IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Alvaro Mejia Garcia
29/02/2024

Outline

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

Executive Summary

- The fact that SpaceX can recover the first stage of a launched rocket, allows the company to save millions of dollars in every mission. However, not always is possible to recover the stage.
- The aim here is to study the influence of different parameters on the landing outcome and **build a Machine Learning (ML) model to predict if the rocket will land successfully or not**.
- Methodologies:
  - SpaceX API and web scraping for data collection, followed by standard techniques of data cleaning
  - Exploratory Data Analysis (EDA) with visualization and SQL queries
  - Interactive visualizations of the launch sites and interactive dashboard
  - ML model building, looking for the best parameters and evaluation accuracy
- Summary of all results:
  - Several parameters, such as the launch location, payload and type of orbit have a correlation with the landing outcome
  - Classification models can be built with remarkable accuracy, but with room for improvement

Introduction

- Project background and context:
  - When a rocket is launched to the space, it's bound to put a satellite into an orbit. This is what is called the "Payload".
  - The other parts of the rocket (also called "stages") are just enormous fuel tanks to reach the right altitude for the payload.
  - Traditionally, the first stages of the rockets fall to the sea when they're empty and are never seen again. However, SpaceX managed to recover the first stage back and reuse it.
  - Thanks to that, the company only spends around 65M$ in a launch, much less than other competitors who spend up to 165M$ per launch.
  - The problem is that not always is possible to recover the 1st stage, due to contingencies during the mission or failures in the landing operation
- What do we want to know?
  - **Is there any chance to predict whether a landing will be successful or not?**
  - **What parameters have influence on a successful or unsuccessful landing?**

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Request data through the SpaceX API
  - Web scraping for historical Falcon 9 launch records
- Perform data wrangling
  - Filter the data to include only Falcon 9 launches
  - Deal with missing values: replace Payload missing values with the mean
- Perform exploratory data analysis (EDA) using visualization and SQL
  - Different plots and queries to get a better understanding of the dataset

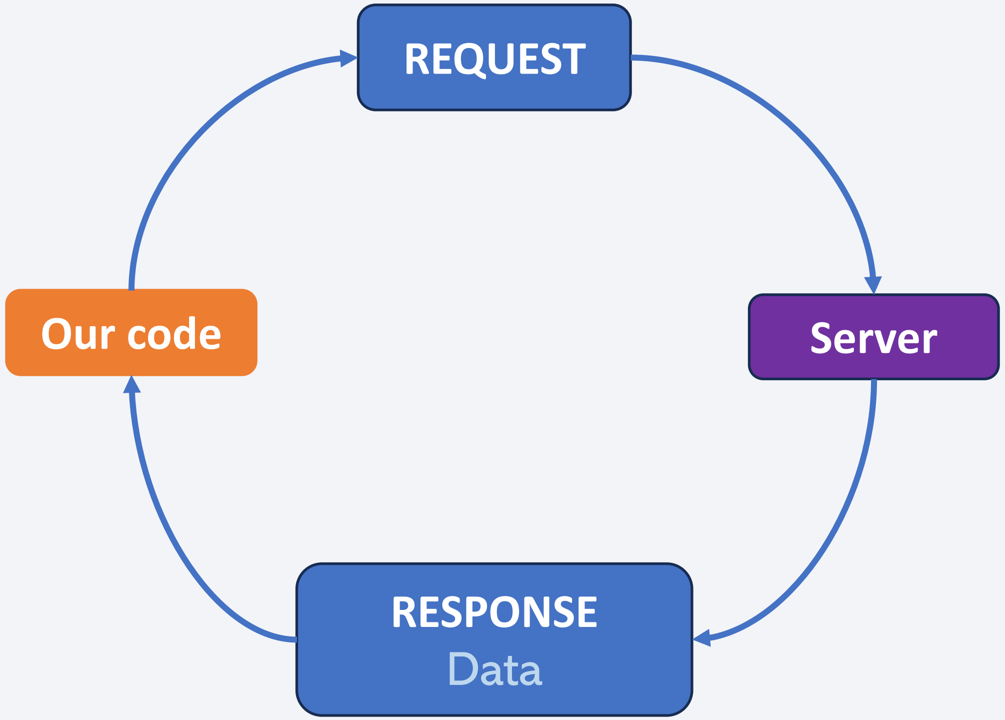
Methodology

Executive Summary

- Perform interactive visual analytics using Folium and Plotly Dash:
    - Create an interactive map with markers for each launch location and distance measurement
    - Create an interactive dashboard to see the influence of the launch location, the payload and the booster version on the landing outcome
- Perform predictive analysis using classification models
    - Optimize with the best hyperparameters
    - Confusion matrix and accuracy measurement

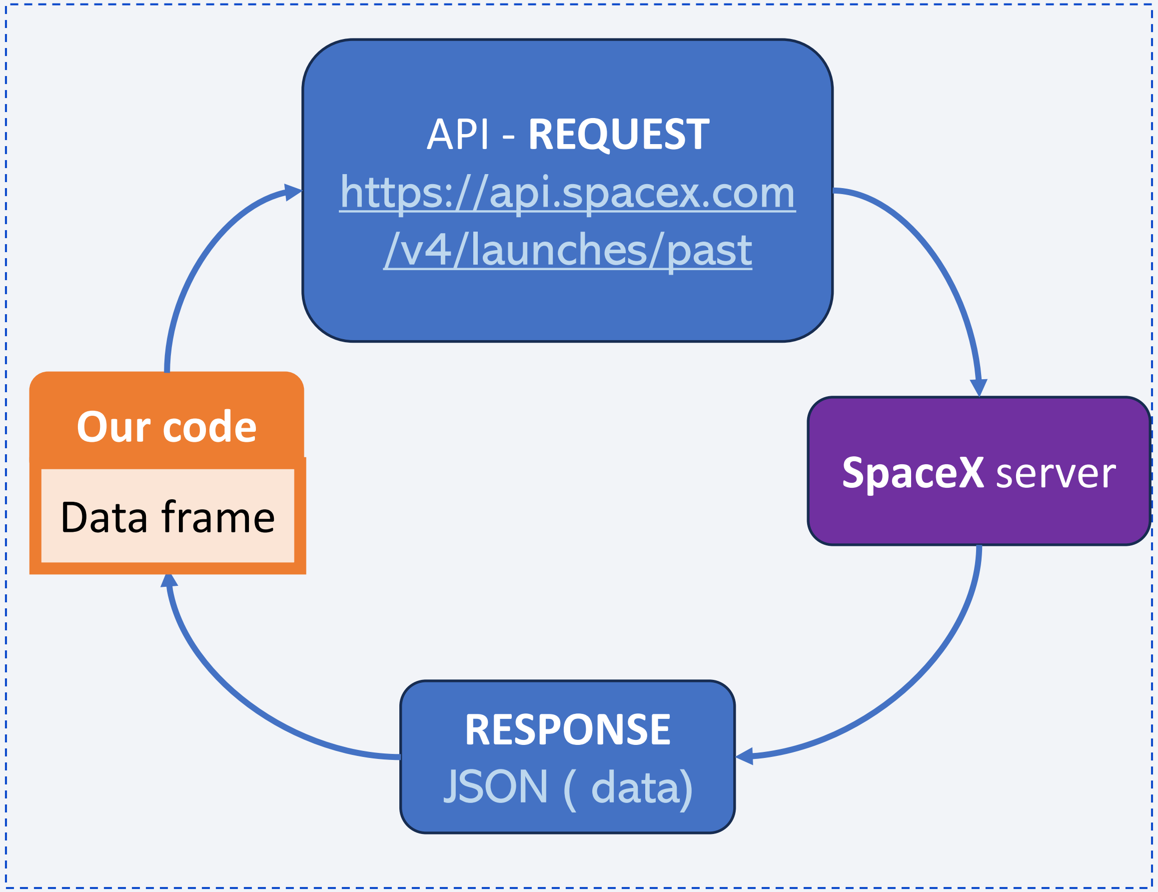
Data Collection

- Data were collected in 2 ways:
  - Through the SpaceX API: https://api.spacex.com/v4/launches/past
  - Using web scraping on Wikipedia: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- These methods consist of a piece of **code** sending a **request to the server** where the data is stored.
- Then, the server sends a **response** back with the **data**



