IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Tudor Bozan
22.08.2025

Outline

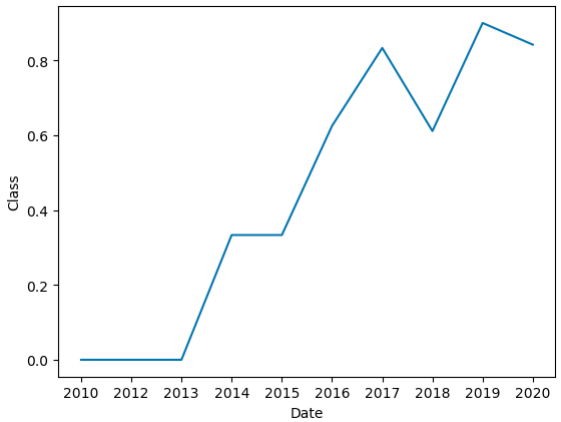
- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- **Project Overview**

Goal: Predict landing success of Falcon 9 (0 = fail, 1 = success).
Data Sources: SpaceX API and Wikipedia (web scraping)
Tools Used: Pandas, Matplotlib, Seaborn, Folium, Plotly Dash, Scikit-learn

- **Visual Insights**



Success rate increases with time

- **Dashboard Summary**

Plotly Dash App:
Interactive filters: Site, Payload
See success rate per Site and how the Payload influences it

- **Model Performance**

Model	Accuracy (train)	Accuracy (test)
Logistic Regression	82%	83.33%
Decision Tree	83.4%	83.33%
Support Vector Machine	84.8%	83.33%
K-Nearest Neighbors	84.8%	83.33%

- **Recommendations**

- this project's solutions can already be used, lightweight, to predict successes based on standard SpaceX feature set with an accuracy over 83-84 %
- a larger dataset could be used in the future

Introduction

Project background

This project aims to create a model for predicting the failure or success of the Falcon 9 landing (first stage).

If this prediction is possible by using available data, then we enable for cost advantage potential, as SpaceX first stage is reusable, compared to competition.

Questions -> focus of the analysis

Based on main features **X**, will the Falcon 9 land or fail to?
What is the performance of the model we can create?
How do several versions of the model compare to each other?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

  → SpaceX REST API and scraping the Wikipedia report tables

- Performed data wrangling

  → From the original data set, we converted the Landing Outcome strings that represented True and False landings in different circumstances to 1 and 0, respectively.

- Performed exploratory data analysis (EDA) using visualization and SQL
- Performed interactive visual analytics using Folium and Plotly Dash
- Performed predictive analysis using classification models

  → Built, tuned, evaluated classification models

Data Collection – SpaceX API

- The data was collected by using the SpaceX REST API at **https://api.spacexdata.com/v4/** accessed on 1st of August 2025.

  → A subset of features was used

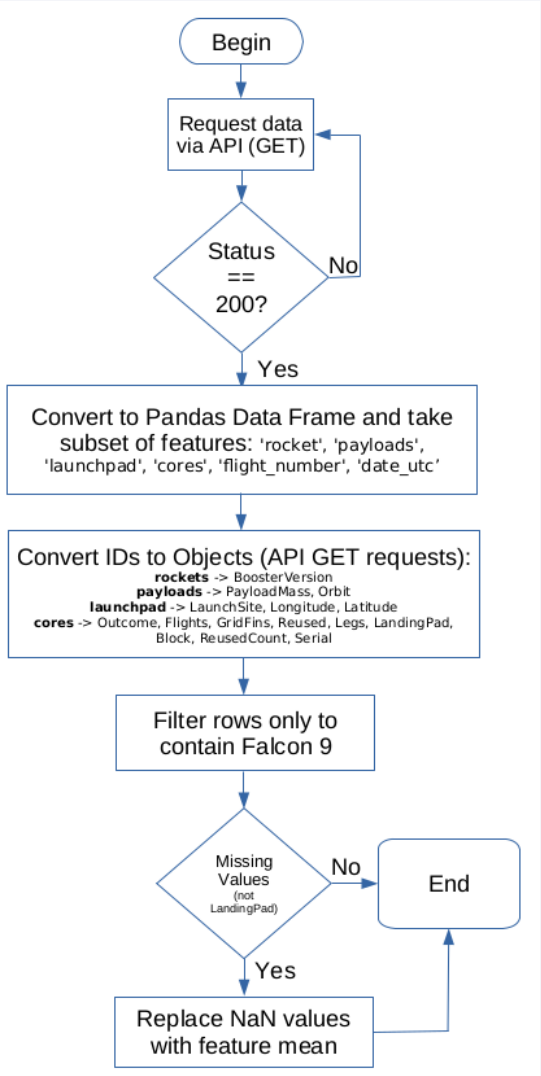
  → Using further API calls the original IDs were converted into usable strings

  → The final set was filtered to ‘Falcon 9’ records only

  → Missing values (not LandingPad) were replaced by feature mean

- GitHub URL of the completed SpaceX API calls notebook:

  https://github.com/tudorbozan/IBM-Data-Science-Specialization-Capstone-Project/blob/main/Lab1-jupyter-labs-spacex-data-collection-api.ipynb



Data Collection – SpaceX API

- The Data Frame looked like this when the process was done:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
4	1	2010-06-04	Falcon 9	6123.547647	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857
5	2	2012-05-22	Falcon 9	525.000000	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.561857
6	3	2013-03-01	Falcon 9	677.000000	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.561857
7	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34.632093
8	5	2013-12-03	Falcon 9	3170.000000	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28.561857

Data Collection - Scraping

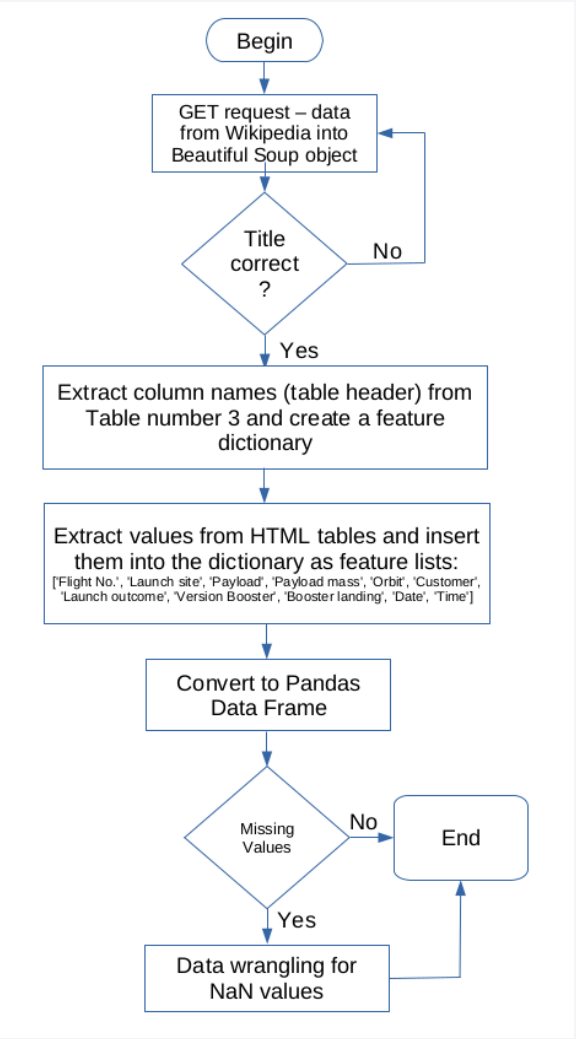
- Read data via GET requests into a BeautifulSoup object from:
  https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
  accessed on 2nd of August 2025

  → Extracted table headers from the HTML tables

  → Parsed the contents into a Pandas Data Frame

- GitHub URL of the completed web scraping notebook:

  https://github.com/tudorbozan/IBM-Data-Science-Specialization-Capstone-Project/blob/main/Lab2-jupyter-labs-webscraping.ipynb



