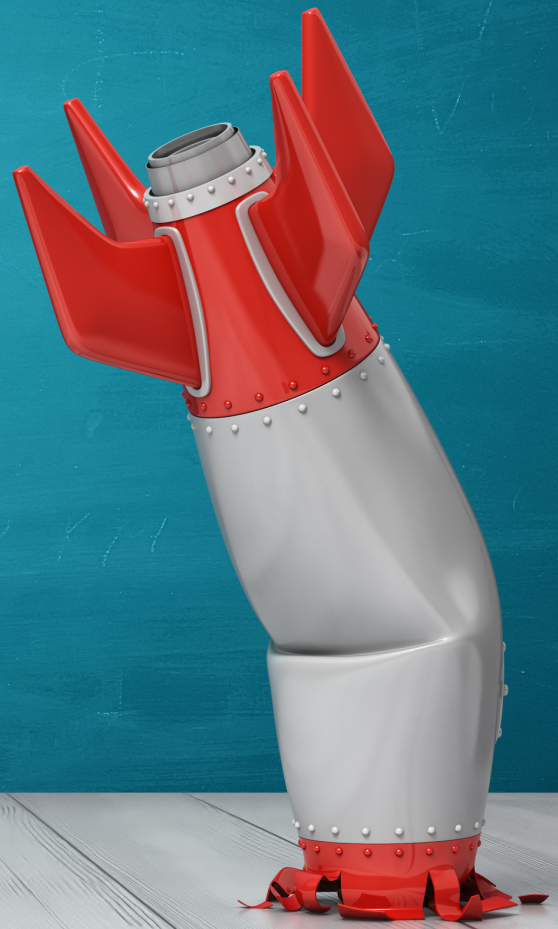
Winning Space Race with Data Science

Juraj Hudák
25/11/2022

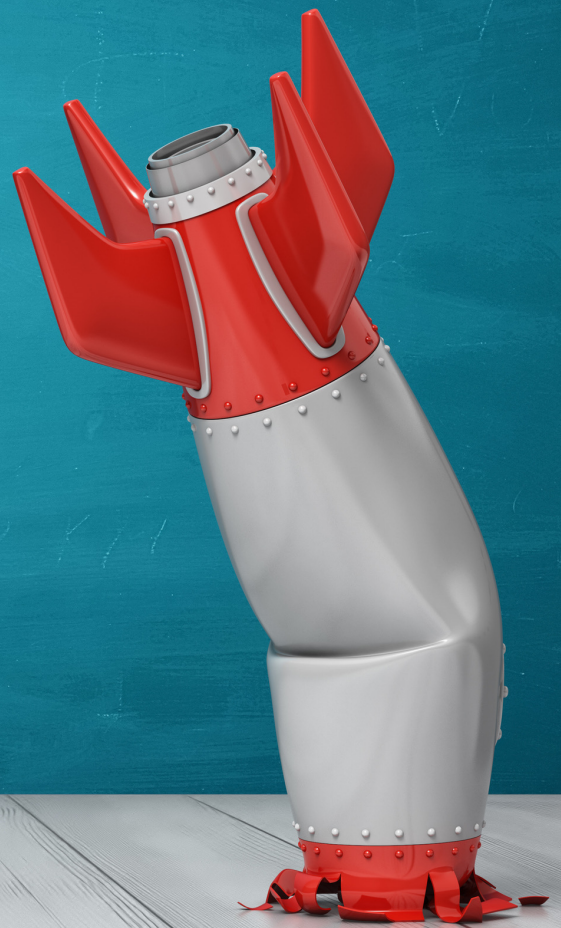
Outline (Table of Contents)



Executive summary

This project focuses on building the optimal model to predict Falcon 9 launches' success from scratch. It involves extracting the necessary data from various sources (APIs, web scraping), cleaning it up, performing introductory data analyses (SQL queries, pandas library) and also visualizing the relationships between variables (Folium, Plotly by Dash) in order to find those used in classification.

Finally, we go through 4 different classification algorithms, tuning their hyperparameters, analyze their results and try to select the optimal one by given criteria.



Introduction

During this project, we aim to predict if the Falcon 9's first stage (by SpaceX) lands successfully.

- SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage;
- if we can determine that the first stage lands, we can also tell the cost of a launch. 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:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Two ways – SpaceX API & web scraping

Specifics:

-> Get the booster's name from ***rocket*** column of the source.

-> From ***launchpad,*** export launch site's name and its longitude & latitude.

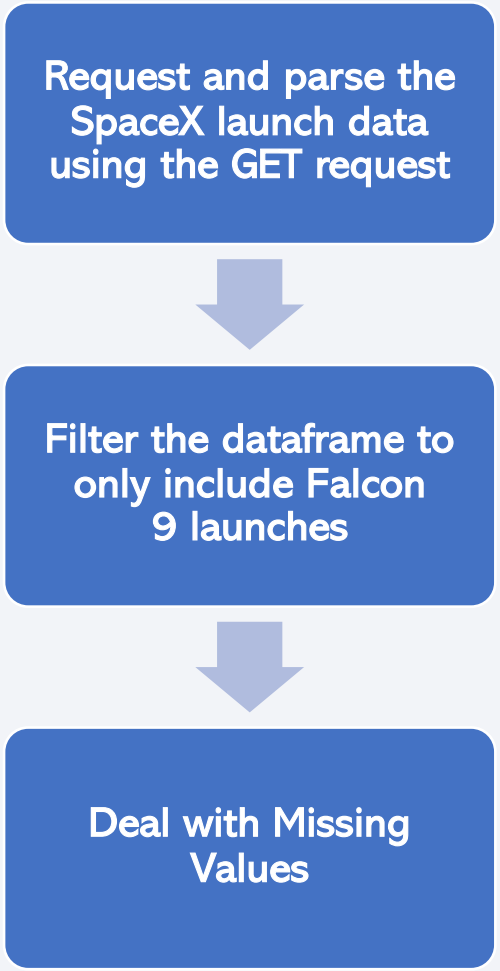
-> From ***payload*** we get the mass of the payload (in kg) and the orbit it's going to.

-> From ***cores*** we would like to learn the outcome of the landing, the type of the landing, number of flights with that core, whether gridfins were used, wheter the core is reused, wheter legs were used, the landing pad used, the block of the core which is a number used to seperate version of cores, the number of times this specific core has been reused, and the serial of the core.

Data Collection – SpaceX API

- Making a get requests to the SpaceX API, getting the partial results in json format and using json_normalize to convert the results into dataframe.
- Basic data wrangling and formatting.
- Completed jupyter notebook:

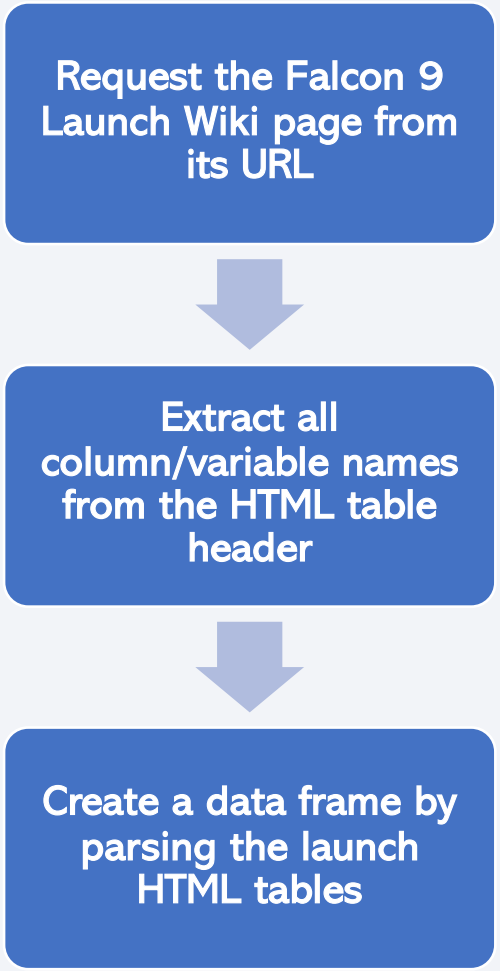
https://github.com/juraj-hudak2/IBM_Data_Science_Professional_repo/blob/main/SpaceX%20Week%201%20-%20Collecting%20the%20data.ipynb



