IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Łukasz Nakonieczny
08.11.2022

Outline

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

Executive Summary

- Summary of methodologies
  - We will use webscraping, API and SQL queries to obtain needed data.
  - Working within Python data ecosystem we will use Pandas for exploratory data analysis (EDA) and Matplotlib, Seaborn and Folium for data visualization.
  - Using Python Dash package, we will prepare dashboard summarizing most important of our findings during EDA stage.
  - Lastly, we will train and deploy machine learning model for purpose of judging is first stage retrieval will be successful or not.
- Summary of all results
  - For payload of the mass up to 8000 kg any launch site will do. For heavier payload we must chose between KSC LC 39A and CCAFS SLC 40.
  - Overall, most successful site is KSC LC 39A.
  - Two most successful booster versions were B1049 and B1051.
  - We found out that for our purpose the most suited machine learning classification model is based on the Decision Tree algorithm and achieved 94% accuracy.

Introduction

- Project background and context

Since commercial space race already begin, we intend to take part in it and claim as large part of the market as its is possible. We will analyze publicly available SpaceX data concerning Falcon 9 launches to gain insights on the first stage retrieval probability. We hope that this allows as to develop competitive pricing strategy.

- Problems you want to find answers

What is optimal choice of launch parameters including landing site choice and booster characteristics. The purpose is to minimize launch cost by maximizing probability of first stage retrieval.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - use of the SpaceX API and webscraping on Falcon 9 launches Wikipedia website
- Perform data wrangling
  - unnecessary data were removed from dataset and missing values were treated
  - features needed for further explorations were added
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - most relevant features were chosen and dummy variables for categorical data were introduced
  - we use set of classifications models on the data to find optimal one for prediction landing outcome
  - for fine tunning and scoring models we used sklearn library

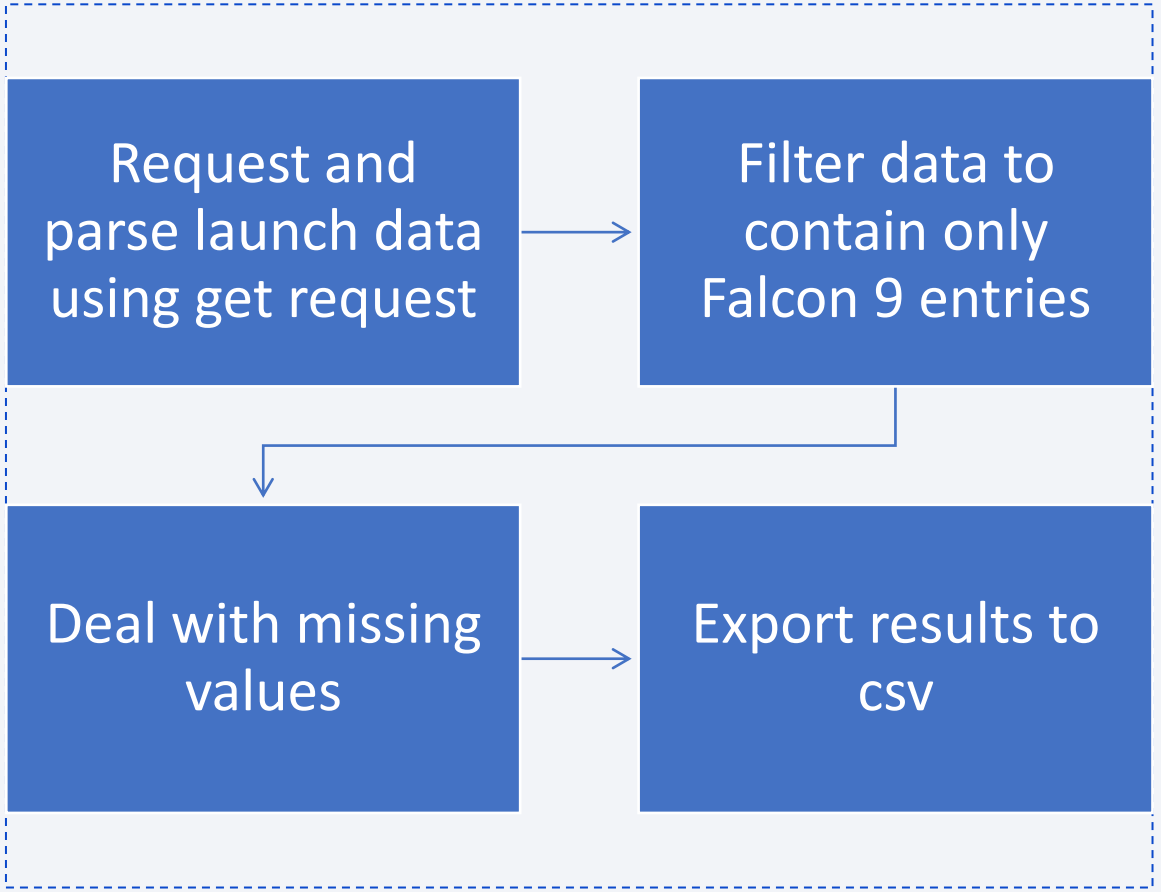
Data Collection

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

Data Collection – SpaceX API

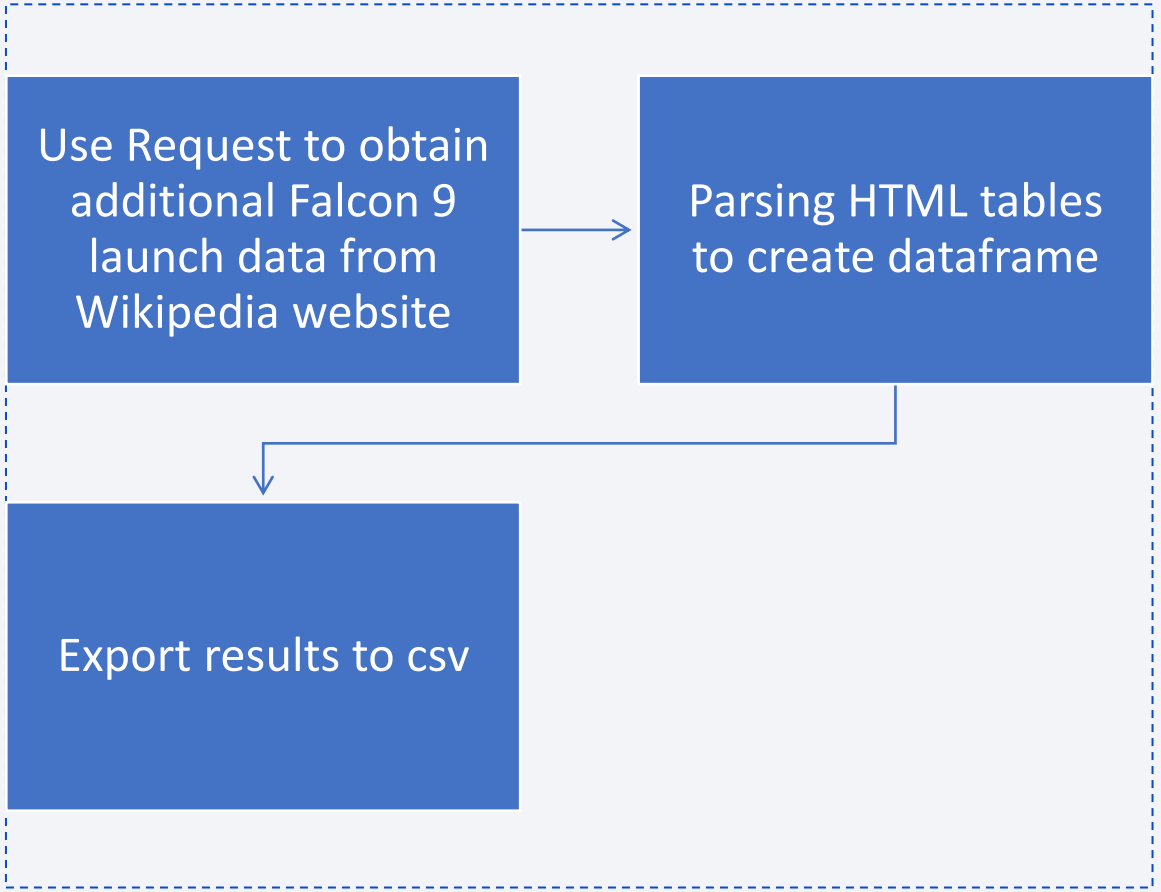
- We used SpaceX API to obtain json file containing data relevant for previous rocket launches (for purpose of further analysis we used snapshot of data provided by IBM)
- We queried additional endpoint of API to obtain data about booster version, launch site name, payload mass, outcome of the landing and detail of the landing process
- We filtered data to contains only entries related to Falcon 9 rocket model
- We replaced missing values in the payload mass column by the mean of the payload mass
- Results were exported to csv file for further use

- https://github.com/LNakonieczny/IBM_Data_Science_Capstone_Project/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

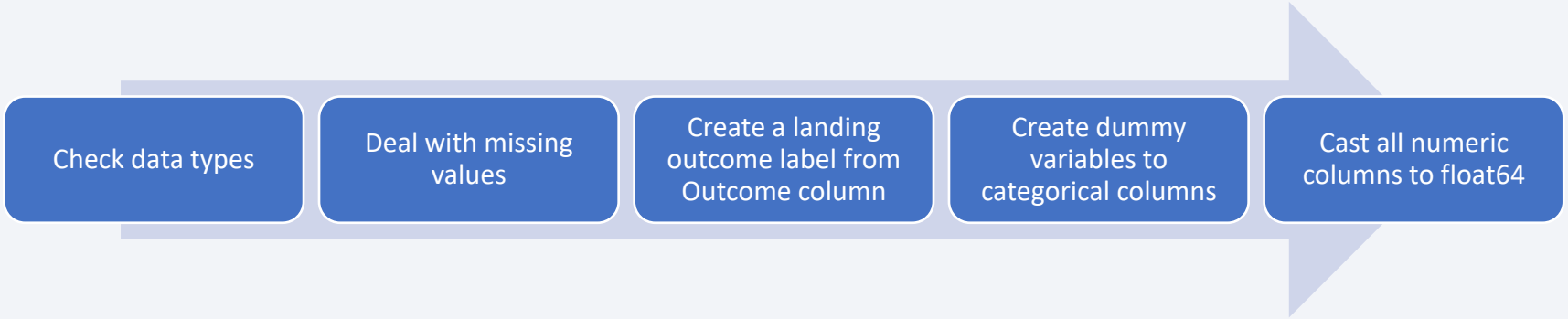


Data Collection - Scraping

- We used GET request to obtain additional data for Falcon 9 launches from Wikipedia website (for purpose of further analysis we used snapshot of this data provided by IBM)
- We used BeautifulSoup to parse obtained HTML response object
- We created pandas dataframe from parsed HTML tables
- We exported results to csv file
- https://github.com/LNakonieczny/IBM_Data_Science_Capstone_Project/blob/main/jupyter-labs-webscraping.ipynb