Data Wrangling

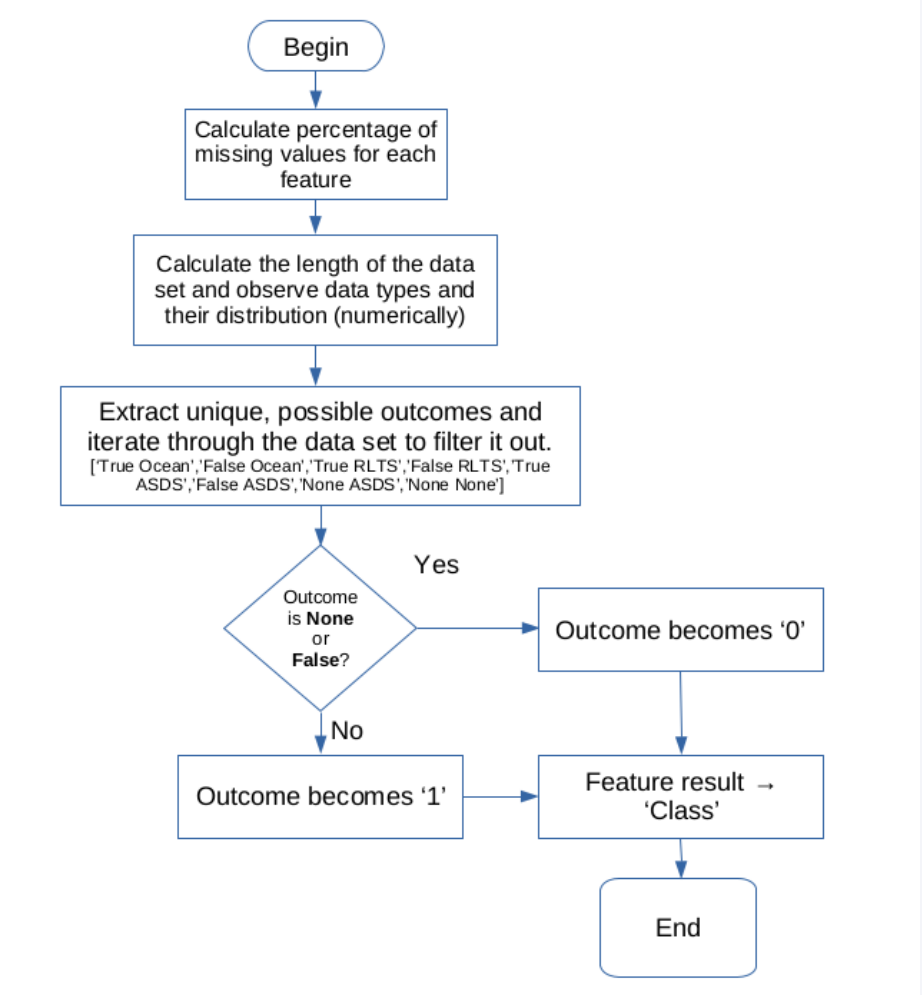
- Data wrangling process

  → Basic statistics on data set

  → Converted the original Outcomes with many possible values to a new target feature → **Class**, that contains only **1** and **0** for **success** and **failure** of the landing, respectively

- GitHub URL of the completed web scraping notebook:

https://github.com/tudorbozan/IBM-Data-Science-Specialization-Capstone-Project/blob/main/Lab3-labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

During EDA, the data has been looked at by means of the following charts:

- Scatter plot of **Payload Mass vs Flight number**, with 3rd dimension being success (1 or 0) → so we could see if there is a trend, as the Flight number increased, also the proportion of successes increased (True)

- Scatter plot of **Launch Site vs Flight number**, with 3rd dimension being success (1 or 0) → so we could see if the trend is kept also for different sites (True)

- Scatter plot of **Launch Site vs Payload Mass**, with 3rd dimension being success (1 or 0) → so we could see if the masses are distributed uniformly (False)

- Bar plot of **Success rate for each Orbit** → could identify the most successful launch Orbits

- Scatter plot of **Orbit vs Flight Number** → to see whether there is a trend (True for some Orbit)

- Scatter plot of **Orbit vs Payload Mass** → to see whether the successes are clustered around some mass values (True for some)

- Line plot of **Success rate vs Year** → so we could see the increase in success rate over time

GitHub URL of completed EDA with data visualization notebook:

https://github.com/tudorbozan/IBM-Data-Science-Specialization-Capstone-Project/blob/main/Lab5-edadataviz.ipynb

EDA with SQL

To assess the data before modeling, the following SQL queries have been used:

- Find out the unique Launch Site names

- Find 5 records where the name o a launch Site starts with ‘CCA’

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

- Display the average payload mass carried by booster ‘F9 v1.1’

- Find the date when the first successful landing outcome in ground pad happened

- Find the names of boosters which have success in drone ship and have payload mass grater than 4000 but below 6000 kg

- List the total count of successes and failures

- List all booster versions that have carried the maximum payload mass

- For year 2015, find the month name, failed outcomes for drone ship, booster version and launch site

- Rank all landing outcomes count between 2010-06-04 and 2017-03-20, in descending order

GitHub URL of completed EDA with SQL notebook:

https://github.com/tudorbozan/IBM-Data-Science-Specialization-Capstone-Project/blob/main/Lab4-jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

To have a better overview, the launch sites have been plotted on a Folium map:

- For each Launch Site, a circle was used to point to the location, along with a marker next to it, to show the site name

- For each Launch Site, the successes and failures were marked as location markers (green for success and red for failure), by using a Marker Cluster

- Distance lines (aerial) were lastly plotted on the map to assess the locations proximity to towns, coast lines, railways or roads

GitHub URL of completed interactive map with Folium map:

https://github.com/tudorbozan/IBM-Data-Science-Specialization-Capstone-Project/blob/main/Lab6-Folium-lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

By using a Plotly Dash app, two user interactions with associated charts have been used:

- Select one particular site or all sites from **a drop-down** → **pie chart** for success rate
    - If All sites are selected → plot true rate for each in a pie
    - If one site is selected → plot true rate vs. false rate for that site in a pie
- Select the **payload mass range (kg)** by using a **slider**:
    - For a particular range selected, display on a **scatter plot the successes and failures for each booster version**

GitHub URL of completed Plotly Dash lab:

https://github.com/tudorbozan/IBM-Data-Science-Specialization-Capstone-Project/blob/main/Lab7-spacex-dash-app.py

Predictive Analysis (Classification)

After standardizing the data by scaling and splitting the data into train and test sets, several models have been built and their performance was assessed by accuracy and confusion matrices.

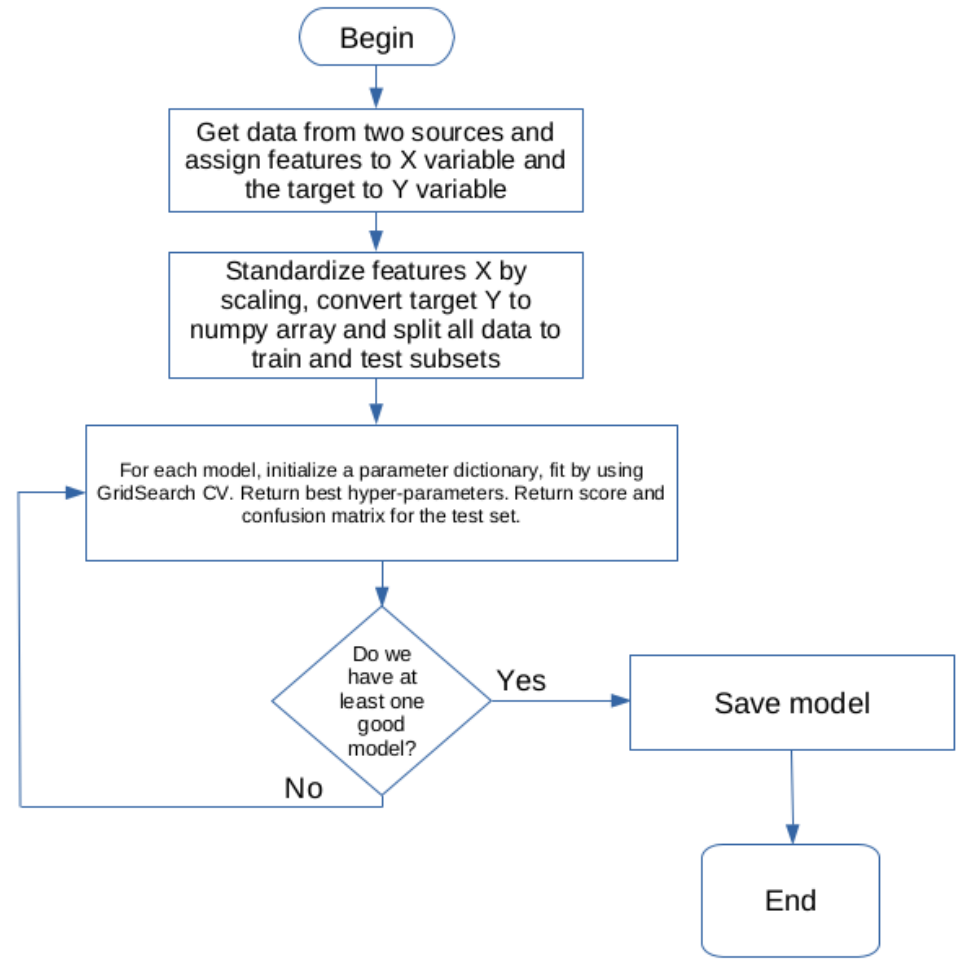
For each model, the hyper-parameters have been chosen by using a Cross Validation technique (GridSearch CV)

The following model types have been built:

- Logistic Regression
- Support Vector Machine
- Decision Tree
- K-nearest neighbors

GitHub URL of completed predictive analysis lab:

https://github.com/tudorbozan/IBM-Data-Science-Specialization-Capstone-Project/blob/main/Lab8-SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb



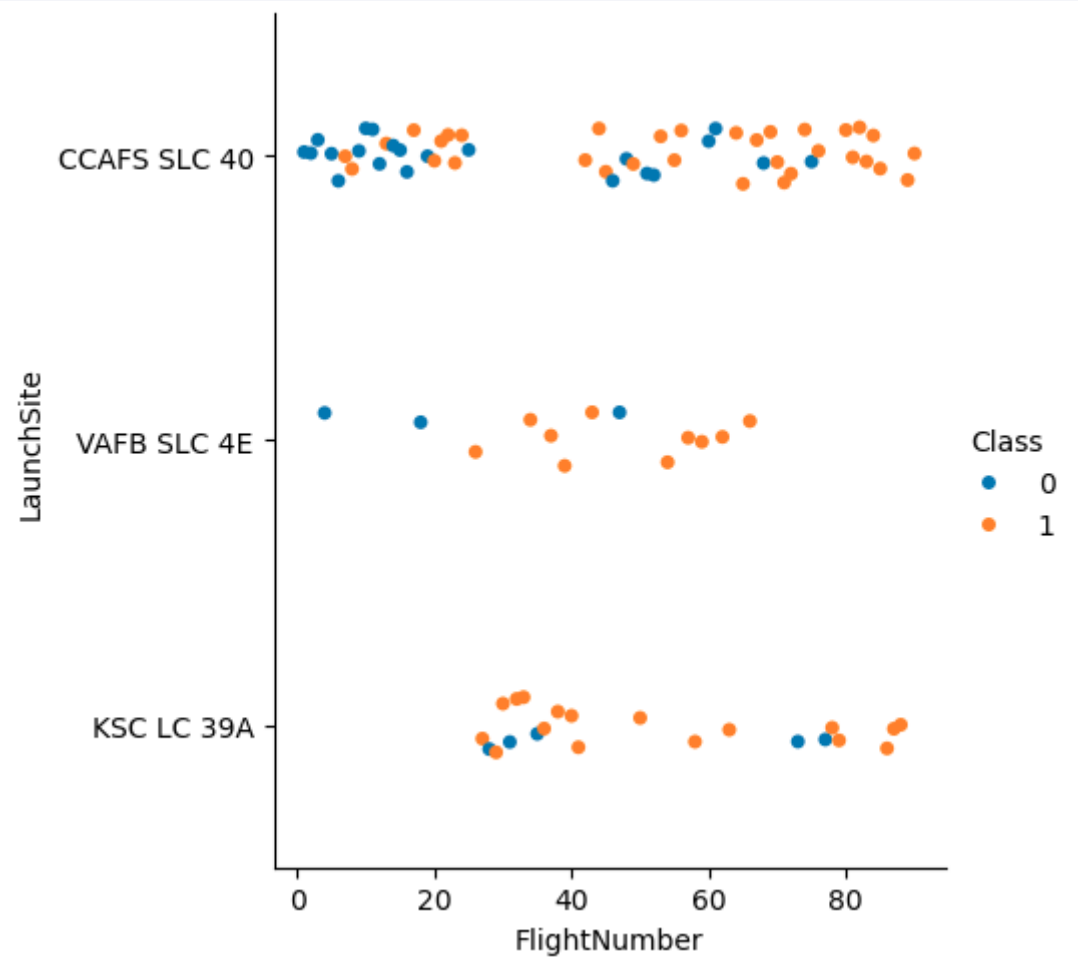
Results

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

Section 2

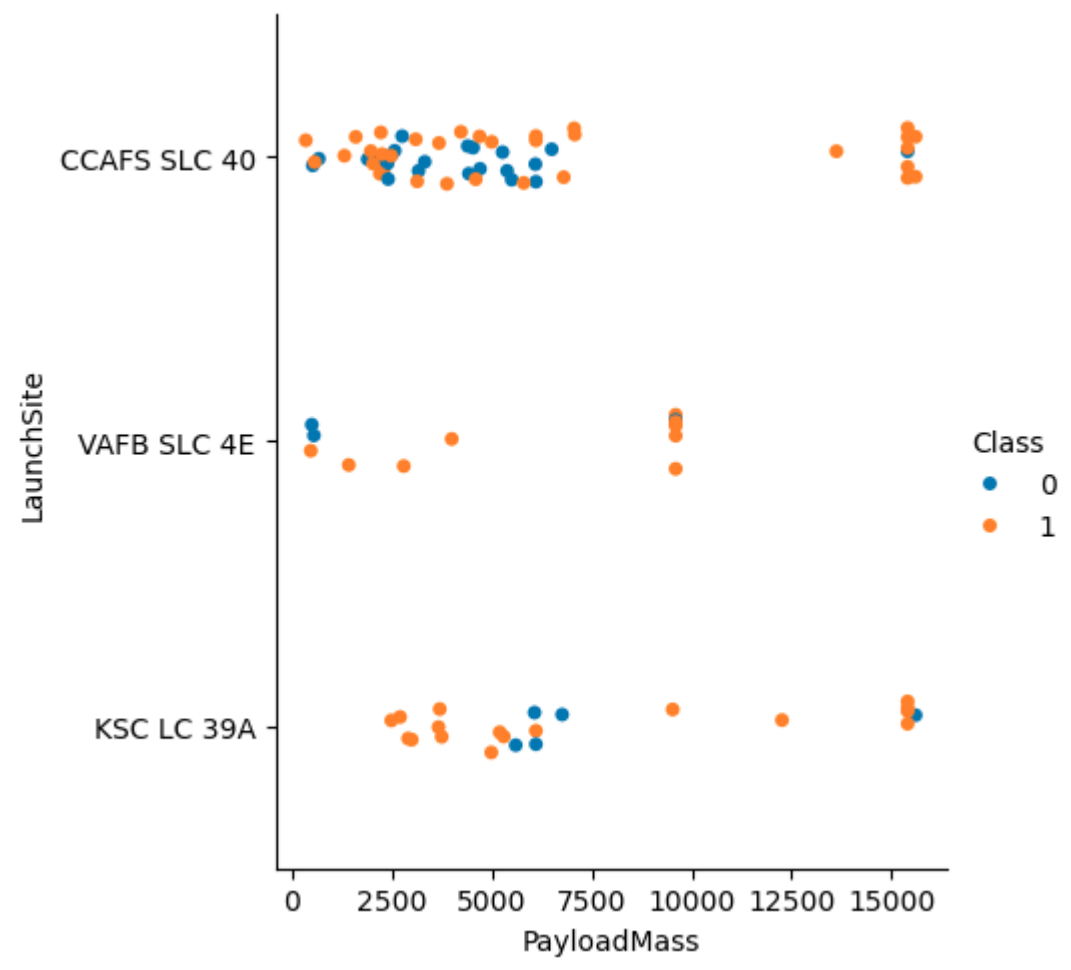
Insights drawn from EDA

Flight Number vs. Launch Site



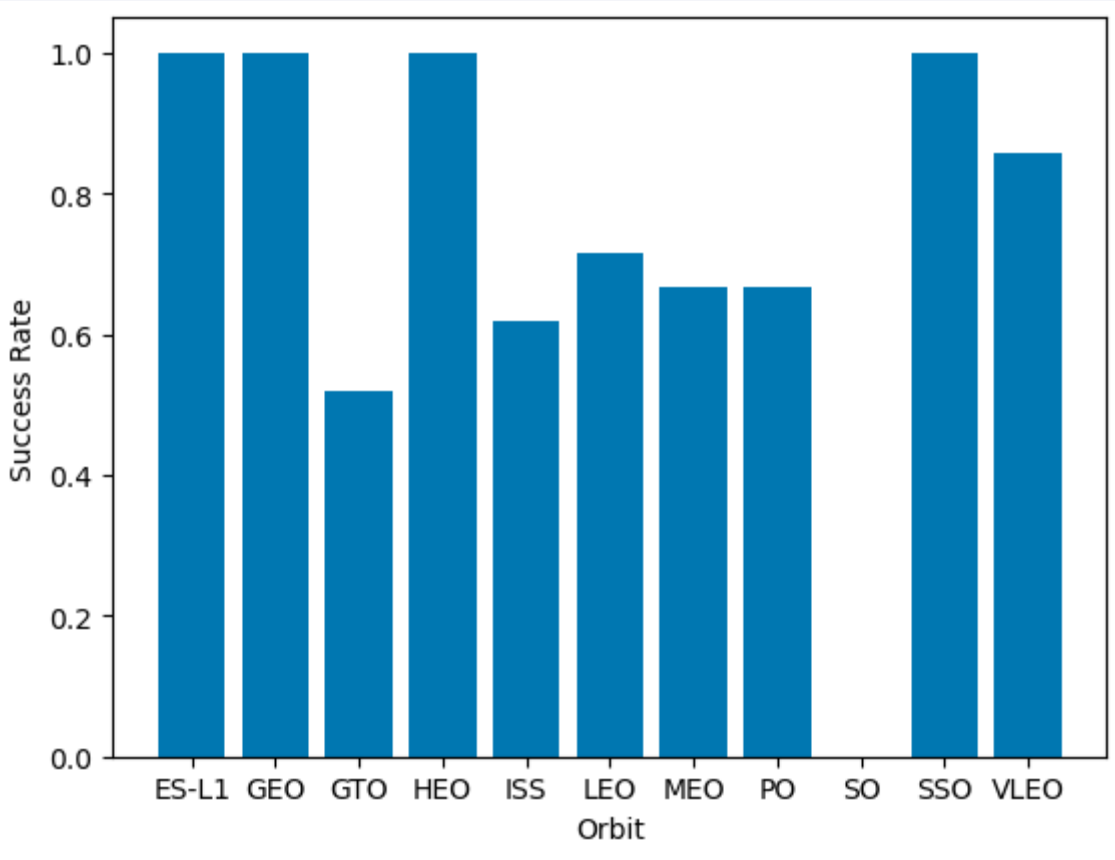
→ In general, when we look at beginning flights, the failure rate is high, then it seems to be decreasing.

