IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Marco Molitor
March 24th 2023

Outline

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

Executive Summary

- Utilize Supervised Learning Techniques to predict Landing Success of SpaceX rockets
- All utilized Learners come to similar result accuracy: 5 in 6 landings are correctly predicted as success or failure
- 1/6 of all events is mispredicted as a landing when, in fact, it did not land
- Our model gives us great confidence for when a landing is predicted as a landing failure
- However, 20% of all cases predicted as landing do not land – caution is recommended

Introduction

- Motivation: Set up a possibly viable business model for SpaceY
- Need to learn about business competitors, esp. SpaceX
- Want to answer the following question

  “Can we predict whether a SpaceX landing is successful?”
- If so, we can find ways to improve on them (or at least best against them)

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected via webscraping Wikipedia page for Falcon 9 Launches, then parsed via BeautifulSoup
- Perform data wrangling
  - Some missing data was mean-imputed, data were label- or one-hot-encoded to ensure no valuable information was lost
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Train-Test-Validation of several supervised Learners & inference

Data Collection

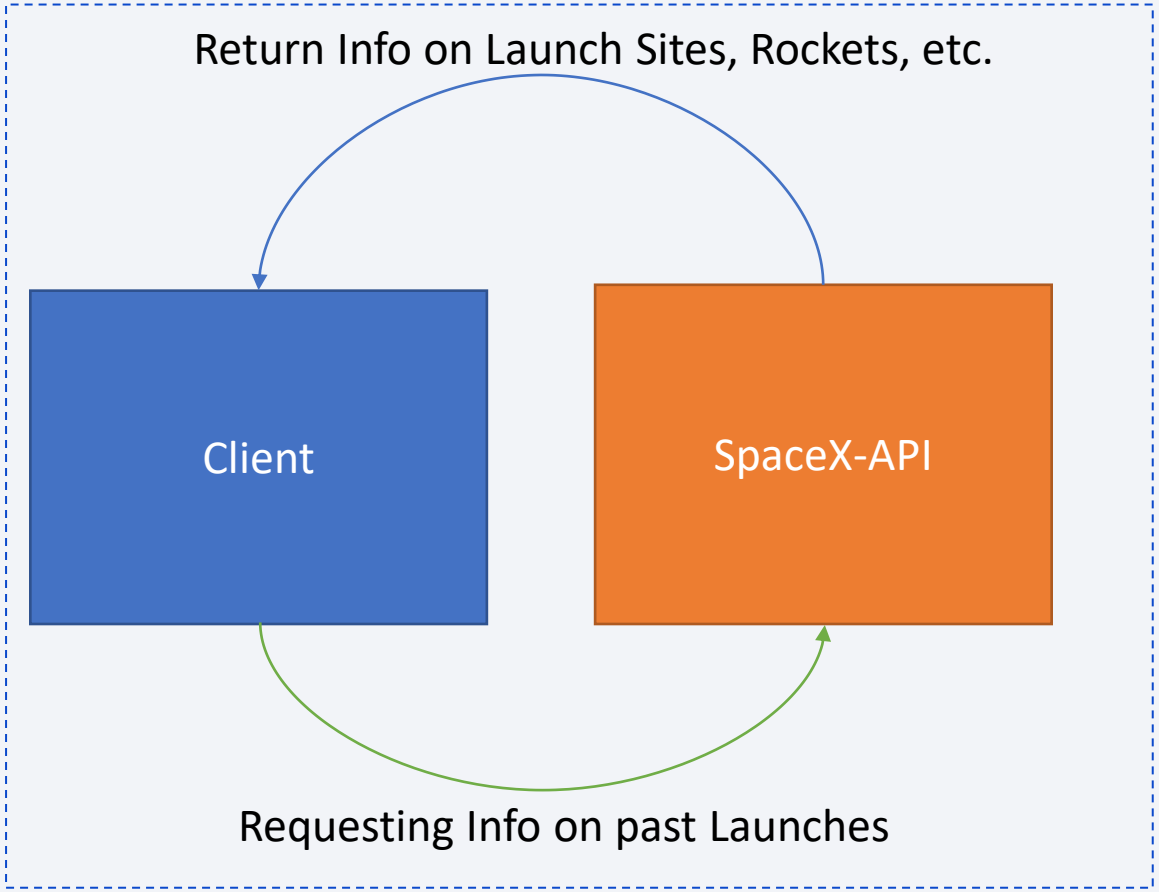
- Data was collected from Wikipedia and SpaceX-Api
- Webscraping both sides and parsing raw data with an HTML-Parser I was able to create a structured data set
- On this data, some data cleaning and preprocessing were performed, i.e. imputation and encoding
- Visualization was undertaken using Seaborn for Python-integrated charts, Folium for displaying geospatial data, and Plotly Dash for Interactive Dashboards
- ML Algorithms were then trained on the Data to predict Launch Successes

Data Collection – SpaceX API

- REST-API of SpaceX was called to gather information regarding their rockets, launch sites, orbital strategies, and other measures

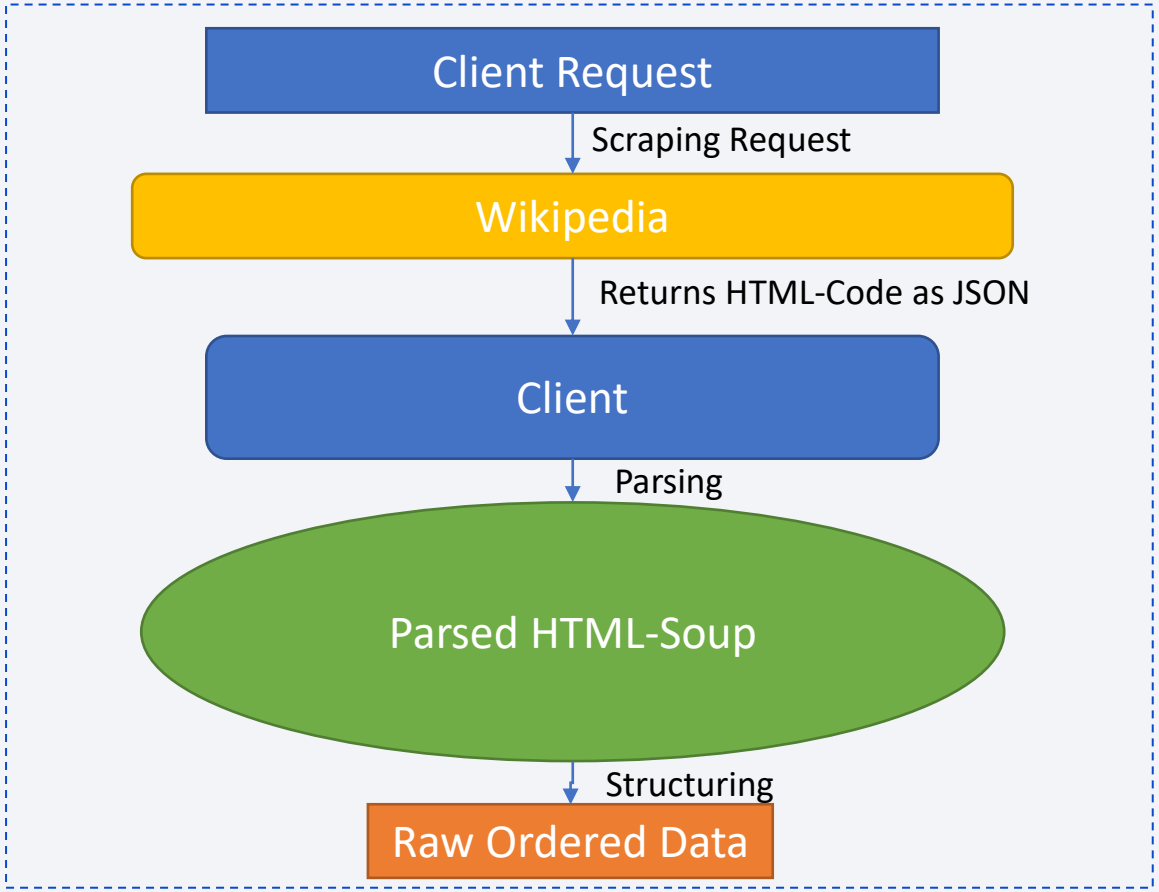
- GitHub:

https://github.com/maggomor/IBM-Certification-Repository/blob/main/01-Data-Collection/DATA_COLLECTION_SPACEX.ipynb



Data Collection - Scraping

- Scraping Rocket Launch Tables from Wikipedia, HTML-Parsing with BeautifulSoup, Data Cleaning and structuring to bring data in raw tabular form
- GitHub for Webscraping: https://github.com/maggomor/IBM-Certification-Repository/blob/main/01-Data-Collection/SpaceX_Webscraping.ipynb



