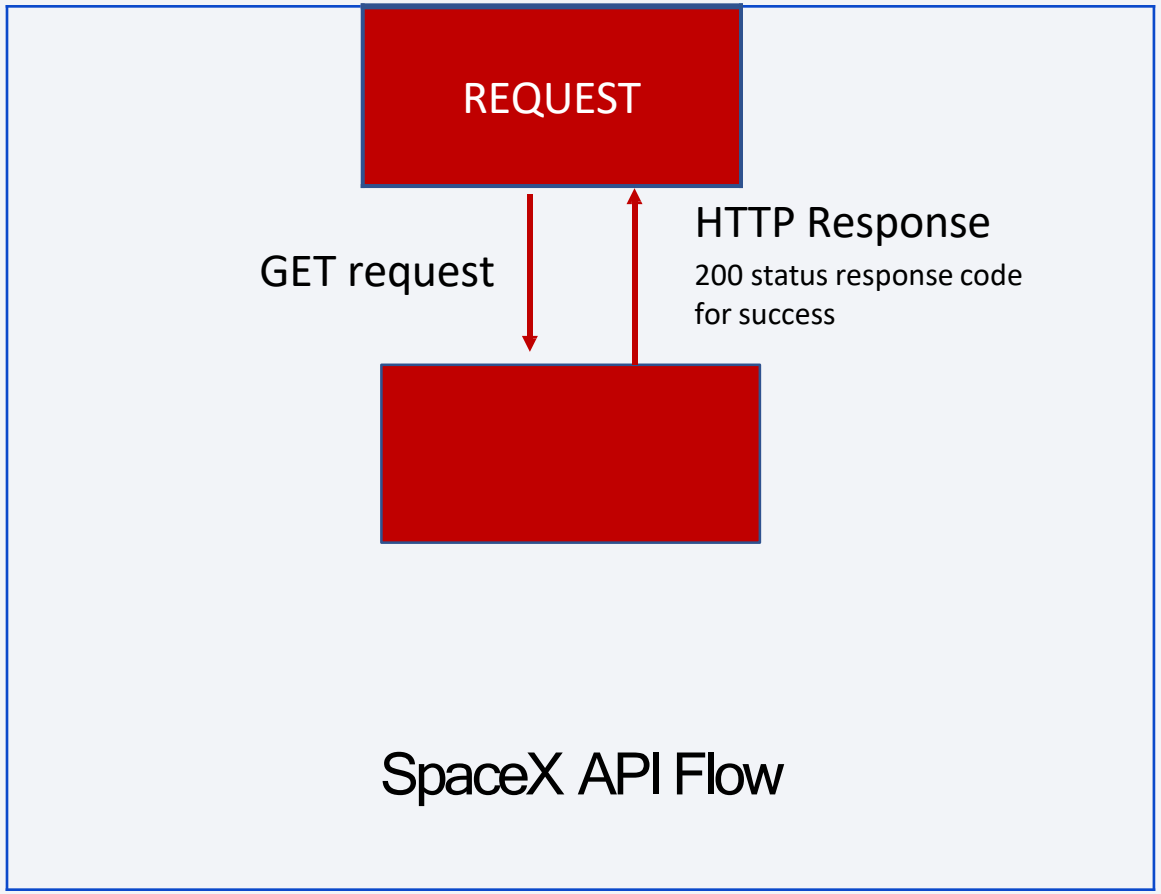
Data Collection

- Rocket Data was collected by making a GET request to SpaceX API to extract information using identification numbers in the launch data using the python request library.
- We then parse and decode the response content as a Json using .json() and turn it into a Pandas dataframe using .json_normalize()
- A series of helper functions was defined to extract extract the relevant columns such as rocket use, launchpads (launch site being used, the longitude, and the latitude.), payload mass to learn the mass of the payload and the orbit that it is going to.
- The data obtained from the different columns were then combined into a dictionary from which we create a Pandas dataframe.
- Falcon 9 historical launch records were web scraped from an HTML table on Wikipedia page titled "List of Falcon 9 and Falcon Heavy launches" to collect records, using the BeautifulSoup library

Data Collection - SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

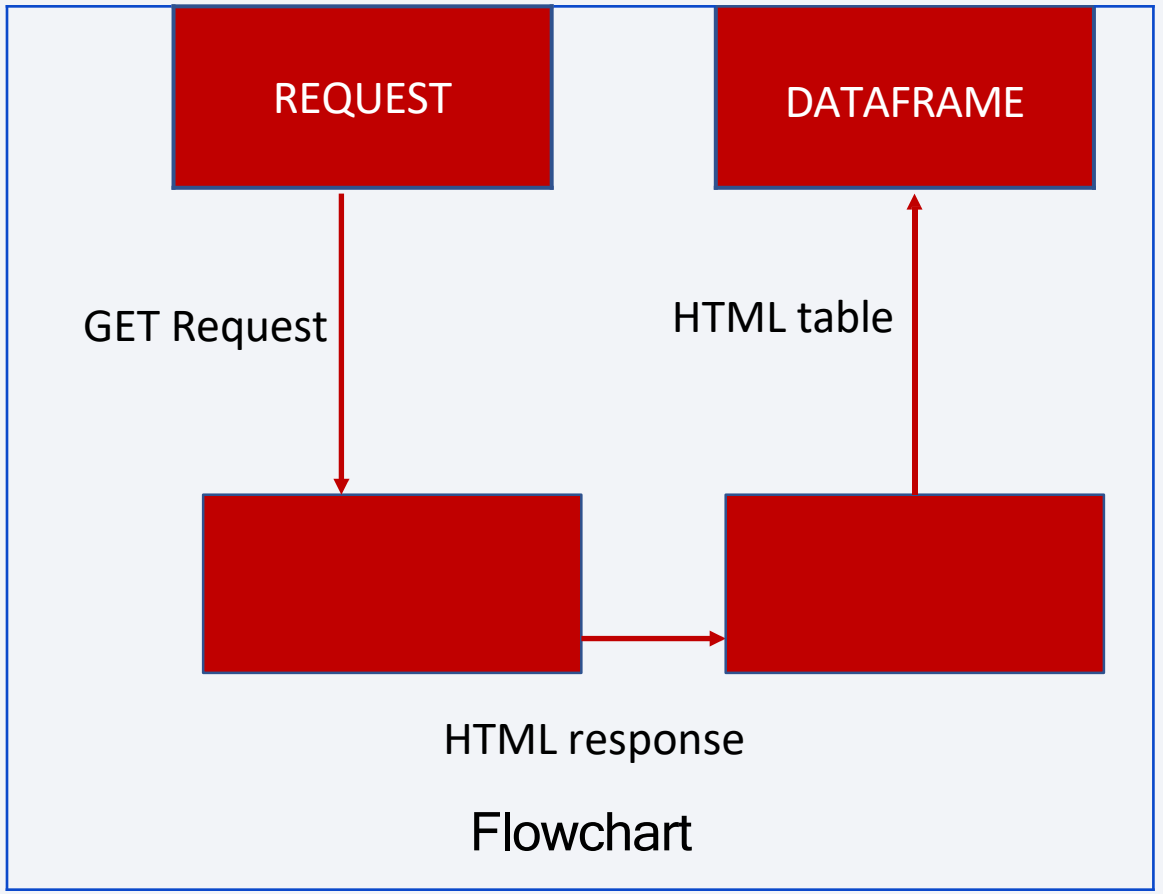
- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose



Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose



Data Wrangling

- Data Wrangling applied used to clean dataset by identifying and working on columns with null values.
- Percentage of missing values for each variable verified.
- Missing values from payload mass column replaced with mean value of the payload mass.
- Al relevant columns for the model should have no null values.
- The data set filtered included featured only the Falcon 9 launch which is of interest to building a model.

Data Wrangling - cont..

- As part of exploratory data analysis, libraries were imported and defined auxiliary functions for processing the dataset.
- Identifying the data types using dataframe dtypes() method.
- Value_counts() method used on the column LaunchSite to determine the number of launches on each site.
- Value_counts() method used on columns: orbit and output to determine the number and occurrence of each orbit, also, to determine the number of landing_outcomes respectively
- The value_count method on the outcome column assigned to a new variable landing_outcomes and used to create landing outcome label for out dataset

Data Wrangling - cont..

- With the output, a list created where the element is zero for the corresponding row in Outcome is in the set bad_outcome; otherwise, it's one. This is assigned to a variable, landing_class:
- This variable landing_class will represent the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully

EDA with Data Visualization

- Exploratory Data Analysis used on the dataset with various visualisations
- First is to visualize how the Flight Number which indicates the continuous launch attempts and Payload variables would affect the launch outcome using catplot from the seaborn library
- Furthermore, exploring the relationship between between 'Flight Number' and 'Launch Site' with scatter point plots using catplot and setting the hue to 'class'
- Different launch sites have different success rates. CCAFSLC-40, has a success rate of about 60 %, while KSCLC-39A and VAFB SLC4E has a success rate of 77%.

EDA with Data Visualization - cont..

- Relationship between launch sites and their payload mass was explored and observed that several successful launches were clustered around payload mass below 7000kg.
- Furthermore, the relationship between success rate of each orbit type. We observed ES-L1, GEO, HEO and SSO orbits show a high success rate
- Checks for each orbit were explored as to whether there is relationship between FlightNumber and Orbit type using scatter point plot. It was observed that in the LEO orbit, the Success were related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.
- Scatter Plot were used to explore the relationship between payload and orbit type where we observe that observe that Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.

EDA with Data Visualization - cont..

- In addition, the trends in the yearly launch success rate by plotingt a line chart with x axis to be Year and y axis to be average success rate, to get the average launch success trend.
- It'sobservedfrom the plot that the success rate has been increasing since 2013 till 2020

EDA with SQL

- To understand the SpaceX dataset, a DB2 database is created in Watson Studio.
- The CSV datasets were uploaded to the database schema creating tables for each CSV file.
- For the 'SPACEXDATASET' table, we updated the Date datatypes to the format DD-MM-YYYY
- The PAYLOAD*MASS*_KG_ datatype changed to INTEGER.
- Python libraries 'sqlachemy'', ibm_db_sa'', ipython-sql' were installed after which the SQL extension using the DB2 magic "%load_ext sql" were loaded.
- Connection to the database using the uri from the DB2 service credentials established.