Data Wrangling



- We checked if all data are of correct type
- We replaced missing values in the payload mass column by the mean of the payload mass
- We created binary variable 'Class' that take value 0 when landing of first stage was failure and 1 when landing was successful
- Using Pandas get_dummies method we transformed categorical variables to the numerical ones
- We casted all numeric columns to float64

- https://github.com/LNakonieczny/IBM_Data_Science_Capstone_Project/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- We used catplot and scatterplot for investigating relations among payload mass , flight number, launch site, and orbit type and they connection to the outcome of the landing of the first stage
- We used barplot to visualize success rate for various orbits types
- We used lineplot to represent change of success rate for all launches over given time period

- https://github.com/LNakonieczny/IBM_Data_Science_Capstone_Project/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

We used SQL queries to obtain information about:

- launch sites,
- payload mass,
- successful and failure landing dates and booster versions.

- https://github.com/LNakonieczny/IBM_Data_Science_Capstone_Project/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- We created and added to folium map following objects:
  - circles to indicate launch sites position on global map,
  - markers to indicate if landing of first stage was a success or failure,
  - lines for measuring distance between launch site and nearest feature like coastline, railroad, highway or city.

- https://github.com/LNakonieczny/IBM_Data_Science_Capstone_Project/blob/main/lab_jupyter_launch_site_location.ipynb

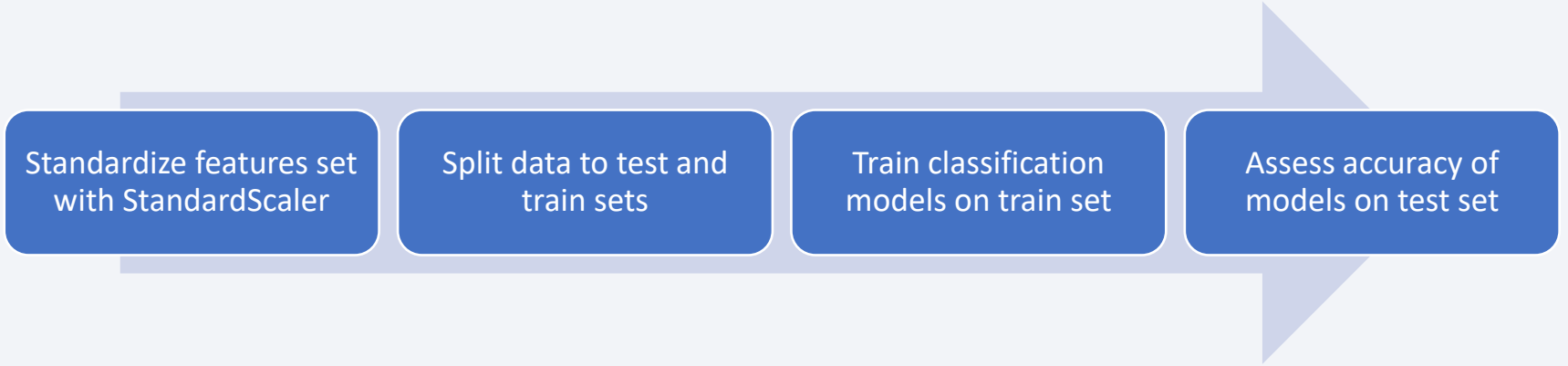
Build a Dashboard with Plotly Dash

We used the following interactions and visualization in dashboard:

- piechart to represent success rate for given launch site,
- scaterplot for visualization of relations between payload mass and success of retrieving first stage,
- dropdown list for drill down on success rate for specific launch site,
- slider to Focus on specific payload mass range.

- https://github.com/LNakonieczny/IBM_Data_Science_Capstone_Project/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

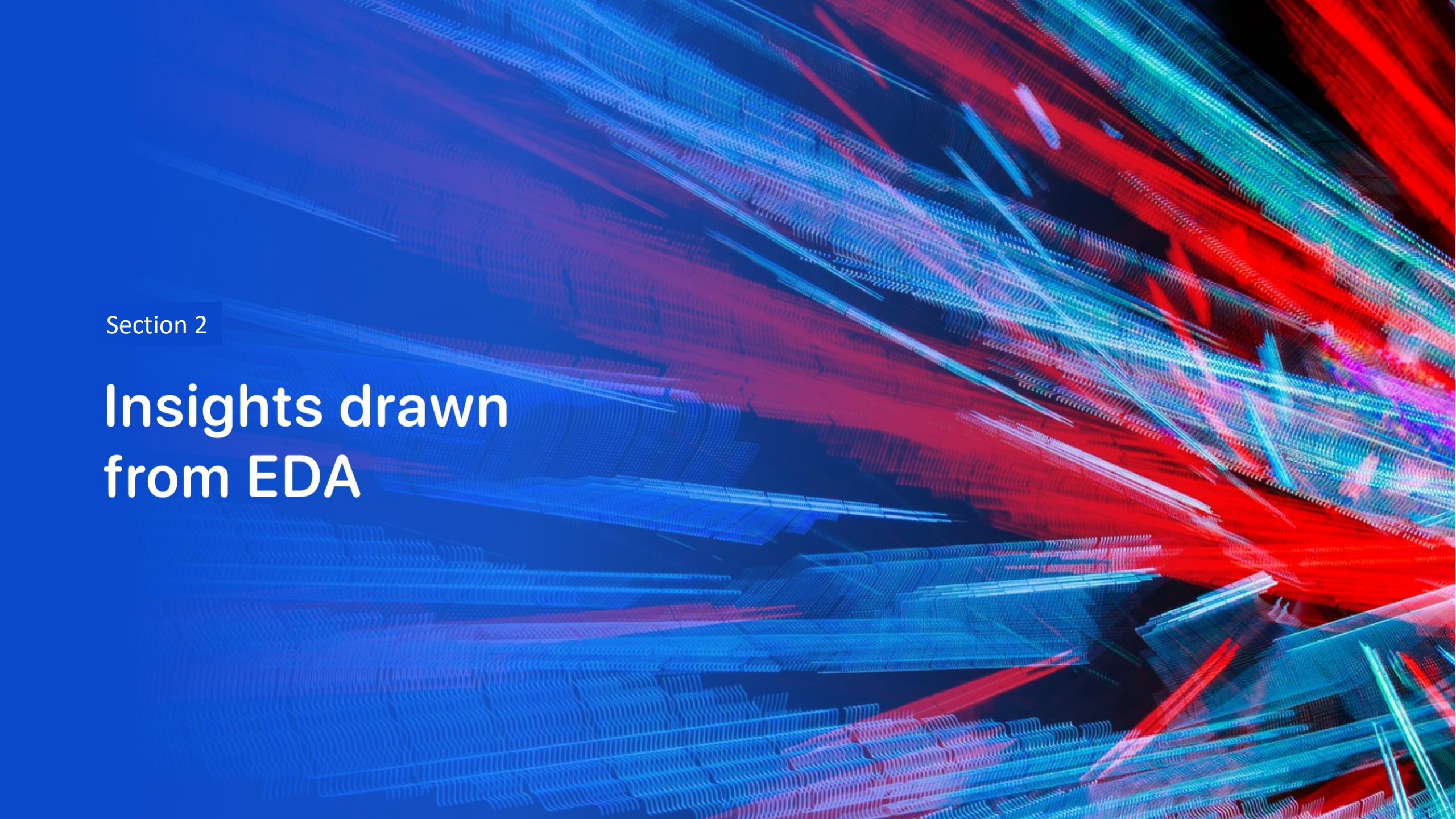


- As a preparatory steps before training of chosen classification models we used sklearn StandardSacer (on features) and subsequently train_test_split on whole dataset
- We trained selected models on train set
- To assess accuracy of the models we used test dataset and score method of sklearn
- We visualized accuracy of models with barchart and plot of confusion matrix for best performing model

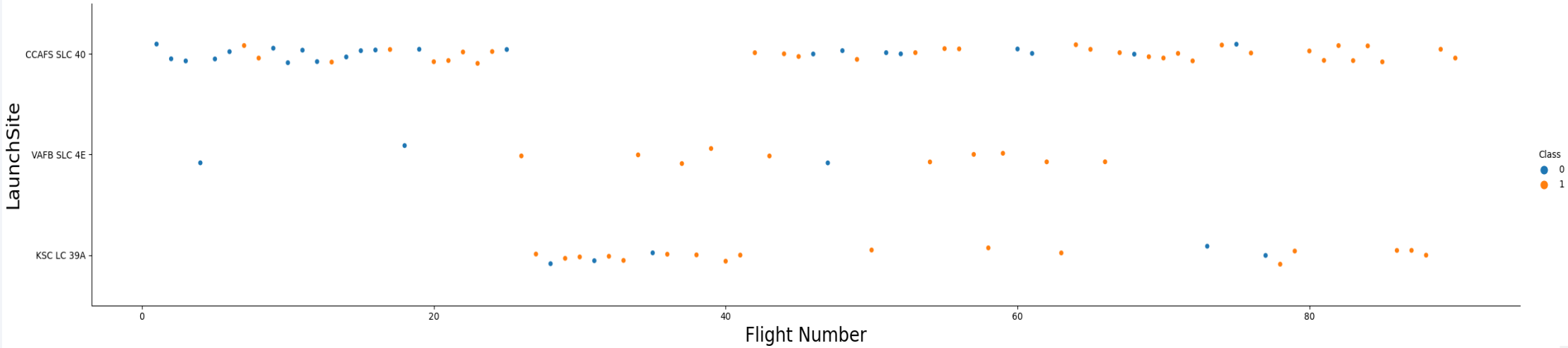
- https://github.com/LNakonieczny/IBM_Data_Science_Capstone_Project/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