Data Wrangling

- Missing Data was mean-Imputed
- A Success-Category was coded as result of various different origin categories with a Mean-Success-Rate of 2/3
- Data-Wrangling on GitHub:

https://github.com/maggomor/IBM-Certification-Repository/blob/main/01-Data-Collection/DATA_WRANGLING.ipynb

EDA with Data Visualization

- Explorative Data Analysis using Scatter- and Barplots to visualize birelational Associations
- GitHub: https://github.com/maggomor/IBM-Certification-Repository/blob/main/02-Data-Wrangling-And-EDA/DATAVIZ_NOTEBOOK.ipynb

EDA with SQL

- Analysing the Amount of different patterns observed in relational database for SpaceX data
- Displaying Failures vs Successes, retrieve list of boosters carrying maximum payload, rank landing outcomes by amount, most often: success
- GitHub: https://github.com/maggomor/IBM-Certification-Repository/blob/main/02-Data-Wrangling-And-EDA/EDA_SQL.ipynb

Build an Interactive Map with Folium

- Interactive map created to gather geological information on SpaceX launching sites, added markers distributing successes and failures across Launching Sites, example of distance of one launching site to infrastructure surroundings
- Success-Markers were added to see if one launching site has systematically better impact on launches, surroundings were located by distance to see how launch sites interact with their environment: usually near the coast, away from big cities and highways, proximity to railways likely due to delivery of mission-critical goods
- GitHub: https://github.com/maggomor/IBM-Certification-Repository/blob/main/03-Data-Visualization/Folium_Launch_Sites.ipynb

Build a Dashboard with Plotly Dash

- Created Pie-Chart and Scatter-Plot on Interactive Plotly Dashboard to visualize Landing Success Rates by Launch Site as well as visualizing correlation between payload and success rate, marked by booster rocket
- Interactivity allows human-readable information-gathering in easy visual form to recognize visual patterns among data
- Github: https://github.com/maggomor/IBM-Certification-Repository/blob/main/03-Data-Visualization/spacex_dash_app%20(1).py

Predictive Analysis (Classification)

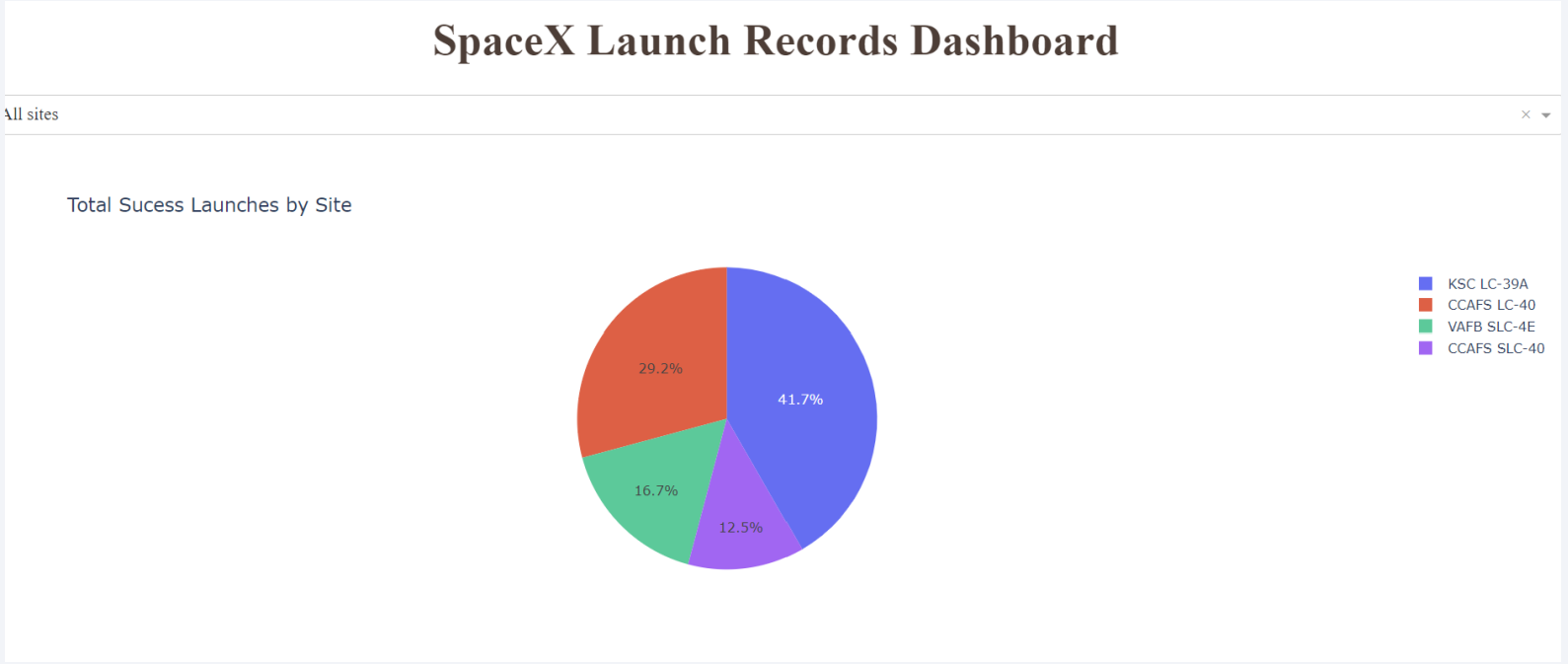
- A Cross-Validated Grid-Search Algorithm was implemented on different Learners to facilitate optimal hyperparameter tuning
- Linear as well as Non-Linear Learners were employed, in particular:

Logistic Regression	Support Vector Machine
Decision Tree Classifier	K-Nearest-Neighbours-Classifier
Random Forest Classifier	Voting-Classifier among other five

GitHub: https://github.com/maggomor/IBM-Certification-Repository/blob/main/04-Machine-Learning/ML_Predictions.ipynb

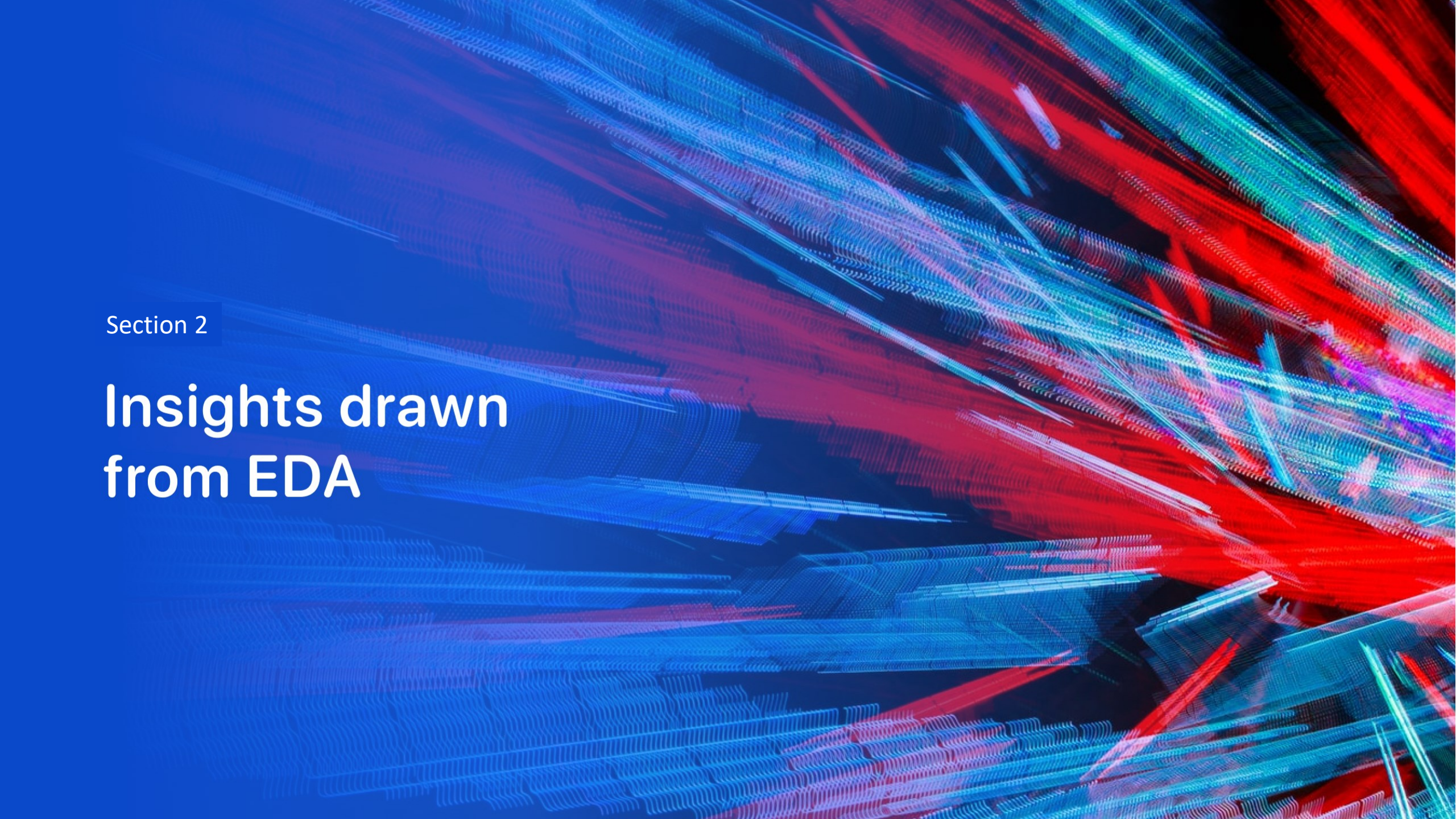
Results

- Exploratory data analysis results: Observations vary substantially between payload, boosters, orbit, Customer, and Launch-Sites
- Predictive results: 5 out of 6 out-of-sample observations are correctly classified



