
IBM Developer
SKILLS NETWORK
Winning Space Race
with Data Science
Tobias Meier
24.09.2021

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- The assignment is about SpaceX and their launch data. The two main data sources were the SpaceX Rest API and Wikipedia page.
- In order to get a better data quality, there were some data wrangling steps done.
- Through SQL we gain some first insights from this data.
- Further there were some data visualization and statistical analysis done.
- The dashboard build with plotly and dash, as well as some map visualization with folium got us more information and insights
- At the end, a few classification models were built to predict the outcome and solve the problem.

Findings:

- The EDA, showed us that the columns payload, orbit type and flight number have a significant impact on the success rate of SpaceX launches.
- The best algorithm for the problem was the Decision Tree Algorithm

Introduction

- SpaceX is one of the most successful commercial space age pioneers. They sent spacecraft to the International Space Station and manned missions to Space. They provided Starlink, a satellite internet constellation providing satellite Internet access. One reason they can do this is the rocket launches are relatively inexpensive. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch.
- The Project outline is:
  - Determine the price of each launch by gathering information about SpaceX and creating dashboards for team
  - Determine if SpaceX will reuse the first stage by training a machine learning model and use public information to predict if SpaceX will reuse the first stage.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data collection through SpaceX REST API (https://api.spacexdata.com/v4/launches/past)
  - Data collection through web scrapping from the Wikipedia page (https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922)
- Perform data wrangling
  - Processed raw data by handling missing values, applying one-hot encoding, calculating training labels
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Standardize the data, split into training data and test data, find best Hyperparameter for SVM, KNN, Decision Tree and Logistic Regression, calculate the accuracy for all the algorithms, choose the best performing algorithm

Data Collection

For the capstone assignment, we worked with SpaceX launch data. In order to gather the data, we used the following data sources and methods:

- REST API: We used “api.spacexdata.com/v4/launches/past” to target a specific endpoint of the API to get past launch data. We performed a GET request using the “requests” library to obtain the launch data, which we used to get the data from the API. Our response was in the form of a JSON, specifically a list of JSON objects. To convert this JSON to a “dataframe”, we used the “json_normalize” function. This function allowed us to “normalize” the structured json data into a flat table. Finally exported it into CSV file.
- Web Scraping: As an alternative way to obtain Falcon 9 Launch data, we used the Python BeautifulSoup package to web scrape some HTML tables on the related Wiki pages that contain valuable Falcon 9 launch records. Then we parsed the data from those tables and convert them into a “Pandas” “dataframe” for further visualization and analysis. We transformed this raw data into a clean dataset. Finally exported it into CSV file.

Data Collection – SpaceX API

1. Sending GET request
2. Collecting JSON response
3. Normalizing json into dataframe
4. Cleaning and converting features
5. Creating a dictionary from selected features
6. Transform dictionary to dataframe
7. Filtering dataframe and exporting into a flat file

GitHub URL of the completed SpaceX API calls notebook:

- https://github.com/ToJoMe/Applied-Data-Science-Capstone/blob/master/Data-Collection-API.ipynb

Data Collection - Scraping

1. Sending GET request
2. Creating BeautifulSoup object from response
3. Finding the related HTML tables
4. Extracting feature set
5. Parsing HTML tables to create dictionary
6. Transform dictionary to dataframe
7. Exporting into a flat file

GitHub URL of the completed web scraping notebook:

- https://github.com/ToJoMe/Applied-Data-Science-Capstone/blob/master/Data-Collection-Web-Scraping.ipynb

Data Wrangling

- In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.

- We converted those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.

- We also processed raw data by handling missing and applying one-hot encoding where necessary.

Github-URL:
https://github.com/ToJoMe/Applied-Data-Science-Capstone/blob/master/Data-Collection-API.ipynb

EDA with Data Visualization

We applied data visualization to gain further insights into the data. In order to do that, we plotted scatter point charts, bar charts and line charts to visually check if there are any relationship between selected features.

We plotted scatter point charts to observe and show relationships between two numeric variables.

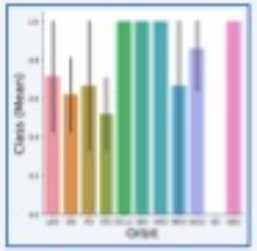
Following charts were plotted:

- Payload mass vs Flight number
- Launch site vs Flight number
- Launch site vs Payload mass
- Orbit vs Flight number
- Orbit vs Payload mas

We plotted bar charts to perform a comparison of metric values across different subgroups

Following charts were plotted:

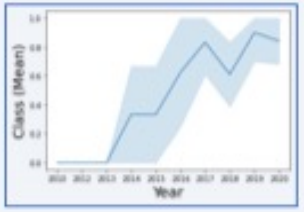
- Orbit vs Succes rate



We plotted line charts to track changes over short or long periods of time.

Following charts were plotted:

- Succes Rate vs Year



- Github-URL: https://github.com/ToJoMe/Applied-Data-Science-Capstone/blob/master/EDA-Visualization.ipynb

EDA with SQL

In order to gain a preliminary understanding and get acquainted with the dataset following queries were performed:

- Get the unique launch sites
- Get the 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was acheived.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- 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

- Github-URL: https://github.com/ToJoMe/Applied-Data-Science-Capstone/blob/master/EDA-SQL.ipynb

Build an Interactive Map with Folium

- We added each site's location on a map using site's latitude and longitude coordinates to gain insights. We used “folium.Circle” to add a highlighted circle area with a text label on these locations.
- We also added markers for each site and created icons showing launch site name.
- In order to visualize multiple launch outcomes for each launch site, we used “MarkerCluster()” and assigned
- color to launch outcomes, green for successful launches (class = 1) and red unsuccessful ones (class = 0).
- To explore and analyze the proximities of launch sites, we added a “MousePosition()” on the map to get coordinate for a mouse over a point on the map
- By using calculate_distance function, we calculated distances between launch sites and closest railways, coastlines and city centers.
- We drew a “PolyLine()” between CCAFS-SLC40 to nearest railway point and coastline point.

