IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Eunan Diamond
07/02/2023

Outline

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

Executive Summary

- Summary of methodologies:
- Data collection using Spacex_api and web scraping the wiki page of falcon 9 launches
- Exploratory Analysis using sql and visualization using pandas and matplotlib
- Interactive Visual Analytics and Dashboard
- Predictive Analysis Using Classification

- Summary of all results:
- Collected data through api and web scraping which allow us to carry out data wrangling, which allow us to find out the outcomes of each mission
- Used the data for EDA, which helped us understand the dataset more.
- Using all this analysis we were able to split the dataset into train and test sets and to predict if a mission will be success or not and what was the best classification technique

Introduction

- Project background and context:

- Space travel is coming more and more affordable. This is due to being able reusable rockets, for example SpaceX. This is because they can reuse the first stage, which saves them millions compared to other companies.

- Problems we want to find answers:

- Working for spaceY, spaceX competitor ,We want to find out how much launch will cost using machine learning model to predict if the first stage will land or not using SpaceX public data.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - We Collected Data using api calls using json calls to spacex_api
  - Web scraping using beautifulsoup library on list of falcon 9 and heavy launches wiki
- Perform data wrangling
  - Used the data collected to calculate each mission outcome and used this to create a outcome label, 0= not success, 1 = success
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Used different ranges of classification techniques such as KNN, Tree classifier, Used cross validation to tune to best parameters and then used best_Score method and confusion matrix to evaluate each model

Data Collection

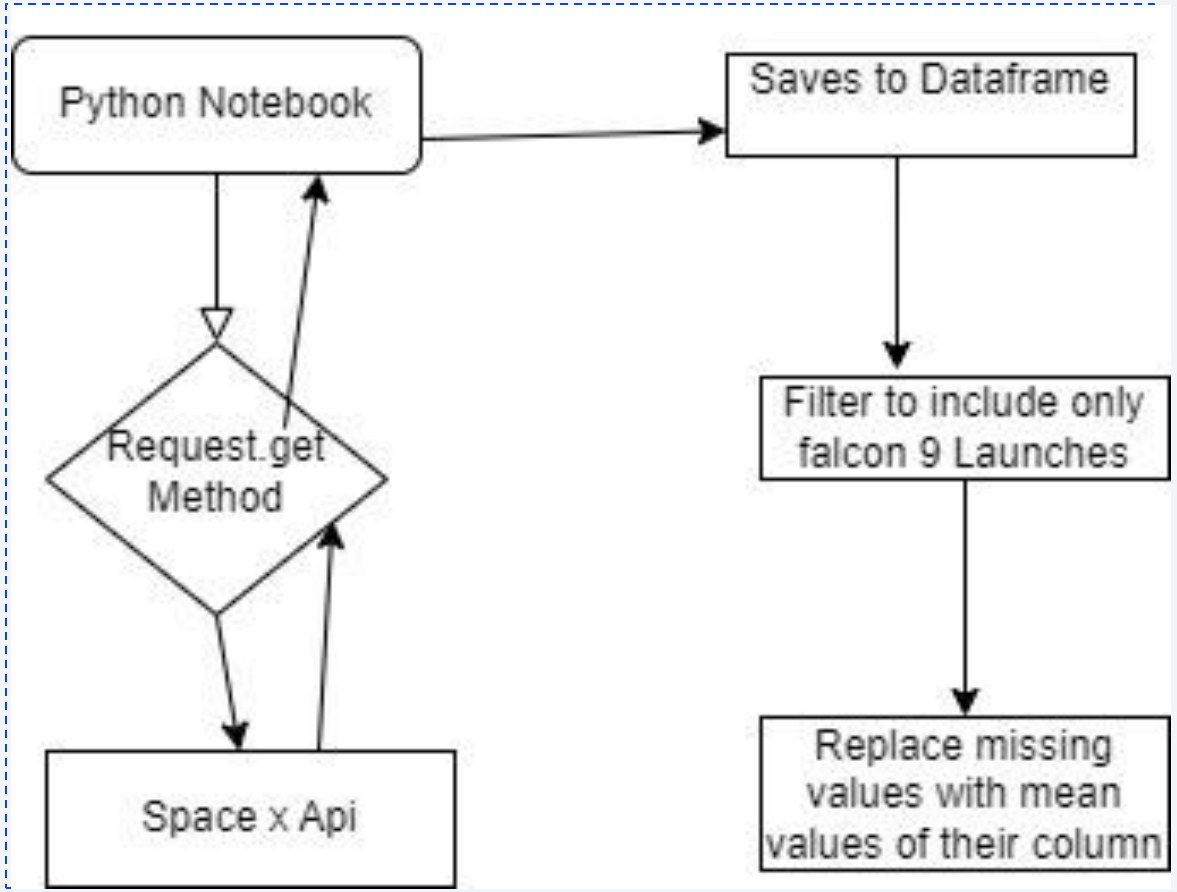
- data sets were collected by:
- Using the public Spacex Api (https://api.spacexdata.com/v4/launches/past) and converting in into a data frame.
- Performed Web scraping on wiki page (https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922) to find table elements showing data on each falcon 9 launch
- Performed Data wangling to outcome label to make it easy for analysis if a launch was successful or not. 0 for not successful, 1 for successful.

Data Collection – SpaceX API

- Space x offer a public api to use. We used the repsonse.get method in python to use it and then save to dataframe.

- Code In Github repo:

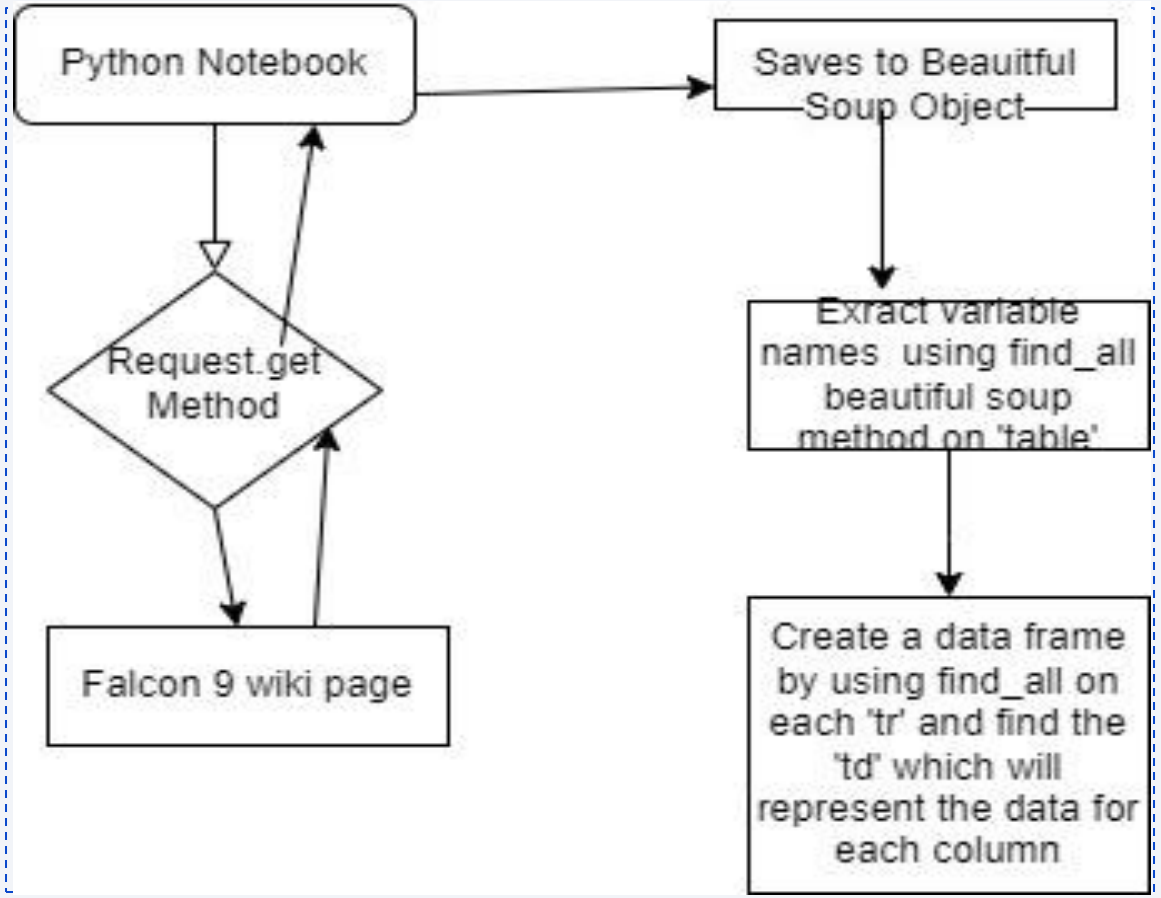
https://github.com/Eunan02/IBM-Data-science-Capstone-Project-/blob/main/Data-Collection-api.ipynb