Data Collection - Scraping

- Webscraping Falcon 9 launch records with ‘BeautifulSoup‘
- Completed jupyter notebook:

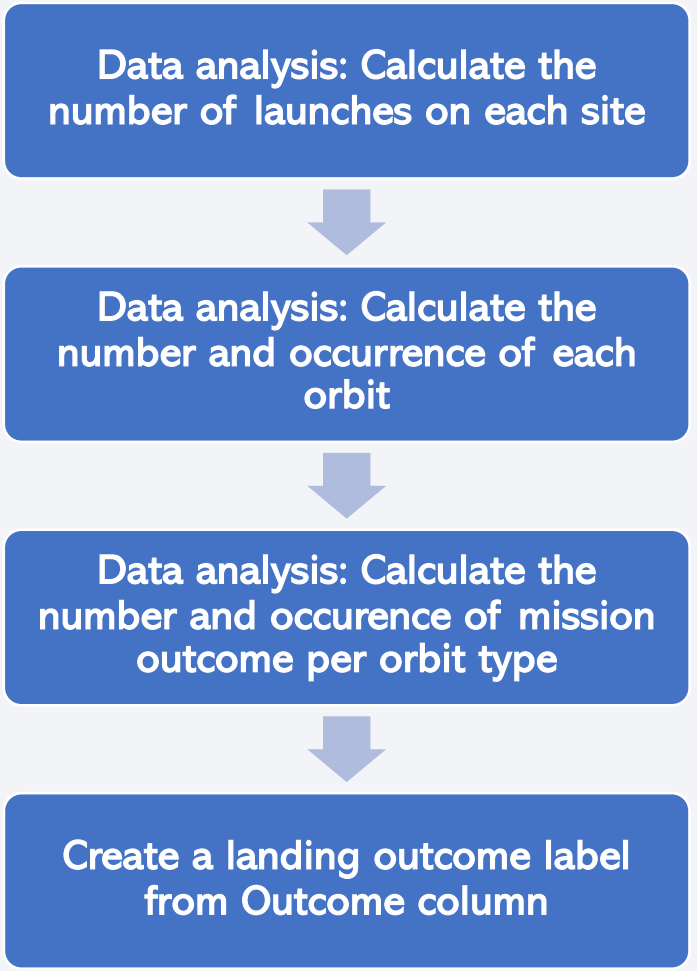
https://github.com/juraj-hudak2/IBM_Data_Science_Professional_repo/blob/main/SpaceX%20Week%201%20-%20Web%20scraping.ipynb



Data Wrangling

- Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.
- Completed jupyter notebook:

https://github.com/juraj-hudak2/IBM_Data_Science_Professional_repo/blob/main/SpaceX%20Week%201%20-%20Data%20wrangling.ipynb



EDA with Data Visualization

Charts to see the launch success progress in time with respect to one other variable:

- Scatterplot of Flight number vs. Payload mass
- Scatterplot of Flight number vs. Launch site
- Scatterplot of Flight number vs. Orbit

Relationship between Payload mass and Launch site (and launch success):

- Scatterplot of Launch site vs. Payload mass

Relationship between Orbit and launch success:

- Bar chart of Orbit vs. Class variable

Relationship between Payload mass and Orbit:

- Scatterplot of Payload mass vs. Orbit

Yearly trend of launch success rates:

- Lineplot of Date vs. Class variable

Completed jupyter notebook:

https://github.com/juraj-hudak2/IBM_Data_Science_Professional_repo/blob/main/SpaceX%20Week%202%20-%20EDA%20with%20Data%20visualization.ipynb

EDA with SQL

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

Completed jupyter notebook:

https://github.com/juraj-hudak2/IBM_Data_Science_Professional_repo/blob/main/SpaceX%20Week%202%20-%20EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

Added objects:

- Markers and circles for all launch sites.
- Creating marker cluster at every launch site, which, after clicking at it, unveils the color (according to success/failure of the launch) labeled icons.
- Distance (number) from CCAFS SLC-40 launch site to the closest coastline and line connecting those two points on map.
- Distance (number) from VAFB SLC-4E launch site to the closest railway and line connecting those two points on map.

Completed jupyter notebook:

https://github.com/juraj-hudak2/IBM_Data_Science_Professional_repo/blob/main/SpaceX%20Week%203%20-%20Launch%20sites%20locations%20analysis%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

Added objects/components and their purpose:

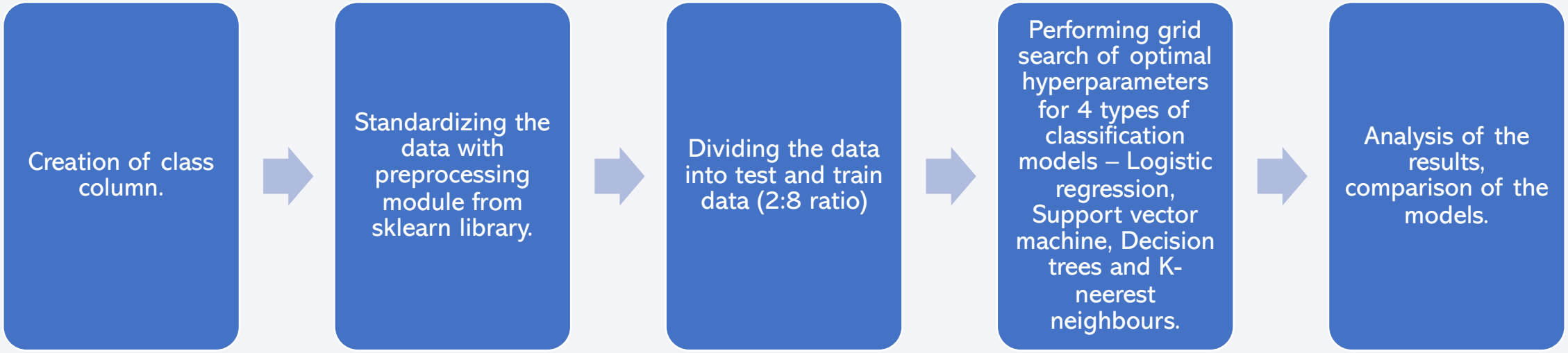
- Launch Site Drop-down Input Component – user is able to view interactive parts of the app based on his/her choice of launch sites – either all of them or one specific.
- Callback function to render ***success-pie-chart*** based on selected site dropdown – user can tell which site has the largest number of successful launches and which site has the highest launch succest rate.
- Range Slider to select Payload interval.
- Callback function to render the ***success-payload-scatter-chart*** scatterplot – user can visually estimate payload ranges / booster versions with highest/lowest launch success rate.

Completed python script:

https://github.com/juraj-hudak2/IBM_Data_Science_Professional_repo/blob/main/SpaceX%20Week%203%20-%20Interactive%20dashboard%20with%20Plotly%20Dash.py

Predictive Analysis (Classification)

Process of classification model building:

