IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Luțu George-Theodor
01.04.2022

Outline

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

Executive Summary

- Summary of methodologies
  - Collected data through SpaceX API and web scraping
  - Exploratory Data Analysis, Data Wrangling, Data Visualization, Interactive Visual, Analytics, Machine Learning Prediction
- Summary of all results
  - Collection of valuable and complex data
  - Using Exploratory Data Analysis the best features that predict success of landings were indentified
  - With Machine Learning the essential characteristics were predicted for the optimal outcome

Introduction

- The main purpose of the project is to evaluate the viability of a new company Space Y to surpass Space X's performance.
- Problems that need answers:
  - The best place to launch rockets
  - Using the collected data, to estimate the total cost for launches and predict their success of landing back

Section 1

Methodology

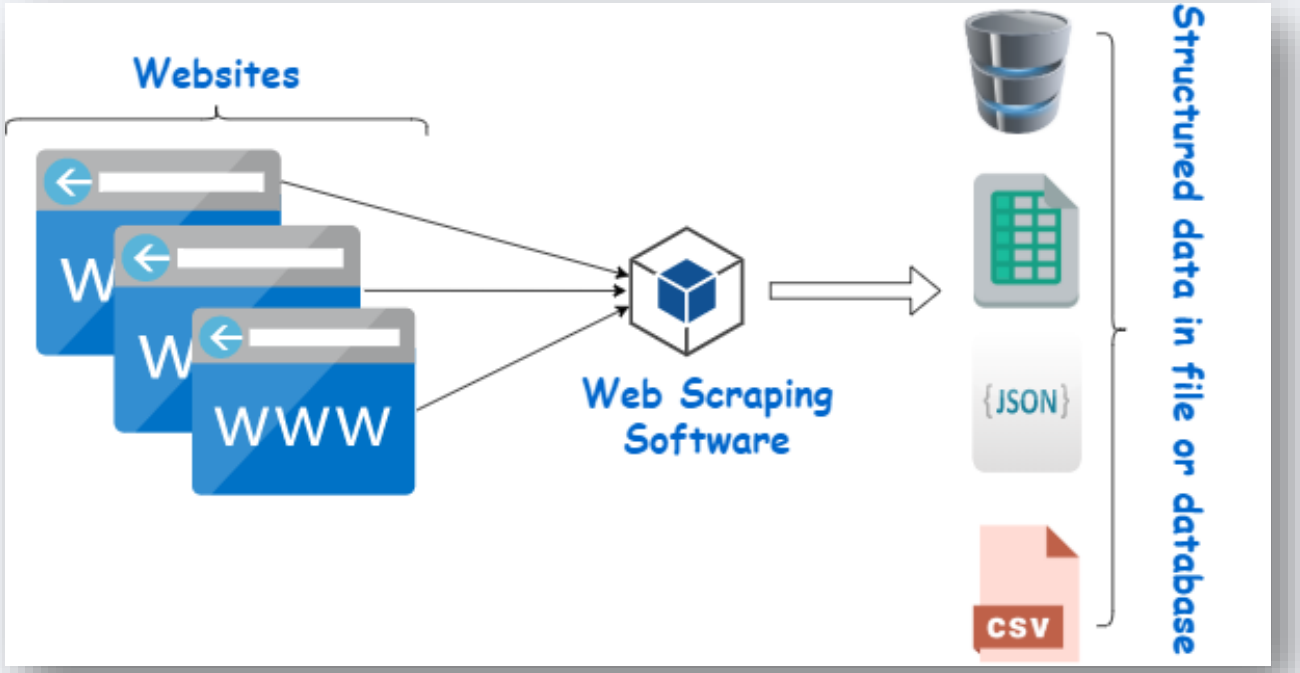
Methodology

Executive Summary

- Data collection methodology:
  - Space X API
  - Web Scraping
- Perform data wrangling
  - After summarizing and analyzing all the important features, the data was enhanced by creating a landing outcome label
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Normalization, Train/Test data, Evaluation

Data Collection

- Data sets were collected through 2 methods:
  - Space X API https://api.spacexdata.com/v4/rockets/
  - Web Scraping https://en.wikipedia.org/wiki/List_of_Falcon/_9/_and_Falcon_Heavy_launches



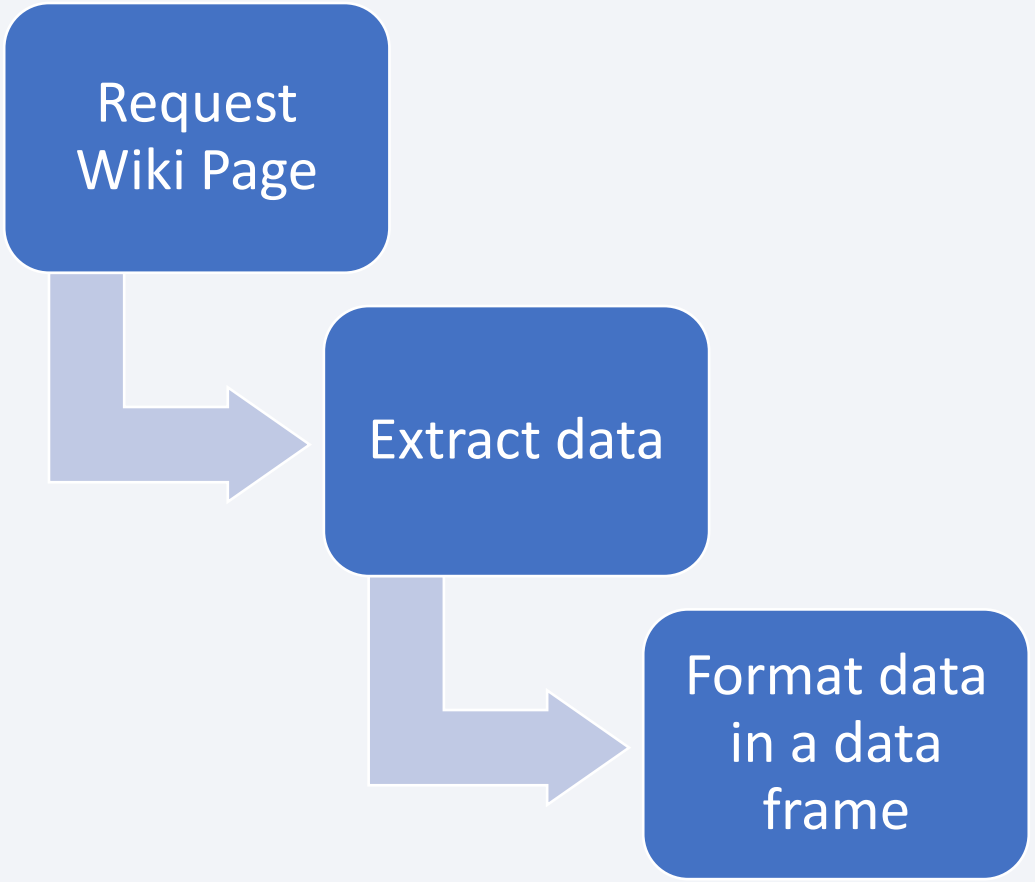
Data Collection – SpaceX API

- The data was obtained through a public API offered by SpaceX
- https://github.com/GeorgeLutu/Applied_data_science_capstone/blob/master/Data%20Collection%20API.ipynb