Data Collection - Scraping

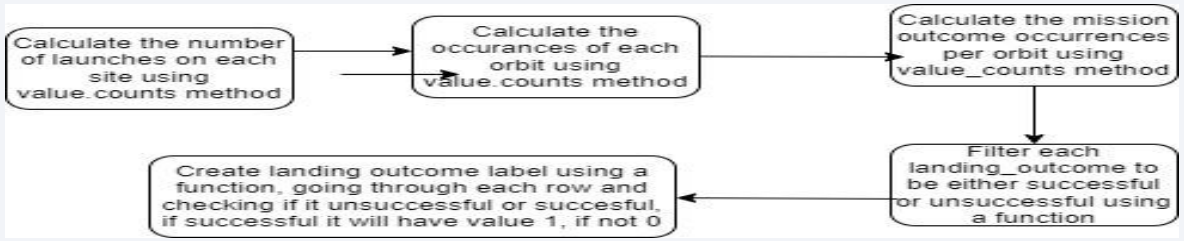
- Using beatifulSoup python library and falcon 9 and heavy launches wiki we were able to get data on the falcon 9 launches

- Link to Github repo:
- https://github.com/Eunan02/IBM-Data-science-Capstone-Project-/blob/main/Data-collection-with-web-scraping.ipynb



Data Wrangling

- Data wrangling was needed to find patterns and determine the labels for the training models.
- We took each landing_outcome column in each row in the data frame to see if they were successful or not, these were in text so we create a function for ones that were unsuccessful to be 0 in our new label and 1 if it was a successful launch
- Code in Github repo: https://github.com/Eunan02/IBM-Data-science-Capstone-Project-/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_1_L3_labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb



EDA with Data Visualization

- We plotted a range of charts of relationships between varaibles and the mission outcome these included :
- Relationship between payload mass and flight number
- Launch site and flight number
- Orbit type and Success rate
- We did this so we could discover which features would be siginifcant or related to tell if a mission will be success or not
- Github: https://github.com/Eunan02/IBM-Data-science-Capstone-Project-/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- We carried out queries such as :
- Total payload mass carried by boosters launched by NASA
- When The 1st successful landing outcome in ground pad was achieved
- Total number of successful and failure outcomes
- Number of successful outcomes between 04-06-2010 and 20-03-2017 in descending order
- Github: https://github.com/Eunan02/IBM-Data-science-Capstone-Project-/blob/main/jupyter-labs-eda-sql-coursera_sqllite%20(1).ipynb

Build an Interactive Map with Folium

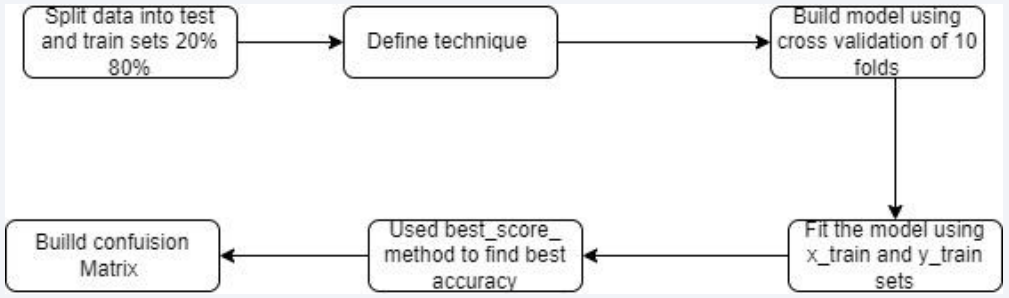
- Created folium map to show geological location of launch sites using markers to plot coordinates, and using add.child method to add to the graph, we used this to also show where each launch took off and if it was successful or not using green and red colours
- Used folium.circle to add the circle radius around each site
- We used these objects as this interactive map makes it easy to understand where each site is located and if there is any patterns with successful launches based of location
- Github: https://github.com/Eunan02/IBM-Data-science-Capstone-Project-/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Created a dropdown menu to filter between each launch site as well as all launch sites
- Pie chart for each launch site and for all sites
- Scatter graph to show relationship between payload mass and if a mission is successful not as well as the booster version category, it is able to scale between the value you want to show for payload mass.
- Dashboard is a great way for anyone to understand the data as it is interactive and can be easily filtered
- Github: https://github.com/Eunan02/IBM-Data-science-Capstone-Project-/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- We first split the data into train and test set , with 20% being used as test data
- For each classification technique we defined the technique, then used cross validation to make the model, with 10 cross validation sets. We then fit the model on the x_train and y_train sets
- We then used the best_score_ method to find the best accuracy and then plotted the confusion matrix
- We used the confusion matrixes and accuracy scores to decide which model was the most accurate
- Github: https://github.com/Eunan02/IBM-Data-science-Capstone-Project-/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

