IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Federico Plazzotta
November 5th 2022

Outline

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

Executive Summary

- **Goal** of the analysis is to predict if the Falcon 9 first stage will land successfully.
- **Data collection** relied on data from public SpaceX API and on scrapping SpaceX Wikipedia page through Beautiful soup library
- **Methods** relied on Exploratory data analysis (EDA) with SQL, Pandas and MatplotLib and on the creation of labels column 'Class' which denotes all the successful landings.
- As **result** a Machine Learning Pipeline capable of predicting if the first stage of the Falcon9 will land given the data was developed with good accuracy score (83%)

Introduction

- Project is developed as final assignment of the Data Science professional Certificate and has the goal of predicting if the Falcon 9 first stage will land successfully, based on the outcome of previous launches, thus determining the cost of a SpaceX launch. This information is useful for an alternate company, SpaceY, which wants to bid against SpaceX for a rocket launch.
- Answer about the probability of success of landing of first stage for launches from different sites is expected

Section 1

Methodology

Methodology

Executive Summary

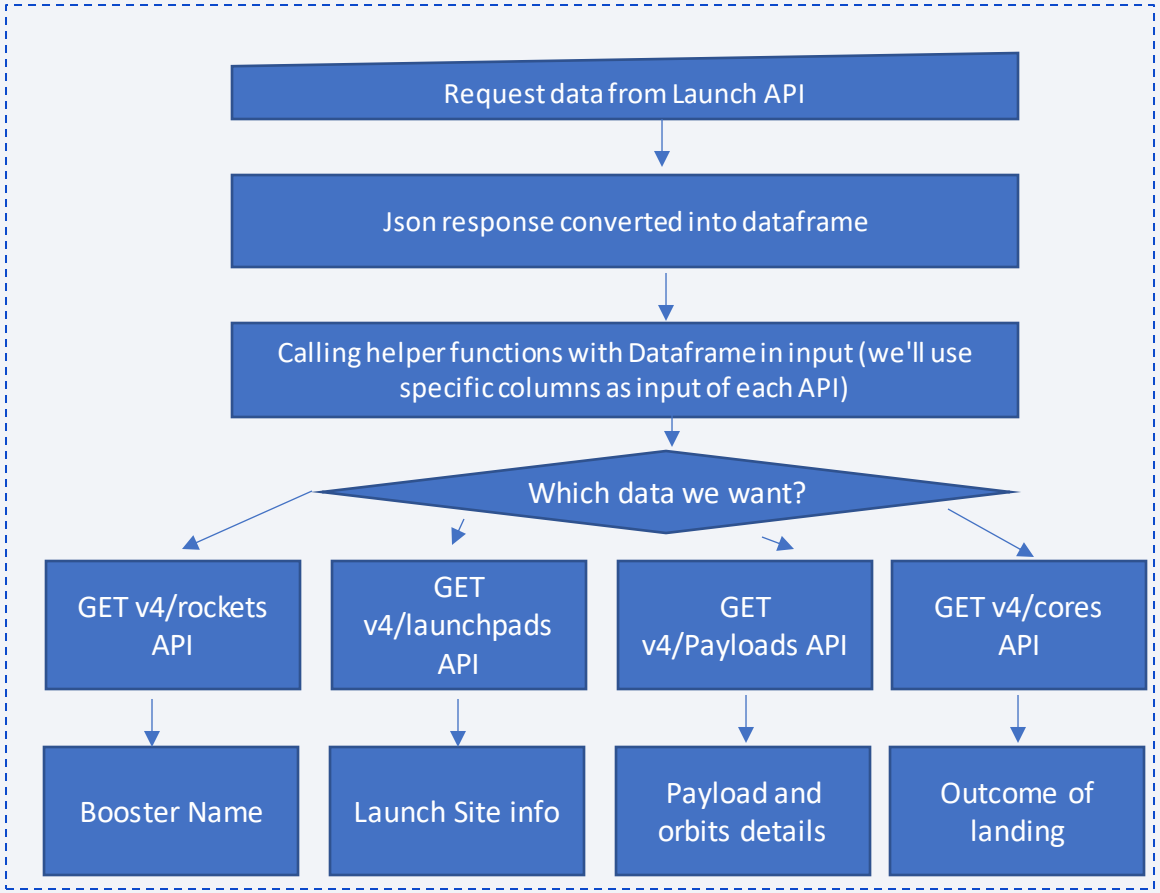
- Data collection methodology:
  - Data was collected from public SpaceX API and on scrapping SpaceX Wikipedia page through Beautiful Soup library
- Perform data wrangling
  - Using Pandas methods data were wrangled to obtain the "outcome" field to represent successful or unsuccessful landings
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - The outcome "Class" was isolated and data were standardized and split into training and test data, then best hyperparameters were evaluated for SVM, Classification Trees and Logistic Regression models using GridSearch from sklearn

Data Collection

- Describe how data sets were collected.
  - REST API Calls
  - WebScraping using BeautifulSoup (see next slides)
- You need to present your data collection process use key phrases and flowcharts