Data Collection - Scraping

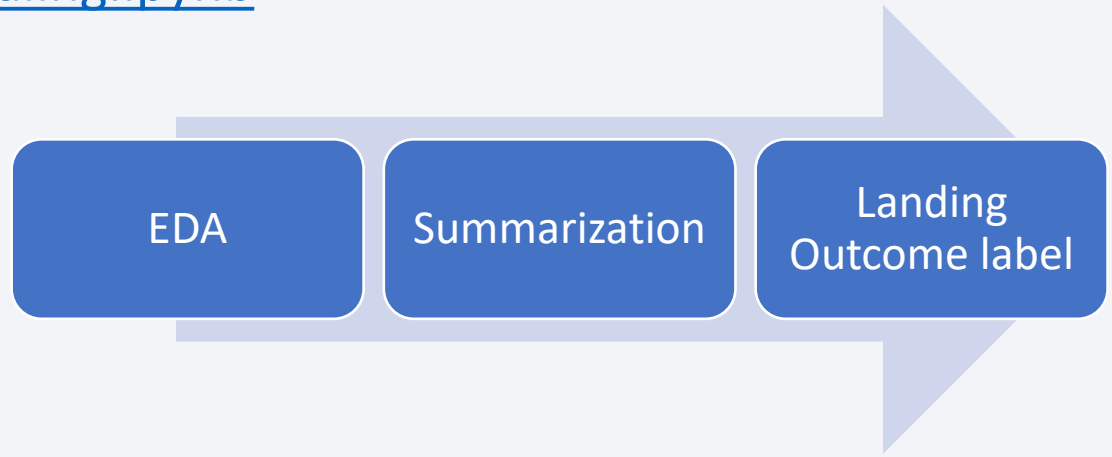
- Using the free site, Wikipedia, data has been web scraped.

- https://github.com/GeorgeLutu/Applied_data_science_capstone/blob/master/Data%20Collection%20Web%20Scraping.ipynb



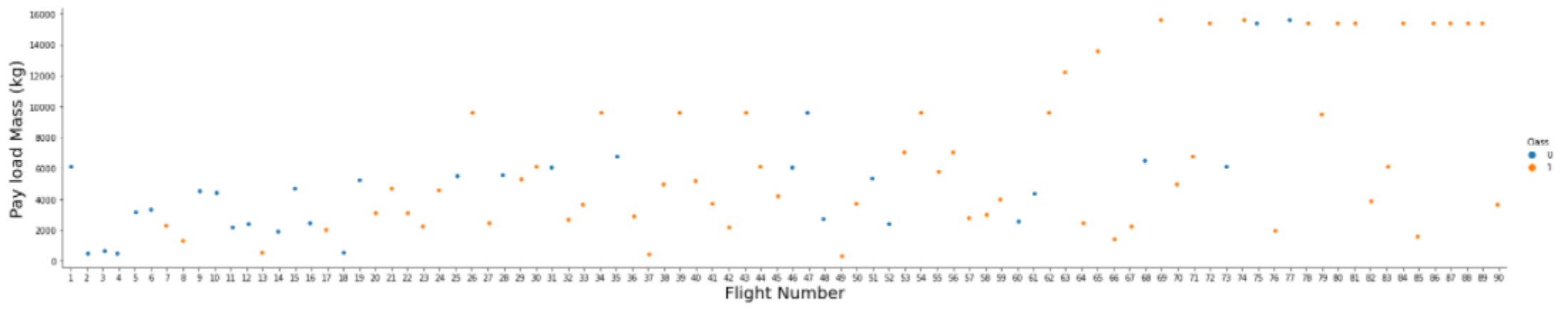
Data Wrangling

- Firstly, Exploratory Data Analysis was used on the dataset
- Proceeded to calculate launches per site, occurrences of each orbit and mission outcome
- In the end, from “Outcome” column the landing outcome label was created.
- https://github.com/GeorgeLutu/Applied_data_science_capstone/blob/master/Data%20Wrangaling.ipynb



EDA with Data Visualization

- In order for the data to be explored, various types of graphs were used, like scatterplots and bar plots
- https://github.com/GeorgeLutu/Applied_data_science_capstone/blob/master/EDA%20Visualization.ipynb
- Example:



EDA with SQL

- A number of 10 SQL queries were performed on the data that was collected:
  - Names of the unique launch sites in the space mission;
  - •Top 5 launch sites whose name begin with the string 'CCA';
  - •Total payload mass carried by boosters launched by NASA (CRS);
  - •Average payload mass carried by booster version F9 v1.1;
  - •Date when the first successful landing outcome in ground pad was achieved;
  - •Names of the boosters which have success in drone ship and have payload mass between 4000 and 6000 kg;
  - •Total number of successful and failure mission outcomes;
  - •Names of the booster versions which have carried the maximum payload mass;
  - •Failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015; and
  - •Rank of the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20.
- https://github.com/GeorgeLutu/Applied_data_science_capstone/blob/master/EDA%20SQL.ipynb

Build an Interactive Map with Folium

- Objects created using Folium Maps:
  - Markers that indicate different launch sites
  - Circles highlighting areas around specific coordinates
  - Marker Clusters that show groups of events in different zones
  - Lines that indicate distances between specific coordinates
- https://github.com/GeorgeLutu/Applied_data_science_capstone/blob/master/Data%20Visualization%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

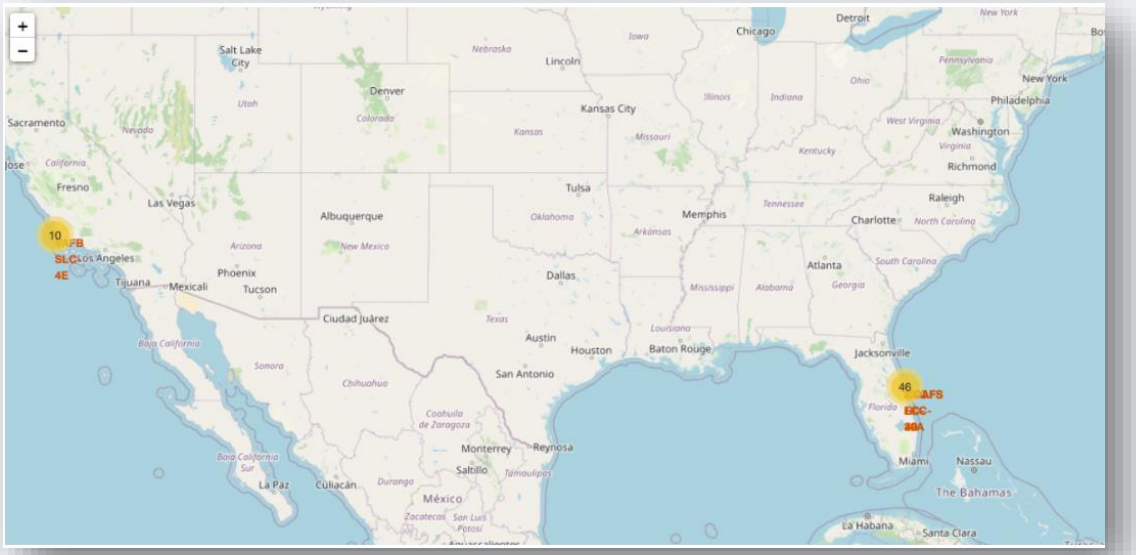
- In order to visualize data, the following graphs were ploted:
  - Percentage of launches by site
  - Payload range
- By doing so, the best place to launch was identified
- https://github.com/GeorgeLutu/Applied_data_science_capstone/blob/master/spacex_dash_app.py