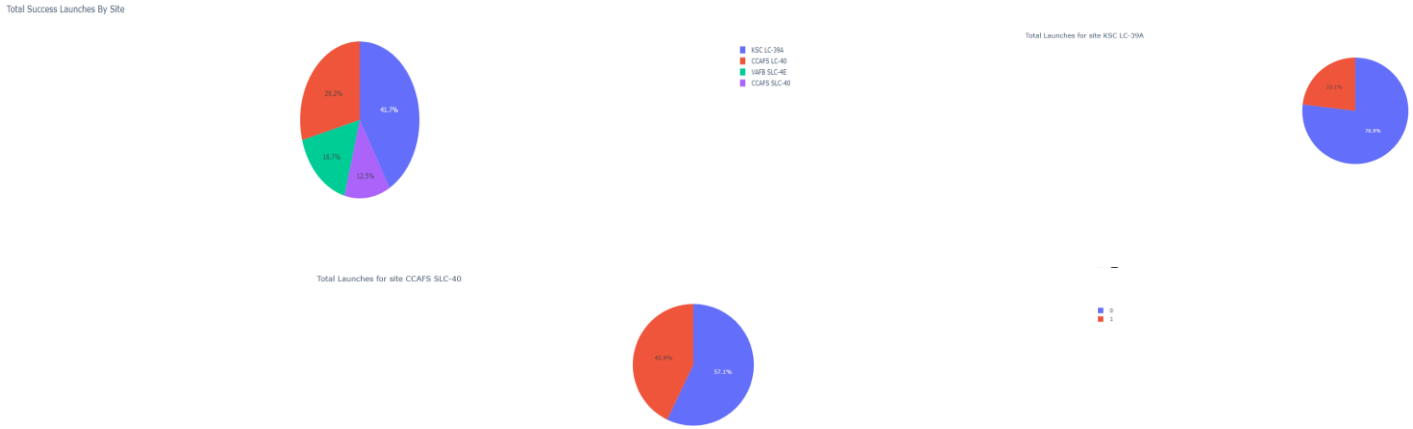


Results

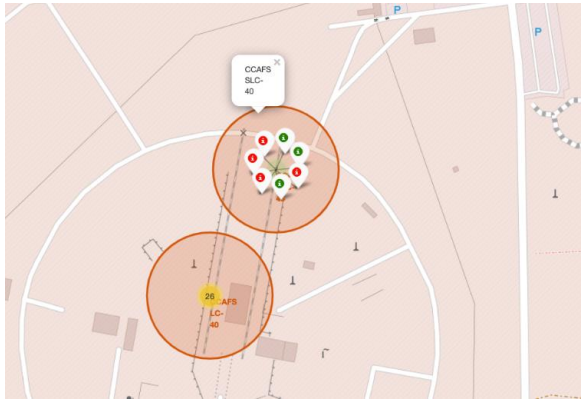
Exploratory data analysis results:

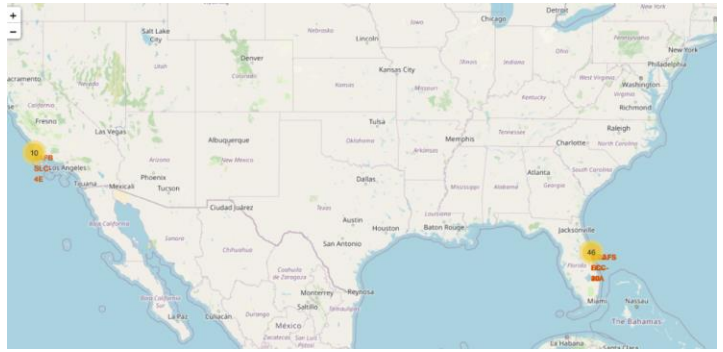
- There are 4 launch sites used by spacex
- The average payload mass for booster version F9 V.1.1 is 2928kg
- There was 100 successful outcomes in the data set and only 1 that failed
- There is no significant correlation between flight number and launch site for mission outcome as well flight number and launch site
- ESL, GEO, HEO, SSO have the highest success rate of nearly one , SO has the lowest with 0
- 2018 was the peak for the success rate of missions, with 0% for all years before 2010

Interactive analytics demo in screenshots



- KSC-LC-39A had biggest Portion of successful launches, It also has the biggest portion of successful launches with 76.9% CCAFS SLC 40 has lowest with 57.1% successful



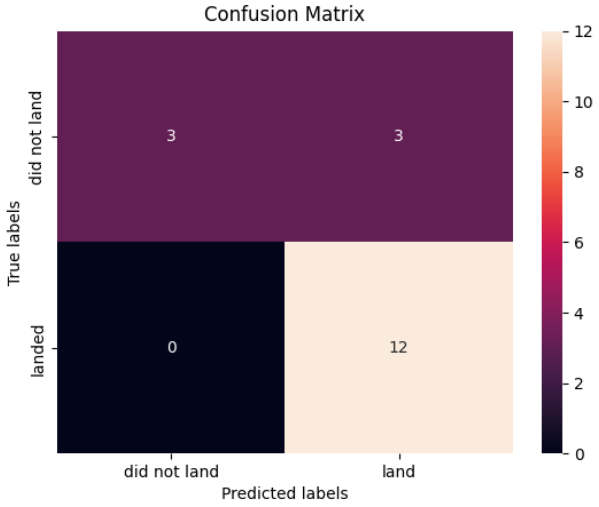


The folium maps show how many launches took off in each location and which were successful or not, This is another way to visual the results I state above

Predictive analysis results

- Results were all the same for logistic regression, support vector machine and knn
- Tree Classifier had the best accuracy out of the 4 models, based off accuracy and confusion matrix

```
0.8333333333333334
```

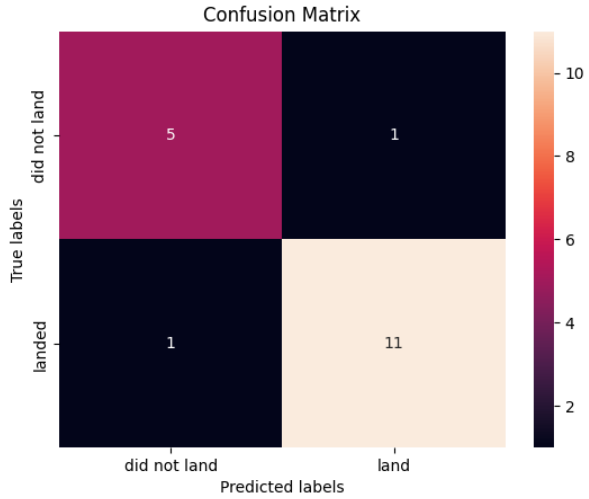



```
In [95]: tree_cv.score(X_test,Y_test)

Out[95]: 0.8888888888888888
```