Payload vs. Launch Site



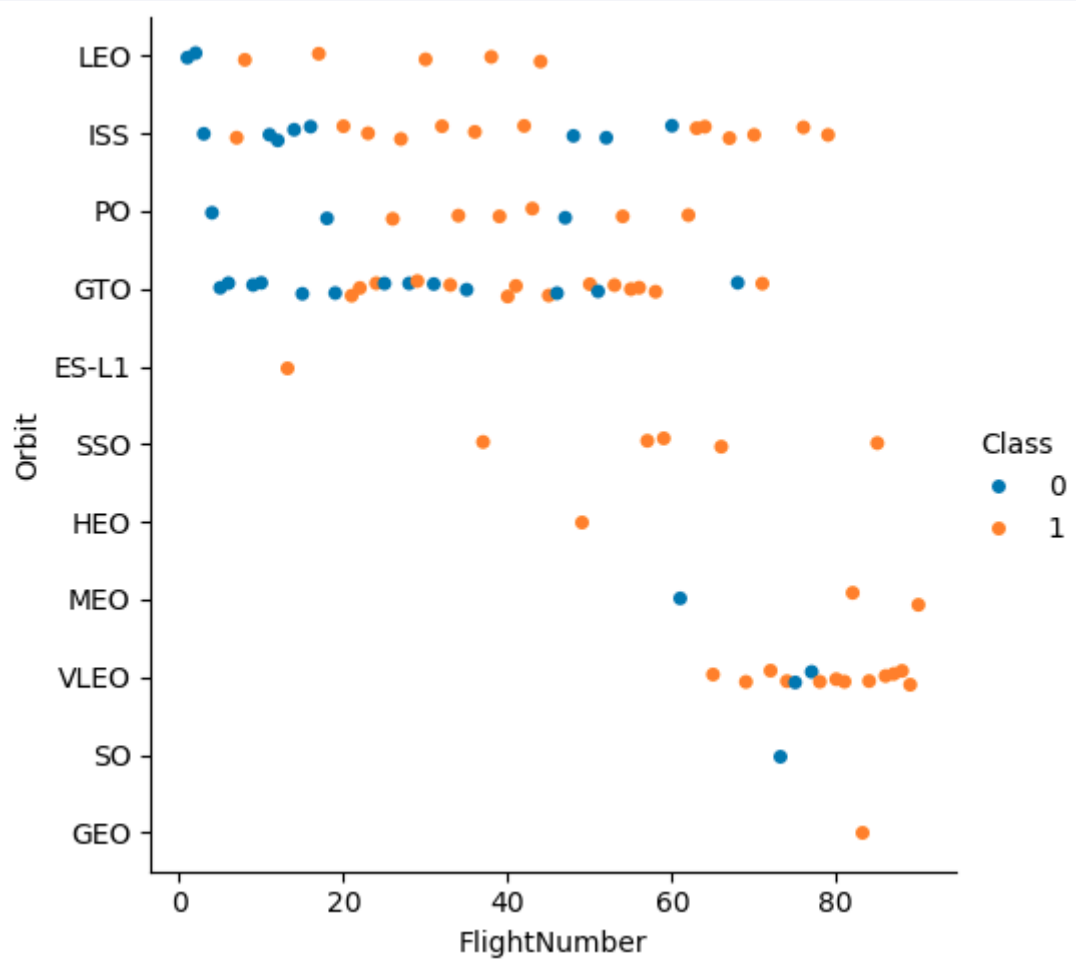
→ This tells us that for some launch sites there are no heavy masses used (as example for VAFB SLC 4E, no mass over 10k kg)

Success Rate vs. Orbit Type



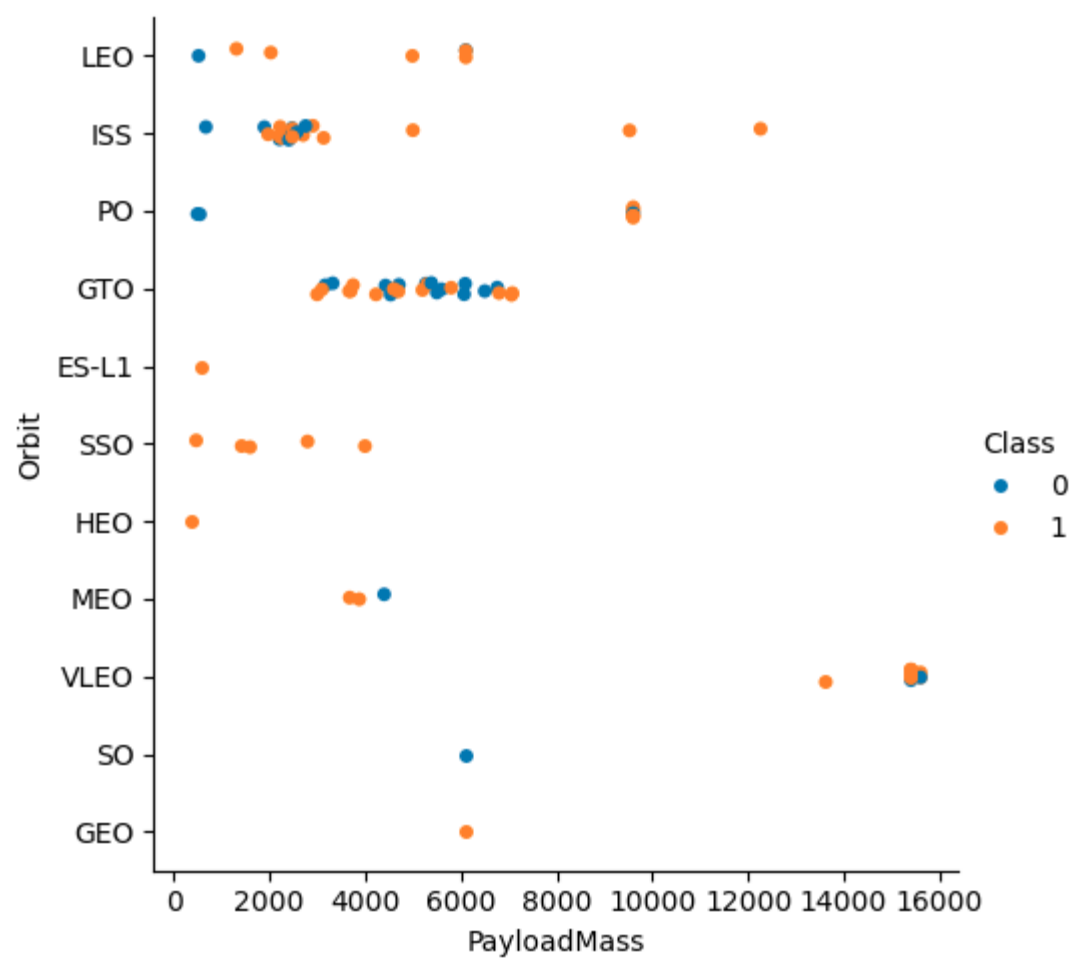
→ES-L1, GEO, HEO, SSO and VLEO orbits score highest

Flight Number vs. Orbit Type



→In the LEO orbit, success seems to be related to the number of flights. Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.