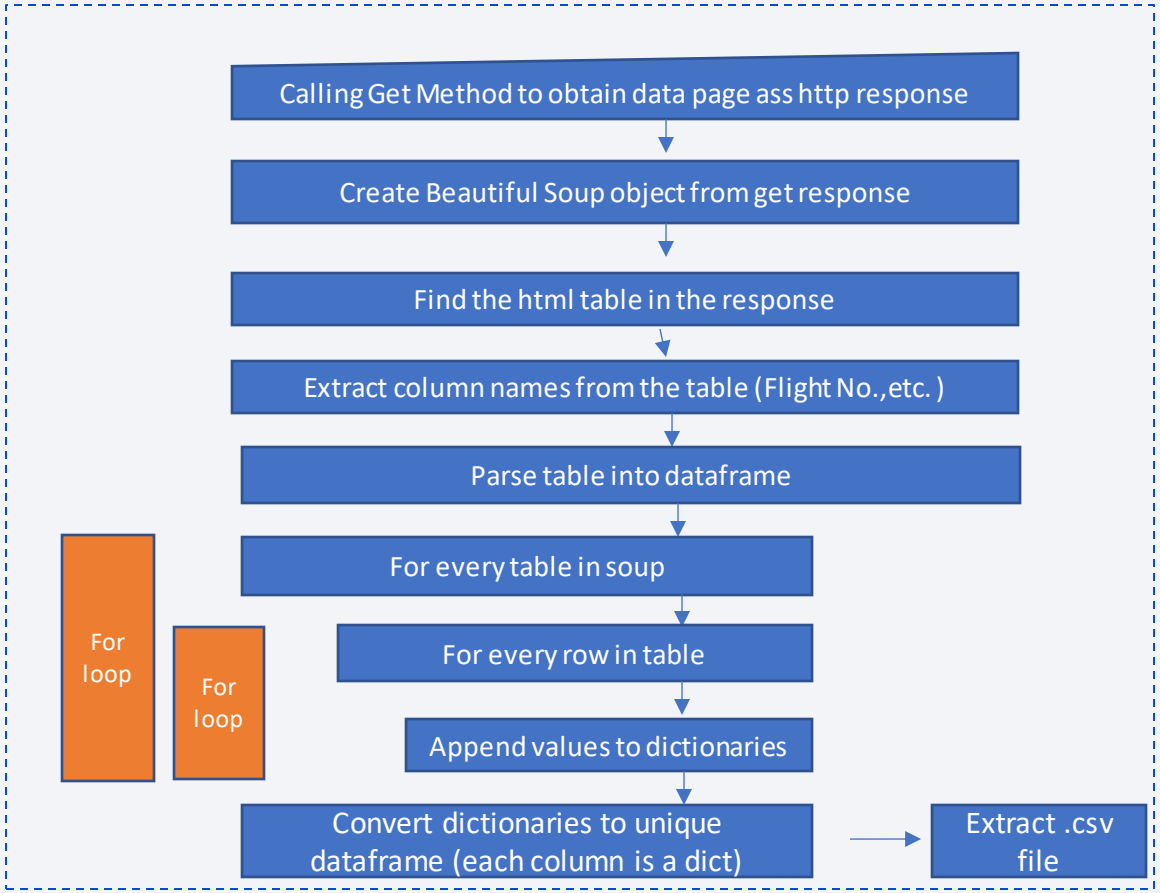
Data Collection – SpaceX API

- Data were collected initially by calling SpaceX GET Launch API:
  - GET /v4/launches/past
    the columns of the dataframe resulting from the response where used as inputs to call the following APIs
    - 1. Request data from rockets api (GET v4/rockets/) -->learn Booster name
    - 2.Requests from launchpads api (GET v4/launchpads/) -->learn launch site, longitude and latitude
    - 3.Requests from payload api (GET v4/payloads/) -->learn mass of payload and its orbit
    - 4. Requests from core api (GET v4/cores/) -->learn outcome of the landing, type of landing and type of core
- GitHub URL of the completed SpaceX API calls notebook
  [https://github.com/fedeplaz/CapstoneProject/blob/main/jupyter-labs-spacex-data-collection-api.ipynb]



Data Collection - Scraping

- Web scraping to collect Falcon9 historical launch records from Wikipedia page using Beautiful Soup
  - Get method to obtain data of page as http response
  - Beautiful Soup Object created from HTML response
  - Find the HTML table in the response
  - Extract column names from the table
  - Parse table into Pandas Dataframe
    - Find_all elements in table and save it into dictionary
    - Convert dictionaries containing each column into a Dataframe
- GitHub URL of the completed web scraping notebook *[https://github.com/fedeplaz/CapstoneProject/blob/main/Week1-jupyter-labs-webscraping.ipynb]*



Data Wrangling

- Describe how data were processed
  - Data were processed using Pandas and Numpy python libraries
  - Loading .csv file with data into a DataFrame and performing preliminary Data Analysis with methods such as:
    - Isnull
    - Dtypes
    - Value_counts
  - Create landing outcome label to represent successful or unsuccessful landings

- You need to present your data wrangling process using key phrases and flowcharts
- GitHub URL of completed data wrangling related notebooks *[https://github.com/fedeplaz/CapstoneProject/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb]*

EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
  - Catplot 1: Class over Flight Number and Payload Mass to check that landing is more likely to be successful after lots of flight numbers and with lower payload mass
  - Catplot 2: Class over Flight Number and Launch Site to check that some sites required more launch tentative than others to become successful
  - Catplot 3: Class over Payload Mass and Launch Site to check that some sites did not launch heavy payloads
  - Bar chart: Success rate vs orbit, to show that launches in some orbit are more successful than others (HEO and SSO are the most successful)
  - Catplot 4: Class over Orbit and Flight Number to check that the outcome for some orbits depends on experience (flight number) while for other orbits not (GTO orbit)
  - Catplot 5 : Class over Orbit and Payload Mass to check for heavy payloads some orbits are more likely to succeed than others (Polar, Leo and ISS)
  - Line Plot: average success rate over year, to show that success rate kept increasing
- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose
  [https://github.com/fedeplaz/CapstoneProject/blob/main/jupyter-labs-eda-dataviz.ipynb]

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed

1. ***Display the names of the unique launch sites in the space mission***
   %sql select distinct("Launch_Site") from "SPACEXTBL";

2. ***Display 5 records where launch sites begin with the string 'CCA'***
   %sql SELECT * FROM "SPACEXTBL" WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;

3. ***Display the total payload mass carried by boosters launched by NASA (CRS)***
   %sql SELECT SUM("PAYLOAD_MASS__KG_") FROM "SPACEXTBL" WHERE "Customer" LIKE 'NASA (CRS)'

4. ***Display average payload mass carried by booster version F9 v1.1***
   %sql SELECT AVG("PAYLOAD_MASS__KG_") FROM "SPACEXTBL" WHERE "Booster_Version" LIKE 'F9 v1.1%'

5. ***List the date when the first succesful landing outcome in ground pad was acheived.***
   %sql SELECT MIN("Date") FROM "SPACEXTBL" WHERE "Mission_Outcome" LIKE 'Success'
6. ***List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000***
   %sql SELECT DISTINCT("Booster_Version") FROM "SPACEXTBL" WHERE "Mission_Outcome" LIKE 'Success' AND "PAYLOAD_MASS__KG_" BETWEEN 4000 and 5999
7. ***List the total number of successful and failure mission outcomes***
   %sql select "Mission_Outcome" as outcome, count(*) from "SPACEXTBL" group by outcome
8. ***List the names of the booster_versions which have carried the maximum payload mass. Use a subquery***
   %sql select "Booster_Version" from "SPACEXTBL" where "PAYLOAD_MASS__KG_" = (select MAX("PAYLOAD_MASS__KG_") from "SPACEXTBL")
9. ***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***
   %sql select substr(Date, 4, 2) as month, "Booster_Version", "Launch_Site" from "SPACEXTBL" where "LANDING _OUTCOME"= 'Failure (drone ship)' and substr(Date,7,4)='2015';
10. ***Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.***
    %sql SELECT "DATE","Landing _Outcome",count("Landing _Outcome")as LANDING_OUTCOME_COUNT,DATE from SPACEXTBL where substr(Date,7,4) || substr(Date,4,2) || substr(Date,1,2) between '20100604 and '20170320' group by "Landing _Outcome" order by count("Landing _Outcome") desc

- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose
  [https://github.com/fedeplaz/CapstoneProject/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb]

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Explain why you added those objects
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification)