Section 2

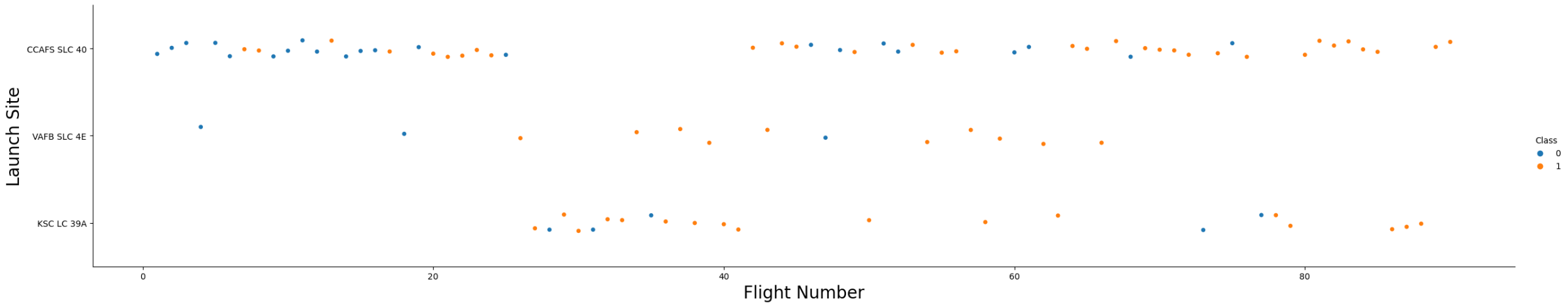
Insights drawn from EDA



Flight Number vs. Launch Site

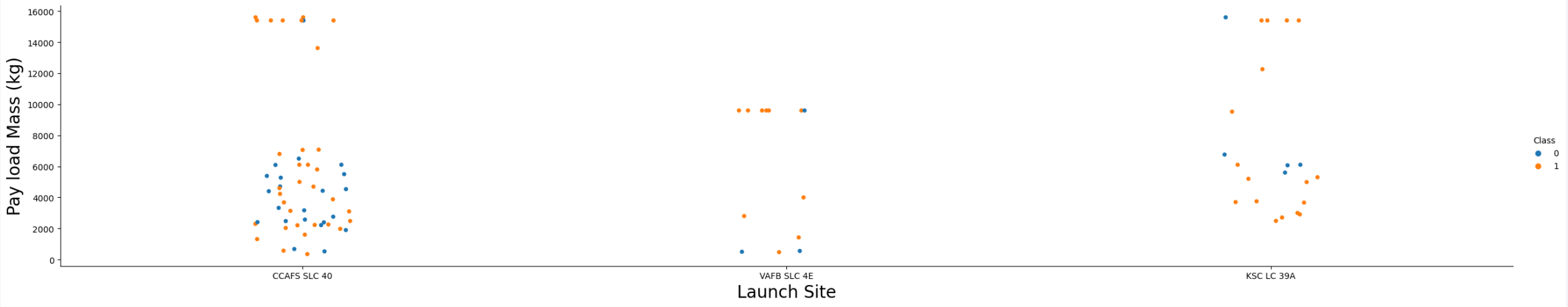
- Show a scatter plot of Flight Number vs. Launch Site

=> Initial Flights located at CCAP SLC-40, later other sites joined



Payload vs. Launch Site

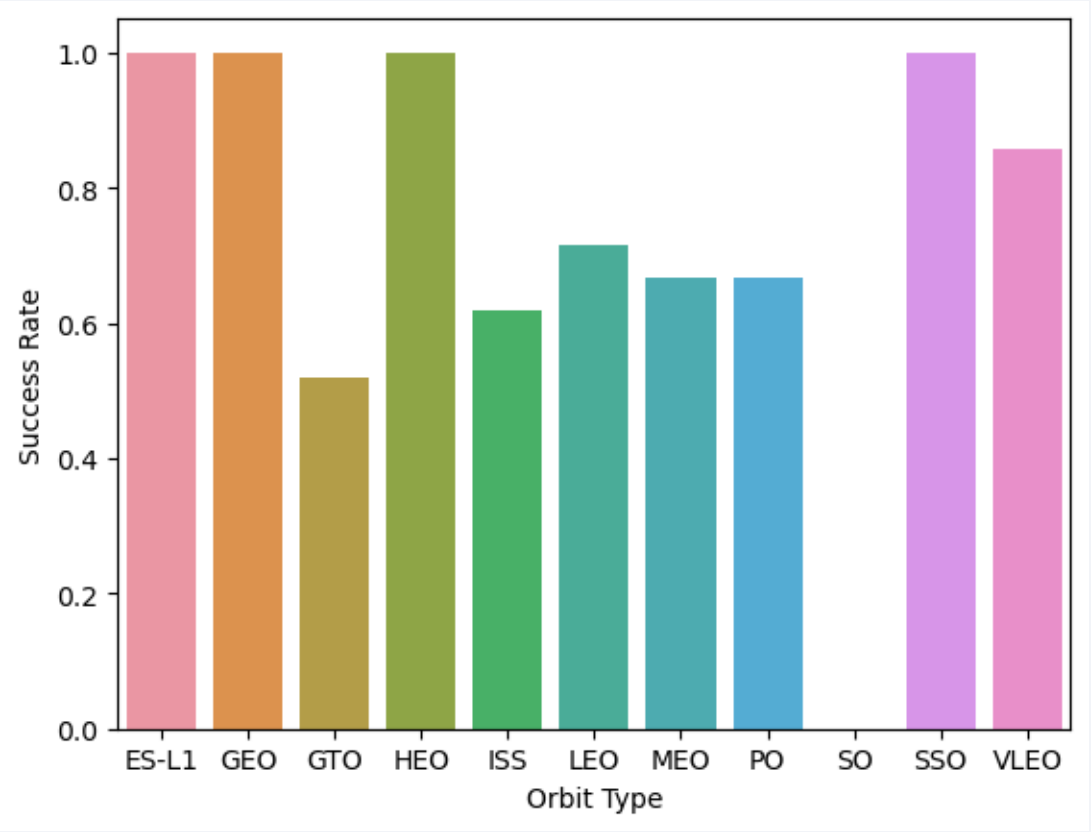
- Show a scatter plot of Payload vs. Launch Site
- Pattern shows concentration among CCAFS SLC 40, especially for heavy payload