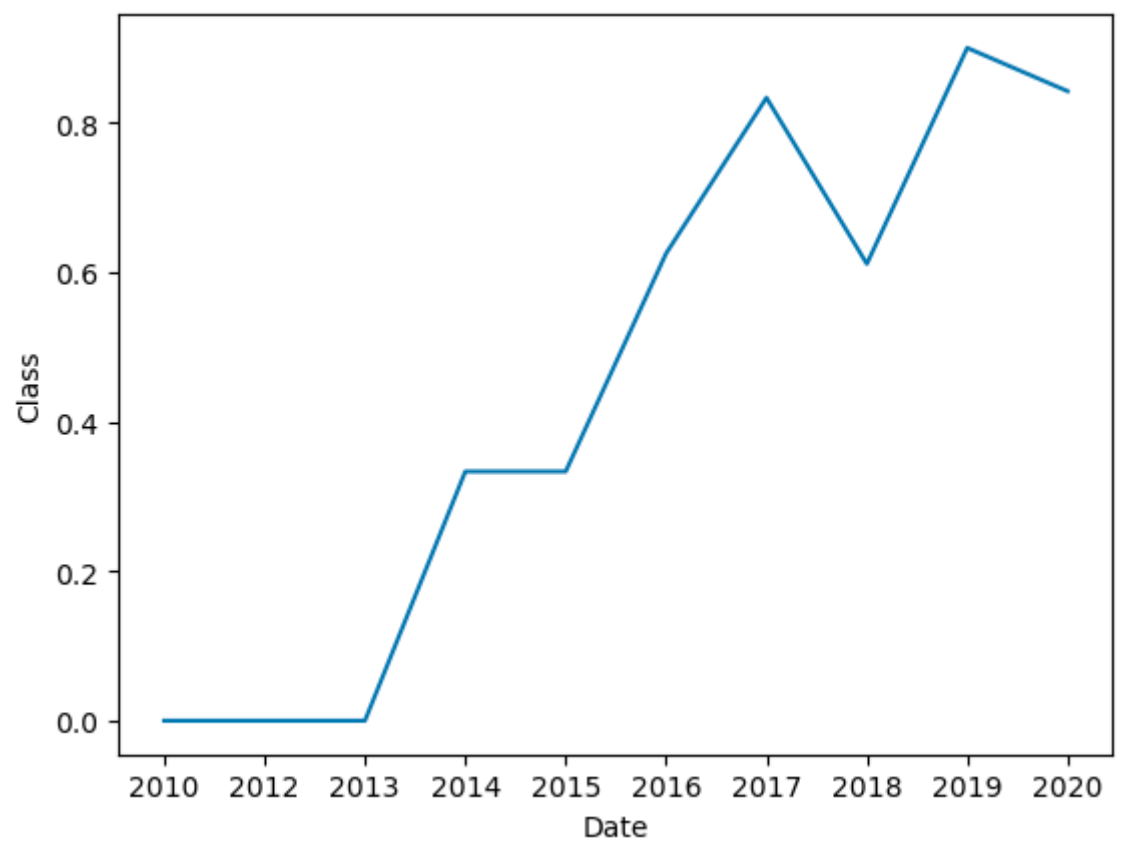
Payload vs. Orbit Type



→With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.

Launch Success Yearly Trend



→The sucess rate since 2013 kept increasing until 2020, with a slight decrease in 2018

All Launch Site Names

All distinct Launch Site names were queried from ‘SPACEXTABLE’ table, and the result is as follows:

Launch Sites:
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Launch Site Names Begin with 'CCA'

5 records where launch sites begin with `CCA` were queried from ‘SPACEXTABLE’ table:

Launch Site:
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

The total payload mass carried by boosters from NASA (customer is NASA CRS) from ‘SPACEXTABLE’ table:

SUM("PAYLOAD_MASS__KG_")	customer
45596	NASA (CRS)

Average Payload Mass by F9 v1.1

The average payload mass carried by booster version F9 v1.1 (kg) from ‘SPACEXTABLE’ table:

avg("PAYLOAD_MASS__KG_")	Booster Version
2928.4	F9 v1.1

First Successful Ground Landing Date

The date of the first successful landing outcome on ground pad found by querying the ‘SPACEXTABLE’ table:

min("Date")	lo
2015-12-22	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 kg from ‘SPACEXTABLE’ table:

Booster_Version	Payload Mass	Mission Outcome
F9 v1.1	4535	Success
F9 v1.1 B1011	4428	Success
F9 v1.1 B1014	4159	Success
F9 v1.1 B1016	4707	Success
F9 FT B1020	5271	Success
F9 FT B1022	4696	Success
F9 FT B1026	4600	Success
F9 FT B1030	5600	Success
F9 FT B1021.2	5300	Success
F9 FT B1032.1	5300	Success
F9 B4 B1040.1	4990	Success
F9 FT B1031.2	5200	Success
F9 B4 B1043.1	5000	Success (payload status unclear)
F9 FT B1032.2	4230	Success
F9 B4 B1040.2	5384	Success
F9 B5 B1046.2	5800	Success
F9 B5 B1047.2	5300	Success
F9 B5 B1046.3	4000	Success
F9 B5B1054	4400	Success
F9 B5 B1048.3	4850	Success
F9 B5 B1051.2	4200	Success
F9 B5B1060.1	4311	Success
F9 B5 B1058.2	5500	Success
F9 B5B1062.1	4311	Success

Total Number of Successful and Failure Mission Outcomes

The total number of successful and failure mission outcomes have been calculated:

match_status	count
Failure	1
Success	100

Boosters Carried Maximum Payload

The names of the booster which have carried the maximum payload mass are:

booster_version	payload_mass
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

The failed landing_outcomes in drone ship, their booster versions, and launch site names for year 2015 are computed:

month	land_out	boost_ver	launch_site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

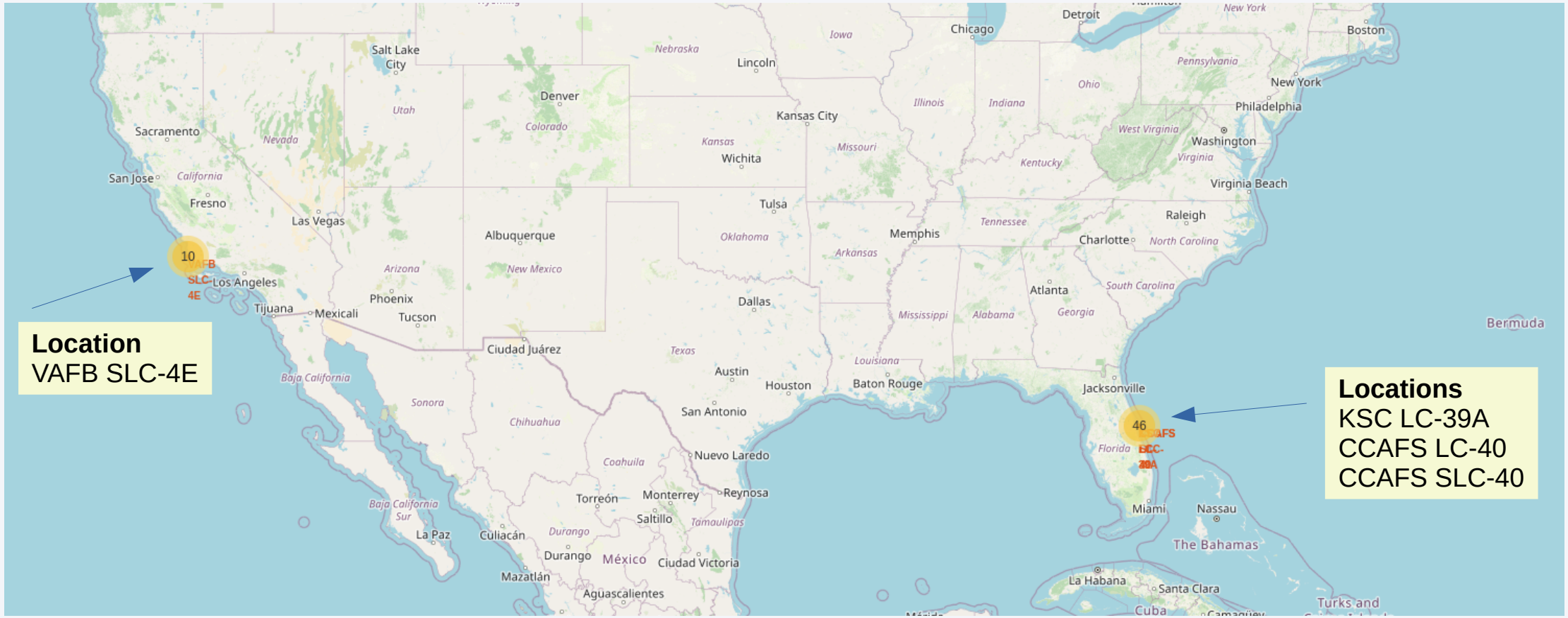
Below is the count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order:

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

Section 3

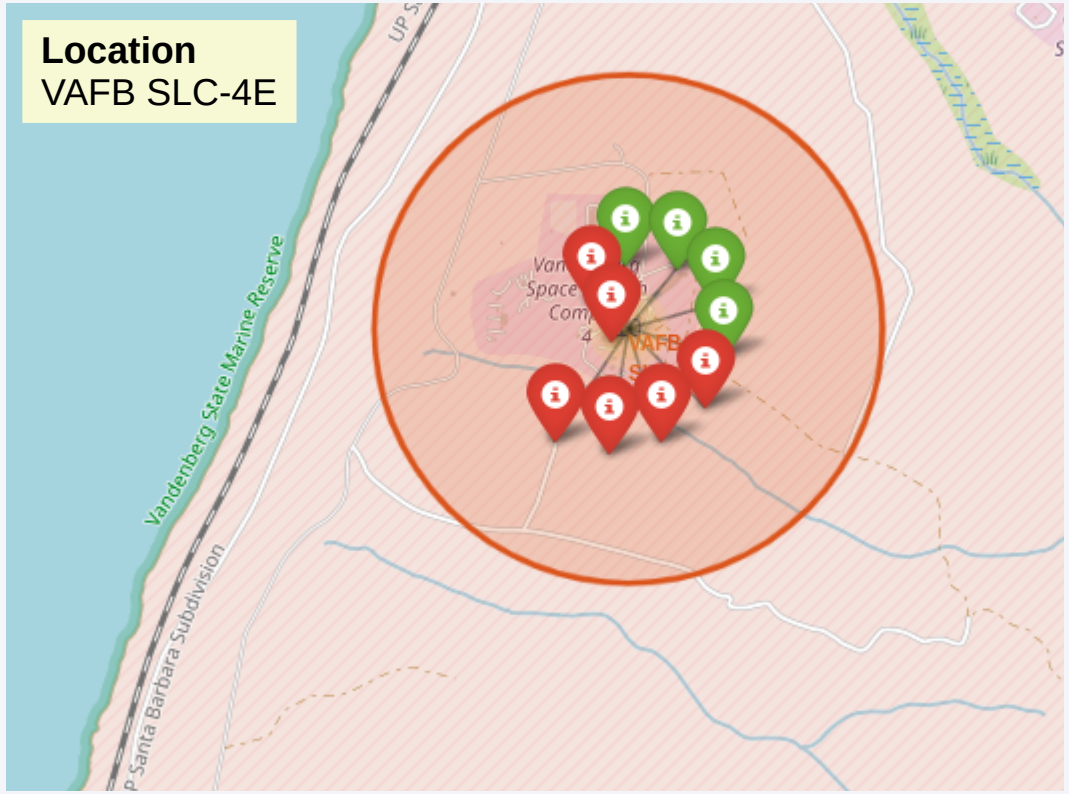
Launch Sites Proximities Analysis

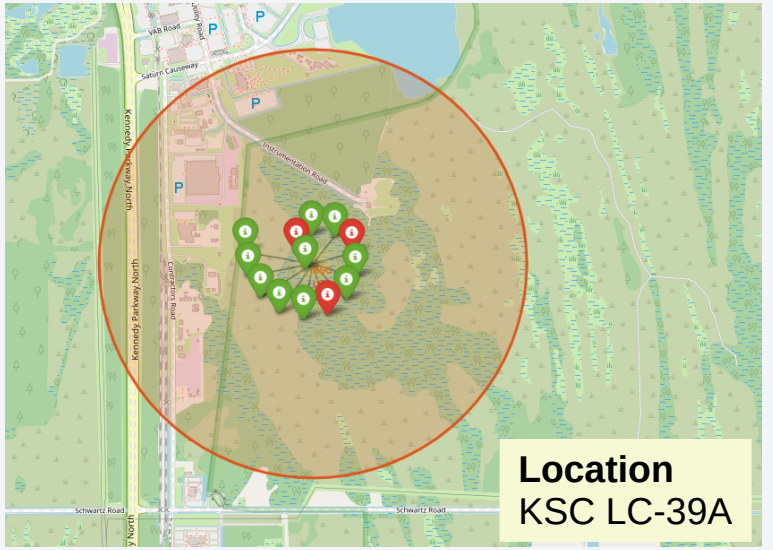
Launch sites mapped: one west, three east



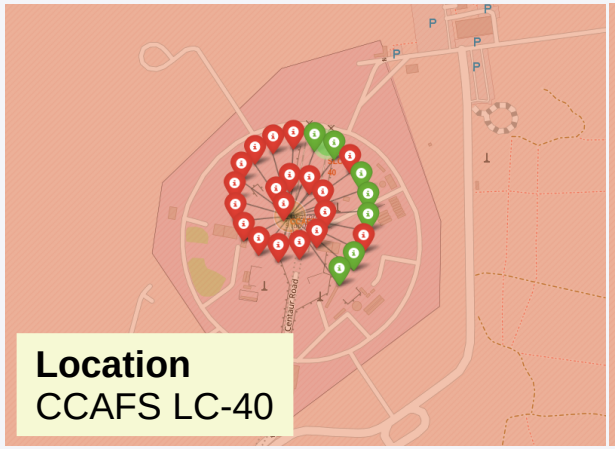
Success is green

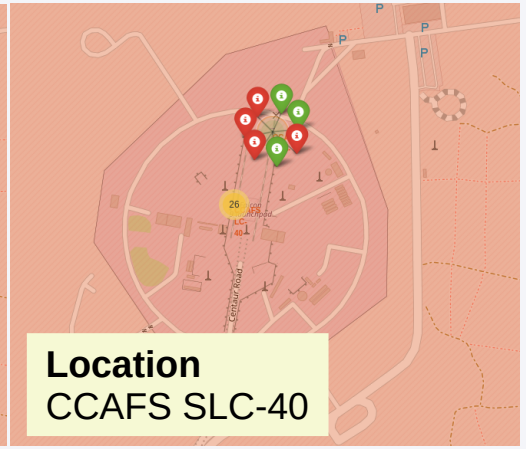
Green → successful oucome
Red → failed outcome



