Winning Space Race with Data Science

Danielle de Pinho Mello
06 April 2022

Outline

Executive Summary

Space exploration can have elevated costs and it can prevent new companies to start in the aerospace manufacturer industry. It is important to observe how successful companies in this field operate to solve this challenge. SpaceX, for example, advertises that their launch cost was reduced by 100 million dollars by reusing the first stage of their rockets. Therefore, this project aims to predict the landing success of the first stage rocket.

In order to achieve this goal, the data about launches of the SpaceX's Falcon 9 were analyzed. The data were collected through an API and web scraping; then it was visualized and manipulated in the EDA stage; prepared for prediction through classification models; and finally, the predictions were made, and the models evaluated.

All models built had a good accuracy score and returned the expected outcome with good confidence.

Introduction

- Space exploration has been an important topic since the space race in the 50s. But launching rockets into space is a business that comes with great costs and risks.
- SpaceX, an aerospace manufacturer, was able to reduce their missions' cost by reusing the first stage of the rocket.
- SpaceX advertises that, for its Falcon 9 rocket, they can reduce the cost from 165 million to 62 million dollars if the first stage lands safely.
- Research question: How can we predict a successful landing of the rocket's first stage?
- Goal: to predict the success a rocket's first stage landing.
- To find the answer and best outcome for this project, data about SpaceX's Falcon 9 launches were analyzed.

Falcon 9

- Falcon 9 is a reusable, two-stage rocket that can transport people and payload into space. It is the first of its kind.
- It is known for the ability to ”refly” the most expensive parts of the rocket, lowering the cost of space access.
- Falcon 9 first launch was in June 2010, as part of the Commercial Orbital Transportation Services (COTS) program.
- In 2012, SpaceX’s began the reusability test program and achieved a successful landing and recovery of a first stage in December 2015 with Falcon 9 Flight 20.

(SpaceX, 2022; Wikipedia, 2022a)

FALCON 9

OVERVIEW

HEIGHT	70 m / 229.6 ft
DIAMETER	3.7 m / 12 ft
MASS	549,054 kg / 1,207,920 lb
PAYLOAD TO LEO	22,800 kg / 50,265 lb
PAYLOAD TO GTO	8,300 kg / 18,300 lb
PAYLOAD TO MARS	4,020 kg / 8,860 lb

Source: SpaceX, 2022

Section 1

Methodology

Methodology

Methodology stages:

- Data collection:
  - The data were collected through the SpaceX API and scraping the Wikipedia page “List of Falcon 9 and Falcon Heavy launches”.
- Data wrangling
  - The data was cleaned, transformed and observed to be able to create a new column that would serve as the target for the models.
- Exploratory data analysis (EDA) using visualization and SQL
- Interactive visual analytics using Folium and Plotly Dash
- Predictive analysis using classification models
  - After features engineering, the data was standardized and split. Classification models were built and evaluated.

Data Collection

- To help answering the problem of this project, it was necessary to collect data about SpaceX and Falcon 9 and observe which part of it could help in the prediction.
- The data was collected in two stages:
  - Through the SpaceX API
  - Web scraping the Wikipedia page “List of Falcon 9 and Falcon Heavy launches”

Objectives:

API	Web scraping
☐ Request to the SpaceX API	☐ Extract a Falcon 9 launch records HTML table from Wikipedia
☐ Clean the requested data	☐ Parse the table and convert it into a Pandas data frame

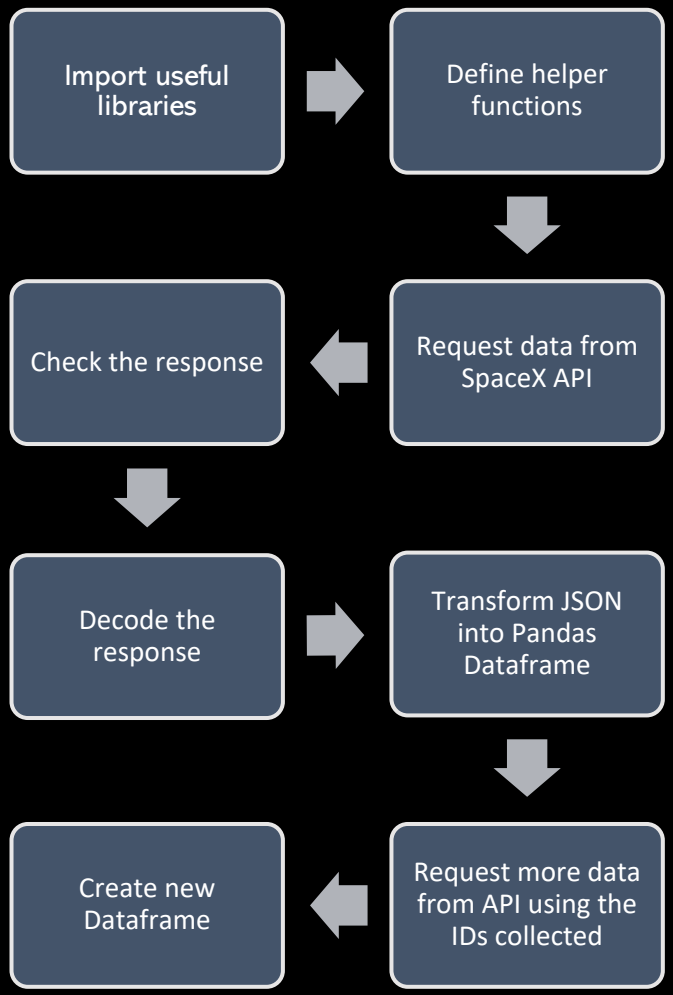
Data Collection – SpaceX API

- Using the requests library, the SpaceX API was accessed, and the rocket launch data was requested;
- The response was a JSON that was transformed to a Pandas Dataframe.

```
j_response = response.json()
data = pd.json_normalize(j_response)
```

- Much of the data present in the Dataframe were identification numbers and did not contain other information. A new access was needed and will be detailed in the next slide.

Full notebook is available on Github: Data Collection API



Data Collection – SpaceX API

- Using the IDs, relevant information was requested.
- The new data collected were, first, stored in lists and later into a dictionary in order to be transformed into a new Pandas Dataframe.

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

Used IDs	Information collected
Rocket	Booster version
Payload	Payload Mass, Orbit
Launchpad	Launch site's name, Latitude, Longitude
Cores	Outcome, landing type, number of flights, gridfins use, core reusability, legs use, landing pad, core block, number of time the core was used, serial of the core.

Data Collection – SpaceX API (Data Wrangling)

- After collecting the data from SpaceX API, the Dataframe was filtered to include only the Falcon 9 launches and checked for missing values.
- There were two columns with missing values: PayloadMass and LandingPad.
- The LandingPad null values represent when landing pads were not used, therefore no action was required.
- For the missing values in the PayloadMass column, the function replace was used to substitute the NAN value with the PayloadMass mean.

```
# Calculate the mean value of PayloadMass column
payloadmass_mean = data_falcon9["PayloadMass"].mean()
# Replace the np.nan values with its mean value
data_falcon9["PayloadMass"] = data_falcon9["PayloadMass"].replace(np.nan, payloadmass_mean)
with the mean
```

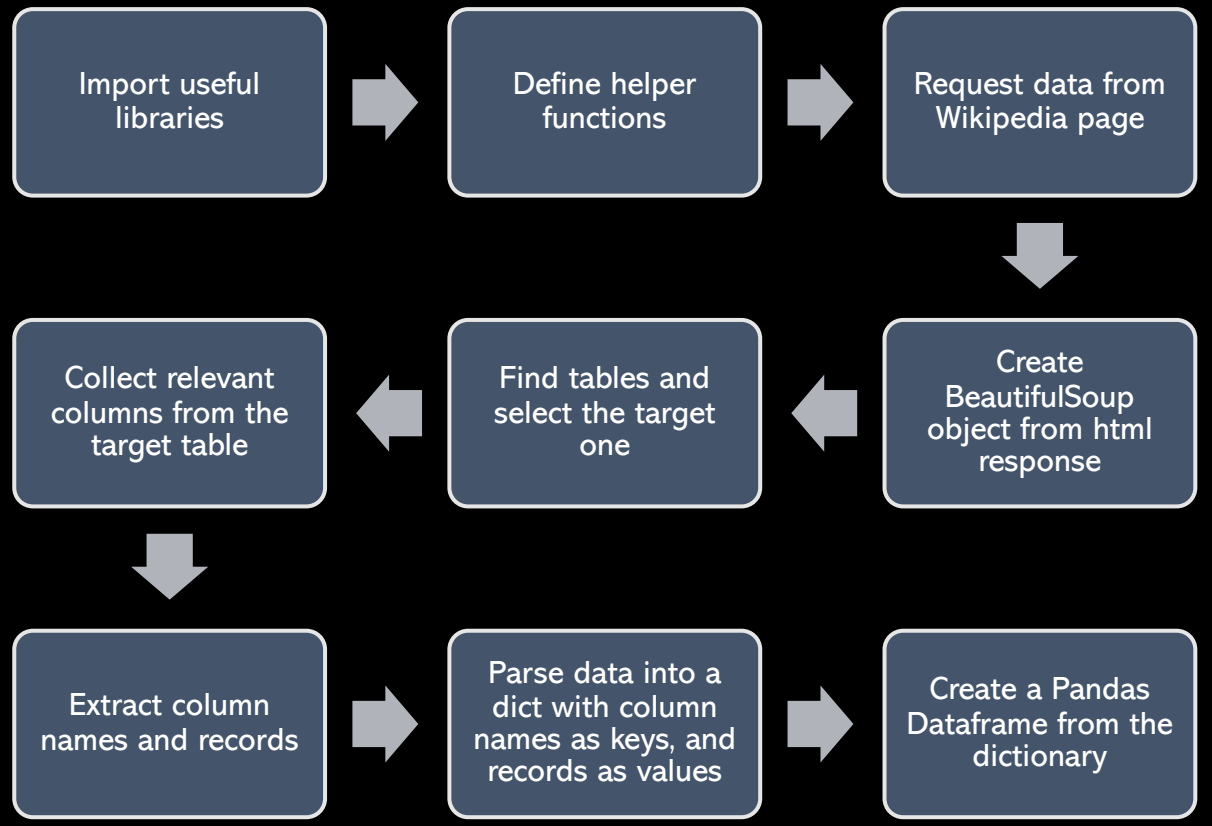
```
data_falcon9.isnull().sum()
```

FlightNumber	0
Date	0
BoosterVersion	0
PayloadMass	5
Orbit	0
LaunchSite	0
Outcome	0
Flights	0
GridFins	0
Reused	0
Legs	0
LandingPad	26
Block	0
ReusedCount	0
Serial	0
Longitude	0
Latitude	0