Findings:

- By visualizing all the objects on a map, we discovered that launch sites are in close proximity to coastline and have access to railroads and keep certain distance away from highways and cities.

Github-URL: https://github.com/ToJoMe/Applied-Data-Science-Capstone/blob/master/Interactive-Visual-Analytics-Folium.ipynb

Build a Dashboard with Plotly Dash

- In order to perform interactive visual analytics on SpaceX launch data in real-time, we build an interactive dashboard using plotly dash. We added following components to dashboard:
  - Added a launch site drop-down input component
  - Added a callback function to render success-pie-chart based on selected site dropdown
  - Added a range slider to select payload
  - Added a callback function to render the success-payload-scatter-chart scatter plot

Findings:

- By using the dashboard, we can analyze the data in real time and answer many question about SpaceX launches such as;
  - Which site has the largest successful launches?
  - Which site has the highest launch success rate?
  - Which payload range(s) has the highest launch success rate?
  - Which payload range(s) has the lowest launch success rate?
  - Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest launch success rate?

Gitbhub-URL: https://github.com/ToJoMe/Applied-Data-Science-Capstone/blob/master/Interactive_Dashboard_with_Ploty_Dash.py

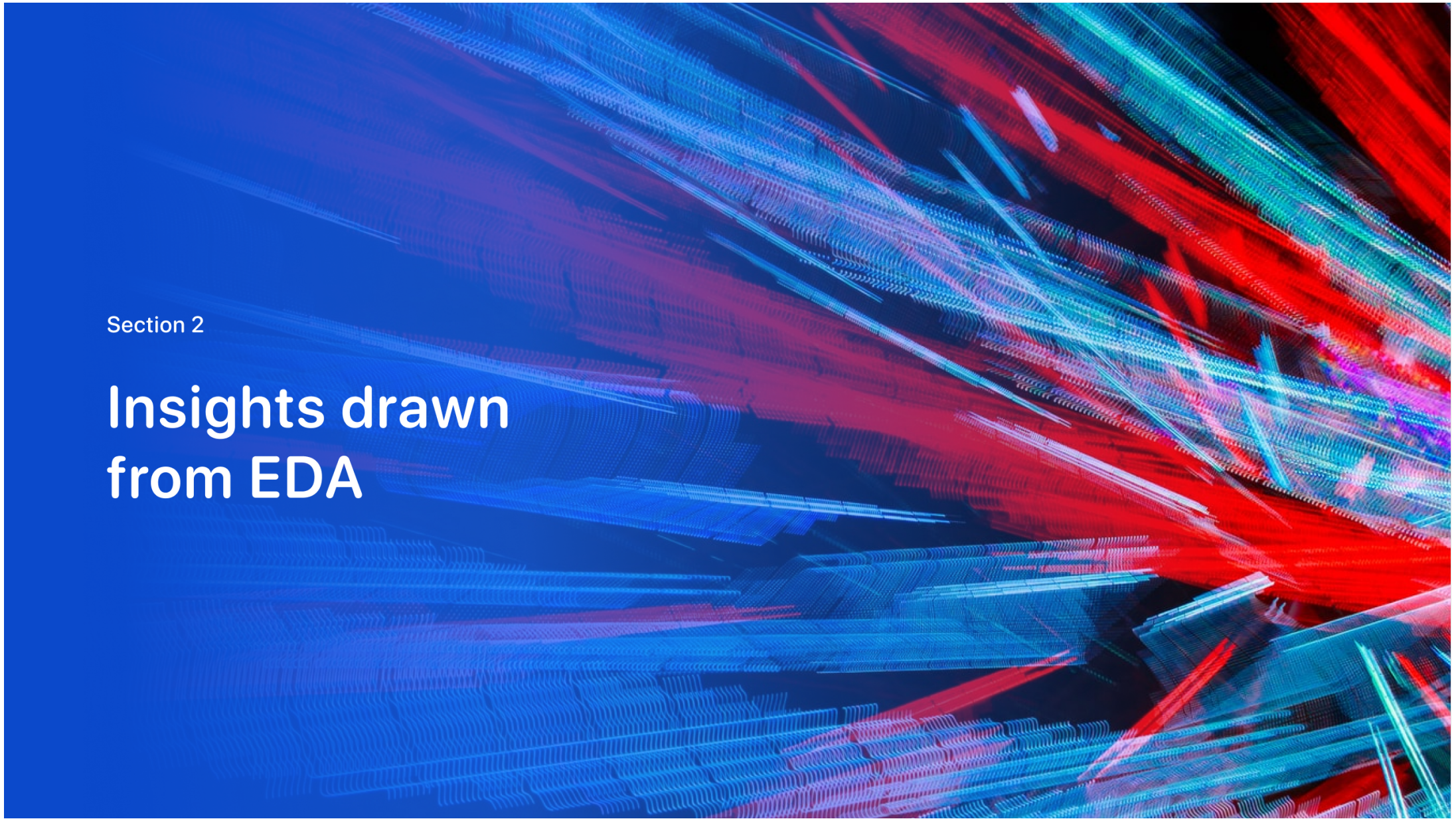
Predictive Analysis (Classification)

- Building model
  - Separating dependent (Y) and independent (X) variables
  - Standardizing independent variables
  - Splitting data into training and test datasets
  - Performing grid search with different algorithms to optimize hyperparameters
  - Training and testing models using different algorithms with optimized hyperparameters
- Evaluating model
  - Checking accuracy for each model
  - Plotting confusion matrix for each model
- Improving model
  - Feature engineering
  - Dimension reduction
- Finding the best performing classification model
  - The model with the best accuracy score is selected as the champion model