Build an Interactive Map with Folium

- As part of the project an interactive Map with folium to visualize various launch sites is built.
- folium.Circle to add a highlighted circle area with a text label on a specific coordinate such as the NASA Johnson Space Center's used.
- Created and added folium.Circle and folium.Marker for each launch site on the site map to allow for easy location on the map
- All launch sites were found to be close to the coastline and far from places of dwelling
- Created a column with different colors - red for failed launch and green for success launch. We then created a marker_cluster which was added to folium.Marker on the map

- From the color-labeled markers in marker clusters, able to easily identify which launch sites have relatively high success rates.

Build an Interactive Map with Folium - cont..

- Added Mouse Position to get the coordinate (Latitude, Longitude) when the mouse is hovered over on the map. As such, when exploring the map, it can easily find the coordinates of any points of interests (such as railway)
- Using the Mouse Position, retrieved the coordinates for the closest coastline for each launch site
- Calculated the distance from each launch site to the nearest coastline and draw a PolyLine to the coastline, as proximities to areas such as railway, highway, residential areas or cities

Build a Dashboard with Plotly Dash

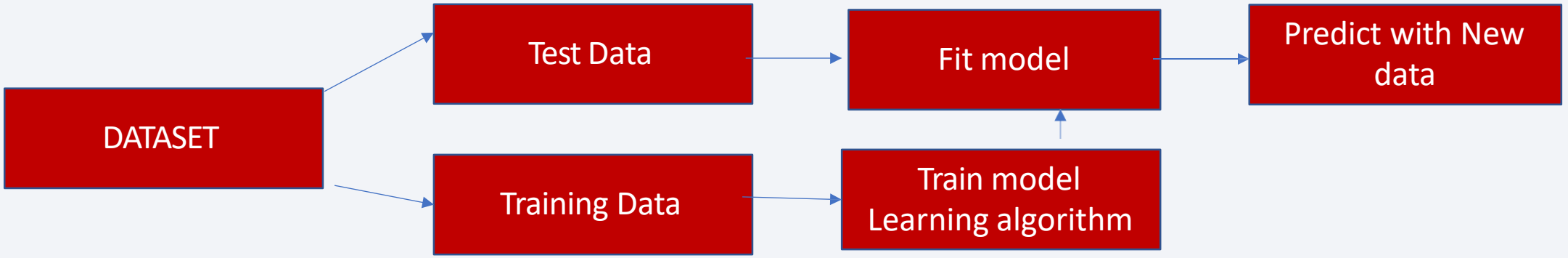
- A Plotly Dash application for users to perform interactive visual analytics on SpaceX launch data in real-time.
- A dropdown added to allow for filtering by site or selection all sites for interactive plots.
- Plotted a pie chart showing the total success launches by site
- Added a point scatter plot with a range slider for varying payload mass showing the correlation between payload mass and the success class by various Booster categories

Predictive Analysis (Classification)

- Added a column class for the outcome in our dataset to begin building our classification model. The class is 0 for failed launches and 1 for successful launches
- Loaded a dataset and select the Class column into a numpy array as our outcome 'Y' variable.
- Selected the features X from our dataset and standardize them by using the preprocessing.StandardScaler() object to transform them.[X=transform.fit_transform(X)]
- Data were split into training and testing using the function train_test_split. The training data is divided into validation data, a second set used for training data; then the models are trained and hyperparameters are selected using the function GridSearchCV.

Predictive Analysis (Classification) - cont..

- We later create various classification models (logistic regression, support vector machine, decision tree classifier, k nearest neighbors), using a cv=10 and the GridSearchCV function to determine the best hyperparameters for each model.
- For each algorithm, we evaluated the model based on the accuracy score , and output confusion matrix, and the classification report including the precision, recall f1 score and support values.
- We found the decision tree classifier to provide a higher accuracy score compared to the other models.

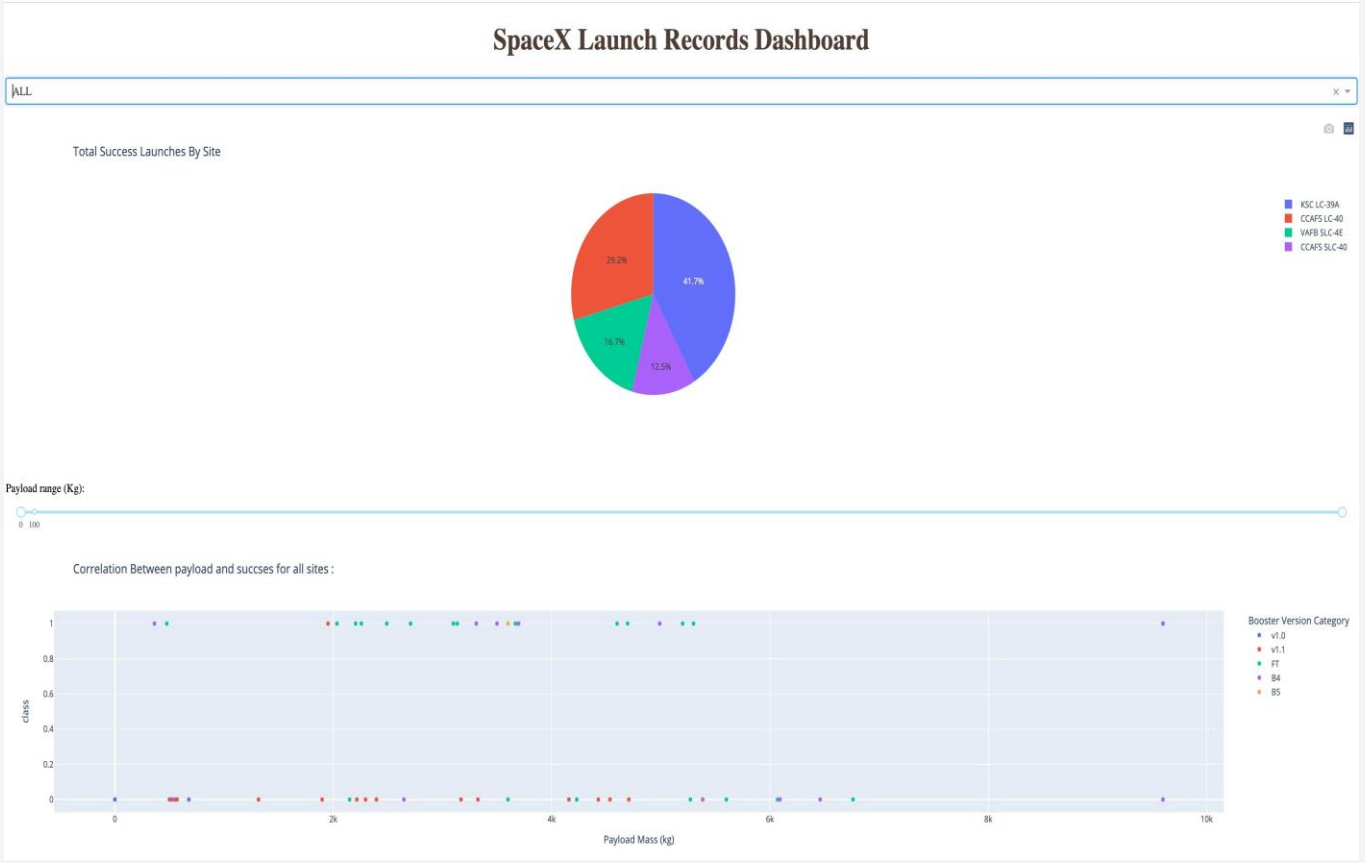


Results

- Different launch sites have different success rates. CCAFS LC-40, has a success rate of about 60 %, while KSCLC-39A and VAFB SLC 4E has a success rate of 77%.
- Observed that several successful launches were clustered around payload mass below 7000kg.
- The relationship between success rate of each orbit type. We observed ES-L1,GEO,HEO and SSOorbits show a high success rate visualized
- Further more, observed 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 GTOorbit.
- In addition, Hour exploration showed that heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.
- Found out that the success rate of launches has been increasing since 2013 till 2020