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


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

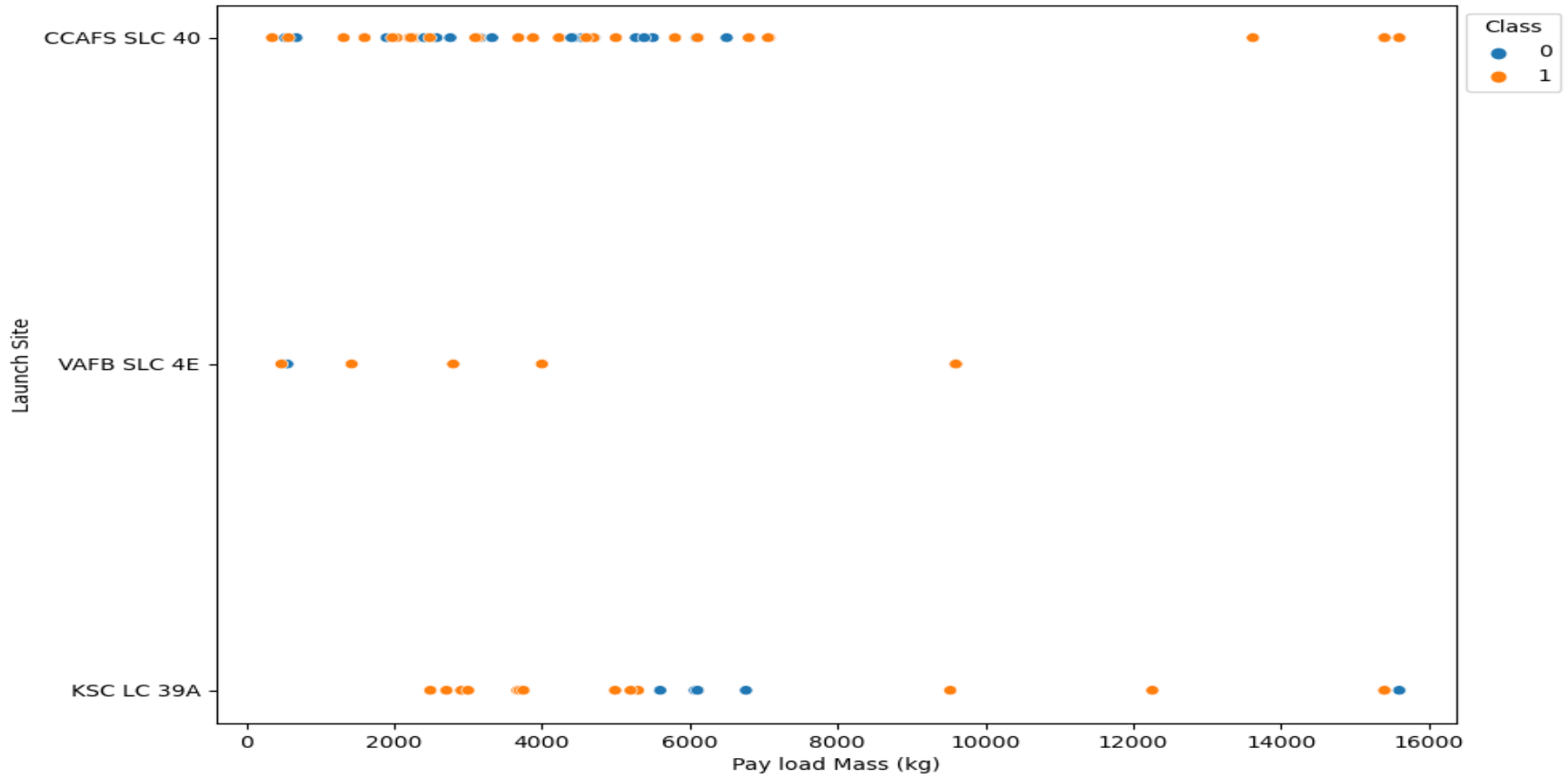


[11]:

	CCAFS SLC 40	VAFB SLC 4E	KSC LC 39A
Date of first launch	2010-06-04	2013-09-29	2017-02-19
Booster Serial	B0003	B1003	B1021

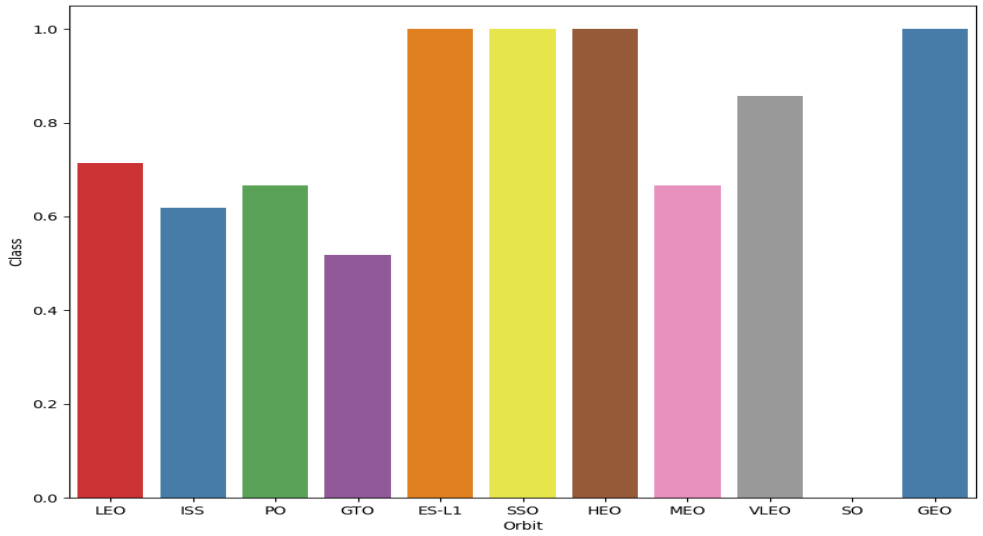
- We see that the CCAFS SLC 40 launch site has the largest number of launches that ended in failure especially in low fligh number sector.
- There are also some launches of this type in VAFB SLC 4E site while they are missing from KSC LC 39A.
- Investigation of collected dater revealed that this is due to the fact that CCAFS SLC 40 was the first launch site used by Falcon 9 rockets. Launches started at 2010 with the booster version B0003. Starting from 2013 VAFB SLC 4E was also used and in 2017 launches from KSC LC 39A started.
- The booster version used in the first launch from KSC LC 39A was B1021. We may connect high success rate for launches from this site to the fact that rocked launched there were operating on advanced version of boosters.

Payload vs. Launch Site



- Now if you observe Payload vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass (greater than 10000).
- We may also observe that for KSC LC 39A there is the cluster of failed launches for payload mass around 6000 kg.

Success Rate vs. Orbit Type

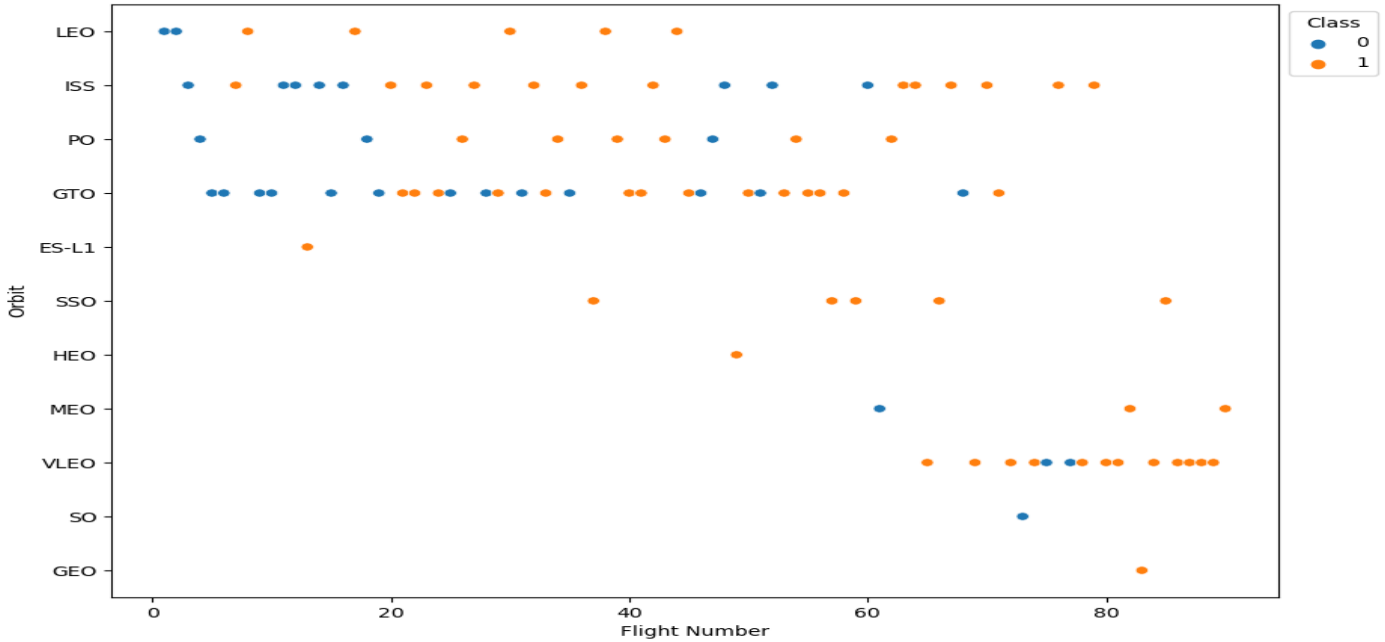


Analyze the ploted bar chart try to find which orbits have high sucess rate.