dtype: int64

Data Collection - Scraping

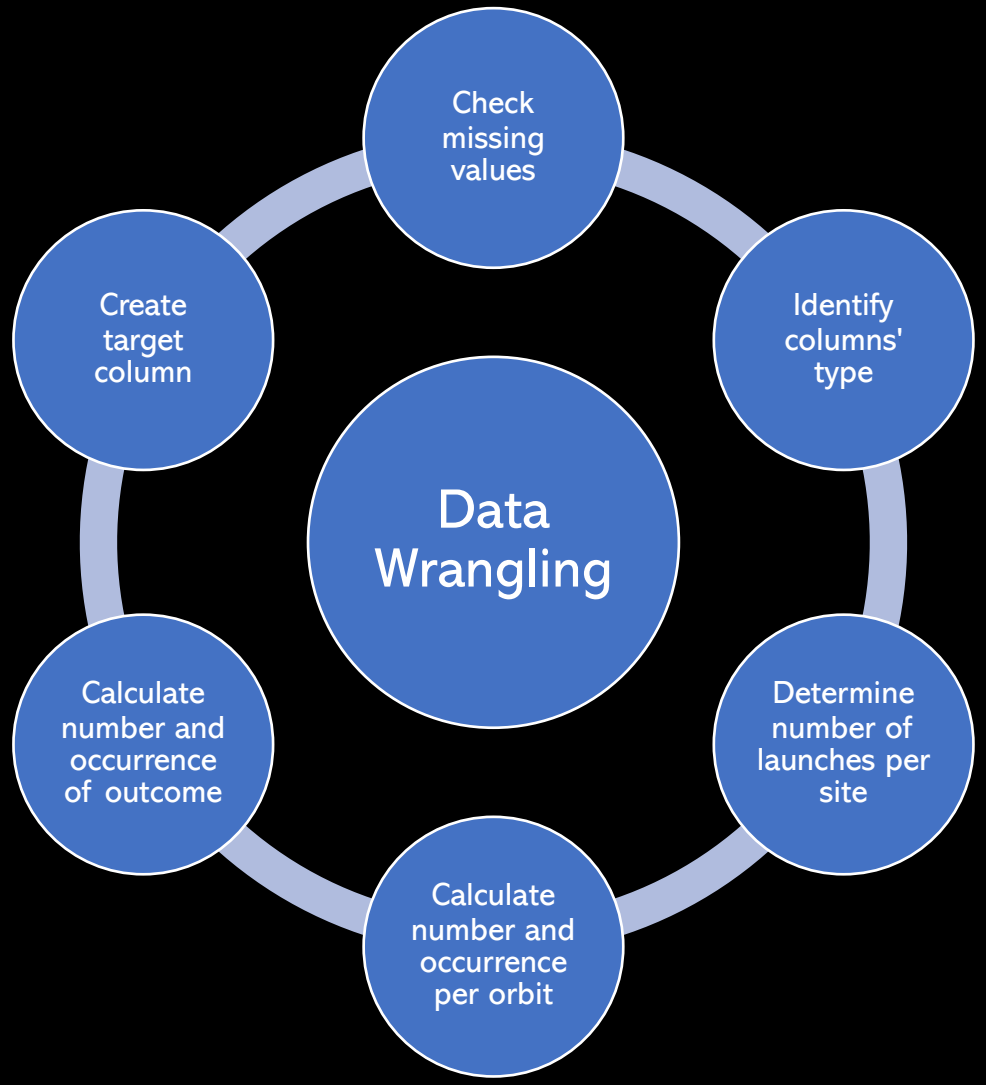
- First, useful libraries, such as Requests and BeautifulSoup were imported.
- The requested html response was used to create a BeautifulSoup object.
- The tables present in the webpage were located and the one containing the proper information was selected.
- The columns with the relevant data were selected and their names served as keys of an empty dictionary.
- The records values from the table rows were extracted and parsed into the dictionary.
- A Pandas Dataframe was created from this dictionary.



Full notebook is available on Github: Data Collection - Web Scraping

Data Wrangling

- Data wrangling consists in the process of cleaning, transforming and mapping data.
- An exploratory data analysis (EDA) was made to find data patterns and to define which label would be used for training supervised models.
- The data wrangling process was divided in:
  - Checking missing values.
  - Identifying the type of columns, if numerical or categorical.
  - Analyzing launches data.
  - Observing landing outcome.
  - Defining a new column with the outcome result that will serve as target, consisting in 1 for landing successfully and 0 for landing failure.



Full notebook is available on Github: Data Wrangling - EDA

Data Wrangling – Defining Target

- To define the target, the data about each launch were analyzed.
- As part of the analysis, it was calculated:
  - The number of launches for each site;
  - The number and occurrences for each orbit;
  - The number and occurrences of mission outcomes.
- It was observed:
  - The landing site with more launches was CCAFS SLC 40 with 55 launches;
  - More launches were made towards the orbit GTO, a geosynchronous orbit that is a high Earth orbit that allows satellites to match Earth's rotation;
  - More than 66% of the launches were able to successfully land the first stage.
- A column named "Class" was created with values that show if a launch had their first stage landed successfully or not:

  1: First stage landed successfully

  0: First stage did not land successfully

Outcome	Frequency	Meaning	Class
True ASDS	41	successfully landed to a drone ship	1
None None	19	failure to land	0
True RTLS	14	successfully landed to a ground pad	1
False ASDS	6	unsuccessfully landed to a drone ship	0
True Ocean	5	successfully landed to a specific region of the ocean	1
False Ocean	2	unsuccessfully landed to a specific region of the ocean	0
None ASDS	2	failure to land	0
False RTLS	1	unsuccessfully landed to a ground pad	0

EDA with Data Visualization

- To analyze possible relations between several variables in the data, a myriad of charts were plotted.
- Summary of charts:
  - Scatter plot between Flight Number and Launch Site
  - Scatter plot between Payload Mass and Launch Site
  - Bar chart observing the success rate for Orbit type
  - Scatter plot between Flight Number and Orbit type
  - Scatter plot between Payload Mass and Orbit type
  - Line chart showing the Launch success yearly trend

Full notebook is available on Github: Data Visualization - EDA

EDA with SQL

- SQL queries performed:
  - Unique launch sites in the space mission
  - First 5 records where launch sites begin with the string 'CCA'
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The date when the first successful landing outcome in ground pad was achieved
  - The names of the boosters which have success landing in drone ship and have payload mass greater than 4000 but less than 6000
  - The total number of successful and failure mission outcomes
  - The names of the booster versions which have carried the maximum payload mass
  - The failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - The ranking of the count of landing outcomes between 2010-06-04 and 2017-03-20, in descending order

Full notebook is available on Github: Data Analysis - EDA with SQL

Build an Interactive Map with Folium

Maps created with Folium:

- All Launch Sites map: it contains circles and markers to highlight the location of each launch site area.
- Success and failed outcomes per site: colored-marker were inserted for each launch to show which one had a successful or unsuccessful outcome.
- Proximities of launch site: With markers, a few sites like railways, highways, cities and coastline, that are in the proximities of a launch site, were highlighted in the map. The distance between them was calculated and displayed on the map. Finally, a line was plot to better show that distance.

Full notebook is available on Github: Data Visualization - Interactive Visualization with Folium

Build a Dashboard with Plotly Dash

A Dashboard was made with Plotly Dash, containing:

- Pie charts with information about the outcome by launch sites.
- A dropdown input was created for the dashboard to allow the selection of a specific launch site or all sites to analyze the data.
- Scatter plots with the relation between payload mass and outcome per booster version, for a selected launch site or for all sites.
- It also contains a slider that allows filtering the scatter plots between a range of payload mass.

Full notebook is available on Github: Data Visualization - Dashboard with Plotly Dash

Predictive Analysis (Classification)

- To be able to predict if a first stage rocket would successfully land, a machine learning pipeline was created, considering the data collected previously.
- Before building the classification models, a features engineering stage was necessary to select and prepare which data would be used to help the prediction.
- Classification models algorithms imported for this project:
  - Logistic Regression
  - Support Vector Machine
  - Decision Tree
  - K Nearest Neighbors

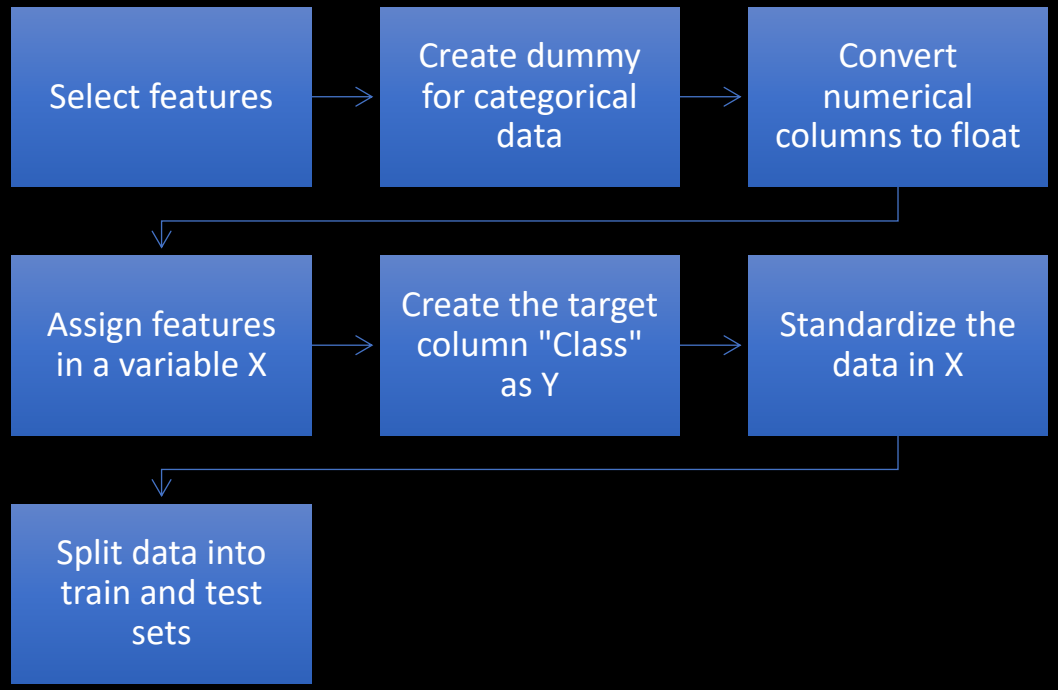
Full notebook is available on Github: Machine Learning Prediction