Predictive Analysis (Classification)

- There were 4 classification models used: Logistic Regression, Support Vector Machine, Decision Tree, K-Nearest Neighbors
- The steps of the process: Data preparation and standardization, Testing of the models, Results comparison
- https://github.com/GeorgeLutu/Applied_data_science_capstone/blob/master/Machine%20Learning%20Prediction.ipynb

Results

- Exploratory data analysis results
  - There are 4 different launch sites
  - The average payload of F9 v1.1 booster is 2.928 kg
  - The first successful landing took place in 2015
  - The number of landing outcomes increased in success over the years
- Interactive analytics results



Results

- Predictive Analysis results
  - The best model to predict successful landing is Tree Classifier with accuracy at over 88%.

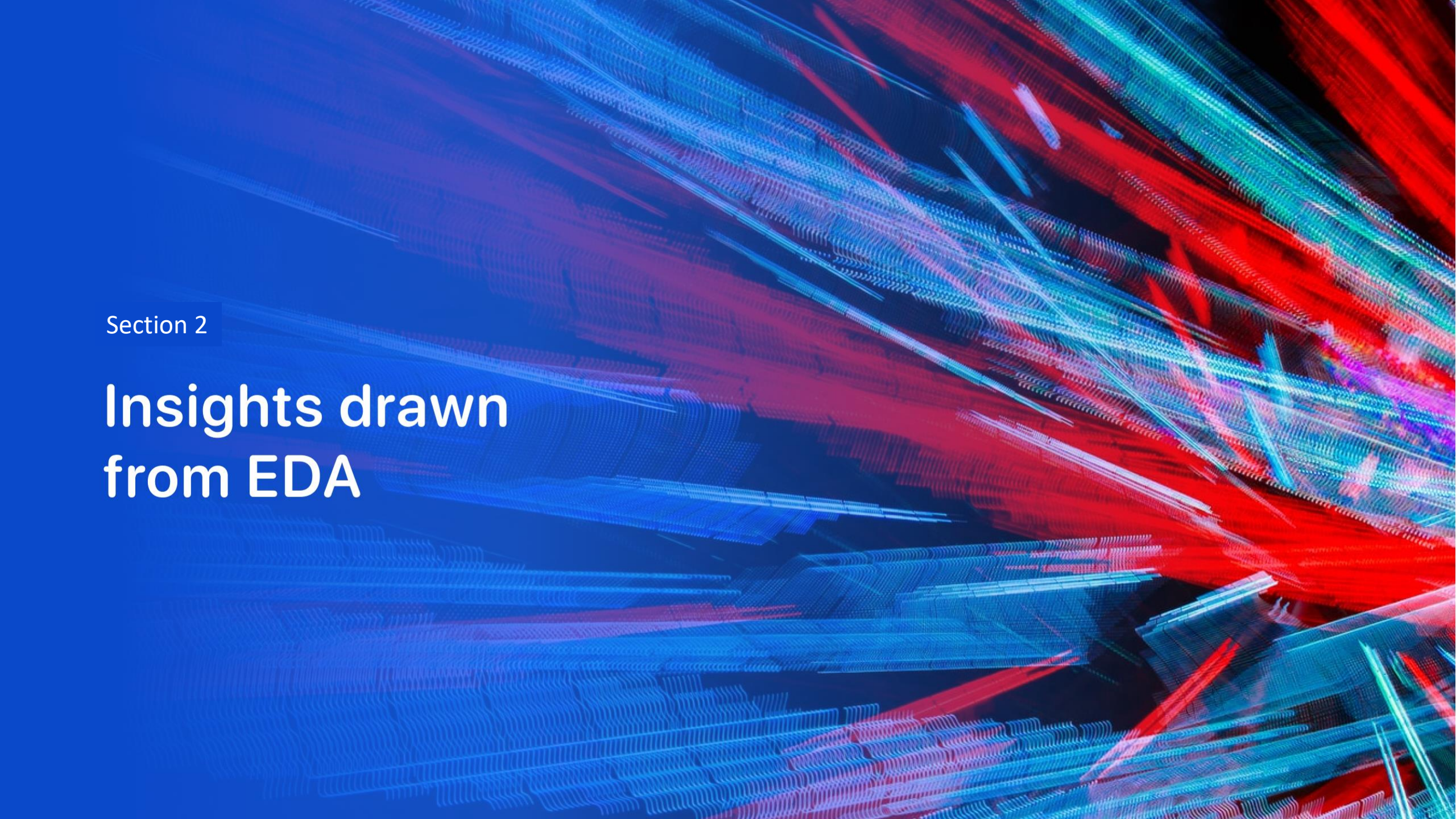
```
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
```

```
tree_cv=GridSearchCV(tree, parameters,cv=10)
tree_cv.fit(X_train,Y_train)
```

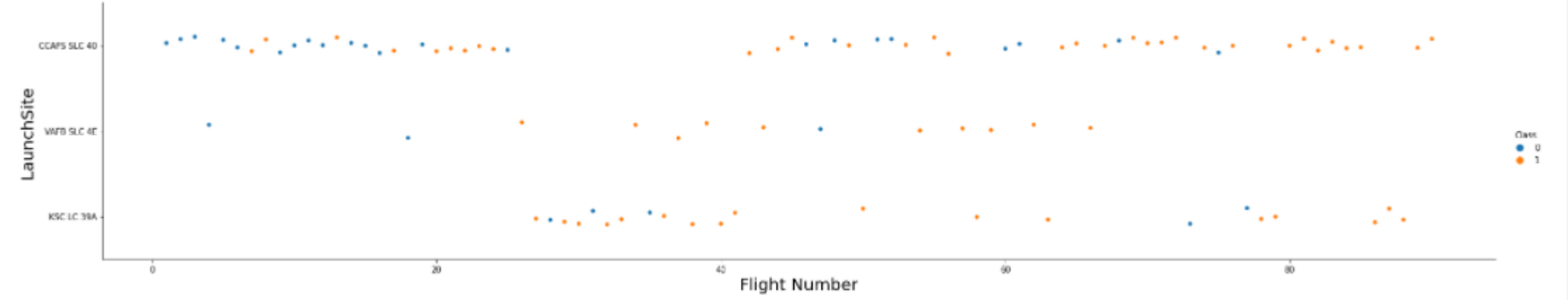
```
GridSearchCV(cv=10, estimator=DecisionTreeClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [2, 4, 6, 8, 10, 12, 14, 16, 18],
                         'max_features': ['auto', 'sqrt'],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [2, 5, 10],
                         'splitter': ['best', 'random']})
```

```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 4, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5,
'splitter': 'best'}
accuracy : 0.8892857142857142
```