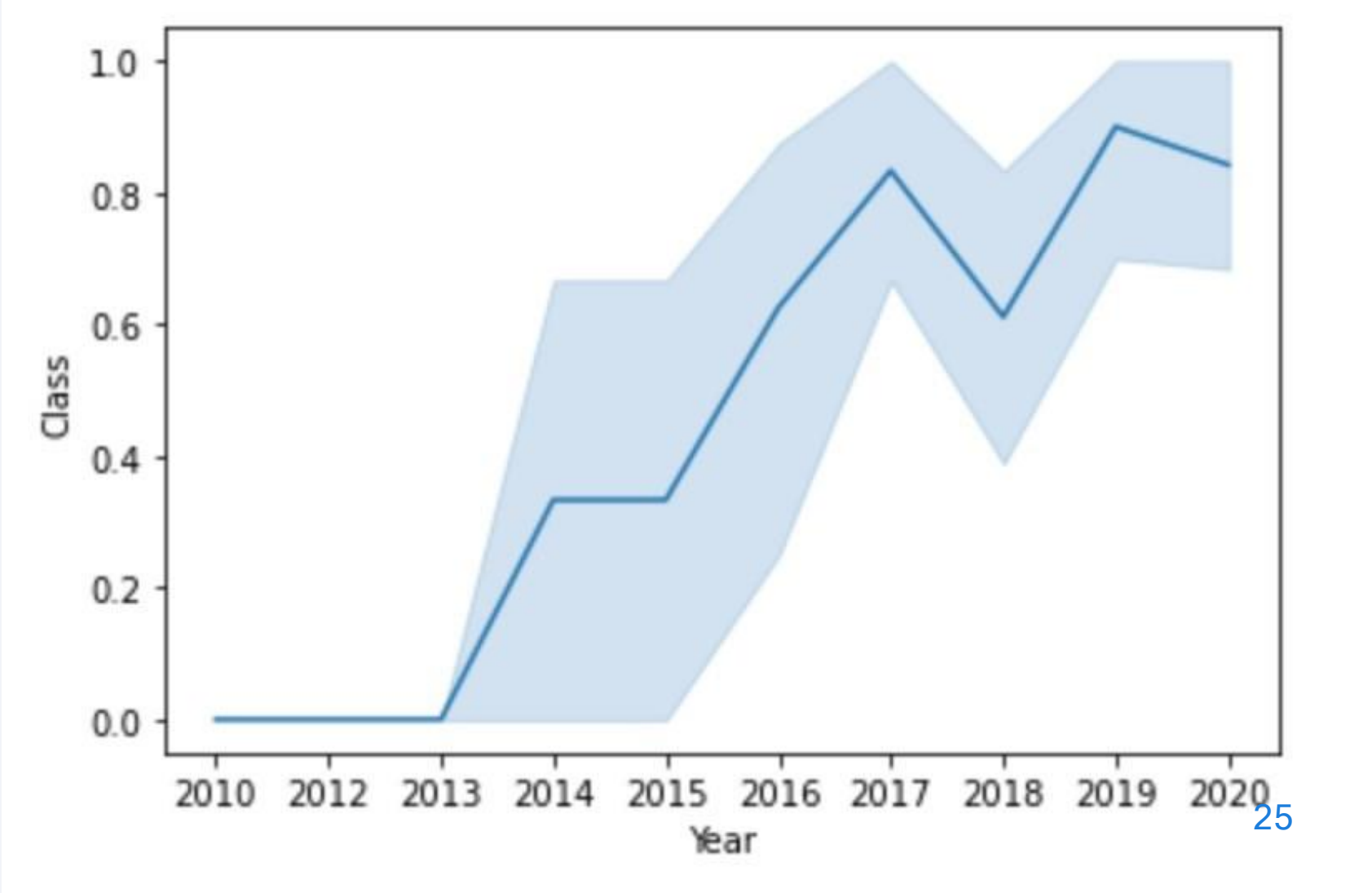
Results - cont ..

- Screenshots depicts interactive analytics



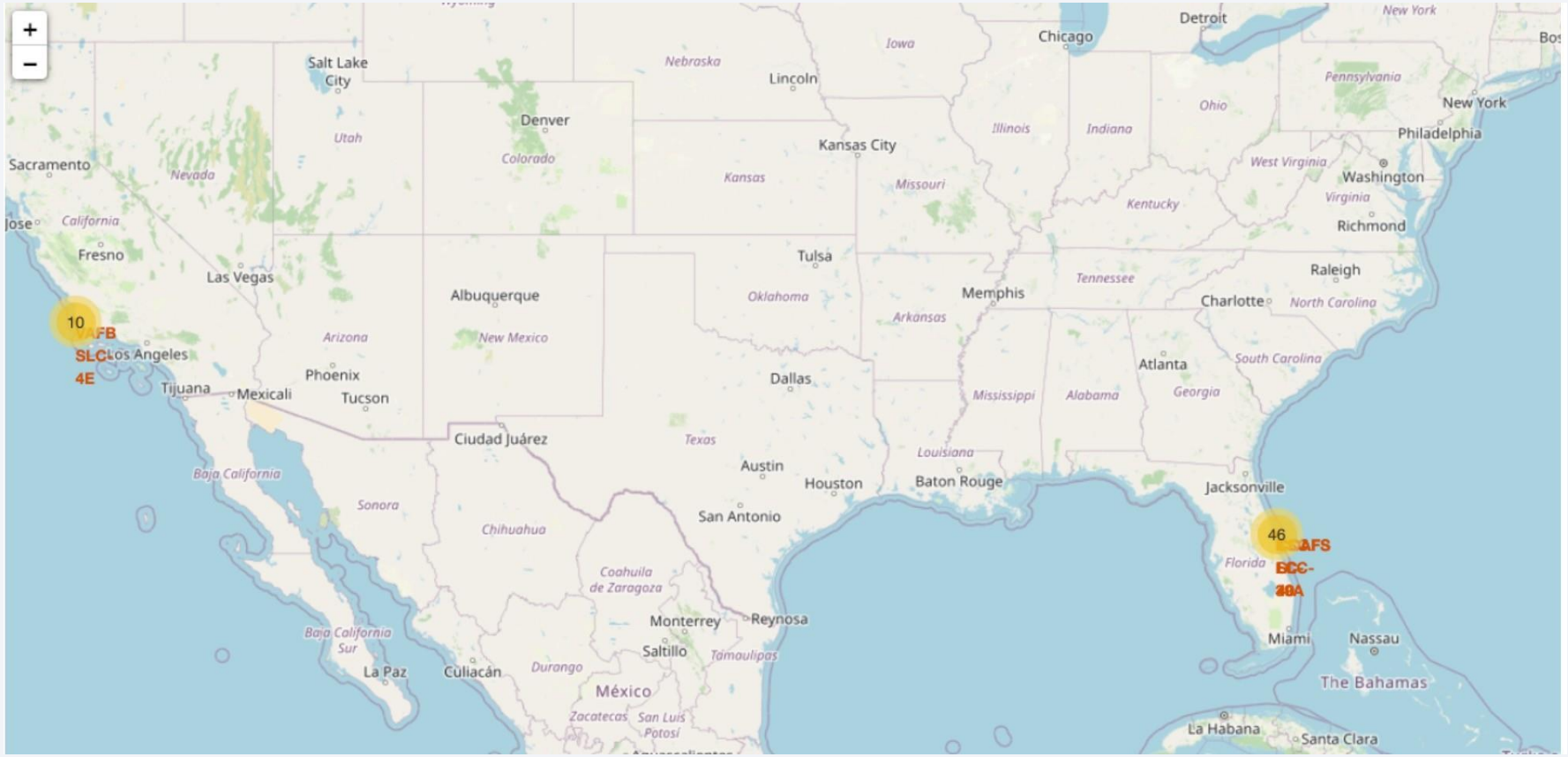
Results - cont ..

- Annual successful launch trend



Results - cont ..

- Geospatial location for launch site

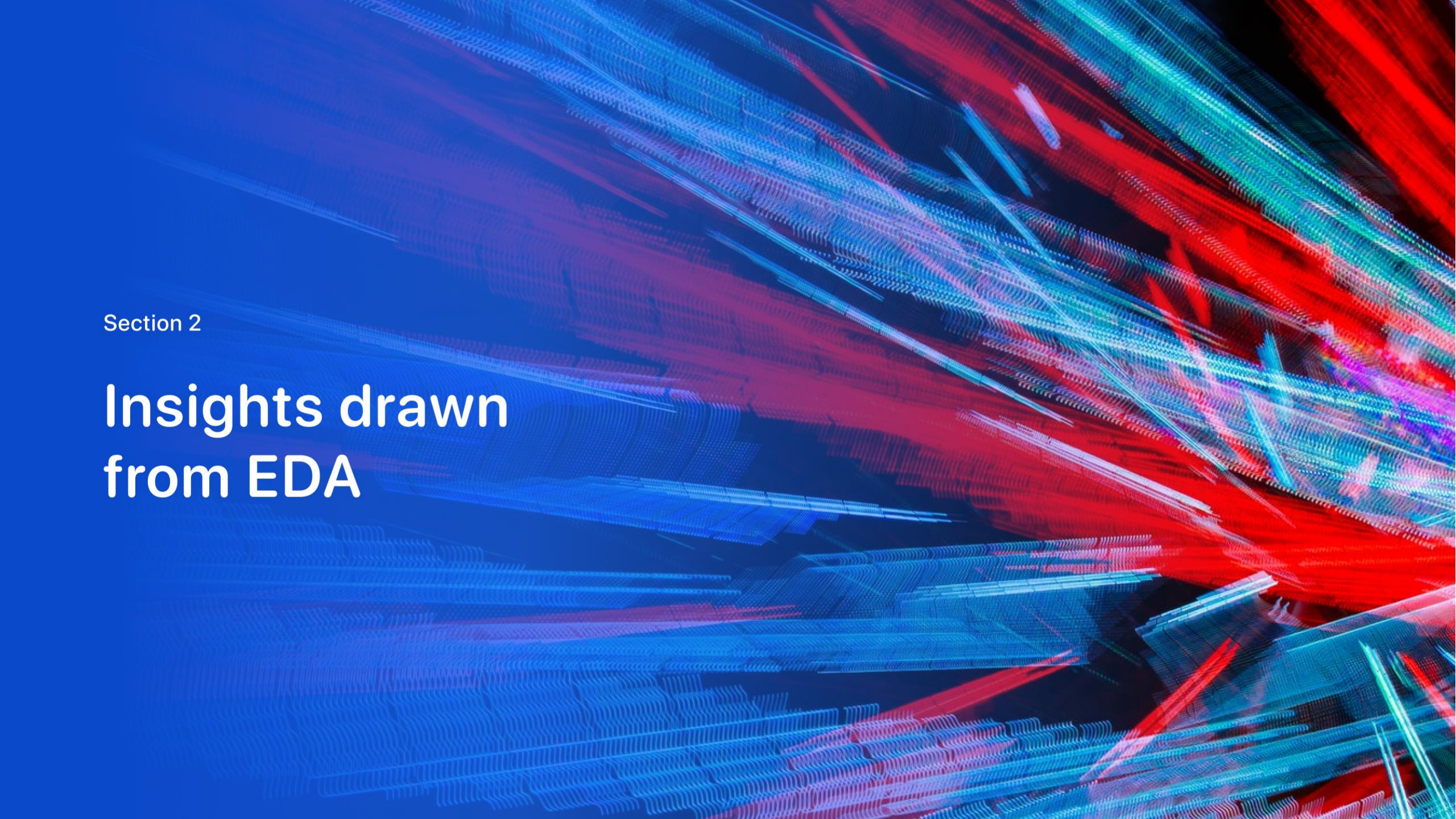


Results - cont ..