Objectives

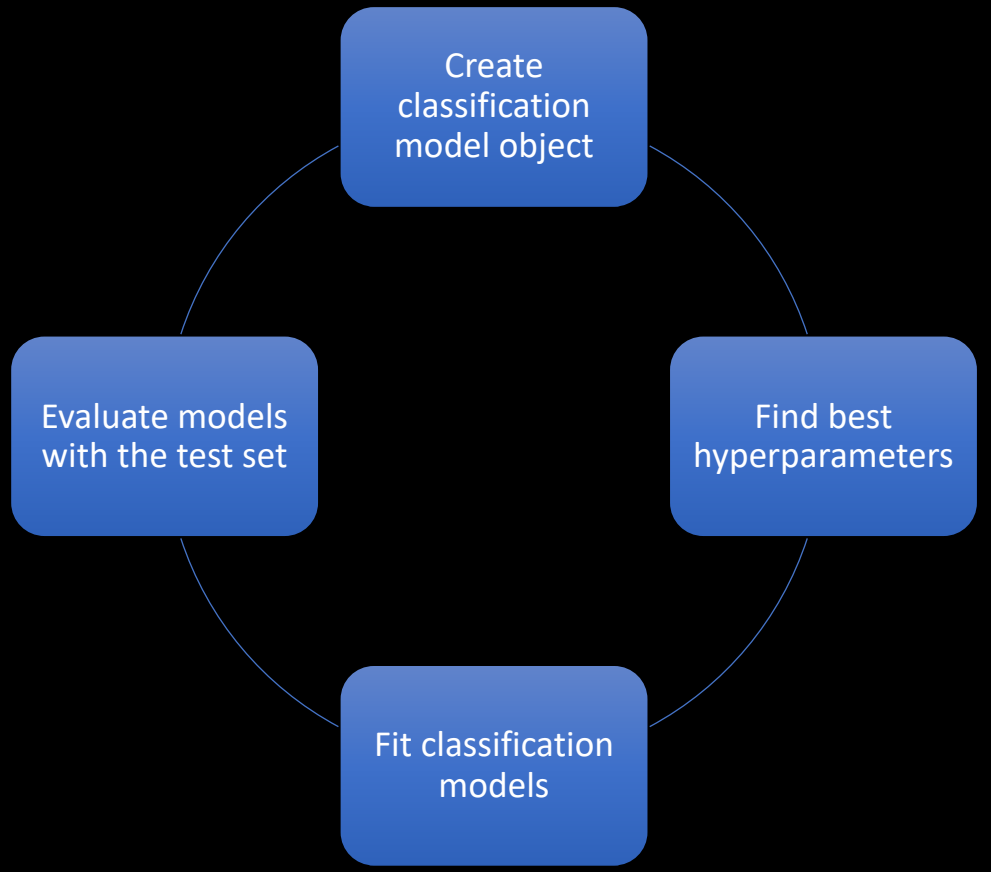
- [ ] Perform EDA and determine training labels
- [ ] Create a column for the class
- [ ] Standardize the data
- [ ] Split into training data and test data
- [ ] Find best hyperparameter
- [ ] Find the best method using test data

Predictive Analysis - Flowcharts

Data preparation process



Model training and evaluation process



Predictive Analysis - Features engineering

- It was selected all features that would be used in the models to predict the success of landing the first stage rocket.
- Dummy variables were created for the categorical columns: Orbits, LaunchSite, LandingPad and Serial.
- All numeric columns were converted to float64.

Selected features

	FlightNumber	PayloadMass	Orbit	LaunchSite	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial
0	1	6104.959412	LEO	CCAFS SLC 40	1	False	False	False	NaN	1.0	0	B0003
1	2	525.000000	LEO	CCAFS SLC 40	1	False	False	False	NaN	1.0	0	B0005
2	3	677.000000	ISS	CCAFS SLC 40	1	False	False	False	NaN	1.0	0	B0007
3	4	500.000000	PO	VAFB SLC 4E	1	False	False	False	NaN	1.0	0	B1003
4	5	3170.000000	GTO	CCAFS SLC 40	1	False	False	False	NaN	1.0	0	B1004

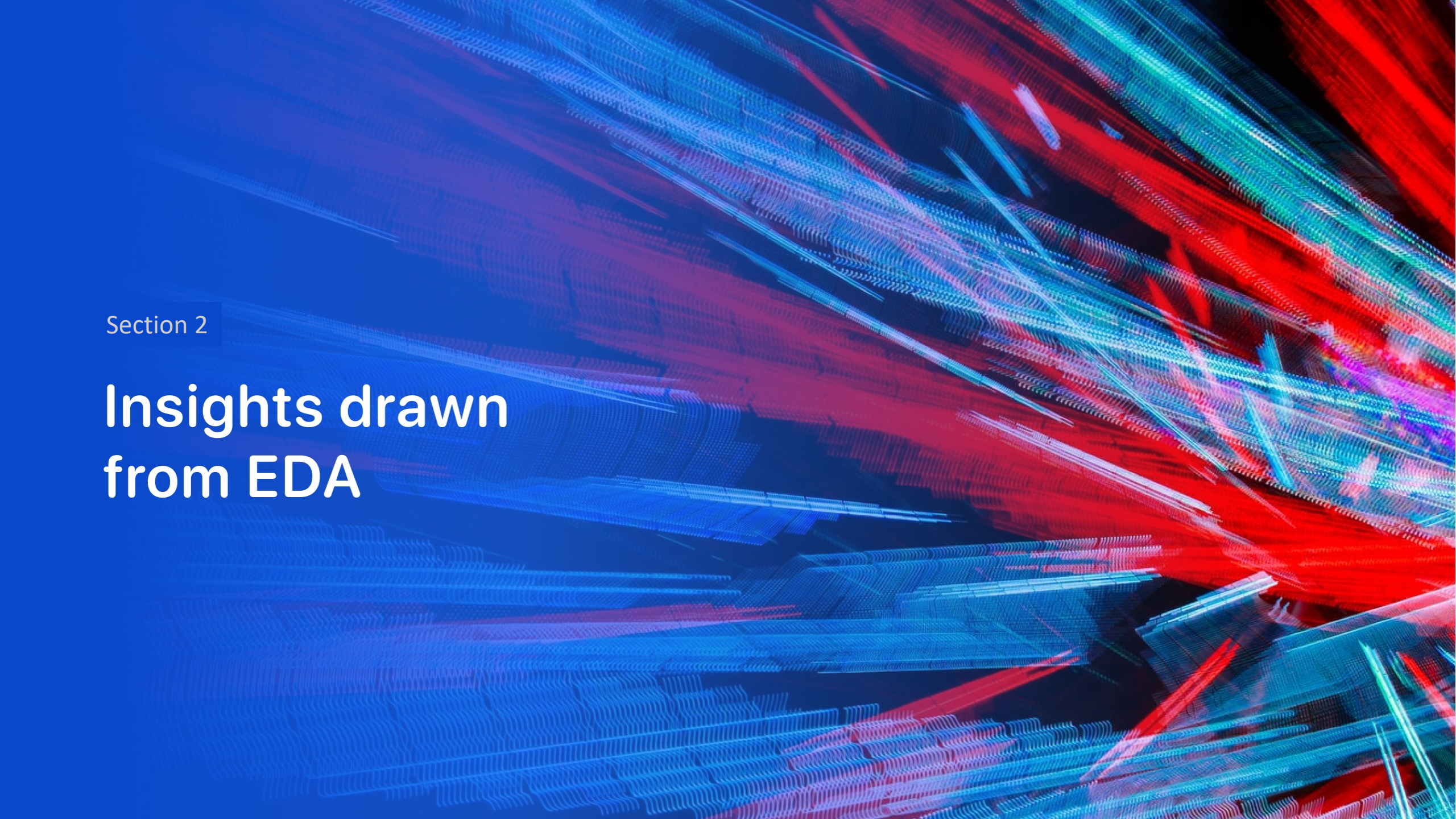
Predictive Analysis – Classification models

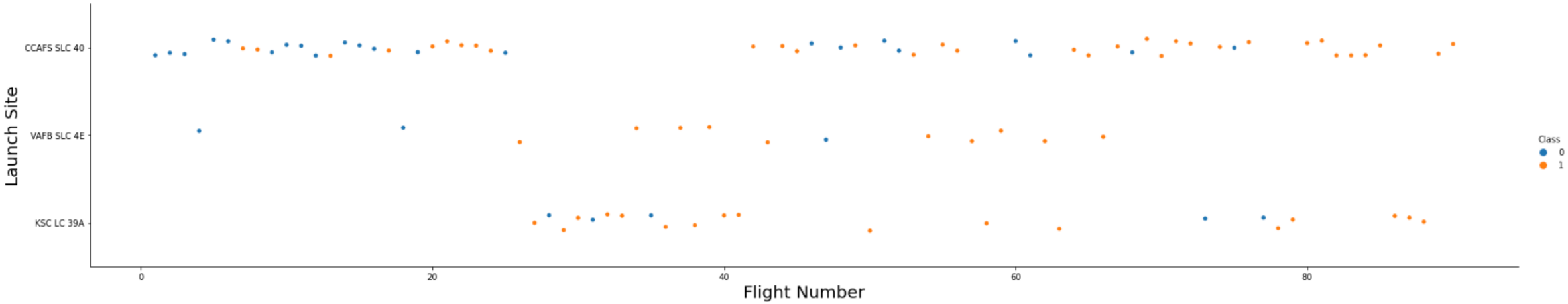
- After features engineering, the data selected were assigned to a variable called X. The data in this variable were standardized with the StandardScaler( ) method.
- A target variable (Y) was assigned with data from the “Class” column and transformed into a Numpy array.
- The variables X and Y were split into two sets: training and test data.
- Using GridSearchCV( ), the best parameters were found. The best parameters for each model can be seen in the table.
- To evaluate the model, the accuracy for each model was calculated using the score method and a confusion matrix was also plotted.

Model	Best parameters
Logistic Regression	{'C': 0.01, 'penalty': ‘12’, 'solver': 'lbfgs'}
SVM	{'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
Decision Tree	{'criterion': 'gini’, 'max_depth': 8, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 10, 'splitter': 'random'}
KNN	{'algorithm': 'auto’, 'n_neighbors': 10, 'p': 1}

Results

- The exploratory data analysis (EDA) allowed us to analyze the relation between variables and to discover the proper ones to use for predicting if a first stage would land successfully.
- The selected columns that were used as features for the models, after EDA stage, were: FlightNumber, PayloadMass, Orbit, LaunchSite, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial.
- The “Class” column was the target for the prediction, that indicates the outcome of the first stage landing.
- Of the four classification models, the Decision Tree model had a better result with the train set, obtaining an accuracy of 0.87.
- With the test set, all models had the same accuracy of 0.83 and the same confusion matrix.

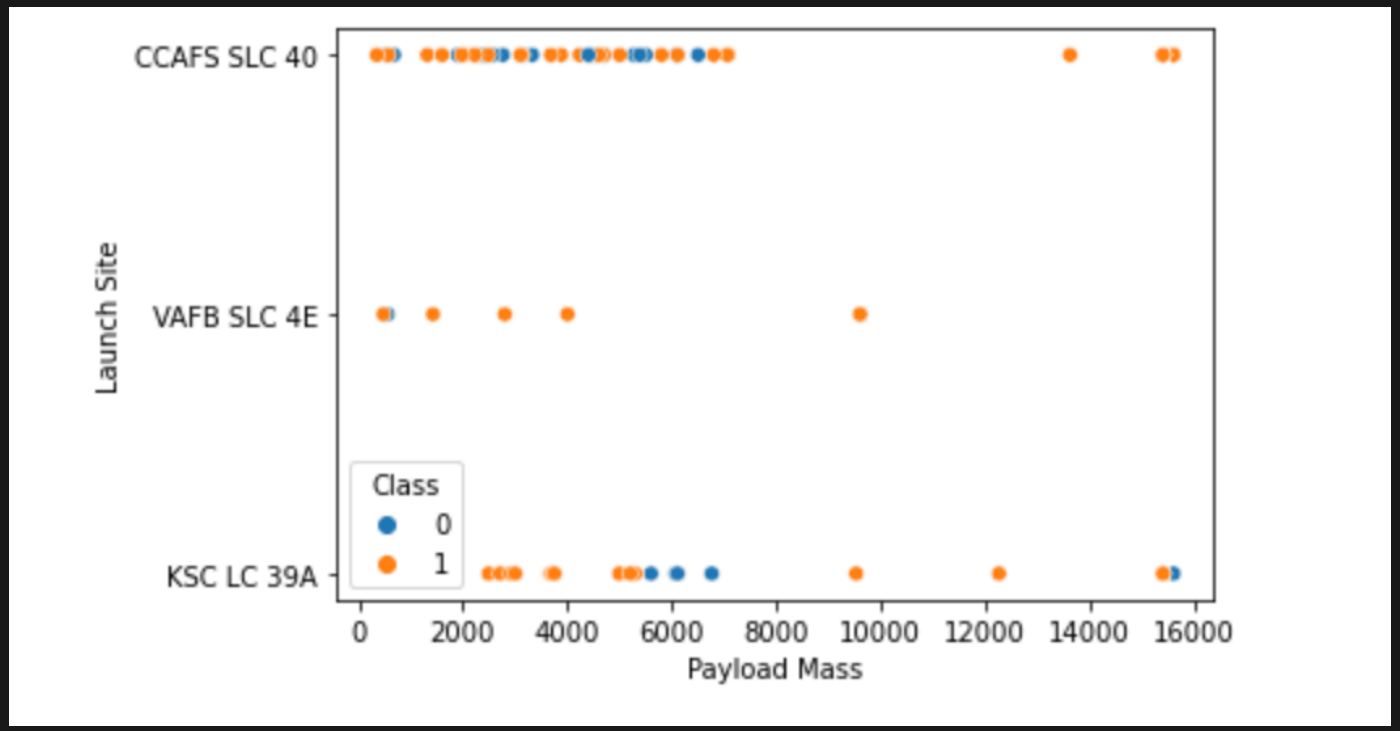

Section 2
Insights drawn from EDA



- The chart shows where each flight was launched and their outcome.
- It is possible to observe that most launches occurred at CCAFS SLC 40.
- It is possible to observe a gap with no flights in CCAFS SLC 40 between flights 25 and 40 and almost all flights in this gap were launched at KSC LC 39A.

