IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Paul Gieske
29 / Feb / 2024

Outline

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

Executive Summary

In this project we collect SpaceX data and we use it to create a classification model to predict where a Falcon 9 flight will have a successful or unsuccessful landing outcome.

We collect data using the SpaceX API and we Scrape data from wikipedia. We wrangle the data to put it in a form that we can use it to create models. We use SQL queries and visualisation to familiarise ourselves with the data. Finally, we train and evaluate a logistic regression model, a k-nearest neighbors model, a classification tree and a support vector machine.

When we evaluate the models we see that all 4 models perform equally well. Changing the distribution of train and testing data points has a significant impact on the model performance. Further research is required.

Introduction

SpaceX has managed to lower operation costs significantly by recovering the rocket. In this manner they are very competitive. The recovery of the rocket is no always successful however. It would be in our strategic interest to be able to predict when the recovery will and won't be successful.

In this project we will attempt to create a model that will predict whether a SpaceX Falcon 9 flight landing outcome will be successful or not. The prediction will be based entirely on public information.

In order to achieve this we will collect publicly available data using the SpaceX api and by web scraping a wikipedia site. We will wrangle and explore the data. Finally we will train and evaluate 4 classification models.

Section 1

Methodology

Methodology

Executive Summary

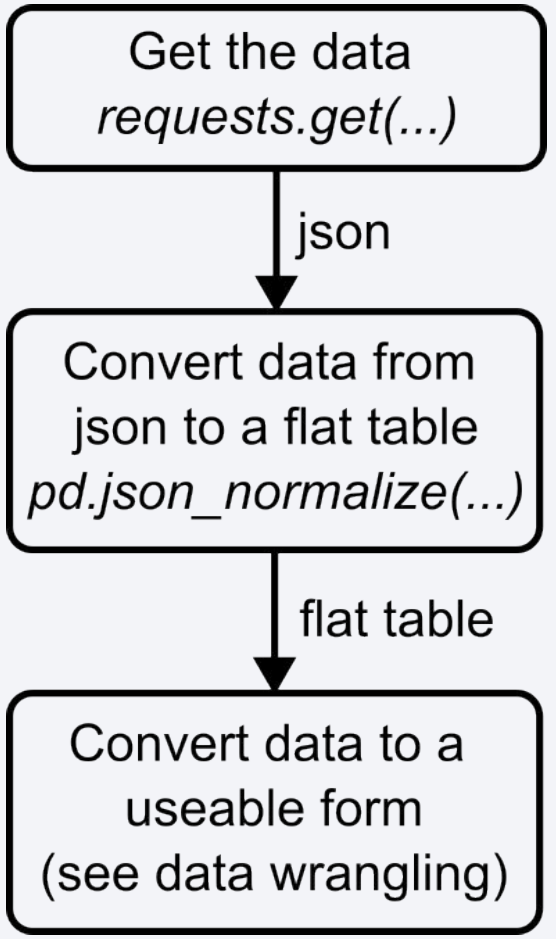
- Data collection methodology:
  - We used the SpaceX API to collect data about
  - We scraped the SpaceX Launches wikipedia page
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

How did we collect the data?

- Two ways:
  - We used the SpaceX API to collect data about
    - SpaceX provides a RESTful API
  - We scraped the wikipedia page:
    - “List of Falcon 9 and Falcon Heavy launches” (link)

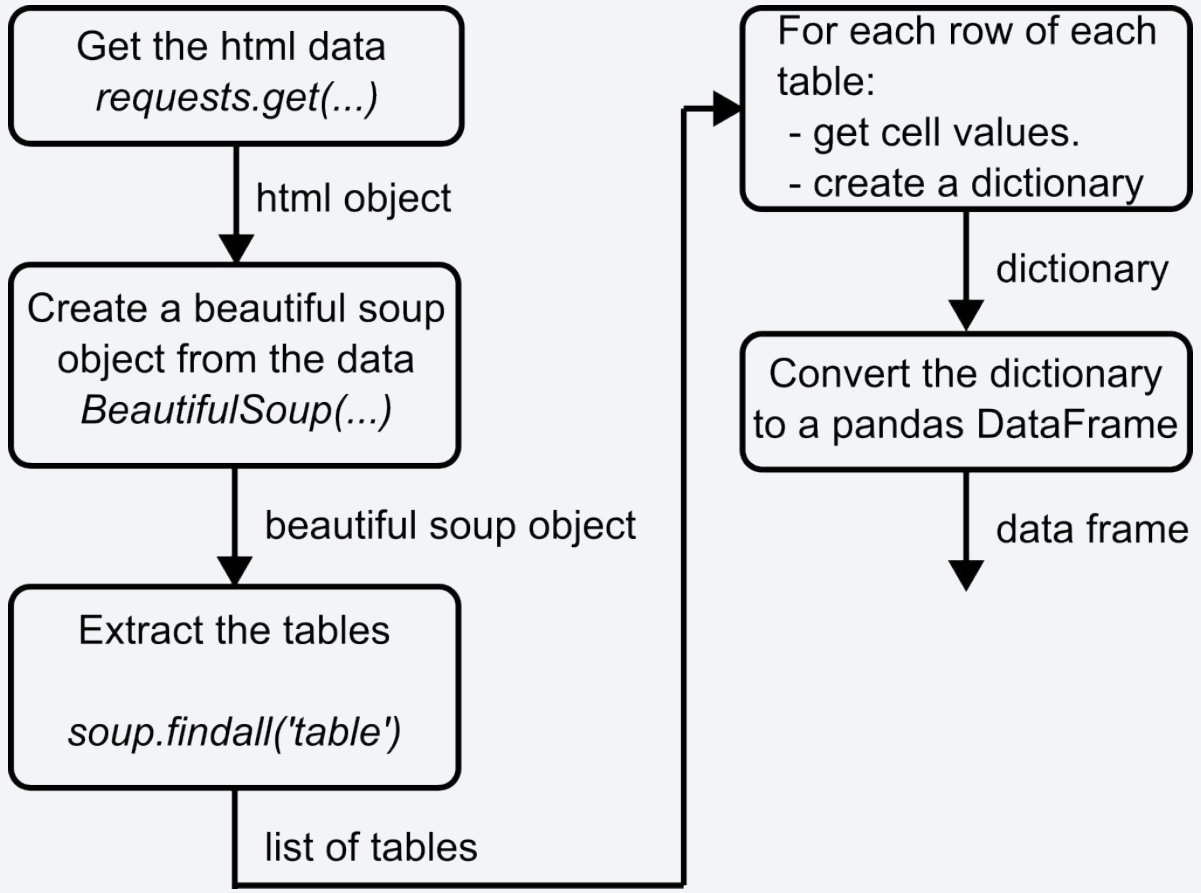
Data Collection – SpaceX API

- How did we collect the data?
  - We used the RESTful API to get the data in json format
  - We converted the json format to a flat table
  - Next we will convert the data to a more useable format
    - (See the section on data wrangling)
- Jupyter: (link)



Data Collection - Scraping

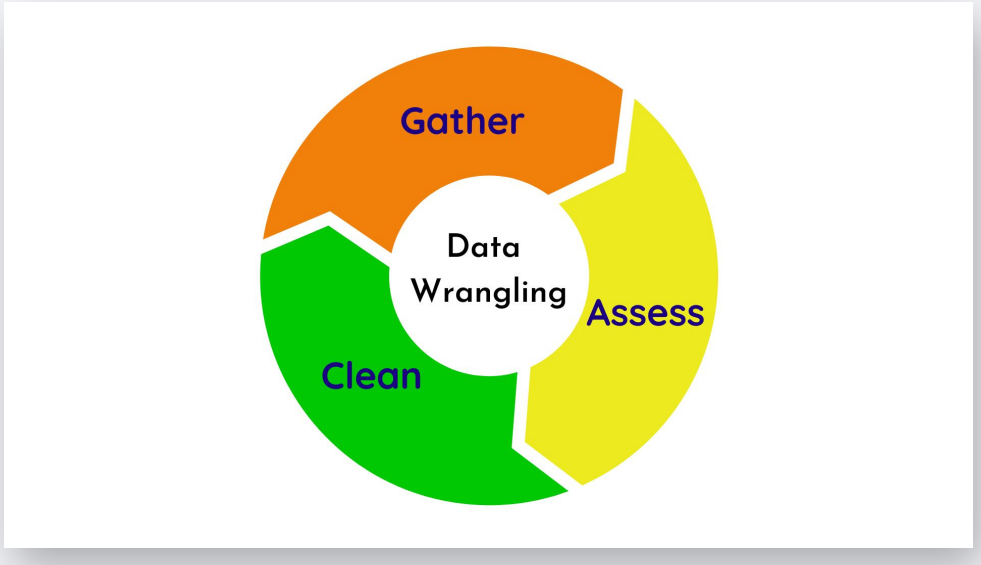
- How did we collect the data?
  - First we get the data
  - We convert it to a beautifulsoup object and extract all the tables
  - We loop through each row of every table to get the values we need and store it in a dictionary of lists
  - Finally we convert the list into a DataFrame



- Jupyter: (link)

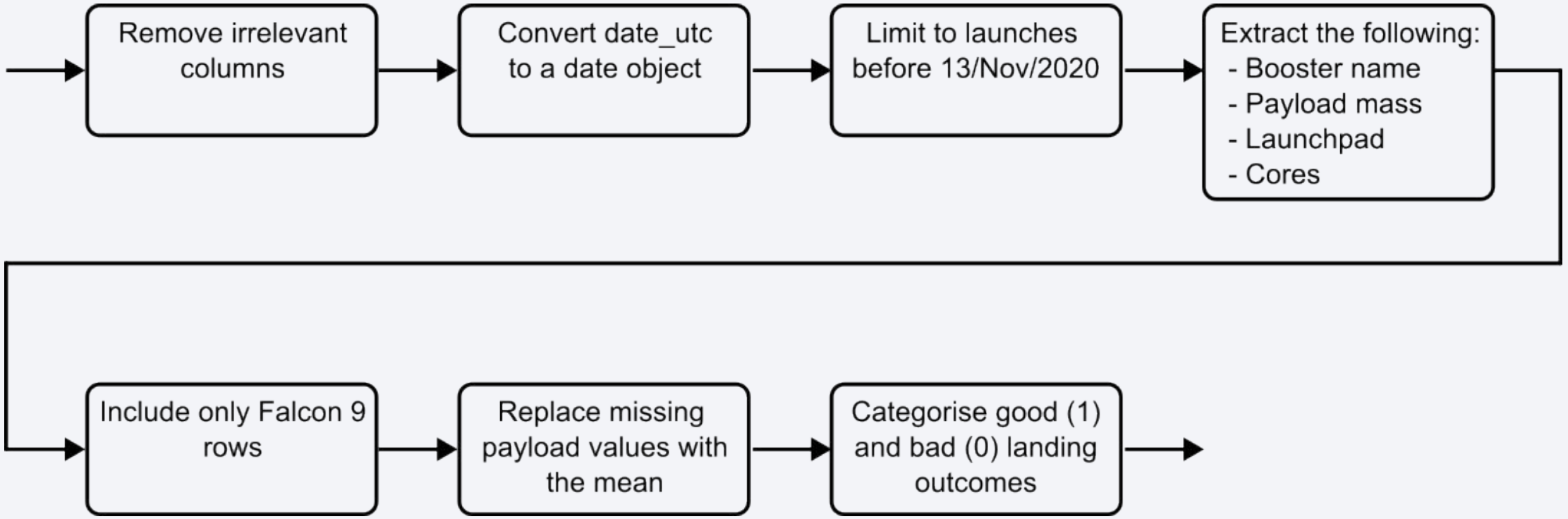
Data Wrangling - Description

- We perform the following steps in order to put the data into a usable format:
  - Remove irrelevant columns
  - Convert date_utc to date object
  - Limit to launches before 13/Nov/2020
  - Extract the following information from various columns
    - Booster name
    - Payload mass
    - Launchpad
    - Cores
  - Filter out rows that do not include Falcon 9
  - Replace missing values in the payload mass with the mean payload
  - Finally we categories good (1) and bad (0) landing outcomes and add it as a new column to the dataset.