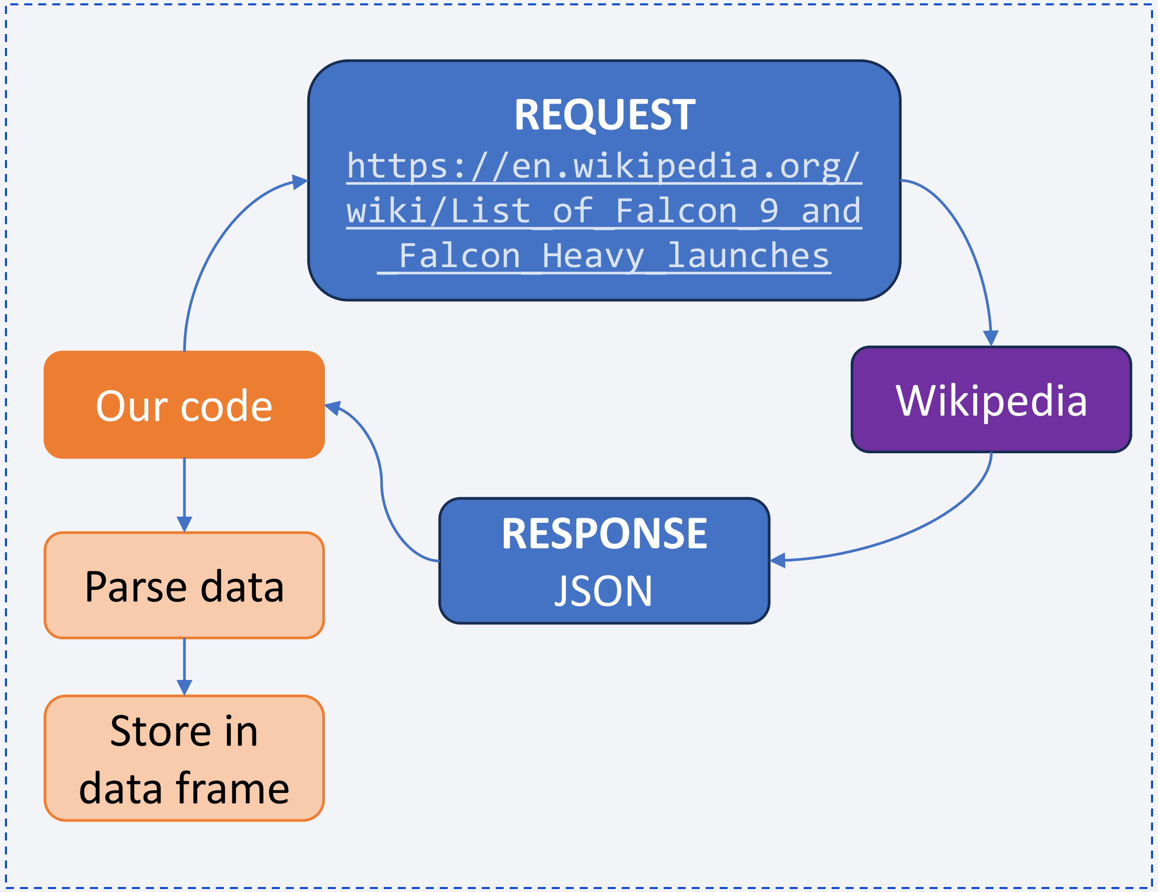
Data Collection – SpaceX API

- Use of SpaceX REST API for data collection.
- Connect to SpaceX server: https://api.spacex.com/v4/
  - Endpoint for history data: launches/past
- Get response (data) in JSON format
- Store data in pandas dataframe
- GitHub URL: https://github.com/umbreon13/Capstone_Applied_Data_Science/blob/main/1-data-collection-api.ipynb



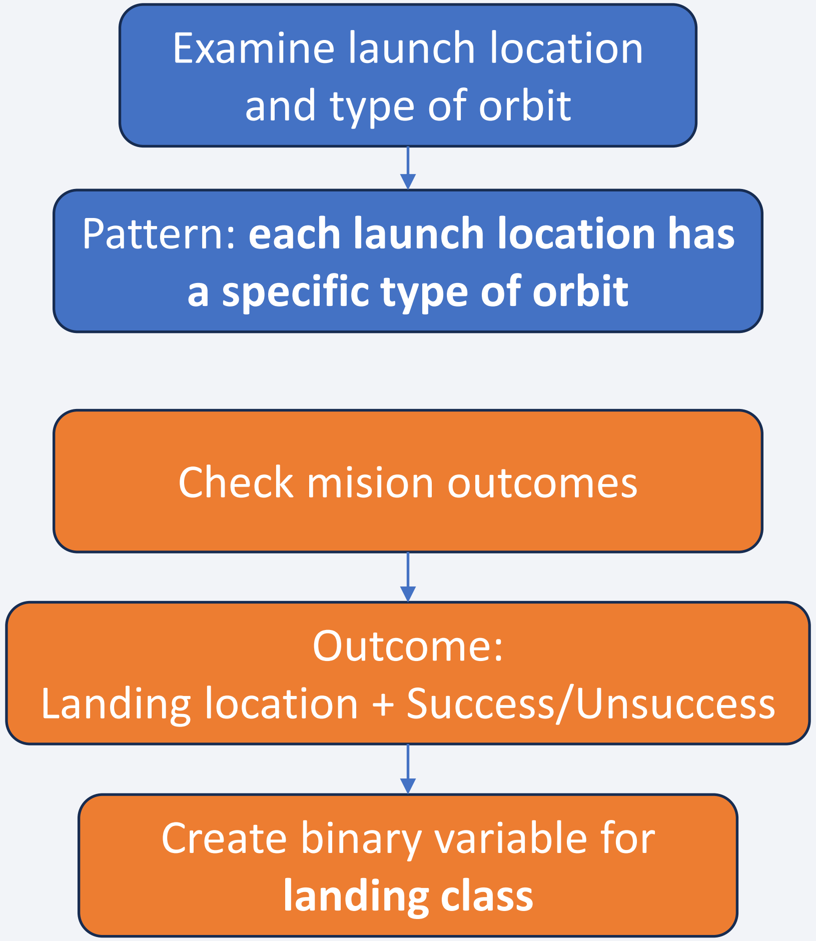
Data Collection - Scraping

- Connect to Wikipedia via **requests**
- Get the **response** from the **server**
- **Identify** the table with the data of interest
- **Parse** the table content
- Store it into a pandas data frame
- GitHub URL: https://github.com/umbreon13/Capstone_Applied_Data_Science/blob/main/2-webscraping.ipynb



Data Wrangling

- Preliminary Exploratory Data Analysis (EDA) is performed to:
    - Find patterns
    - Determine training labels
- Check launch places, **each for a dedicated orbit**.
- Check the mission outcome: successful/unsucc. landing + landing place
- Create **binary landing outcome label**: landing_class: [0: unsuccessful, 1: successful]
- GitHub URL: https://github.com/umbreon13/Capstone_Applied_Data_Science/blob/main/3-Data_wrangling.ipynb



EDA with Data Visualization

- Charts plotted:
  - Payload vs. Flight number
  - Launch site vs. Flight number
  - Launch site vs. Payload
  - Success rate vs. Orbit type
  - Orbit type vs. Flight number
  - Payload vs. Orbit type
  - Success rate vs. year
- GitHub URL: https://github.com/umbreon13/Capstone_Applied_Data_Science/blob/main/5-eda-data-visualization.ipynb

EDA with SQL

- Perform EDA with SQL for a better understanding of the SpaceX dataset:
  - Display the name of the unique launch sites
  - Display 5 records where launch site is CCAFS LC-40
  - Display the total payload carried by the NASA (CRS) launchers
  - Average payload carried by booster version F9 v1.1
  - Date of the first successful landing on ground pad
  - Names of the boosters with successful landings on drone ship and PL between 4000 and 6000kg
  - List the total number of successful and failure missions
  - Names of the boosters with maximum payload:
  - Records with the month name, booster version and launch site for the year 2015 where **landing outcome in drone ship is failure**
  - Count the landing outcomes types between 2010-06-04 and 2017-03-20
- GitHub URL: https://github.com/umbreon13/Capstone_Applied_Data_Science/blob/main/4-eda-sql.ipynb

Build an Interactive Map with Folium

- Build an interactive map to analyze each launch location and their outcomes
- Several objects were created over the map:
  - Circles: to point out the location of launch sites
  - Marker cluster to deal with multiple overlapping markers, used to point out the outcomes of the landing class in each launch location
  - Mouse position: to get the coordinates of each point the cursor is hoovering on
  - Polyline: to draw lines from launch sites to points of interest (for example, the coastline, a railway, an airport, etc.)
- GitHub URL: https://github.com/umbreon13/Capstone_Applied_Data_Science/blob/main/6-lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Interactive dashboard – plots added:
  - Pie chart – proportion of successful launches per location (including all sites together)
    - Compares the different success rates between different locations
    - Allows to see the importance of the location on the mission outcome
  - Success rate per Payload and Booster version – interactive, allows to display different ranges of PL
    - Compares the success rates between different payloads
    - Discover the range of payloads with better success rate
    - Determine if there's any influence of the booster version on the outcome
- GitHub URL: https://github.com/umbreon13/Capstone_Applied_Data_Science/blob/main/7-spacex_dashboard_app.py

Predictive Analysis (Classification)

- We must **classify** the mission outcome to know if the 1st stage will land or not
- Build the model: predictors (X) and target (Y)
    - Our target is the landing class: 0 means unsuccessful; 1 is successful
    - The predictors are the rest of variables used to predict the target → must be scaled
- Split the dataset: train set (80% of the data) and test set (20%)
- Model improvement: test different computation algorithms and hyperparameters for each model:
    - Linear Regression
    - Support Vector Machine (SVM)
    - Decision Trees
    - K-Nearest Neighbors
- Model evaluation: check the confusion matrix and the accuracy of the best performing parameters to compare models
- GitHub URL: https://github.com/umbreon13/Capstone_Applied_Data_Science/blob/main/8-SpaceX_Machine_Learning_Prediction.ipynb