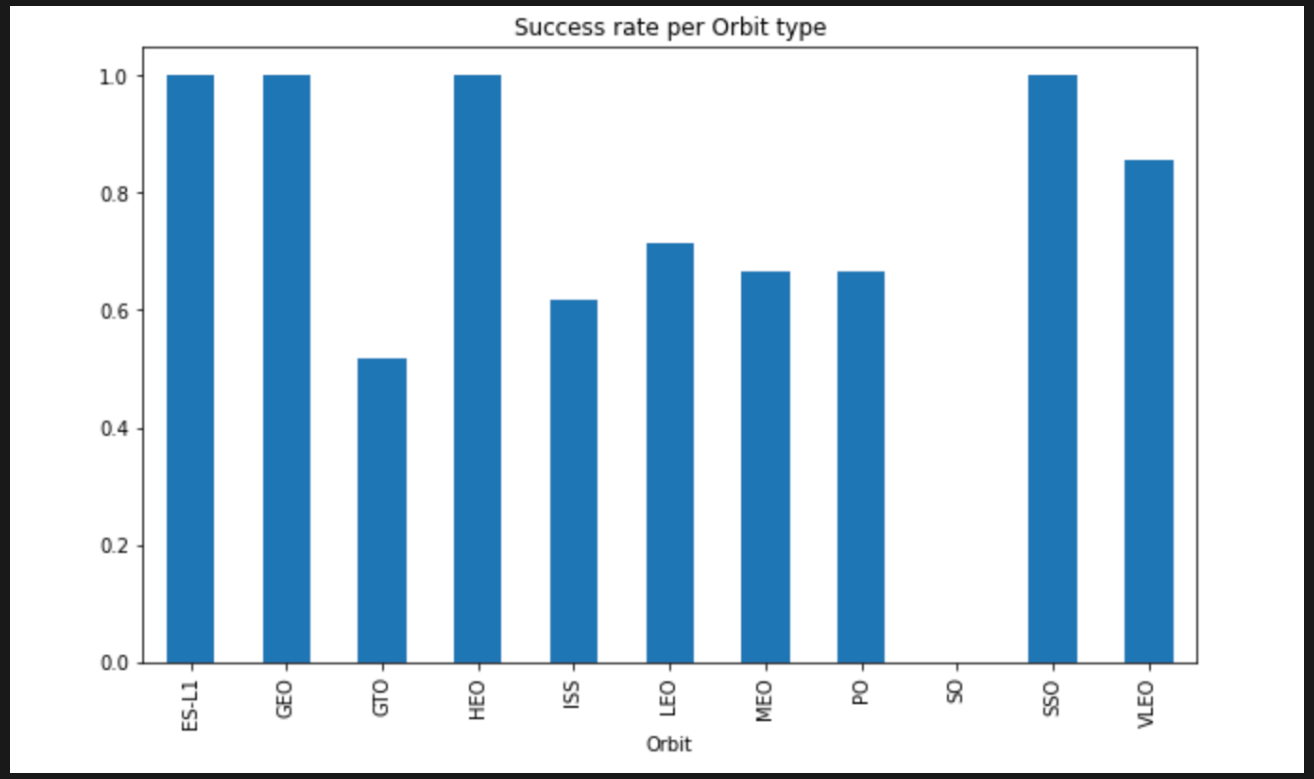
Flight Number vs. Launch Site

Payload vs. Launch Site



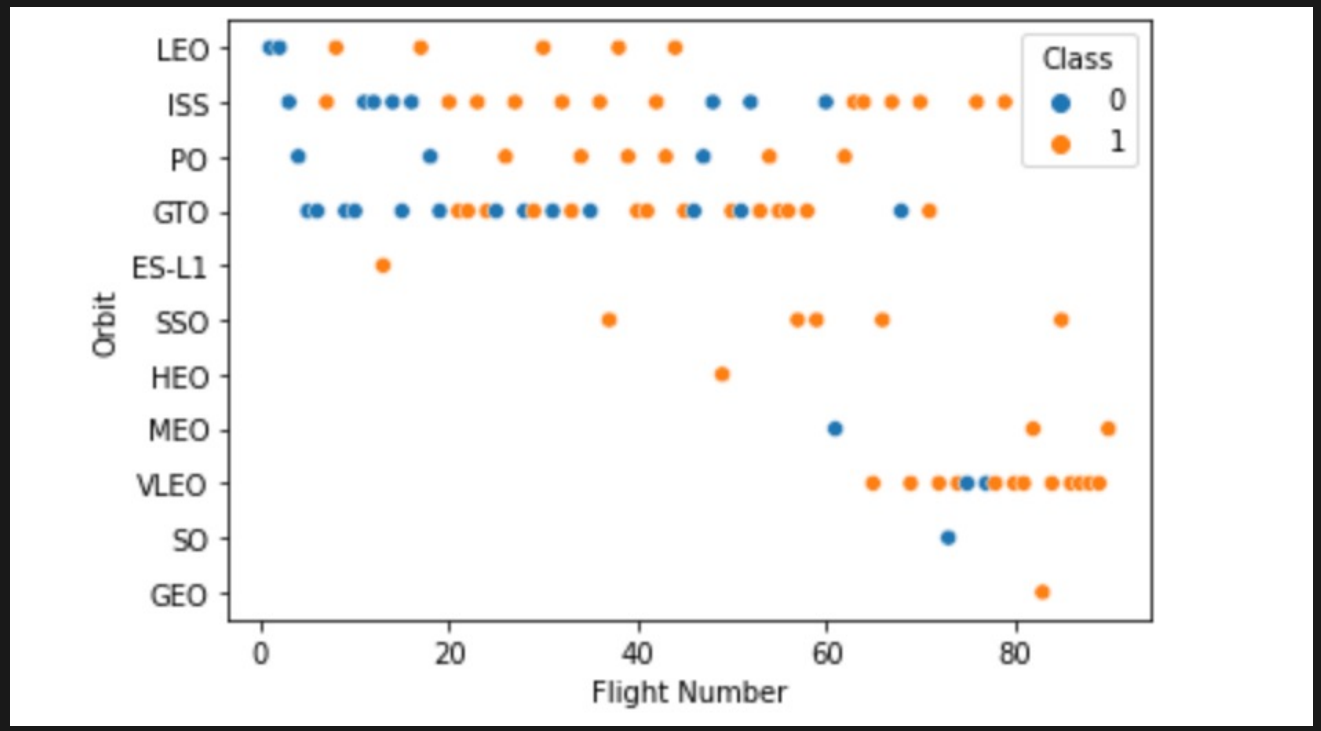
- The chart show the payload mass of each flight per launch site.
- Almost all payload mass are under 10000 kg.
- No heavy payload was launched at VAFB SLC 4E.

Success Rate vs. Orbit Type



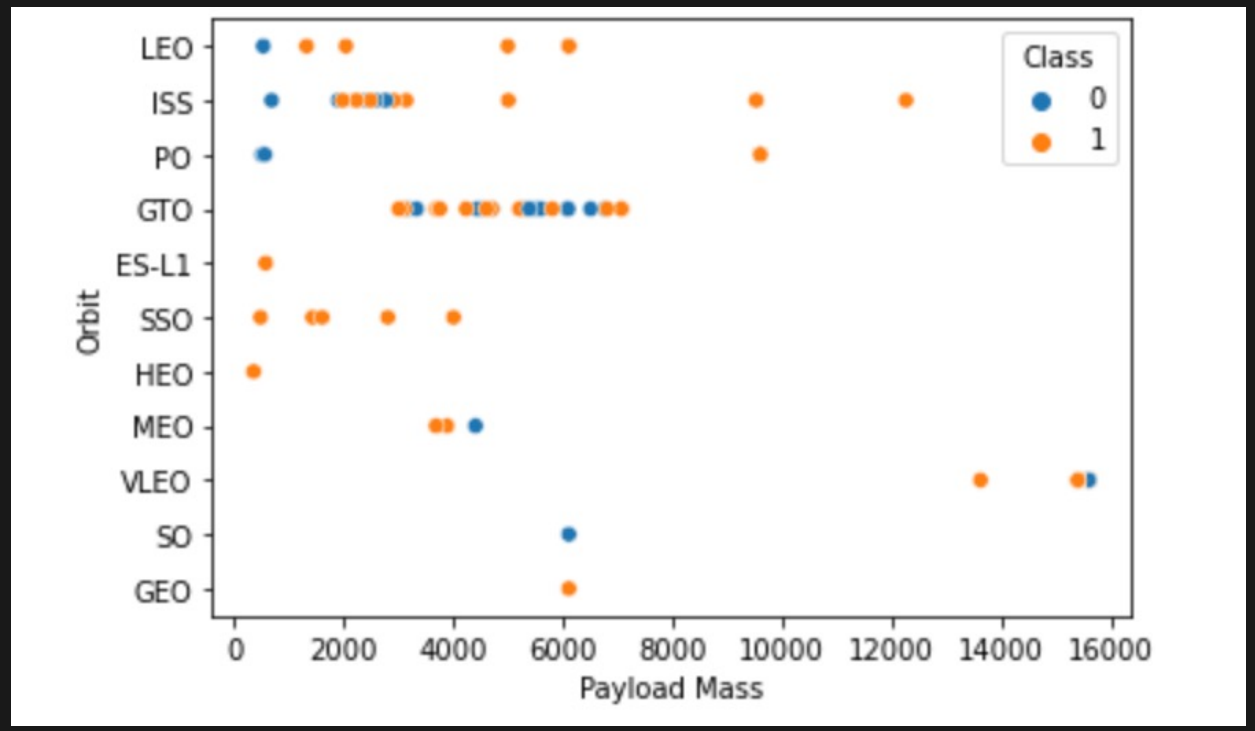
- All missions were successful in four orbits:
  - ES-L1
  - GEO
  - HEO
  - SSO
- The GTO orbit has the least success rate. Half of the missions to GTO were successful in landing the first stage.