- Juptyer: (Datacollection by API, Data Wrangling)

Data Wrangling - Flowchart



EDA with Data Visualization

- To gain insight into the data we plotted the following charts:
  - Scatter: flight number vs launch site categorised by landing outcome (gives an overview of success/failed landings per site depending on flight number)
  - Scatter: payload mass vs launch site categorised by landing outcome (gives an overview of success/failed landings per site depending on payload mass)
  - Bar: landing outcome success rate for each orbit (gives an overview of which orbits have higher and lower landing outcomes)
  - Scatter: flight number vs orbit categorised by landing outcome (gives an overview of success/failed landings per orbit depending on flight number)
  - Scatter: payload mass vs orbit categorised by landing outcome (gives an overview of success/failed landings per orbit depending on payload mass)
  - Lineplot: year vs average landing outcome success rate (gives an overview of how the success rate has changed over time
- See Section 2 of this presentation for the plots.
- Jupyter: (link)

EDA with SQL

- To explore the data the following SQL commands were performed (see the Jupyter notebook linked below for the commands):
  - COUNT the number of launches be launch site
  - List five records where the launch site starts with ‘CCA’
  - Find the total payload mass for the customer ‘NASA (CRS)‘
  - Find the total payload mass for the booster ‘F9 v1.1‘
  - Find the date of the first successful landing outcome
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - List the total number of successful and failure mission outcomes
  - List the names of the booster_versions which have carried the maximum payload mass
  - 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
  - Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order
- Jupyter: (link)

Build an Interactive Map with Folium

- We also created an interactive folium map. We added the following elements to the map:

Location	Element added	Reason
NASA JSC	Red circle with popup	To easily be able to locate NASA JSC
	Marker	
Each SpaceX launch site	Red circle with popup	To easily be able to locate each launch site
	Marker	
	Marker cluster with individual landing outcome successes and failures marked in green and red respectively	To get a overview of success / fail rates per launch site location
CCAFS SLC-40	Distance lines and markers from the site to: - the nearest coastline - the nearest railway - the nearest highway - the nearest city	To get an easy overview of the relative distances involved

- Jupyter: (link)

Build a Dashboard with Plotly Dash

- We also created an dash. We added the following elements to the dash:
  - A drop down list where the user can select a launch site or ‘All Sites’
  - A slider where the user can select the payload range
  - A pie chart
    - If the user selected ‘All sites’ the pie chart shows the number of successful landing outcomes for each site
    - If the user select one of the site the pie chart shows the proportion of successful and unsuccessful landing outcomes for that site
  - A scatter plot
    - With the given payload range on the x-axis the plot shows the landing outcome of each site.
    - Each flight is categorized by the booster version
- Python: (link)

Predictive Analysis (Classification)

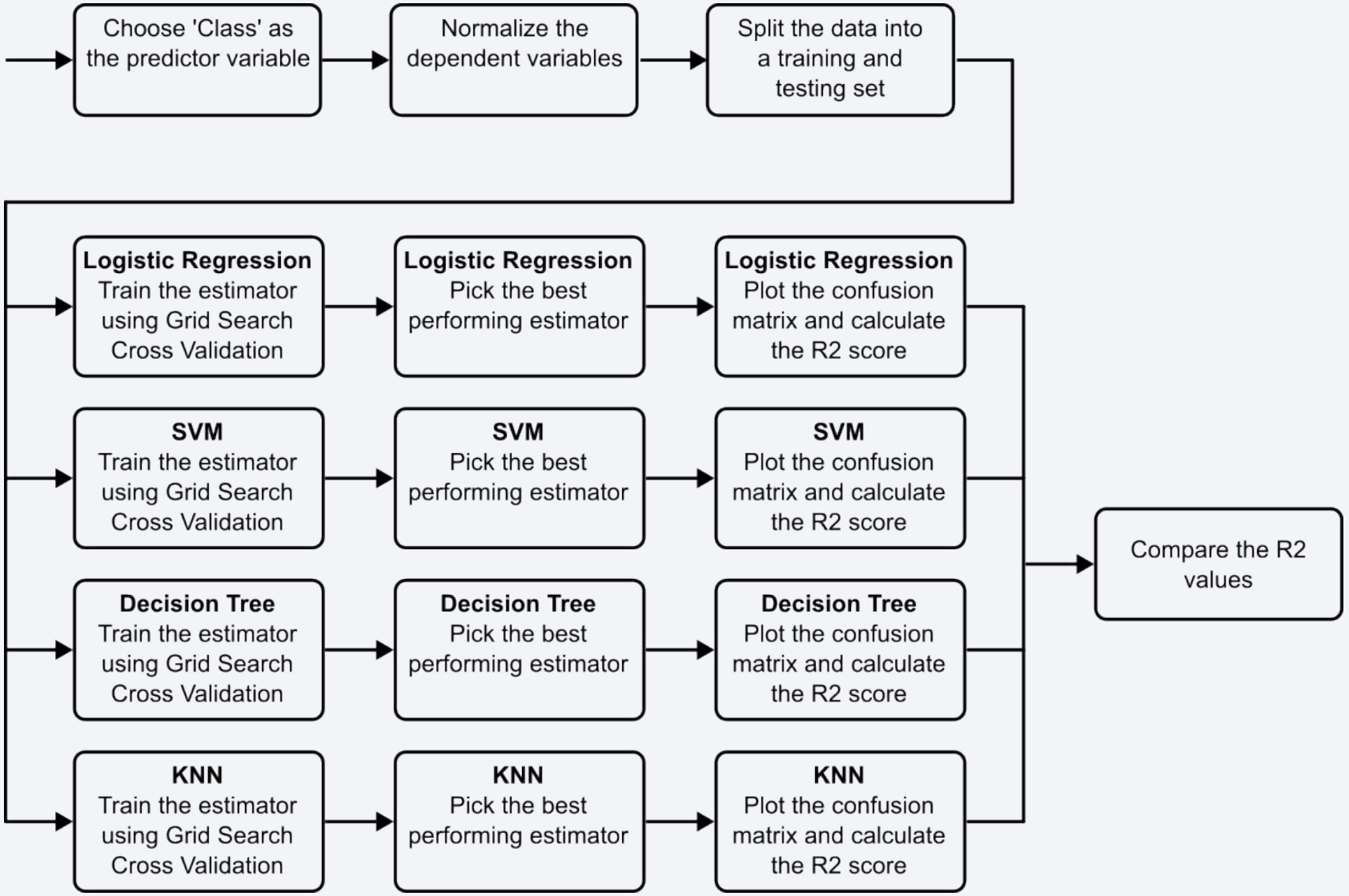
- In this section the goal is to create an estimator that predicts the landing outcome based on the data we have.
- We will build and compare the following types of classification models:
  - Logistic Regression
  - Support Vector Machine
  - Decision Tree Classifier
  - K Nearest Neighbors

- Jupyter: (link)

Predictive Analysis (Classification)

- We train and evaluate the estimators in the following way:
  - We choose ‘Class’ ad our predictor variable
  - We normalize the dependent variables
  - We split the data into a training set and a test set
  - For each model that we:
    - Train the estimator using Grid Search Cross Validation on the training data
    - We select the best performing model
    - We calculate the R2 score and plot the confusion matrix using the test data
- Comparing the R2 score of each model we can select which estimator is most suited for the task

Predictive Analysis (Classification) - Flow Chart



Results

- Exploratory data analysis results

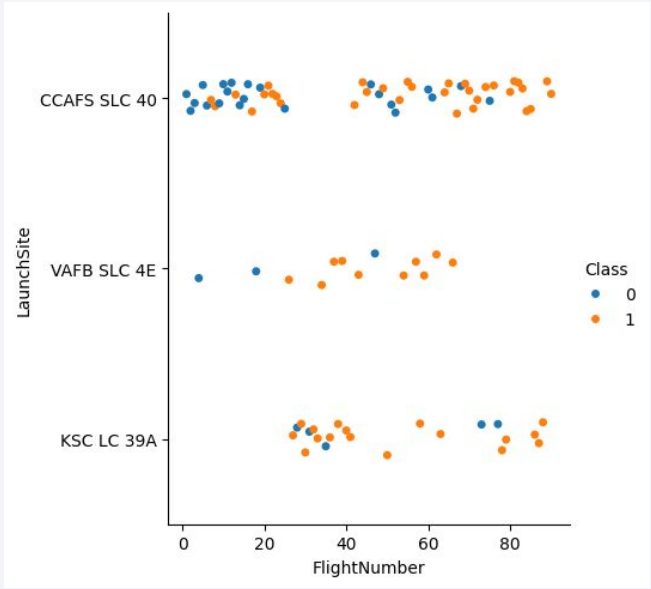
Flights per launch site	
CCAFS LC-40	26
CCAFS SLC-40	34
KSC LC-39A	25
VAFB SLC-4E	16