Model development process

Separate **predictors** and **target**

↓

Split data set: **Train & Test**

↓

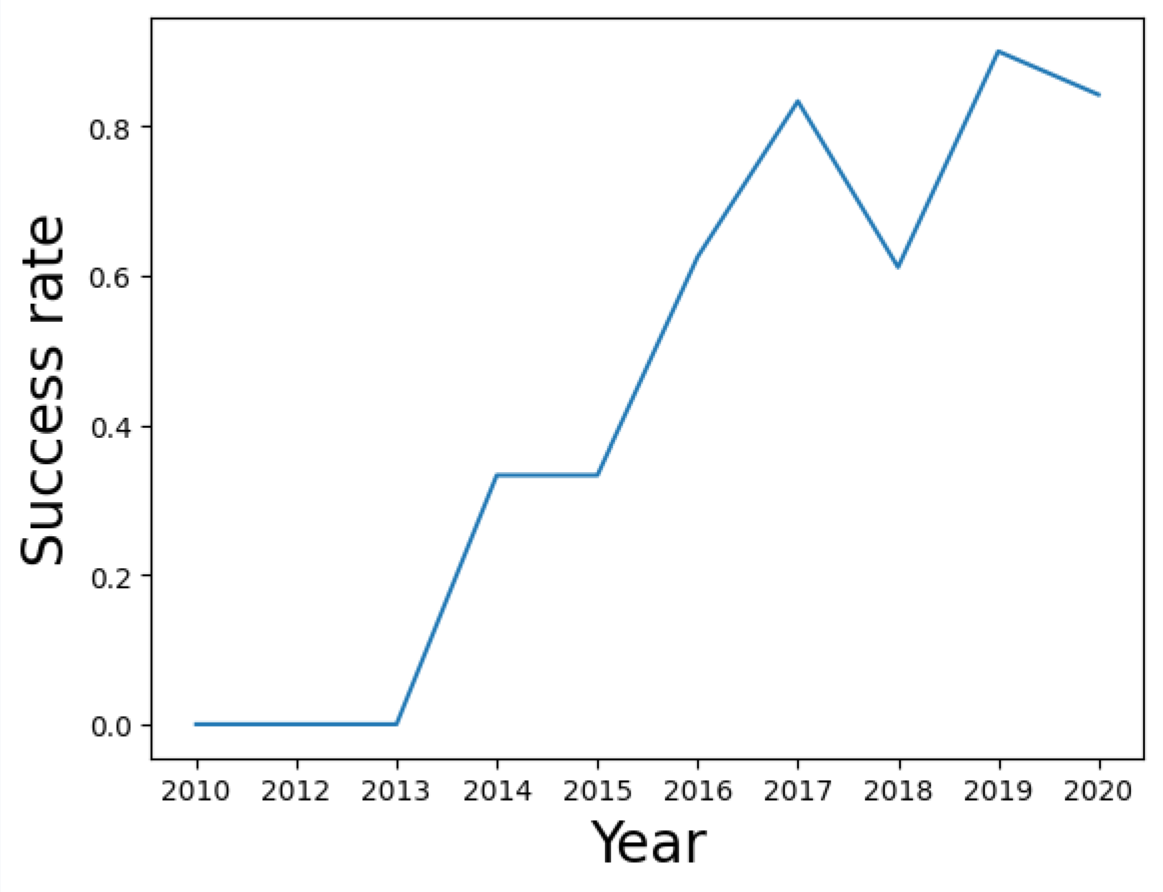
Find the best **hyperparameters** for each model

↓

Evaluation by comparing models

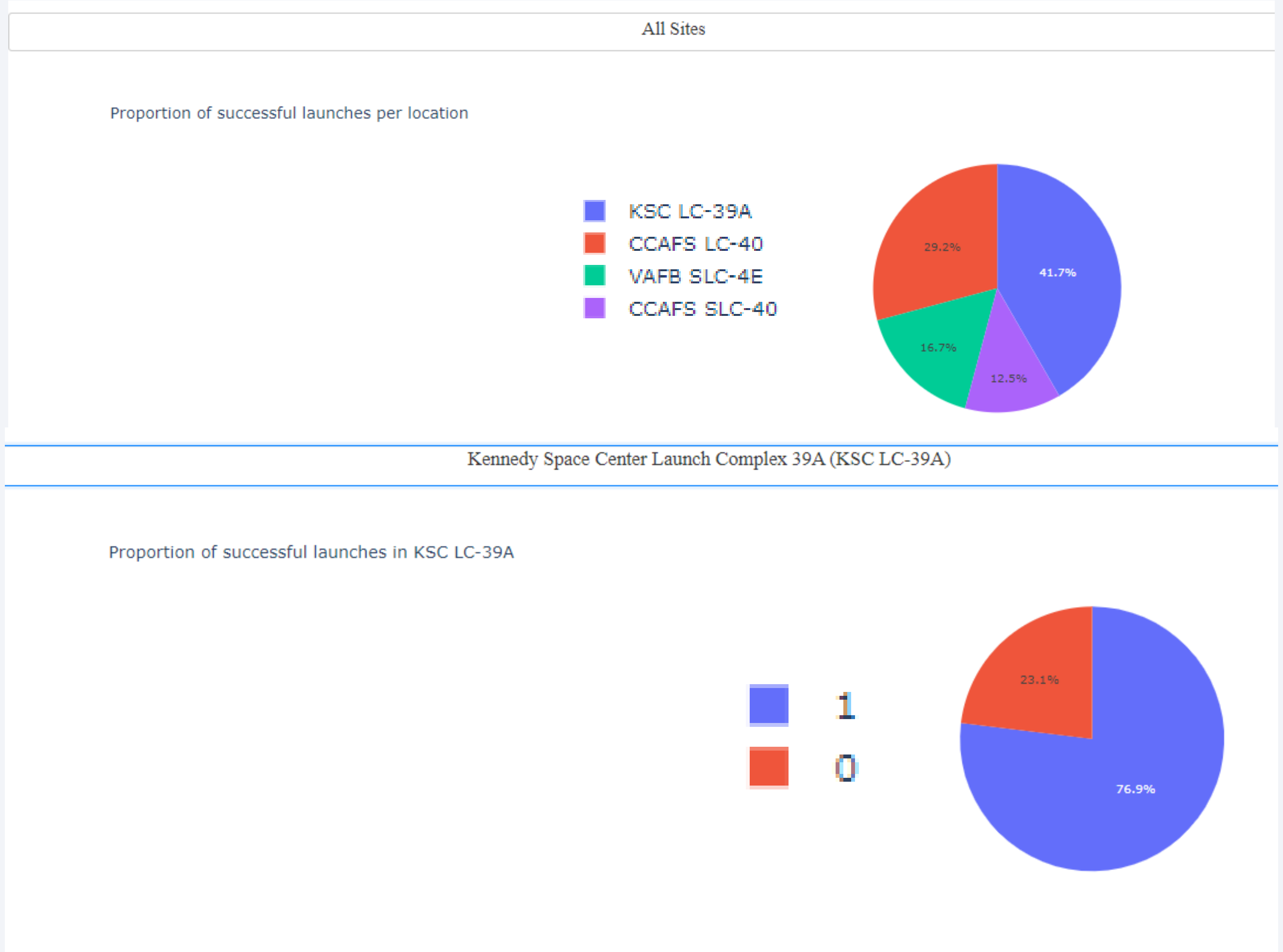
Results

- Exploratory data analysis results:
  - Success rate increases over time
  - Payload and site location have an important influence on the success rate
  - In some cases, the type of orbit can also influence the outcome
  - Each launch location is dedicated to specific types of orbit
  - VAFB SLC 4E was never used as a launch site for payloads heavier than 10000kg



Results

- Interactive analytics demo in screenshots:
  - KSC LC-39A is the launch site with the highest proportion of successful landings
  - 76.9% of its launches successfully recovered the 1st stage