Success Rate vs. Orbit Type

- Success Rate varies greatly depending on orbital type

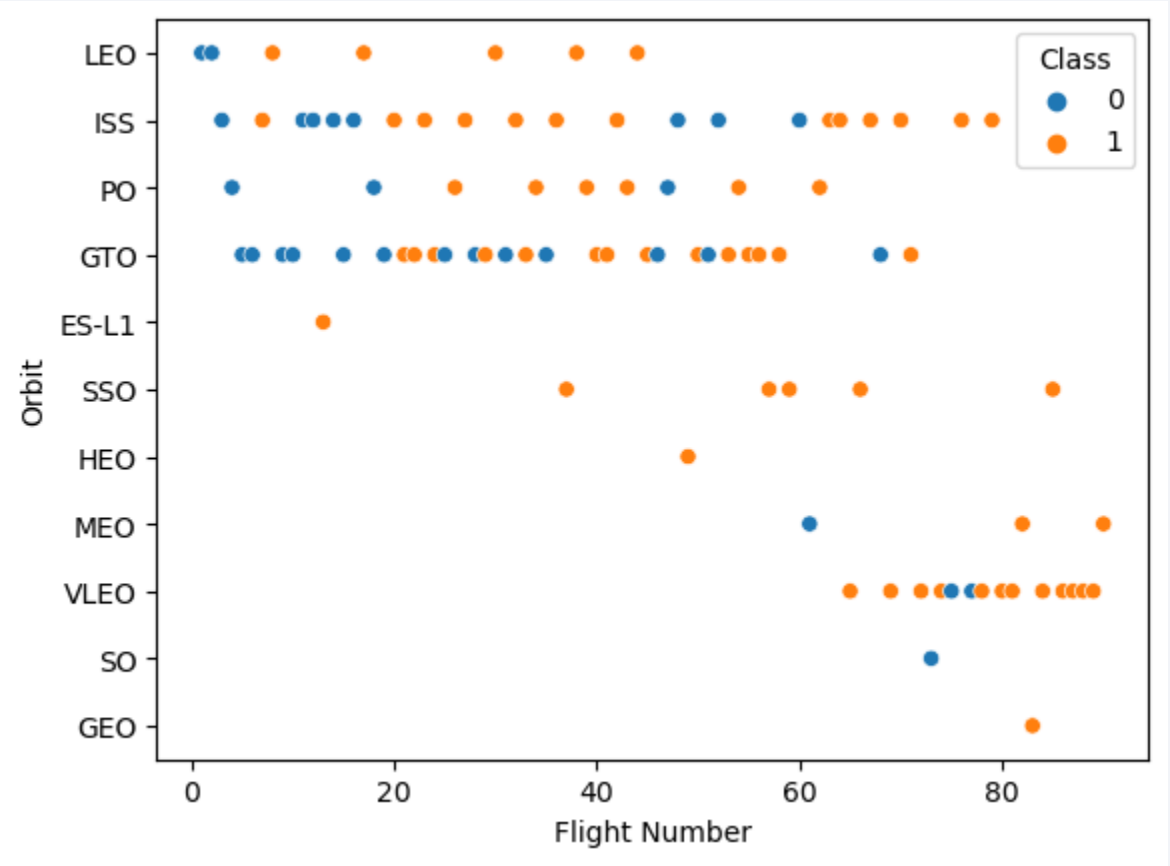
- Certain easier ones like GEO (low-earth-orbit) have 100% success rate, whereas others have much more meager success rate



Flight Number vs. Orbit Type

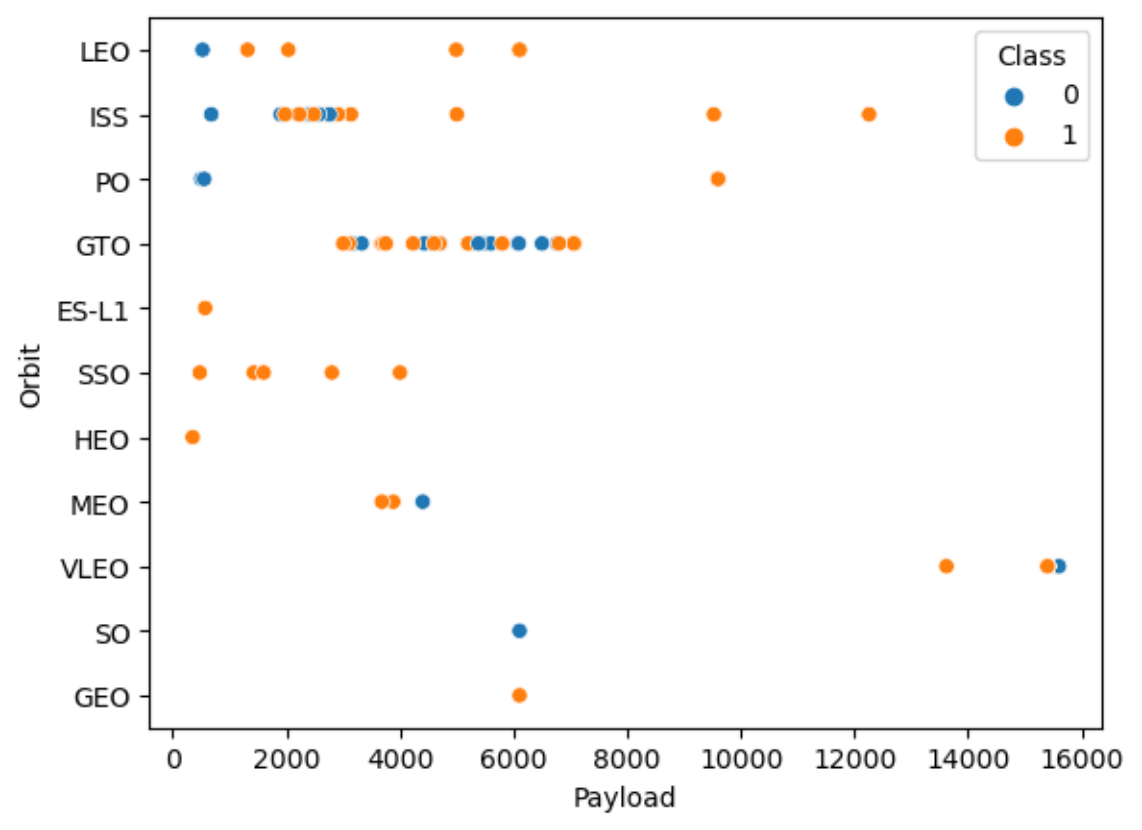
- Certain orbits (e.g. VLEO – Very Low Earth Orbit) only very late, main business in ISS / GTO (space station / sattelite orbit)

- Many of these are successes, especially as flight number increases (=> flights in later years)



Payload vs. Orbit Type

- Plotting Payload-VS-Orbit
- Very heavy loads (>10k) only in low earth orbit, very low payload also for further flights (SSO). Medium range often for GTO (satellite outbringing). ISS mostly low payload, only recently trust established and also heavier payloads to ISS (all successes)

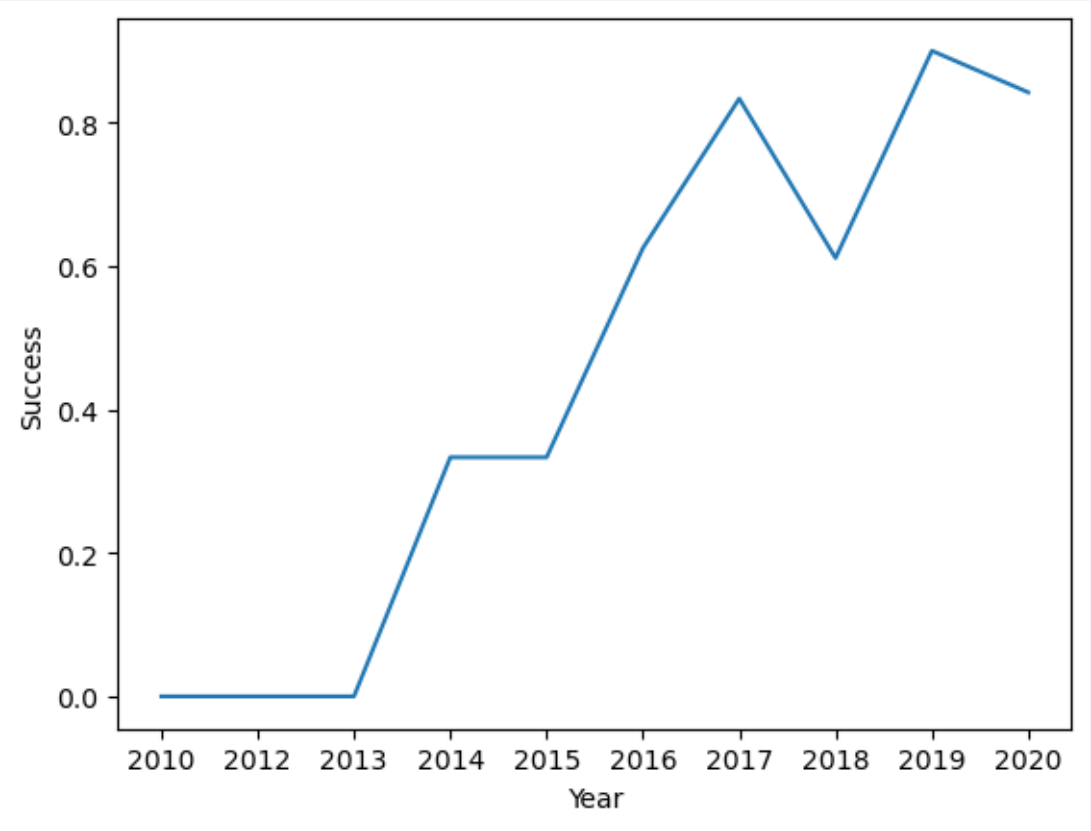


Launch Success Yearly Trend

- Success Rate consistently rises over the years

- Over time, SpaceX has consistently been learning, especially initial starts often incorporated intentional failures to gather failure-data in their systems

=> Apparently helped as success rate increased vastly



All Launch Site Names

- Unique launch sites can be seen to the right
- SQL Query to get all distinct values from database table

```
%%sql
SELECT DISTINCT Launch_Site FROM SPACEXTBL;
```

[30]

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

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

Launch Site Names Begin with 'CCA'

- Example of five records for “CCA” launch site
- Queried 5 records from table that locates all launch-sites that begin with “CCA”

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

[31] Python

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

Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit
18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO
15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)
07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)
00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)
15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)