Flight Number vs. Orbit Type



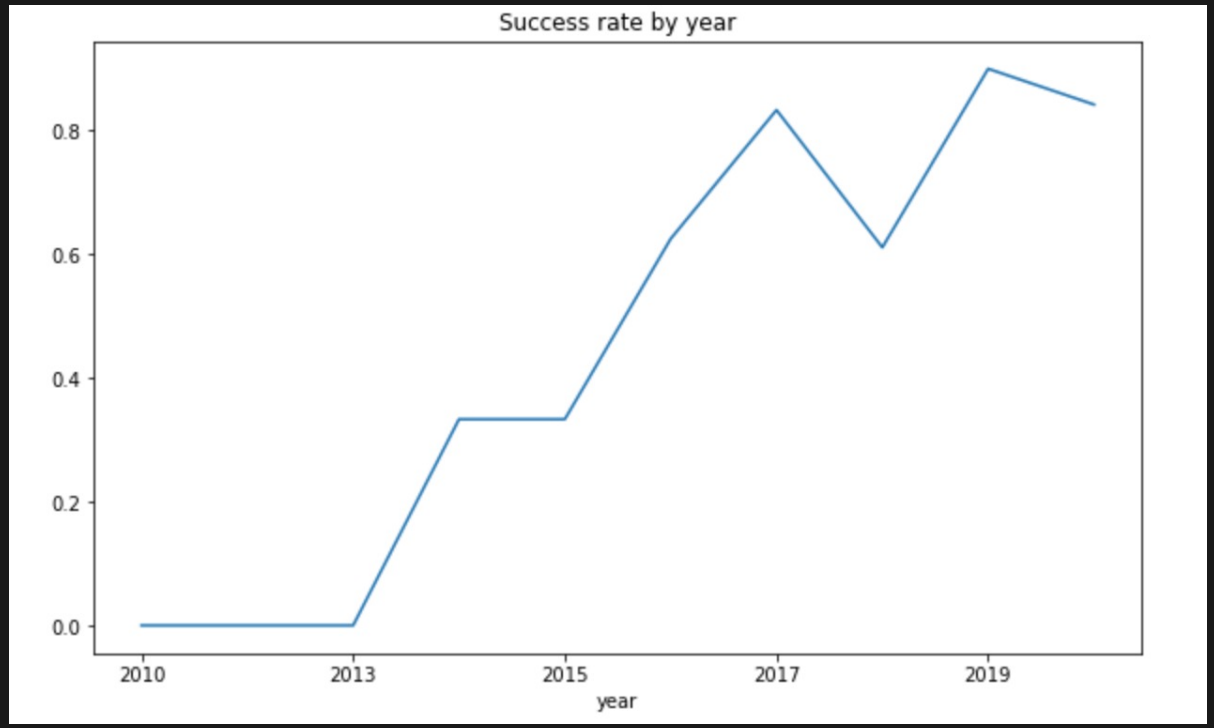
- It is possible to observe that the number of flights to LEO orbit seems to be related with the mission success. After the initial failures, all flights were successful in landing the first stage.
- The same does not apply to GTO orbit, which concentrates the higher number of missions that failed in landing the first stage, but regardless of number of flights.

Payload vs. Orbit Type



- It seems that with payload mass greater than 5000 kg, the rate of successful landings are higher for Polar, Leo and ISS orbits.
- For GTO the same could not be observed, since both landing results (successful and failure) are present, through all payload mass range.

Launch Success Yearly Trend



- With this line chart, it is possible the notice an increase of successful rate from 2013.
- Although there was no year that did not have at least one unsuccessful first stage landing

All Launch Site Names

- The SQL clause SELECT DISTINCT allows to retrieve unique values in the select column from a table.

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

```
 * ibm_db_sa://nh
Done.
```

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

- The first 5 records, with all columns, where Launch Site name began with ‘CCA’ was selected using the WHERE clause combined with the operator LIKE, to be able to find records that matched the initial part of the expression.

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

Python

```
 * ibm_db_sa://nhj38780
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

Launch Site Names Begin with 'CCA'

Total Payload Mass

- The total payload mass carried by boosters from NASA (CRS) was calculated using the aggregator function SUM, that allows all values in the selected column to be summed. To filter for the right records, a WHERE clause was used.

```
%%sql SELECT SUM(payload_mass__kg_) as Sum FROM SPACEXTBL
WHERE customer = 'NASA (CRS)';
```

```
 * ibm_db_sa://nhj3878
Done.
```

SUM
45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1 was calculated using the aggregator function AVG and to find the right records a WHERE clause was used in combination with the operator LIKE, to return all records of F9 v.1.1, regardless of its ending variation.

```
%%sql SELECT AVG(payload_mass__kg_) as Average FROM SPACEXTBL
WHERE booster_version LIKE 'F9 v1.1%';
```

```
 * ibm_db_sa://nhj3878
Done.
```

average
2534

First Successful Ground Landing Date

- To find the first successful ground pad landing date, the MIN aggregator function was used to retrieve to minimum value in the column date. With the WHERE clause, the correct type of landing outcome was filtered.

```
%%sql SELECT MIN(date) as date FROM SPACEXTBL
WHERE landing__outcome = 'Success (ground pad)';
```

```
 * ibm_db_sa://nhj387
Done.
```

DATE
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- To query for the booster versions which have a successful landing and payload mass between 4000 and 6000, a WHERE clause was used in combination with the AND operator.

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

```
 * ibm_db_sa://n
Done.
```

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

Total Number of Successful and Failure Mission Outcomes

- To verify the number of successful and failure mission outcomes, a GROUP BY clause was used in the ‘mission_outcome’ column and their unique values were counted with the aggregator function COUNT.

```
%%sql SELECT mission_outcome, Count(*) as total FROM SPACEXTBL
GROUP BY mission_outcome;
```

```
 * ibm_db_sa://nhj38780:
Done.
```

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

Boosters Carried Maximum Payload

- To find the booster that carried the maximum payload mass, the use of a subquery was required. The query returned booster versions where payload mass was equal to the maximum value for payload mass. To be able to use the MAX aggregator function in the WHERE clause, a subquery was made.

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

```
 * ibm_db_sa://nhj38
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

- The columns with the booster version, landing outcome and the launch site values were selected according to their landing outcome and date, through a WHERE clause and the LIKE operator for finding all 2015 records.

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

```
 * ibm_db_sa://nhj38780:
Done.
```

booster_version	landing__outcome	launch_site
F9 v1.1 B1012	Failure (drone ship)	CCAFS LC-40
F9 v1.1 B1015	Failure (drone ship)	CCAFS LC-40

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

- To find the count of all types of landing outcome between the selected dates, first the records were filtered by date with the WHERE clause. Next, a GROUP BY was used to count the results for each landing outcome.
- Finally, the result was ranked with the ORDER BY clause in descending order.