Github-URL: https://github.com/ToJoMe/Applied-Data-Science-Capstone/blob/master/Machine-Learning-Prediction.ipynb

Results

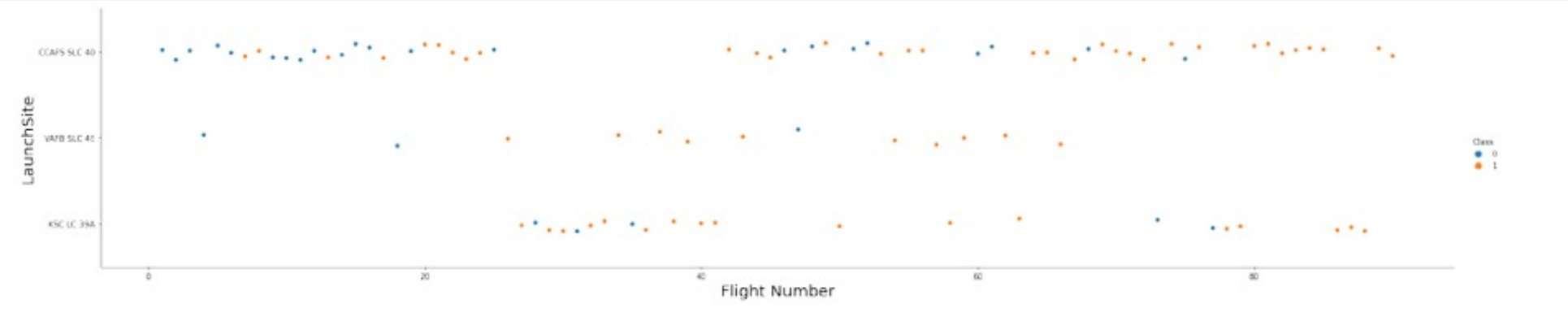
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results



Section 2

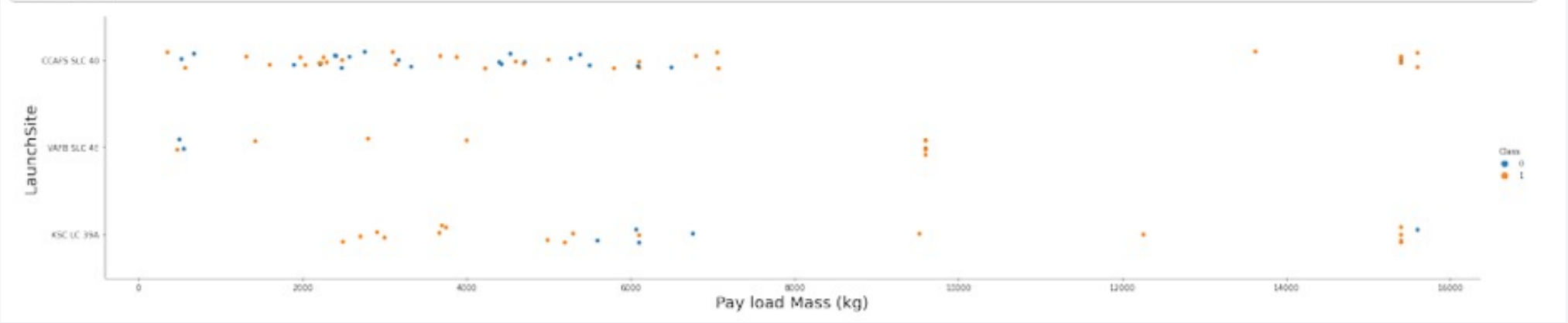
Insights drawn from EDA

Flight Number vs. Launch Site



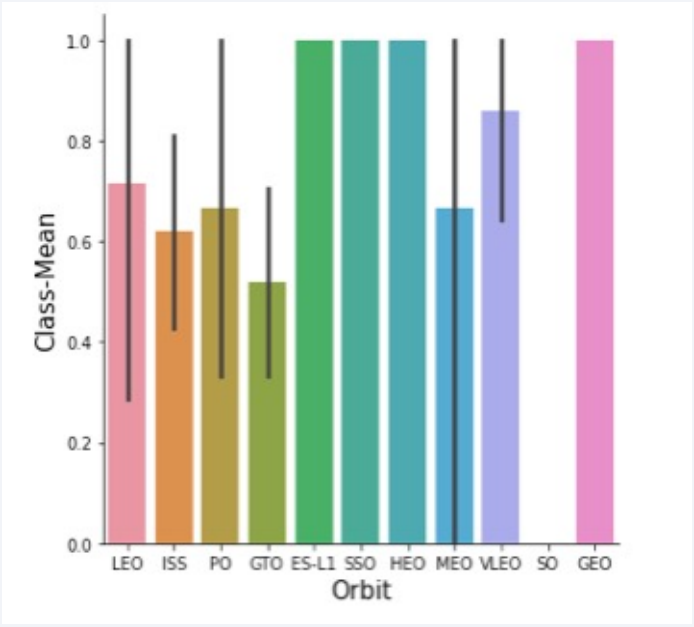
- Explanation: There seems no strong relationship between launch site and flight number. Overall, it tends that higher flight numbers are more successful than the early ones, which makes sense.