Total Payload Mass

- Total Payload carried for NASA is 48213kg
- Rows where “NASA” was listed as customer were summed up along the payload-column

Task 3

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

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE
CUSTOMER LIKE '%NASA (CRS)%';
```

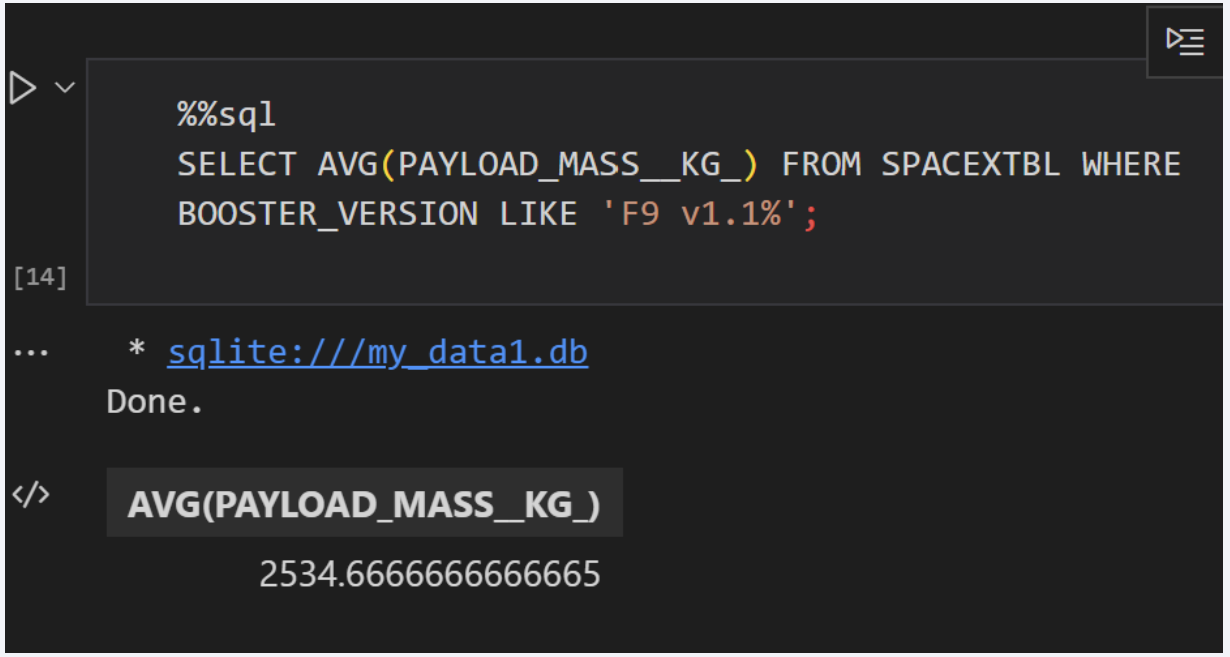
[13] Python

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

SUM(PAYLOAD_MASS__KG_)
48213

Average Payload Mass by F9 v1.1

- Average payload carried by F9 v1.1 booster is 2534.67kg
- Average payload was calculated for rows where booster version matched F9 v1.1



First Successful Ground Landing Date

- First successful landing outcome on ground pad took place 22nd December 2015
- Searching for “MIN(DATE)” does not work as Date is in DD-MM-YYYY format and thus not interpreted correctly as a Date by SQL
- SQL would give out 01-05-2017 as first successful ground pad landing year which would be 1.5 years too late

```
%%sql
SELECT * FROM SPACEXTBL WHERE
LANDING_OUTCOME LIKE '%Success (ground pad)%'
ORDER BY DATE DESC;
-- Shows the actual correct date because
-- DATE is here misinterpreted and not understood
--as daytime as it is in DD-MM-YYYY format
--Therefore, min-command does not work
```

Python

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_
22-12-2015	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034
19-02-2017	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490
18-07-2016	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257
15-12-2017	15:36:00	F9 FT B1035.2	CCAFS SLC-40	SpaceX CRS-13	2205
14-08-2017	16:31:00	F9 B4 B1039.1	KSC LC-39A	SpaceX CRS-12	3310
08-01-2018	01:00:00	F9 B4 B1043.1	CCAFS SLC-40	Zuma	5000
07-09-2017	14:00:00	F9 B4 B1040.1	KSC LC-39A	Boeing X-37B OTV-5	4990

Successful Drone Ship Landing with Payload between 4000 and 6000

- Boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

- The F9 managed that landing in different specifications

```
%%sql
SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ > 4000 AND
PAYLOAD_MASS__KG_ < 6000 AND
LANDING_OUTCOME LIKE '%DRONE%';
```

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

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

Total Number of Successful and Failure Mission Outcomes

- Total number in this dataset for F9 launches had 100 successes, only one failure
- Subquery was used to select counts for observations that where failures and succeses

List the total number of successful and failure mission outcomes

```
%%sql
SELECT COUNT(ORBIT) AS FAILURE_COUNT,
(SELECT COUNT(ORBIT) AS SUCCESS_COUNT FROM SPACEXTBL
WHERE MISSION_OUTCOME NOT LIKE '%Failure%') AS SUCCESS_COUNT
FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE '%Failure%';
```

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

FAILURE_COUNT	SUCCESS_COUNT
1	100

Boosters Carried Maximum Payload

- Boosters that have carried maximal payload can be seen to the right
- Query was used using subquery for filtering by payload

```
%%sql
SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_)
FROM SPACEXTBL);
```

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

- Failed Landings in drone ship, their booster versions, and launch site names in 2015

- Matching year on 2015 and searching for months, landing outcomes, booster versions, and launch sites

```
%%sql
SELECT substr(Date,4,2) AS Month, LANDING_OUTCOME,
BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL
WHERE substr(Date,7,4)='2015' AND
LANDING_OUTCOME LIKE '%Failure%';
```

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

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	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 between June 4th 2010 and March 20th 2017 in descending order

- Selection was performed on date and successes where counted grouped by Landing-Outcome
- The most common outcome is a regular success