From the barplot above we see that ES-L1, SSO, HEO and GEO orbits have success rate of 100%. On the other hand, SO orbit has 0% success rate.
Yet if we take a look at the flight number that were destined at these orbits, we see that for all cases there was only single flight. Therefore, any conclusions about success rate for these orbits are unreliable.
Among the rest most successful were:

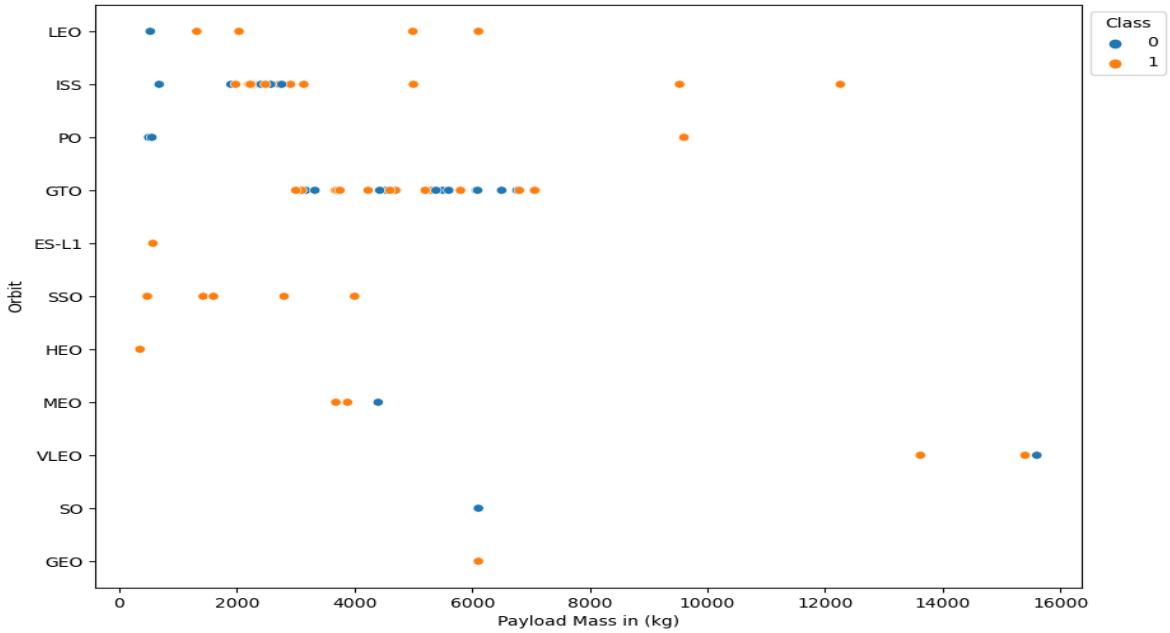
- VLEO (14 attempted flights) with success rate around 85%
- LEO (7 attempted flights) with success rate around 70%.

Flight Number vs. Orbit Type



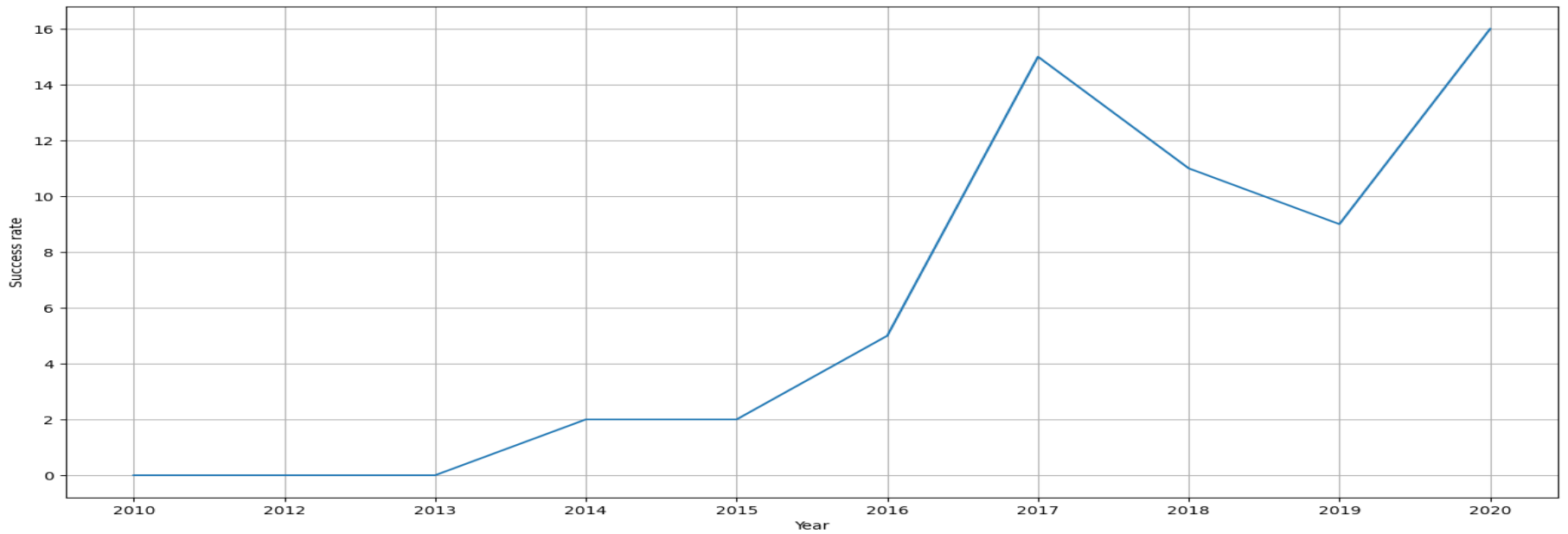
- You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.
- For all orbits as the flight number increase so does the success rate.

Payload vs. Orbit Type



- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing (unsuccessful mission) are both there here.
- We may observe that for two most distant orbits (GEO and MEO) there was no payload heavier than 6000 kg.

Launch Success Yearly Trend



- We can observe that the sucess rate since 2013 kept increasing till 2020
- We also see that there is dip in success rate between 2017 and 2018. Further investigation revealed that in 2017 and 2018 there were significant number of new boosters introduced. This may indicate that SpaceX team try to solve some persistent construction problems, or they were testing new approach in they booster construction.

All Launch Site Names

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

[10]:
```
%%sql
SELECT DISTINCT("Launch_Site") FROM SPACEXTBL;
```

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

[10]:

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

We queried provided sqlite database for unique launch site names.
For this purpose, we used sql magic extension of JupyterLab.

Launch Site Names Begin with 'CCA'

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

[11]:
```
%%sql
SELECT * FROM SPACEXTBL
WHERE "Launch_Site" LIKE 'CCA%'
LIMIT 5;
```

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

[11]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

We used WHERE clause with LIKE condition for filter our query.

Total Payload Mass

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

[12]:
```
%%sql
SELECT SUM("PAYLOAD_MASS__KG_") AS payload_by_NASA FROM SPACEXTBL
WHERE "Customer" LIKE '%NASA (CRS)%';
```

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

[12]:

payload_by_NASA
48213

According to our data NASA expedited 48213 kg of payload to various orbits.

Average Payload Mass by F9 v1.1

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

[9]:
```
%%sql
SELECT "Booster_Version", AVG("PAYLOAD_MASS__KG_") AS AVG_payload FROM SPACEXTBL
GROUP BY "Booster_Version"
HAVING "Booster_Version" LIKE '%F9 V1.1%';
```

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

[9]:

Booster_Version	AVG_payload
F9 v1.1	2928.4
F9 v1.1 B1003	500.0
F9 v1.1 B1010	2216.0
F9 v1.1 B1011	4428.0
F9 v1.1 B1012	2395.0
F9 v1.1 B1013	570.0
F9 v1.1 B1014	4159.0
F9 v1.1 B1015	1898.0
F9 v1.1 B1016	4707.0
F9 v1.1 B1017	553.0
F9 v1.1 B1018	1952.0

[10]:
```
%%sql
SELECT AVG("PAYLOAD_MASS__KG_") AS AVG_payload FROM SPACEXTBL
WHERE "Booster_Version" LIKE '%F9 V1.1%';
```

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

[10]:

AVG_payload
2534.6666666666665

- Average payload transported by F9 v1.1 is 2534.7 kg.
- Using GROUP BY statement with HAVIG clause to calculate average payload carried by various v1.1 booster subversions.

First Successful Ground Landing Date

[16]:
```
%%sql
SELECT Date FROM SPACEXTBL
WHERE "Landing _Outcome" LIKE "Success (ground pad)" AND
Date = (SELECT MIN(Date) FROM SPACEXTBL
        WHERE "Landing _Outcome" LIKE "Success (ground pad)");
```

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

[16]:

Date
01-05-2017

We used subquery to find that first successful ground pad landing date is 01.05.2017

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

[17]:
```
%%sql
SELECT "Booster_Version" FROM SPACEXTBL
WHERE "Landing _Outcome" LIKE "Success (drone ship)" AND
"PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000;
```

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

[17]:

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

We used WHERE clause with BETWEEN condition to find out that there were four booster versions that carried between 4000 and 6000 kg of payload and successfully landed on drone ship.

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

[18]:
```
%%sql
SELECT "Mission_Outcome", COUNT("Mission_Outcome") FROM SPACEXTBL
GROUP BY "Mission_Outcome";
```

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

[18]:

Mission_Outcome	COUNT("Mission_Outcome")
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Provided database contains entries for 101 mission outcomes for which 100 ended in success and 1 in failure. To find this we used GROUP BY statement and COUNT aggregate function.

Boosters Carried Maximum Payload

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

[19]:
```
%%sql
SELECT DISTINCT("Booster_Version") FROM SPACEXTBL
WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL);
```

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

[19]:

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

Using subquery, we found out that boosters that carried maximum payload belonged to B5 family.

2015 Launch Records

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.

[20]:
```
%%sql
SELECT
CASE SUBSTR(Date, 4, 2)
    WHEN '01' THEN 'January'
    WHEN '02' THEN 'February'
    WHEN '03' THEN 'March'
    WHEN '04' THEN 'April'
    WHEN '05' THEN 'May'
    WHEN '06' THEN 'June'
    WHEN '07' THEN 'July'
    WHEN '08' THEN 'August'
    WHEN '09' THEN 'September'
    WHEN '10' THEN 'October'
    WHEN '11' THEN 'November'
    WHEN '12' THEN 'December'
END AS month,
"Landing _Outcome", "Booster_Version", "Launch_Site"
FROM SPACEXTBL
WHERE SUBSTR(Date, 7, 4) = '2015' AND "Landing _Outcome" LIKE 'Failure (drone ship)';
```

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

[20]:

month	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

We found out two entries for failed drone ship landings. On technical side we used CASE statement to translate month number to name.