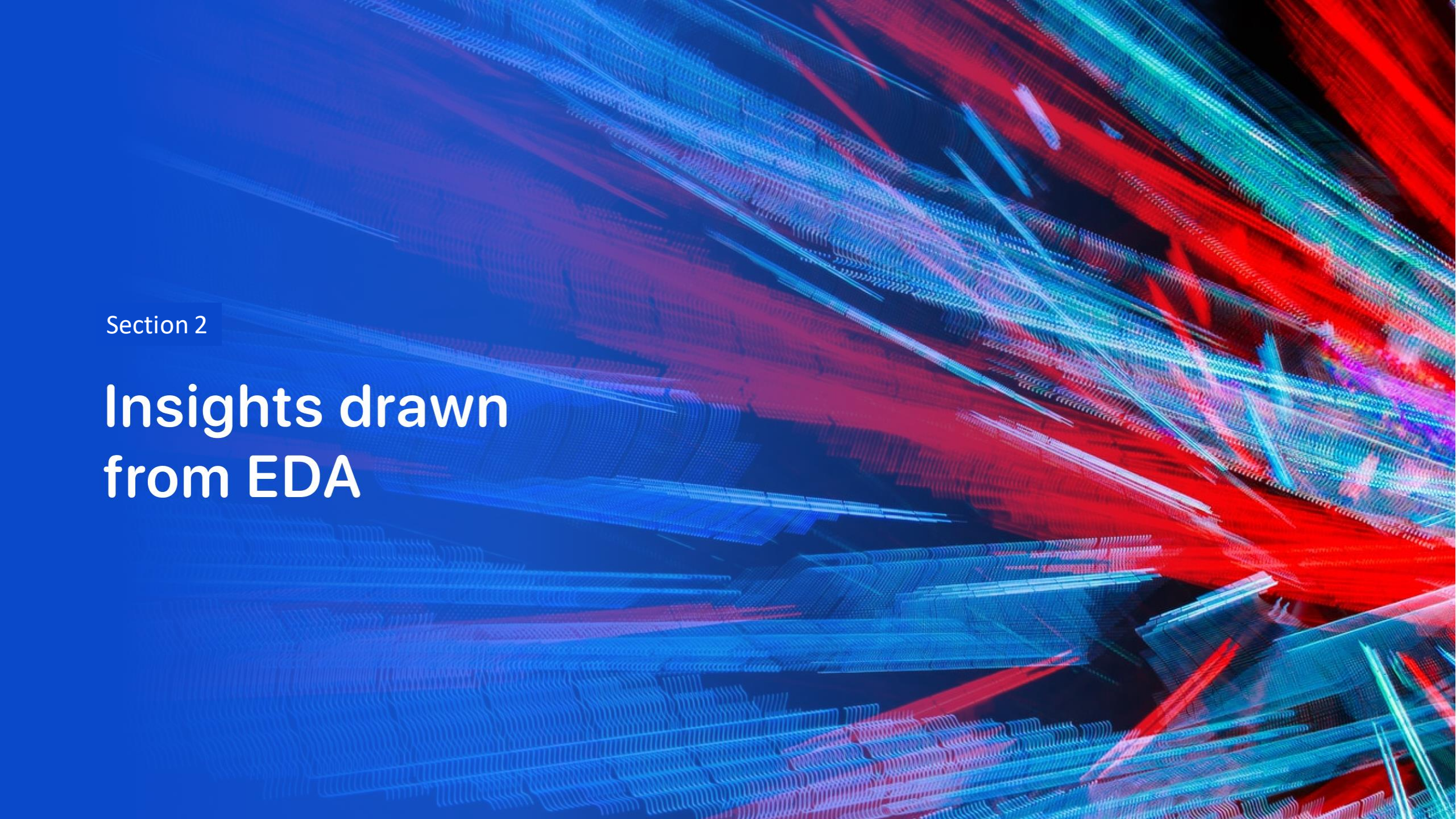
We can plot the confusion matrix

```
In [97]: yhat = tree_cv.predict(X_test)
         plot_confusion_matrix(Y_test,yhat)
```