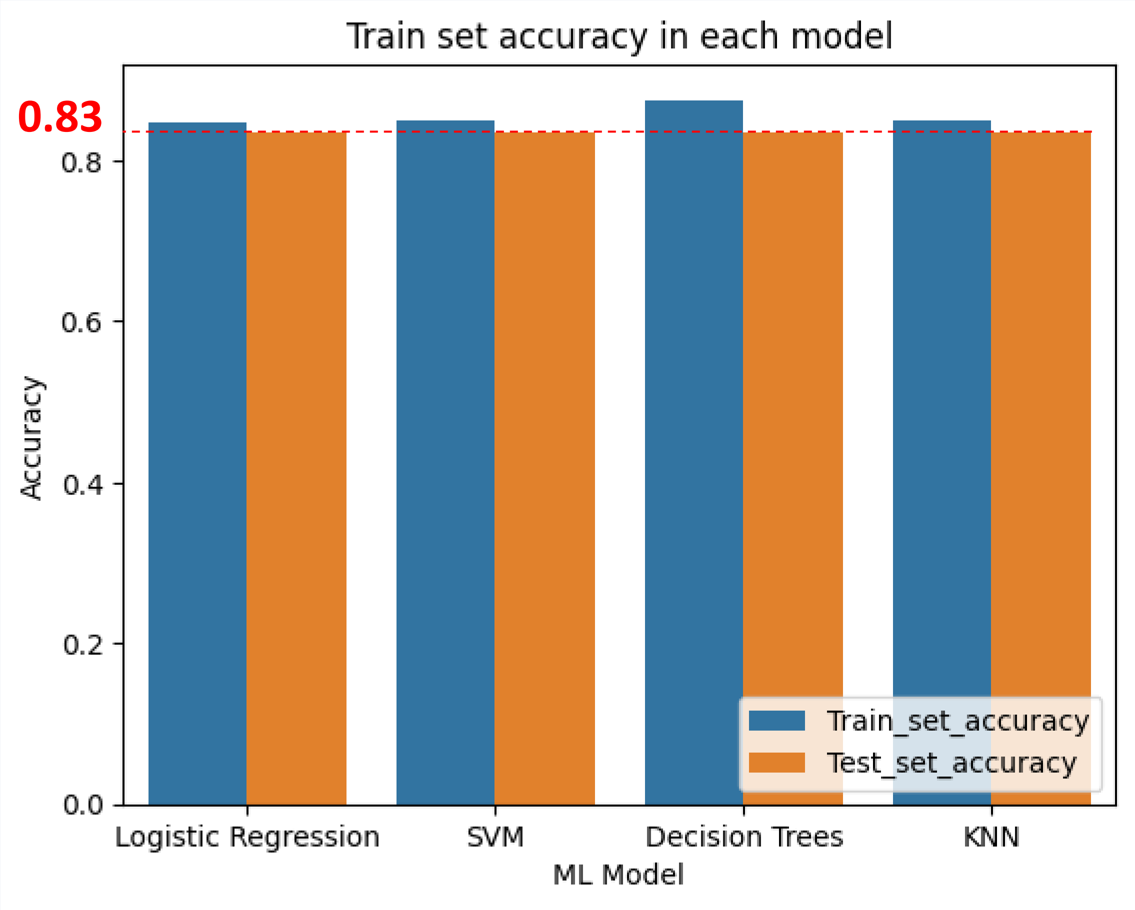
Results

- Success rate depending on PL and booster version:
  - FT version has the highest success rate
  - Higher success rate between 2000kg to 5500kg

Results

- Predictive analysis results
  - Same **test set accuracy** for the 4 models
  - Similar train accuracy in all of them
  - Small size of the data set

	Model	Train_set_accuracy	Test_set_accuracy
0	Logistic Regression	0.846429	0.833333
1	SVM	0.848214	0.833333
2	Decision Trees	0.875000	0.833333
3	KNN	0.848214	0.833333

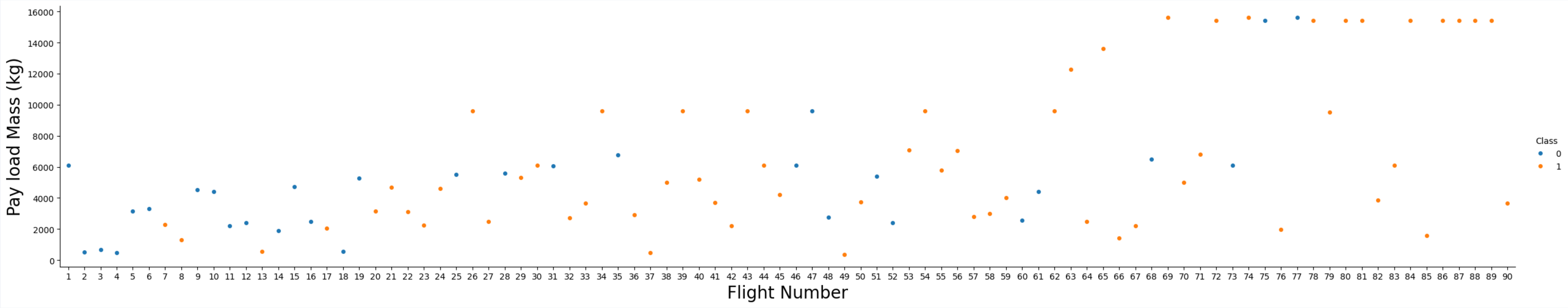


Section 2

Insights drawn from EDA

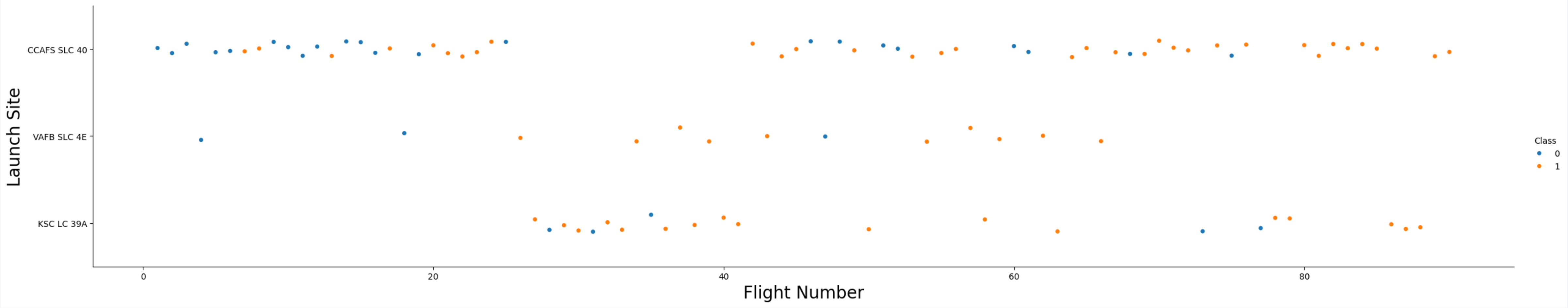
Payload vs. Flight number

- Success rate increases over time
- High failure rate at the beginning
- Quite good success rate for massive payloads (>10000kg)



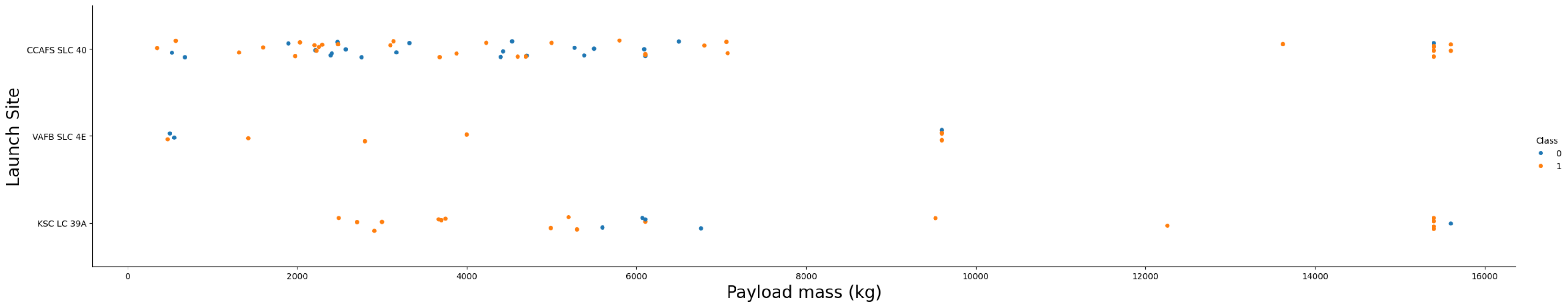
Launch site vs. Flight number

- Success rate increases over time
- There are some periods of inactivity in all launch sites
- Different success rates