- From the confusion matrix on logistic regression model, it was observed that logistic regression can distinguish between the different classes. Clearly, the major problem is false positives. Logistic regression had an accuracy score of 83.33
- The best tuned kernel for SVM was the sigmoid kernel with a gamma value of 0.013. SVM resulted in a score of 83.33
- Decision tree resulted in the best score of 88.88 whereas KNN also had a score of 83.33

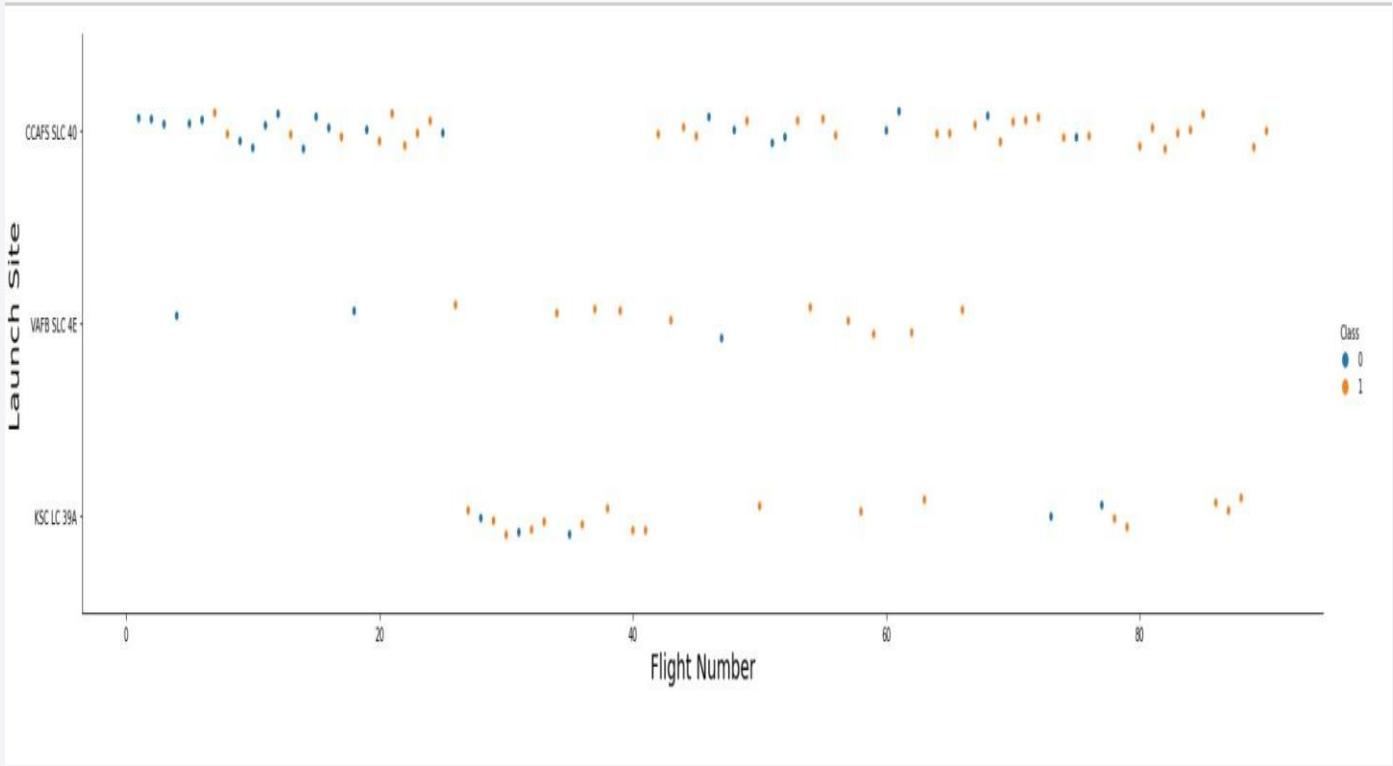
Section 2

Insights drawn from EDA



Flight Number vs. Launch Site

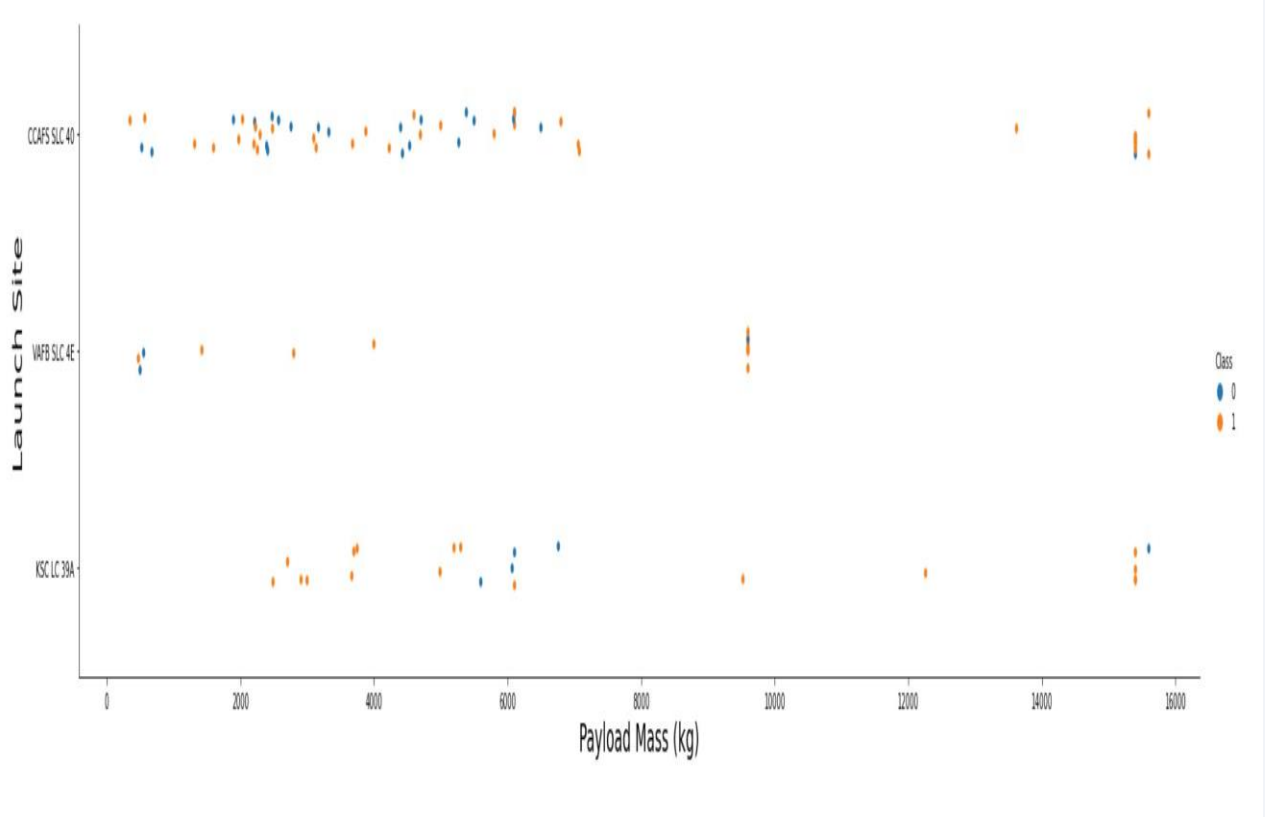
- Observed different launch sites have different success rates. CCAFS LC-40, has a success rate of 60 %, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.



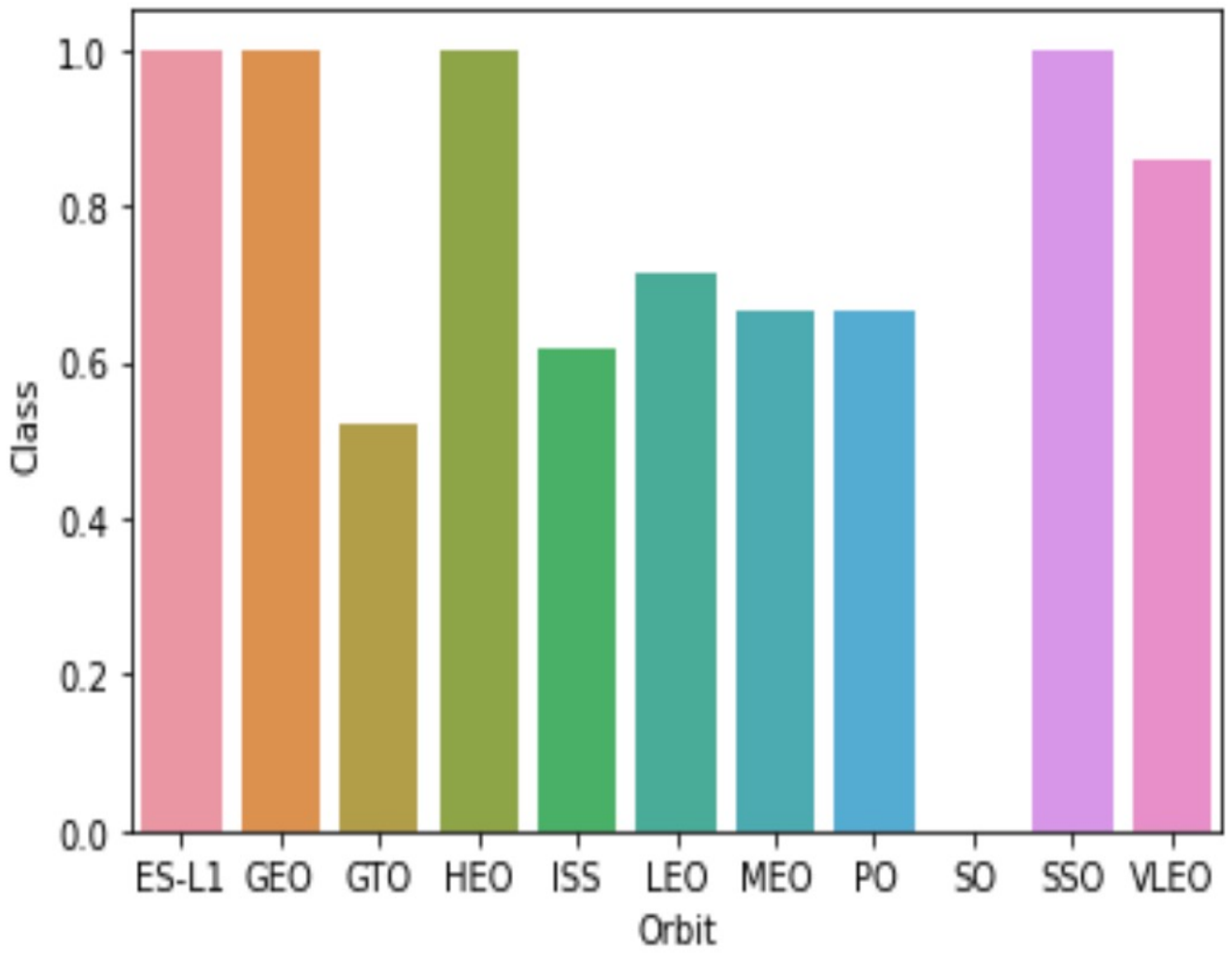
Payload vs. Launch Site

- From the plot Several flights are clustered around payload mass below 7000.
- Site KSC LC had a high proportion of success for

  payloads between 2000 and 6000
- Site CCAFS has about 83.33% success rate for payload mass above 8000 and about 55% for payloads below 8000 whereas VAFB has about 66% success for lower payloads below 8000 and about 75% for payloads above 8000