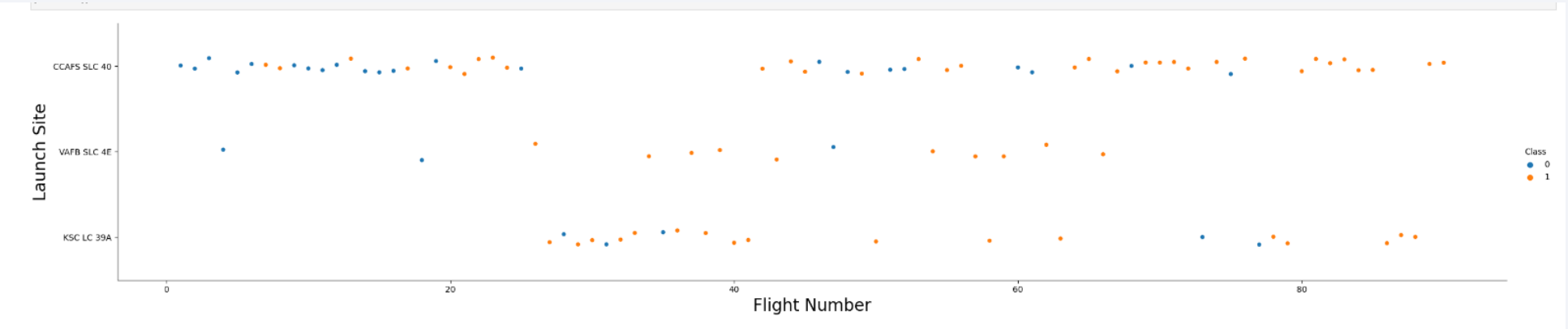


Section 2

Insights drawn from EDA

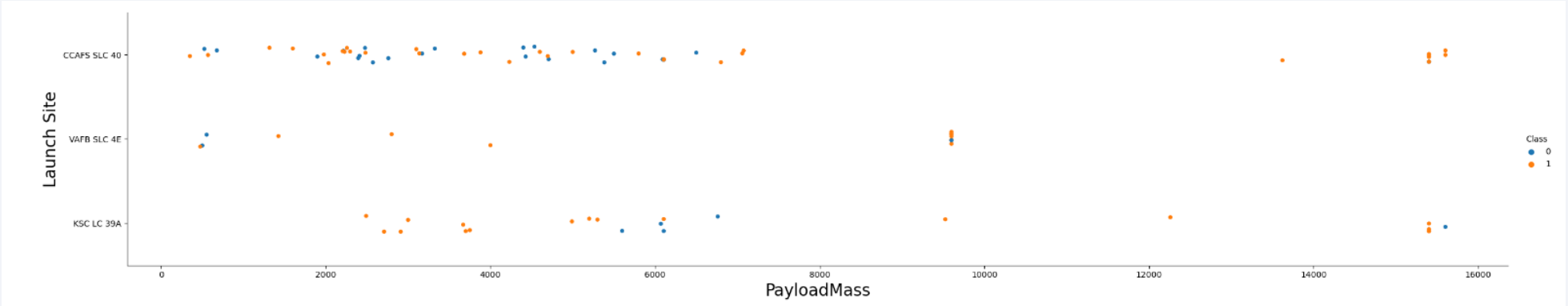


Flight Number vs. Launch Site



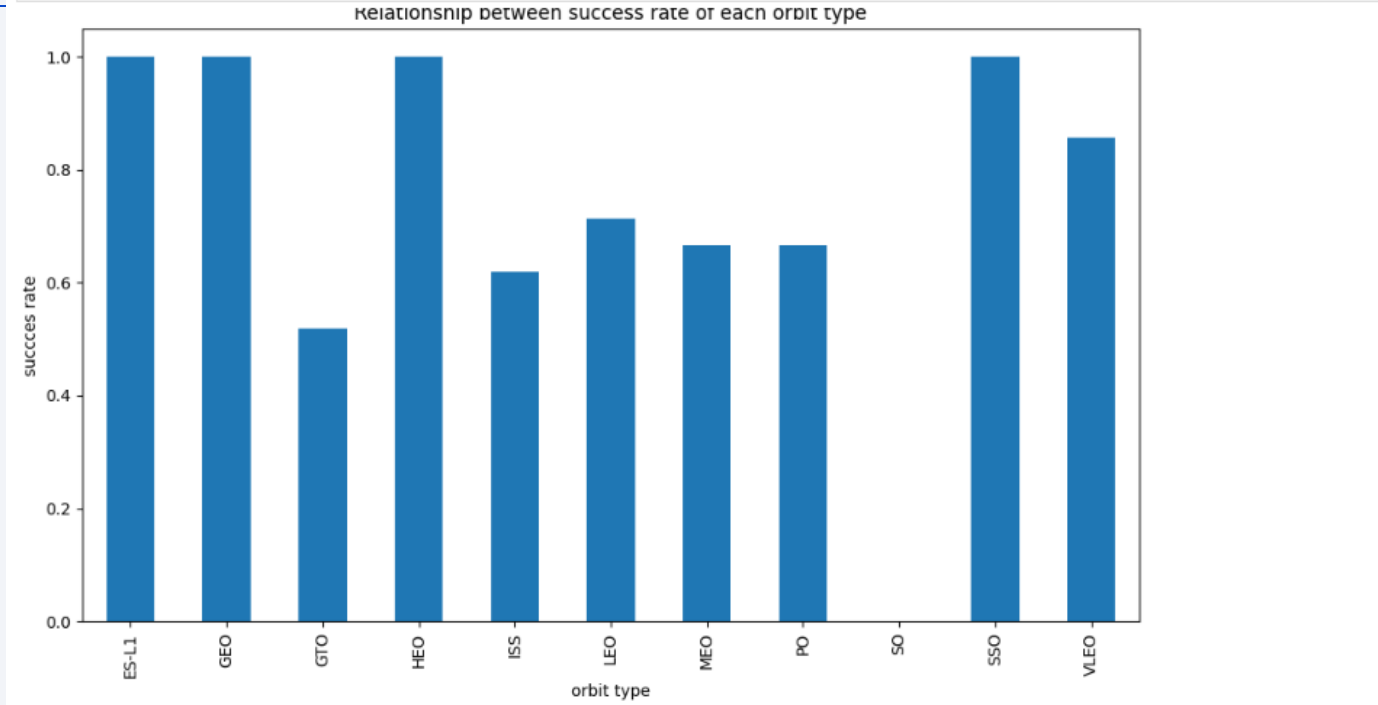
- CCAPS SLC 40 had the most launches
- As the flight number increased the more launches that were successful
- VAFB SLC 4E had the least launches
- KSC LC 39A had highest proportion of successful launches to unsuccessful launches

Payload vs. Launch Site



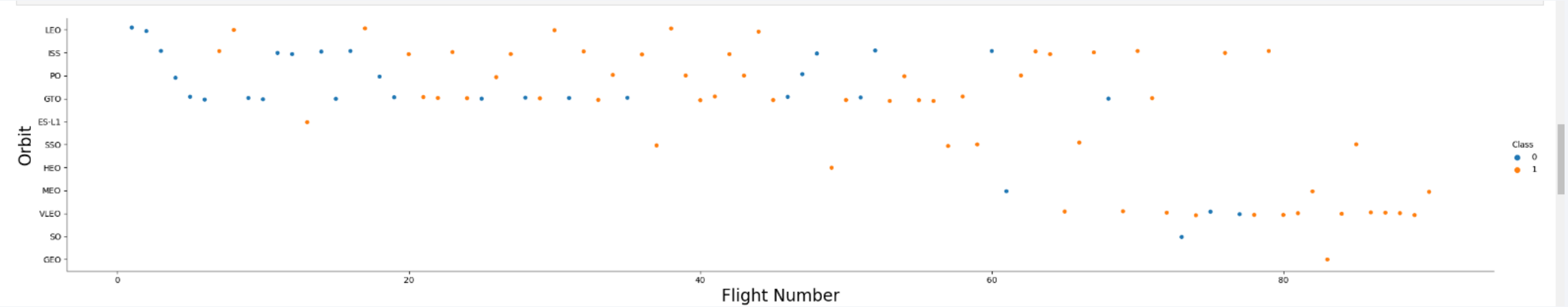
- All bar one over 10000kg were successful
- Most failures had between 30k and 60k kg
- CCAPS SLC 40 had most failures
- Most really low mass (under 30k) and really high mass (over 90k) were successful

Success Rate vs. Orbit Type



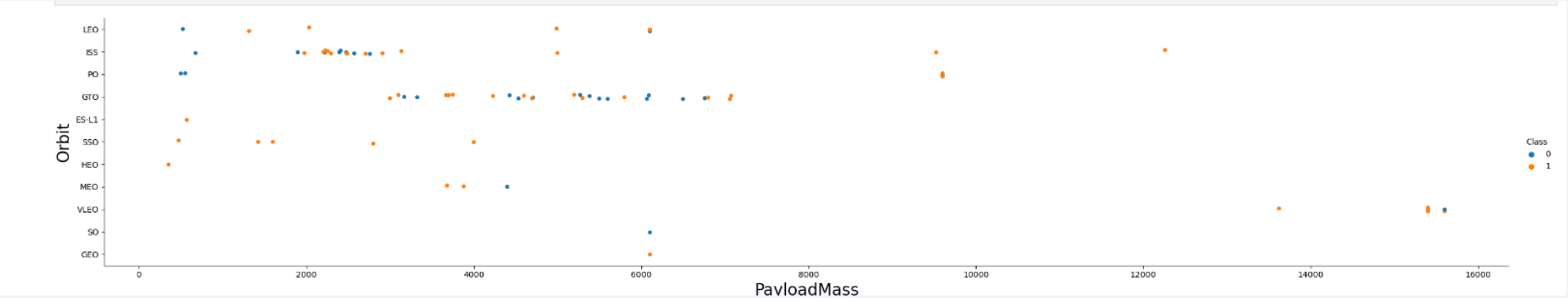
- ESL-1, GEO, HEO, SSO had highest success rate which is around 98%
- SO had lowest with 0%
- Most orbit types have success rate over 60% except for GTO and SSO

Flight Number vs. Orbit Type



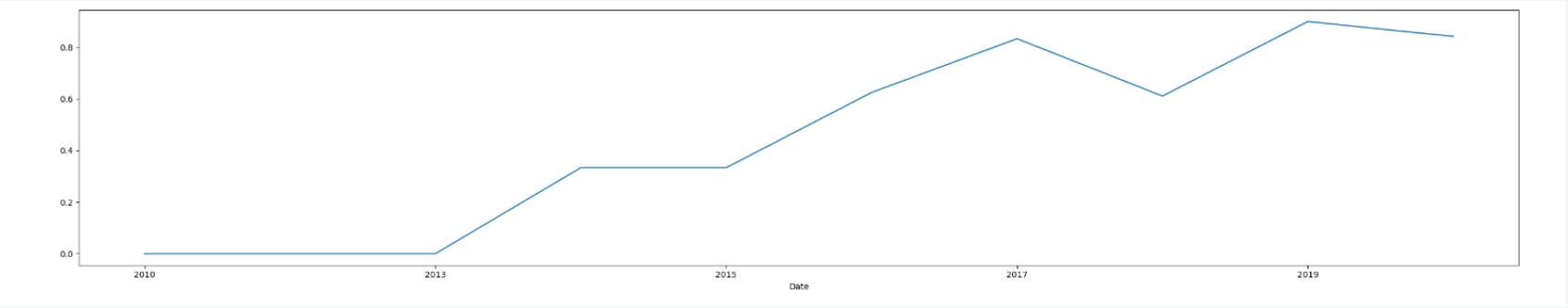
- As Flight number increases there is more successful landings for every orbit
- Gto Starts with 5 unsuccessful landings, but as flight number increases it gets more and more successes
- LEO starts with 2 unsuccessful, but the rest are successful
- Every orbit last flight was successful
- All flight numbers below 10 were failures
- Show the screenshot of the scatter plot with explanations