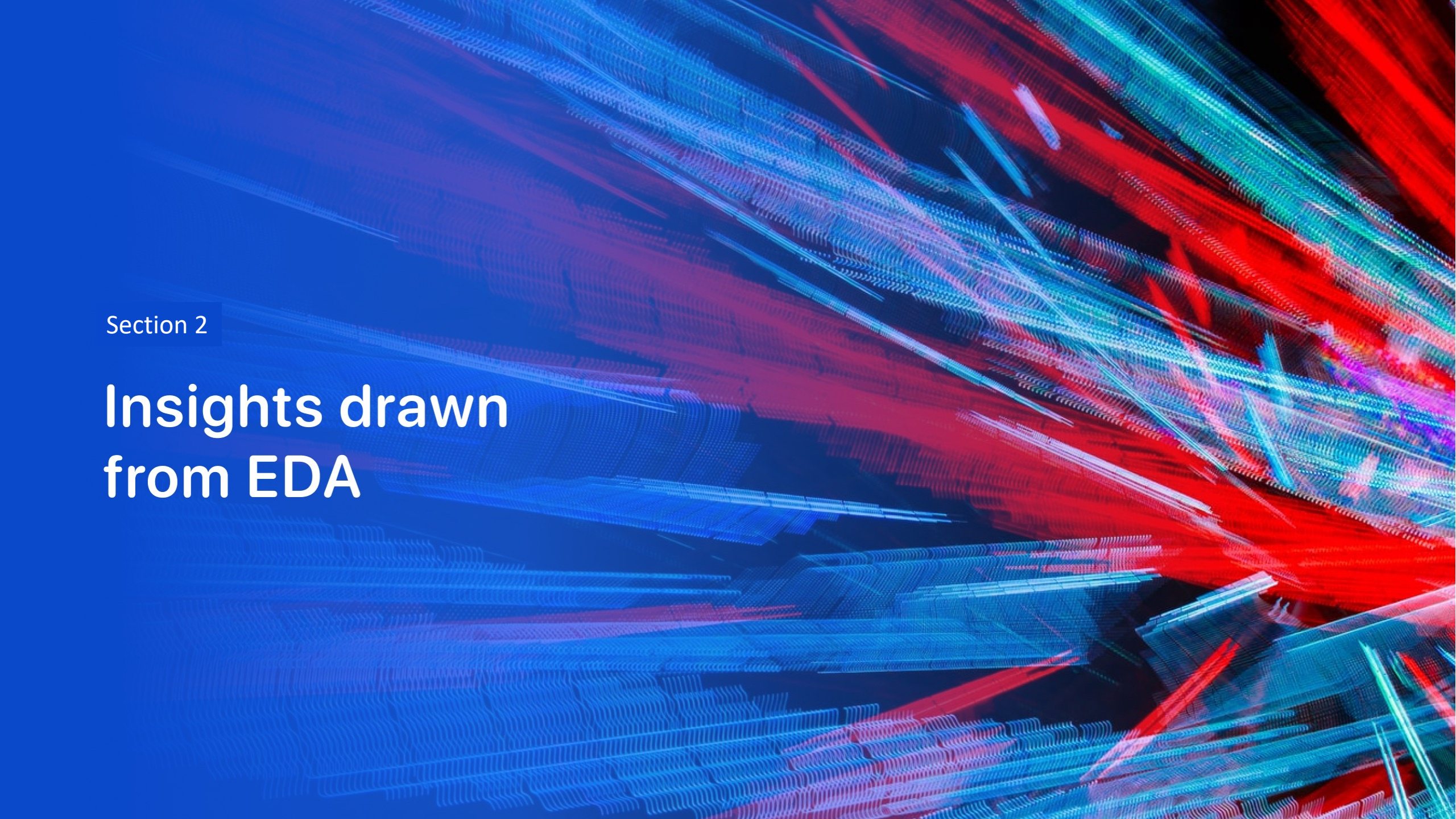


Completed jupyter notebook:

https://github.com/juraj-hudak2/IBM_Data_Science_Professional_repo/blob/main/SpaceX%20Week%204%20-%20Machine%20learning%20prediction.ipynb

Results

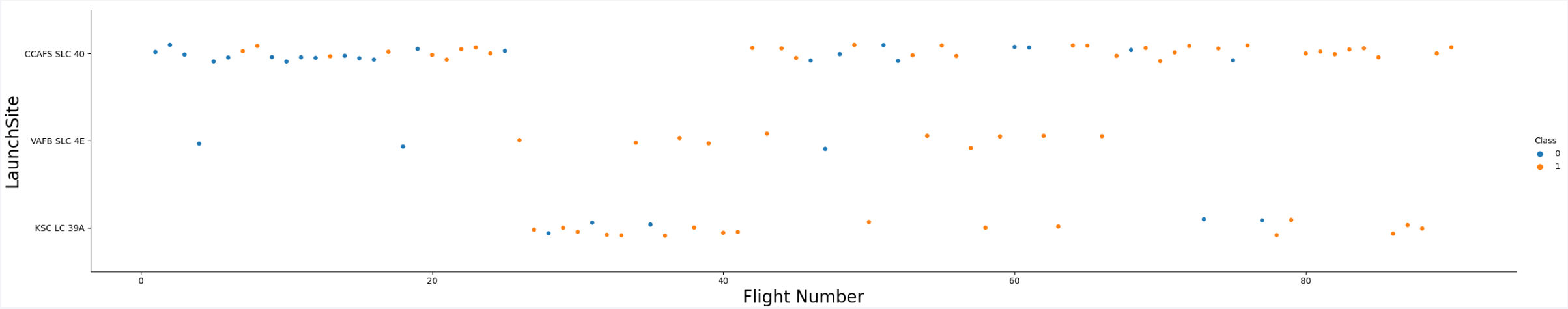
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

Remarks:

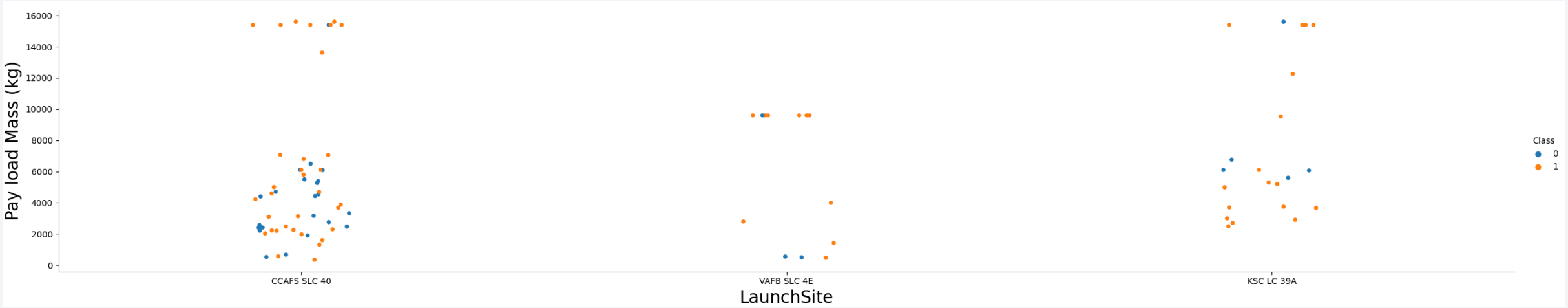
- Out of first 25 launches, 23 started at CCAFS SLC-40 site.
- Even after adding other two sites, it remained as the one most used to launch Falcon 9s.



Payload vs. Launch Site

Remarks:

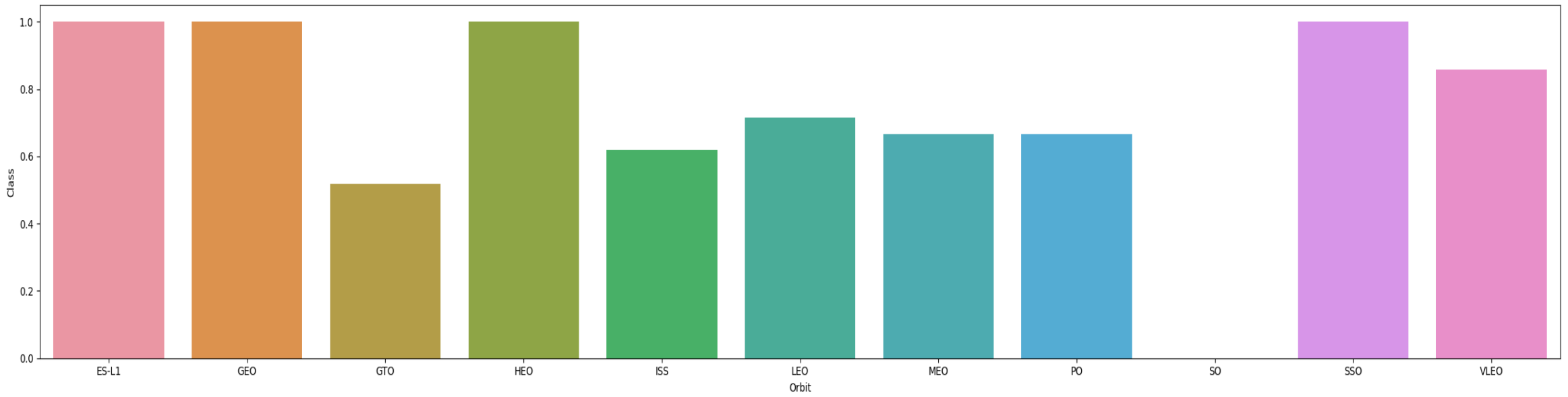
- VAFB SLC 4E site was used only for payload not greater than 10000kg.