Payload vs. Launch Site



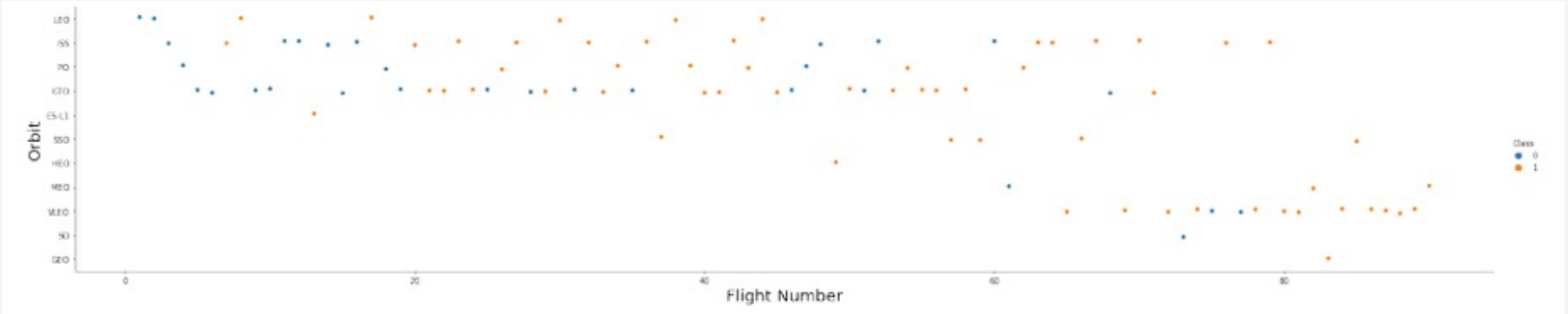
- Explanations: CCAFS SLC 40 has higher success rate when the payload is greater then ~6.500 kg. KSC LC 39A on the other hand has higher success rate with payloads less than ~5.500 kg.

Success Rate vs. Orbit Type



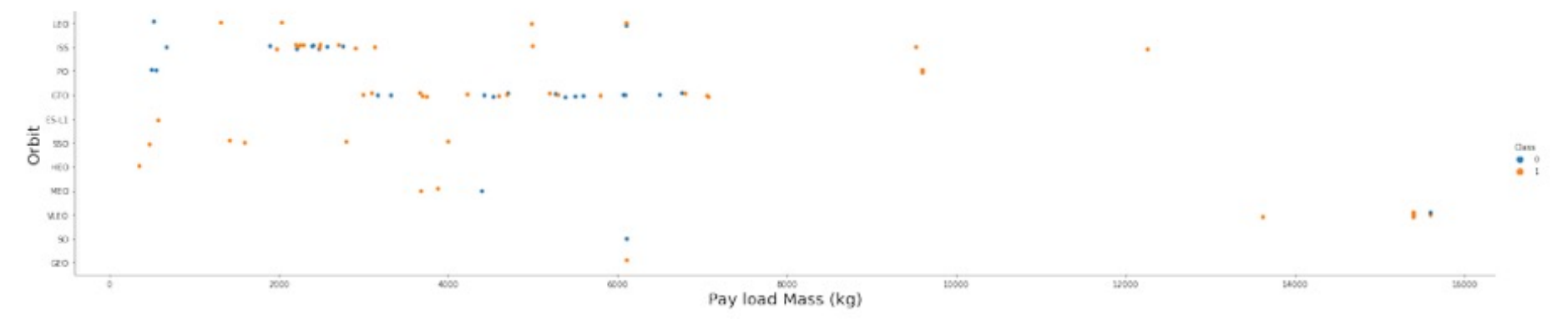
- Explanation: The orbit type with the highest success-rate was ES-L1, GEO, HEO and SSO with a 100% success-rate.

Flight Number vs. Orbit Type



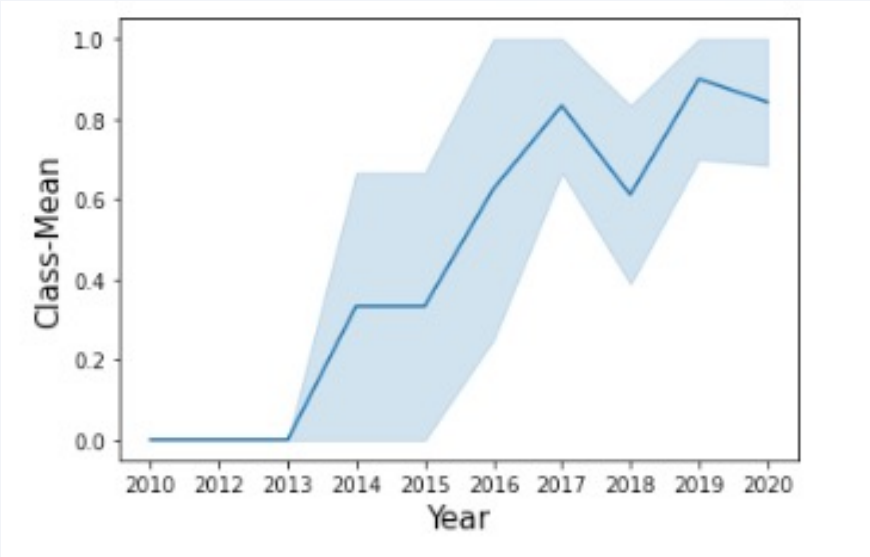
- Explanation: In the LEO orbit the success is related to the flight number because at the beginning there were some failures and after that just success full flights. No relationship for example in the GTO orbit between Orbit and flight number.

Payload vs. Orbit Type



- Explanation: The observation is that heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.

Launch Success Yearly Trend



- Explanation: We can observe that the success rate since 2013 kept increasing till 2017 and then dropped in 2018. In 2019, success rate reached the highest point between 2010 and 2020.