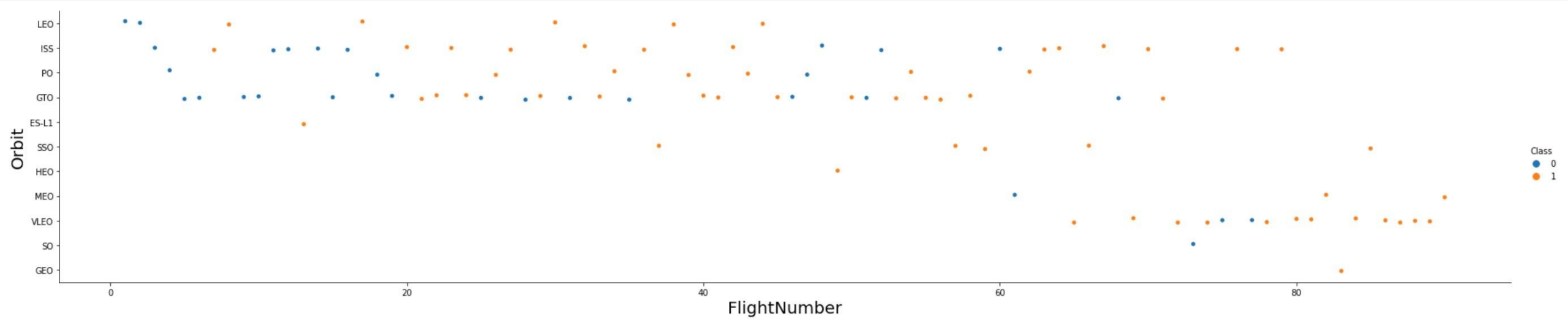


Success Rate vs. Orbit Type



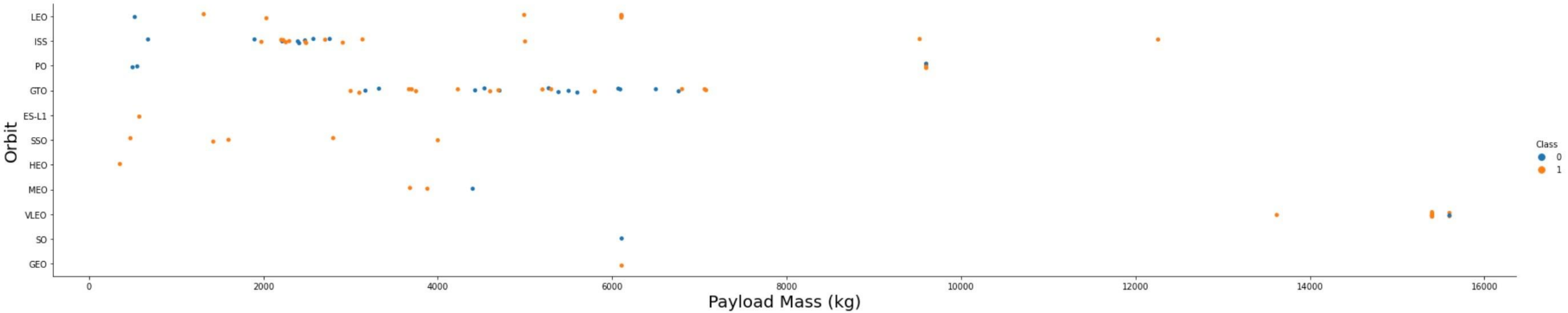
- The bar plot shows success rate on ES-L1, GEO, HEO and SSO orbit types

Flight Number vs. Orbit Type



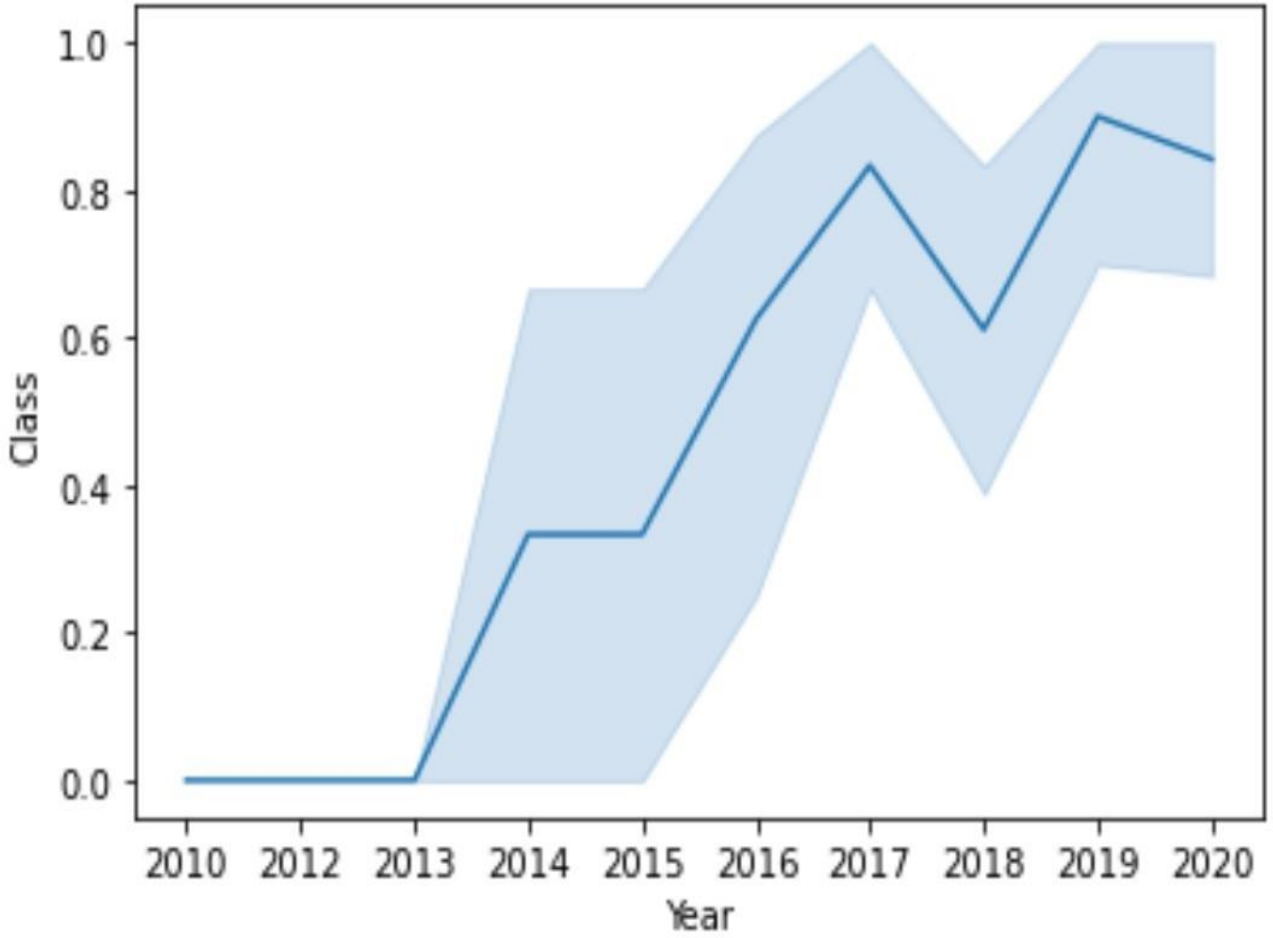
- For LEO orbit the success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type



- Heavy payloads have a negative influence on GTOorbits and positive on Polar LEO (ISS) orbits oberserved.