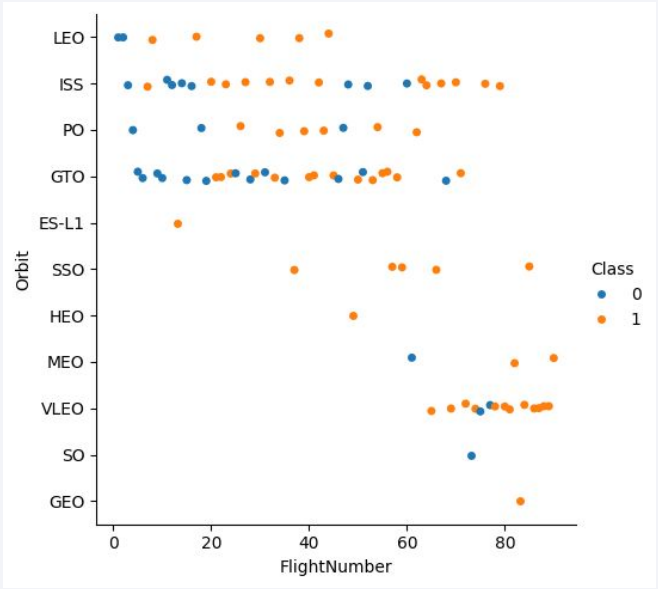
Total payload mass: NASA
45 596 kg

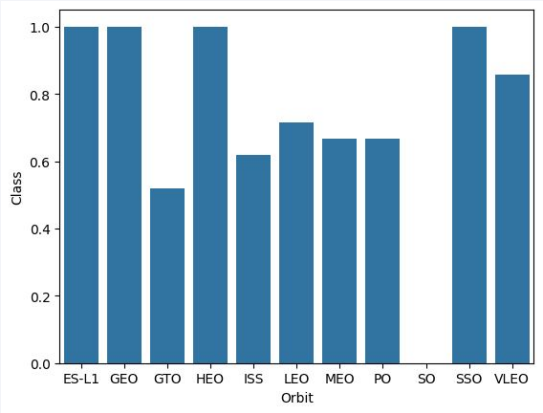
Average payload mass: F9 v1.1
2 928 kg

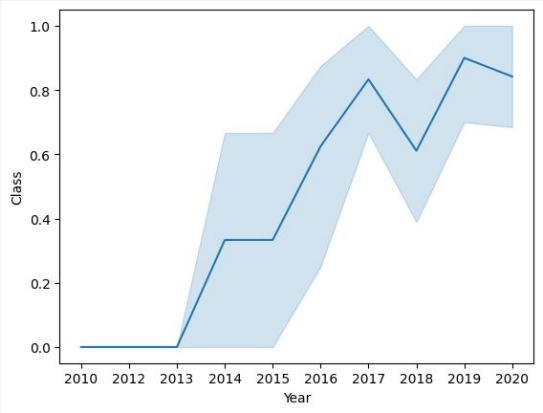
Mission successes and failures	
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Number of landing outcomes	
Success (drone ship)	12
No attempt	12
Success (ground pad)	8
Failure (drone ship)	5
Controlled (ocean)	4
Uncontrolled (ocean)	2
Precluded (drone ship)	1



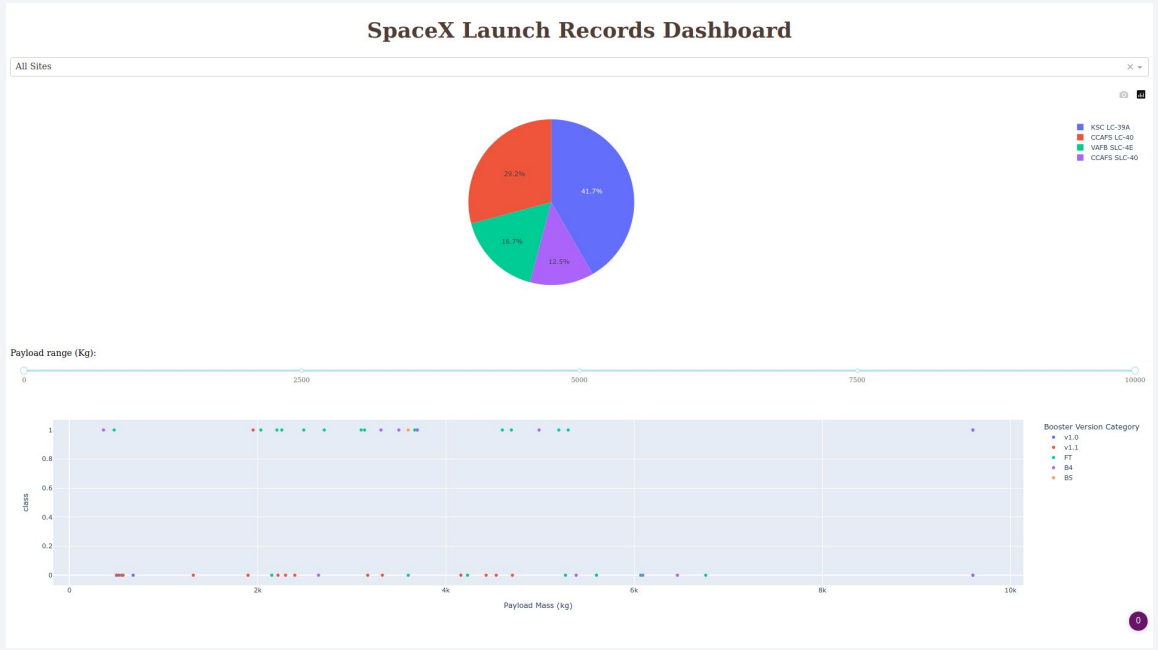


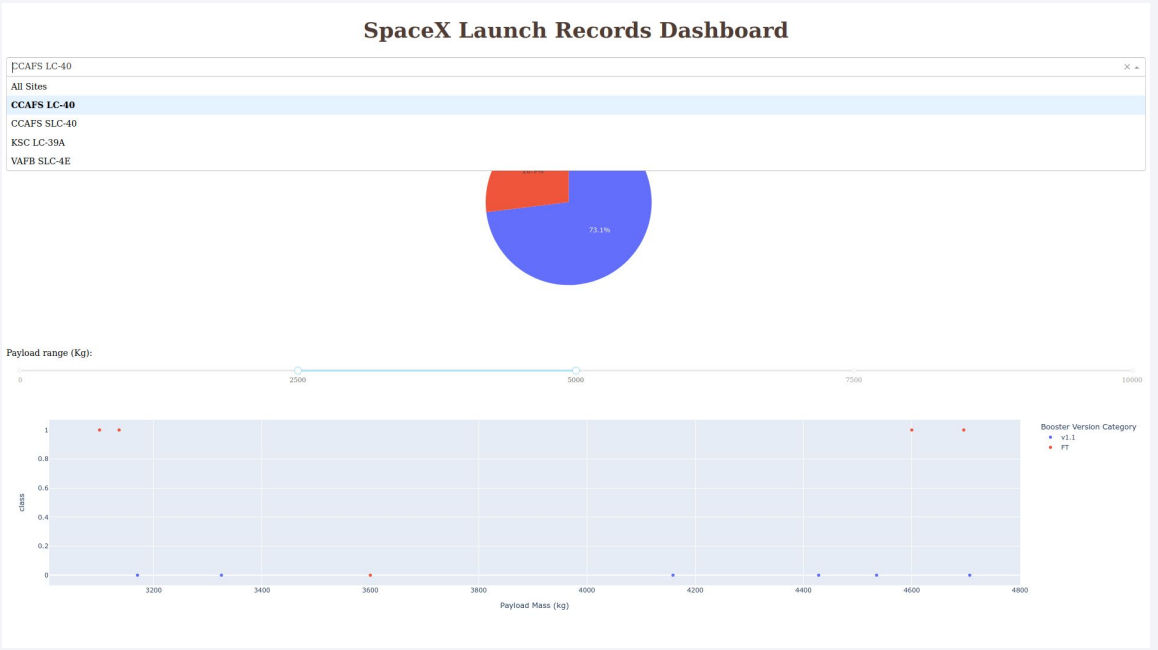




Results

- Interactive analytics demo in screenshots

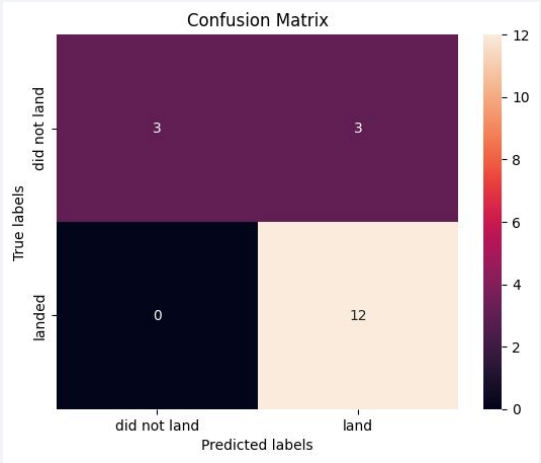




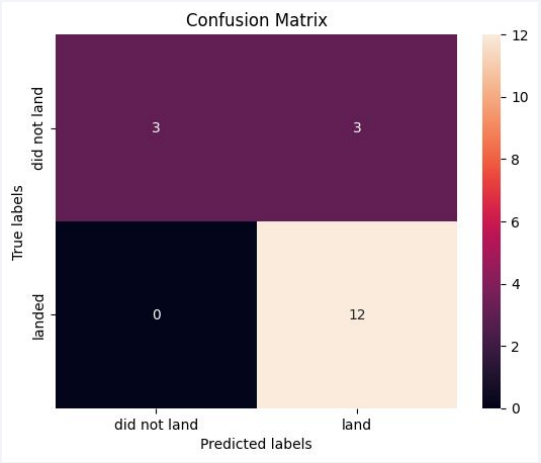
Results

- Predictive analysis results:

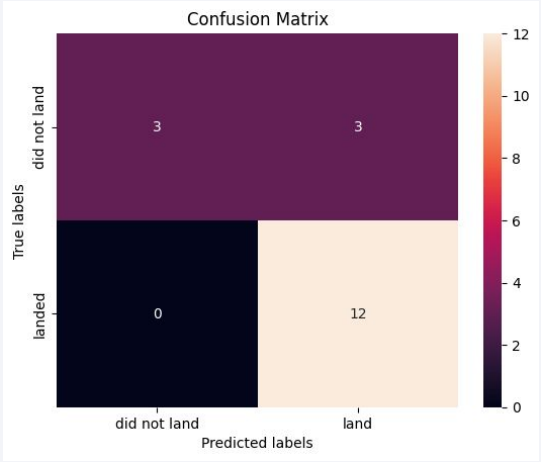
Logistic Regression: score = 83%



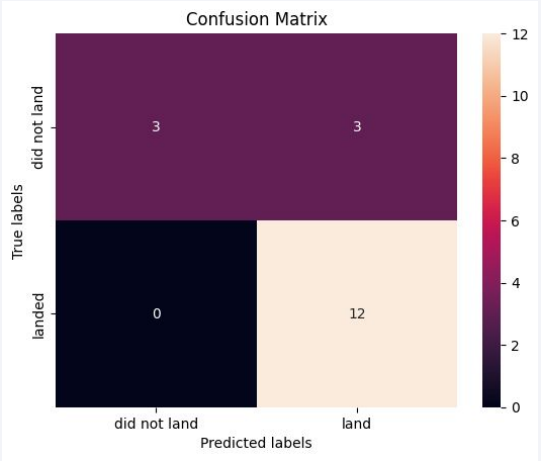
SVM: score = 83%



Classification Tree: score = 83%

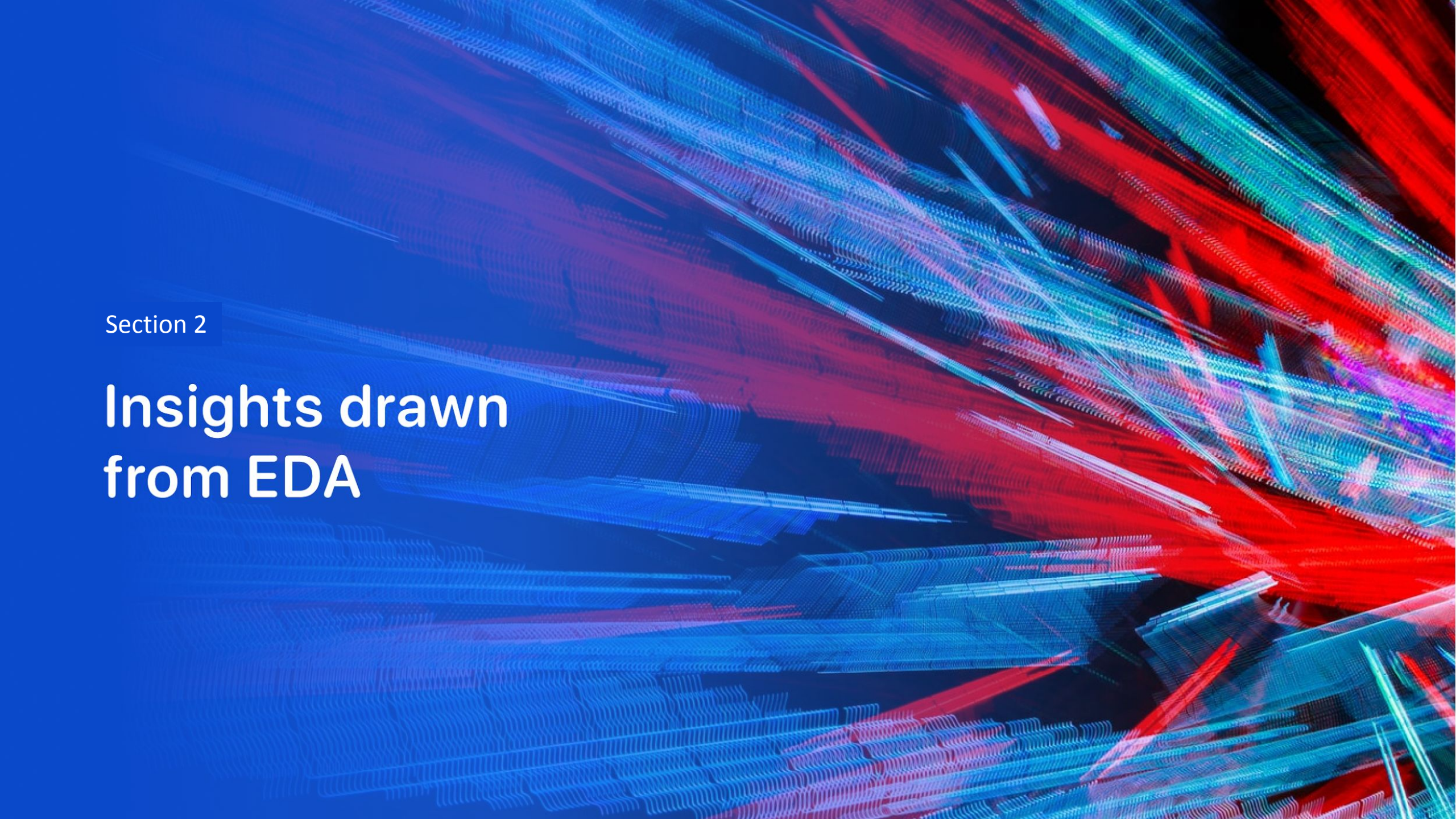


KNN: score = 83%

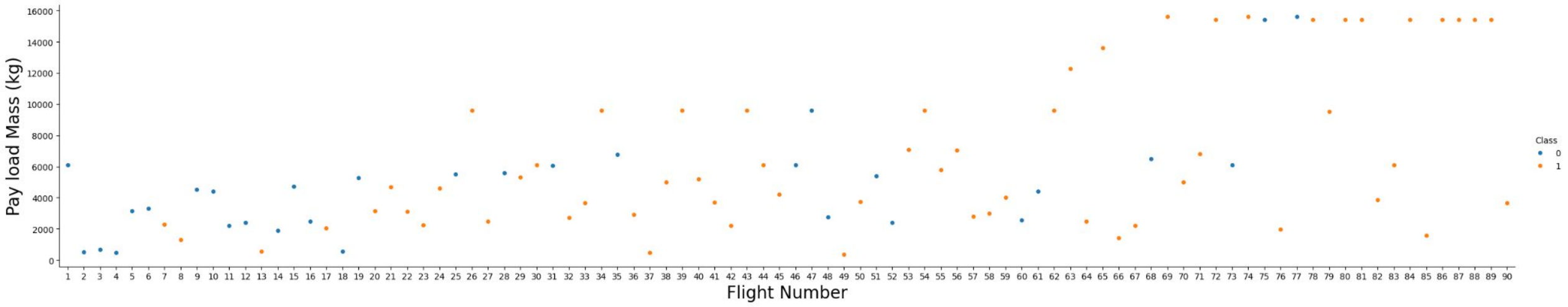


Section 2

Insights drawn from EDA



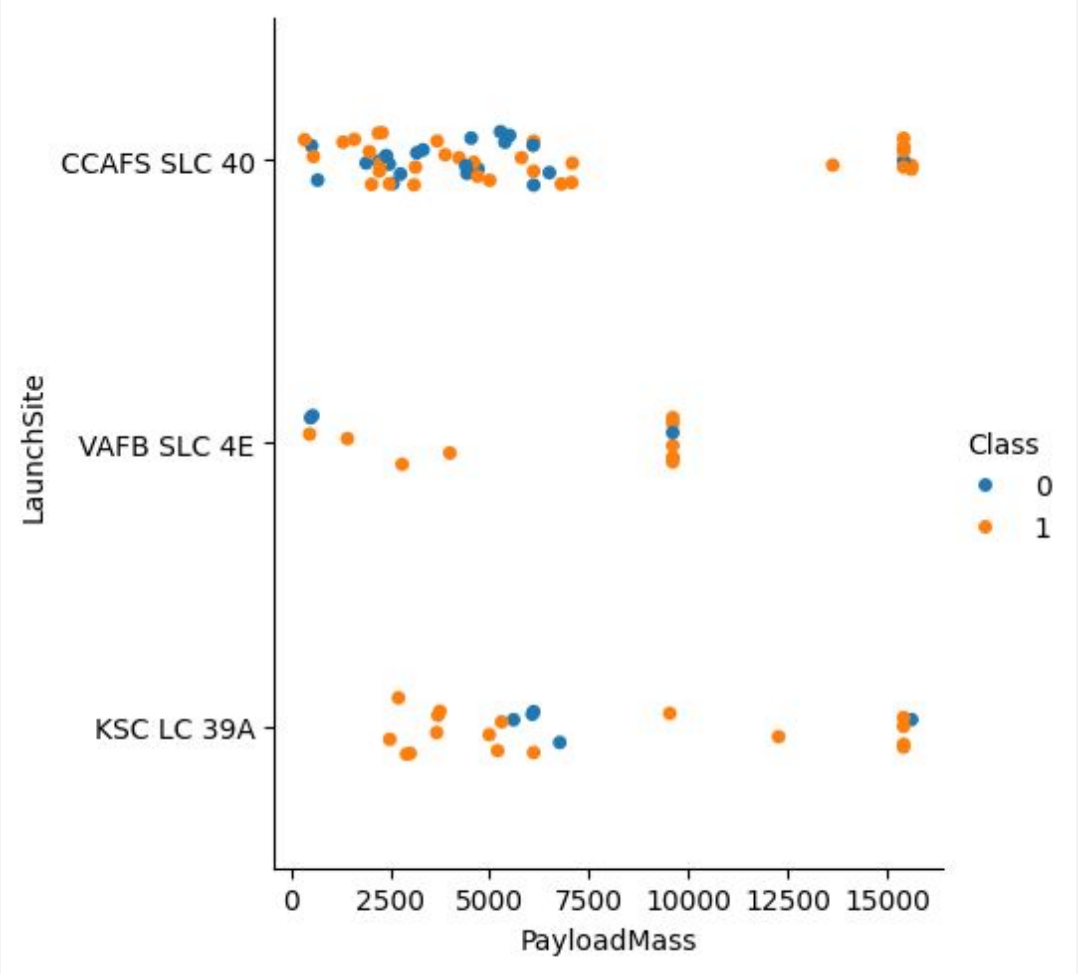
Flight Number vs. Launch Site



- Show the screenshot of the scatter plot with explanations

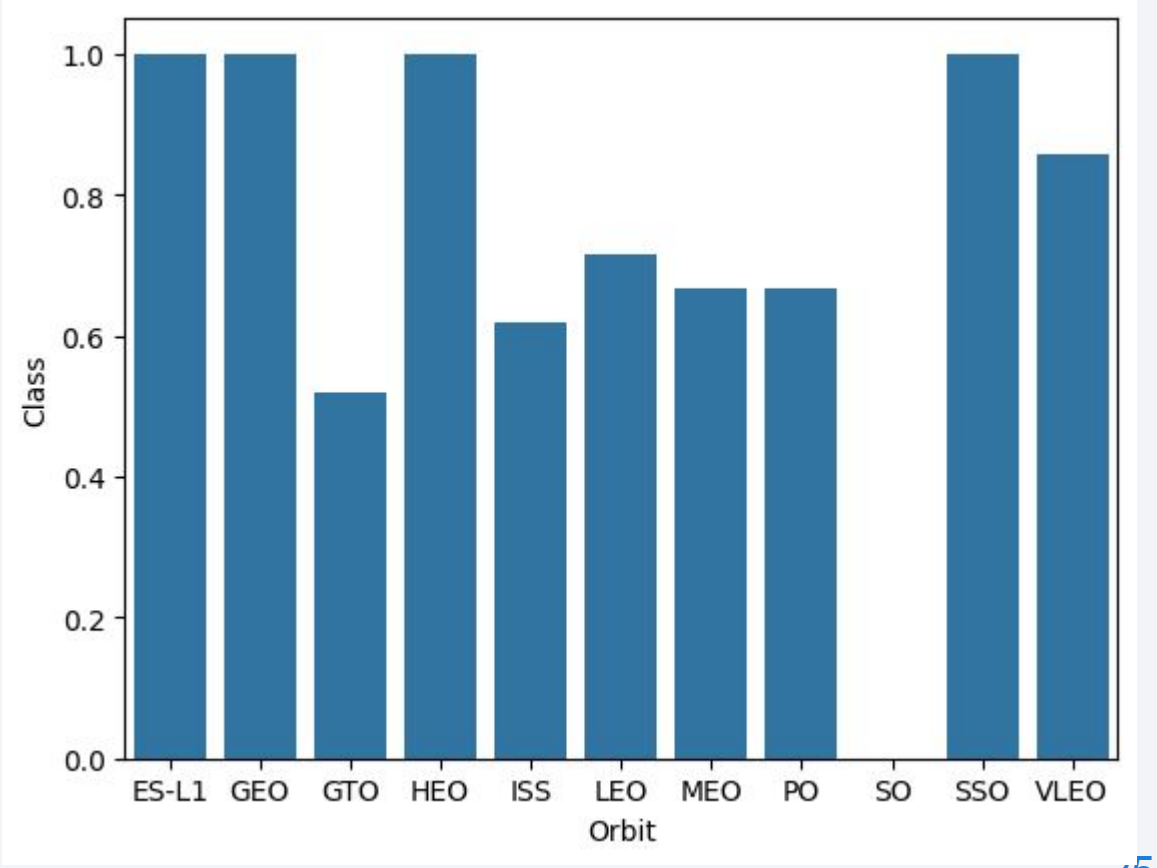
Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site