Payload vs. Orbit Type



- SO only had one launch and it failed, where geo had 1 and it was successful
- Nearly all launches over 100k were successful except one from VLEO
- Most launches under 10k were unsuccessful including all from PO, LEO AND ISS

Launch Success Yearly Trend



- All launches before 2013 were failures
- After 2013 there was a rise to around 30%
- Between 2015 and 2017 there was a really big rise from around 30% to 75%
- Before a slight decline in 2018 and then a high of 80% in 2019

All Launch Site Names

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

In [22]:
```
%sql select distinct launch_site from SPACEXTBL
```

```
 * ibm_db_sa://lsj12633:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
   sqlite:///my_data1.db
Done.
```

Out[22]:

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

- This simple query listed the names all of sites, which are shown above

Launch Site Names Begin with 'CCA'

Task 2

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

```
%sql select * from SPACEXTBL WHERE LAUNCH_SITE like 'CCA%' limit 5;
```

```
 * ibm_db_sa://lsj12633:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
   sqlite:///my_data1.db
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

Here is 5 sites that begin with CCA, which are all ccafs lc-40 site and it is clear from this the orbit type from this site is LEO

Total Payload Mass

Task 3

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

```
7]: %sql select sum(payload_mass__kg_) from SPACEXTBL where CUSTOMER like 'NASA (CRS)' ;
 * ibm_db_sa://lsj12633:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
   sqlite:///my_data1.db
Done.
7]:    1
       45596
```

- Using sum function and filtering for Nasa we could see total mass for all nasa launches

Average Payload Mass by F9 v1.1

Task 4

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

```
[31]: %sql select avg(payload_mass__kg_) from SPACEXTBL where booster_version like 'F9 v1.1' ;
```

```
 * ibm_db_sa://lsj12633:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
   sqlite:///my_data1.db
Done.
```

[31]:

1
2928

- Using avg function we were able to see the average sum for booster f9 v1.1 which wasn't that high at 2928kg

First Successful Ground Landing Date

```
[46]: %sql select DATE from spacextbl WHERE Mission_outcome like '%Success%' and landing__outcome like '%ground pad%'  order by DATE  limit 1;

 * ibm_db_sa://lsj12633:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
   sqlite:///my_data1.db
Done.
```

t[46]:

DATE
2015-12-22

- This statement shows that the first successful ground landing was in 2015, based off our previous EDA we saw that the number of successful landings started to increase massively in that year

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

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

In [50]:
```
%sql select distinct BOOSTER_VERSION from spacextbl where Mission_outcome like '%Success%' and landing__outcome like '%drone ship%' and payload_mass__
```

```
 * ibm_db_sa://lsj12633:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
   sqlite:///my_data1.db
Done.
```

Out[50]:

booster_version
F9 FT B1021.2
F9 FT B1031.2
F9 FT B1020
F9 FT B1022
F9 FT B1026

- There is 5 booster versions than had successful drone ship landing between these payload masses, you can see them above

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql select count(*) from spacextbl where Mission_outcome like 'Success%';
```

```
 * ibm_db_sa://lsj12633:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
   sqlite:///my_data1.db
Done.
```

1
100

```
%sql select count(*) from spacextbl where Mission_outcome like 'Failure%';
```

```
 * ibm_db_sa://lsj12633:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
   sqlite:///my_data1.db
Done.
```

1
1

- There was 100 success missions and only one that was a failure, this tells us that the majority of launches in this dataset were successful.

Boosters Carried Maximum Payload

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

In [62]:
```
%sql select distinct booster_Version from spacextbl where payload_mass__kg_ = (select max(payload_mass__kg_) from spacextbl);
```

```
 * ibm_db_sa://lsj12633:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
   sqlite:///my_data1.db
Done.
```

Out[62]:

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

- Above the query shows that there is 12 booster versions that have carried the max payload

2015 Launch Records

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

In [82]:
```
%sql select MonthName(DATE), LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_sITE from spacextbl WHERE YEAR(DATE) = 2015 AND LANDING__OUTCOME = 'Failure (dr
```

```
 * ibm_db_sa://lsj12633:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
   sqlite:///my_data1.db
Done.
```

Out[82]:

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

- There is 2 landings that failed by drone ship in 2015 one in january and one in april and they both came from the same launch site in ccafs lc-40

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

Task 10

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

In [97]:
```
%sql select landing__outcome ,count(*) from spacextbl WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' and landing__outcome like 'Success%'  group by
```

```
 * ibm_db_sa://lsj12633:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
   sqlite:///my_data1.db
Done.
```

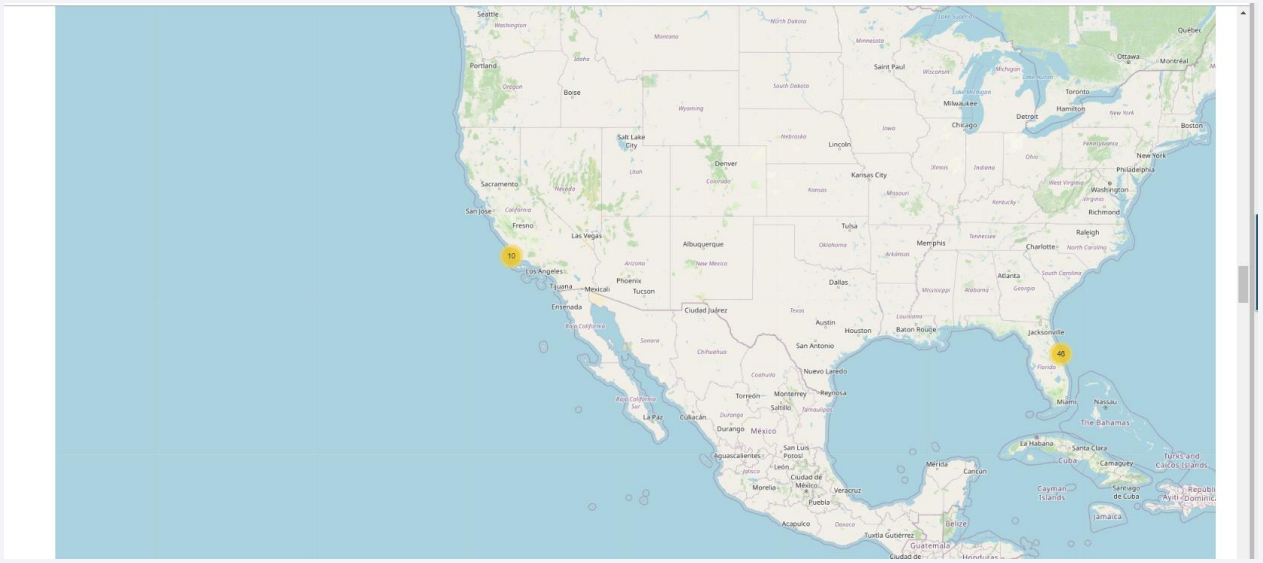
Out[97]:

landing__outcome	2
Success (drone ship)	5
Success (ground pad)	3

- There was 5 successful drone landings and 2 successful ground pad landings between june 2010 and march 2017