Launch Success Yearly Trend



- The graph show that from 2013, the success rate has generally been increasing annually to 2020

All Launch Site Names

```
%sql SELECT DISTINCT launch_site FROM SPACEXTBL
```

```
 * ibm_db_sa://tjy22290:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqblod8lcg.databases.appdomain.cloud:3119
8/bludb
Done.
```

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

- We use the DISTINCT launch site keyword to query the SPACEXTBL table to retrieve distinct sites within our dataset

Launch Site Names Begin with 'CCA'

```
%sql SELECT * FROM SPACEXTBL WHERE launch_site like 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://tjy22290:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:3119
8/bludb
Done.
```

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

- We used the LIMIT 5 keyword to limit the query result to 5 to retrieve 5 records from the table

Total Payload Mass

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) as PAYLOAD_BY_NASA_CRS FROM SPACEXTBL WHERE customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://tjy22290:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31198/bludb
Done.
```

payload_by_nasa_crs
45596

- The total payload by using the SUM(payload_mass) command is calculated.

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as AVG_PATLOAD_BY_F9 FROM SPACEXTBL WHERE booster_version LIKE 'F9 v1.1%'
```

```
 * ibm_db_sa://tjy22290:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:3119
8/bludb
Done.
```

avg_patload_by_f9
2534

- To calculate the average payload mass carried by booster version F9 v1.1, we use the AVG function on the payload mass column after which we limit the average to booster version F9 v1 from the where clause

First Successful Ground Landing Date

```
%sql SELECT MIN(DATE) as first_successful_date FROM SPACEXTBL WHERE landing__outcome = 'Success (ground pad)'
```

```
 * ibm_db_sa://tjy22290:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:3119
8/bludb
Done.
```

first_successful_date
2015-12-22

- Retrieved the the date when the first successful landing outcome in ground pad was achieved by using the min function on the DATE and limit to cases where landing_coutcome= 'Success (ground pad)'

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select DISTINCT booster_version  FROM SPACEXTBL  WHERE landing__outcome = 'Success (drone ship)' AND  PAYLOAD_MAS
S__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000
```

```
 * ibm_db_sa://tjy22290:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:3119
8/bludb
Done.
```

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

- To list the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000, we use DISTINCT to retrieve unique booster_version for successful landing landing__outcome = 'Success (drone ship)' AND PAYLOAD_MASS_KG_ > 4000 AND PAYLOAD_MASS_KG_ < 6000

Total Number of Successful and Failure Mission Outcomes

```
%sql select count(*) as total_number, mission_outcome  FROM SPACEXTBL  GROUP BY mission_outcome
```

```
 * ibm_db_sa://tjy22290:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:3119
8/bludb
Done.
```

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

- The total number of successful and failure mission outcomes by using the count keyword and grouping by outcome
- The result shows a significantly high number of success rate

Boosters Carried Maximum Payload

```
%sql SELECT DISTINCT booster_version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPAC
EXTBL)
 * ibm_db_sa://tjy22290:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:3119
8/bludb
Done.
```

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

- Listed the names of the booster which have carried the maximum payload mass by using a subquery to determine the max payload and selecting the record which has such value with the .

2015 Launch Records

```
%sql SELECT landing__outcome, booster_version,launch_site FROM SPACEXTBL WHERE YEAR(DATE) = 2015 AND landing__outcome
like  '%drone ship)'
```

```
 * ibm_db_sa://tjy22290:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:3119
8/bludb
Done.
```

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40

- Listed the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015 by using the where clause to limit YEAR = 2015 AND landing _outcome like '%drone ship)'

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

```
%sql SELECT COUNT(*),landing__outcome FROM SPACEXTBL WHERE DATE BETWEEN  DATE('2010-06-04') AND DATE('2017-03-20')  AN
D landing__outcome IN ('Failure (drone ship)','Success (ground pad)') GROUP BY  landing__outcome ORDER BY COUNT (*) DE
SC ;
```

