



IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Shiju Raj S
25-Nov-2021



Outline

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

Executive Summary

In this commercial Space age, various Companies are making space travel affordable for everyone. The most successful is SpaceX which include sending spacecraft to the International Space Station and manned missions to Space. SpaceX rocket launches are relatively inexpensive as much of the savings is because SpaceX can reuse the first stage.

Summary of methodologies

- 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 collected from various models depicts compared with actual predict an over 90% 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**

- Space travelling is fast becoming a commercial business and concerns are high as to how to make commercial space travel less expensive.
- To achieve this the cost of travel should include the reduction in cost for rocket launches.
- SpaceX Falcon9 rocket launches has been relatively cheap costing about of \$62 million; when compared to other players
- SpaceX is able to provide much of the savings to others costing between \$165 million dollars to \$200 million because SpaceX can reuse the first stage.

- **Problems you want to find answers**

- This project aims to determine and predict the success of first stage and will be using existing information from previous launches. If we can determine whether the first stage will be successful, then the cost of a launch can be determined. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

Section 1

Methodology

Methodology

Executive Summary

- **Data collection methodology:**
 - Data collection was achieved using SpaceX RESTAPI 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 Payload Mass with the average payload mass from the dataset

Methodology contd..

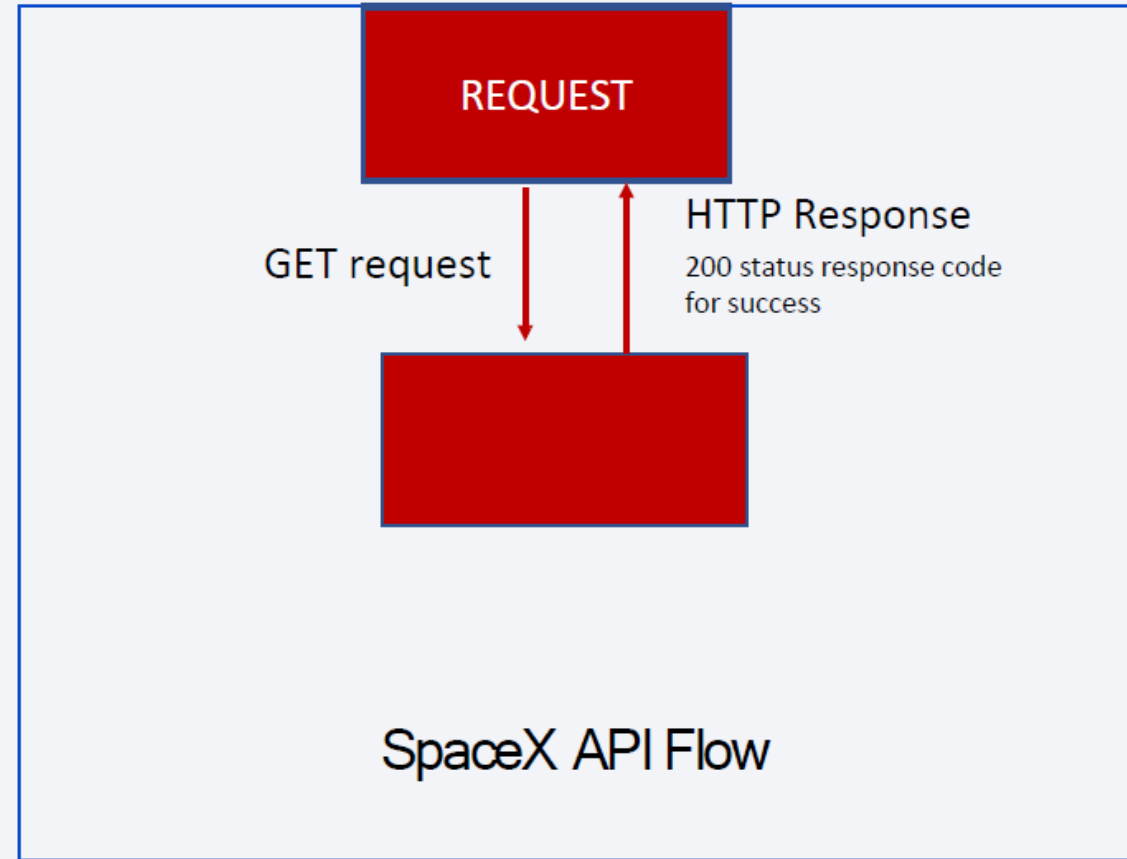
- **Perform exploratory data analysis (EDA) using visualization and SQL**
 - Exploratory Data Analysis is the first step of any data science project and can be used to automatically determine if the Falcon 9's second stage will land. Some attributes can be used to determine if the first stage can be reused with the help of various visualizations
- **Perform interactive visual analytics using Folium and Plotly Dash**
 - Success and failure Analysis is performed using interactive dashboards created in Folium and Plotly
- **Perform predictive analysis using classification models**
 - How to build, tune, evaluate classification 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 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.
- Falcon9 historical launch records were web scraped from an HTML table on Wikipedia page titled "ListofFalcon9andFalconHeavylaunches" to collect records, using the Beautiful Soup library

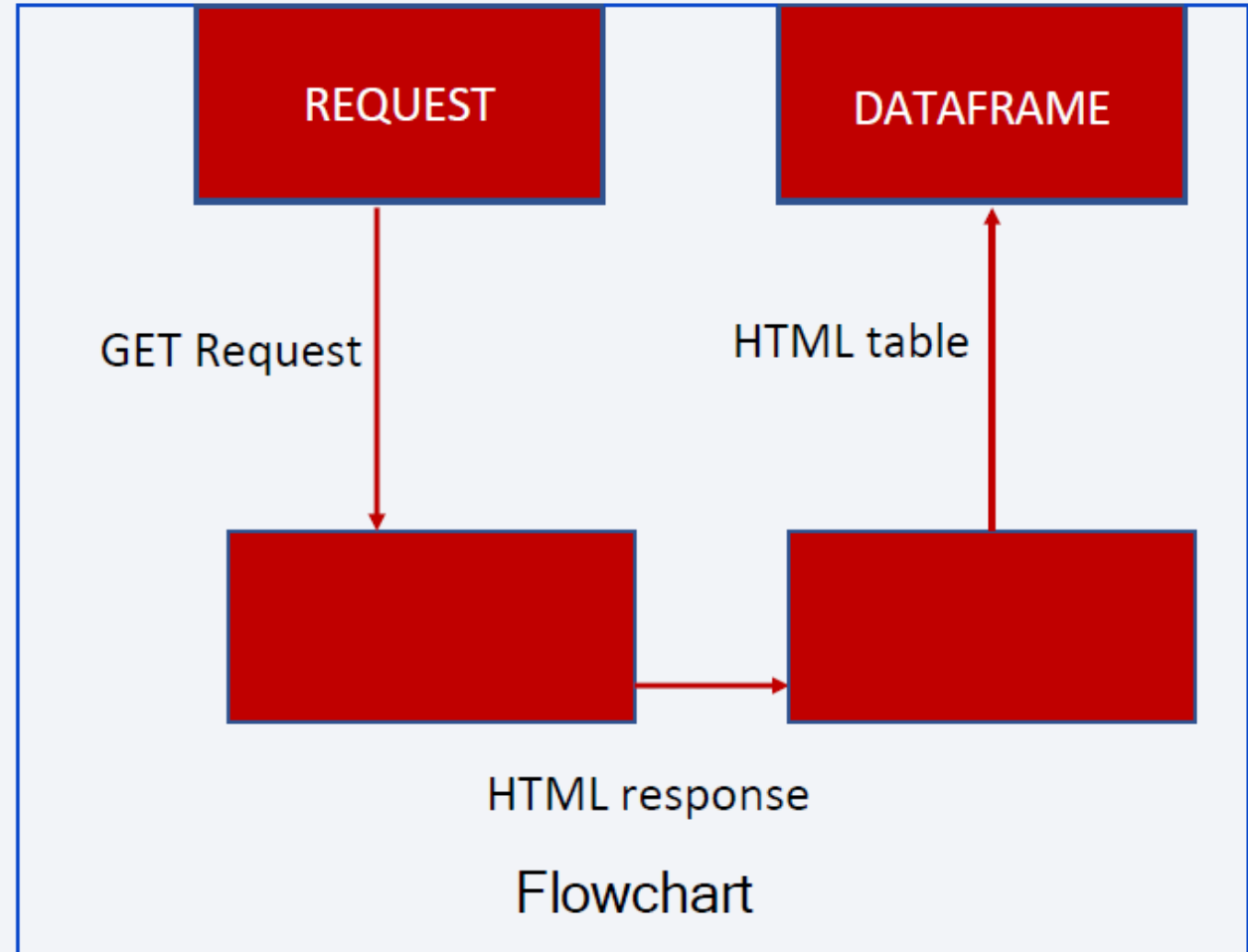
Data Collection – SpaceX API

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



Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts



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.
- All 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.
- As part of exploratory data analysis, libraries were imported and defined auxiliary functions for processing the dataset.
- Identifying the data types using dataframe types() method.