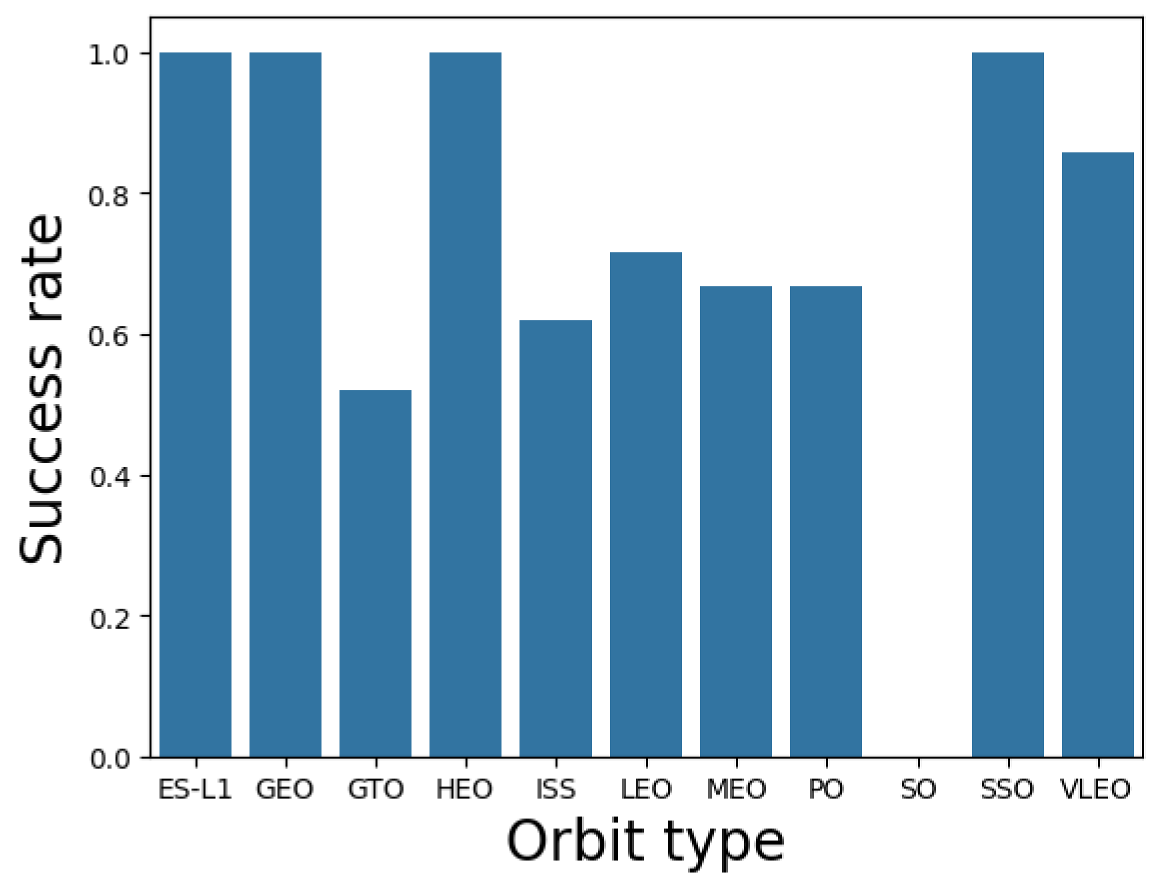
Payload vs. Launch Site

- Massive payloads launched in CCAFS SLC 40 or KSC LC 39A
- VAFB SLC 4E did not provide launches for PL > 10000kg
- Unsuccessful landings can be found in different PL ranges



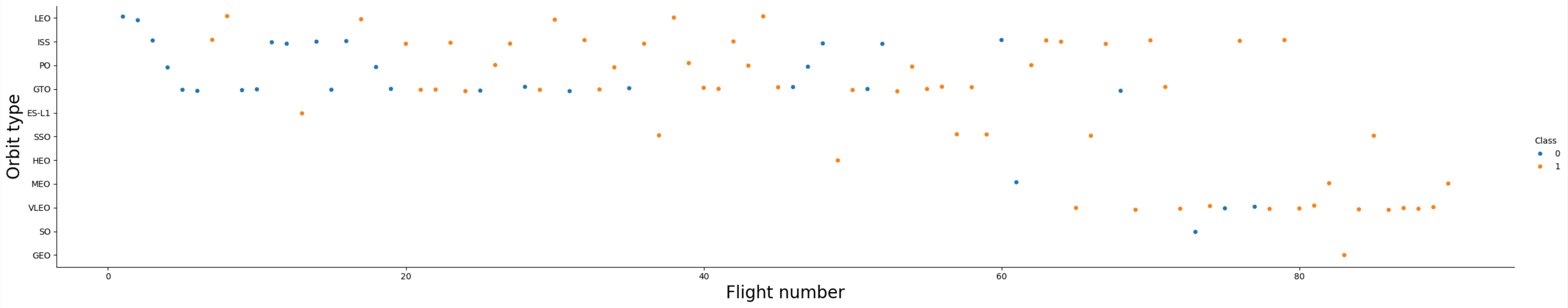
Success Rate vs. Orbit Type

- Some types have a 100% of successful landings:
  - ES-L1
  - GEO
  - HEO
  - SSO
- Especially low rate in GTO orbits (50%)



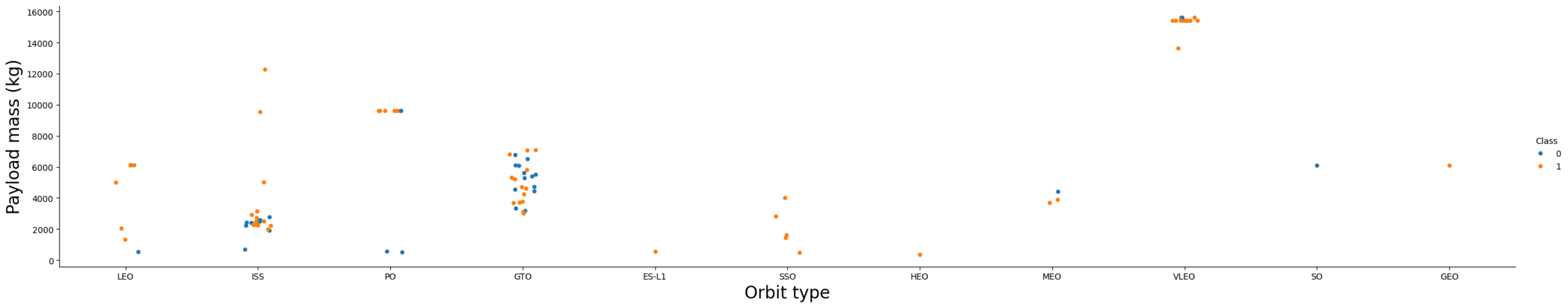
Flight Number vs. Orbit Type

- Success rate increases with flight number for LEO orbits
- Others like GTO have no relationship



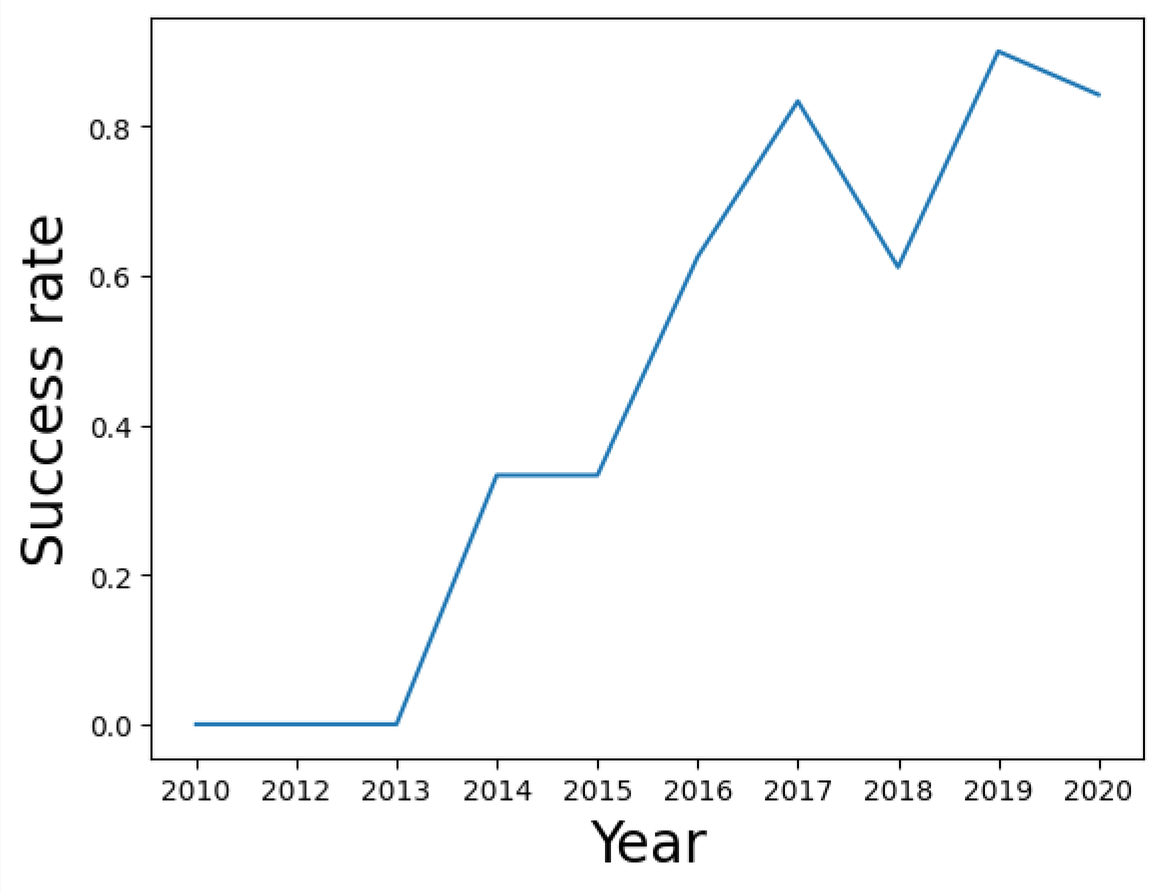
Payload vs. Orbit Type

- LEO, ISS and PO orbits have better success rate for heavy loads
- In GTO orbits, there's no relationship at all