Success Rate vs. Orbit Type

Remarks:

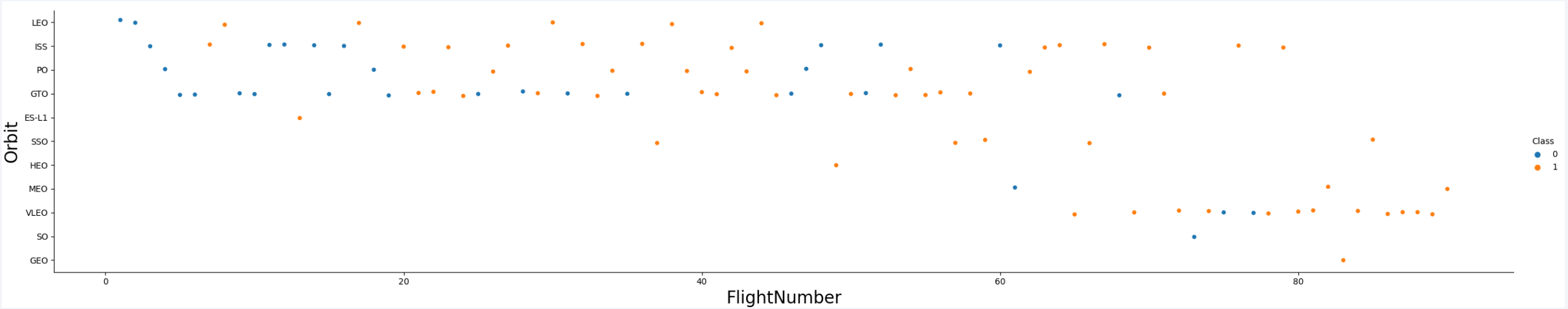
- There are 4 orbits with 100% success rate – ES-L1, GEO, HEO and SSO.



Flight Number vs. Orbit Type

Remarks:

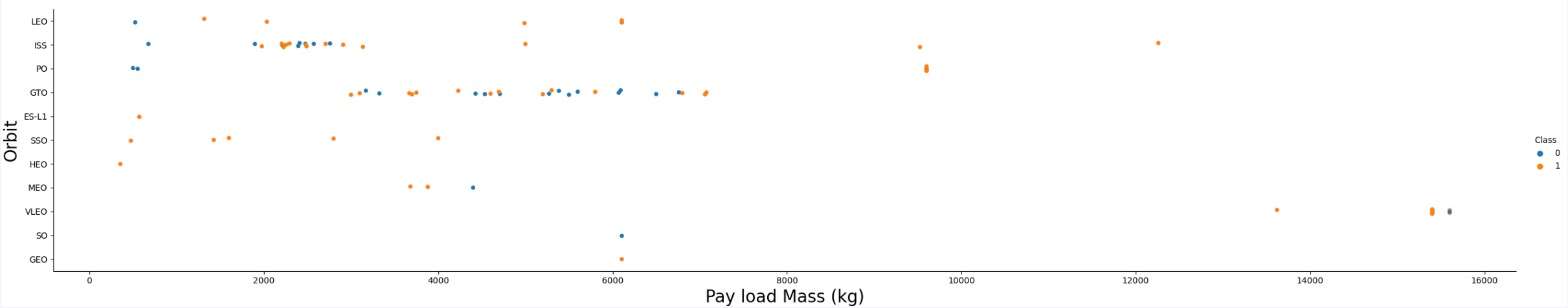
- Initially, only orbits LEO, ISS, PO AND GTO were mostly used, but gradually others, namely VLEO orbit have joined.
- There seems to be no relationship between flight number when in GTO orbit.



Payload vs. Orbit Type

Remarks:

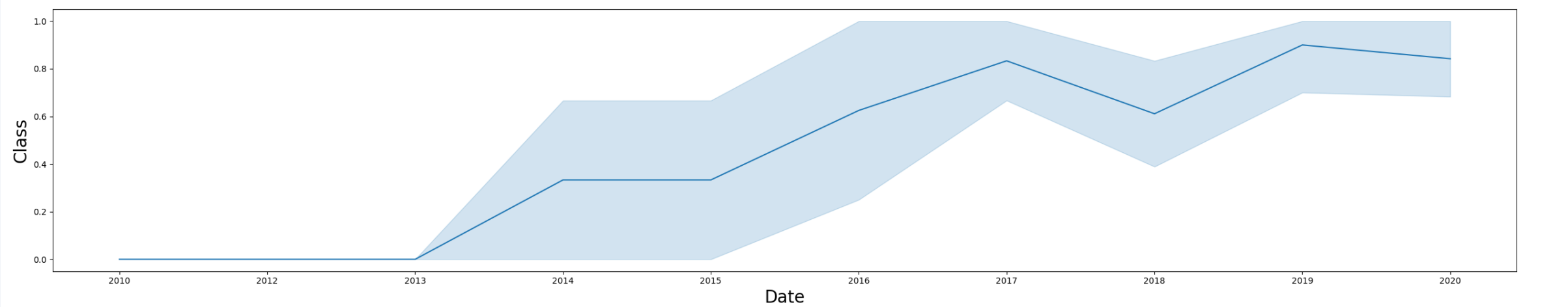
- For heavy payloads, there is higher positive landing rate for orbits PO, ISS and VLEO.



Launch Success Yearly Trend

Remarks:

- Since 2013, the success rate has been steadily increasing (with small dip in 2018).



All Launch Site Names

- There are 4 unique launch sites:

Task 1

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

```
%sql SELECT UNIQUE launch_site FROM SPACE_TABLE
```

Python

```
 * ibm_db_sa://plm72100:***@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'

Task 2

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

```
%sql SELECT * FROM SPACE_TABLE WHERE launch_site LIKE 'CCA%' LIMIT 5
```

Python

```
* ibm_db_sa://plm72100:***@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

- Total mass carried by boosters launched by NASA (CRS) is 45596kg.

Task 3

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

```
%sql SELECT sum(payload_mass__kg_) AS total_payload FROM SPACE_TABLE WHERE customer = 'NASA (CRS)'
```

Python

```
 * ibm_db_sa://plm72100:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

total_payload
45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by F9 v1.1 booster is 2928kg.

Task 4

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

```
%sql select avg(payload_mass__kg_) as avg_payload from SPACE_TABLE where booster_version = 'F9 v1.1'
```

Python

```
 * ibm_db_sa://plm72100:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

avg_payload
2928

First Successful Ground Landing Date

- The first successful landing in ground pad was on the 01-05-2017.

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint:Use min function

[10]:
```
%sql select MIN(Date) as Date FROM SPACEXTBL WHERE "Landing _Outcome" = 'Success (ground pad)'
# where LandingOutcome = 'Success (ground pad)'
```

```
 * sqlite:///my_data1.db
Done.
```

[10]:

Date
01-05-2017

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

```
%sql SELECT unique booster_version FROM SPACE_TABLE where mission_outcome = 'Success' and landing__outcome like '%drone ship%' and payload_mass__kg_ between 4000 and 6000
```

Python

```
 * ibm_db_sa://plm72100:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

```
%sql SELECT mission_outcome, count(*) as counts from SPACE_TABLE group by mission_outcome
```

Python

```
 * ibm_db_sa://plm72100:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

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

Boosters Carried Maximum Payload

- These 12 boosters have carried the maximum payload mass.