Data Wrangling (contd..)

- 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 our dataset
- With the output, a list created where the element is zero for the corresponding row in Outcome is in these 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 '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 VAFBSLC4E has a success rate of 77%.

EDA with Data Visualization (contd..)

- 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 Flight Number 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 (contd..)

- In addition, the trends in the yearly launch success rate by plotting 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's observed from the plot that the success rate has been increasing since 2013 till 2020

<https://github.com/shijurajs/Capstone/blob/master/EDA%20with%20Data%20Visualization.ipynb>

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 PAYLOADMASS_KG datatype changed to INTEGER.
- Python libraries 'sqlalchemy', '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.
- <https://github.com/shijurajs/Capstone/blob/master/EDA%20with%20SQL.ipynb>

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 sitemap to allow for easy location on the map
- All launch sites were found to be close to the coast line 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-labelled markers in marker clusters, able to easily identify which launch sites have relatively high success rates

Build an Interactive Map with Folium (contd..)

- 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

<https://github.com/shijurajs/Capstone/blob/master/Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb>

Build a Dashboard with Plotly Dash

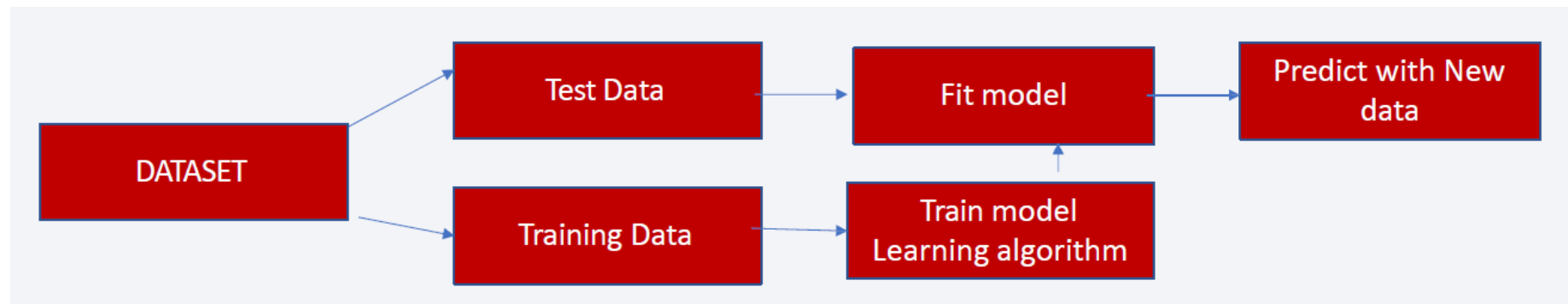
- A Plotly Dash application for users to perform interactive visual analytics on SpaceX launch data in real-time.
- A drop down added to allow for filtering by site or selection all sites for interactive plots.
- Plotted a pie chart showing the total success launches by site
- Added a point scatterplot 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 pre-processing. 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) (contd..)

- 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.



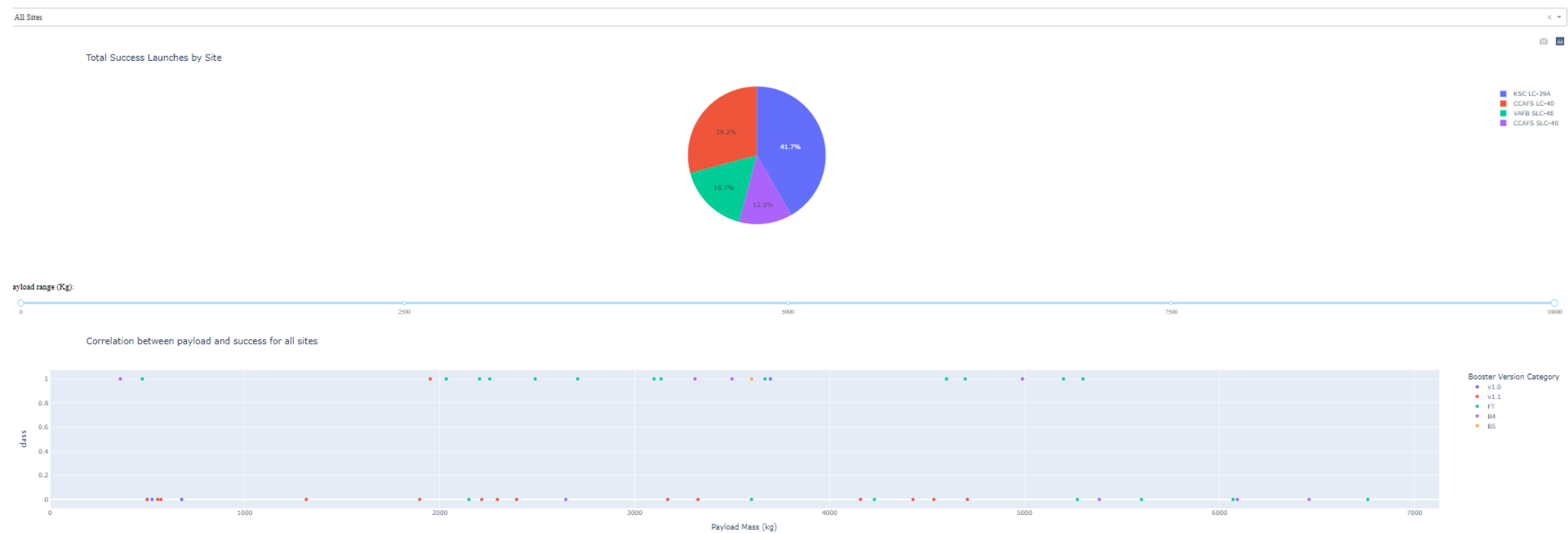
Results

- Different launch sites have different success rates. CCAFSLC-40, has a success rate of about 60%, while KSCLC-39A and VAFBSLC4E 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 SSO orbits show a high success rate visualized
- Furthermore, 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.
- 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 (contd..)