```
 * ibm_db_sa://tjy22290:***@0c77d6f2-5da9-48a9-81f8-86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:3119
8/bludb
Done.
```

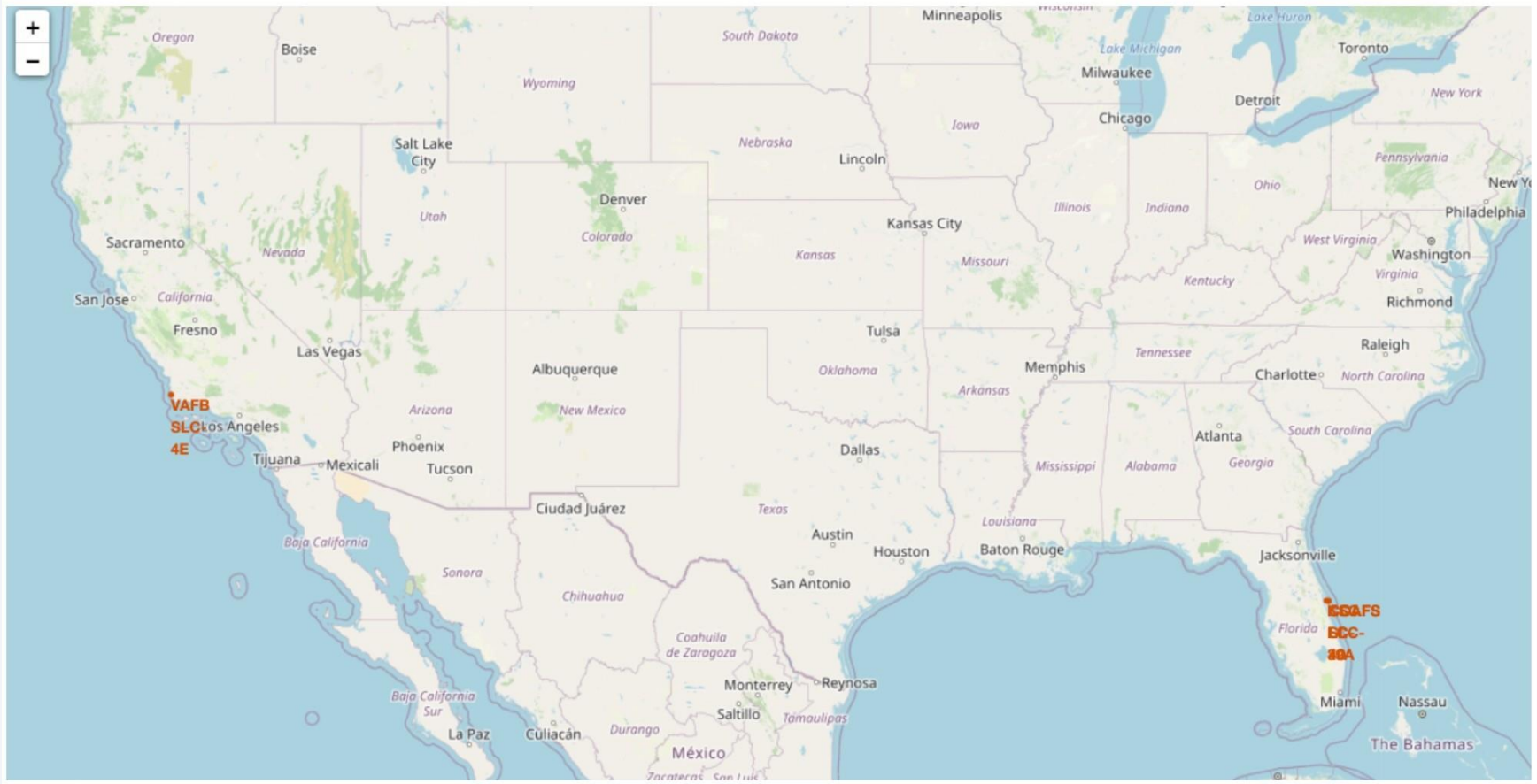
1	landing__outcome
5	Failure (drone ship)
3	Success (ground pad)

- Ranked the outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order with the ORDER BY DESCkeyword while limiting the reords for DATE between 2010-06-04 and 2017-03-20 with the where clause

Section 4

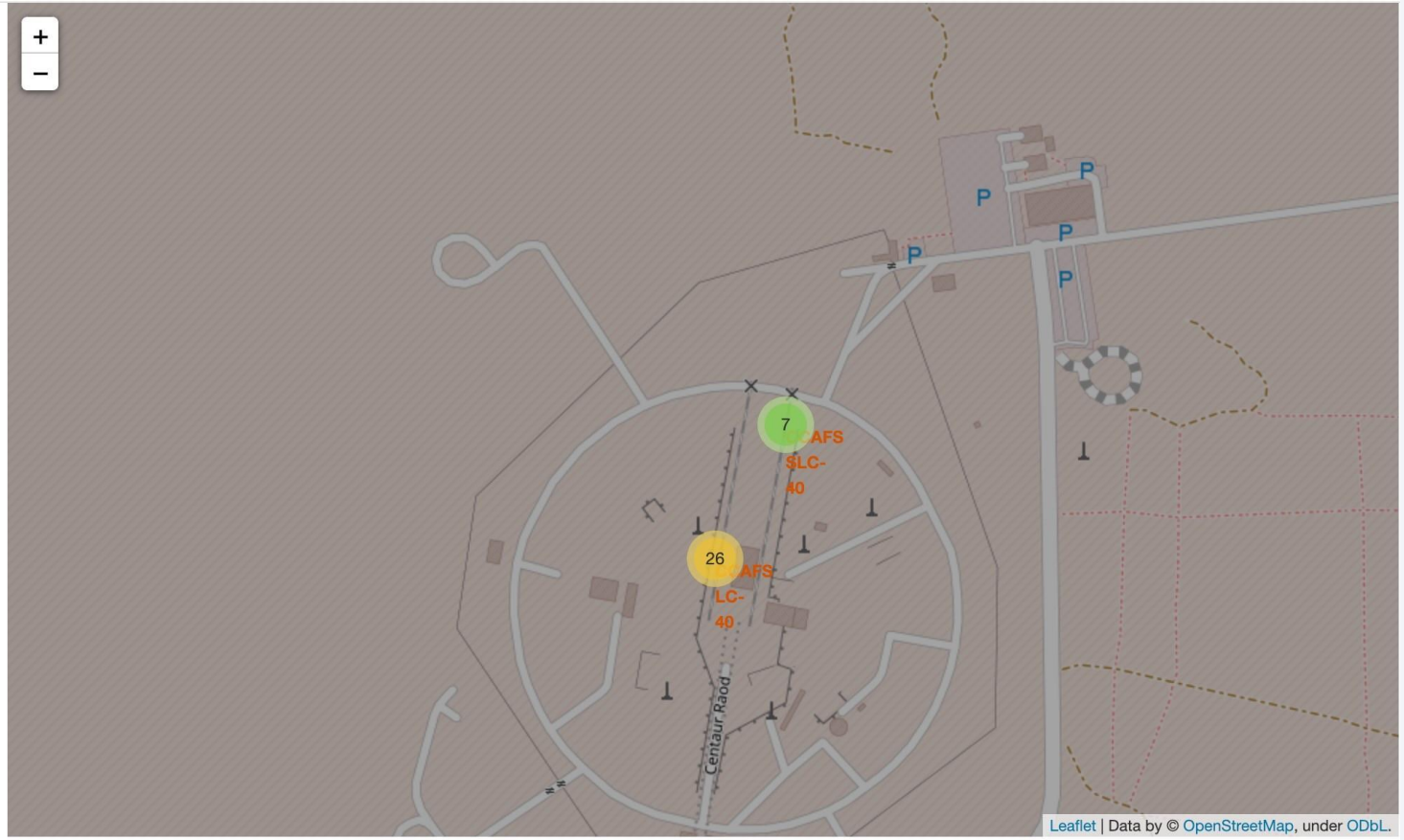
Launch Sites Proximities Analysis

Map showing launch sites



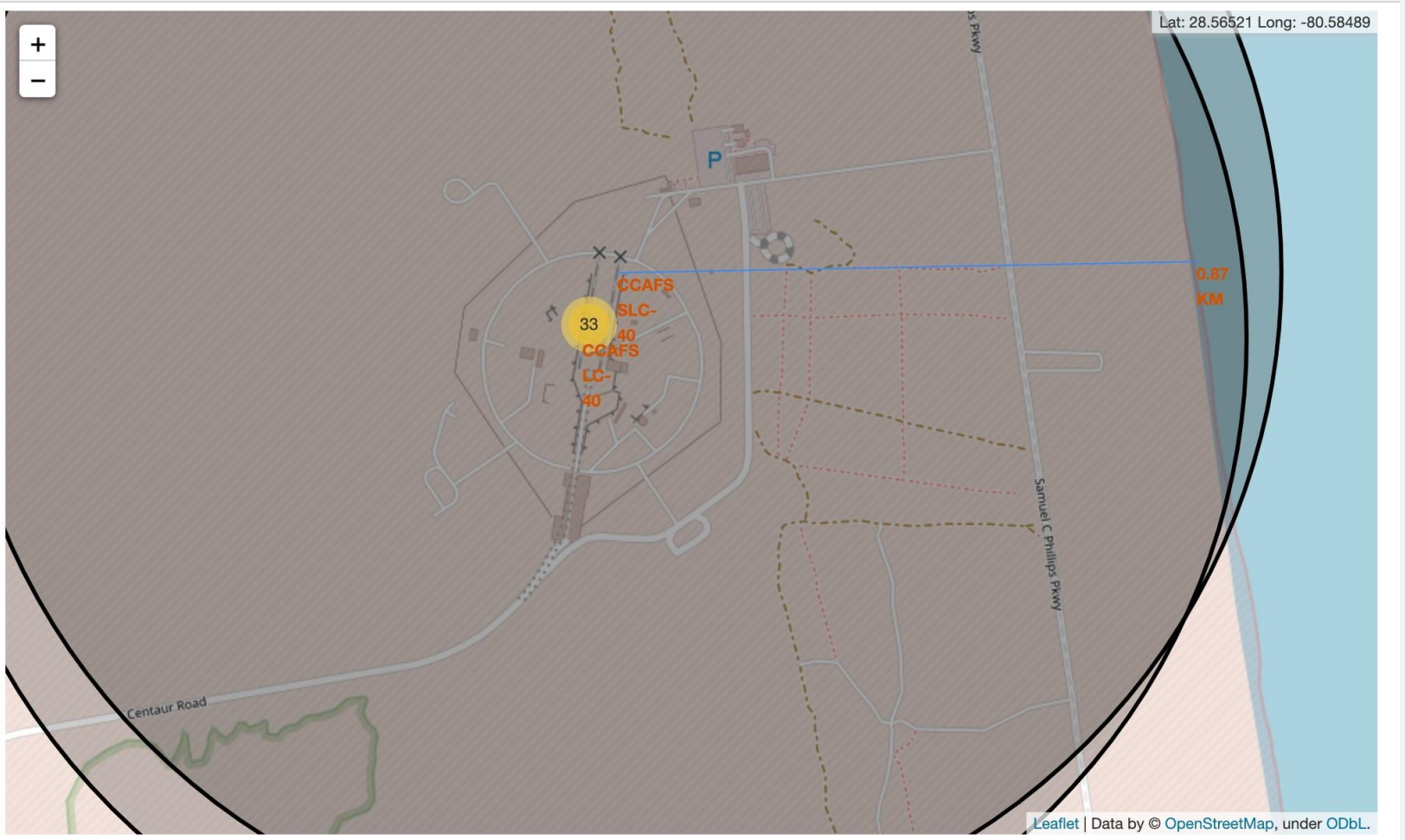
- Map showing launch sites. Sits are mark and label for easy identification

Map showing different marker color based on class

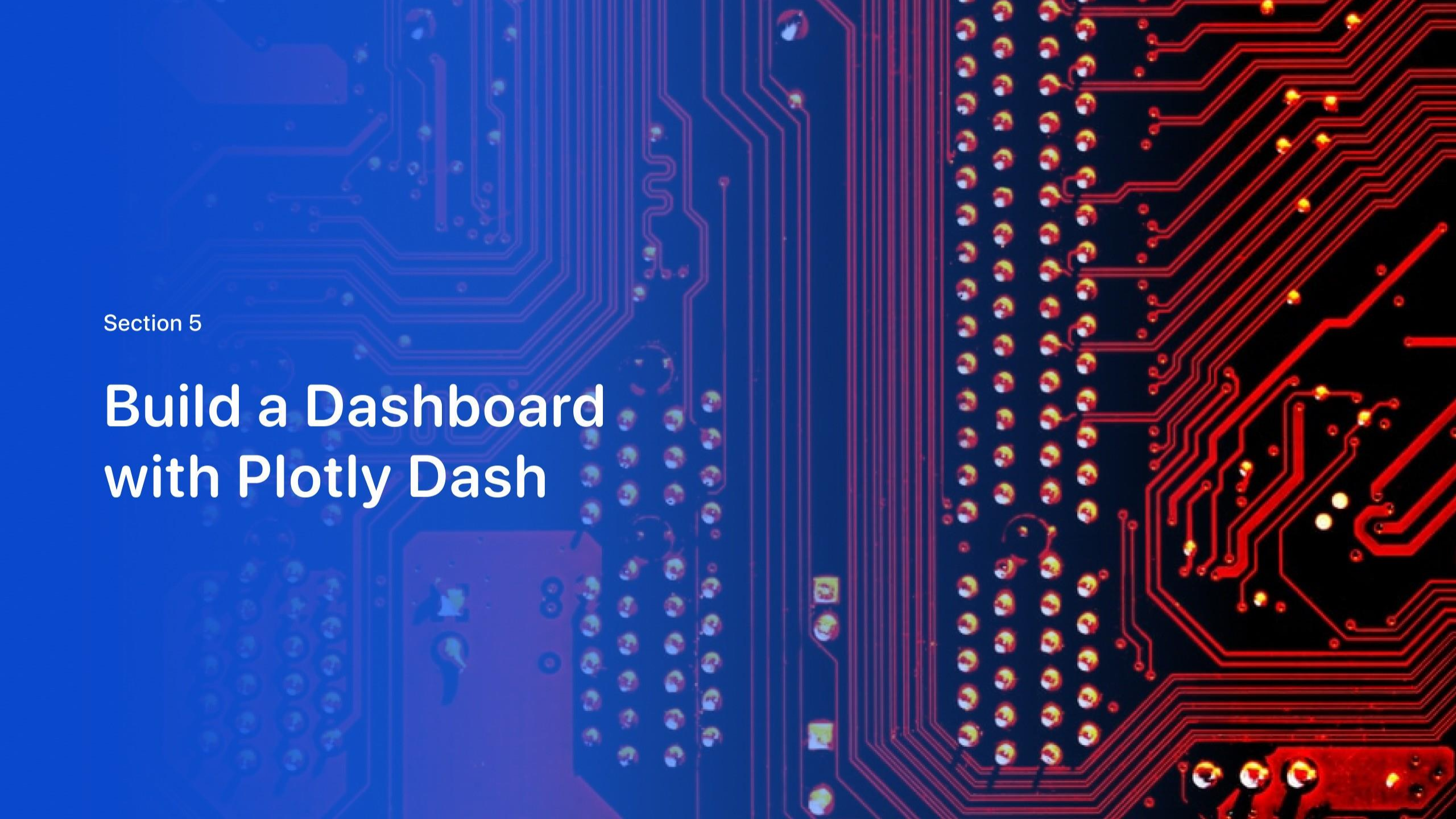


- Green markers on the plot show successful launch site

Map showing polyline to nearest coatsline



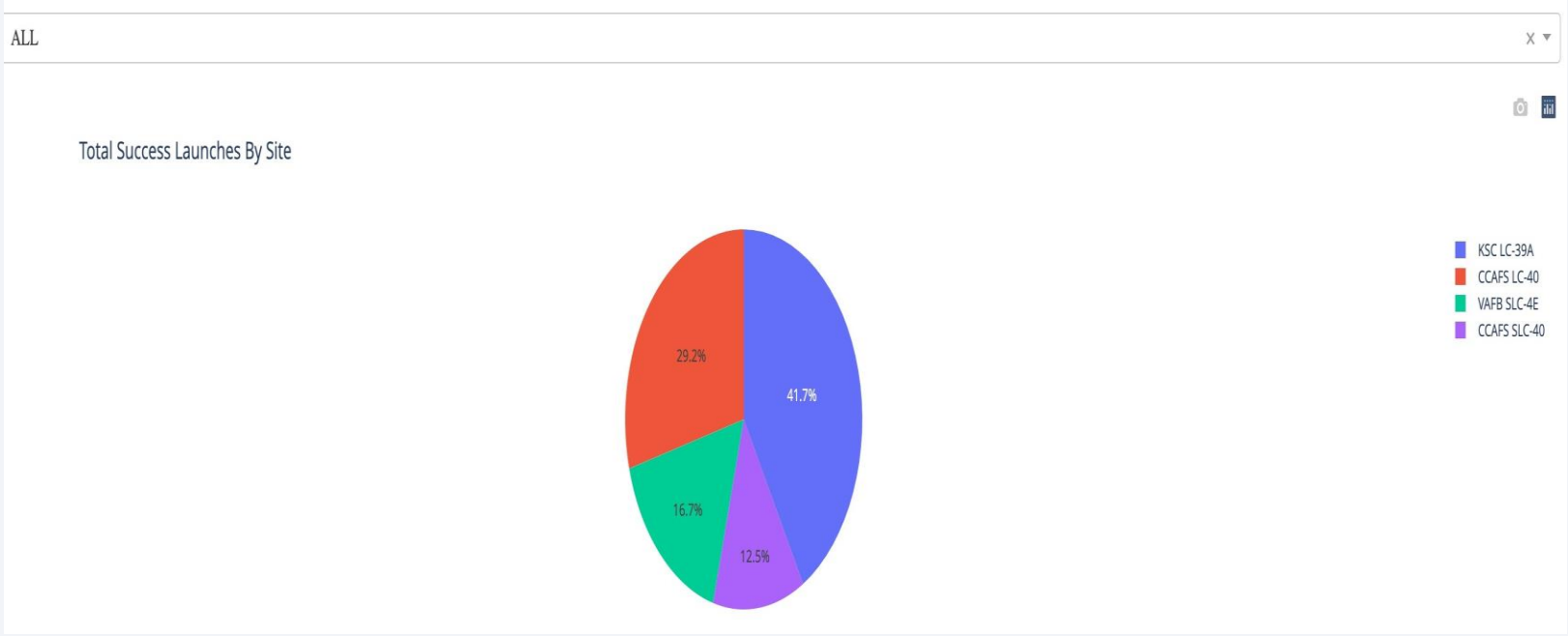
- The map shows a polyline from the coastline to the launch site.
- The PolyLine could be drawn for proximities such as railway, highways and cities



Section 5

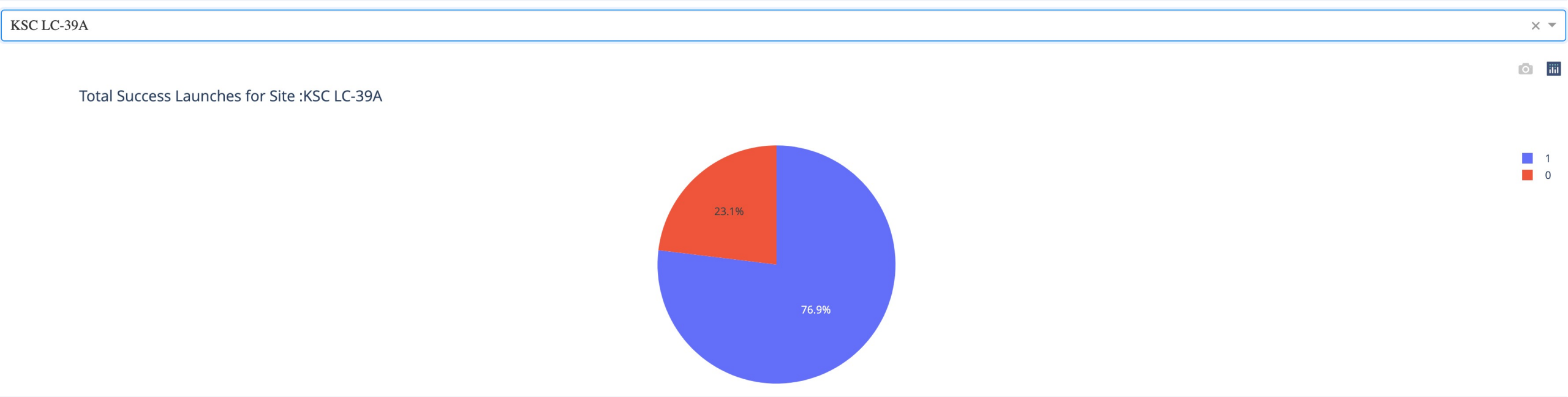
Build a Dashboard with Plotly Dash

Total success Launch by site



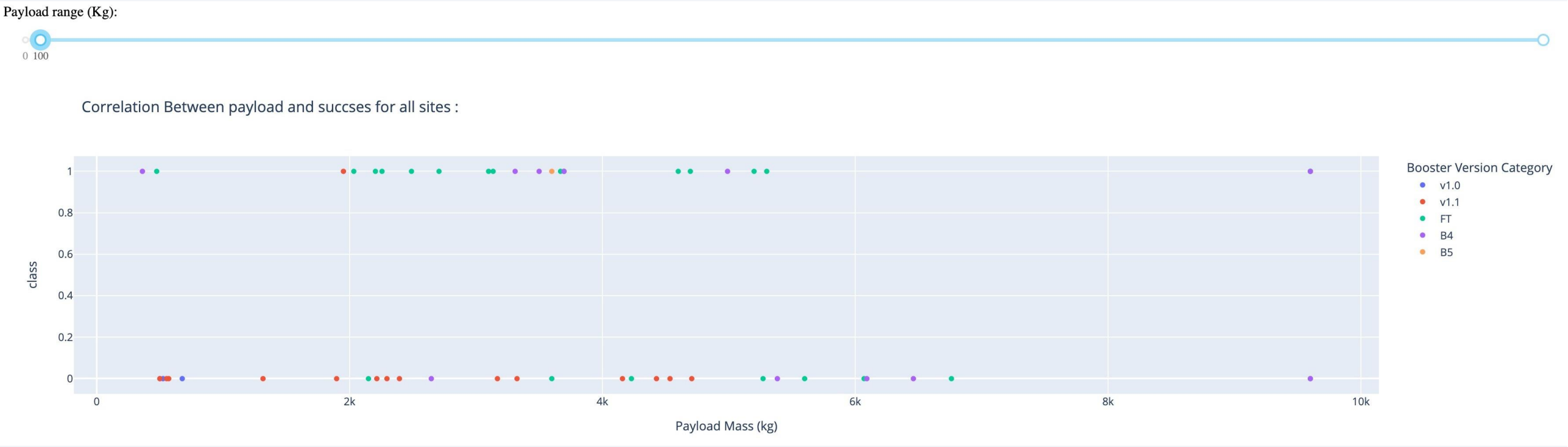