```
%%sql
SELECT COUNT(LANDING_OUTCOME) AS SUCCESS_COUNT,
LANDING_OUTCOME FROM SPACEXTBL
WHERE DATE > '04-06-2010' AND
DATE < '20-03-2017'
GROUP BY LANDING_OUTCOME
ORDER BY SUCCESS_COUNT DESC;
```

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

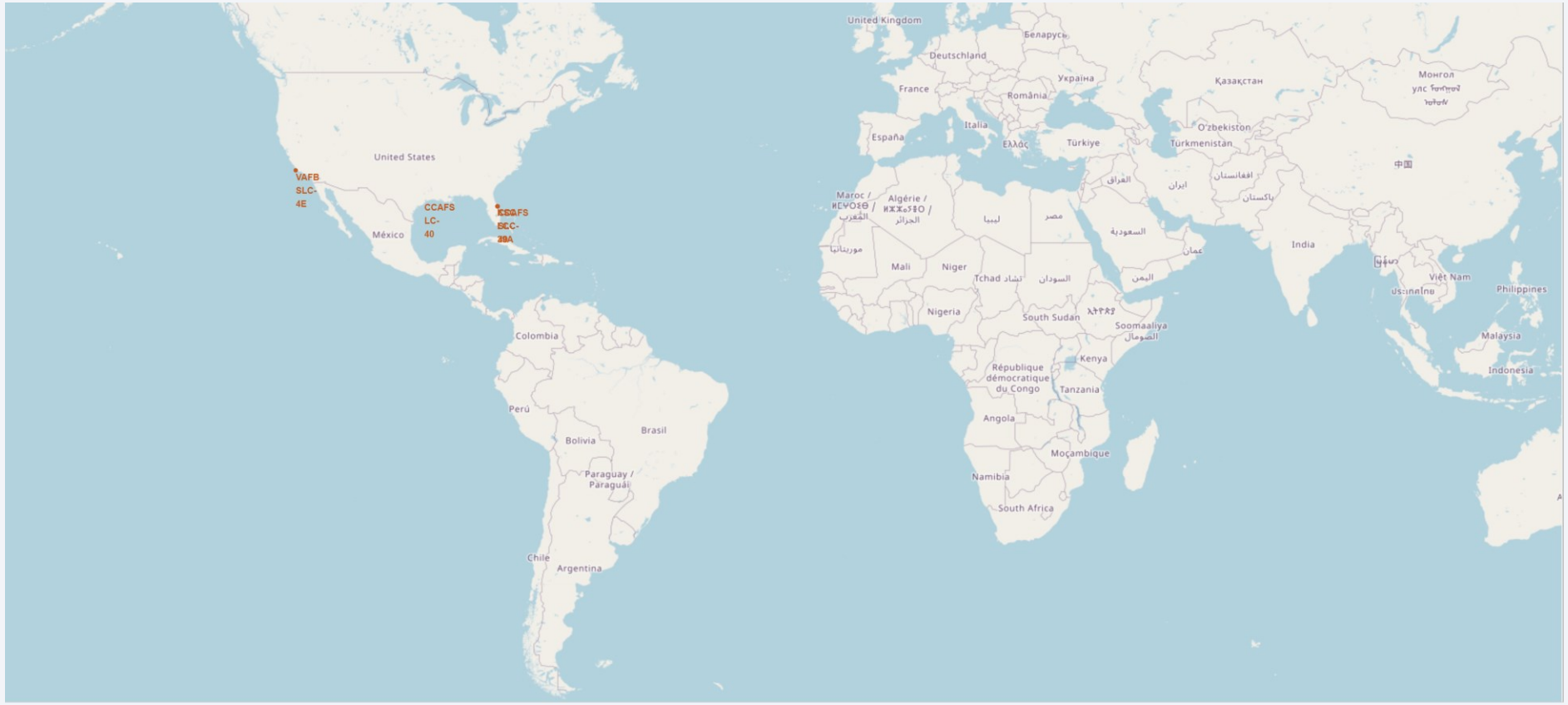
SUCCESS_COUNT	Landing_Outcome
20	Success
10	No attempt
8	Success (drone ship)
6	Success (ground pad)
4	Failure (drone ship)
3	Failure
3	Controlled (ocean)
1	No attempt
1	Failure (parachute)

Section 3

Launch Sites Proximities Analysis

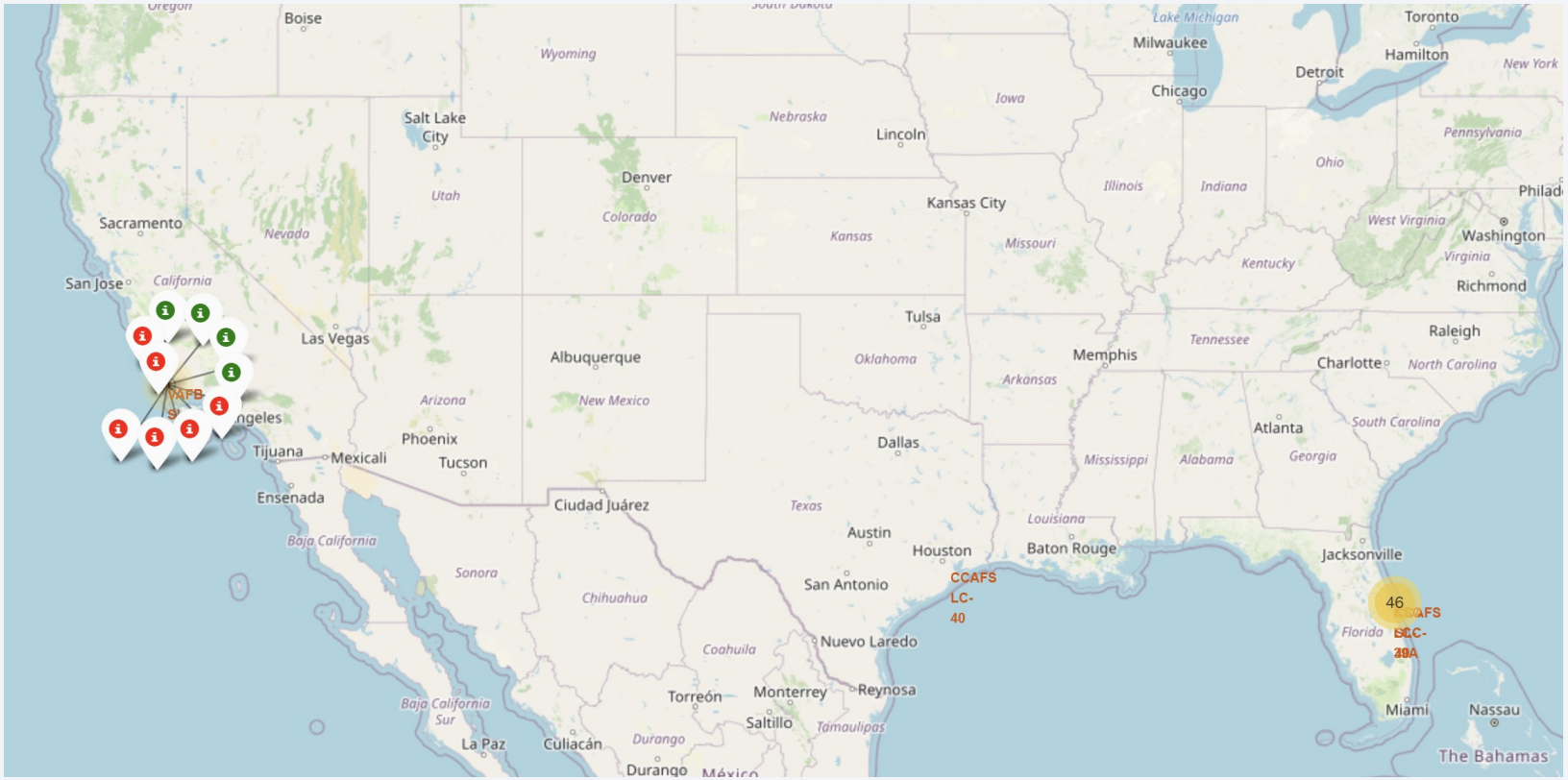
Location of SpaceX Launch Sites

- As can be seen, all SpaceX Launch sites are in coastal proximity located in the USA



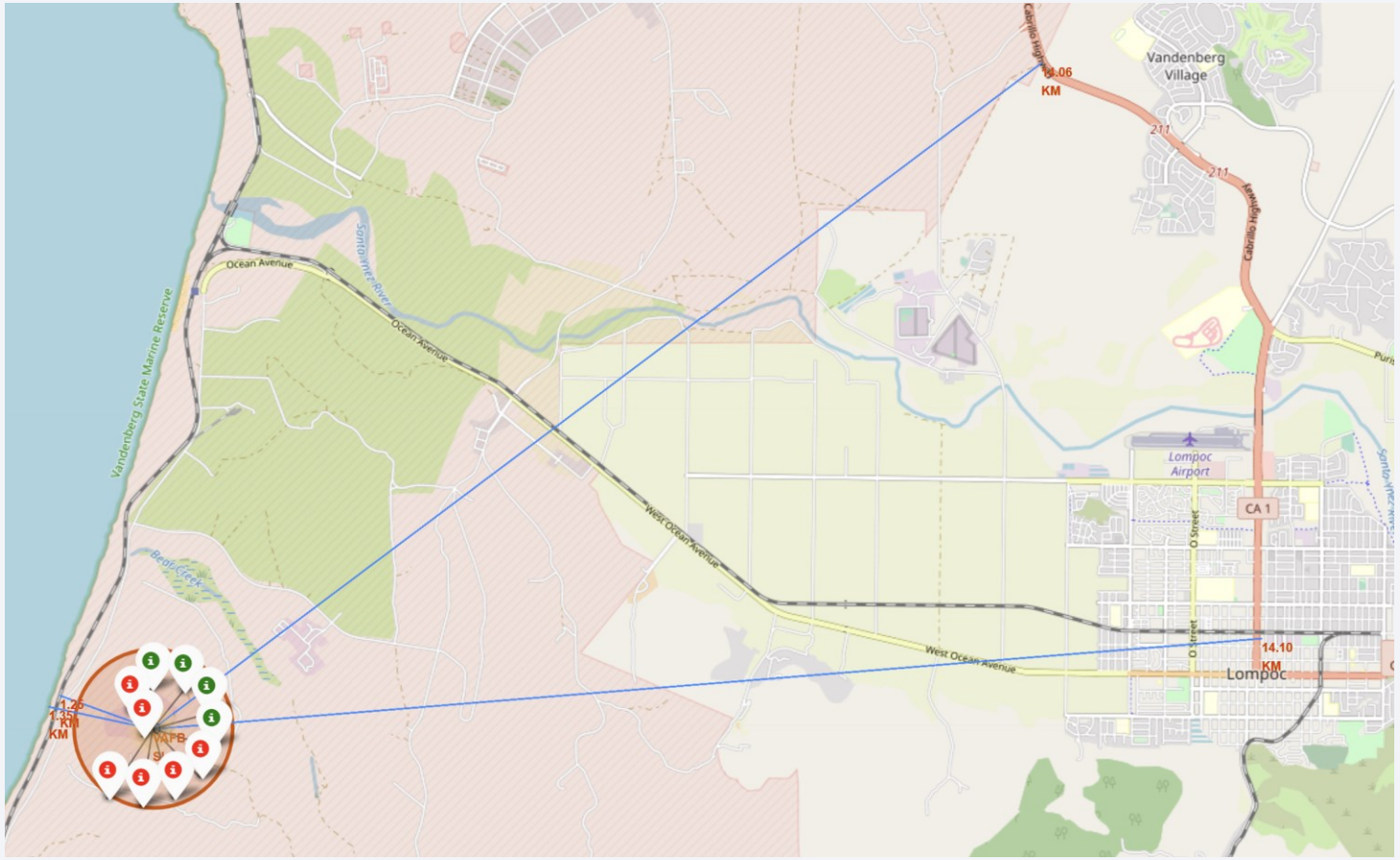
Visualizing Failures by Launch Site

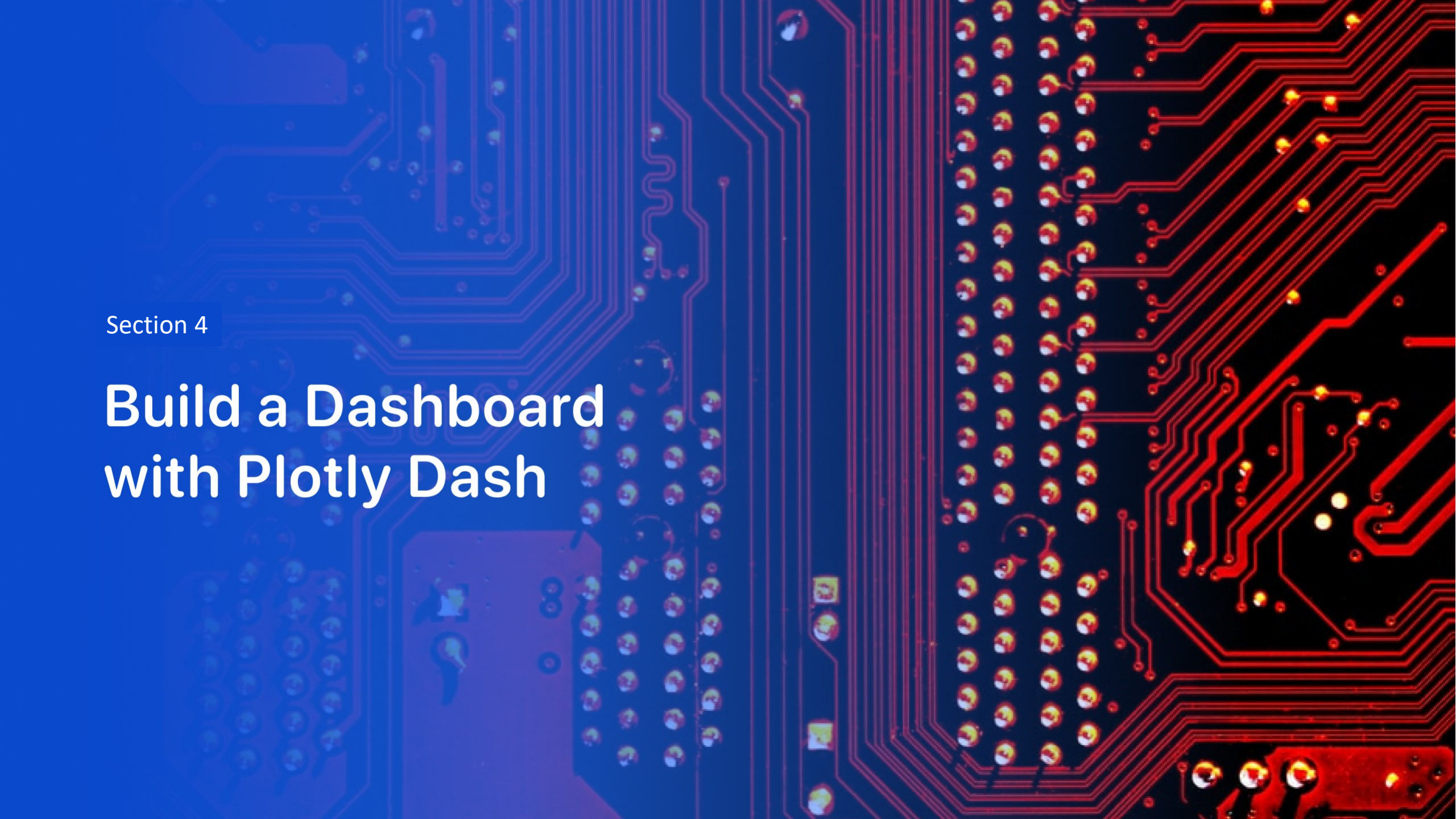
- Example: West Coast Launch Site shows mostly failure starts



Distance to Infrastructure

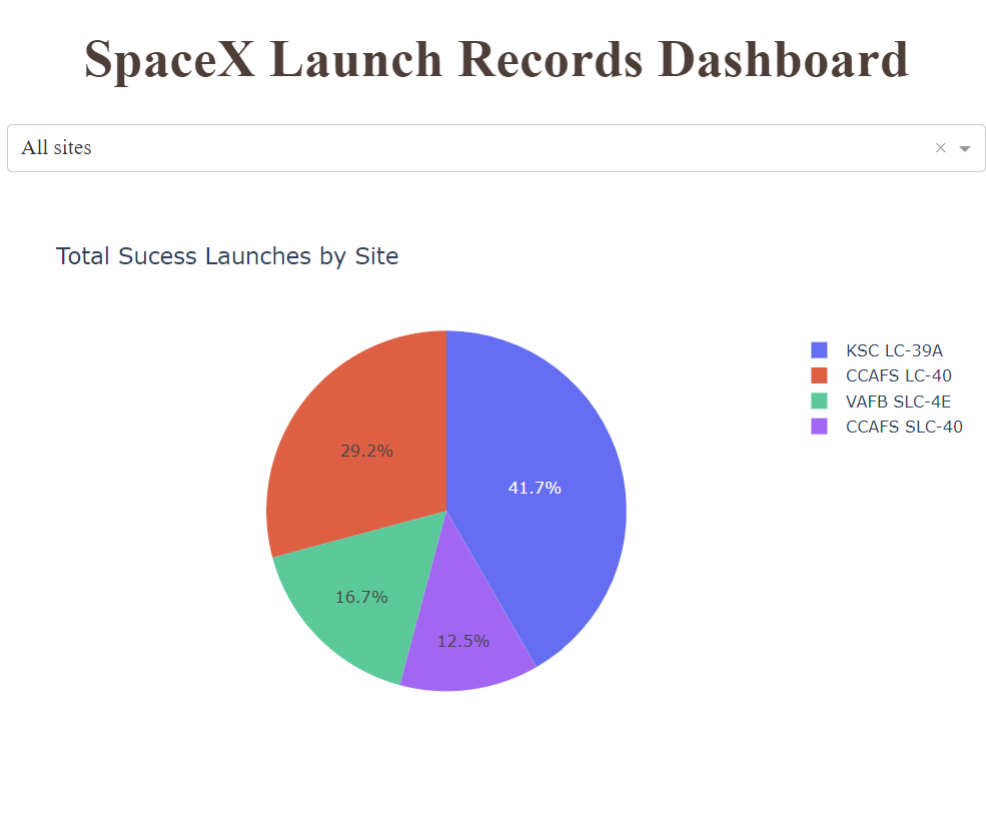
- Launch Sites away from Cities and Highways, but close to coast and railway (first two pose hazards for civilians, last two necessary infrastructure)