All Launch Site Names

- Unique 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 DISTINCT(Launch_Site) FROM SPACEXTBL;
```

```
 * ibm_db_sa://ffy74292:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.clou
d:32536/BLUDB
Done.
```

2]:

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

- With this query we get all distinct values in the Launch-Site column

Launch Site Names Begin with 'CCA'

```
%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

```
 * ibm_db_sa://ffy74292:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:3253
6/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

- This query get 5 rows where the launch-site starts with the string “CCA”, we set the limit to 5 so we just get 5 records.

Total Payload Mass

```
%sql SELECT customer, SUM(payload_mass__kg_) AS Total_Payload_Mass FROM SPACEXTBL WHERE customer = 'NASA (CRS)' GROUP
BY customer;
```

```
 * ibm_db_sa://ffy74292:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:3253
6/BLUDB
Done.
```

customer	total_payload_mass
NASA (CRS)	45596

- This query shows the total payload mass which was carried out by the customer 'NASA (CRS)'.

Average Payload Mass by F9 v1.1

```
%sql SELECT booster_version, AVG(payload_mass__kg_) AS average_payload_mass_by_F9v1 FROM SPACEXTBL WHERE booster_vers
ion = 'F9 v1.1' GROUP BY booster_version;
```

```
 * ibm_db_sa://ffy74292:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:3253
6/BLUDB
Done.
```

booster_version	average_payload_mass_by_f9v1
F9 v1.1	2928

- This query shows what the average payload was, which was carried out by the booster version 'F9 v1.1'

First Successful Ground Landing Date

```
%sql SELECT landing__outcome, MIN(DATE) AS first_successful_outcome_ground_pad FROM SPACEXTBL WHERE landing__outcome
= 'Success (ground pad)' GROUP BY landing__outcome;
```

```
 * ibm_db_sa://ffy74292:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:3253
6/BLUDB
Done.
```

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

- The first successful ground landing date was on ‘2015-12-22’

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT booster_version FROM SPACEXTBL WHERE landing__outcome = 'Success (drone ship)' AND payload_mass__kg_ BETW
EEN 4000 AND 6000;
```

```
 * ibm_db_sa://ffy74292:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:3253
6/BLUDB
Done.
```

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

- The query shows which booster version where able to have an successful drone ship landing with a payload mass between 4000 and 6000 kg.

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT (CASE WHEN mission_outcome LIKE 'Success%' THEN 'Success' ELSE 'Failure' END) AS outcome, COUNT(Mission_O
utcome) AS total_number FROM SPACEXTBL GROUP BY (CASE WHEN mission_outcome LIKE 'Success%' THEN 'Success' ELSE 'Failu
re' END)
```

```
 * ibm_db_sa://ffy74292:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:3253
6/BLUDB
Done.
```

outcome	total_number
Failure	1
Success	100

- In total there were 100 successful missions and 1 mission with the status failure

Boosters Carried Maximum Payload

```
%sql SELECT DISTINCT(booster_version) FROM SPACEXTBL WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM SP
ACEXTBL);
```

```
 * ibm_db_sa://ffy74292:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:3253
6/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

- This query shows the names of booster version, which carried out the max payload mass in kg, this was 15600 kg.

2015 Launch Records

```
%sql SELECT DISTINCT(landing__outcome), booster_version, launch_site, date FROM SPACEXTBL WHERE landing__outcome =  '
Failure (drone ship)' AND YEAR(Date) = 2015;
```

```
 * ibm_db_sa://ffy74292:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:3253
6/BLUDB
Done.
```

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

- This query shows the booster version and launch sites where failure with drone ship happened in the year 2015.

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

```
%sql SELECT landing__outcome, COUNT(landing__outcome) as total from SPACEXTBL WHERE Date BETWEEN '2010-06-04' AND '20
17-03-20' GROUP BY landing__outcome ORDER BY COUNT(landing__outcome) DESC
```

```
 * ibm_db_sa://ffy74292:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:3253
6/BLUDB
Done.
```

landing__outcome	total
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

- This query shows the ranking 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, in descending order.

Section 4

Launch Sites Proximities Analysis

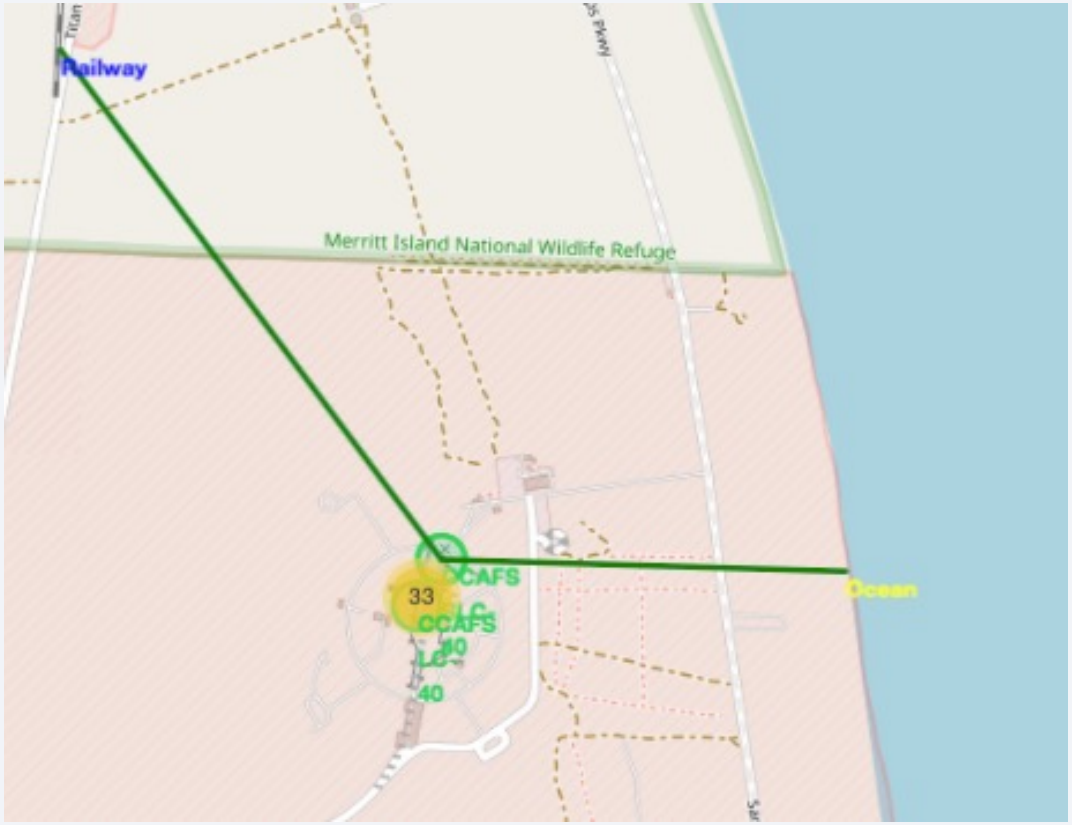
Launch-Site-Marker

- The Map shows all Launch-Sites of the organization at this map, to get an overview where the Launch-Sites are located. One on the West-Coast and three on the East-Coast. All close to the equator