Launch Success Yearly Trend

- Success rate kept on **increasing over time**
- Slight drop in 2018
- Experience is an important factor



All Launch Site Names

- Cape Canaveral Launch Complex 40 (CCAFS LC-40)
- Cape Canaveral Space Launch Complex 40 (CCAFS SLC-40)
- Kennedy Space Center Launch Complex 39A (KSC LC-39A)
- Vandenberg Space Launch Complex 4 (VAFB SLC-4E)

```
%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
```

✓ 0.0s

```
 * 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'

- 5 records where launch site is CCAFS LC-40:
  - NASA is the customer for 4 of these records and SpaceX in one of them
  - All these missions are for LEO orbits
  - There are 2 failure landings with parachute and 3 in which there wasn't even an attempt

```
%%sql
SELECT * FROM SPACEXTABLE
WHERE "Launch_Site" LIKE "CCA%"
LIMIT 5
```

✓ 0.0s Python

```
* sqlite:///my_data1.db
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	0	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	0	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	0	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	0	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	0	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- The total payload carried by the NASA (CRS) launchers is 45596kg

```
%%sql
SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTABLE
WHERE "Customer" == "NASA (CRS)"
```

✓ 0.0s

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

SUM("PAYLOAD_MASS__KG_")
45596

Average Payload Mass by F9 v1.1

- Average payload carried by booster version F9 v1.1
  - The average PL is **2534.7kg**. This means that most of the payloads are not excessively heavy compared to the massive payloads (over 10000kg) of some missions

```
%%sql
SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTABLE
WHERE "Booster_Version" LIKE "%F9 v1.1%"
```

✓ 0.0s

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

AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

First Successful Ground Landing Date

- Date of the first successful landing on ground pad
  - First, check the possible landing outcomes

```
%sql SELECT DISTINCT "Landing_outcome" FROM SPACEXTABLE
✓ 0.0s
```

  - Then, get the date: 22nd of December 2015

```
%%sql
SELECT MIN("Date") AS "First_success_ground_pad" FROM SPACEXTABLE
WHERE "Landing_Outcome" == "Success (ground pad)"
✓ 0.0s

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

First_success_ground_pad
2015-12-22

Landing_Outcome
Failure (parachute)
No attempt
Uncontrolled (ocean)
Controlled (ocean)
Failure (drone ship)
Precluded (drone ship)
Success (ground pad)
Success (drone ship)
Success
Failure
No attempt

Successful Drone Ship Landing with Payload between 4000 and 6000

- Names of the boosters with successful landings on drone ship and PL between 4000 and 6000kg:
  - There are 4 different versions for this specific PL range and landing

```
%%sql
SELECT DISTINCT "Booster_Version" FROM SPACEXTABLE
WHERE "Landing_Outcome" == "Success (drone ship)"
AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000
```

✓ 0.0s

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

- List the total number of successful and failure missions
  - Regardless of the landing outcome, there's only one unsuccessful mission
  - High reliability on mission success

```
%%sql
SELECT "Mission_Outcome", COUNT(*) AS "Count" FROM SPACEXTABLE
GROUP BY "Mission_Outcome"
```

✓ 0.0s

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

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

Boosters Carried Maximum Payload

- Names of the boosters with maximum payload:
  - 12 versions were able to carry 15600kg
  - These are less common, given that the average PL is around 2500kg

```
%%sql
SELECT DISTINCT "Booster_Version", "PAYLOAD_MASS__KG_" FROM SPACEXTABLE
WHERE "PAYLOAD_MASS__KG_" == (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE)
```

✓ 0.0s

* sqlite:///my_data1.db
Done.

Booster_Version	PAYLOAD_MASS__KG_
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

- Records with the month name, booster version and launch site for the year 2015 where **landing outcome in drone ship is failure**
  - In this period, 2 launches had this scenario
  - Both launched from CCAFS LC-40

```
%%sql

SELECT SUBSTR(Date, 6, 2) AS "Month", "Landing_Outcome", "Booster_Version", "Launch_site" FROM SPACEXTABLE
WHERE "Landing_Outcome" == "Failure (drone ship)"
AND SUBSTR(Date,0,5) == "2015"
```

✓ 0.0s

```
 * 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

- In this specific scenario, we find that a remarkable part of the launches didn't even make an attempt of landing: 10 of them
- There are only 8 cases in which the landing was successful:
  - 5 in drone ship
  - 3 in ground pad

```
%%sql
SELECT "Landing_Outcome", COUNT("Landing_Outcome") AS "Count" FROM SPACEXTABLE
WHERE "Date" BETWEEN "2010-06-04" AND "2017-03-20"
GROUP BY "Landing_Outcome"
```

✓ 0.0s

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

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

Section 3

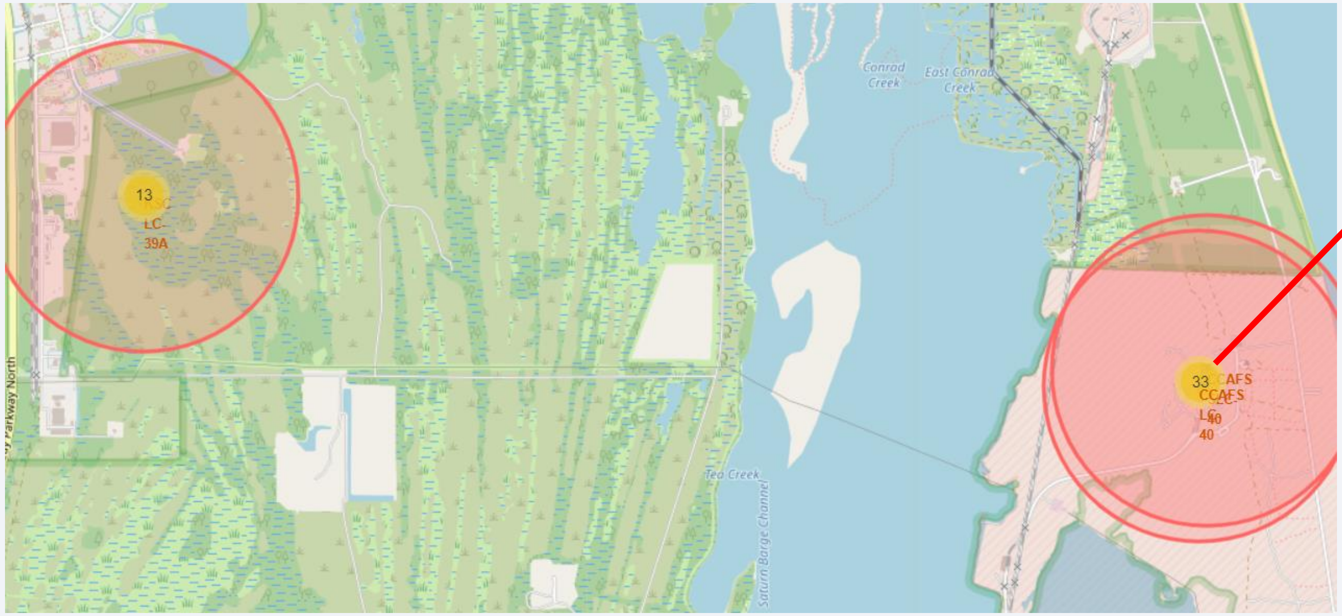
Launch Sites Proximities Analysis

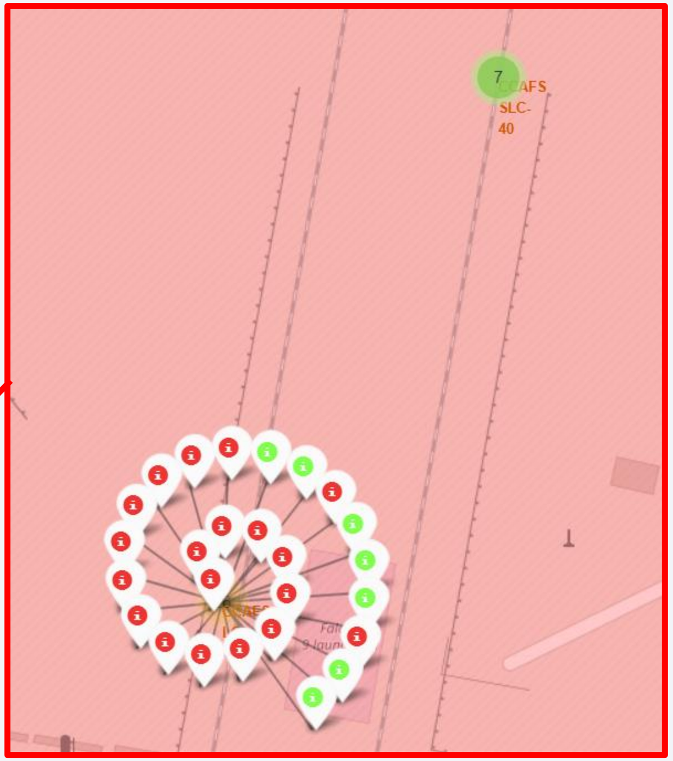
Launch sites' locations

- Circles marking the launch sites:
  - All locations close to coastline
  - VAFB SLC-4E on the West Coast close to LA; rest of sites on the East Coast in Florida