Task 8

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

```
%sql SELECT booster_version FROM SPACE_TABLE WHERE payload_mass__kg_ = (SELECT max(payload_mass__kg_) FROM SPACE_TABLE)
```

Python

```
 * ibm_db_sa://plm72100:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

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

2015 Launch Records

- There is 1 failed landing with given conditions.

Task 9

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

```
%sql SELECT landing__outcome, booster_version, launch_site FROM SPACE_TABLE where mission_outcome like '%Failure%' and landing__outcome like '%drone ship%' and year(DATE) = 2015
```

Python

```
 * ibm_db_sa://plm72100:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

landing__outcome	booster_version	launch_site
Precluded (drone ship)	F9 v1.1 B1018	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:

Task 10

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, count(landing__outcome) as counts FROM SPACE_TABLE where DATE between '2010-06-04' and '2017-03-20' group by landing__outcome order by counts desc
```

Python

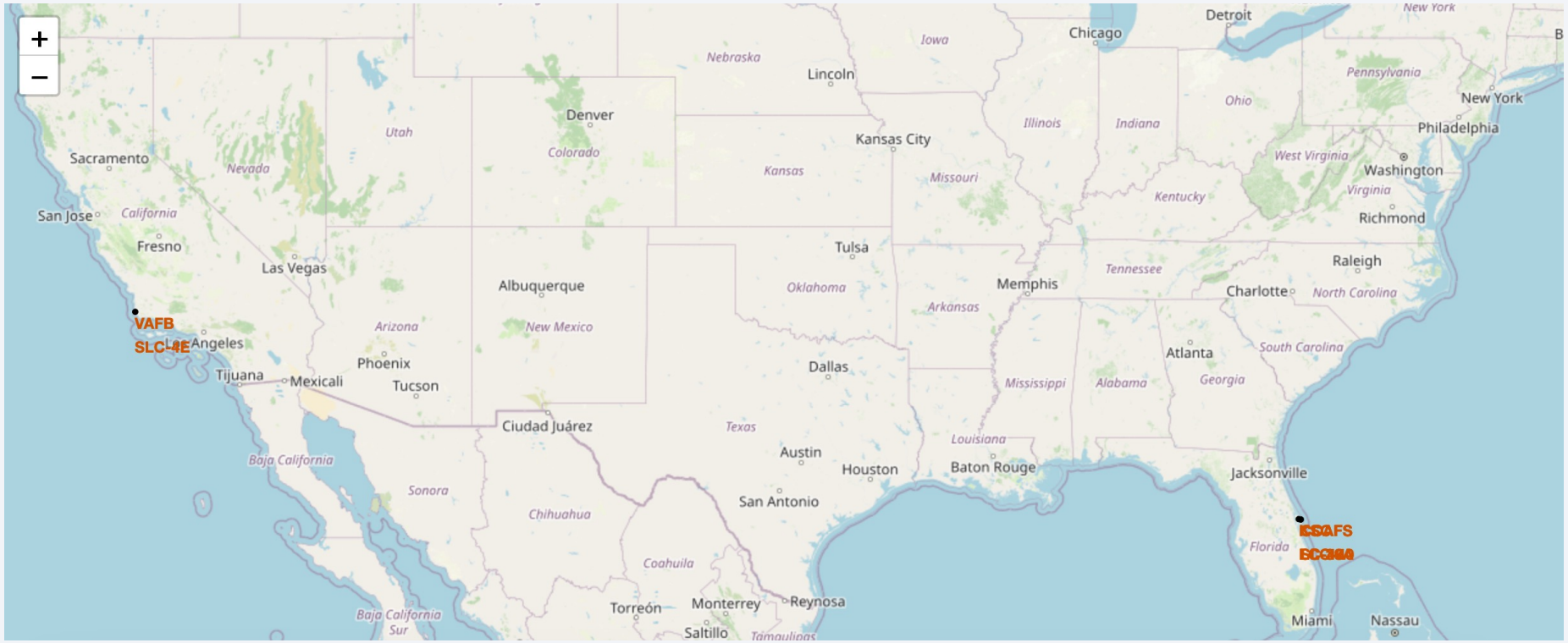
```
 * ibm_db_sa://plm72100:***@55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31929/bludb
Done.
```

landing__outcome	counts