- Show the screenshot of the scatter plot with explanations



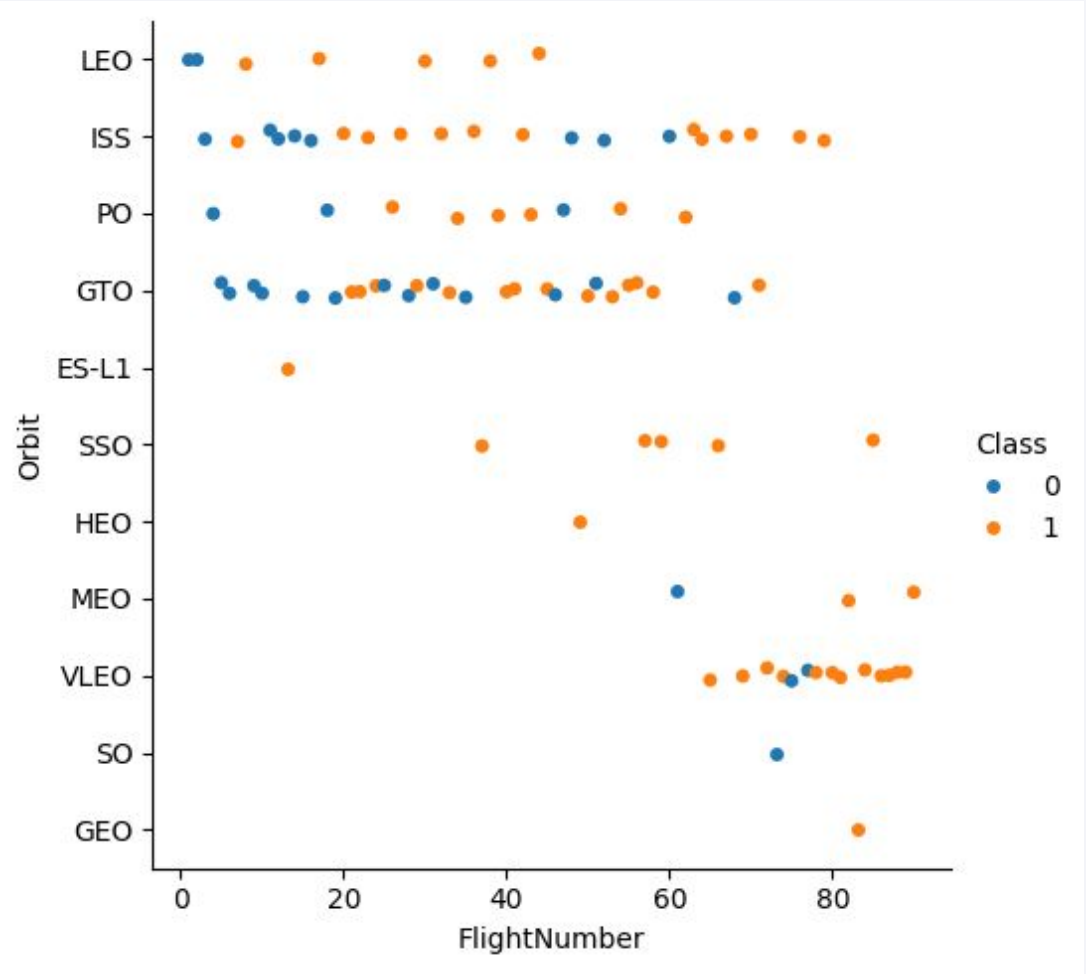
Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type
- Show the screenshot of the scatter plot with explanations



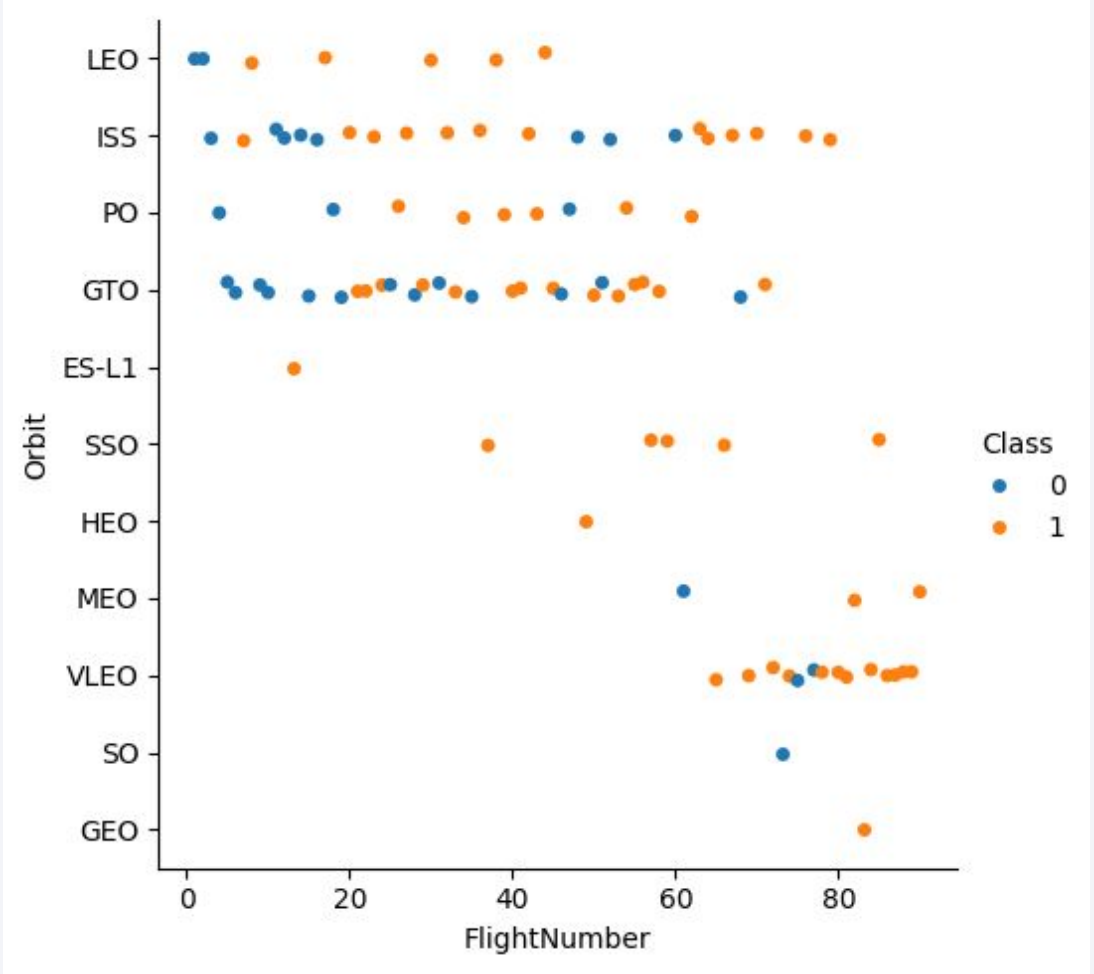
Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type
- Show the screenshot of the scatter plot with explanations



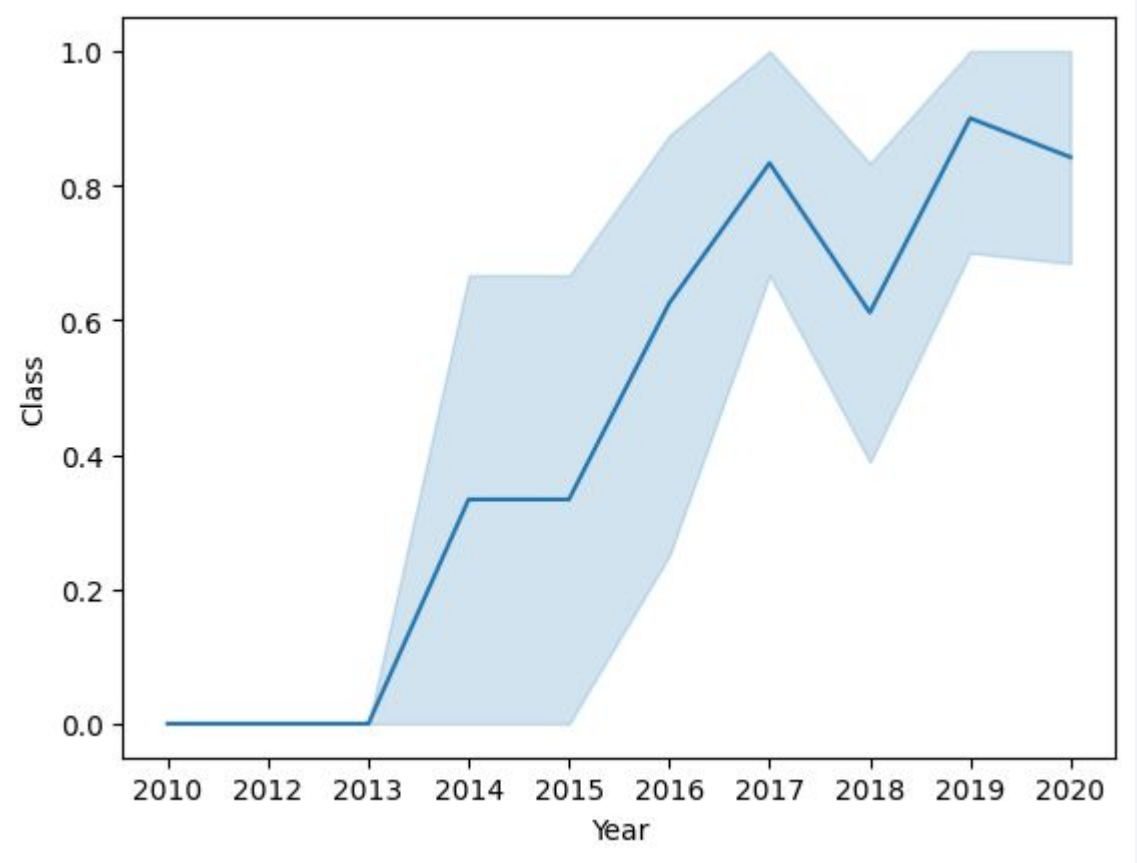
Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type
- Show the screenshot of the scatter plot with explanations