Markers for different outcomes

- Marker cluster to deal with multiple overlapping markers
- Marker colors depending on the outcome:
  - Green: successful landing (1) / Red: unsuccessful landing (0)





Distance to coastline

- Polylines added with marker to indicate the distance between a launch site and a point of interest
- Distance from CCAFS SLC-40 to coastline: 0.86km

Distance to airport

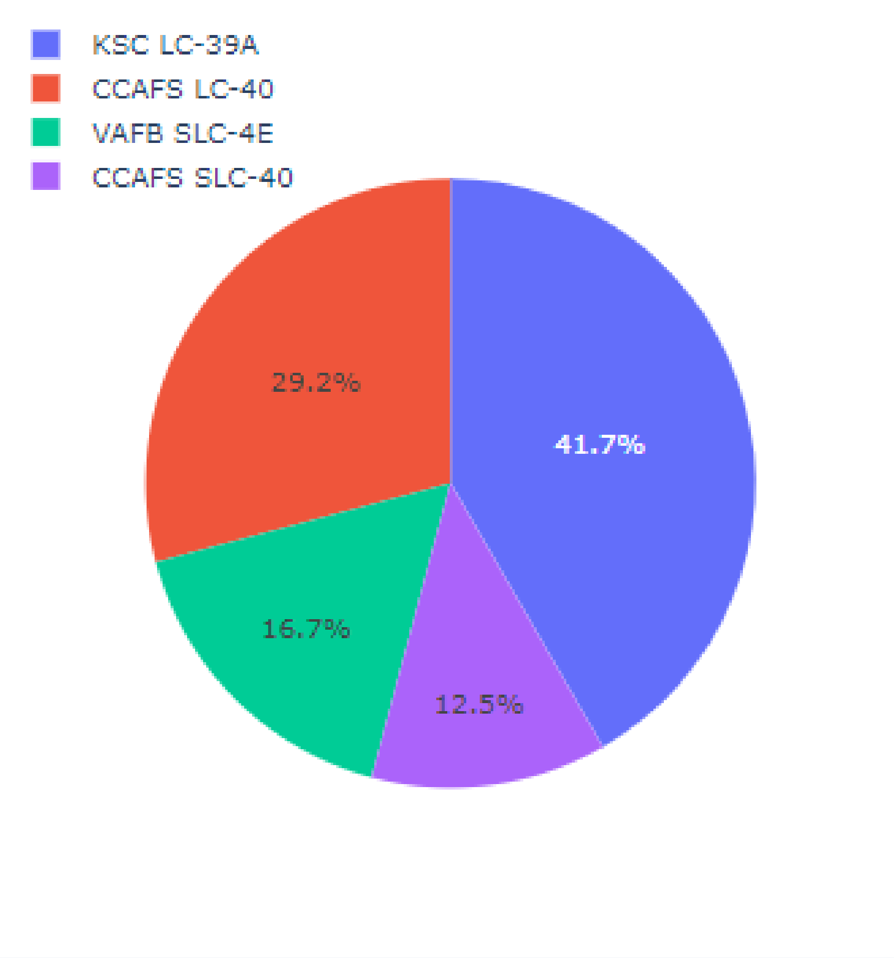
- Distance from CCAFS SLC-40 to Space Coast Regional Airport: 22.38km

Section 4

Build a Dashboard with Plotly Dash

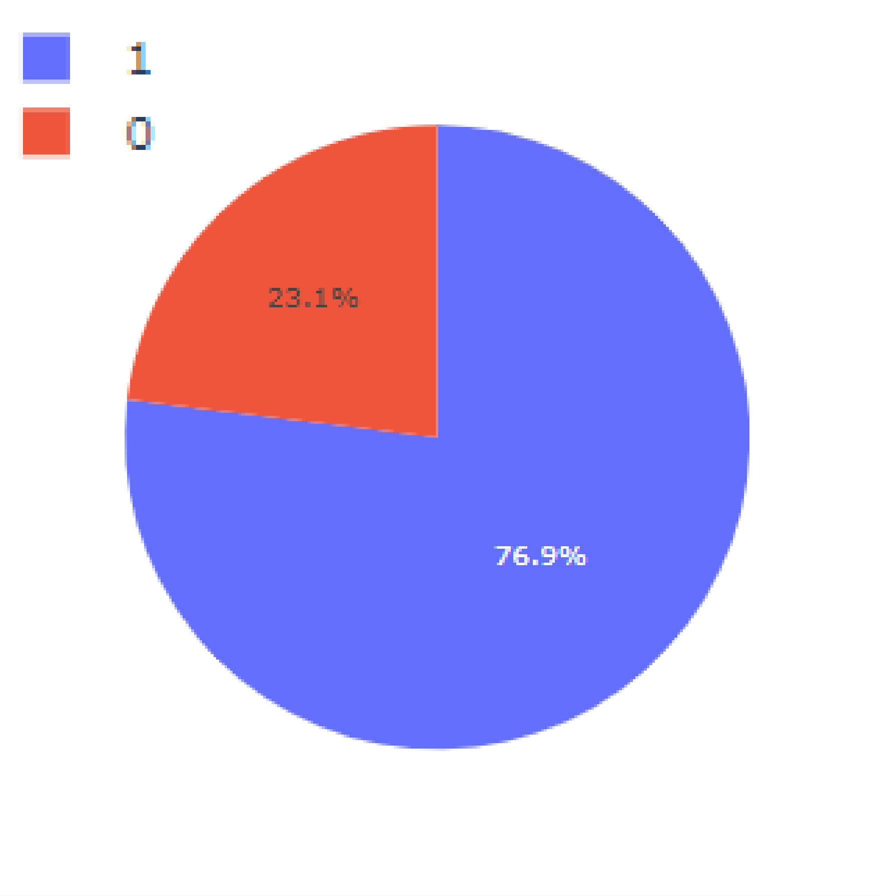
Successful launches in all sites

- The KSC LC-39A has the largest proportion of successful launches of all sites: 41.7%
- The smaller proportion is in CCAFS SLC-40: 12.5%
- For the CCAFS LC-40 there's the 29.2 % of successful launches
- VAFB SLC-4E has a 16.7%



KSC LC-39A successful launches

- The KSC LC-39A also has the largest success rate
- Examining this site, almost a 77% of all its launches were successful
- Only a 23% were failures

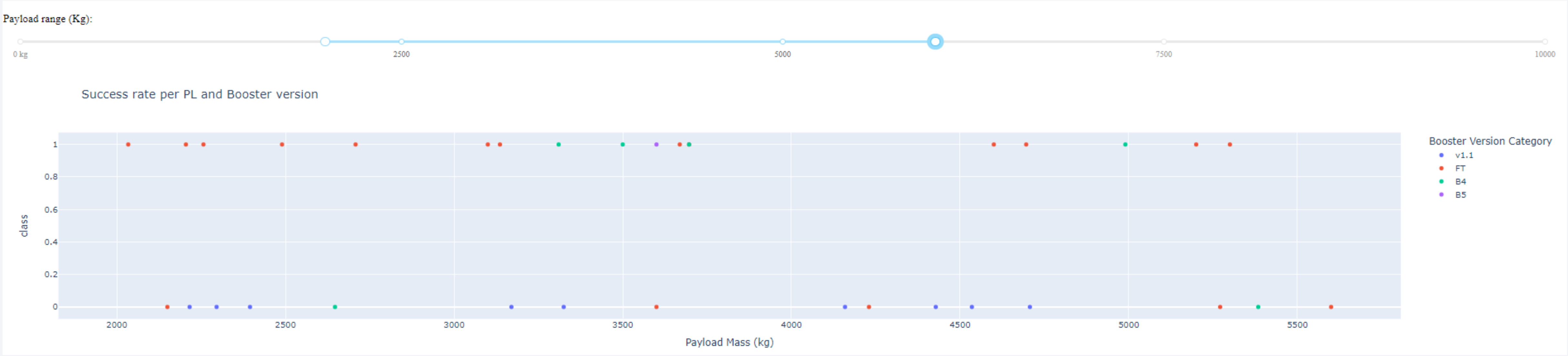


Payload vs. Launch outcome

- This is the launch outcome depending on the **Payload** and considering the **booster version**
- The highest success rate takes place approximately between 2000kg and 5500kg

Payload vs. Launch outcome (2000-5500kg)

- Looking closer to this range, we can see a specific booster version with more frequency among the successful cases
- The **Booster version FT has the highest success rate**
- On the other hand, the v1.1 has the lowest success rate