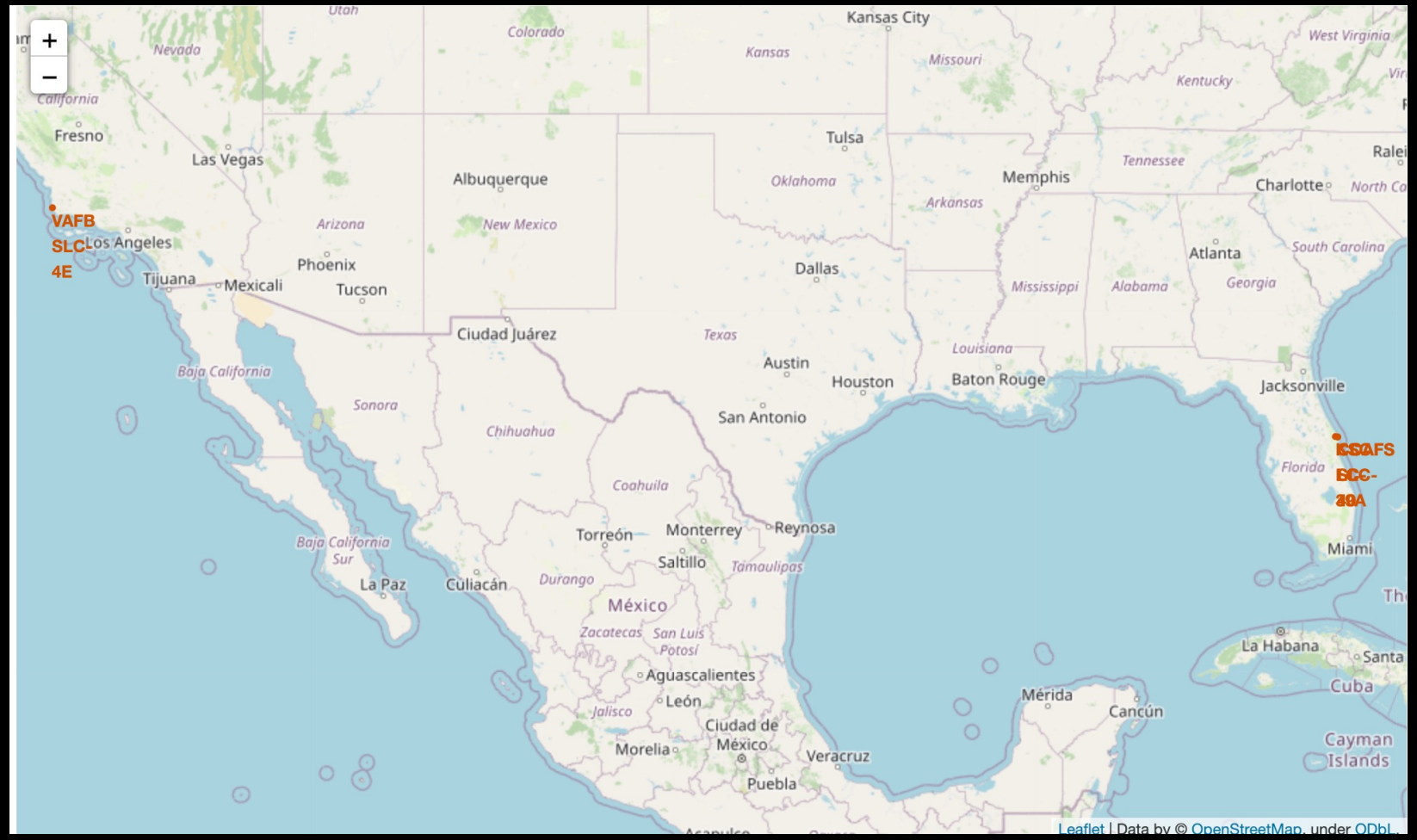
```
%%sql SELECT landing__outcome, COUNT(*) as total FROM SPACEXTBL
WHERE date >= '2010-06-04'
AND date <= '2017-03-20'
GROUP BY landing__outcome
ORDER BY total DESC;
```

```
 * ibm_db_sa://nhj38
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

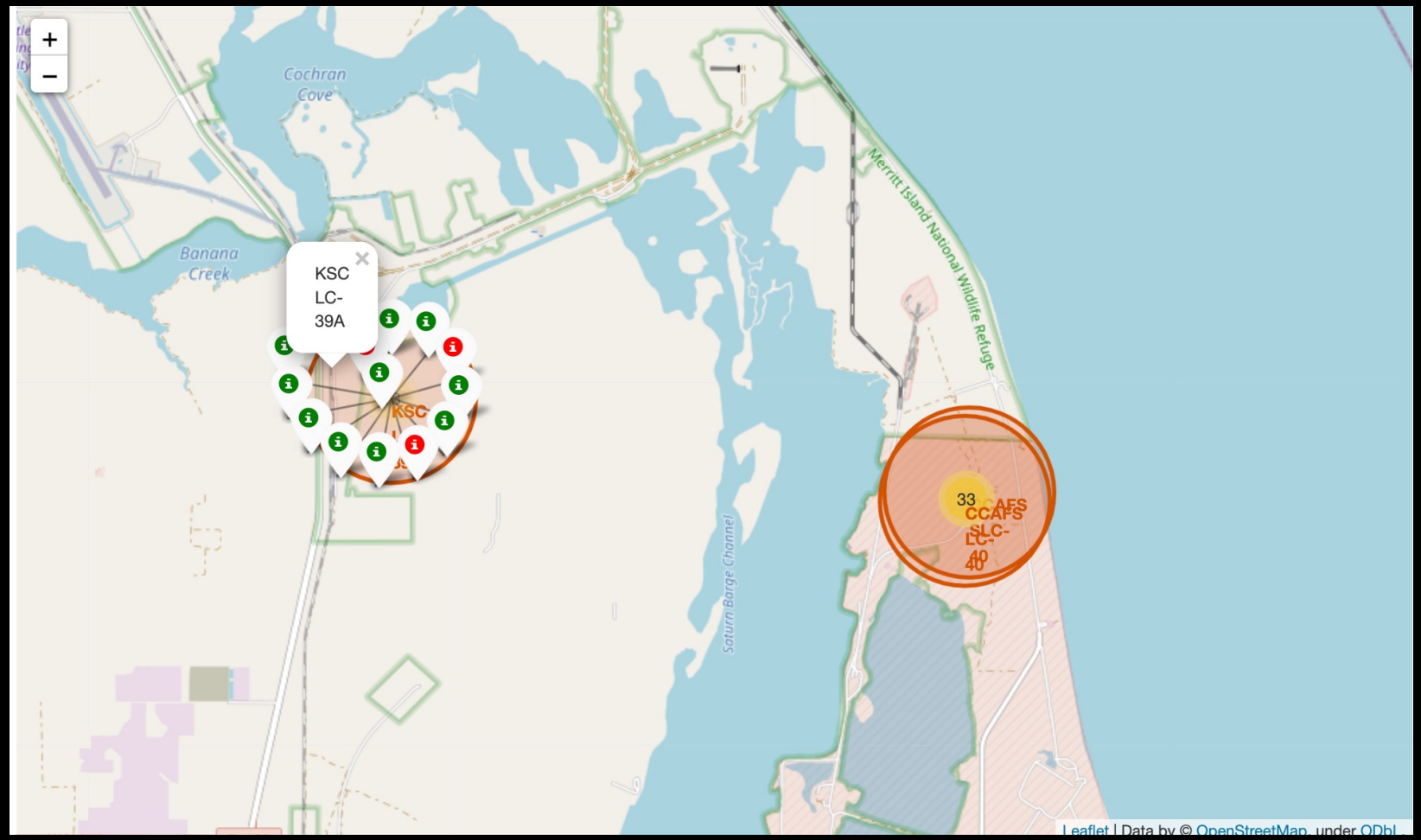
Section 3

Launch Sites Proximities Analysis



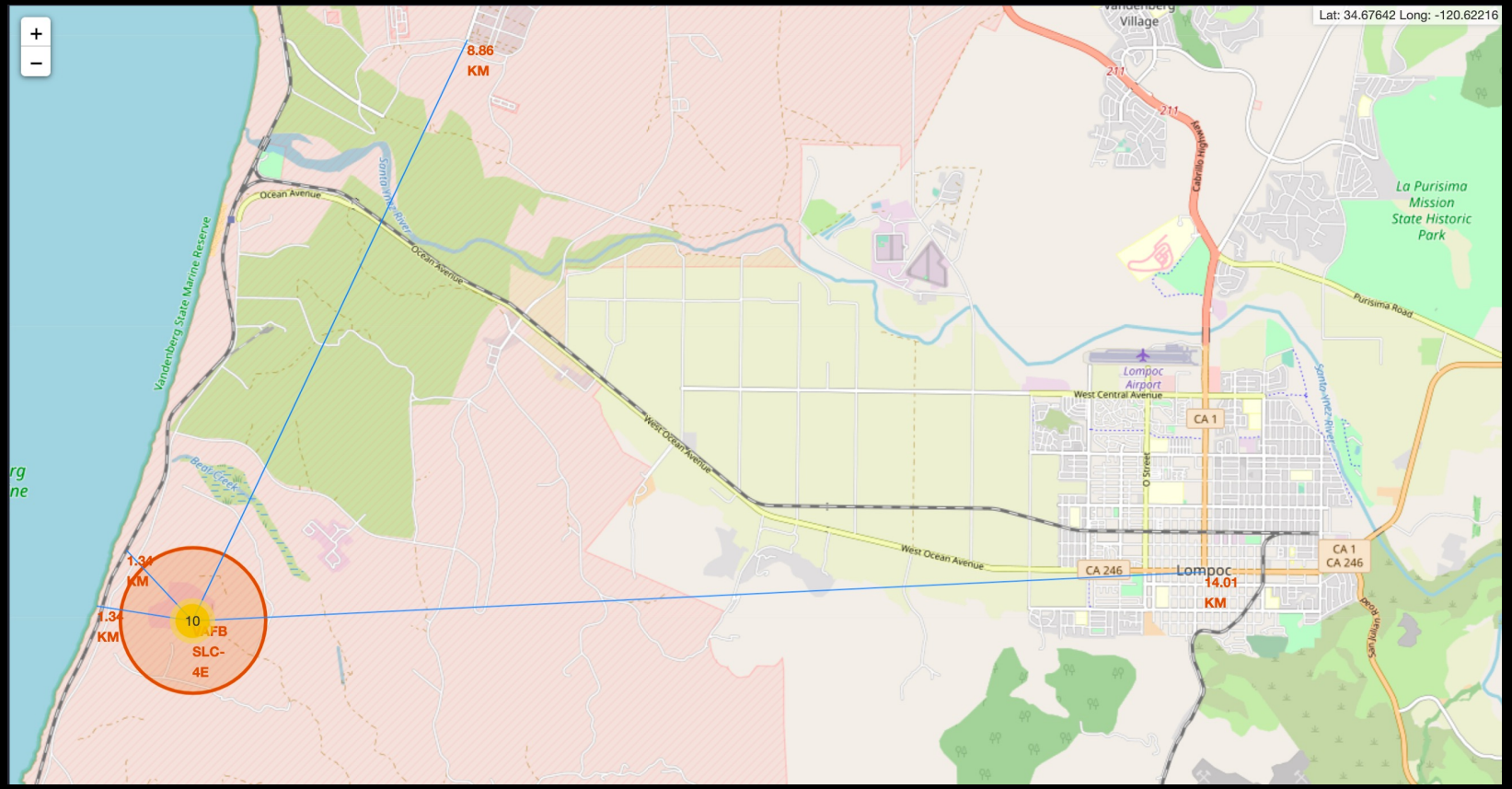
All Launch Sites

- All launch sites were inserted in an interactive world map (a folium Map Object), that allows zooming into each launch location. The latitude and longitude were used for this purpose.
- A circle and a marker was also inserted on the map, to highlight each launch area.
- Most launch sites are in the east coast of the United States.
- All lunch sites are in the south part of the US and very near the ocean.



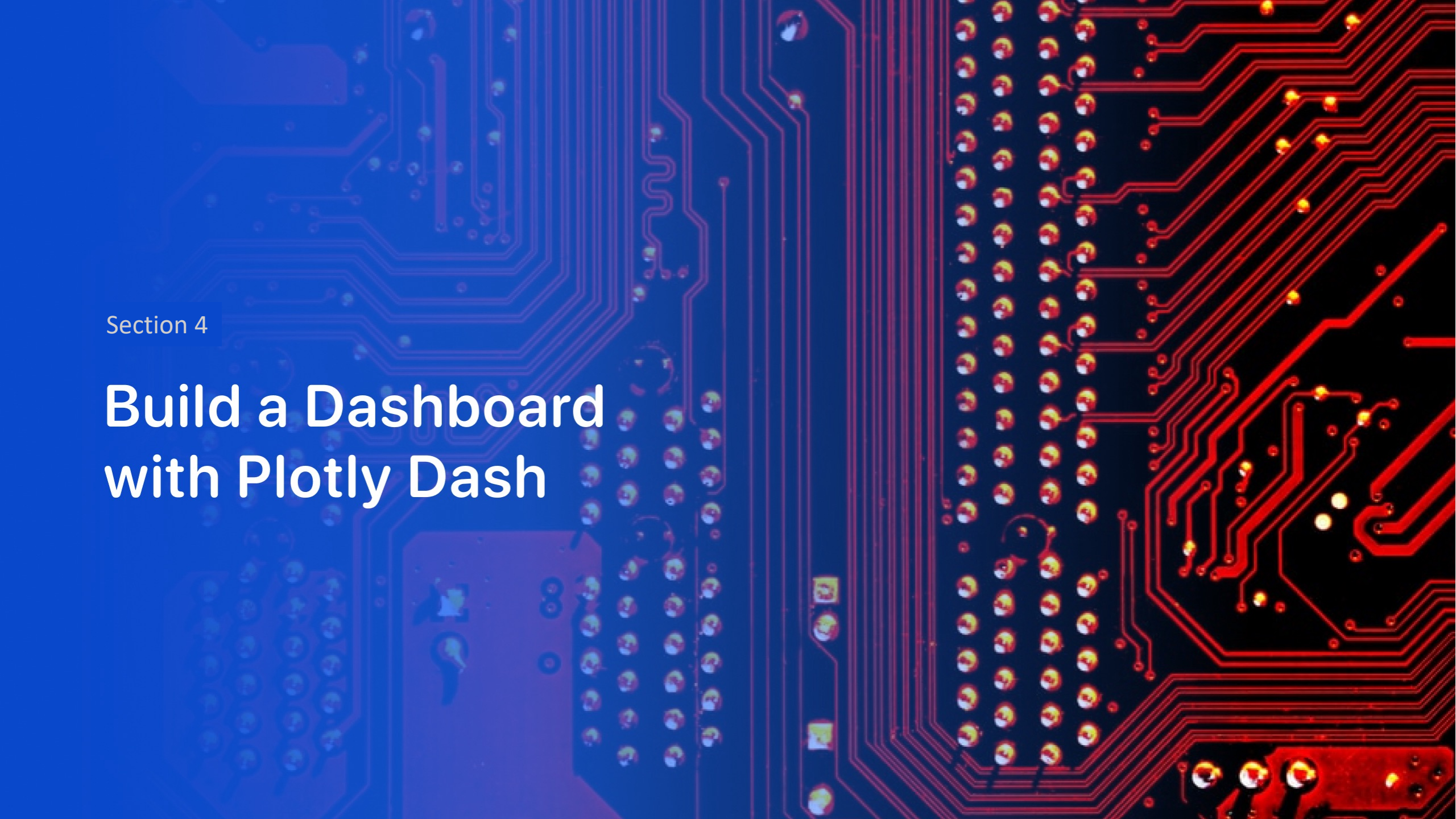
Success and failed outcomes per site

- To be able to observe which site have a high rate of success, the launch outcome was added to the map.
- Color-labeled markers were added for each record, according to its outcome: green for successful ones, and red to unsuccessful.
- A MarkerCluster object is used to insert many markers that have the same coordinate, simplifying the map.
- The site that was chosen in the image, shows that this site launch (KSC LC- 39A) had 3 unsuccessful outcomes and 10 successful.



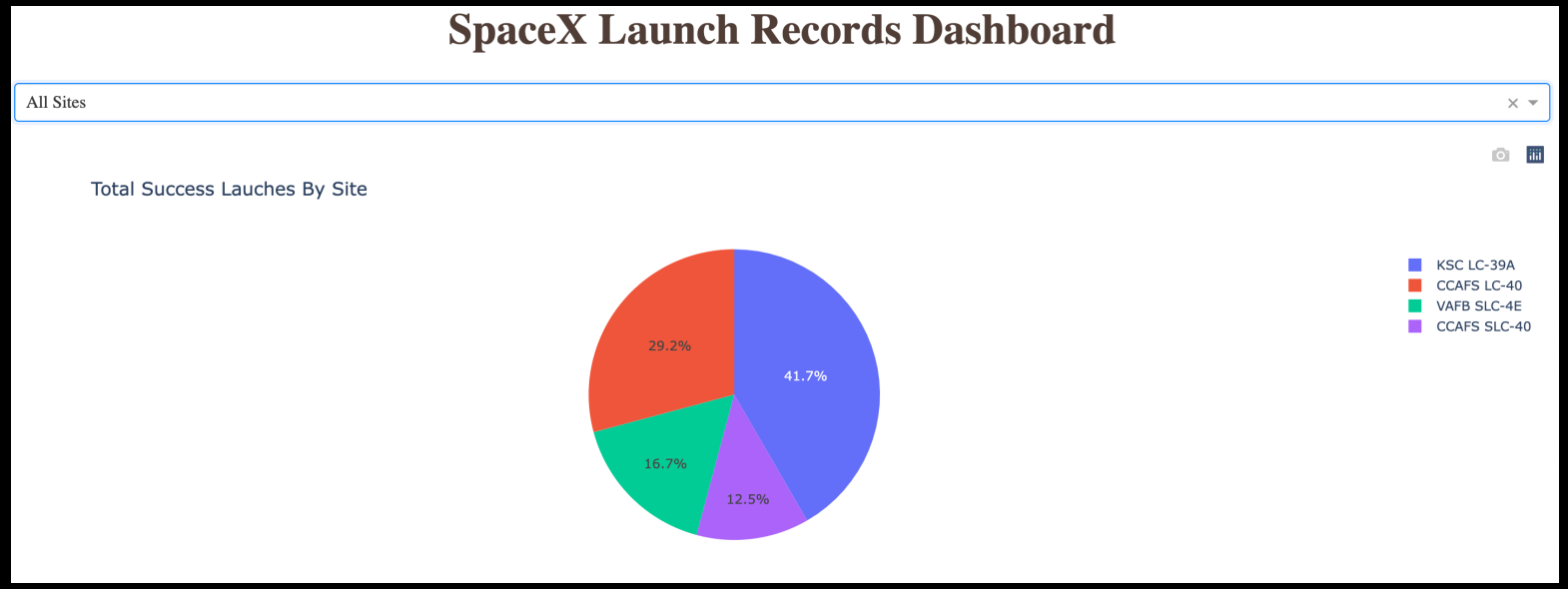
Proximities of launch site

- First, a MousePosition object was added on the map to get coordinate for points of interests.
- Several nearby sites were analyzed, and their coordinates collected, such as: railways, highways, coastline and cities.