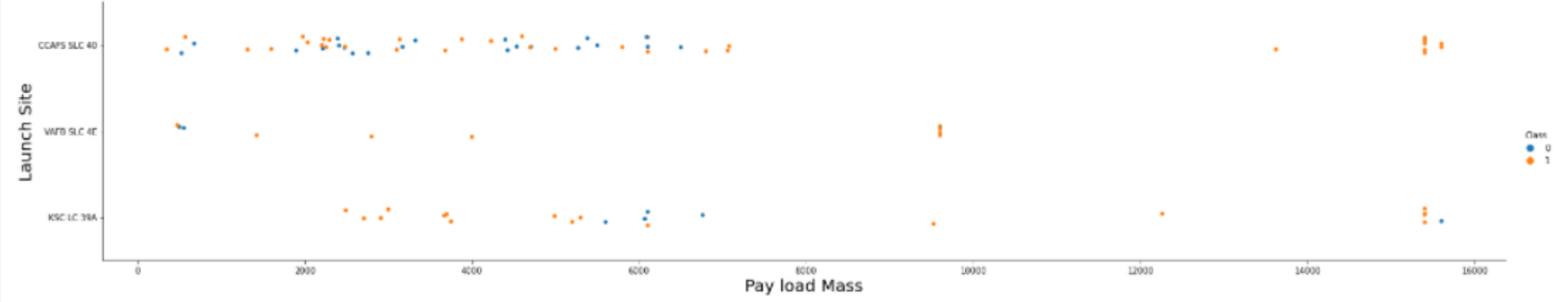

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



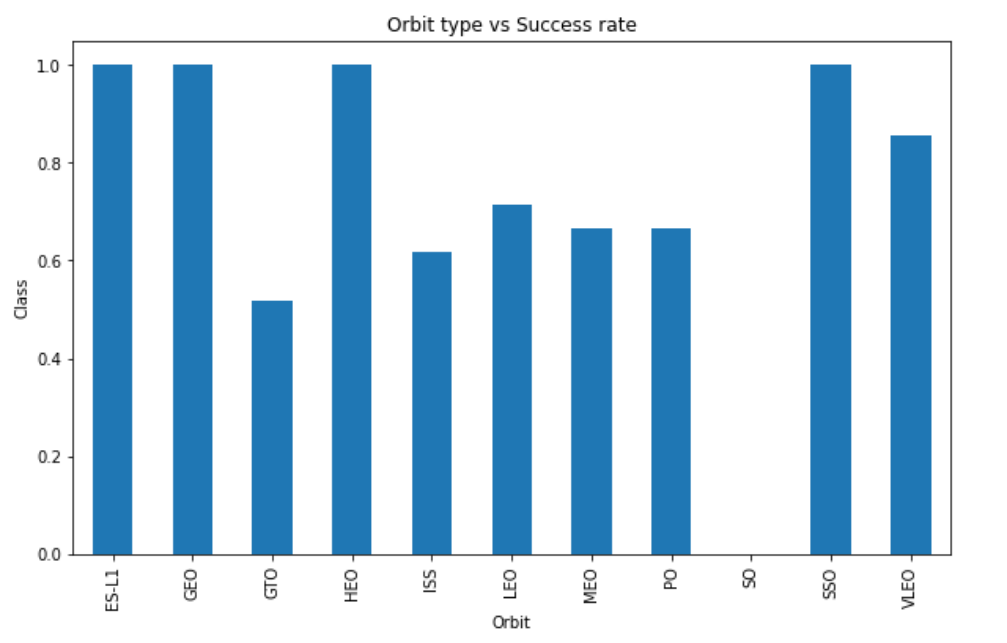
- Looking at the scatter plot above, the best launch site is CCAF5 SLC 40, where most launches were successful
- Also we can see that general success rate has improved over time.

Payload vs. Launch Site



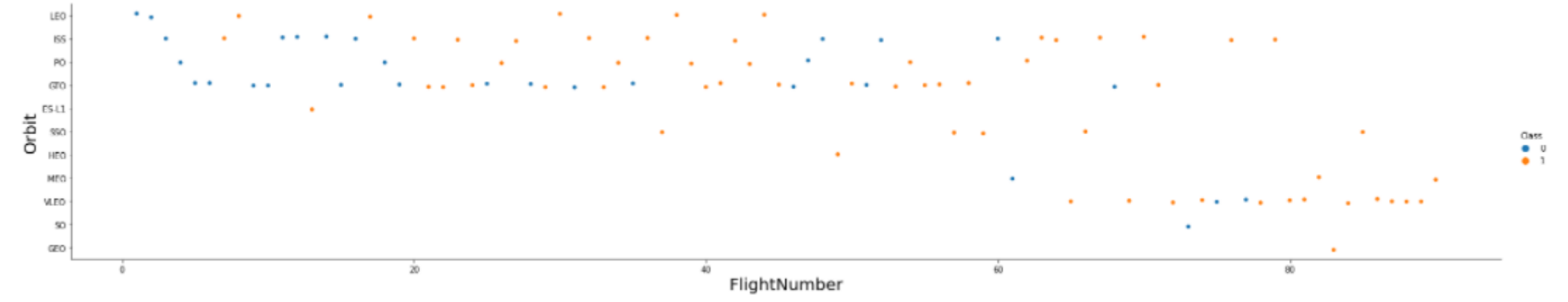
- Looking at the scatter plot above, payloads with mass over 9000 kg have good success rate
- Also with weight over 12000 kg payloads available are only CCAFS SLC 40 and KSC LC 39A

Success Rate vs. Orbit Type



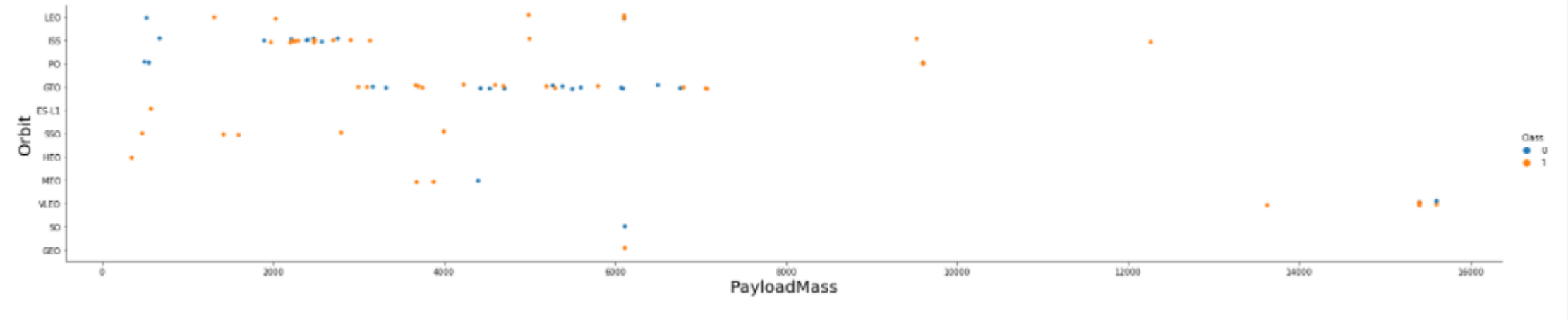
- Best success rate is seen in the following orbits:
  - ES-L1
  - GEO
  - HEO
  - SSO