Launch Success Yearly Trend

- Show a line chart of yearly average success rate
- Show the screenshot of the scatter plot with explanations



All Launch Site Names

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

- The above launch site names were found using the SQL keyword DISTINCT
- The SQL command was:

```
select DISTINCT Launch_Site from SPACEXTABLE
```

Launch Site Names Begin with 'CCA'

- The following represent 5 records where the launch site name begins with ‘CCA’

Date	Launch_Site	Orbit
2010-06-04	CCAFS LC-40	LEO
2010-12-08	CCAFS LC-40	LEO (ISS)
2012-05-22	CCAFS LC-40	LEO (ISS)
2012-10-08	CCAFS LC-40	LEO (ISS)
2013-03-01	CCAFS LC-40	LEO (ISS)

- Note: not the whole record is shown, because that would be very messy
- The SQL command to retrieve this data was:

```
select Date, Launch_Site, Orbit from SPACEXTABLE WHERE
Launch_Site LIKE 'CCA%' LIMIT 5
```

Total Payload Mass

- The total payload carried by boosters from NASA (CRS) is 45 596 kg
- This was determined using the following SQL command:

```
select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE WHERE
Customer == "NASA (CRS)"
```

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1 is 2928.4 kg.
- This was determined using the following SQL command:

```
select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE WHERE
Booster_Version == "F9 v1.1"
```

First Successful Ground Landing Date

- The date of the first successful landing outcome on ground pad is 2015-12-22

- This was determined using the following SQL command:

```
select Date from SPACEXTABLE WHERE Landing_Outcome ==
'Success (ground pad)' order by Date LIMIT 1;
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- The following lists the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

Booster	Landing Outcome	Payload Mass (kg)
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

- This was determined using the following SQL command:

```
select Booster_Version, Landing_Outcome,
PAYLOAD_MASS__KG_ from SPACEXTABLE  WHERE
Landing_Outcome == 'Success (drone ship)' and
PAYLOAD_MASS__KG_>4000 and PAYLOAD_MASS__KG_<6000;
```

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

Mission successes and failures	
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

- This was determined using the following SQL command, and slightly modified manually:

```
select Mission_Outcome, count(Mission_Outcome) from
SPACEXTABLE group by Mission_Outcome
```

Boosters Carried Maximum Payload

- This table lists the names of the boosters which have carried the maximum payload mass
- This was determined using the following SQL command:

```
SELECT Booster_Version FROM SPACEXTABLE WHERE
PAYLOAD_MASS__KG_ == (SELECT
MAX(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

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

- This list shows the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

Month	Landing Outcome	Booster Version	Launch Site
1	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
4	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- This was determined using the following SQL command:

```
select substr(Date,6,2) as 'Month', Landing_Outcome,
Booster_Version, Launch_Site from SPACEXTABLE where
substr(Date,1,4) == '2015' and Landing_Outcome ==
'Failure (drone ship)'
```

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

- The table ranks the count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order

Number of landing outcomes	
Success (drone ship)	12
No attempt	12
Success (ground pad)	8
Failure (drone ship)	5
Controlled (ocean)	4
Uncontrolled (ocean)	2
Precluded (drone ship)	1

- This was determined using the following SQL command:

```
select Landing_Outcome, count(*) as 'count' from
SPACEXTABLE where Date > '20100601' and Date <
'20170320' group by Landing_Outcome order by count
desc
```

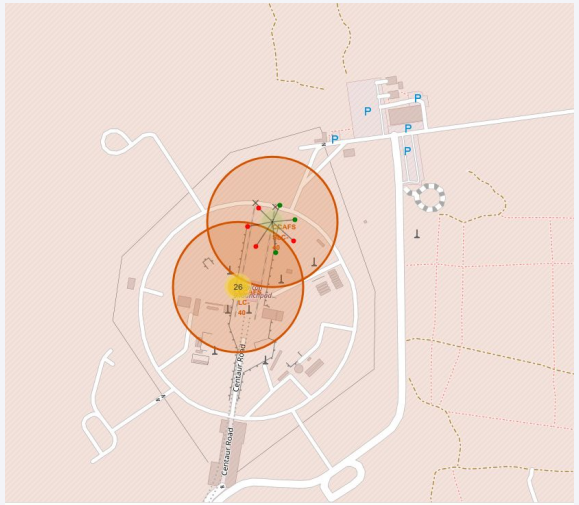

Section 3
Launch Sites
Proximities Analysis

Launch site Locations

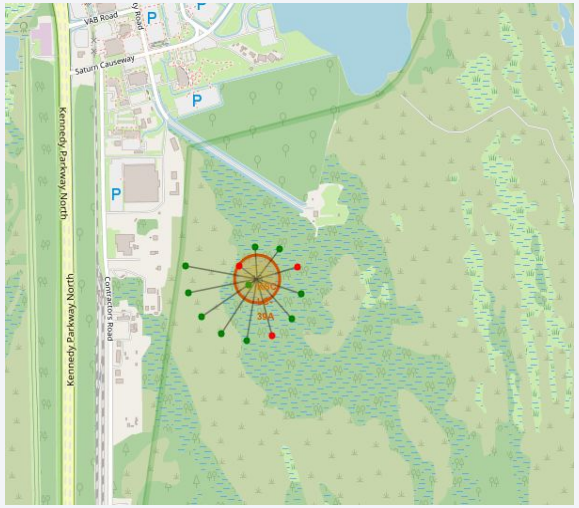
- There are 4 launch sites:
  - 1 in California
  - 3 in Florida