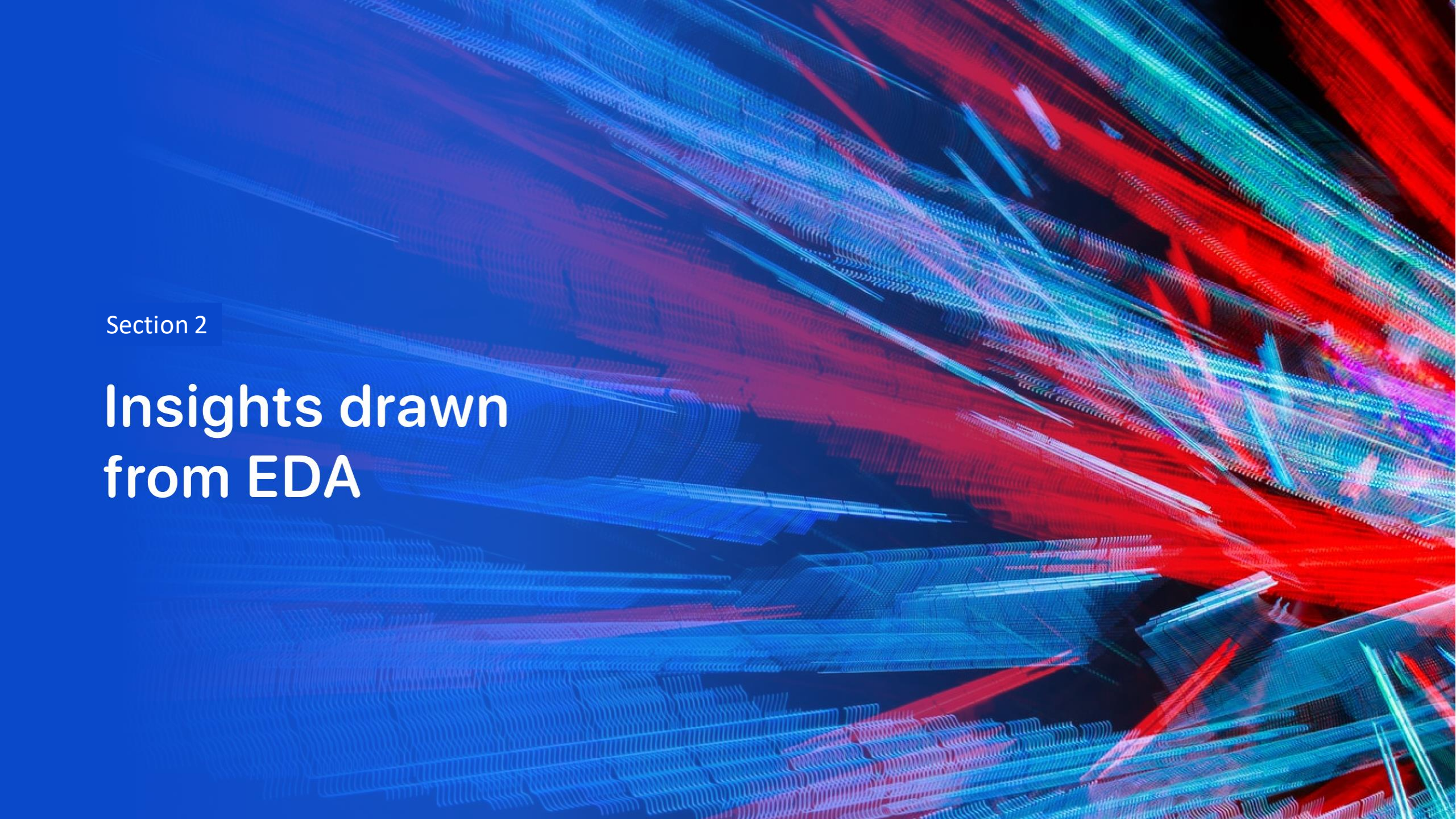
- Summarize how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

Results

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

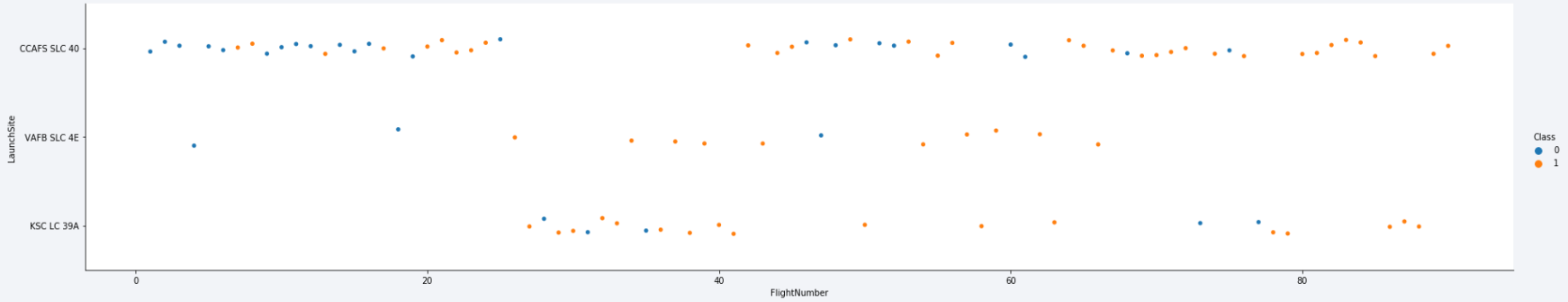
Section 2

Insights drawn from EDA



Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site

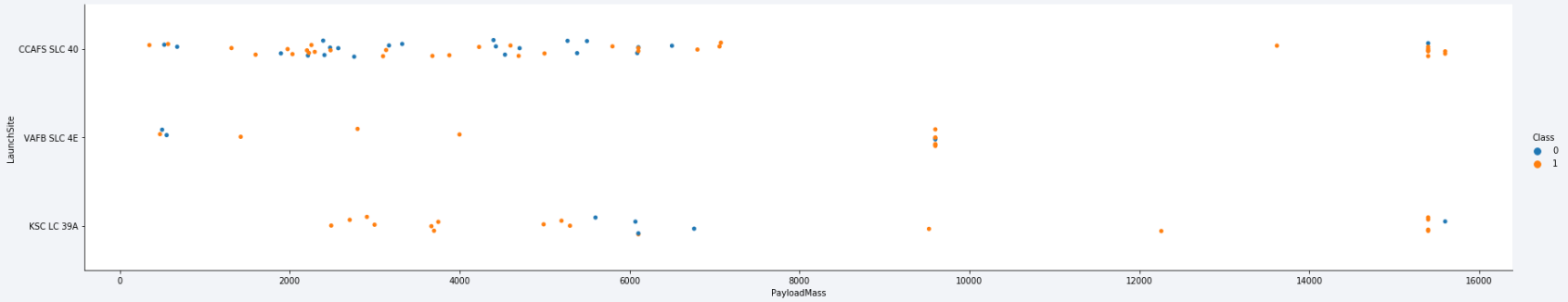


- Show the screenshot of the scatter plot with explanations

Landing "Class" (1 = successful) for different Flight Number and Launch Site, to check that some sites required more launch tentative than others to become successful, but that overall the percentage of success increased with flight number

Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site



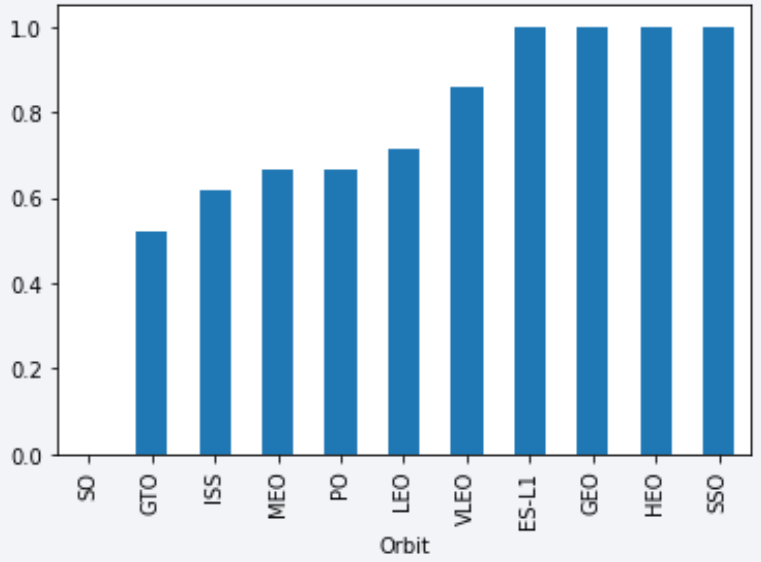
- Show the screenshot of the scatter plot with explanations

Landing "Class" (1 = successful) for different Launch Site and Payload Mass, to check that in some sites heavy payloads launches were not performed

Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type

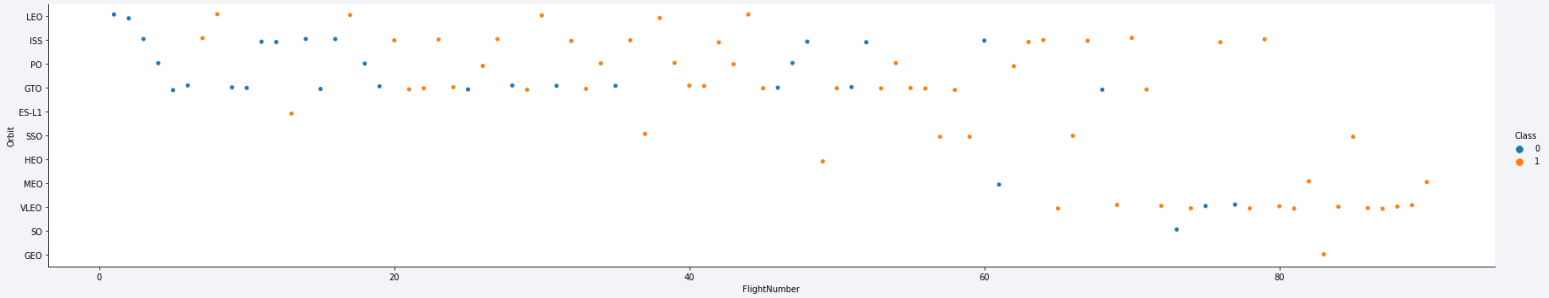
- Show the screenshot of the scatter plot with explanations



Success rate for different orbit types ordered by success rate (maximum = 1). As you can see HEO and SSO orbits guaranteed the highest success rate

Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type

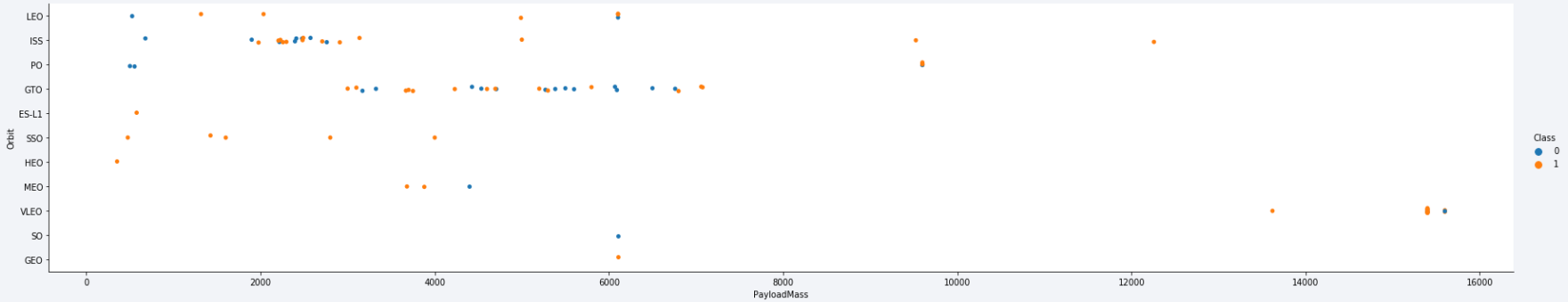


- Show the screenshot of the scatter plot with explanations

Landing "Class" (1 = successful) for different Flight Number and Orbits, to check that the outcome for some orbits depends on experience (flight number) while for other orbits not (GTO orbit)

Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type



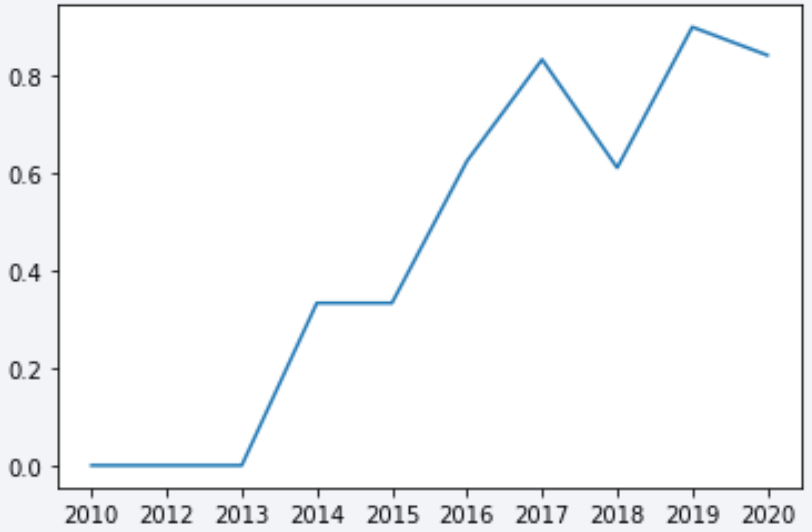
- Show the screenshot of the scatter plot with explanations

Landing "Class" (1 = successful) for different Orbit and Payload Mass, to check that for heavy payloads some orbits are more likely to succeed than others (Polar, Leo and ISS)

Launch Success Yearly Trend

- Show a line chart of yearly average success rate

- Show the screenshot of the scatter plot with explanations



average landing success rate over year (maximum = 1), to show that success rate kept increasing over time

All Launch Site Names

- Find the names of the unique launch sites
- Present your query result with a short explanation here

```
%sql select distinct("Launch_Site") from "SPACEXTBL";
```

Distinct returns the unique names of the particular column "Launch_Site"

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Present your query result with a short explanation here

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

Thanks to wildcard "%" we are selecting every item starting with CCA and limiting the results to the first 5 occurrences

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here

```
%sql SELECT SUM("PAYLOAD_MASS__KG_") FROM "SPACEXTBL" WHERE "Customer" LIKE 'NASA (CRS)'
```

Summing all the values in the payload column, if the "Customer field" is the required one. This leads to the total payload carried by alllaunches performed by that customer

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here

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

Calculate the average on the fields containing booster version F9 v1.1

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here

```
%sql SELECT MIN("Date") FROM "SPACEXTBL" WHERE "Mission_Outcome" LIKE 'Success'
```

Return the minimum date (the oldest) of the occurrences where mission outcome is successful

Successful Drone Ship Landing with Payload between 4000 and 6000

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

- Present your query result with a short explanation here

```
%sql SELECT DISTINCT("Booster_Version") FROM "SPACEXTBL" WHERE "Mission_Outcome" LIKE 'Success' AND "PAYLOAD_MASS__KG_" BETWEEN 4000 and 5999
```

Select only the distinct names of boosters where the outcome is "Success",and filter by payload mass less than 6000 (5999 included) and greater than 4000

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here

%sql select "Mission_Outcome" as outcome, count(*) from "SPACEXTBL" group by outcome

Count all the occurrences of the "Mission_outcome" field, rename such field as "outcome" and group the anwers according to their value

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

```
%sql select "Booster_Version" from "SPACEXTBL" where "PAYLOAD_MASS__KG_" = (select
MAX("PAYLOAD_MASS__KG_") from "SPACEXTBL" )
```

Returning the booster version matching the highest value in payload mass using a subquery

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

- Present your query result with a short explanation here

```
%sql select substr(Date, 4, 2) as month, "Booster_Version", "Launch_Site" from "SPACEXTBL" where "LANDING
_OUTCOME"= 'Failure (drone ship)' and substr(Date,7,4)='2015';
```

Selecting the Month out of the date field and the launch site where the outcome of the landing was failure, and filter only by those happening in the year 2015

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

- 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

- Present your query result with a short explanation here

```
%sql SELECT "DATE","Landing_Outcome",count("Landing_Outcome")as LANDING_OUTCOME_COUNT,DATE from SPACEXTBL where substr(Date,7,4) || substr(Date,4,2) || substr(Date,1,2) between '20100604 and '20170320' group by "Landing_Outcome" order by count("Landing_Outcome") desc
```

Select date, outcome and its count where year month day is between those specified in the requirements. Then group by the landing outcome (to show outcome and its count) an order descending