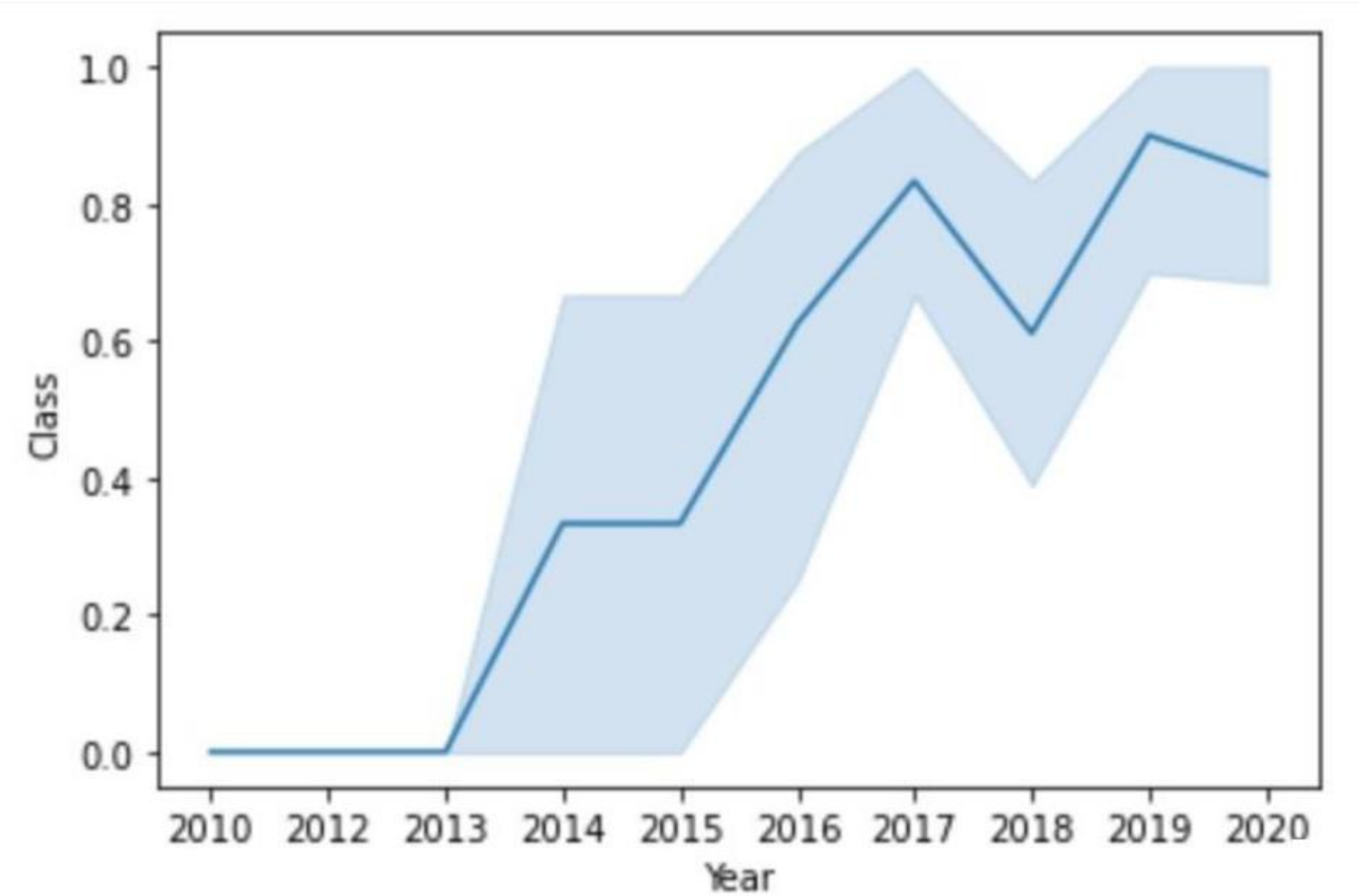
Screenshots depicts interactive analytics

SpaceX Launch Records Dashboard



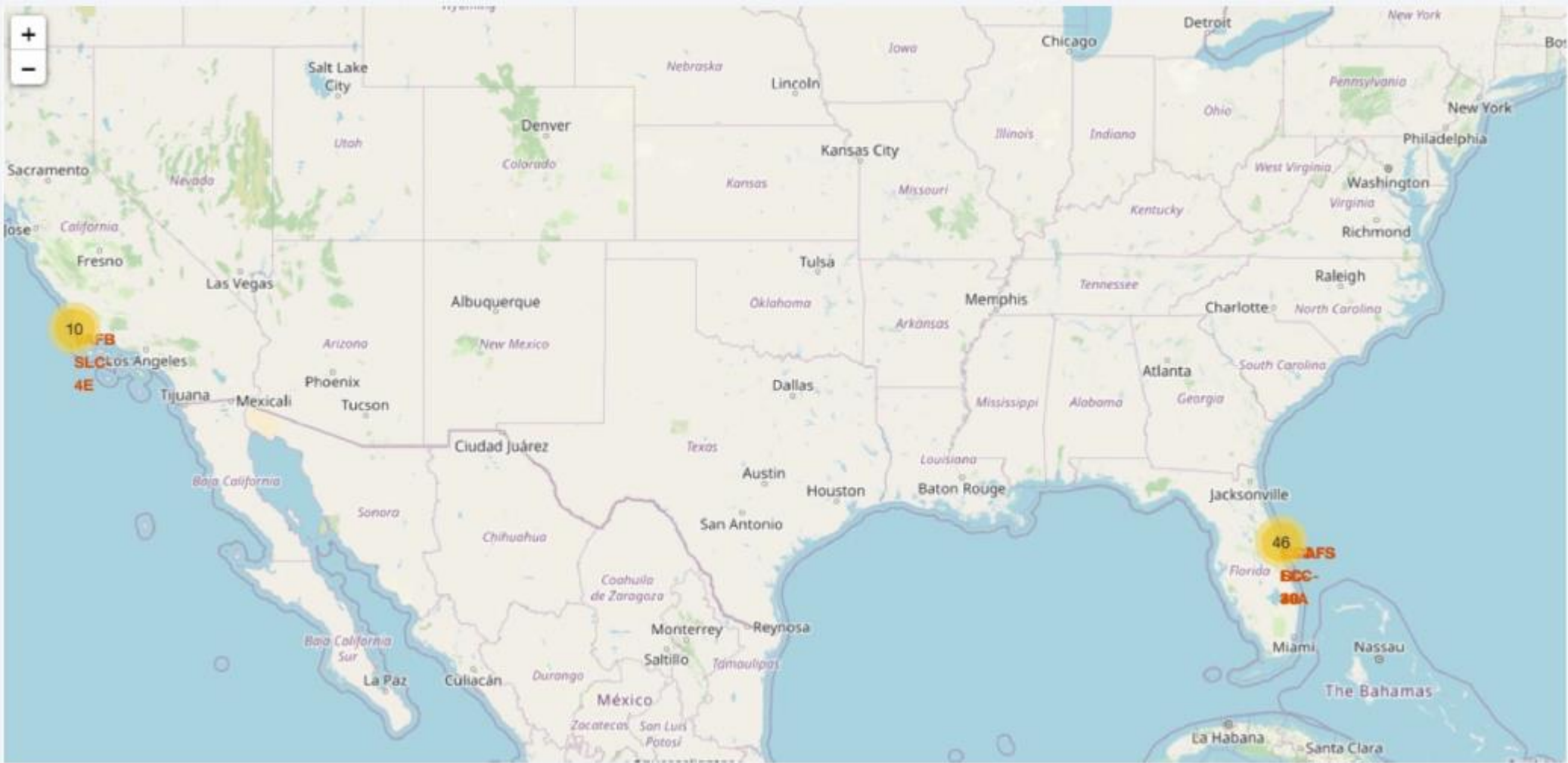
Results (contd..)

Annual Successful Launch Trend



Results (contd..)