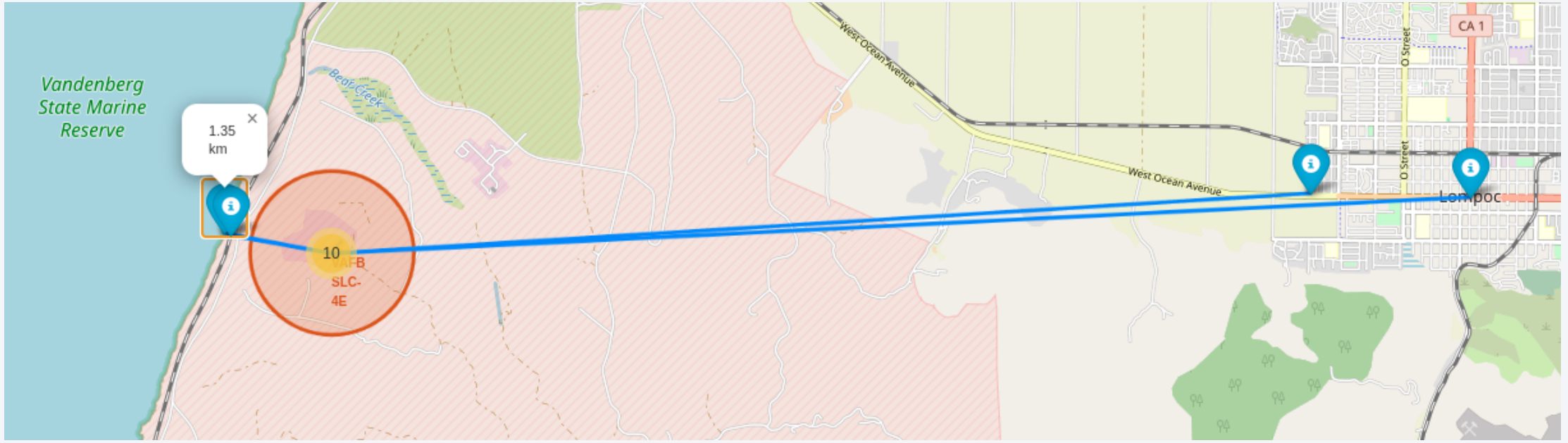


← best success rate location

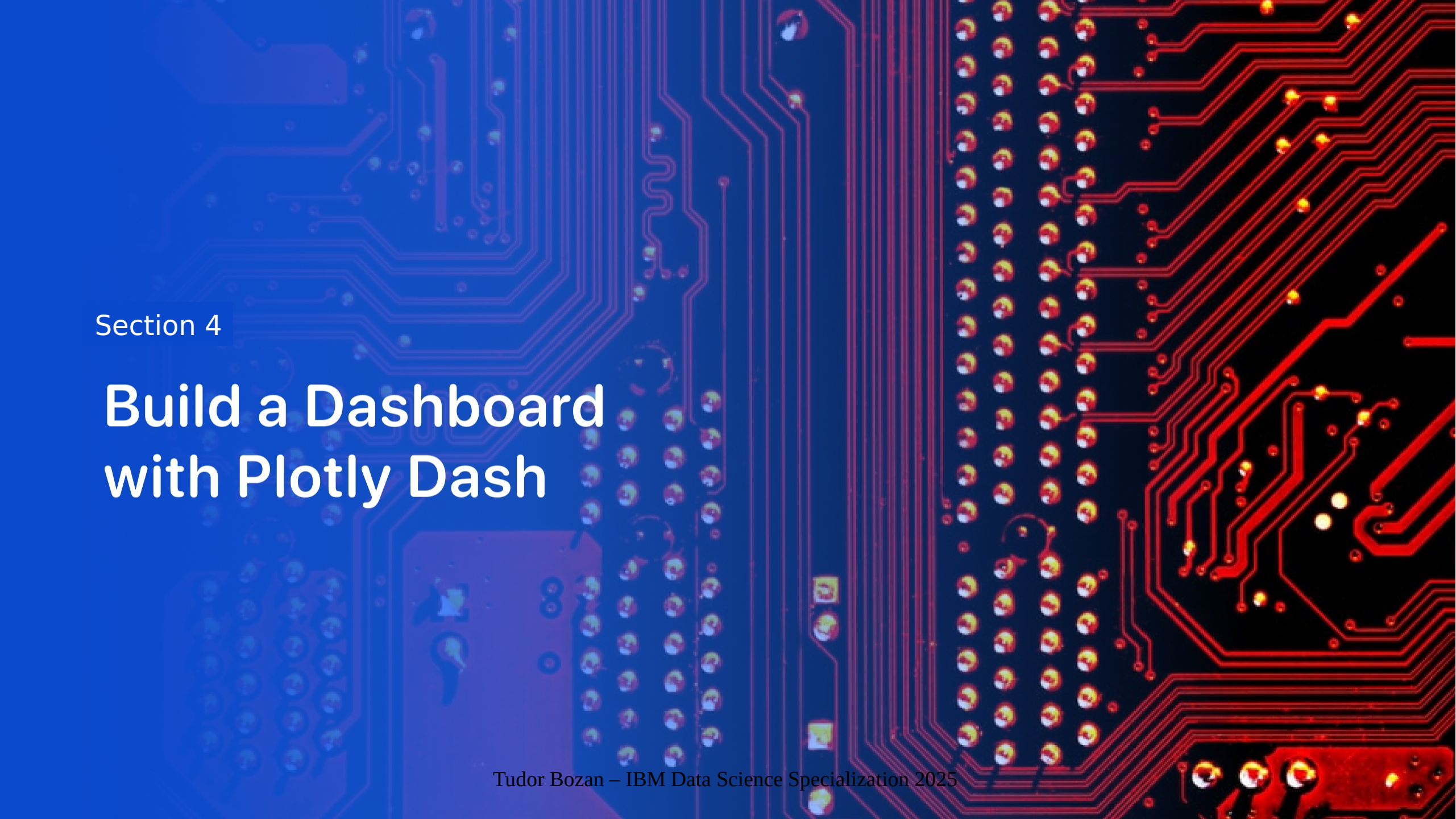




VAFB SLC-4E proximity to nearest POI



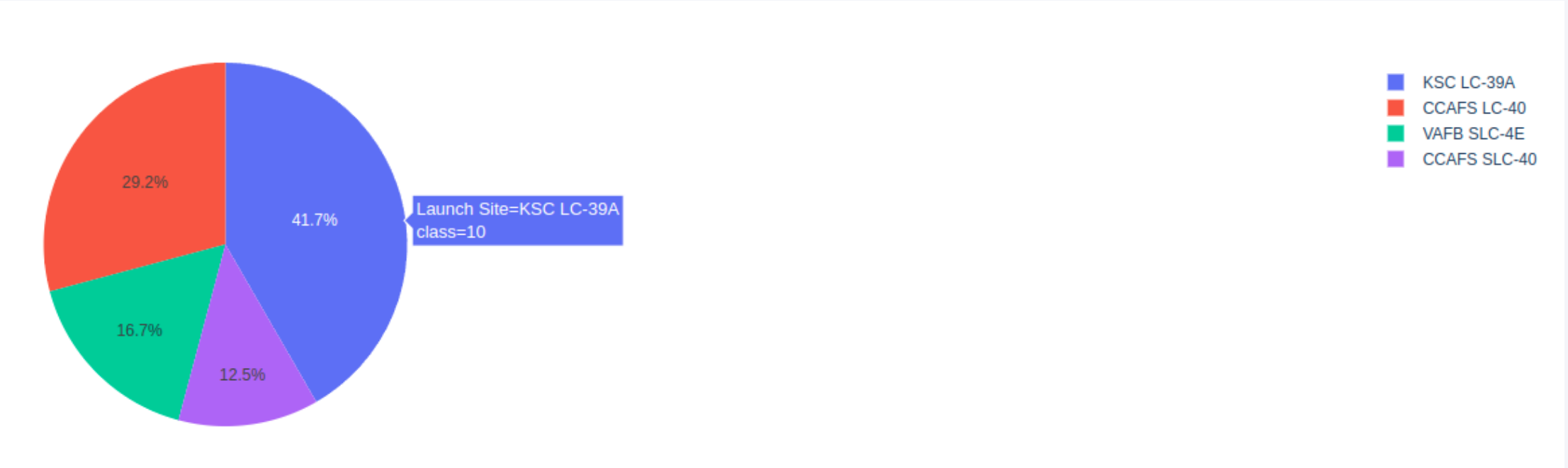
Nearest POI for the location **VAFB SLC-4E** could be determined interactively on the map, such as:
- Coastline: 1.35 km
- Railway: 1.27 km
- Town: 12.04
- Highway: 14.01 km



Section 4

Build a Dashboard with Plotly Dash

Success count per site



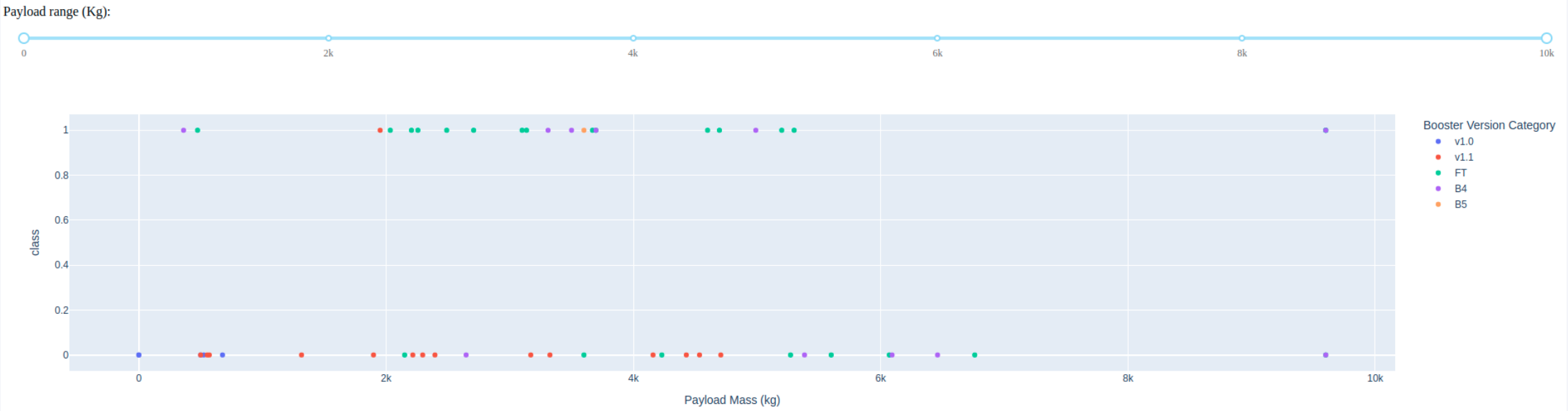
The count of successes for each site is displayed in an interactive pie chart that the user can click on.

Site ‘KSC LC-39A’ has the highest success from all of them combined.

Launch site KSC LC-39A – highest success rate

The success rate for this site is 76.9 %, the highest from all analyzed sites.

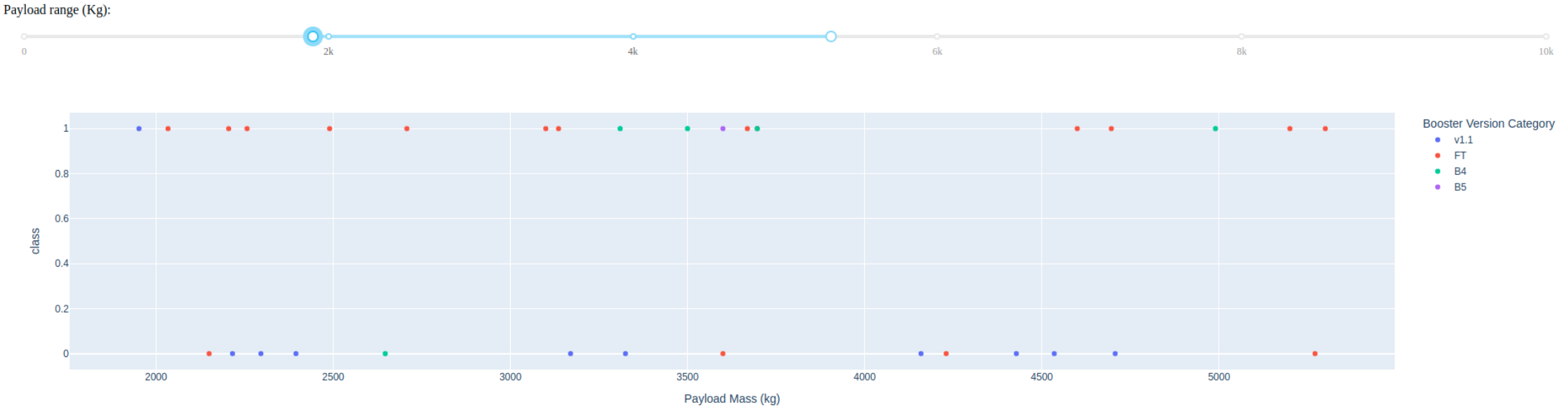
Launch Outcome vs Payload Mass (kg) – all sites



Looking at the full range (0 to 10k kg) we see the most successes are between ~2k and ~5k kg, and that most of the times it seems that we have failures.