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

[26]:
```
%%sql
SELECT "Landing _Outcome", COUNT(*) AS COUNT
FROM SPACEXTBL
GROUP BY "Landing _Outcome"
HAVING (SUBSTR(Date, 7, 4) || '-' || SUBSTR(Date, 4, 2) || '-' || SUBSTR(Date, 1, 2)) BETWEEN "2010-06-04" AND "2017-03-20"
ORDER BY COUNT DESC;
```

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

[26]:

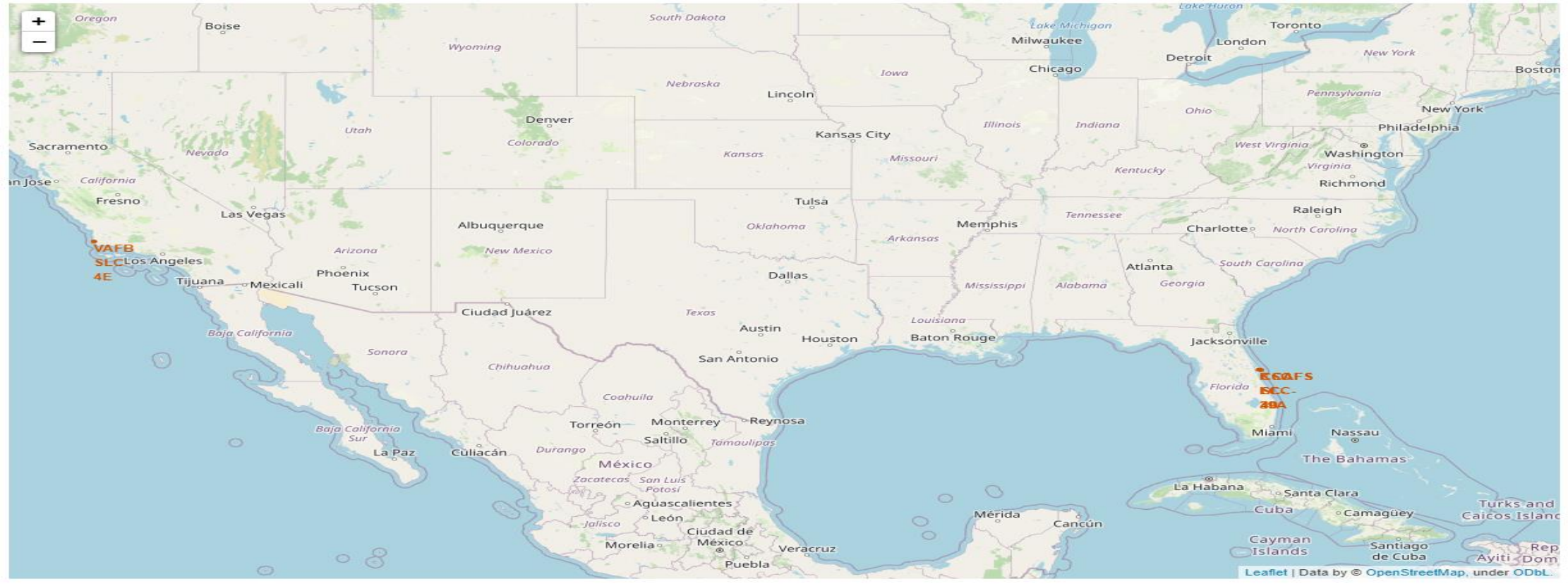
Landing _Outcome	COUNT
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

We done this the hard way in sqlite where dates are given in string format and year is last in sequence.

Section 3

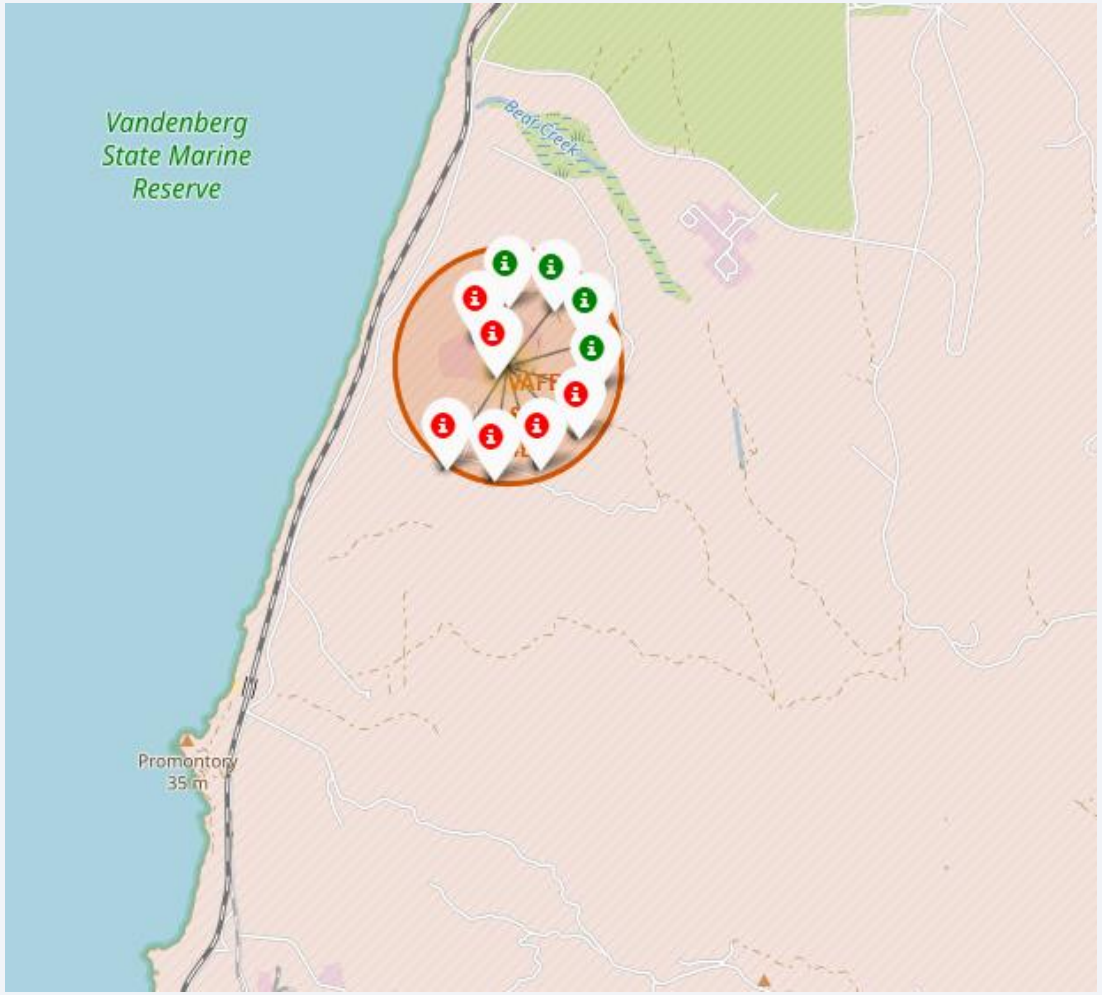
Launch Sites Proximities Analysis

Location of the SpaceX launch sites



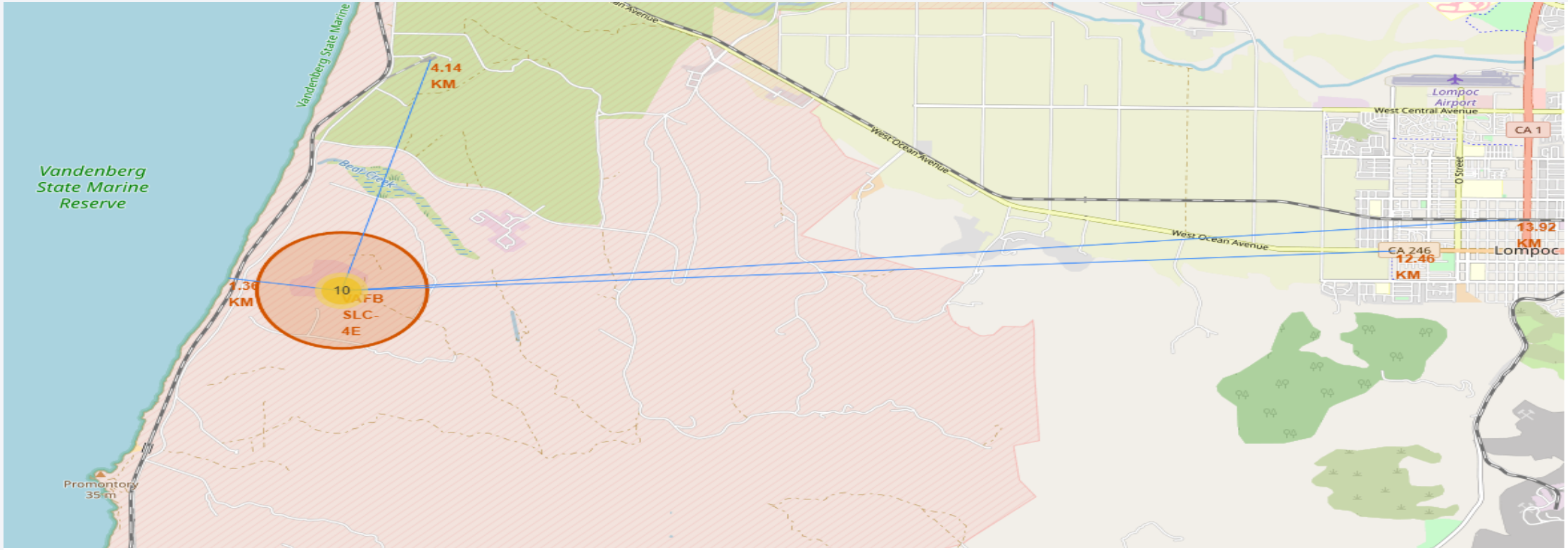
We observe that all launch sites are in vicinity of coastline and in relative distance from large urban areas.