Launch site across Geospatial Location

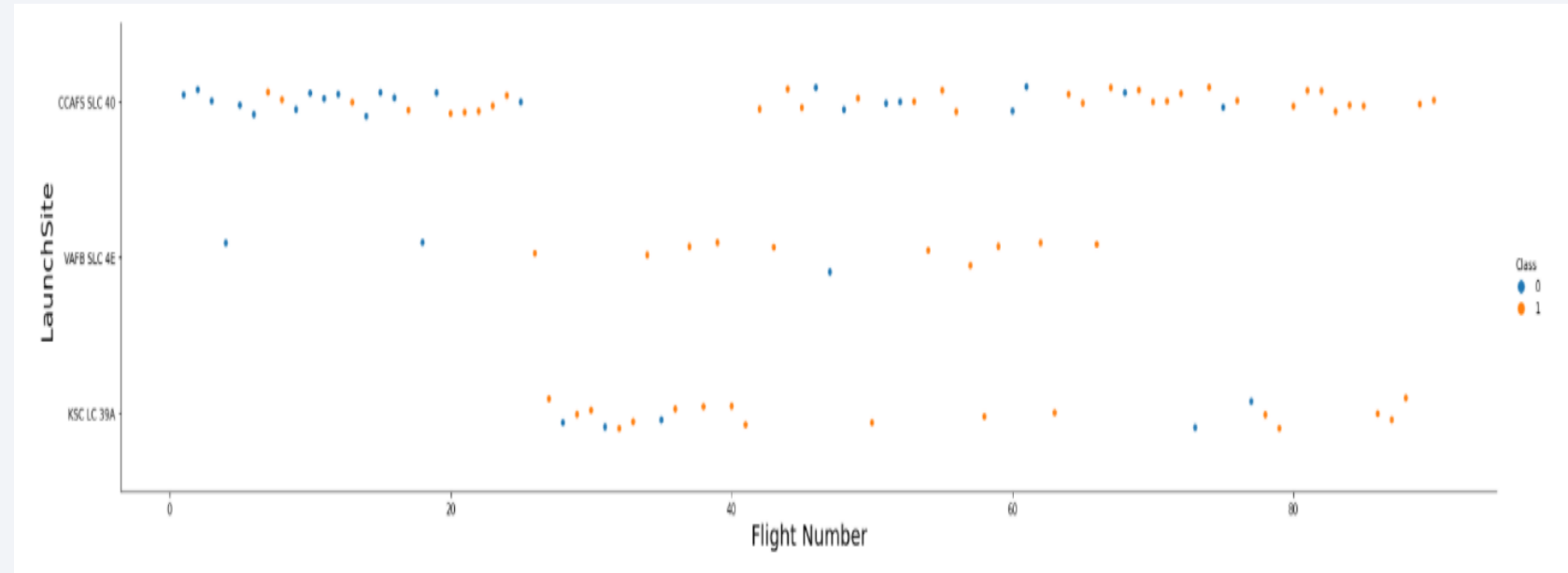


Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

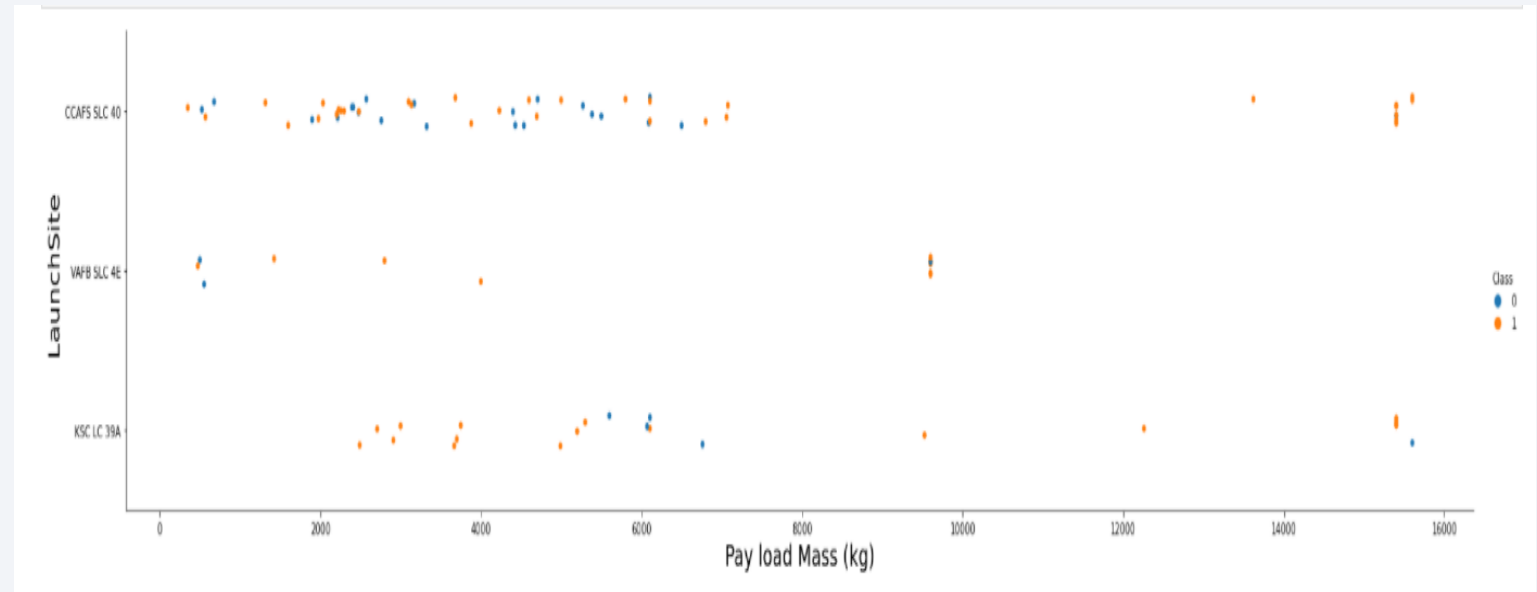
- Observed different launch sites have different success rates. CCAFS LC-40, has a success rate of 60%, while KSC LC-39A and VA FB SLC 4 E has a success rate of 77%.



Payload vs. Launch Site

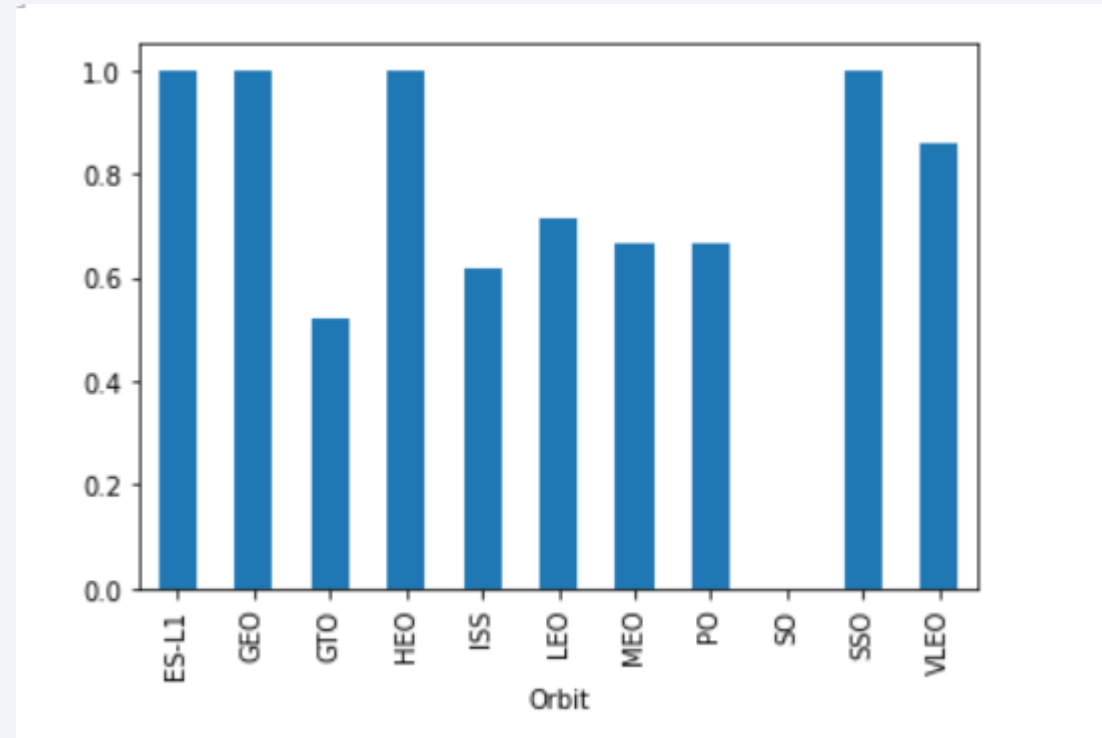
From the plot Several flights are clustered around payload mass below 7000.

- Site KSC LC had a high proportion of success for payloads between 2000 and 6000
- Site CCA FS has about 83.33% success rate for payload mass above 8000 and about 55% for payloads below 8000 whereas VA FB has about 66% success for lower payloads below 8000 and about 75% for payloads above 8000