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


Section 3
Launch Sites
Proximities Analysis

Folium Map Screenshot: all launch sites

- Folium map that includes all launch sites used by SpaceX – situated at the west and east coast of the USA.

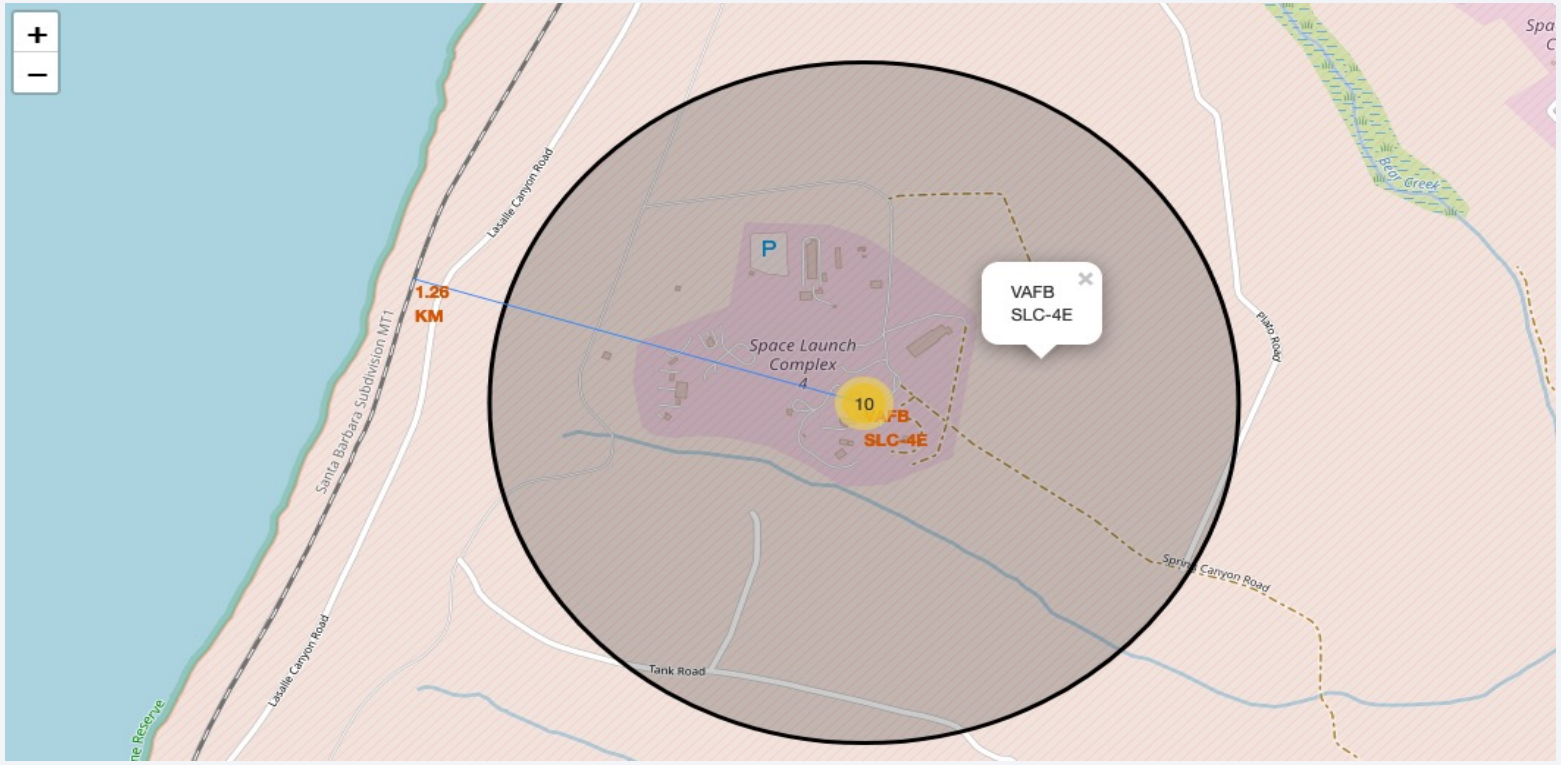


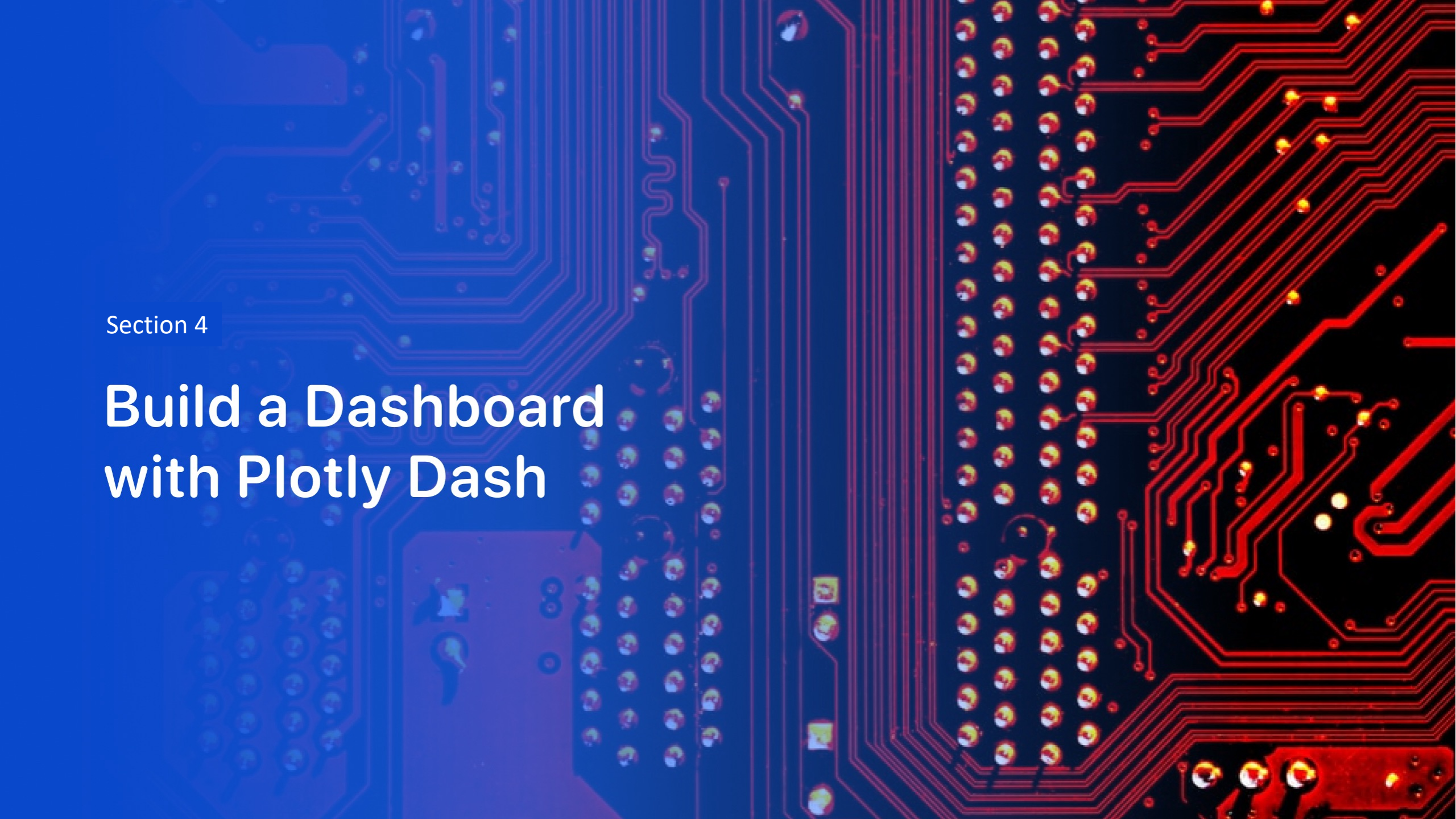
Folium Map Screenshot: markers with icons

- Folium map with markers, that include icon info property – green in case of positive outcome of the launch, red in case of negative one.

Folium Map Screenshot: line to the closest railway

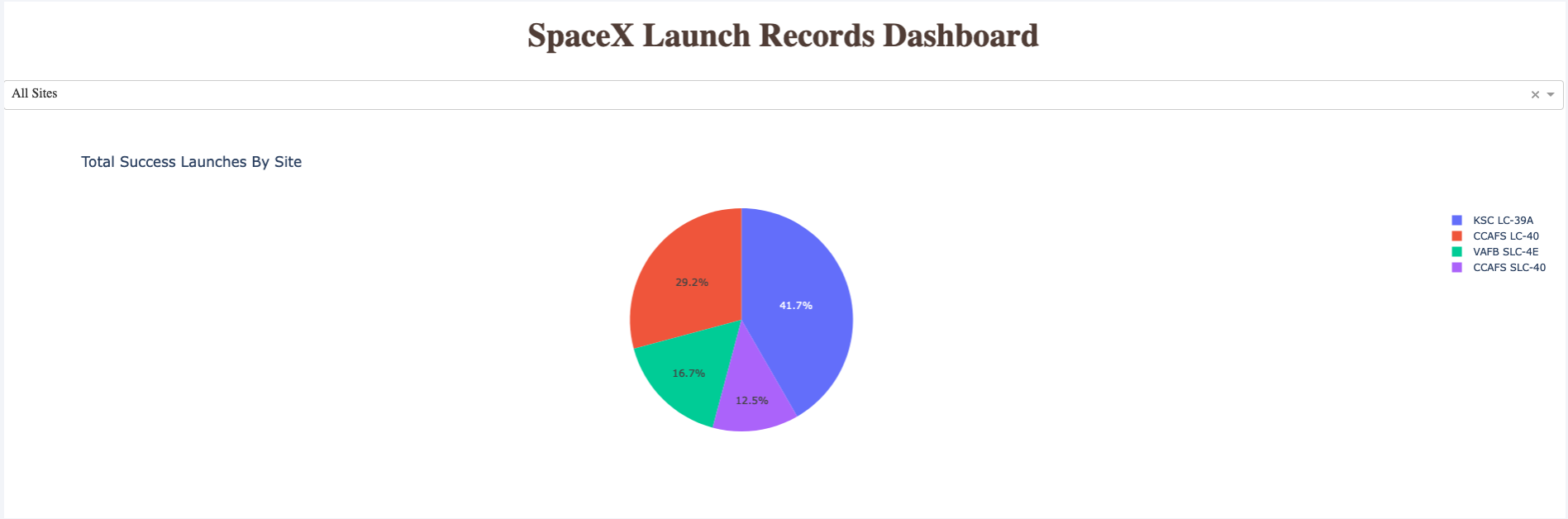
- Folium map with line depicting the distance (1.26km) from VAFB SLC-4E site to the closest railway point.




Section 4
Build a Dashboard with Plotly Dash

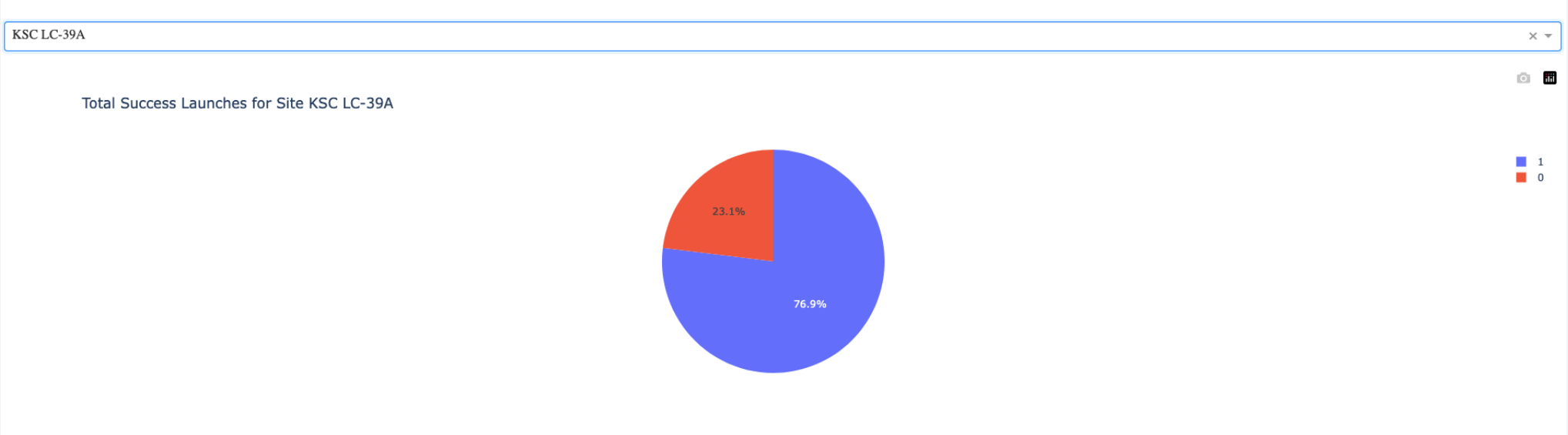
Dashboard outputs: successful launches by site

- The site with the highest number of successful launches is KSC LC-39A (41.7%), followed by CCAFS LC-40 (29.2%), VAFB SLC-4E (16.7%) and CCAFS SLC-40 (12.5%).



Dashboard outputs: success ratio of KSC LC-39A site

- Around 3/4 of launches from KSC LC-39A succeeded, which is very optimistic in a long run.

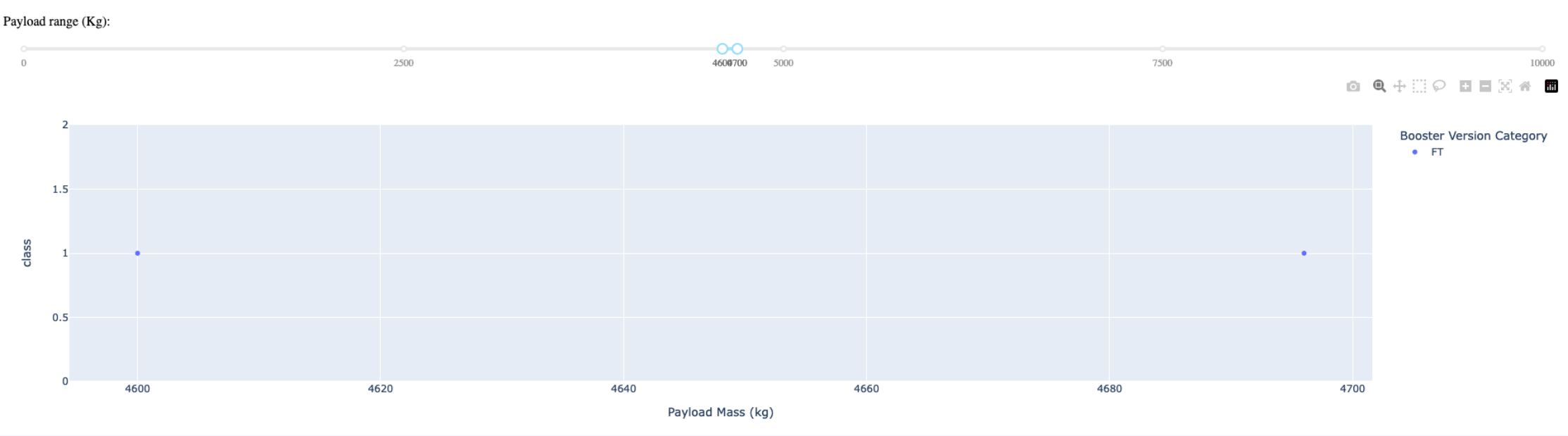


Dashboard outputs: Launch outcomes based on Booster version