Section 3
Launch Sites
Proximities Analysis

Launch Sites location markers



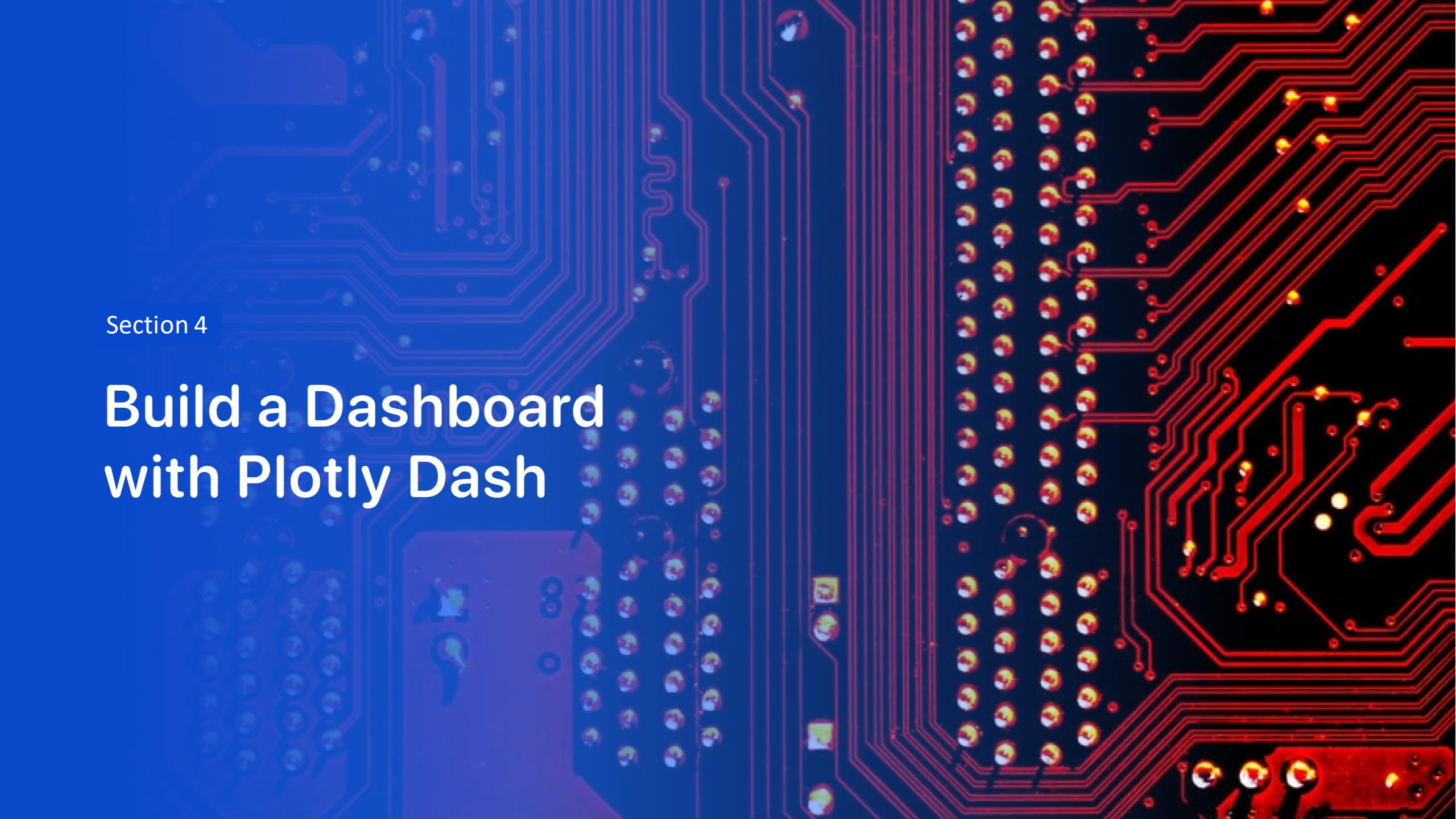
- This screenshots show that 10 launches took place on the west coast and the large majority of launches with 46 took place on the east coast.

Colour coded labeled launch outcomes

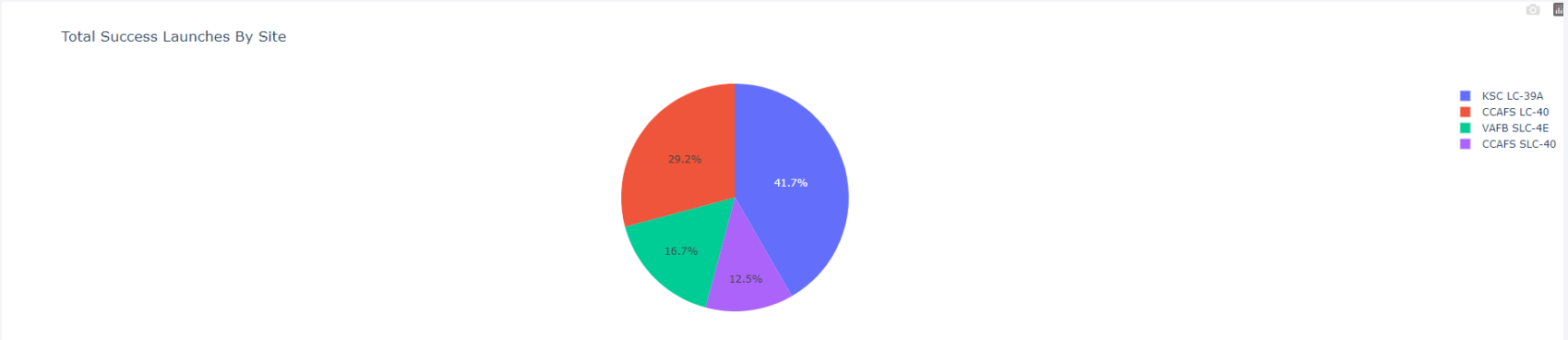
- This screenshot shows one of the launch sites, zoom and clicked in, green shows it is successful launch and red unsuccessful. It is clear from this particular screenshot this launch site had more unsuccessful launches than successful.

<Folium Map Screenshot 3>

- Replace <Folium map screenshot 3> title with an appropriate title
- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed
- Explain the important elements and findings on the screenshot