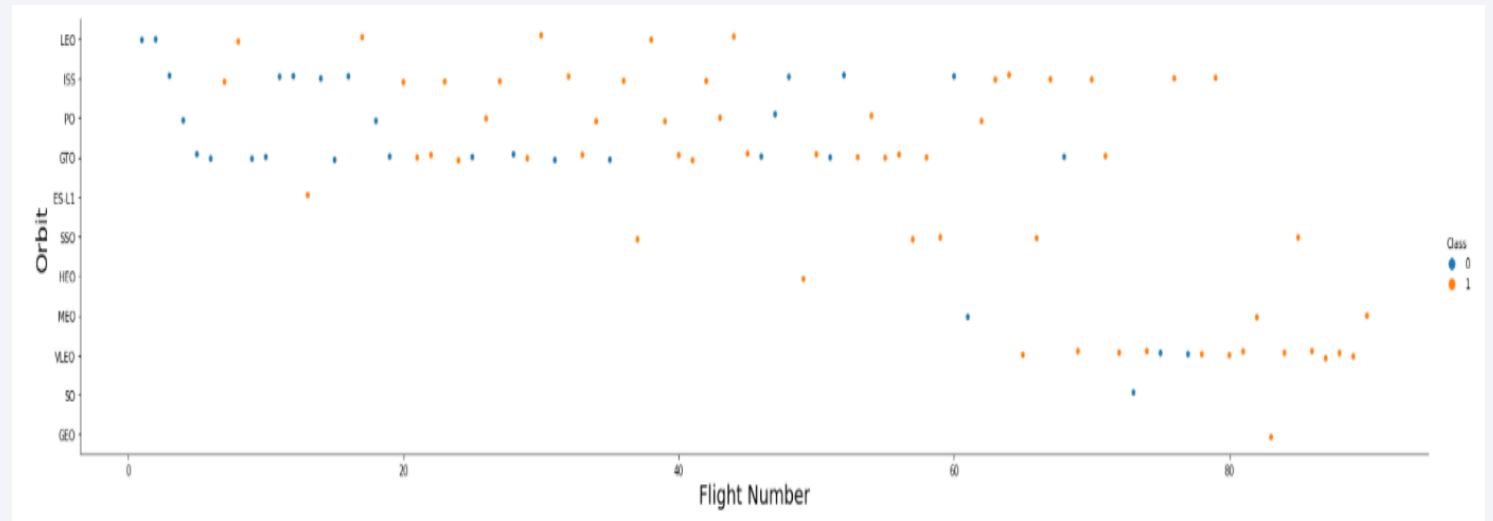
Success Rate vs. Orbit Type

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



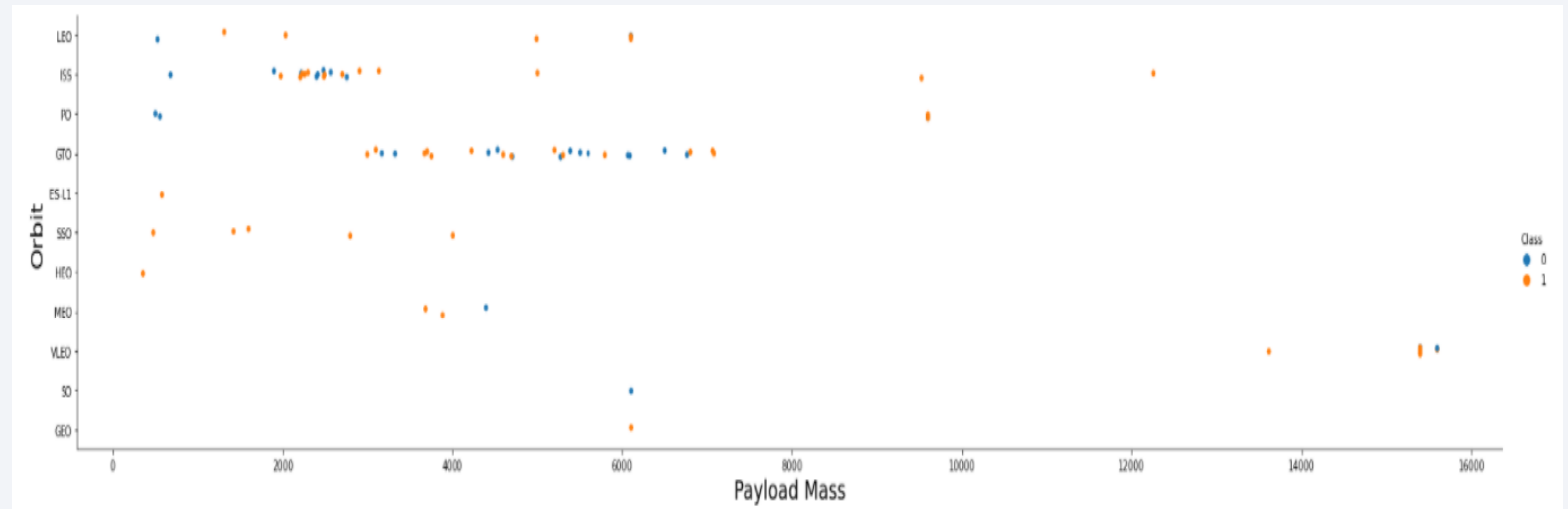
Flight Number vs. Orbit Type

- 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



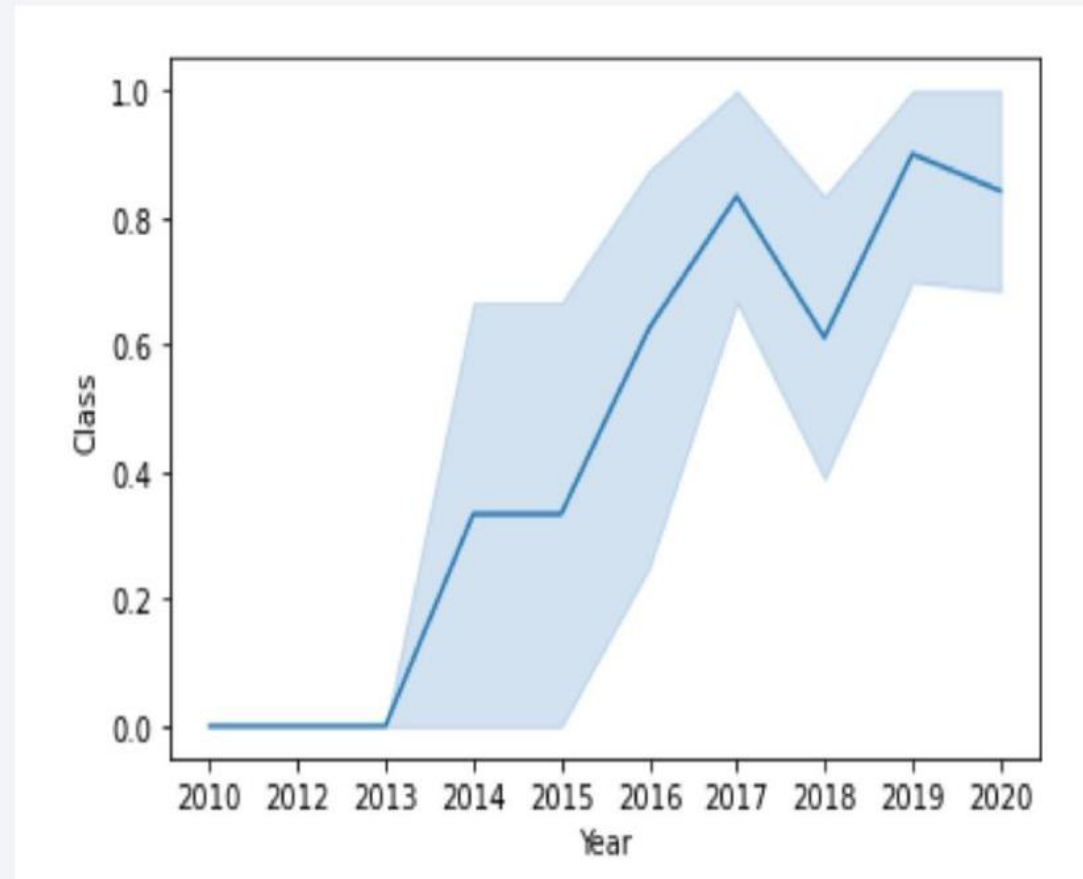
Payload vs. Orbit Type

- Heavy pay loads have a negative influence on GTO orbits and positive on Polar LEO(ISS) orbits observed.



Launch Success Yearly Trend

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



All Launch Site Names

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

```
In [20]: %sql SELECT DISTINCT LAUNCH_SITE FROM YGD26047.SPACEXTBL;
```

```
* ibm_db_sa://ygd26047:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90108kqb1od8lcg.databases.appdomain.cloud:31929/bludb  
Done.
```

```
Out[20]: launch_site
```

```
CCAFS LC-40
```


```
CCAFS SLC-40
```

```
KSC LC-39A
```

```
VAFB SLC-4E
```

Launch Site Names Begin with 'CCA'

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

In [9]:  %sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5 ;

* ibm_db_sa://df149331:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90108kqb1od8lcg.databases.appdomain.cloud:31498/bludb
 ibm_db_sa://ygd26047:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90108kqb1od8lcg.databases.appdomain.cloud:31929/bludb
 Done.