- Technically, the most successful booster version seems to be B5 (1/1), but as it was rarely used, we cannot simply generalize. In fact, FT booster looks as the most reliable (15/23, approx. 65,21%).
- Greater failure rate for lower payloads can be explained by early testing launches with low cargo.

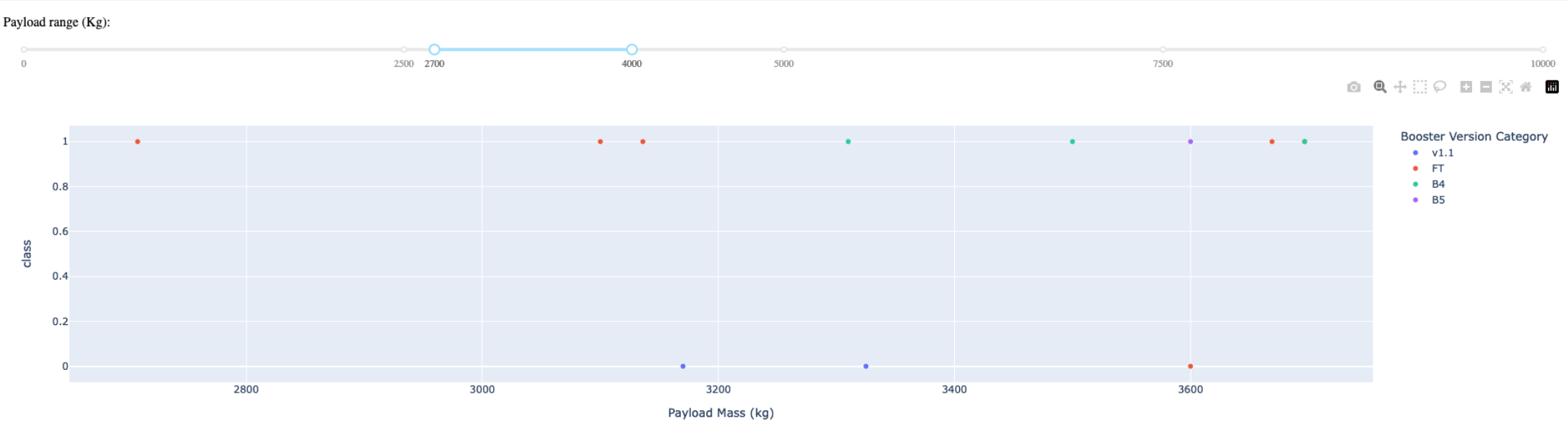
Dashboard outputs: Launch outcomes based on Payload range

- One can easily select payload interval narrow enough (e.g., 4600-4700kg), so that all launches within are successful. Though, we cannot simply generalize anything by that due to small sample size.



Dashboard outputs: Launch outcomes based on Payload range

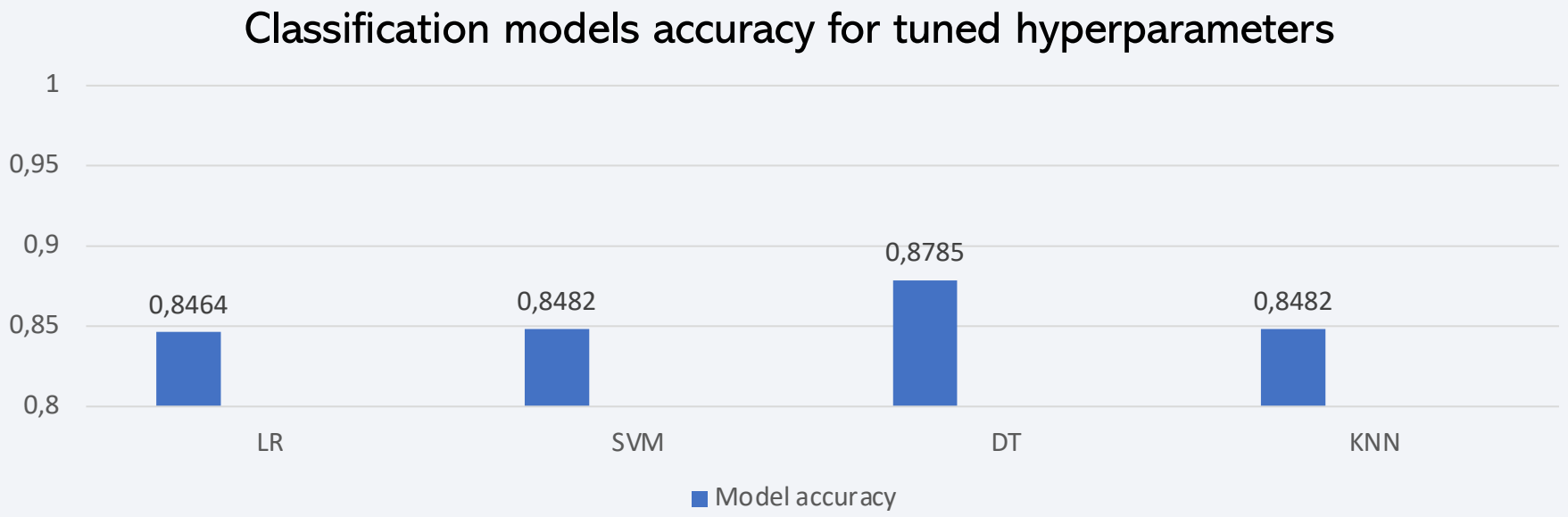
- The promising range seems to be around 2700-4000kg, where most of the launches succeeded (8/11).



Section 5

Predictive Analysis (Classification)

Classification Accuracy



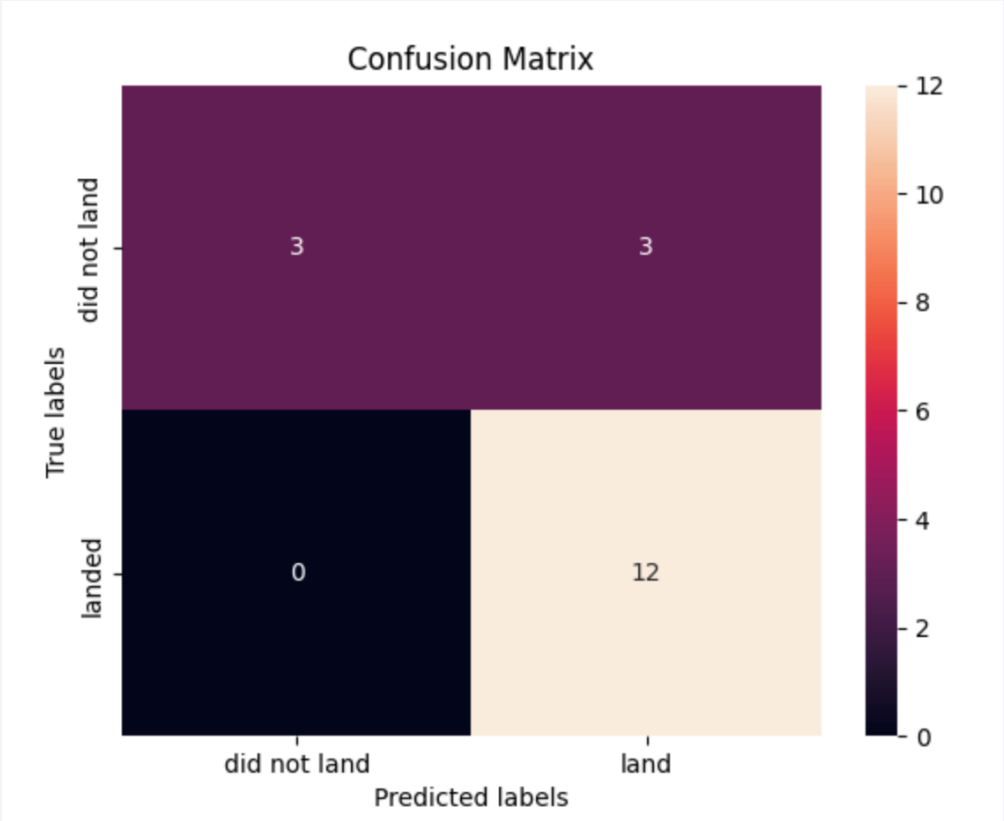
However, accuracy for given test data is 0.8333 (83,33%) for all 4 models!

<u>Best tuned hyperparameters:</u>

- Logistic regression (LR) – {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs‘};
- Support vector machine (SVM) – {'C': 1.0, 'gamma': 0.031622776601683793, 'kernel': 'sigmoid‘};
- Decision trees (DT) – {'criterion': 'entropy', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'random‘};