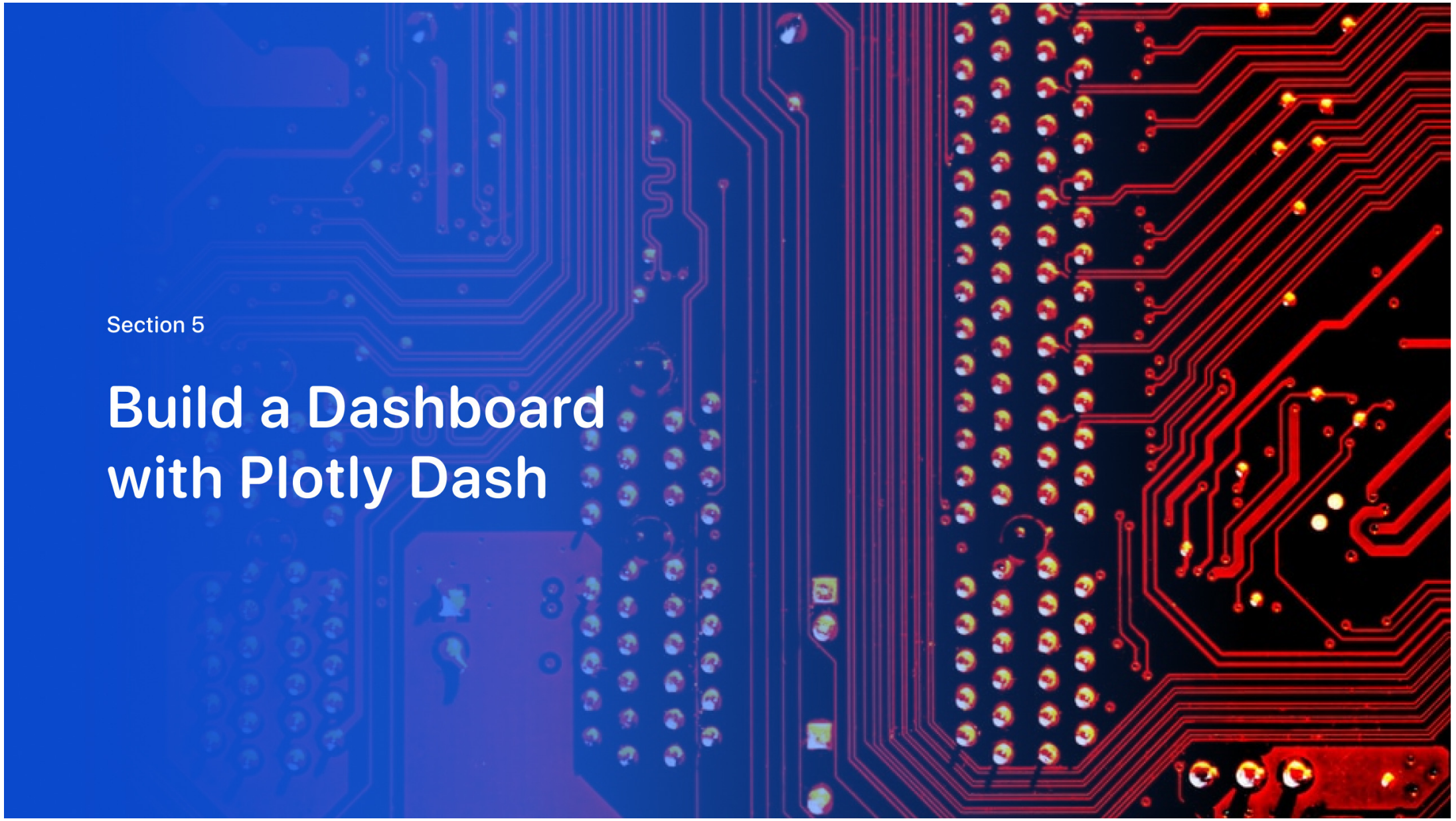
Launch-Outcomes at each Launch-Site

- The Map shows all Launch-Sites with each outcomes. This gives an insight at how many launches where at each site and how much were successful at each site. CCAFS LC-40 was the launch site with the most launches, but the success rate is low. KSC LC 39 has the second most launches and has a decent success rate.