Out[9]:

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

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

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

```
In [28]: %sql SELECT SUM(payload_mass__kg_)total_Payload_mass FROM YGD26047.SPACEXTBL WHERE CUSTOMER='NASA (CRS)';
```

* ibm_db_sa://ygd26047:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8l1cg.databases.appdomain.cloud:31929/bludb
Done.

```
Out[28]: total_payload_mass
```

45596

Average Payload Mass by F9 v1.1

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 F9v1 from the where clause

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

```
In [12]: %sql SELECT AVG(payload_mass_kg_)Avg_payload_mass FROM SPACEXTBL WHERE booster_version LIKE 'F9 v1.1%';  
* ibm_db_sa://df149331:***@3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31498/bludb  
  ibm_db_sa://ygd26047:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb  
Done.
```

```
Out[12]: avg_payload_mass  
         2534
```


First Successful Ground Landing Date

- Retrieved the date when the first successful landing outcome in groundpad was achieved by using them in function on the DATE and limit to cases where landing_outcome= 'Success(groundpad)'

In [32]:

```
%sql SELECT MIN(DATE) FIRST_GROUNDPAD_SUCCESS FROM YGD26047.SPACEXTBL WHERE landing__outcome='Success (ground pad)';
```

```
* ibm_db_sa://ygd26047:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90108kqb1od8l1cg.databases.appdomain.cloud:31929/bludb  
Done.
```

Out[32]: **first_groundpad_success**

```
2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- 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__outcome='Success(drone ship)' AND PAYLOAD_MASS_KG_>4000 AND PAYLOAD_MASSKG_<6000

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

In [36]: `%sql SELECT * FROM YGD26047.SPACEXTBL WHERE payload_mass__kg_>4000 and payload_mass__kg_<6000 and landing__outcome='Success (drone ship)' ;`

* ibm_db_sa://ygd26047:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.

Out[36]:

DATE	time_utc	booster_version	launch_site	payload	payload_mass_kg	orbit	customer	mission_outcome	landing_outcome
2016-05-06	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-08-14	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-10-11	22:53:00	F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105	5200	GTO	SES EchoStar	Success	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- 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

```
In [37]: %sql SELECT mission_outcome, count(*) Outcome_Count FROM YGD26047.SPACEXTBL group by mission_outcome ;
```

```
* ibm_db_sa://ygd26047:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb  
Done.
```

```
Out[37]:
```

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

Boosters Carried Maximum Payload

- 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 maximum.

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

```
In [39]: %sql SELECT distinct booster_version FROM YGD26047.SPACEXTBL WHERE payload_mass__kg_ in (select max(payload_mass__kg_) from YGD26047.SPACEXTBL ) ;
```

```
* ibm_db_sa://ygd26047:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb  
Done.
```

```
Out[39]: 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
```

2015 Launch Records

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

In [54]: `##sql SELECT DATE, launch_site, booster_version, Landing_outcome FROM YGD26047.SPACEXTBL WHERE Landing_outcome='Failure (drone ship)' and YEAR(DATE)=
%sql SELECT * FROM SPACEXTBL where DAYNAME(DATE)='Friday' LIMIT 5`

* ibm_db_sa://ygd26047:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.

Out[54]:

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)
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt
2014-04-18	19:25:00	F9 v1.1	CCAFS LC-40	SpaceX CRS-3	2296	LEO (ISS)	NASA (CRS)	Success	Controlled (ocean)
2016-03-04	23:35:00	F9 FT B1020	CCAFS LC-40	SES-9	5271	GTO	SES	Success	Failure (drone ship)
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)

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

- 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 DESC keyword while limiting the records for DATE between 2010-06-04 and 2017-03-20 with the where clause

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 [52]:

```
%sql SELECT LANDING__OUTCOME, COUNT (LANDING__OUTCOME) LANDING__OUTCOME_COUNT FROM YGD26047.SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' A
```