Flight Number vs. Orbit Type



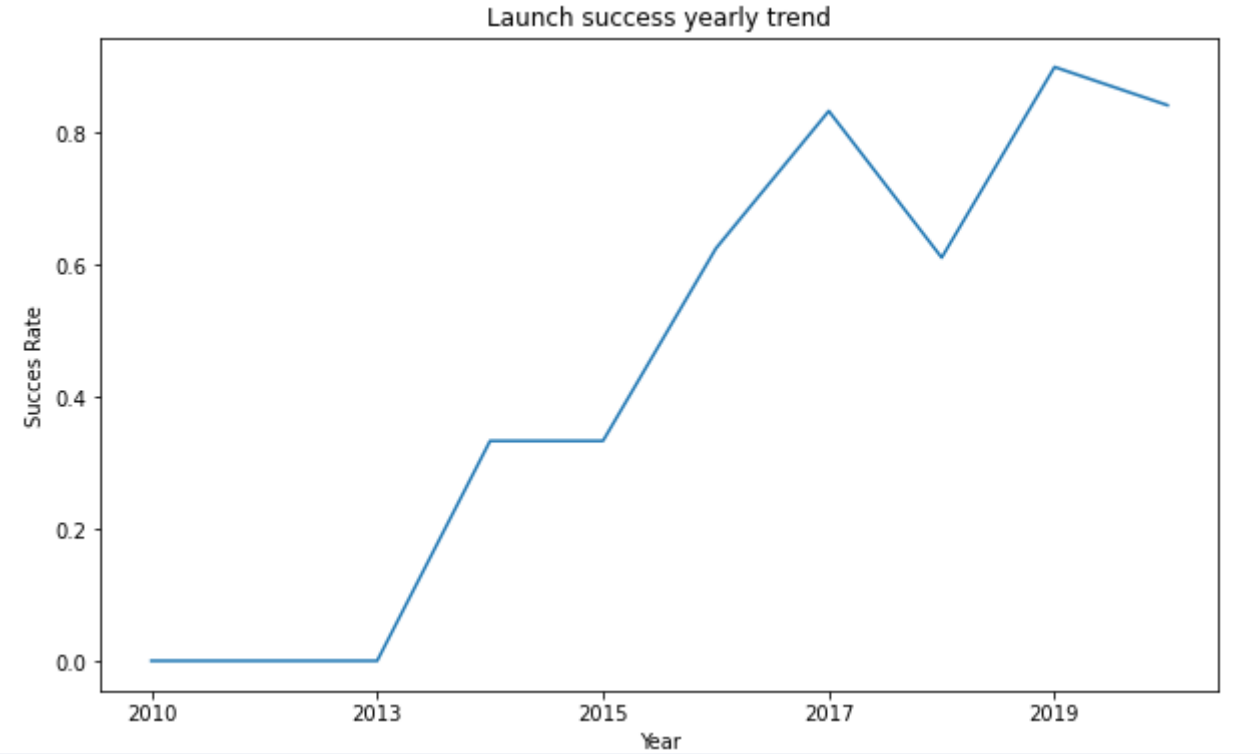
- Success rate improved over time to most orbits
- VLEO orbit is a new and frequently used orbit

Payload vs. Orbit Type



- ISS orbit has the widest range of payload mass and a good success rate
- There are fewer launches to SO and GEO compared to the others

Launch Success Yearly Trend



- Success rate started increasing meaningfully since 2013 and kept until around 2020

All Launch Site Names

- According to the collected data, there are 4 launch sites:
  - CCAFS LC-40
  - CCAFS SLC-40
  - KSC LC-39A
  - VAFB SLC-4E

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

```
%%sql
SELECT UNIQUE(LAUNCH_SITE) from SPACEXTABLE
```

```
 * ibm_db_sa://lcw28067:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/BLUDB
Done.
```

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

Launch Site Names Begin with 'CCA'

```
%%sql
SELECT * from SPACEXTABLE WHERE LAUNCH_SITE LIKE 'CCA%';
```

```
 * ibm_db_sa://lcw28067:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/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

Total Payload Mass

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) from SPACEXTABLE WHERE PAYLOAD LIKE '%CRS%'
```

```
 * ibm_db_sa://lcw28067:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/BLUDB
Done.
```

1
111268

Average Payload Mass by F9 v1.1

```
%%sql
Select AVG(PAYLOAD_MASS__KG_) from SPACEXTABLE WHERE BOOSTER_VERSION = 'F9 v1.1'
```

```
 * ibm_db_sa://lcw28067:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/BLUDB
Done.
```

1
2928

First Successful Ground Landing Date

```
%%sql
Select min(Date) from SPACEXTABLE where LANDING__OUTCOME = 'Success (ground pad)'
```

```
 * ibm_db_sa://lcw28067:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/BLUDB
Done.
```

1
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
%%sql
Select BOOSTER_VERSION from SPACEXTABLE where LANDING__OUTCOME = 'Success (drone ship)' and 400<PAYLOAD_MASS__KG_<6000
```

```
 * ibm_db_sa://lcw28067:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/BLUDB
Done.
```

booster_version
F9 B4 B1045.1

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%%sql
Select SUM(nr) from (SELECT COUNT(*) as nr from SPACEXTABLE GROUP BY LANDING__OUTCOME)
```

```
 * ibm_db_sa://lcw28067:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/BLUDB
Done.
```

1
101

Boosters Carried Maximum Payload

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

```
%%sql
Select BOOSTER_VERSION,PAYLOAD_MASS__KG_ from SPACEXTABLE where PAYLOAD_MASS__KG_= (SELECT MAX(PAYLOAD_MASS__KG_) from SPACEXTABLE )
```

```
 * ibm_db_sa://lcw28067:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/BLUDB
Done.
```

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

2015 Launch Records

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

```
%%sql
SELECT DATE,LANDING__OUTCOME,BOOSTER_VERSION,LAUNCH_SITE from SPACEXTABLE where LANDING__OUTCOME LIKE '%Failure%' and DATE like '2015%'
```

```
 * ibm_db_sa://lcw28067:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/BLUDB
Done.
```

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

```
%%sql
Select LANDING__OUTCOME,RANK() OVER (Order BY LANDING__OUTCOME DESC) as rank from SPACEXTABLE WHERE '2010-06-04'<DATE and DATE<'2017-03-20' GROUP BY LA
```