Section 4
Build a Dashboard with Plotly Dash

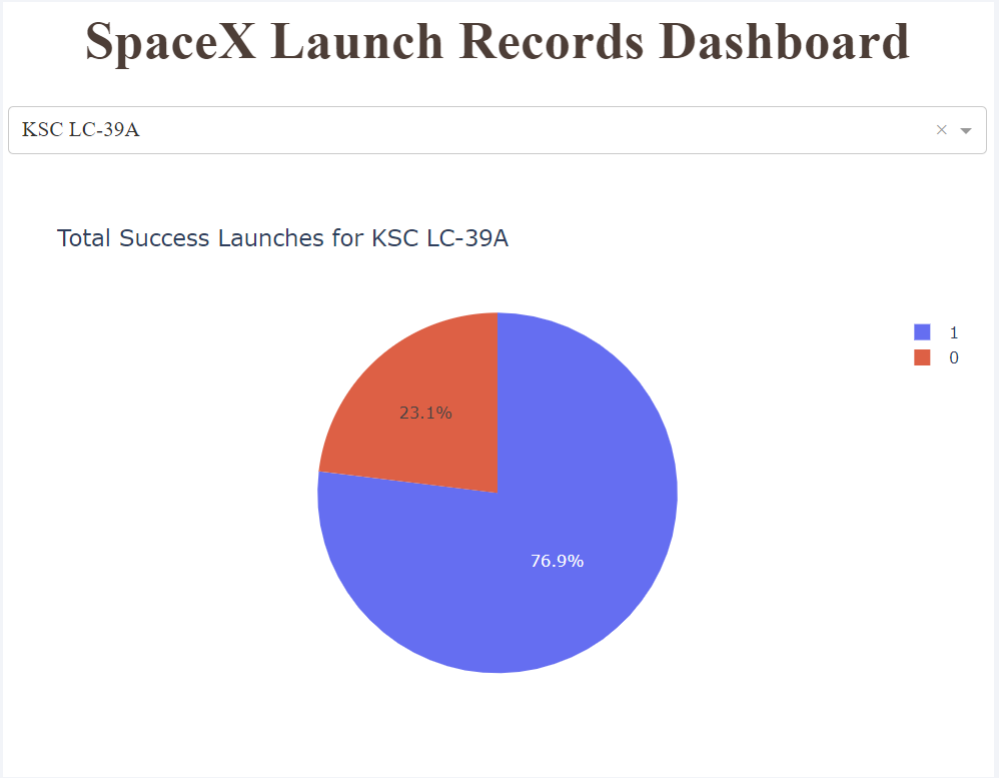
Dashboard Detailing Total Success Launches per Site

- Site KSC holds the biggest share of successful launches
- However, combined both CCAFS sites have a similar count
- The VAFB site seems to either be poorly run or an experimental test site



Successes of Site KSC LC-39A

- Per Dropdown we see the Statistics for the KSC LC-39A site
- Roughly ¾ of all launches from this site were successful



Successes by Payload

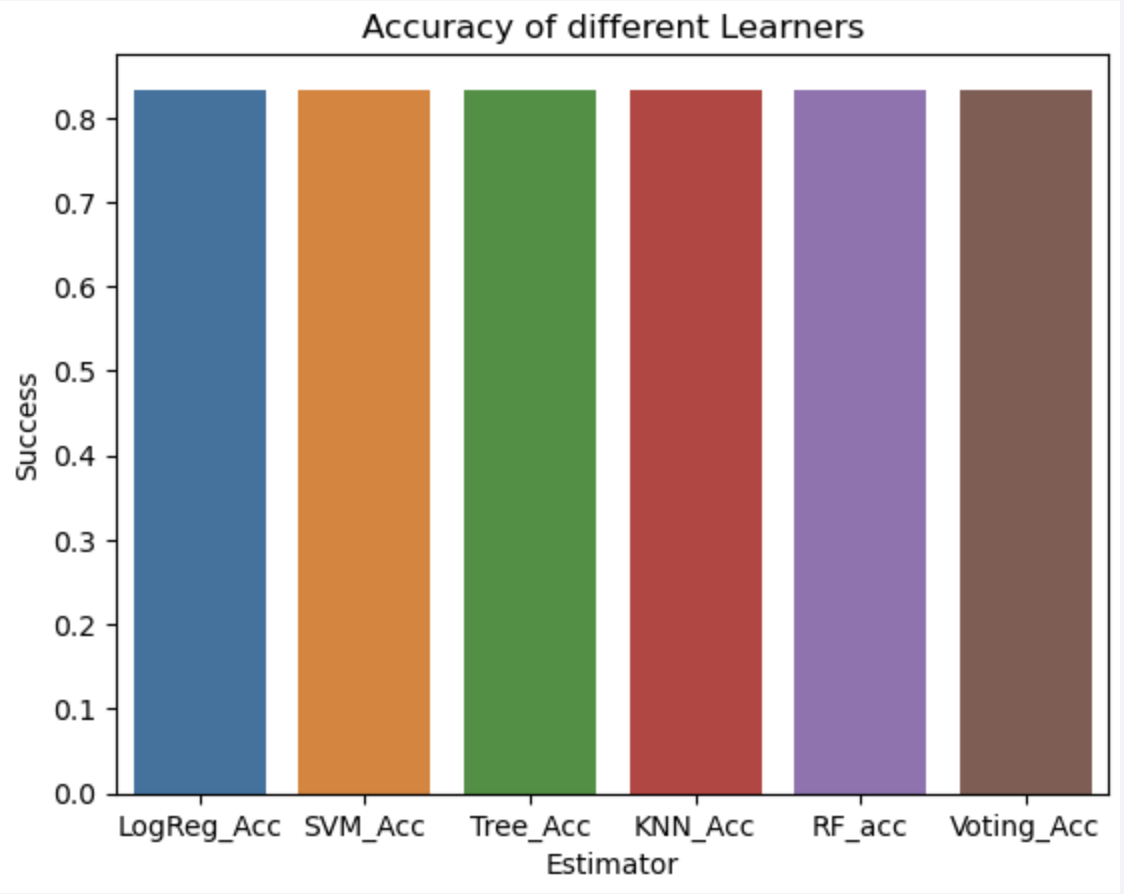
- FT Boosters most successful for intermediate range, for overall range equally

Section 5

Predictive Analysis (Classification)

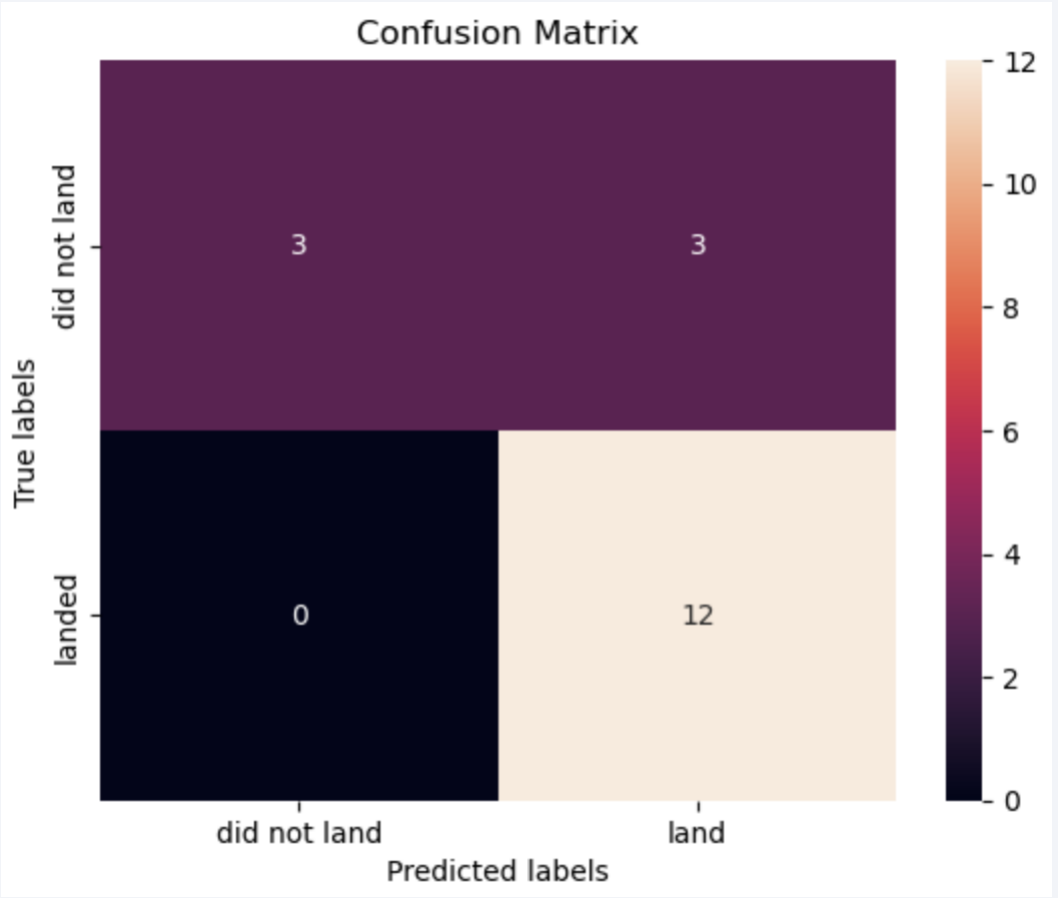
Classification Accuracy

- All Models achieve an excellent accuracy of 5/6
- Depending on the Seed, both the Tree- as well as the Random-Forest-Estimator can sometimes over- / underperform
- Better not to overinterpret; Voting Estimator most consistent at 5/6



Confusion Matrix

- 12 out of 15 flights that landed were correctly predicted to do so
- 3 out of 6 flights that did not land were correctly predicted to do so
- In contrast, overall only 1 in 6 predictions ended up being a false positive
- This speaks of SpaceX's business model: sometimes Failure is intentional



Conclusions

- Classification is equally successful among all models
- Yet, Decision-Tree-Classifier and Random-Forest-Classifier proved a little stronger dependent on the chosen Seed (i.e. open to random fluctuation)
- A Voting-Algorithm that captures the Ensemble of Different Learners seems most robust, capturing 5/6 of all correct observations
- We can be fairly confident in predictions of failures (no false negatives), but need to be somewhat more cautious with false positives (some tests were predicted successes but ended up failures)
- This may, however, be consistent with SpaceX's known strategy of sometimes intentionally crashing to gather crash test data
- In that case, they potentially crashed intentionally flights that would have landed

Appendix

- All relevant Code and Images can be found in this github repository:

https://github.com/maggomor/IBM-Certification-Repository

Thank you!