The most successful Booster Version seem to be FT, and the most unsuccessful, v1.1

Launch Outcome vs Payload Mass (kg) – all sites

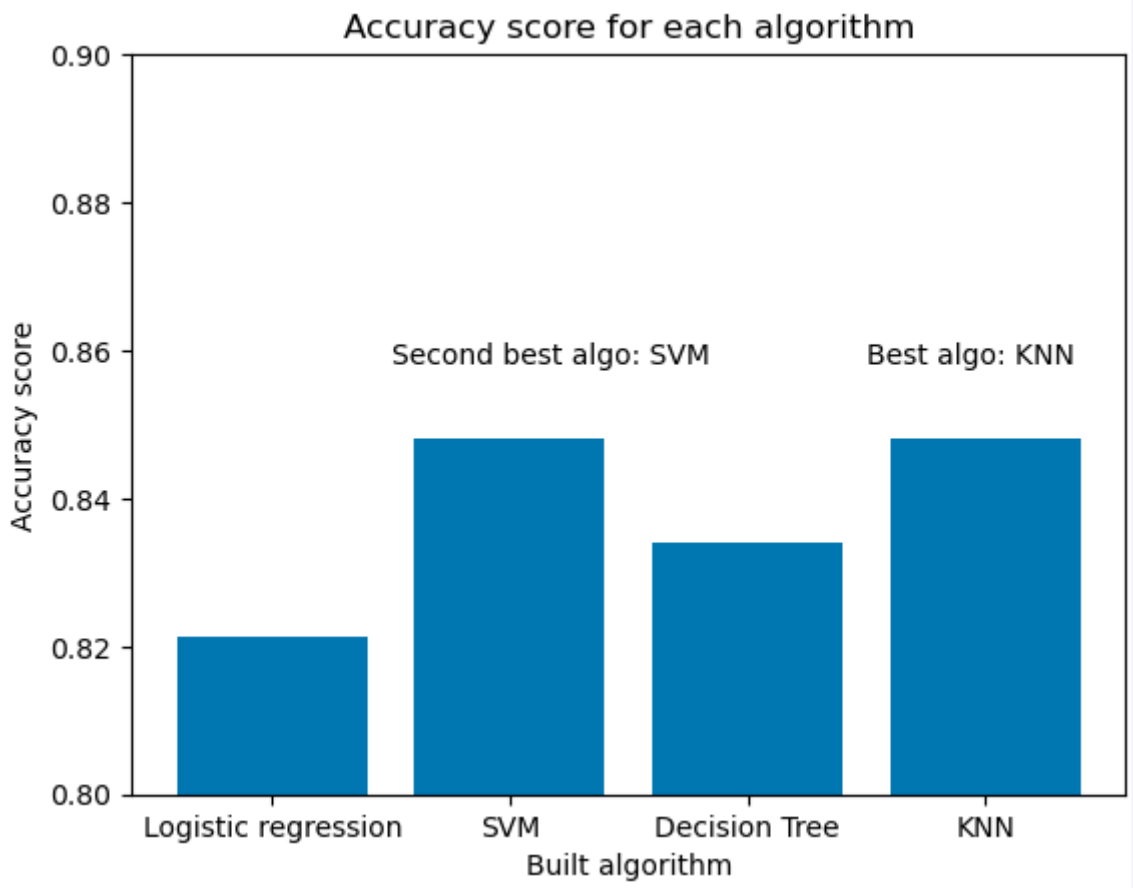


Zooming in at just before 2k and right after 5k kg, where most successes are, we see that FT still holds the most successes. Booster Version B4 also looks good in this range, whereas v1.1 and B5 not quite.

Section 5

Predictive Analysis (Classification)

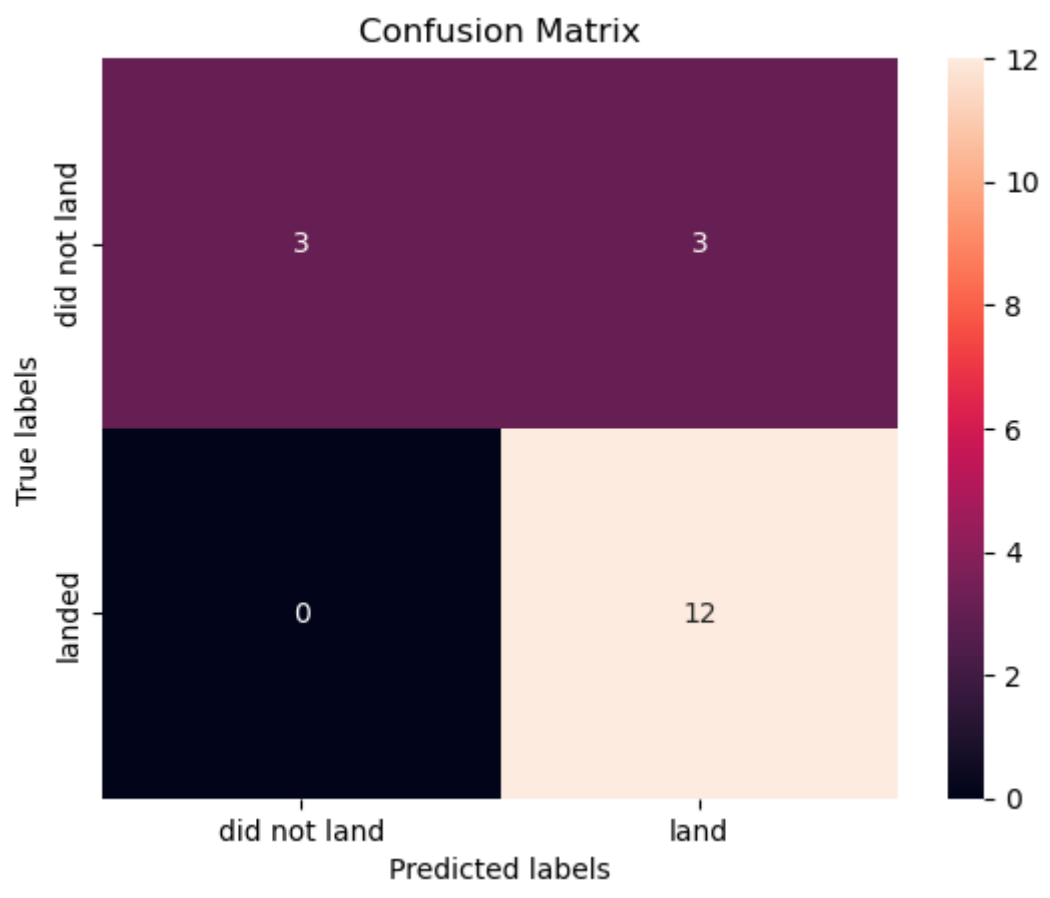
Classification Accuracy



KNN performed best on training data, closely followed by SVM, both at about 84 % accuracy.

All algorithms have about 83% accuracy on test data.

Confusion Matrix



On given test data, the best algorithm, KNN, had a good confusion matrix:

→ only 3 false positives → to be improved further (train and test on larger data sets or make use of hyper-parameter tuning further)
→ no false negatives

Conclusions

→ Success rate had increased over time at SpaceX, in general

→ The most successful sites have been identified, as well as other features

→ Using given features for all launch sites, better or worse, the built models could correctly predict a landing success with an accuracy of over 83-84%

→ There is a variety of four models to choose from, to be used depending on platform

Appendix

Tuned (best) hyperparameters and accuracy output for each trained model:

1. Logistic Regression

```
tuned hpyerparameters :(best parameters)  {'C': 0.1, 'penalty': 'l2', 'solver': 'lbfgs'}
accuracy : 0.8214285714285714
```

2. Support Vector Machine

```
tuned hpyerparameters :(best parameters)  {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
accuracy : 0.8482142857142856
```

3. Decision Tree

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 5, 'splitter': 'random'}
accuracy : 0.8339285714285714
```

4. K – Nearest Neighbors

```
tuned hpyerparameters :(best parameters)  {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
accuracy : 0.8482142857142858
```

Thank you!