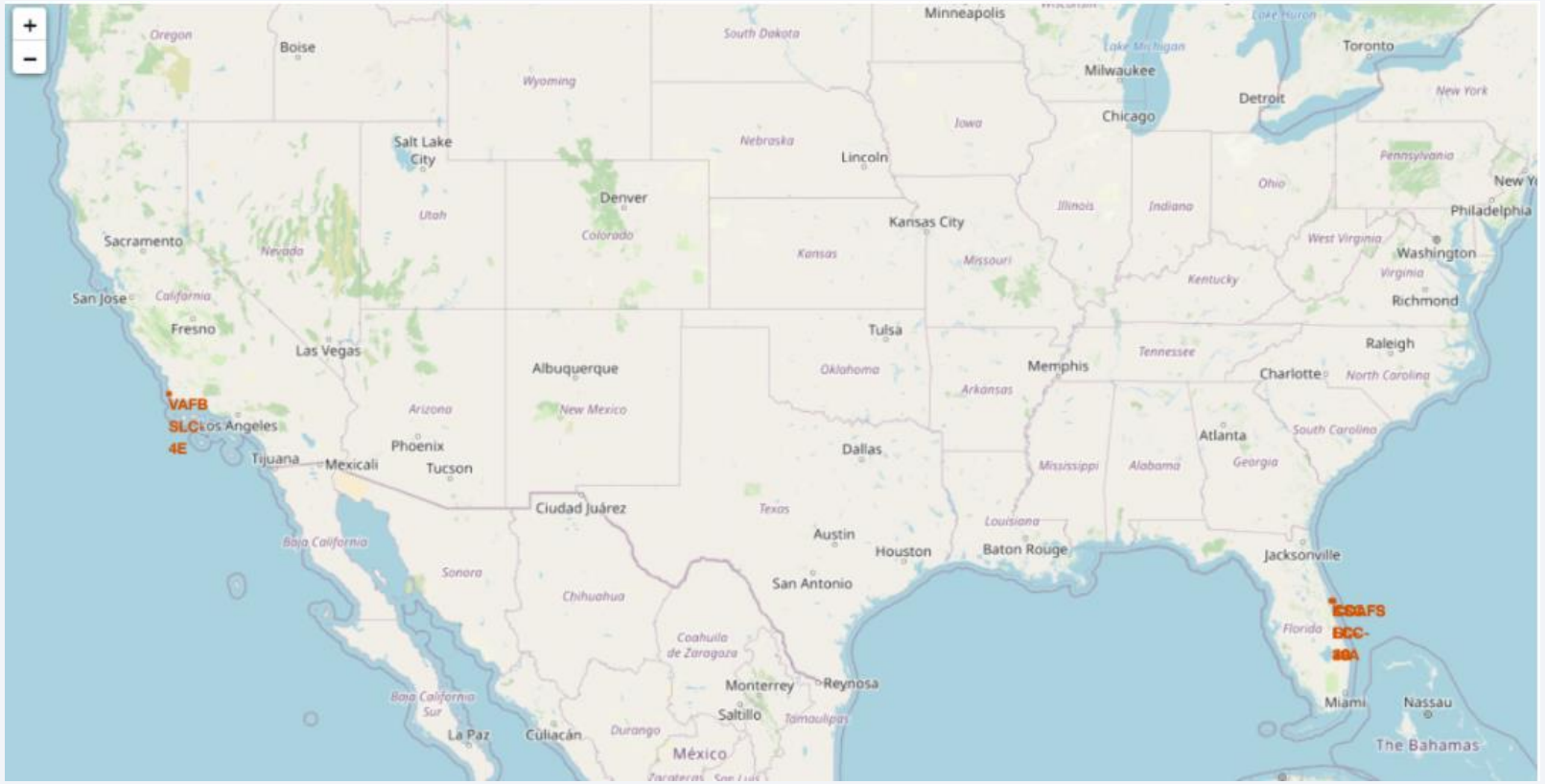
```
 * ibm_db_sa://lcw28067:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/BLUDB
Done.
```

landing__outcome	RANK
Uncontrolled (ocean)	1
Success (ground pad)	2
Success (drone ship)	3
Precluded (drone ship)	4
No attempt	5
Failure (parachute)	6
Failure (drone ship)	7
Controlled (ocean)	8