Section 3
Launch Sites
Proximities Analysis

<Folium Map Screenshot 1>

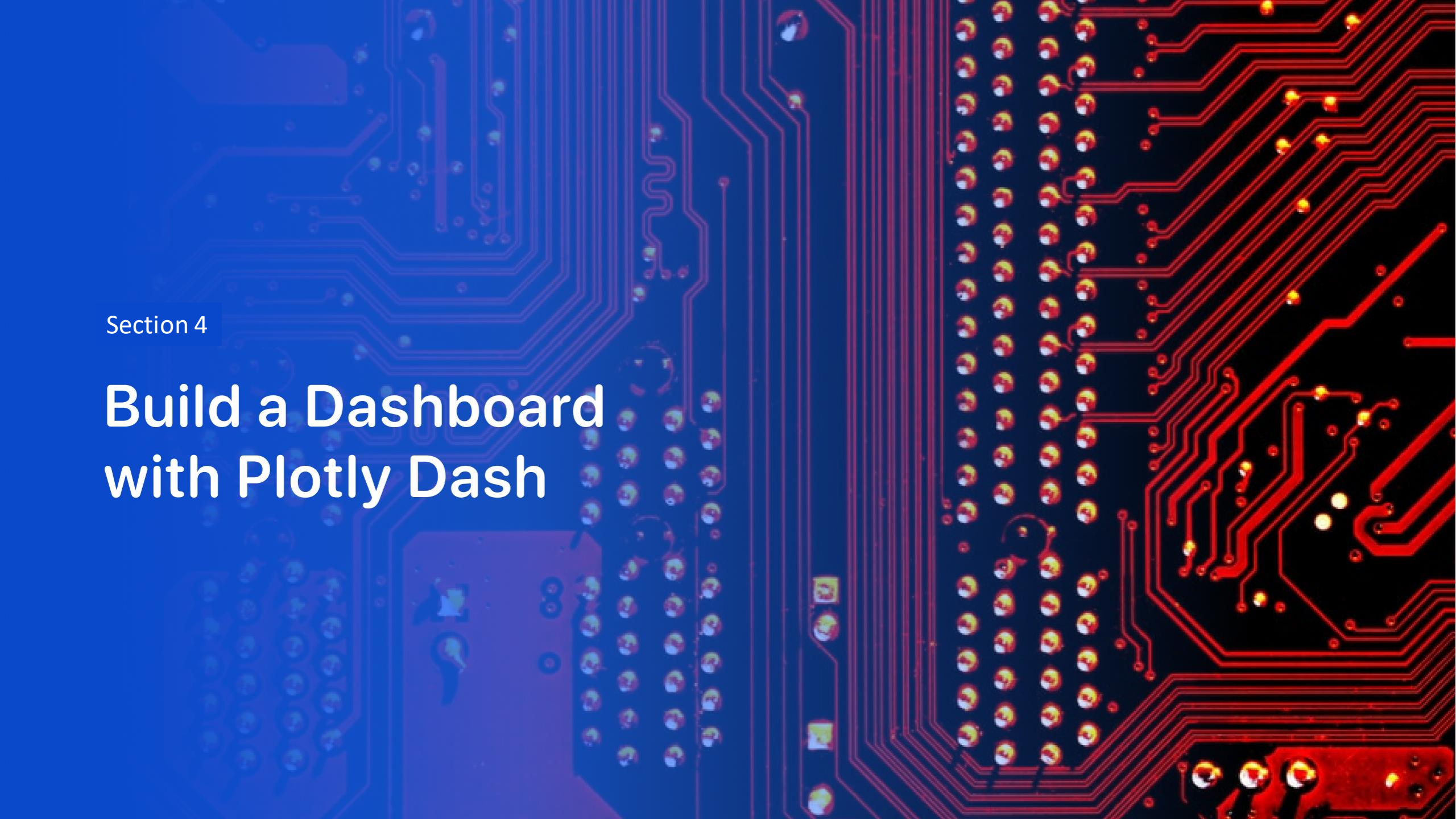
- Replace <Folium map screenshot 1> title with an appropriate title
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 2>

- Replace <Folium map screenshot 2> title with an appropriate title
- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map
- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 3>

- Replace <Folium map screenshot 3> title with an appropriate title
- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed
- Explain the important elements and findings on the screenshot


Section 4
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

- Replace <Dashboard screenshot 1> title with an appropriate title
- Show the screenshot of launch success count for all sites, in a piechart
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 2>

- Replace <Dashboard screenshot 2> title with an appropriate title
- Show the screenshot of the piechart for the launch site with highest launch success ratio
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 3>

- Replace <Dashboard screenshot 3> title with an appropriate title
- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider
- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.