```
* ibm_db_sa://ygd26047:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
```

Done.

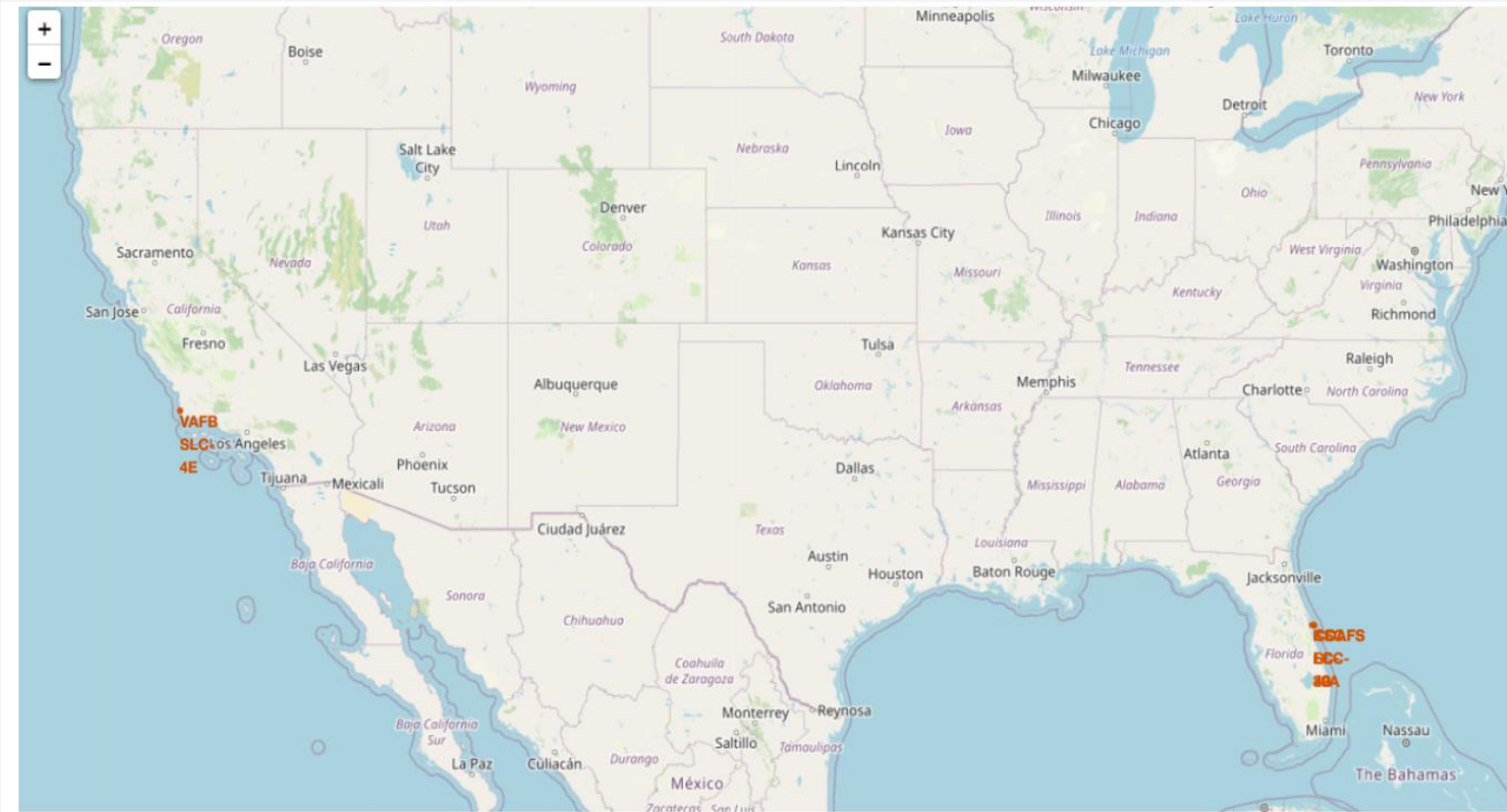
Out[52]:

landing_outcome	landing_outcome_count
Failure (drone ship)	5
Success (ground pad)	3

Section 4

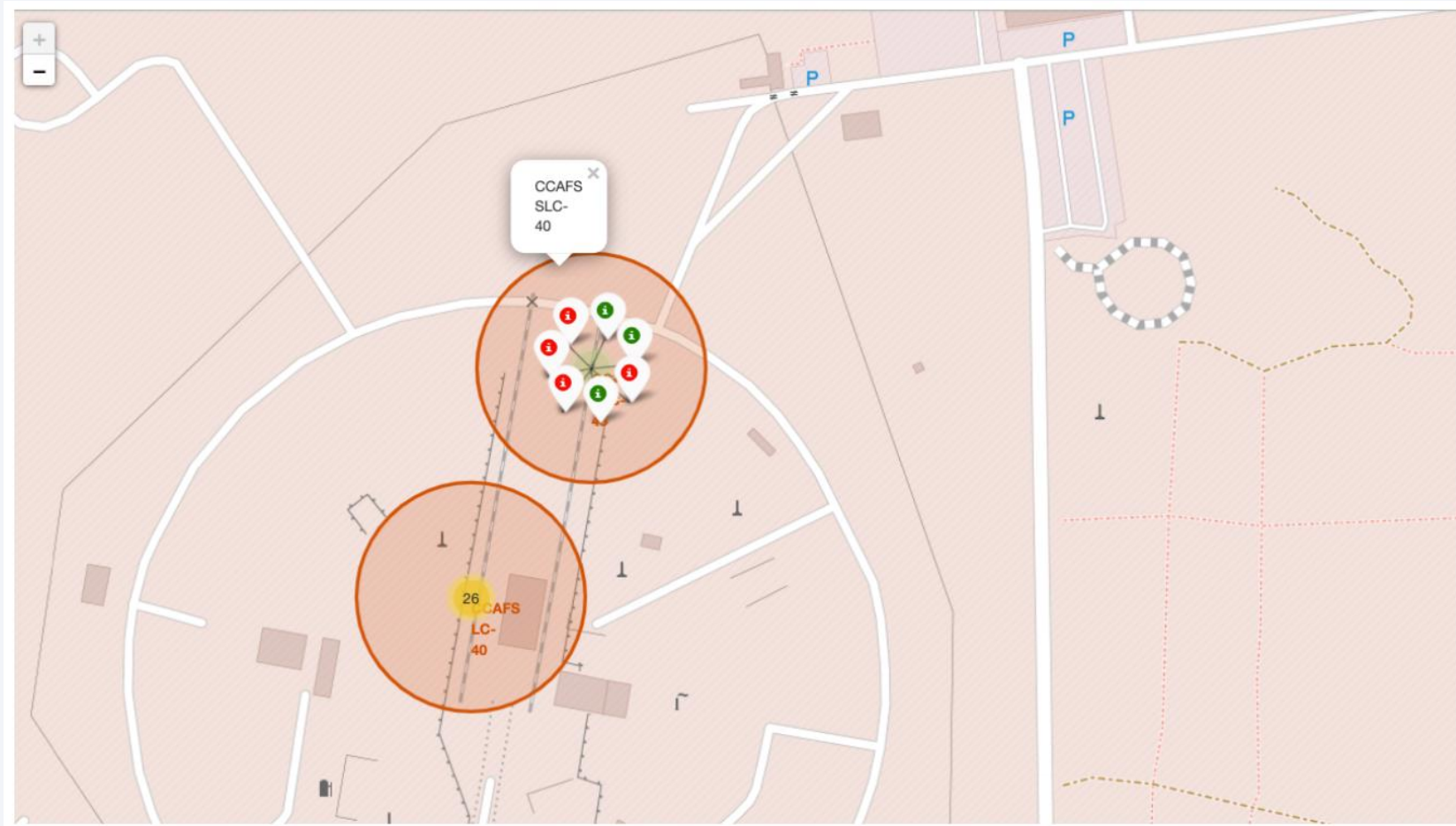
Launch Sites Proximities Analysis

Map with Launch Sites



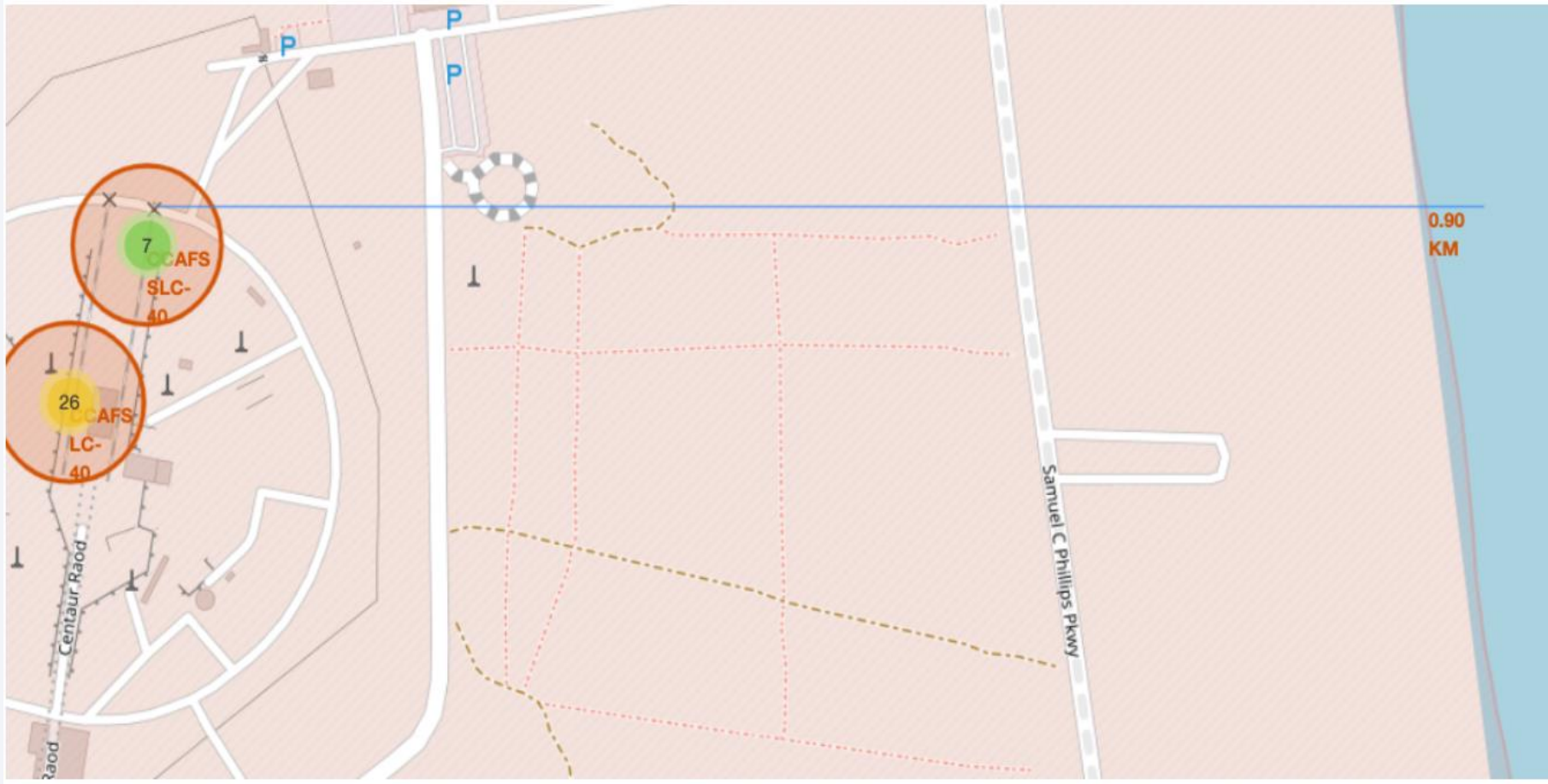
- Map showing launch sites. Sites are marked and labeled for easy identification

Map showing different marker color based on class



- Green markers on the plot show successful launchsite

Map showing polyline to nearest Coastline



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

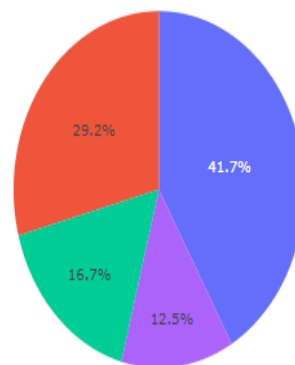
Section 5

Build a Dashboard with Plotly Dash

Total Success Launch by Site

- Observed that KSC LC has a high proportion of success launches of 41.7% followed by CCAFS with 29.2%, then VAFB SLC and CCA FS with 16.7% and 12.5% respectively.

Total Success Launches by Site