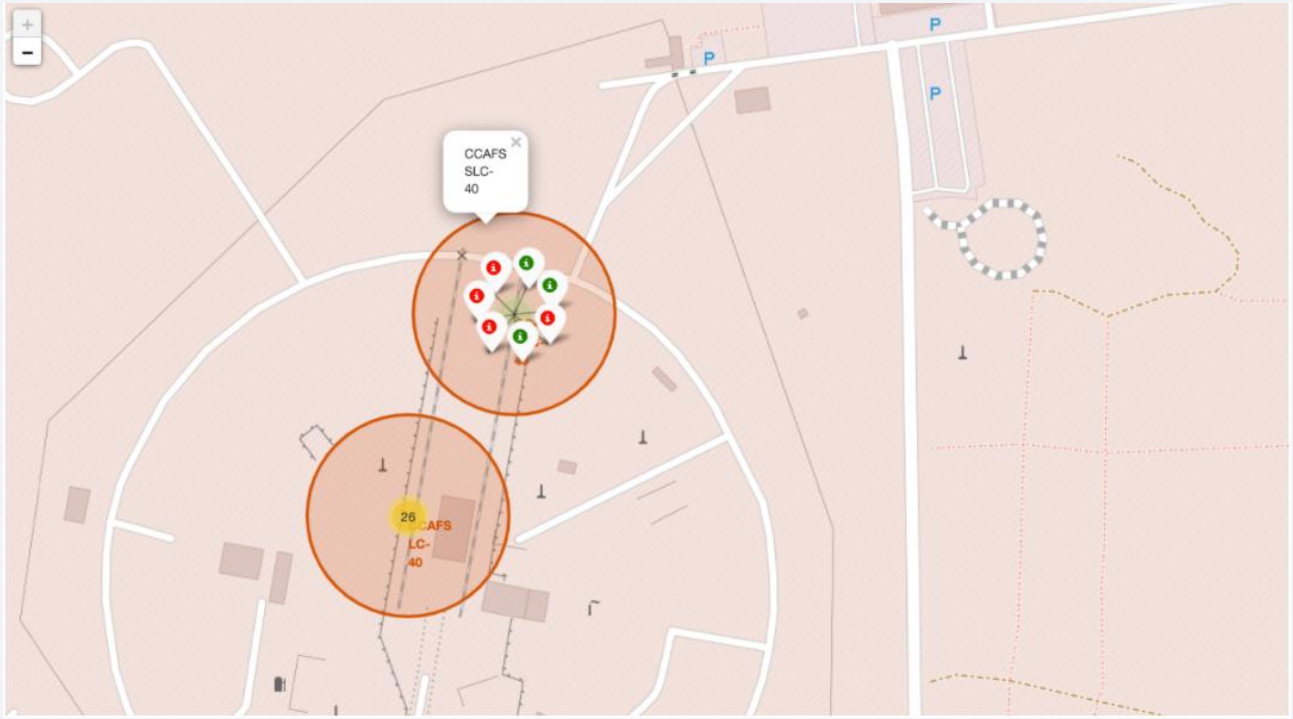
Section 3

Launch Sites Proximities Analysis

Launch sites spread

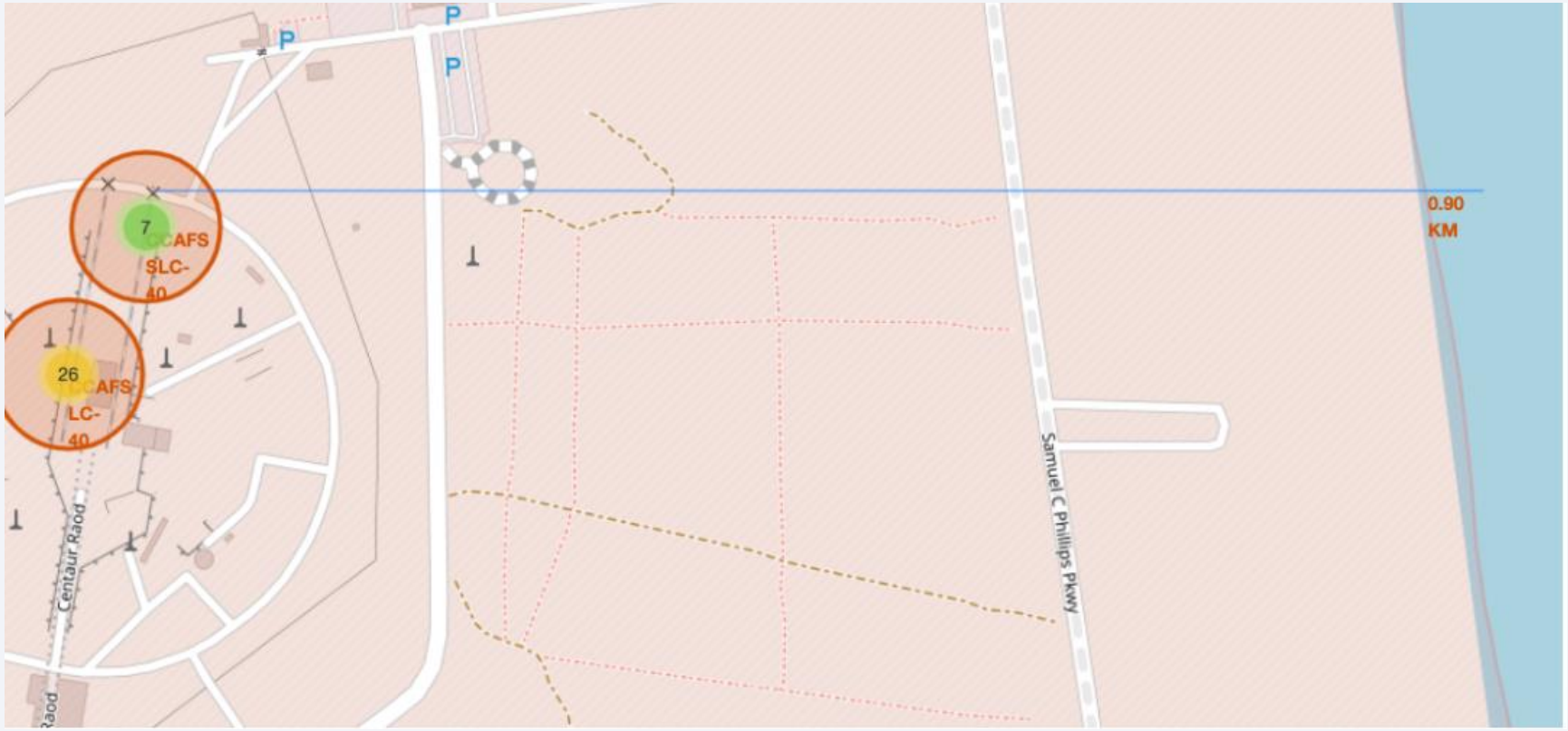


Launch Outcomes

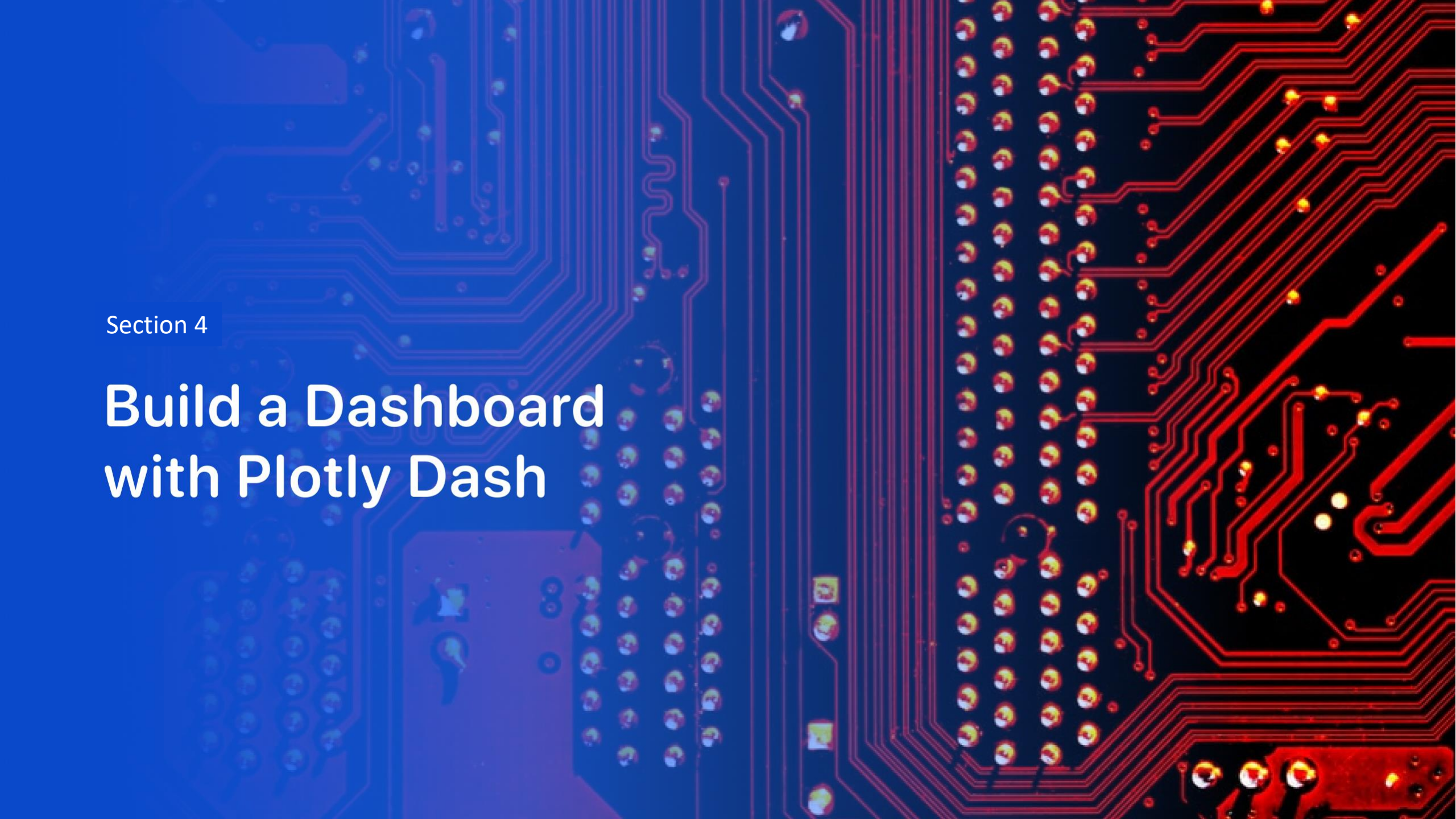


- Green markers represent successful launches and red ones indicate failed launches