Launch outcome for VAFB SLC 4E base



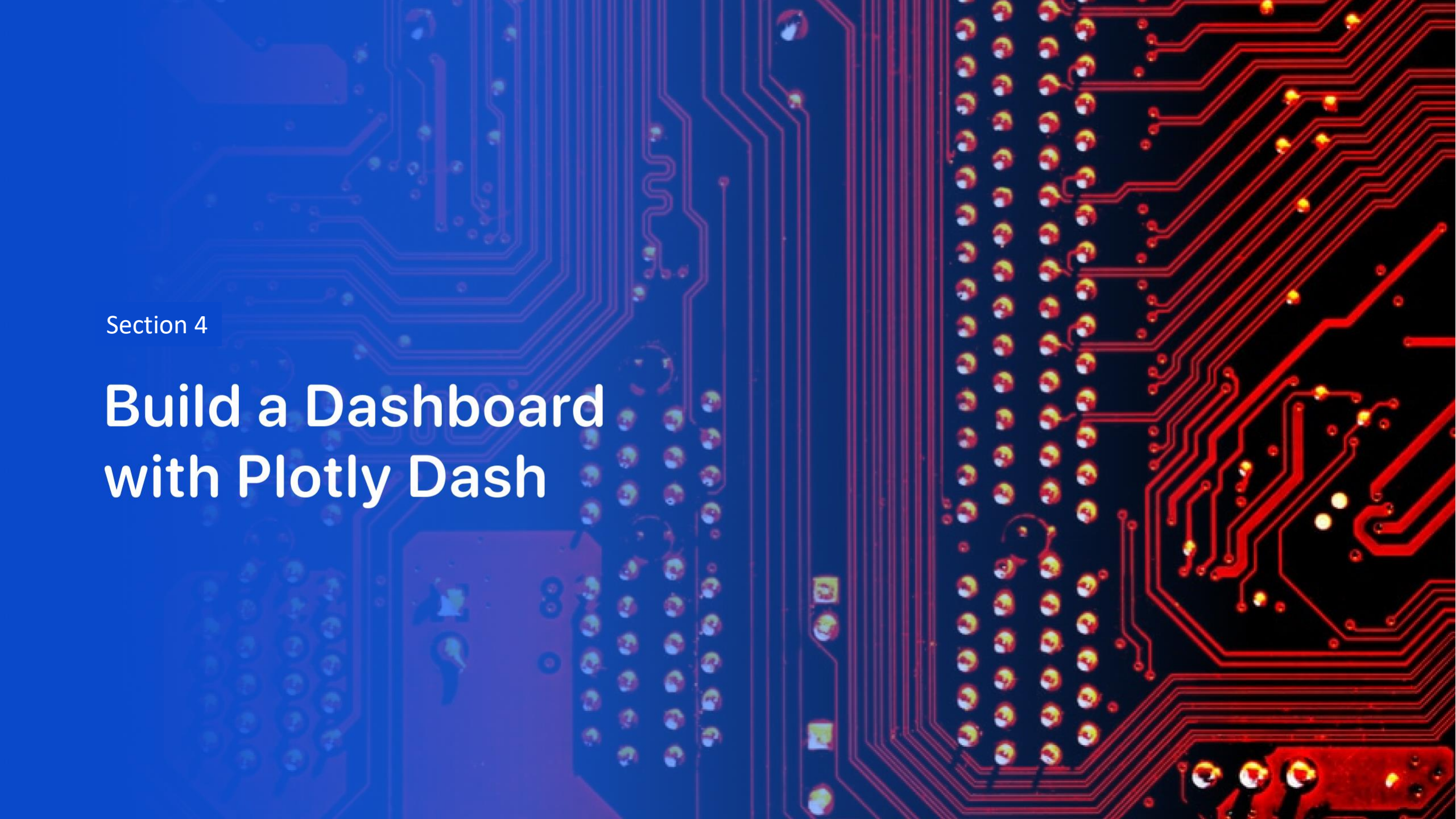
We used markers to indicate launch outcome.
Green ones represent successful landing of the first stage and the red failed attempt.

Launch site surrounding landscape and infrastructure

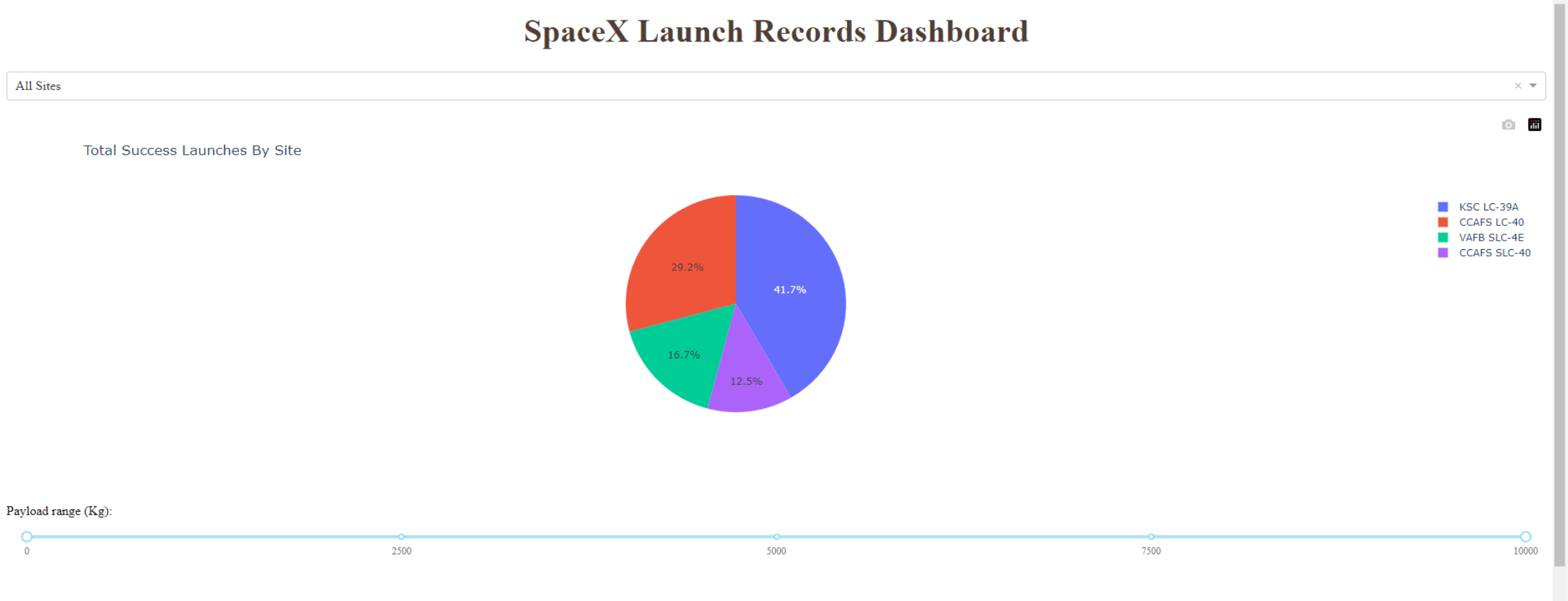


Characteristics of launch site locations:

- proximity to rail lines or highways which is important for transporting spaceships, rockets, fuel, and payload
- proximity of coastline and/or dessert region and away from cities to minimize damage to the infrastructure and causalities in case of rocket explosion