- Markers were inserted for one of each point of interest, and the distance between them and the launch site was calculated.
- A PolyLine was draw connecting the launch site with each point of interest, and the distance was displayed, as well.
- It is possible to observe how close the coastline is, specially comparing with the nearest city.


Section 4
Build a Dashboard with Plotly Dash

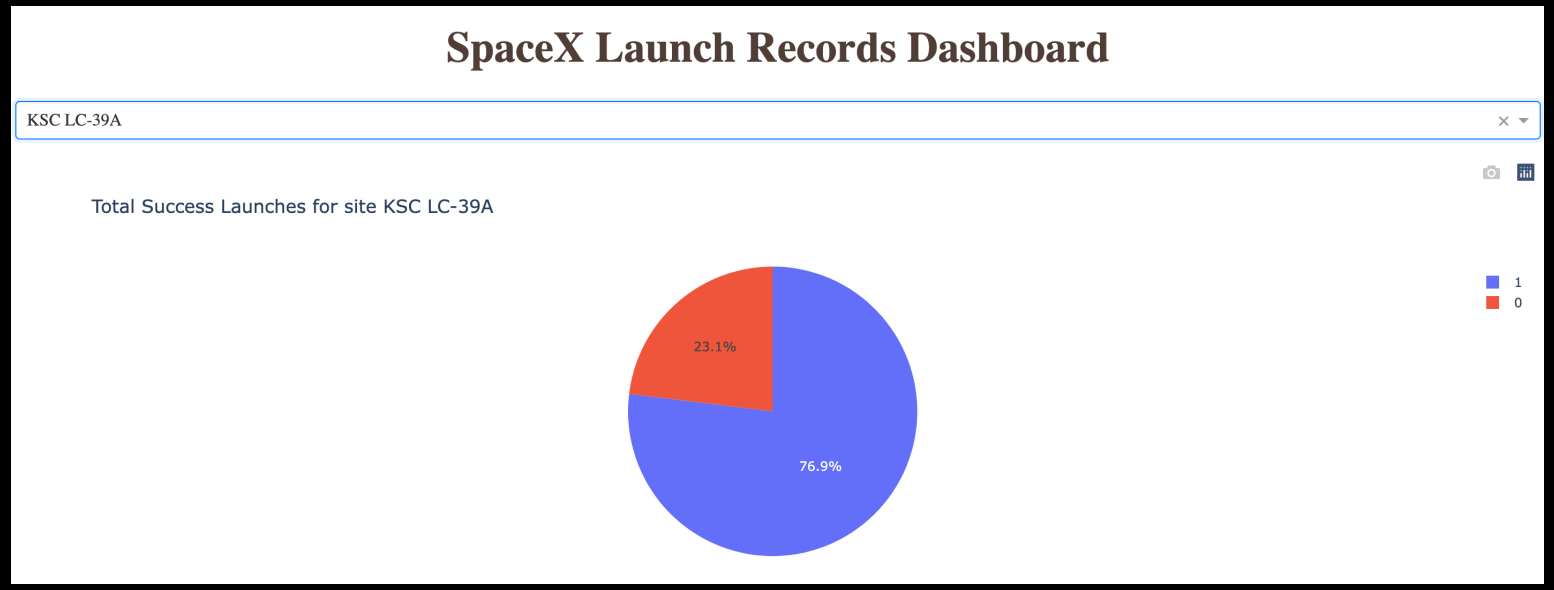
Total success launches by site

- The pie chart shows the success outcome by launch sites.
- KSC LC 39A have the most successful outcome results of all launch sites. 41,7% of all successful outcomes were launched at this site.
- CCAFS SLC-40 have the least successful outcome results.



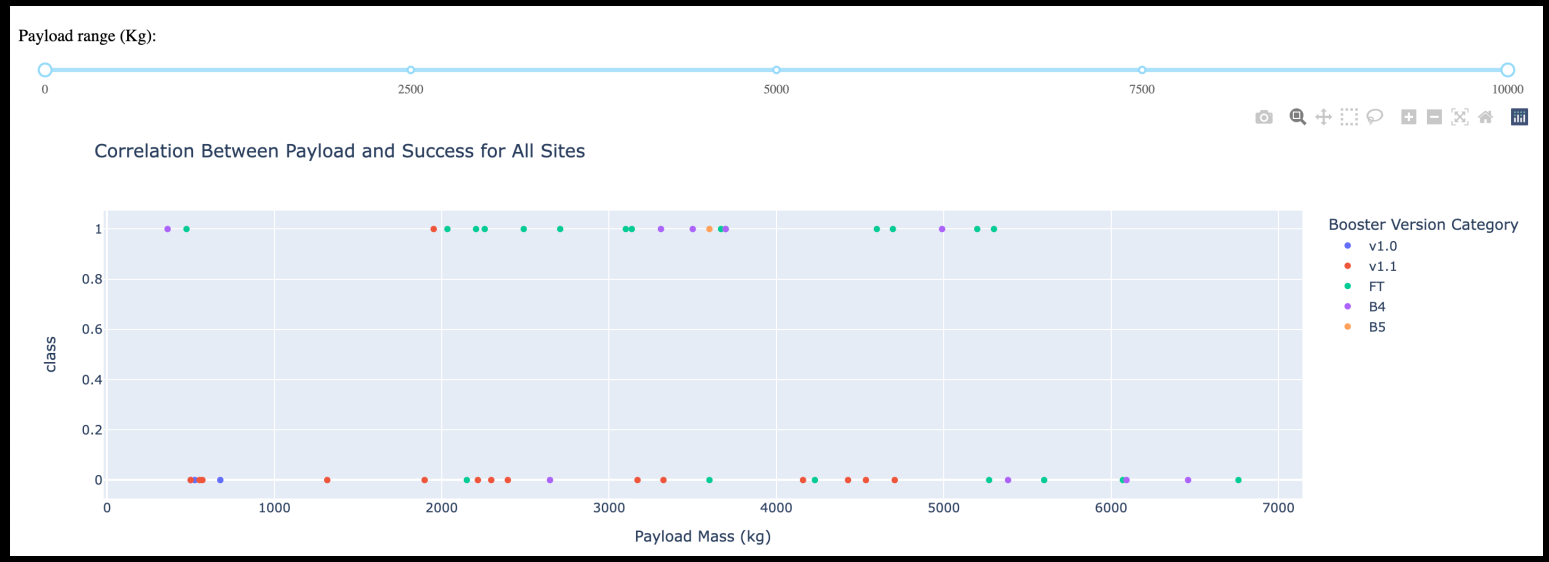
Site with the highest launch success ratio

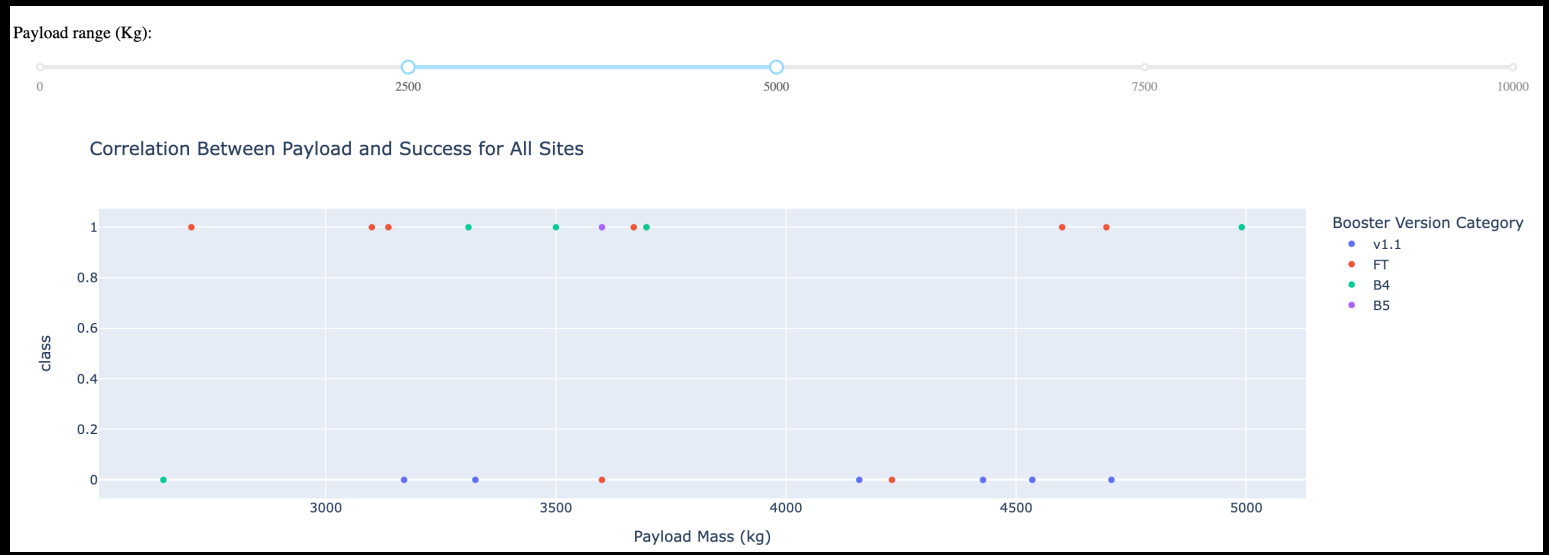
- The pie chart shows the success and failure outcomes for KSC LC-39A.
- Considering all launches at KSC LC-39A, this site have a good outcome rate. Almost 77% of its launches had a successful outcome.
- All other sites had less than 50% of successful outcomes.



Payload vs. Launch Outcome

- The scatter plots show the relation between payload mass and launch outcomes by booster version.
- The dashboard offered a slider that enables filtering results by a range of payload mass.
- Observing booster versions, the FT booster had more successful outcomes and the V1.1 had more failures.
- For heavier payloads, only two booster versions had launches. Both had more failures than successful outcomes.



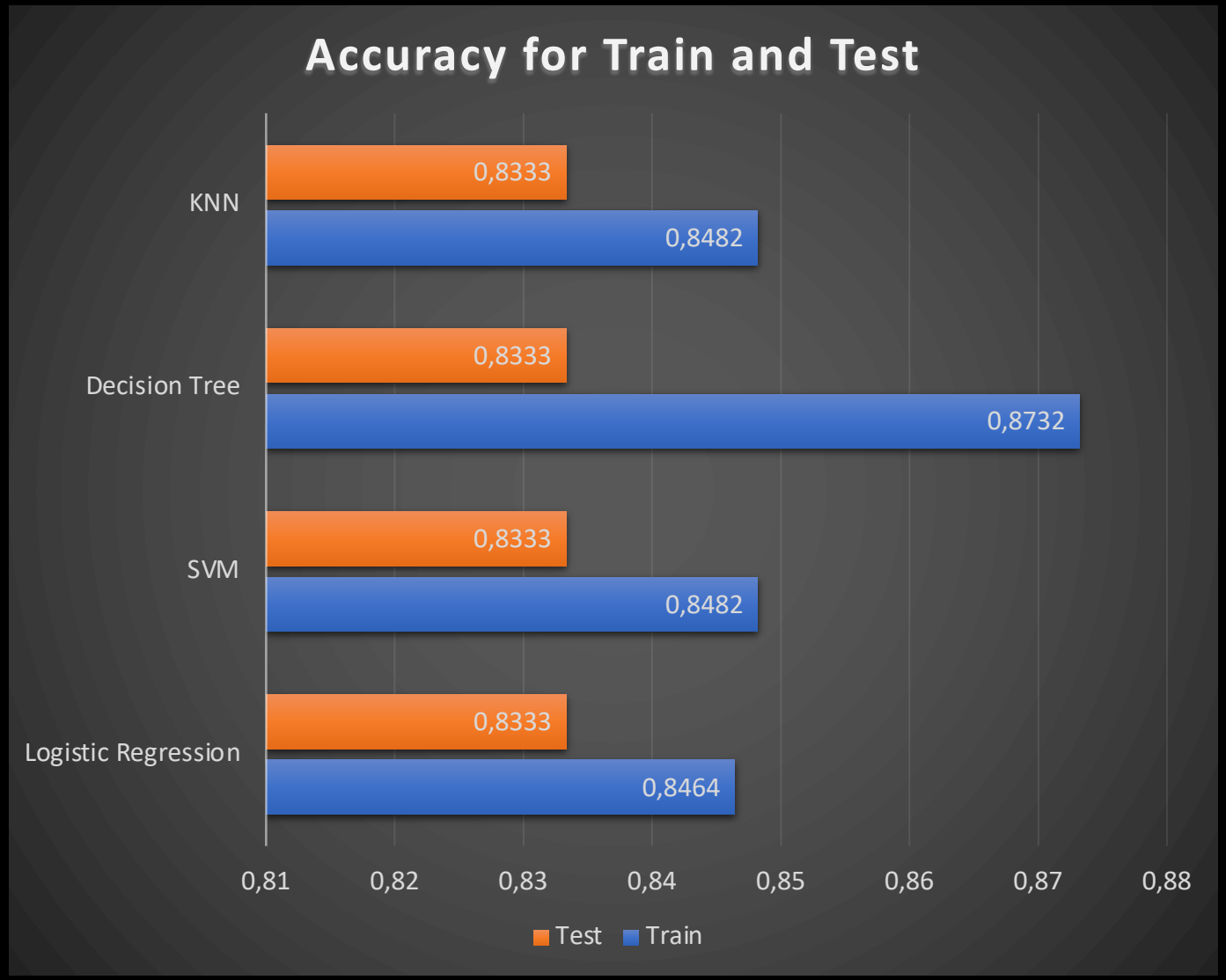


Section 5

Predictive Analysis (Classification)

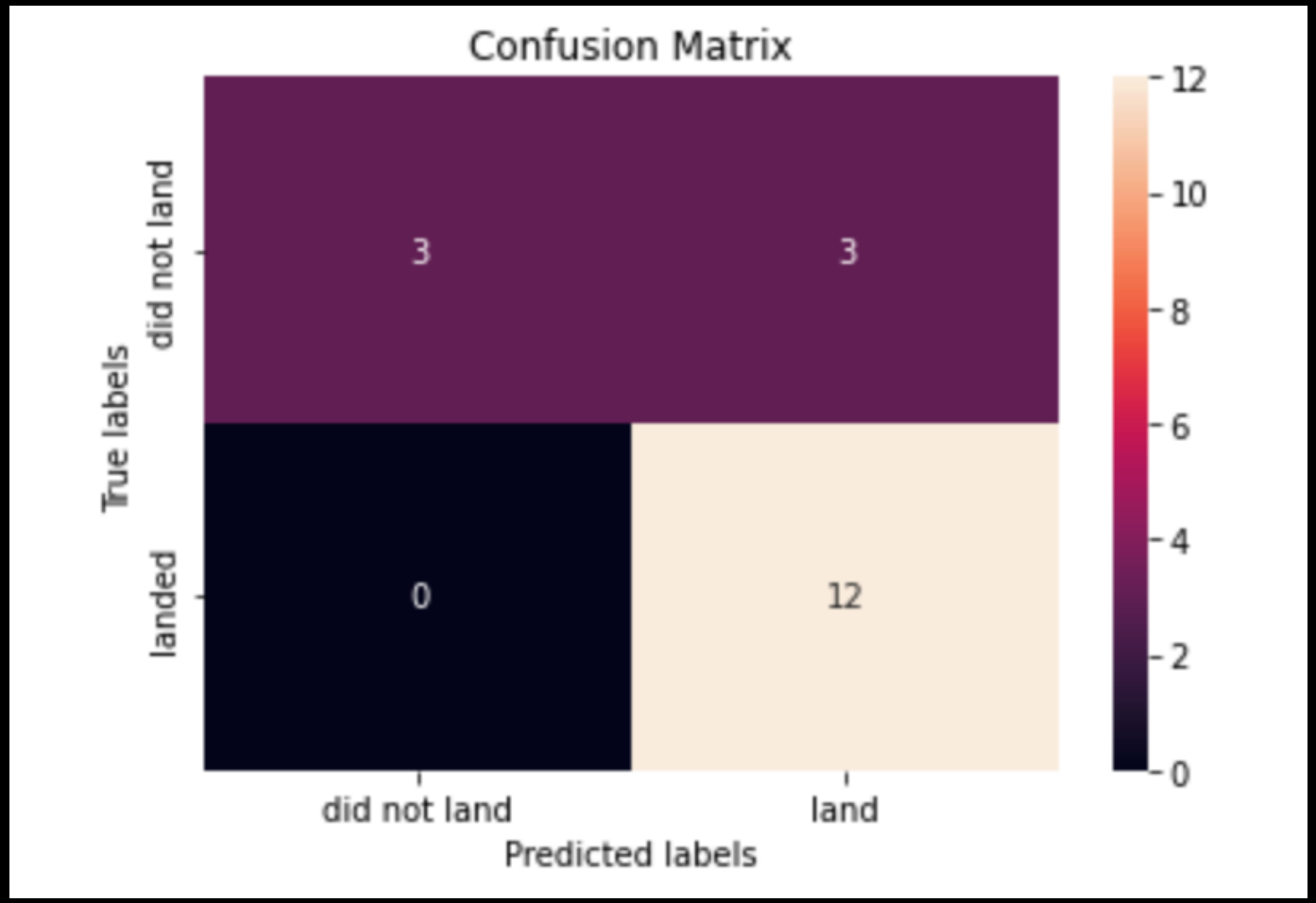
Classification Accuracy

- All models had the same accuracy score for the test set.