Total Success Launches for site KSCLC-39A

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

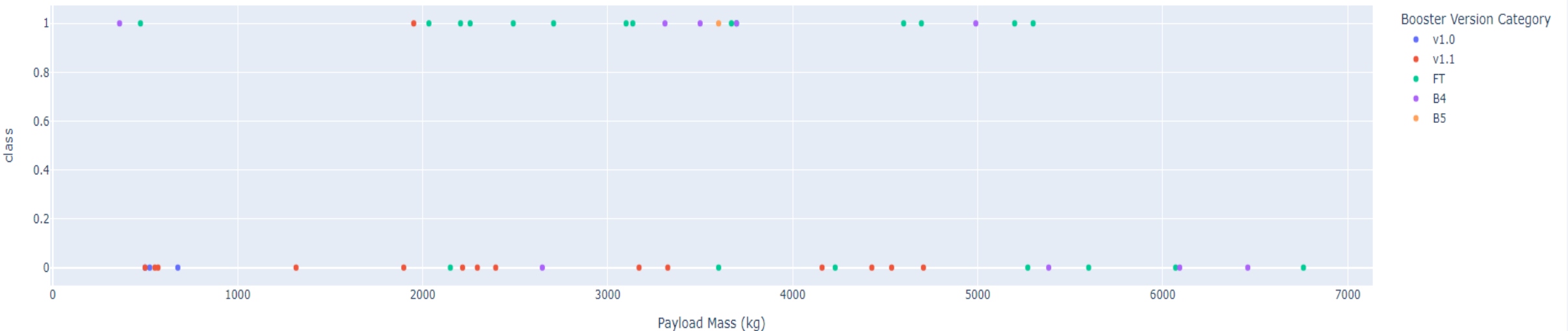
Total Success Launches for site KSC LC-39A



Correlation between Payload and Success rate

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

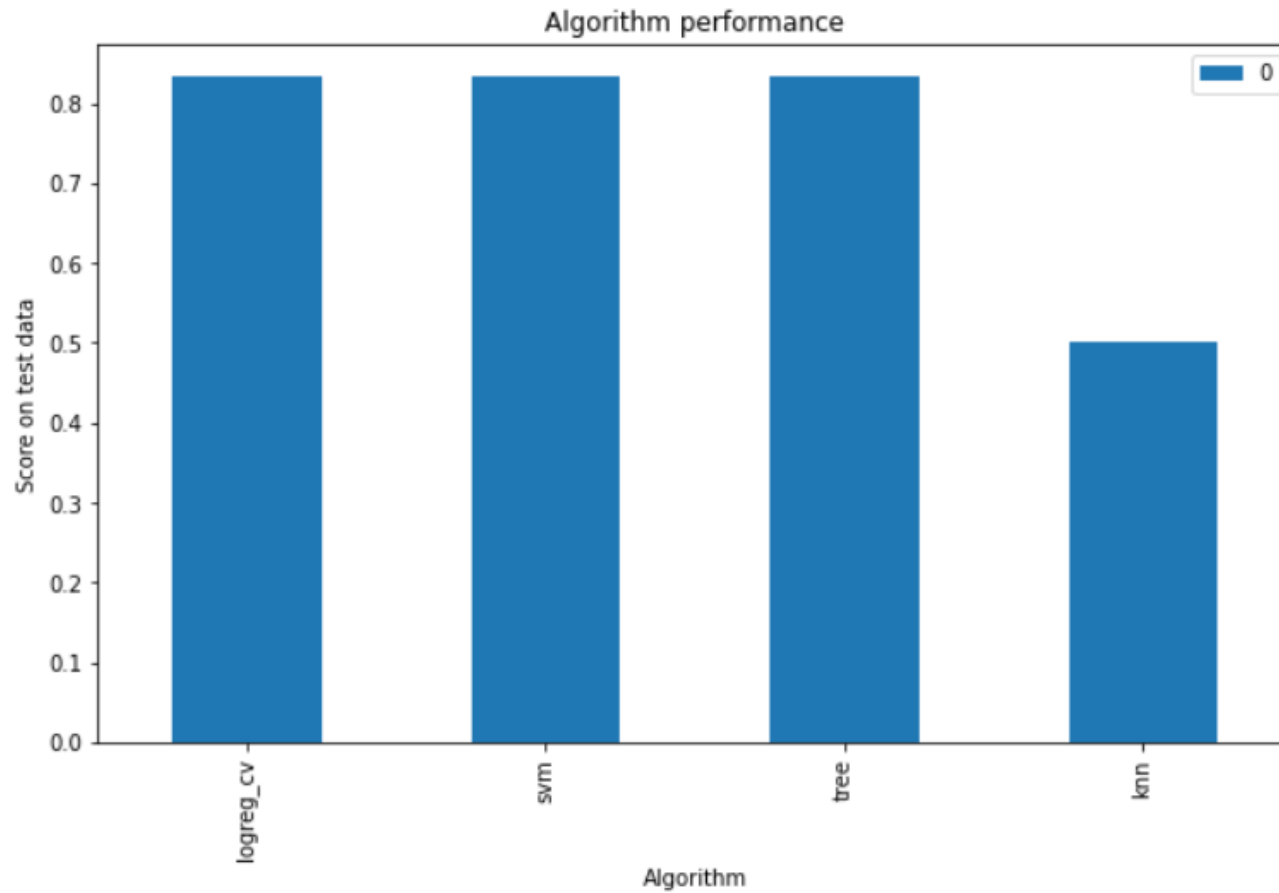
Correlation between payload and success for all sites



Section 6

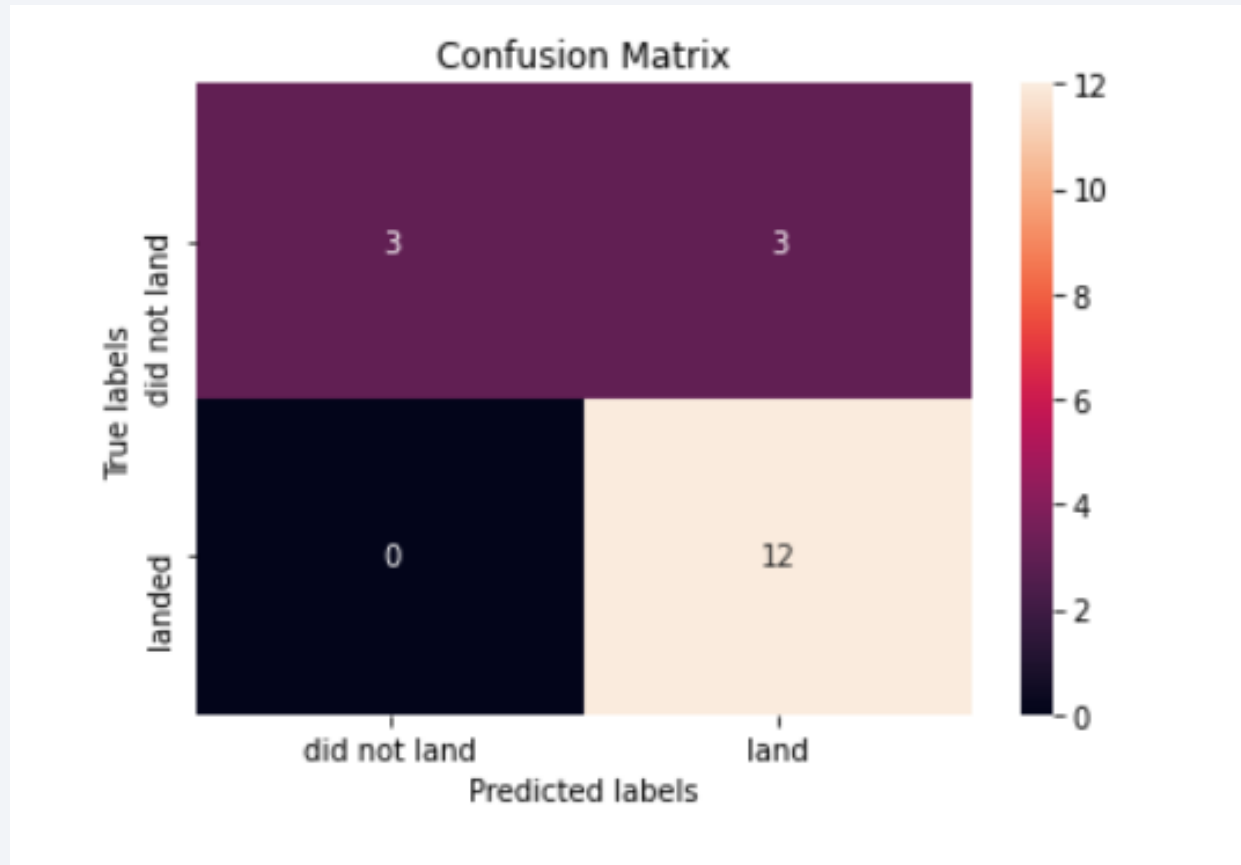
Predictive Analysis (Classification)

Classification Accuracy



- Logical Regression, Support Vector Machine and Decision Tree classifier had similar accuracy ranges

Confusion Matrix



- The SVM 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. CCAFSLC-40, has a success rate of about 60%, while KSCLC-39A and VAFBSLC4E 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
- Except KNN all other algorithms had similar accuracy of 83.3% in predicting the outcome for launch

Appendix

- <https://github.com/shijurajs/Capstone>

Thank you!