Success rate by location

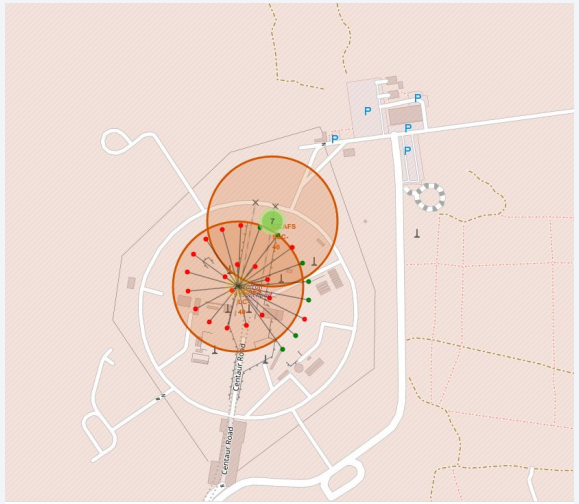
CCAF SLC-40



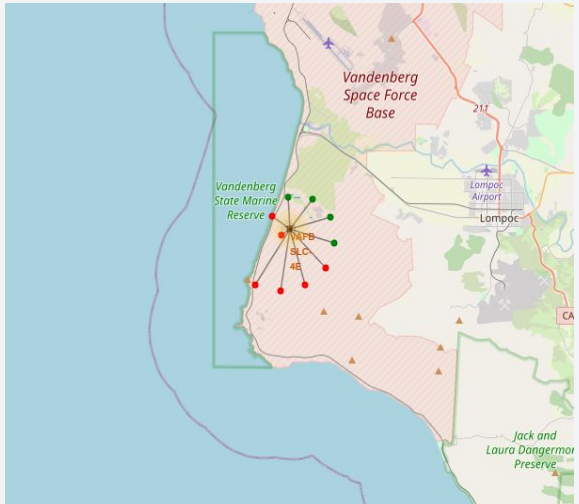
KSC LC-39A



CCAFS LC-40



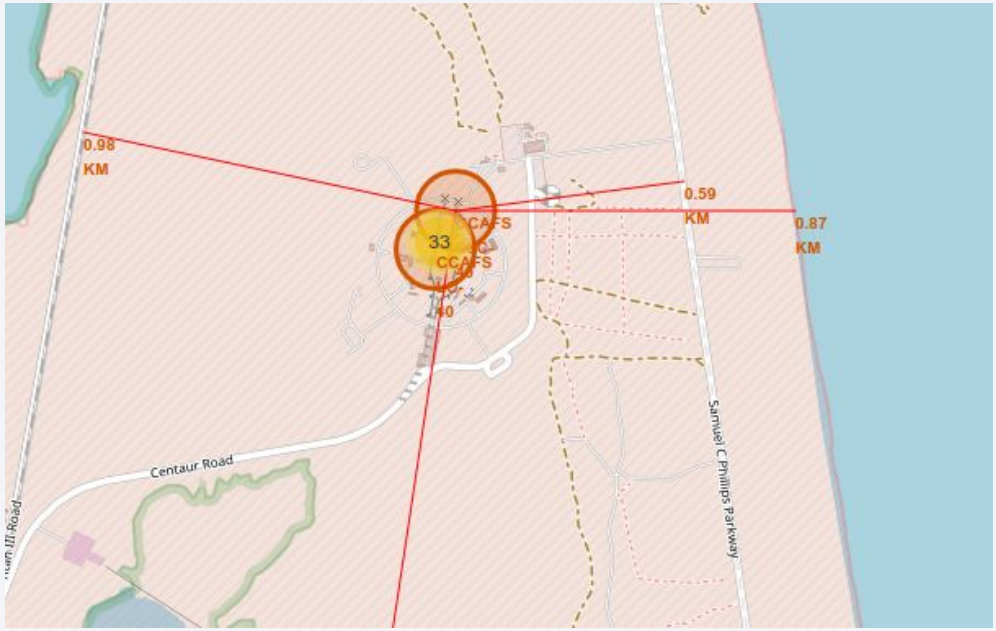
VAFB SLC-4E

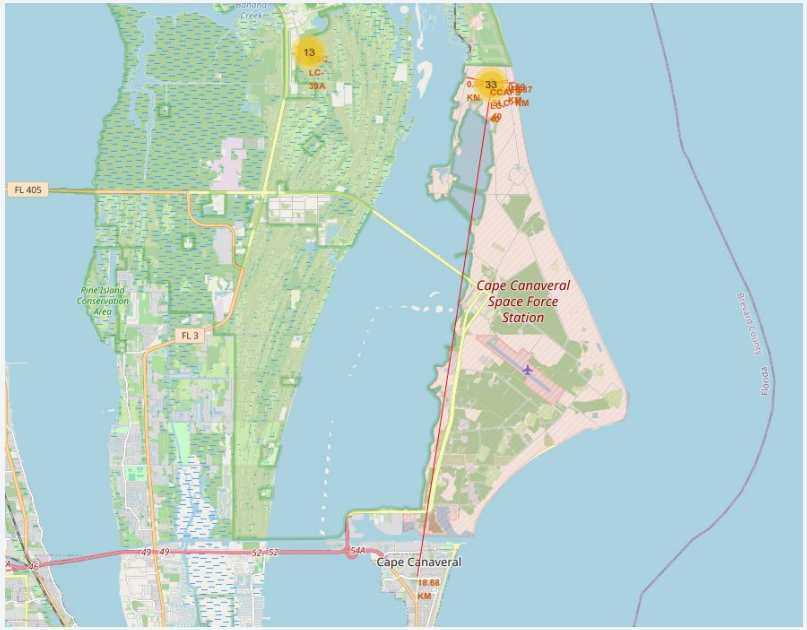


These maps illustrate the success rates of the landings per launch site location.

Green dots show successful landings and red dots show unsuccessful landings

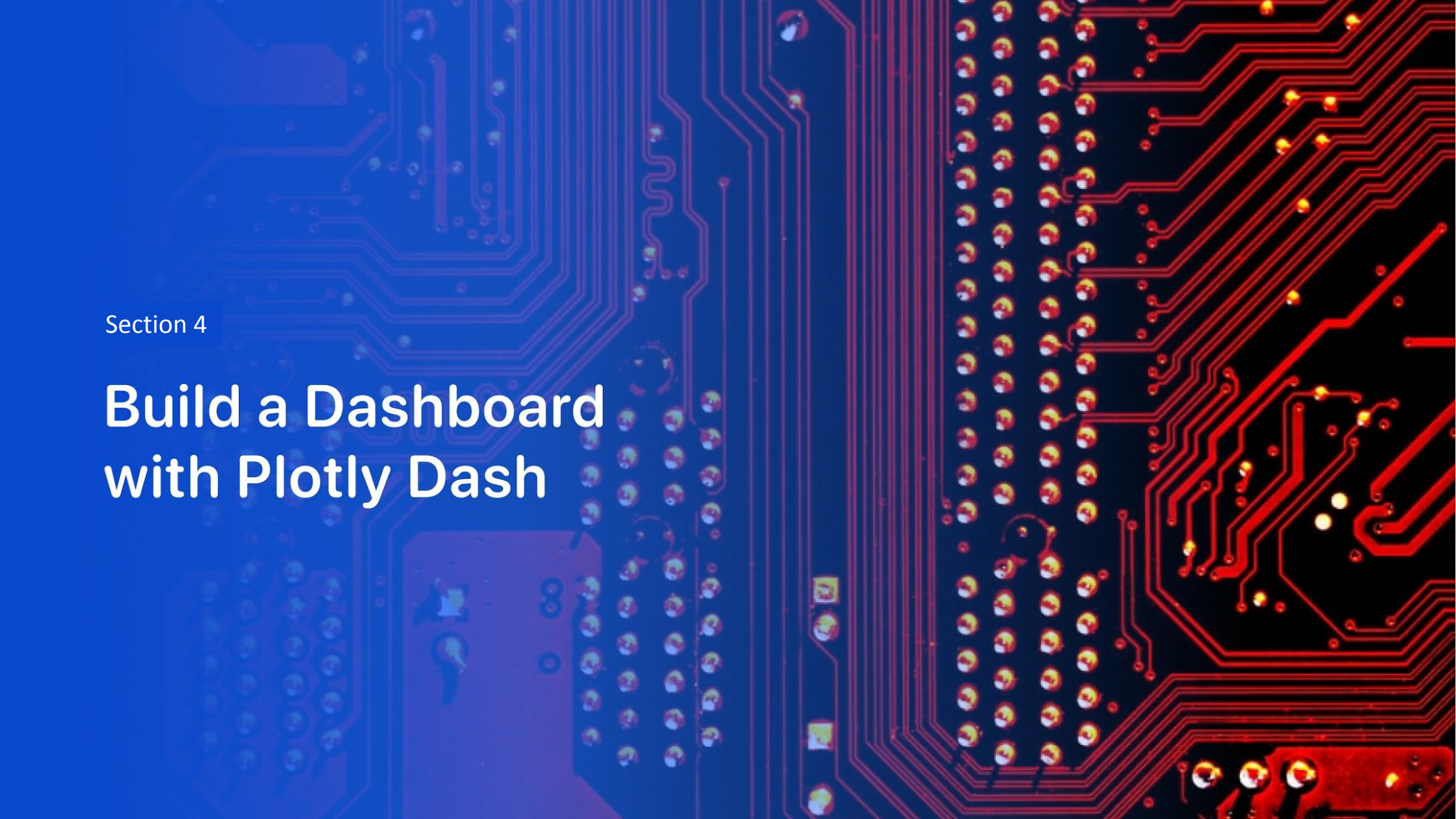
Launch site distance from coast, rail, highway and city





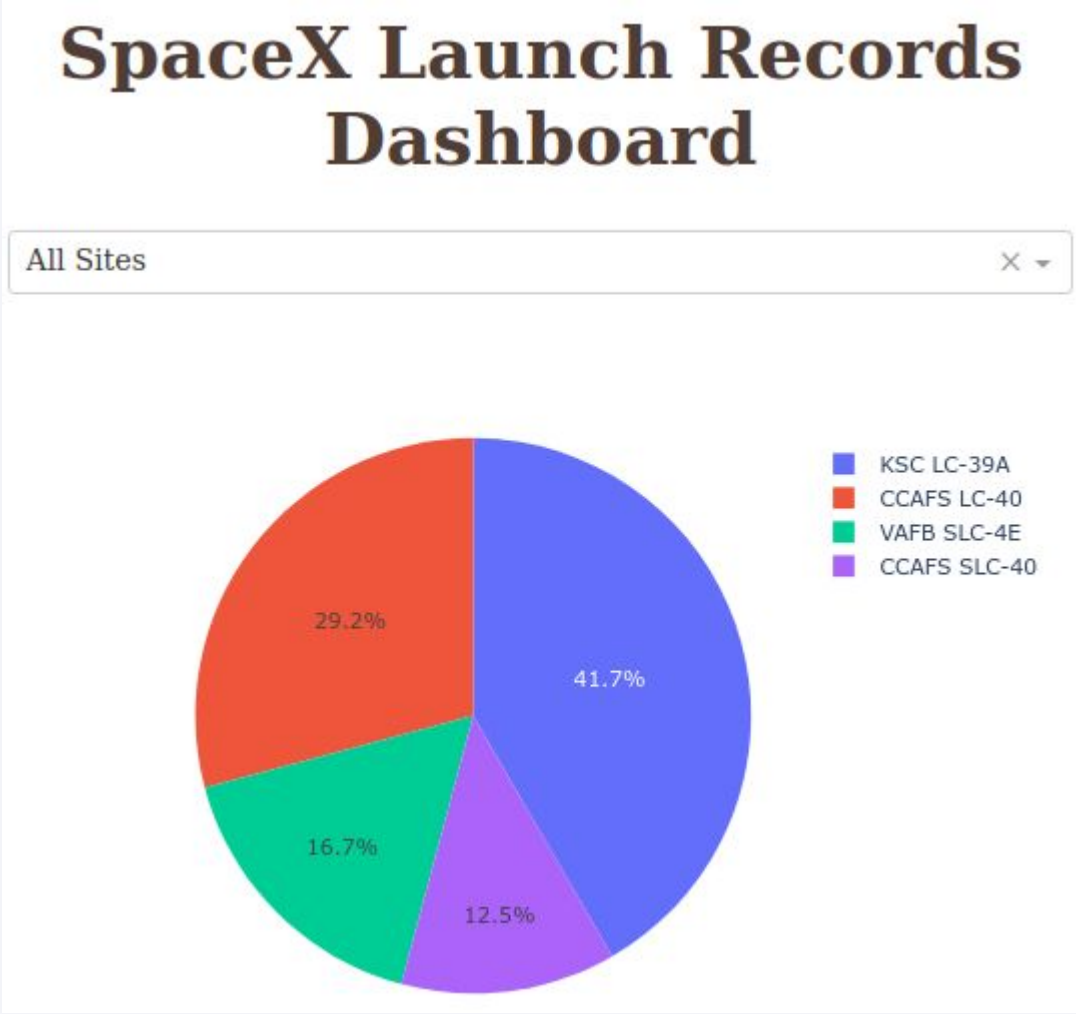
The maps above show the distances of CCAF SLC-40 from the nearest:

- Coast: 0.87 km
- Railroad: 0.98 km
- Highway: 0.59 km
- City: 18.69 km


Section 4
Build a Dashboard with Plotly Dash

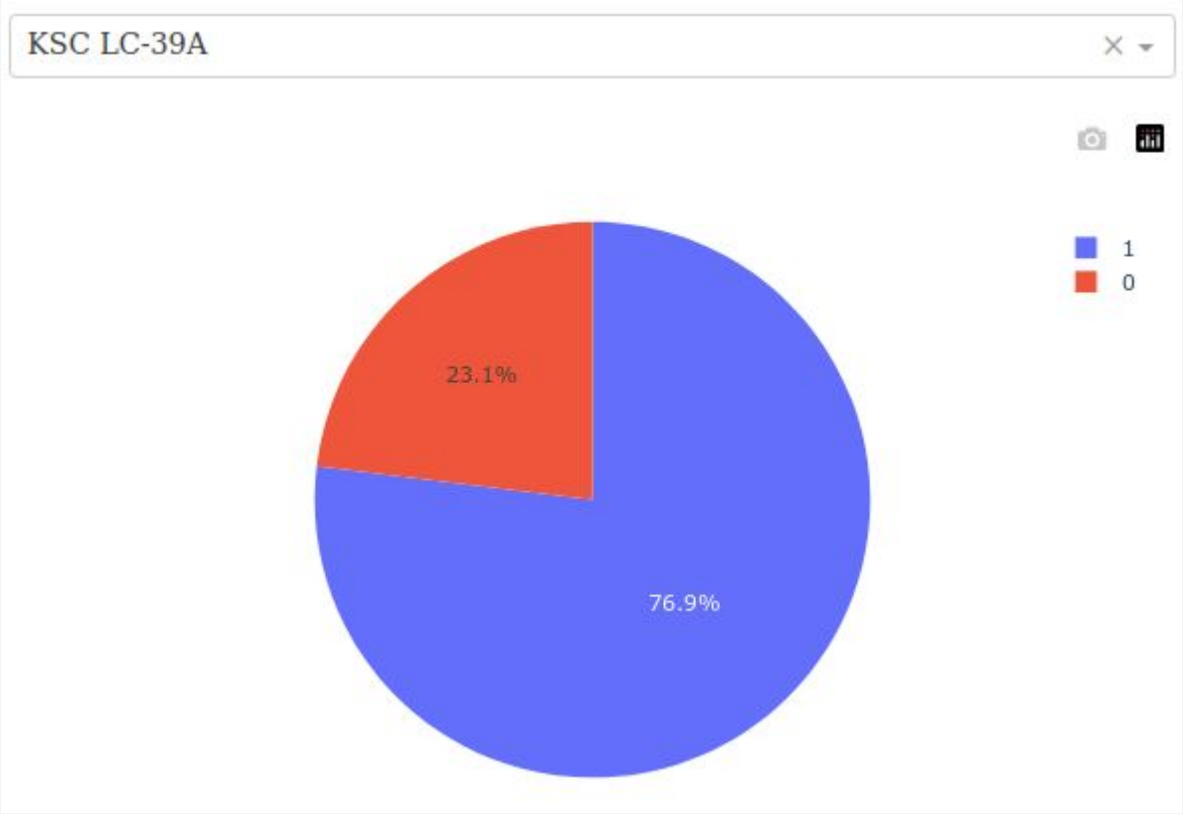
Number of flights with a successful landing outcome

- The pie chart shows the number of flights with a successful landing outcome per launch site.



Success rate for KSC LC-39A

- KSC LC-39A has the highest success rate.
- It has a success rate of 76.9%



Success Rate for a Payload Mass of 5000 to 7500 kg

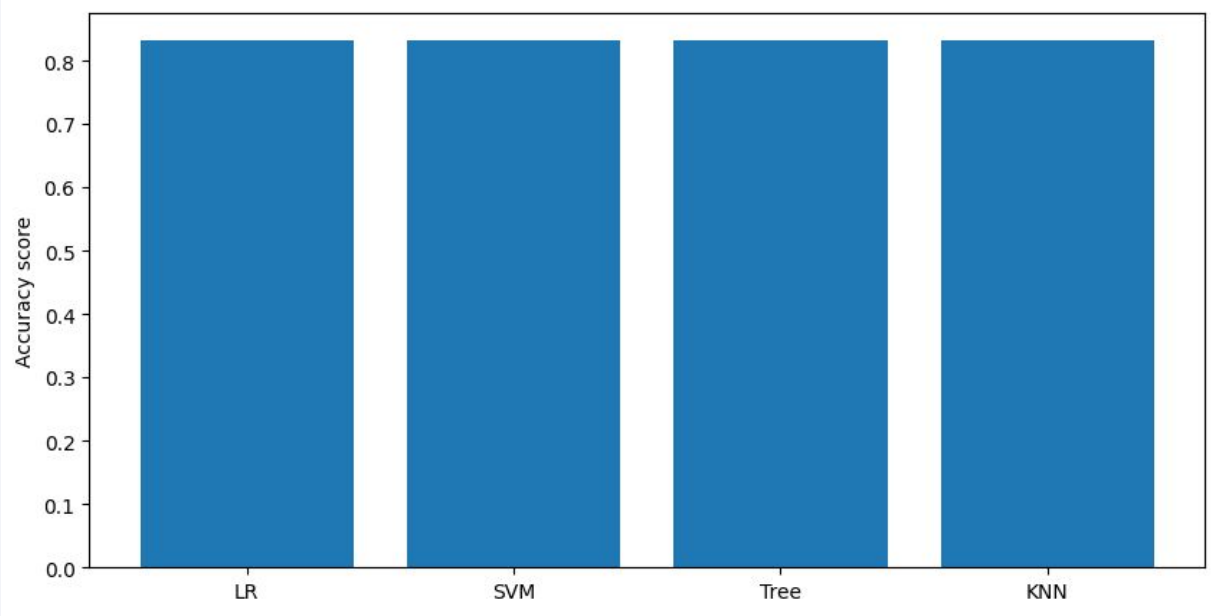
- The plot shows the landing outcome success rate for flights with a payload mass of between 5000 kg and 7500 kg.

Section 5

Predictive Analysis (Classification)

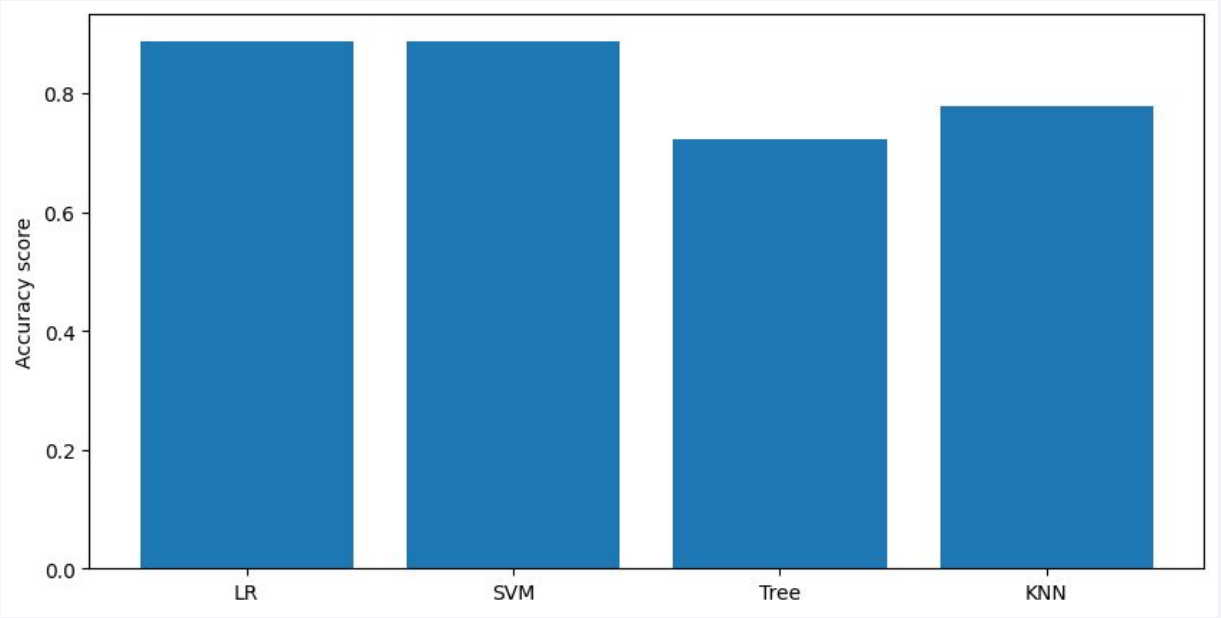
Classification Accuracy

- The chart shows the R2 score of each method we evaluated
- It seems that, based on the test data, they all perform equally well.
- Considering the nature of the data, abd how small the test data is this is actually possible
- However: considering that this is an exercise it seems highly unlikely that the exercise designer would create this situation.
- Did I do something wrong? Did other people get the same result as me?
- … I gues I'll find out soon.



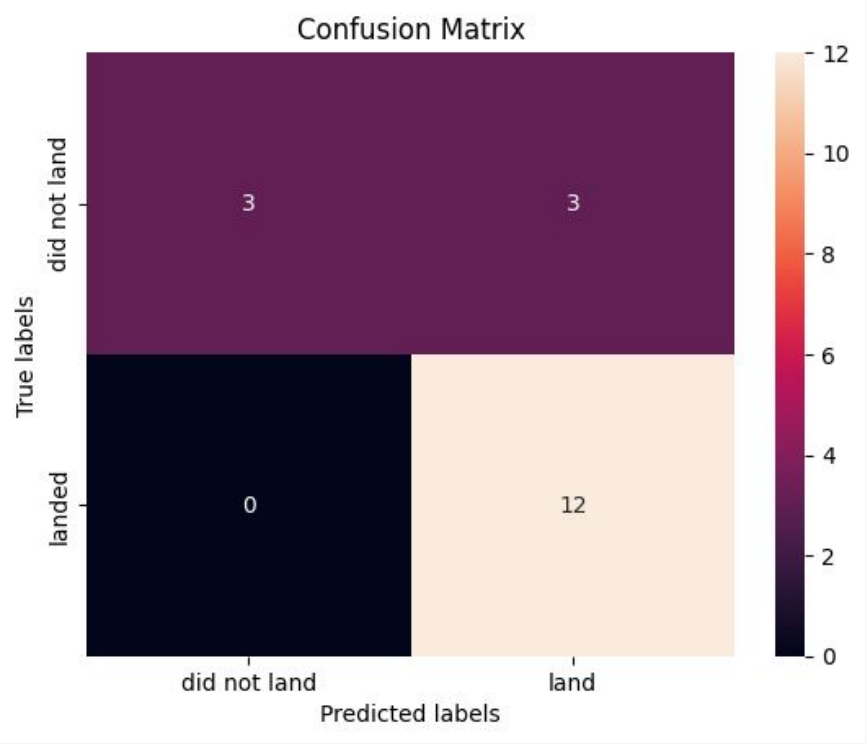
Classification Accuracy

- Incidentally:
  - Changing the `random_state` variable in the `train_test_split` results in significant differences in the scores.
  - The plot shows the result when using a `random_state` of 1 instead of 2 as specified in the exercise.
  - It would be interesting to see how sensitive these results are to changing the composition of the training and testing data sets.



Confusion Matrix

- Surprisingly, all models performed equally well and resulted in the same confusion matrix when evaluated on the test data.
  (See the previous 2 slides for an analysis)



- 3 landings were correctly predicted to fail (True Negative)
- 12 landings were correctly predicted to succeed (True Positive)
- 3 landings we incorrectly predicted to succeed (False Positive)
- 0 landings we incorrectly predicted to fail.

Conclusions

- It appears that all classification models performed equally well.
- A preliminary investigation suggests that the performance of the models is highly sensitive to which data points have been selected.
- Further investigation recommended.

Appendix

- Whole project github (link). In addition to the links in the previous slides, the project also includes:
  - Flowcharts (link)
  - Machine Learning random_state Investigation (link)

Thank you!