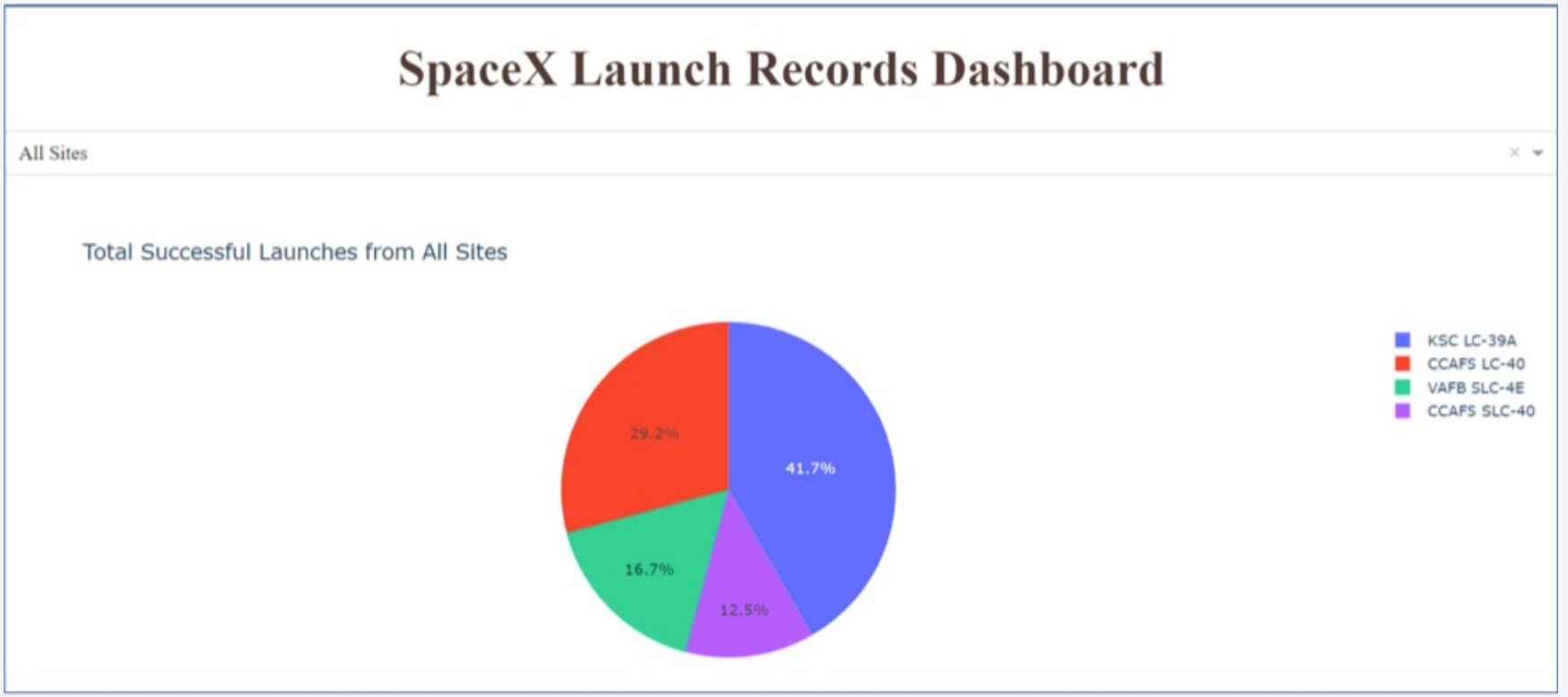
Launch-Site CCAFS SLC-40 and proximities around



- The Map shows the Launch-Site CCAFS SLC-40 and the proximities like a railway or the nearest ocean around. This gives an insight of proximities around each Launch-Site, which may cause some influence on outcomes and circumstances at the launch-site.

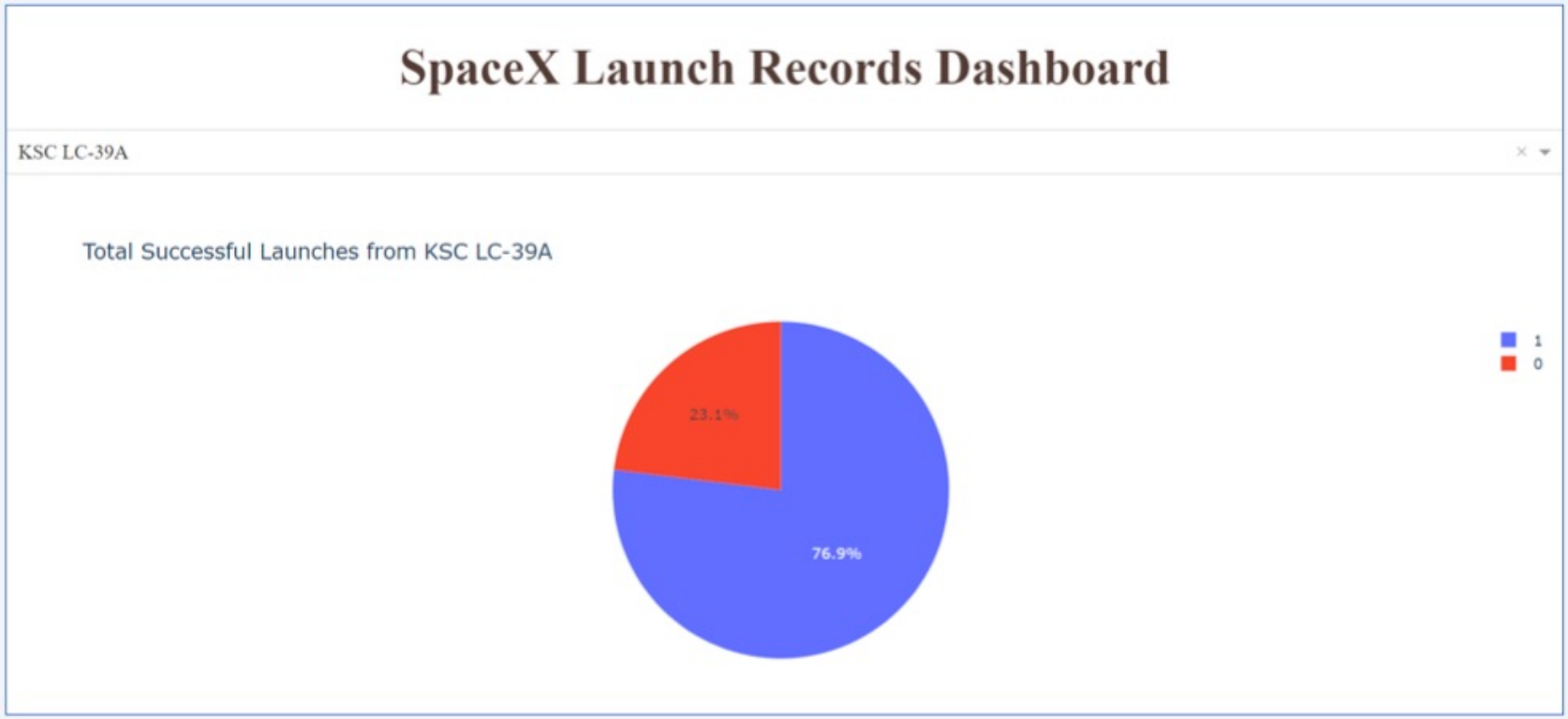

Section 5
Build a Dashboard with Plotly Dash

Overview of all Launch-Sites



The launch site KSC LC-39A had the most successful launches overall.