Section 5

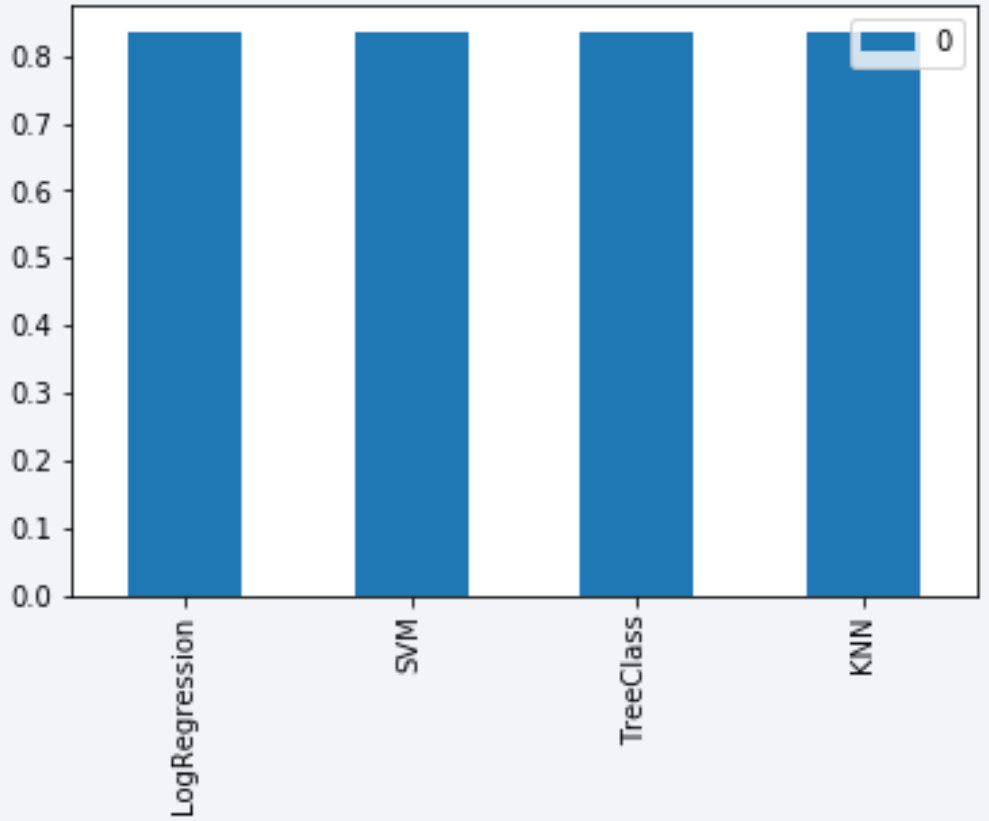
Predictive Analysis (Classification)

Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart

- Find which model has the highest classification accuracy

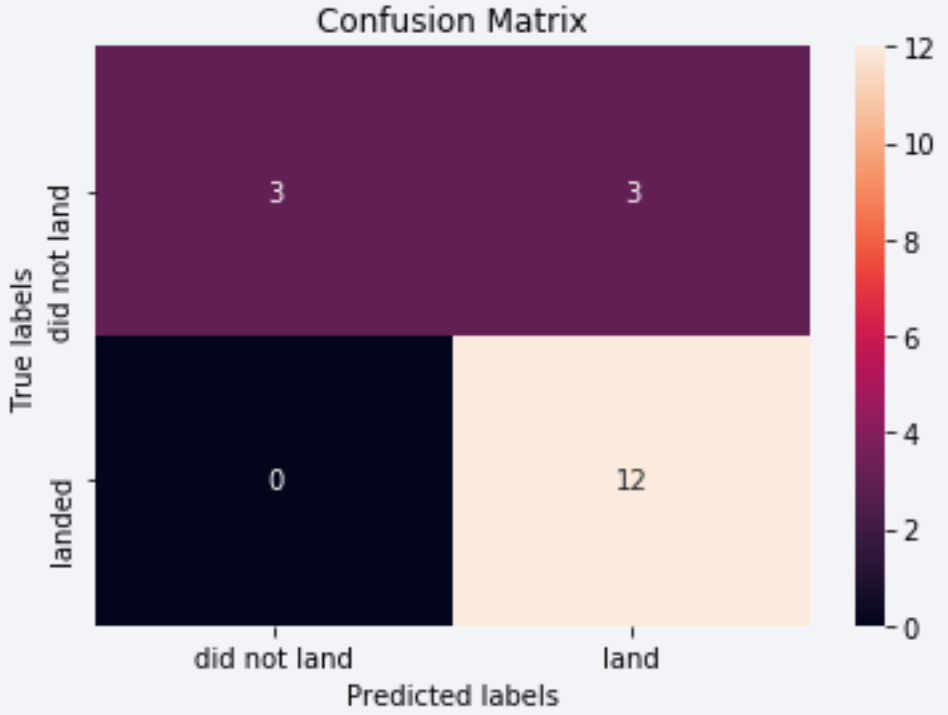
All models have the same accuracy of roughly 83,3%



Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation

All models can distinguish between the different classes, the major problem is the presence of false positives. In fact, 3 did not land were predicted to be landed



Conclusions

- We could create several predictive models allowing to define if the landing will be successful with a accuracy of roughly 83%
- Our models are accurate but struggle a bit with false positives
- Percentage of success is affected by several factors, such as orbit, launch site payload and so on, all these factors are included in our models
- Percentage of successful landing increased over the years

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project
  - We could perform more analysis on the predictive models, such as enphasize the choices in tree classification models
  - We could consider the success rate over year in our predictive model, to estimate the amount of successful landings in the future (e.g. in 2025)

Thank you!