- As for the train set, the Decision Tree model had a slightly better accuracy than the others.



Confusion Matrix

- All models had the same Confusion Matrix.
- In the confusion matrix, it is possible to see that the sensibility is high, since the models predict correctly all first stage that did not land.



Conclusions

- All classification models built and evaluated for this project had the same result with the test data, although the Decision Tree model had a slightly better result with the train data.
- Observing the confusion matrix, it is possible to see that the models have a high sensibility, obtaining false positives.
- Considering the cost of losing the first stage, it would be better to have more specificity, since false positives may lead to unplanned costs.
- Despite the high sensibility, all models had a good accuracy and would predict the expected outcome with good confidence.

References

- SpaceX. (2022). SpaceX homepage. Retrieve April 10, 2022 from https://www.spacex.com/
- Wikipedia. (2022a). List of Falcon 9 and Falcon Heavy launches. Retrieved April 10, 2022, from https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1081249351
- Wikipedia. (2022b). SpaceX. Retrieved April 10, 2022, from https://en.wikipedia.org/w/index.php?title=SpaceX&oldid=1081243394

Appendix A – List of Success rate by year

- This code shows the successful rate of landing the first stage rocket by year.

```
success_rate_year = df1.groupby("year")["Class"].mean()
success_rate_year
```

```
]: year
   2010    0.000000
   2012    0.000000
   2013    0.000000
   2014    0.333333
   2015    0.333333
   2016    0.625000
   2017    0.833333
   2018    0.611111
   2019    0.900000
   2020    0.842105
   Name: Class, dtype: float64
```

Thank you!