- Observed that KSCLC has a high proportion of success launches of 41.7% followed by CCAFSwith 29.2%, then VAFB SLCand CCAFS with 16.7% and 12.5% respectively.

Total Success Launches for site KSC LC-39A



- Observed that KSC LC-39A site has about 77% success rate

Correlation between payload and success rate



- Observed high cluster of success bwteeen payloads 2k and 6k.
- Booster category FT has a high success rate

Section 6

Predictive Analysis (Classification)

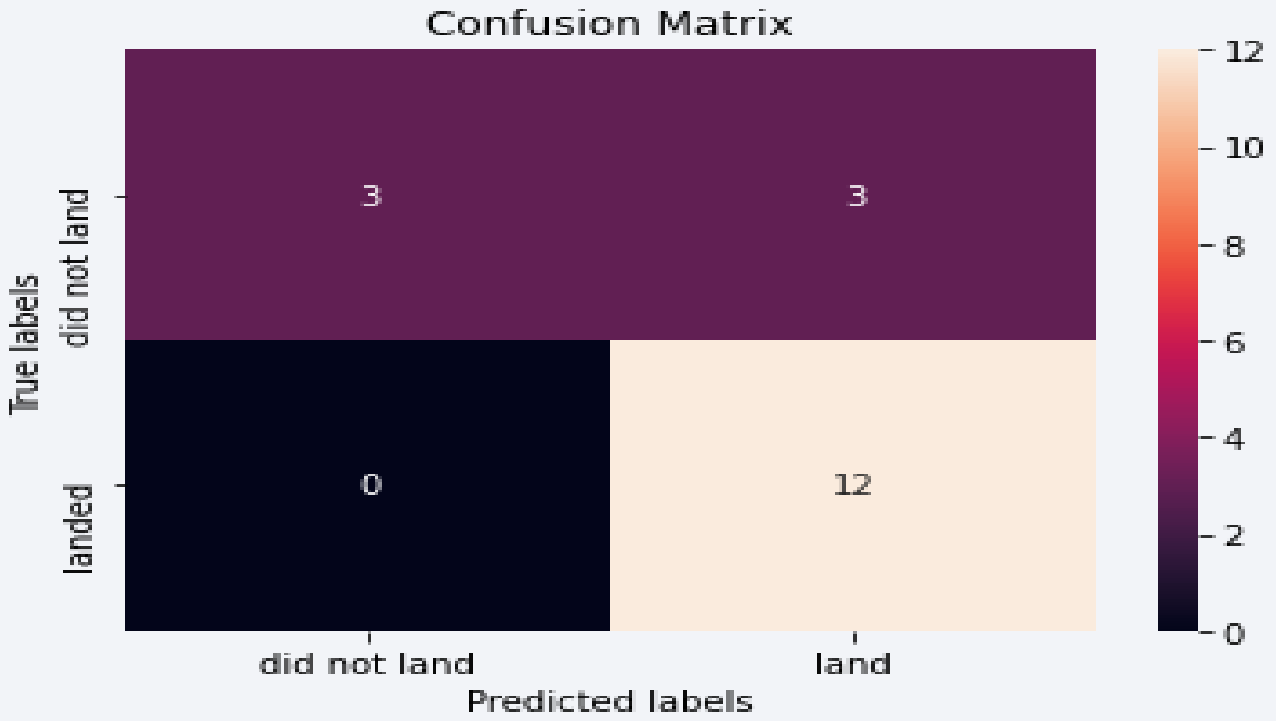
Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart

- Find which model has the highest classification accuracy

Confusion Matrix

- The model performed well with an accuracy score of 88.89%.
  From the confusion matrix, the model predicted a high proportion accurately with a few false negatives.



Conclusions

- Launch success rate has been increasing since 2013 to 2020
- We see that different launch sites have different success rates. CCAFS LC-40, has a success rate of about 60%, while KSC LC-39A and VAFB SLC 4E has a success rate of 77%.
- Observed 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.
- Observed a high correlation between payload mass and success of launch
- Decision tree classifier had the highest accuracy of 88.88% in predicting the outcome a a launch

Appendix

- https://github.com/KeneOkey/Capstone.git

Thank you!