Section 5

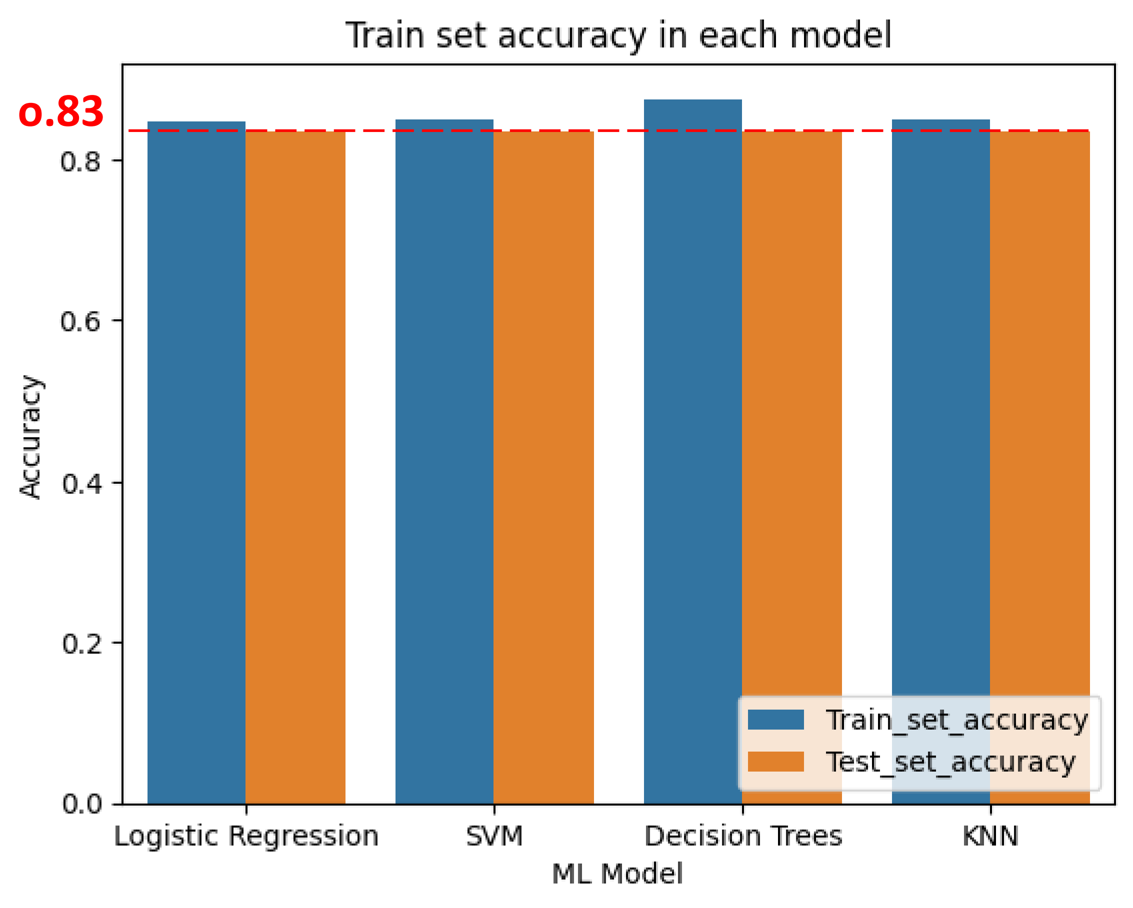
Predictive Analysis (Classification)

Classification Accuracy

- Predictive analysis results
  - Same **test set accuracy** for the 4 models
  - Similar train accuracy in all of them
  - Small size of the data set

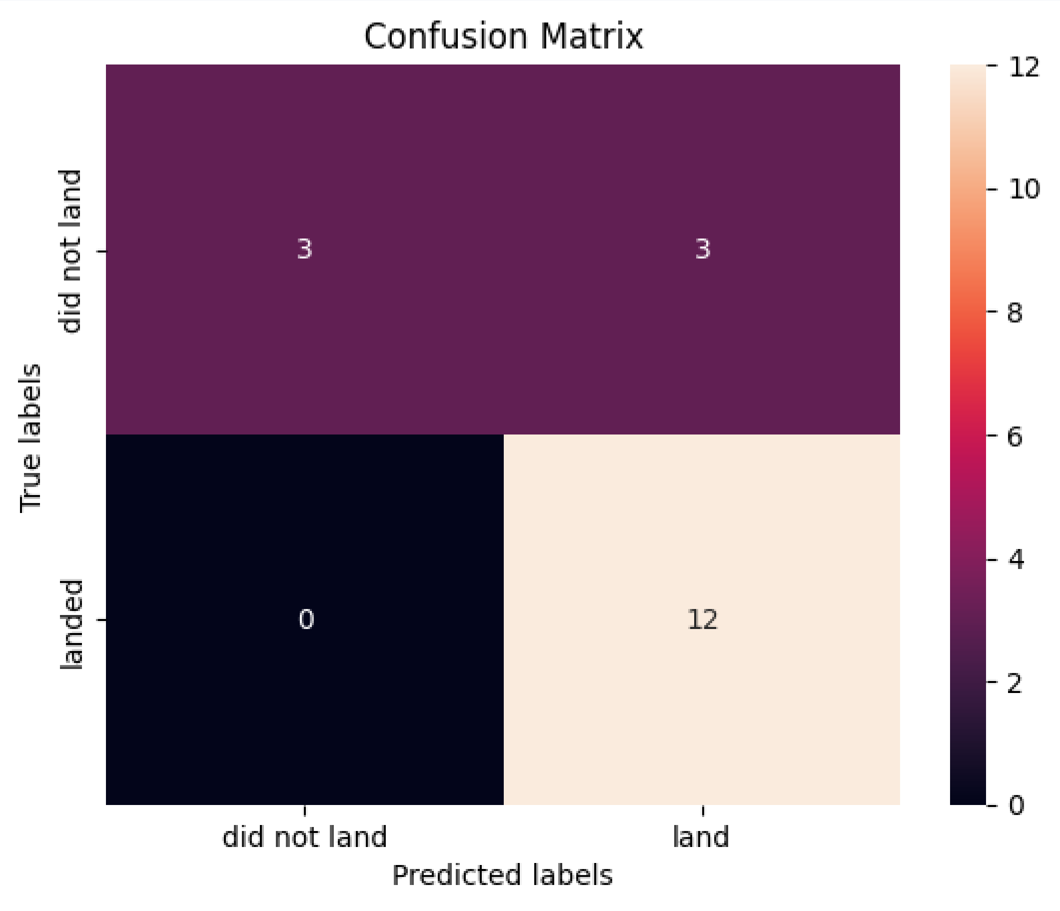
	Model	Train_set_accuracy	Test_set_accuracy
0	Logistic Regression	0.846429	0.833333
1	SVM	0.848214	0.833333
2	Decision Trees	0.875000	0.833333
3	KNN	0.848214	0.833333

0.83



Confusion Matrix

- The confusion matrix turns out to be the same for the 4 models due to the small size of the data set and similar accuracy
- We can see that the landed cases were all correctly predicted
- However, on the unsuccessful cases, half of them were predicted as 'landed' when they were not
- Therefore, the major problem are the **False positives**



Conclusions

- It is possible to build a ML model to predict the landing outcome in a mission, taking into account several parameters of previous launches
- Parameters such as the **payload**, **launch site location** and even the **type of orbit** (in some cases) turned out to have an important influence on the landing outcome
- The **interactive visualization** techniques are extremely helpful to compare different data sets and ranges that can unveil interesting insights
- The models have **remarkable accuracy** (over 80%) for the testing sets, but the confusion matrix reveals a **problem with the False positives**
- The **data set is still small**, so it would be convenient to carry out further development when the data amount grows more

Appendix

- Parameters used for each ML model development:

  - For logistic regression:

```
parameters ={'C':[0.01,0.1,1],
            'penalty':['l2'],
            'solver':['lbfgs']}
```
✓ 0.0s

  - For SVM:

```
parameters = {'kernel':('linear', 'rbf','poly','rbf', 'sigmoid'),
             'C': np.logspace(-3, 3, 5),
             'gamma':np.logspace(-3, 3, 5)}
svm = SVC()
```
✓ 0.0s

  - For Decision Tree:

```
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
```
✓ 0.0s

  - For KNN:

```
parameters = {'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
              'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'],
              'p': [1,2]}

KNN = KNeighborsClassifier()
```
✓ 0.0s

Appendix

- Find the best performance model (code snippets):

```
model_list = [logreg_cv, svm_cv, tree_cv, knn_cv]
model = ['Logistic Regression', 'SVM', 'Decision Trees', 'KNN']
train_acc = [mod.best_score_ for mod in model_list]
test_acc = [mod.score(X_test, Y_test) for mod in model_list]

df = pd.DataFrame()
df['Model'] = model
df['Train_set_accuracy'] = train_acc
df['Test_set_accuracy'] = test_acc

df
```

✓ 0.0s

	Model	Train_set_accuracy	Test_set_accuracy
0	Logistic Regression	0.846429	0.833333
1	SVM	0.848214	0.833333
2	Decision Trees	0.875000	0.833333
3	KNN	0.848214	0.833333

- Plot accuracy:

```
df1 = pd.melt(df, id_vars="Model", var_name="Train/Test", value_name="Accuracy")

ax = sns.barplot(x='Model', y='Accuracy', hue='Train/Test', data=df1)
plt.title('Train set accuracy in each model')
plt.xlabel('ML Model')
plt.ylabel('Accuracy')
plt.legend(loc='lower right')
```

✓ 0.1s

Thank you!