KSC LC-39A



- KSC LC-39A has 76.9% success rate on launches.

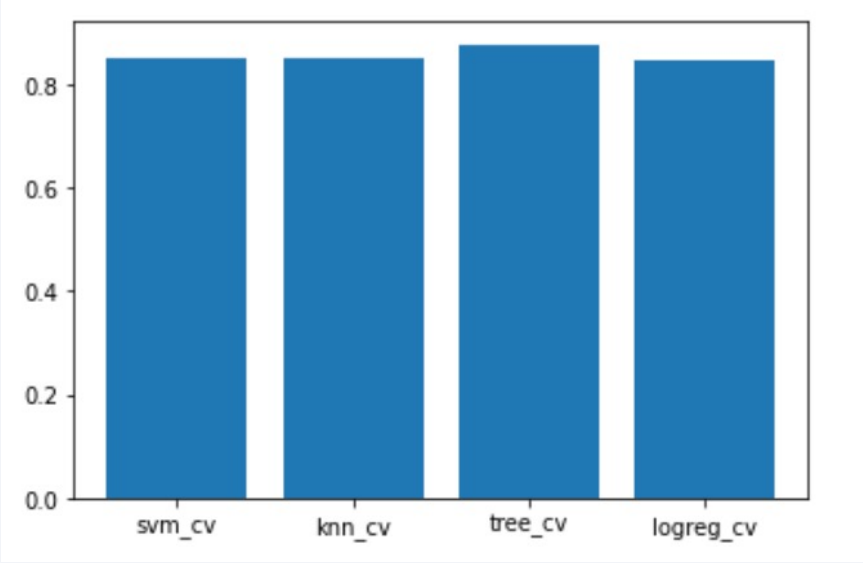
Payload vs. Launch Outcome

- FT booster had 80% success rate when the payload was between 2.000 and 4.000 kg.
- V1.1 booster failed every launch for the same payloads.

Section 6

Predictive Analysis (Classification)

Classification Accuracy



All 4 algorithms produced accuracy ratios quite close to each other:

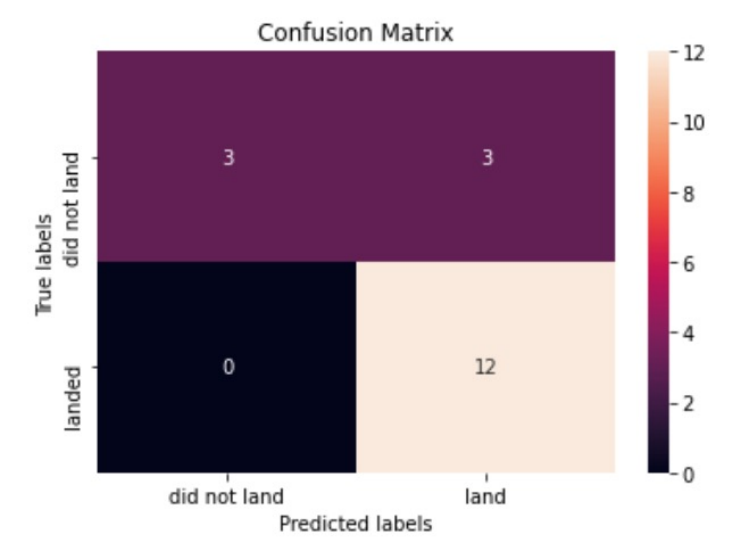
- K-Nearest Neighbors: 0.848,
- Support Vector Machines: 0.848,
- Decision Tree: 0.876,
- Logistic Regression: 0.846

Decision tree algorithm has the highest classification accuracy.

Optimized hyperparameters were as follows:

- Criterion: gini
- Max depth: 6
- Max_features: auto
- Min samples leaf: 2
- Min samples split: 5
- Splitter: random

Confusion Matrix



- Decision Tree is chosen as the best algorithm based on the high accuracy ratio.
- Decision Tree can distinguish between the different classes.
- The major problem is false positives, where algorithm predicts that the rocket would land but in fact it didn't.

Conclusions

- SpaceX launch sites are in close proximity to coastline and have access to railroads and they have certain distance away from highways and cities.
- Launch sites are located in Florida and California, close to equator.
- CCAFS LC-40 is the launch site with the most launches, but the success rate is fairly low. KSC LC-39 has the
- second most launches and has a success rate of 76.9%. It's the most successful launch site among 4.
- Low weighted payloads perform better. Payloads between 2.500 and 5.000 kg have the best success rate. FT booster had 80% success rate when the payload was between 2.000 and 4.000 kg.
- ES-L1, SSO, HEO and GEO orbits have 100% success rate.
- As the number of flights rises, success rate also rises. We can observe that the success rate since 2013 kept increasing till 2017 and then dropped in 2018. In 2019, success rate reached the highest point between 2010 and 2020.
- Decision Tree is chosen as the best algorithm based on the high accuracy ratio. But model incorrectly labels significant number of failed landings as successful landings.

Thank you!