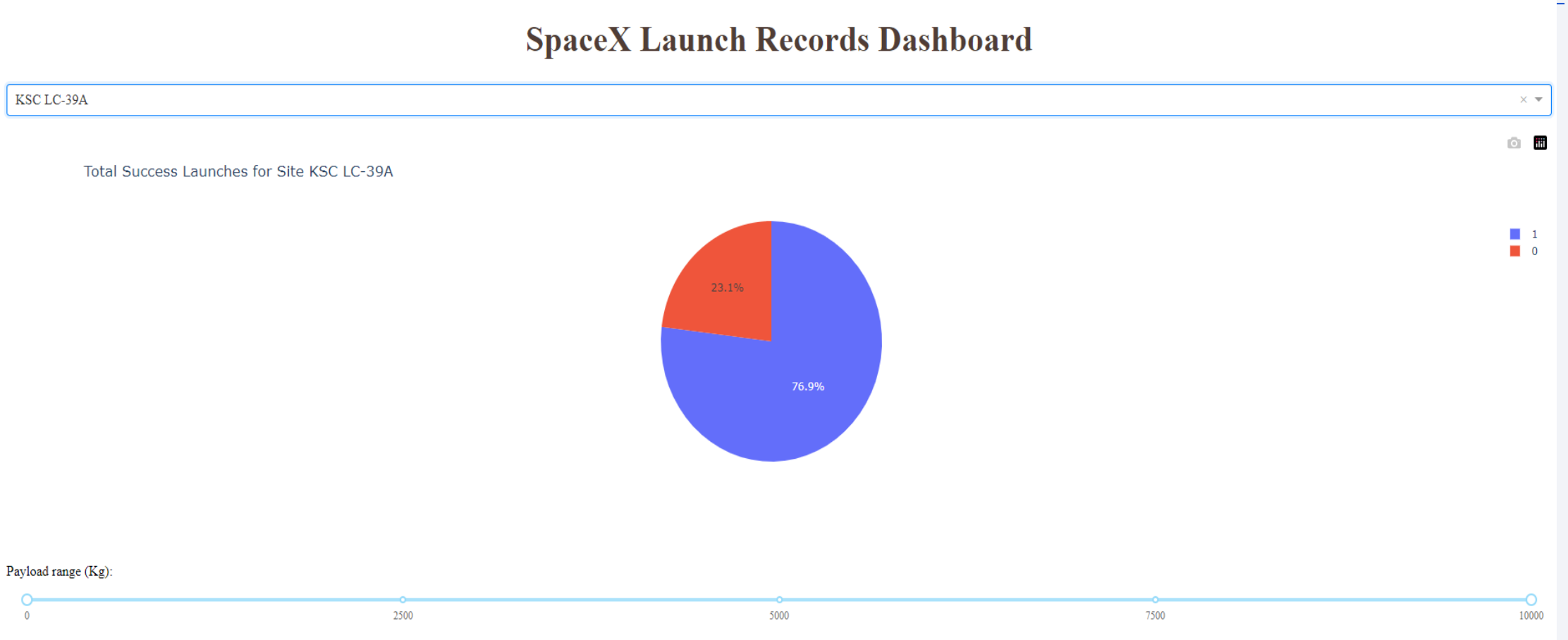

Section 4
Build a Dashboard with Plotly Dash

Percentage of successful launches for all sites



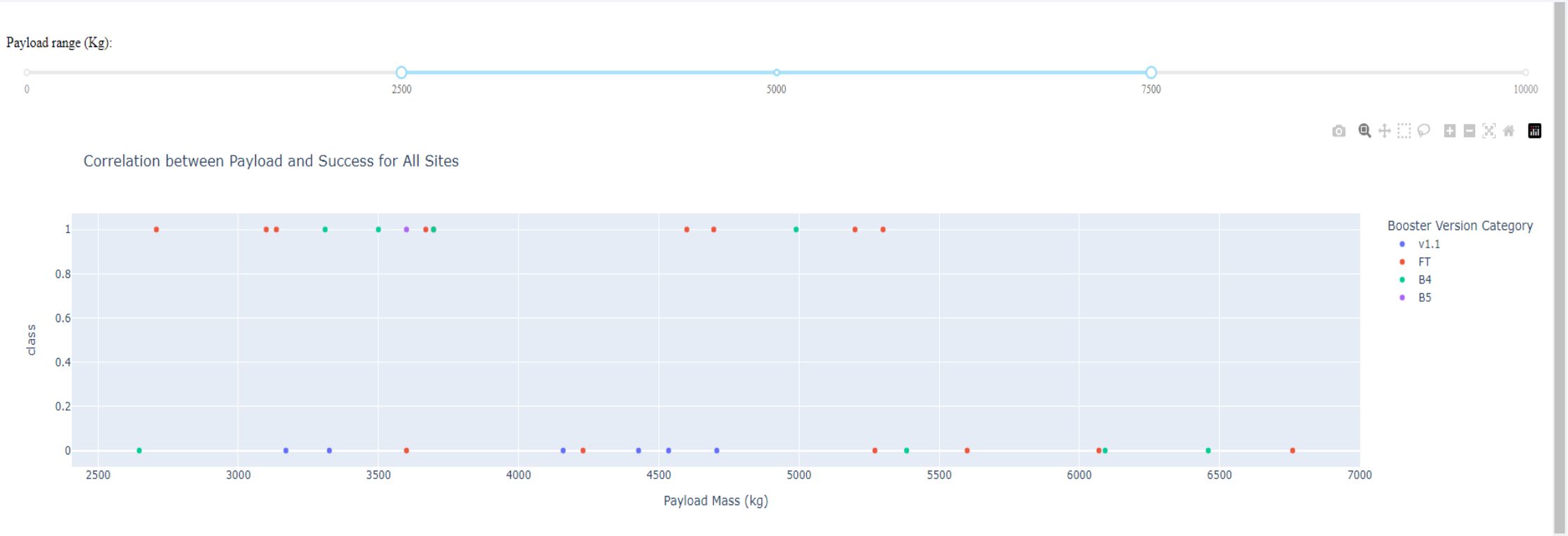
- Most successful site is KSC LC-39A followed by CCAFS LC-40.
- Remaining site have success rate below 20%.

First stage landing statistic for KSC LC-39A site



For this site first stage was successfully landed for almost 77% of cases.

Payload vs Launch Outcome for All Sites

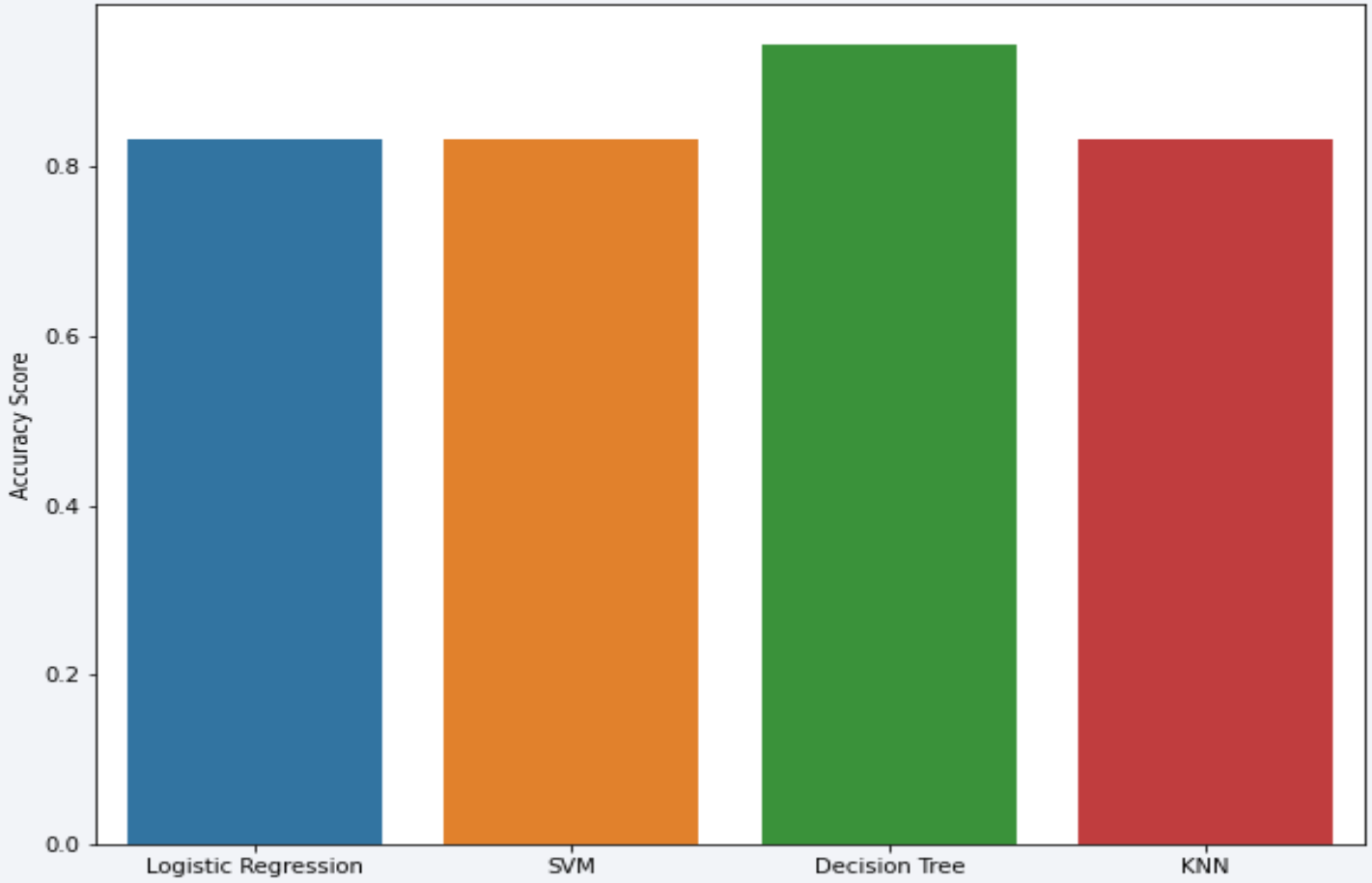


- We see that for payload mass range between 2500 and 7500 kg most successful was FT booster.
- Worst preforming booster was model v1.1 which was earliest booster introduced.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

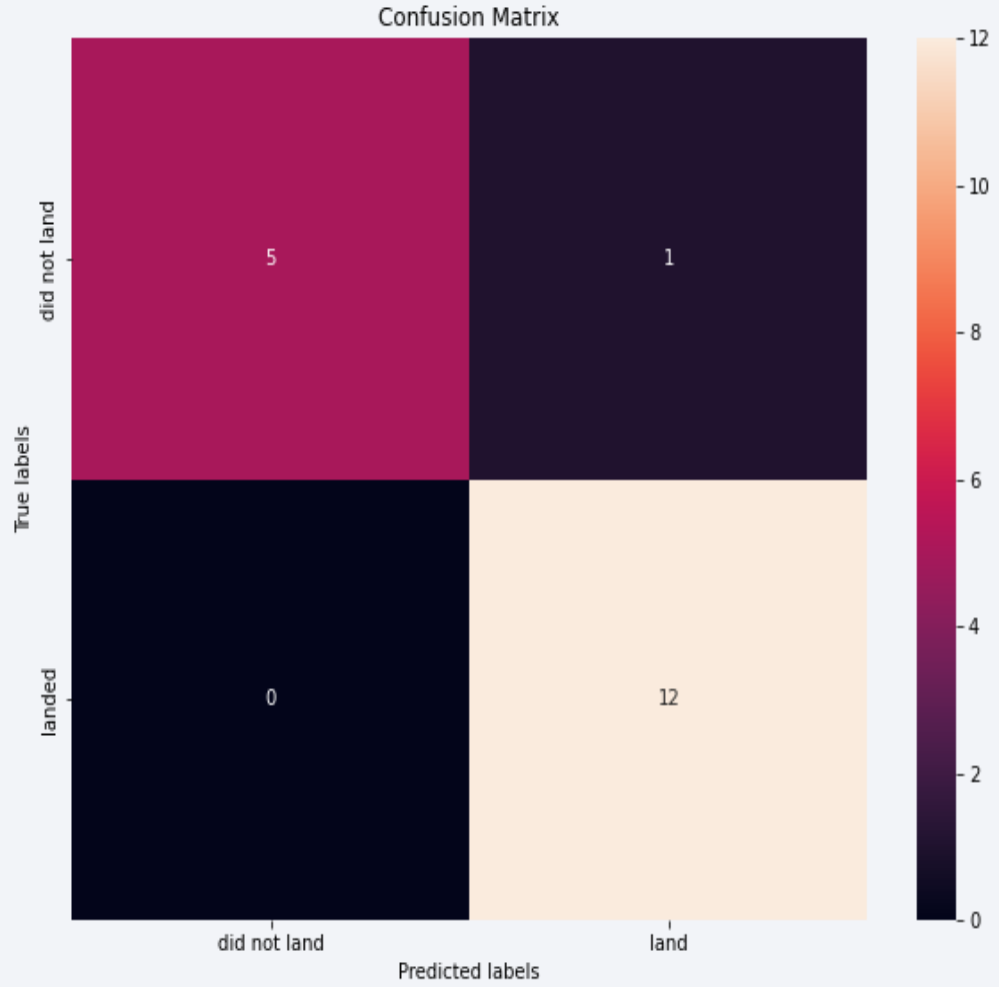


- Best classification model was Decision Tree
- Best hyperparameters were:

```
[51]: criterion             entropy
      max_depth                  18
      max_features             sqrt
      min_samples_leaf            2
      min_samples_split          10
      splitter                 best
      dtype: object
```