- K-neerest neighbours (KNN) – {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}

Confusion Matrix

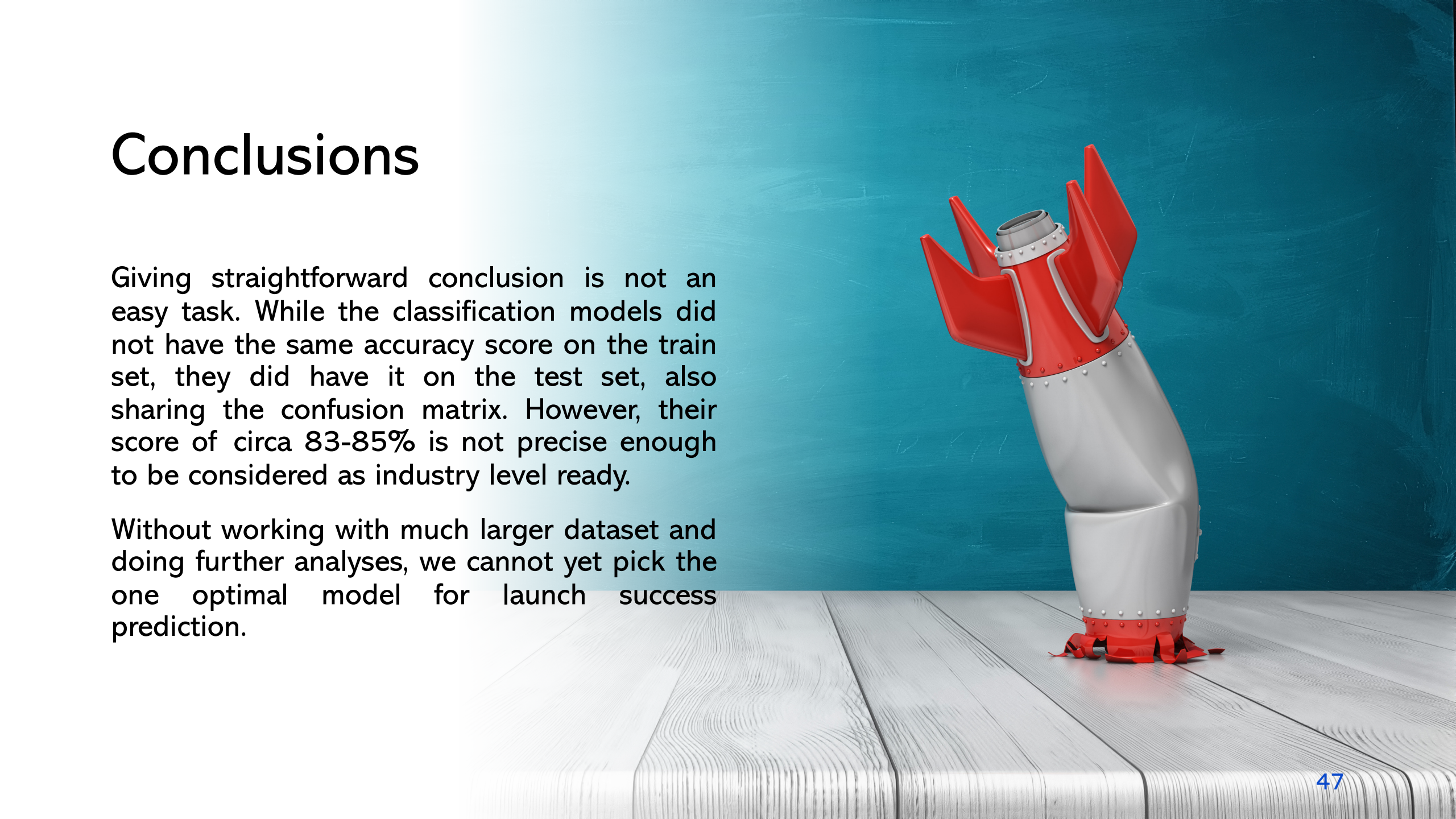
- As the accuracy for the test set of data is identical for all 4 models, this confusion matrix represents all of them.
- Positives: there are no false negatives – meaning that no launches, which were predicted to fail succeeded.
- Negatives: there are 3 cases of false positives, which are even more dangerous than false negatives – as the predicted successful launches in fact failed.
- Only 18 datapoints in test set – models' comparison and any conclusions based on them should be interpreted with the benefit of doubt.



Conclusions

Giving straightforward conclusion is not an easy task. While the classification models did not have the same accuracy score on the train set, they did have it on the test set, also sharing the confusion matrix. However, their score of circa 83-85% is not precise enough to be considered as industry level ready.

Without working with much larger dataset and doing further analyses, we cannot yet pick the one optimal model for launch success prediction.



Appendix

https://github.com/juraj-hudak2/IBM_Data_Science_Professional_repo.git

Github repository with all my project jupyter & python files

Thank you!