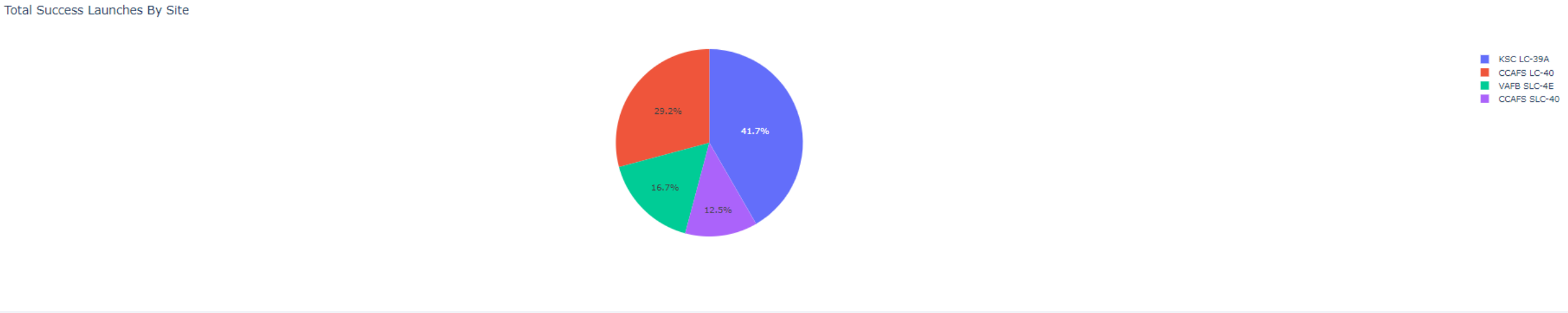
Safety



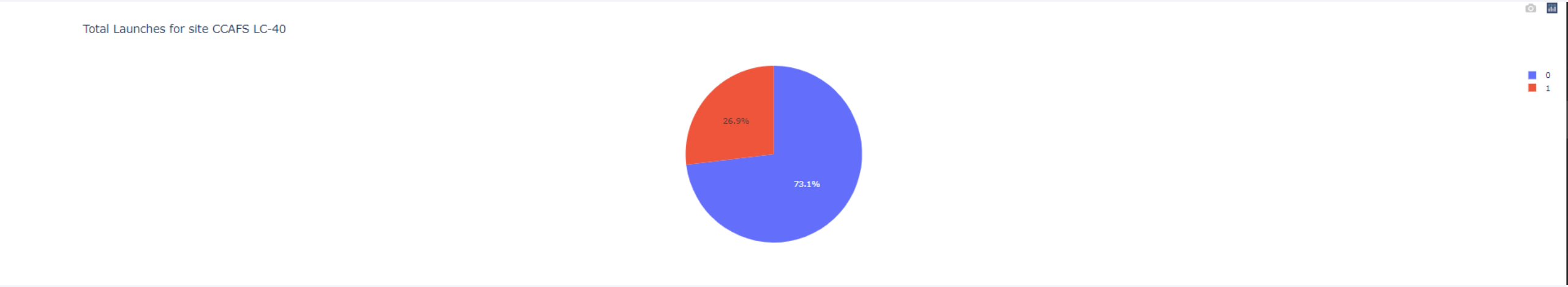
- The map shows how the launch site clusters are at least 0.9 km from the sea


Section 4
Build a Dashboard with Plotly Dash

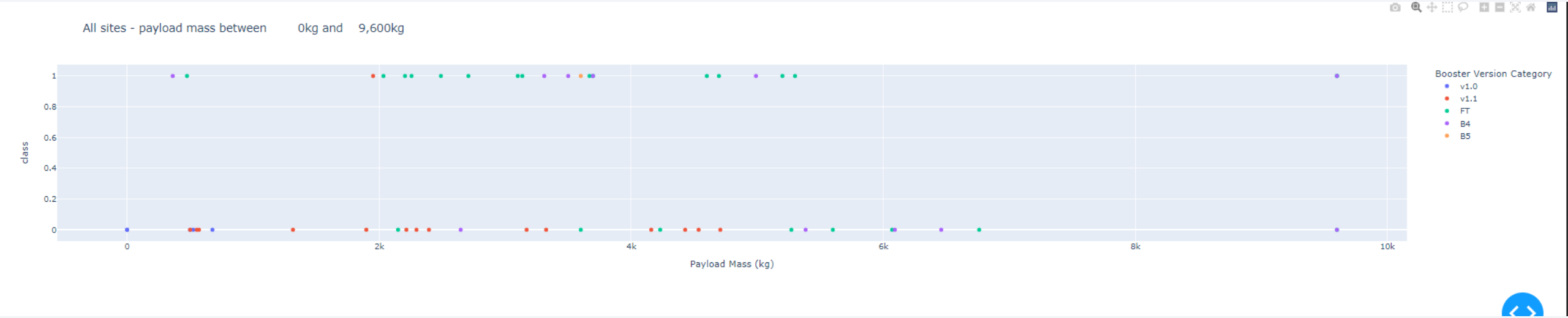
Total Succes by Launch site



Success launches for site CCAFS LC-40



Payload Mass vs. Outcome for All sites



Section 5

Predictive Analysis (Classification)

Classification Accuracy

- 4 classification models were used, and their accuracies are similar
- The model with the highest classification accuracy is the Decision Tree

TASK 8

Create a decision tree classifier object then create a `GridSearchCV` object `tree_cv` with cv = 10. Fit the object to find the best parameters from the dictionary `parameters`.

```
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
```

```
tree_cv=GridSearchCV(tree, parameters,cv=10)
tree_cv.fit(X_train,Y_train)
```

```
GridSearchCV(cv=10, estimator=DecisionTreeClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [2, 4, 6, 8, 10, 12, 14, 16, 18],
                         'max_features': ['auto', 'sqrt'],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [2, 5, 10],
                         'splitter': ['best', 'random']})
```

```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 4, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5,
'splitter': 'best'}
accuracy : 0.8892857142857142
```

TASK 9

Calculate the accuracy of tree_cv on the test data using the method `score`:

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

```
0.9444444444444444
```

Confusion Matrix

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

- The Confusion Matrix of Decision Tree classifier provides information that proves it's accuracy. Big numbers of true positives and true negatives in comparison with the false.

Conclusions

- The best launch site is KSC LC-93A
- Using payloads over 7000 kg is less risky
- Decision Tree is the best model to predict successful landing in the future

Appendix

- All the notebooks and the dash code (plus screenshots) are uploaded on the Github page

Thank you!