- Accuracy calculated by scikit-learn score method was 94%

Confusion Matrix



We present confusion matrix for Decision Tree classifier for test dataset:

- no. of false positives is 1
- no. of false negatives is 0

Classification report containing information in precision and recall:

	precision	recall	f1-score	support
0	1.00	0.83	0.91	6
1	0.92	1.00	0.96	12
accuracy			0.94	18
macro avg	0.96	0.92	0.93	18
weighted avg	0.95	0.94	0.94	18

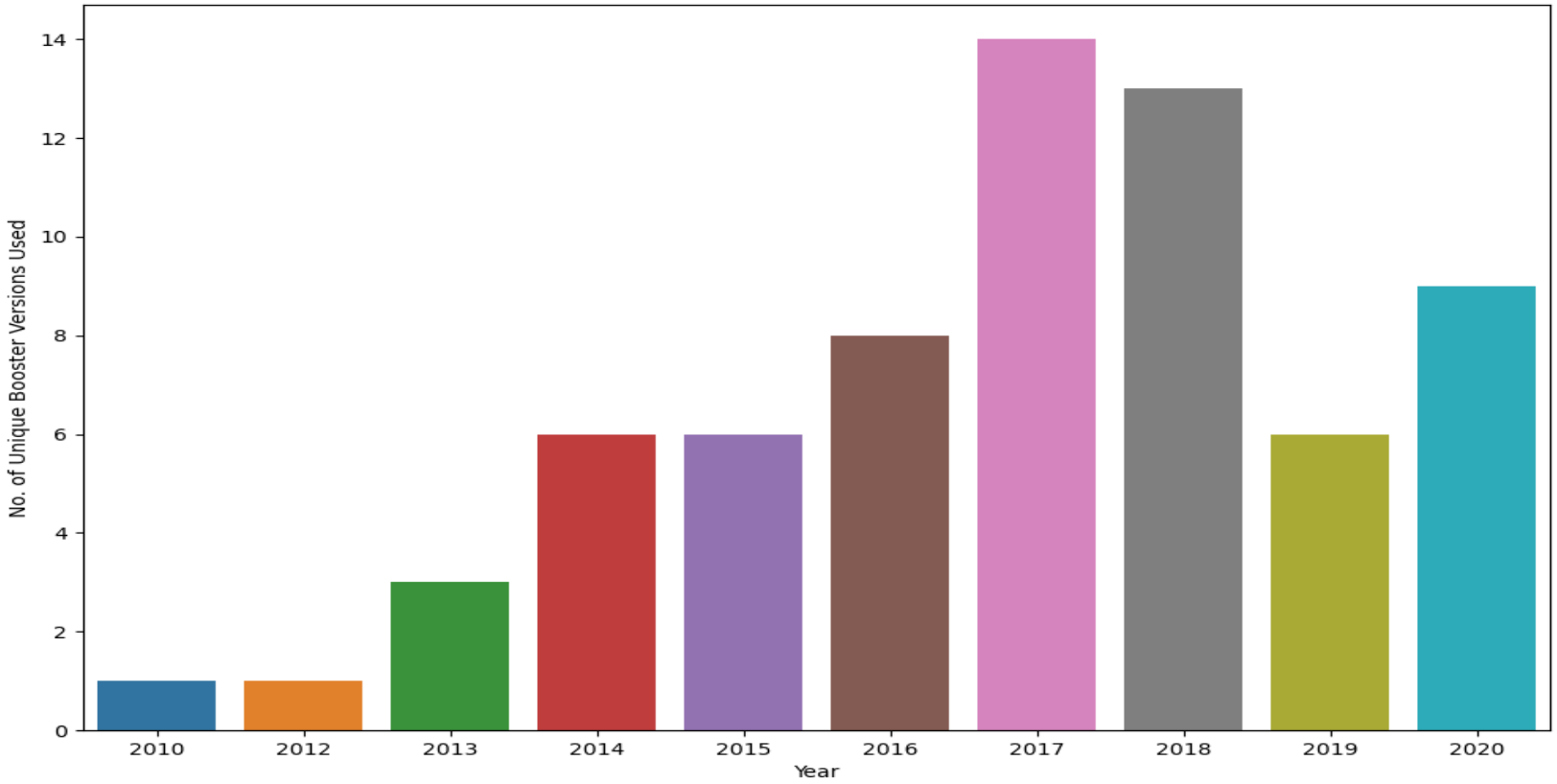
Conclusions

- Collected data indicates that overall, most successful site is KSC LC 39A.
- If using it is not possible or cost effective, we may pick alternative based on the payload mass:
  - for mass up to around 8000 kg any of remaining site will do,
  - for payload above 8000 kg second to best will be CCAFS SLC 40.
- Two most successful booster versions were B1049 and B1051 so it is worth to take as close look at they characteristics as it is possible.
- For judging on success on the first stage retrieval for given launch parameters we choose machine learning model based on Decision Tree classifier which achieved 94% accuracy rate on predictions.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Appendix: unique booster versions used per year

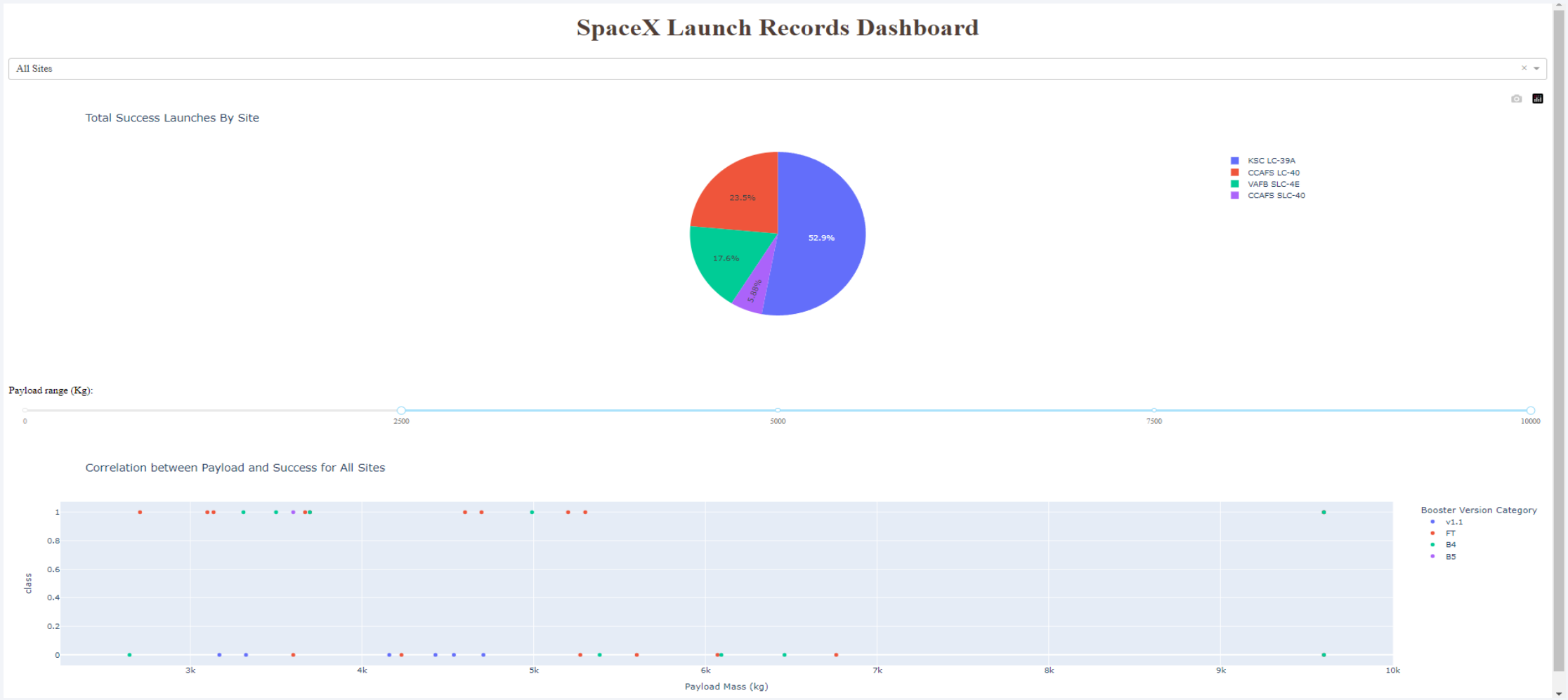


Appendix: most successful booster version

```
[47]: df[df['Class'] == 1].groupby('Serial').count().sort_values('Date', ascending=False)['Date'].head(2)

[47]: Serial
      B1049    6
      B1051    5
      Name: Date, dtype: int64
```

Appendix: SpaceX dashboard for specific payload range



Appendix: list of GitHub archive files

- Link for GitHub repo: https://github.com/LNakonieczny/IBM_Data_Science_Capstone_Project
- List of files used and created during project preparation:

```
SpaceX_Machine Learning Prediction_Part_5.ipynb
Spacex.csv
dash_downloads.ipynb
jupyter-labs-eda-dataviz.ipynb
jupyter-labs-eda-sql-coursera_sqllite.ipynb
jupyter-labs-spacex-data-collection-api.ipynb
jupyter-labs-webscraping.ipynb
lab_1.csv
lab_2.csv
lab_3.csv
lab_jupyter_launch_site_location.ipynb
labs-jupyter-spacex-Data wrangling.ipynb
my_data1.db
spacex_dash_app.py
spacex_launch_dash.csv
spacex_launch_geo.csv
spacex_web_scraped.csv
```

Thank you!