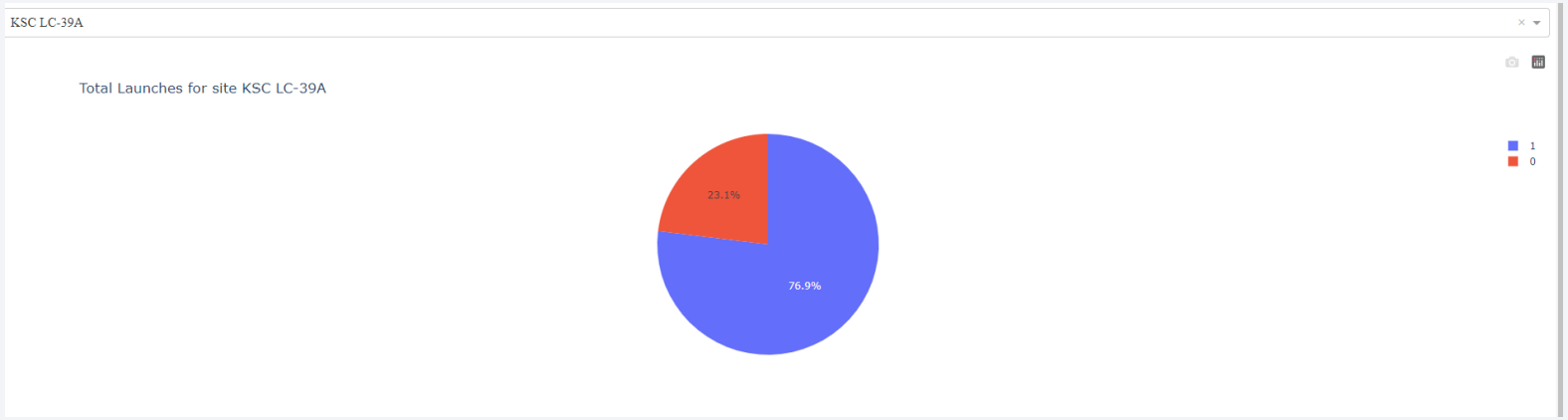

Section 4
Build a Dashboard with Plotly Dash

All sites Pie chart



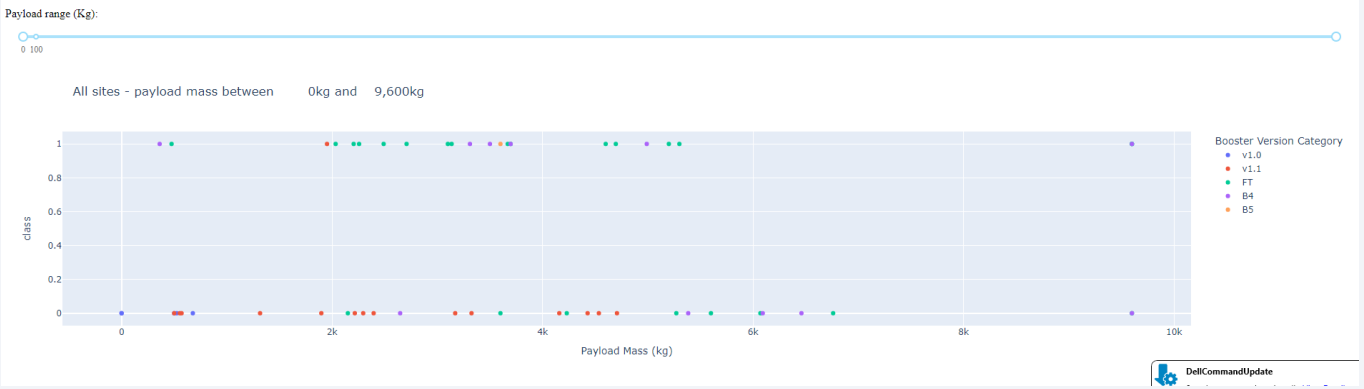
- As the pie chart shows, KSC LC-39A is the launch site with the highest proportion of successful launches with 41.7%, whereas CCAFS SLC-40 has the lowest with 12.5%

Launch site with highest launch success ratio



- As well as having have the most proportion of all successful launches, KSC LC-39A has the highest proportion between successful and unsuccessful launches with 76.6% successful, the next closest is CCAFS LC-40 with 73.1%

Launch Vs Payload Outcome scatter plot



- FT was the most successful launcher
- V 1.1 was the most unsuccessful launcher
- There was more unsuccessful launches than successful in this screenshot
- Most launches were under 6k

Section 5

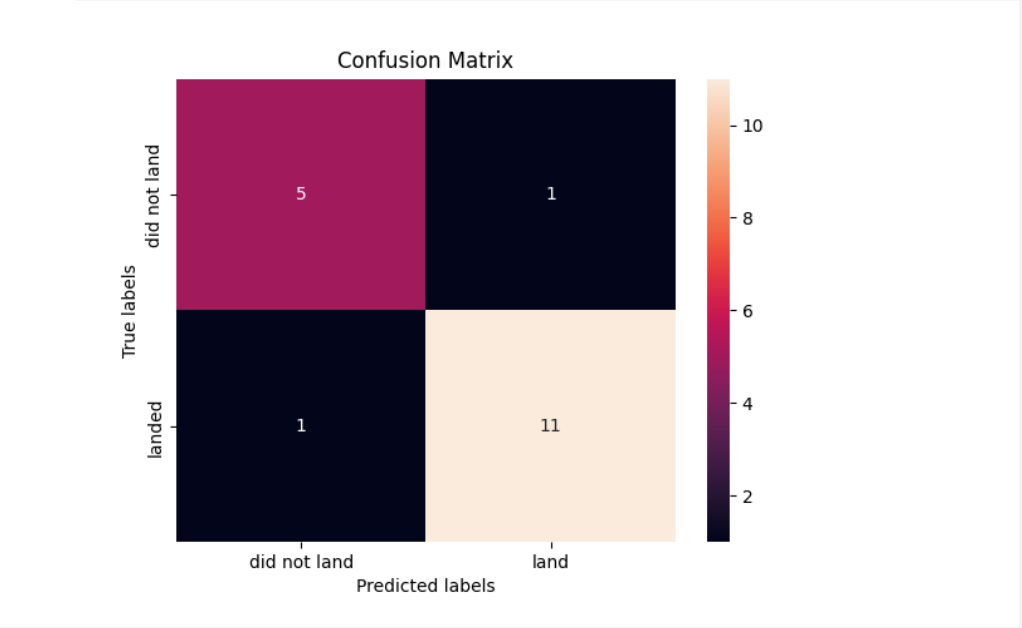
Predictive Analysis (Classification)

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



- This is the confusion Matrix for the tree classifier, it is better than all the other ones because it is has less false positives, with only 1 whereas the other 3 classifiers had 3 false positives. Although this one has 1 false negative and the other 3 had 0, the model is more accurate because of the less false positives, therefore making in the best model to predict if a mission will be success or not

Conclusions

- As time went on the more successful launches there was
- The most successful launches seem to be the ones with low load mass or ones with very high mass
- The most successful launch site was KSC LC-39A
- Tree Classifier was the best classifier because of the best accuracy and the lowest false positives

Appendix

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
outcomes=[]
for i,outcome in enumerate(landing_outcomes):
    if outcome in bad_outcomes:
        outcomes.append(0)
    else:
        outcomes.append(1)

print(outcomes)

landing_class=outcomes
```

```
# Calculate the mean value of PayloadMass column
mean=data_falcon9['PayloadMass'].mean()
mean
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass']= data_falcon9['PayloadMass'].replace(to_replace= np.nan, value= mean)
```

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

```
]: %sql select landing__outcome ,count(*) from spacextbl WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' and landing__outcome like 'Success%'  group by

 * ibm_db_sa://lsj12633:***@21fecfd8-47b7-4937-840d-d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31864/BLUDB
   sqlite:///my_data1.db
```

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
outcomes=[]
for i,outcome in enumerate(landing_outcomes):
    if outcome in bad_outcomes:
        outcomes.append(0)
    else:
        outcomes.append(1)

print(outcomes)

landing_class=outcomes
```

- Above I show a couple of code snippets that took of lot of work to get , both python and sql

Thank you!