IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nicholas Vivenzi
April 6, 2024

Outline

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

Executive Summary

Summary of methodologies

- **Understanding the dataset** Data collection, Data wrangling
- **Exploratory Data Analysis** (EDA) using SQL
- **Interactive Data visualization** using Folium Plotly Dash
- **Predictive data analysis** developing ML classification model
- **Final report** presentation of the main results

Executive Summary

In this work:

- The SpaceX databases have been analyzed, using rest APIs and webscraping of the Wikipedia page
- **Predecitive Machine Learning** analysis has been performed on the databases, using 4 different classification methods:

✓ Logistic Regression
✓ Support Vector Machine
✓ Decision Tree Classifier
✓ K Nearest Neighbors

All the methods have given the same result with an accuracy predicition of **83%**

All the methods slightly **overestimate the successful landing rate**, more data are required to perform robust prediction

Introduction

- Project background and context

✓ **Space X** is the leading private company in commercial space travels

✓ **Main advantage**: cost reduction due first stage vector recover

✓ First stage recovery in Falcon 9 rocket landing can decrease the mission cost (165 mln $ → 62 mln $)

✓ Space Y company would like to be competitive in commercial space business

- Aim of this project

Predicting successful Falcon 9 **first stage recovery** using **machine learning models**

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Space X REST API & Webscraping SpaceX Wikipedia page
- Perform data wrangling
  - Historical launch data → label for supervise model training → “Outcome” column
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - After standardizing data, splitting it into training and test sets. Performing analysis according to different model types and finally tuning models using GridSearchCV

Data Collection

- **API**
  - ✓ Database source: Open source REST API for SPACE X
  - ✓ Main steps:
    1. Request the Database and 'translate' into a Python dictionary
    2. Focus only on Falcon 9 data
    3. Replace the missing values with the average of the other values
- **Webscraping**
  - ✓ Database source: Wikipedia page *List of Falcon 9 and Falcon heavy launches*
  - ✓ Main steps:
    1. Request the data from the URL Wikipage
    2. Extract column and variables names
    3. Conversion into a Pandas database

Data Collection – SpaceX API

- GitHub URL of the completed SpaceX API calls notebook (https://github.com/nicholasviv95/IBM_ML_Capstone_project_final_exam/blob/main/NV_lab_1.0_data_collection.ipynb), for peer-review purpose

Task 1
- Request SpaceX APIs → .json file
- Conversion to Pandas Dataframe
- Create a Python dictionary

Task 2
- Data filtering and wrangling
- Include only Falcon 9 data

Task 3
- Dealing with missing values
- Replace the missing values with their mean

Data Collection - Scraping

- GitHub URL of the completed web scraping notebook (https://github.com/nicholasviv95/IBM_ML_Capstone_project_final_exam/blob/main/NV_lab_1.1_webscraping.ipynb), for peer-review purpose

Task 1
- Request Falcon 9 launch Wiki page from URL
- Get the BeautifulSoup object

Task 2
- HTML table
- Extract all column and variables names

Task 3
- Parse the HTML table
- Create a dictionary/dataframe

Data Wrangling

- **Aim:** creating a database column to label the data for training supervised learning models
- Starting point: “Outcome” column of the database
- Final labels definition:
  - Label 0: the landing is failed or data incomplete
  - Label 1: the landing is successful

- GitHub URL of the completed data wrangling related notebook (https://github.com/nicholasviv95/IBM_ML_Capstone_project_final_exam/blob/main/NV_lab_1.2_data_wrangling.ipynb), for peer-review purpose

EDA with Data Visualization

- Python Libraries: matplotlib and seaborn
- Aim: generating scatterplots, pie charts, bar charts and line charts to develop a preliminary understanding of the data
- Plots shown in this presentation: Flight Number Vs Launch Site, Payload Vs Launch Site, Success Rate Vs Orbit Type, Flight Number Vs Orbit Type, Payload Vs Orbit Type, Launch Success Yearly Trend
- GitHub URL of the completed EDA with data visualization notebook (https://github.com/nicholasviv95/IBM_ML_Capstone_project_final_exam/blob/main/NV_lab_2.1_eda.ipynb), for peer-review purpose

EDA with SQL

- Databases have been queried using SQL Python integration, with the aim of preliminarily understanding the database
- Main query information required: Launch Site names, Launch Site Names Begin with 'CCA', Total Payload Mass, Average Payload Mass by Falcon 9 v 1.1, First successful ground landing date, Successful Drone Ship Landing with Payload between 4000 and 6000 kg, Total Number of Successful and Failure Mission Outcomes, Boosters Carried Maximum Payload, 2015 Launch Records and Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- GitHub URL of the completed EDA with SQL notebook (https://github.com/nicholasviv95/IBM_ML_Capstone_project_final_exam/blob/main/NV_lab_2.2_eda_sql.ipynb), for peer-review purpose

Build an Interactive Map with Folium

- Python library: Folium
- Aim: visualizing the locations of the launch sites, understanding the relationship between locations and landing success, understanding the reasons for the launch site choice
- All the launches have been marked with a color label showing the success/unsuccess of the launch
- Launch site distance calculated: coastline, cities, railways and highways
- GitHub URL of the completed interactive map with Folium map (https://github.com/nicholasviv95/IBM_ML_Capstone_project_final_exam/blob/main/NV_lab_3.1_Folium.ipynb), for peer-review purpose

Build a Dashboard with Plotly Dash

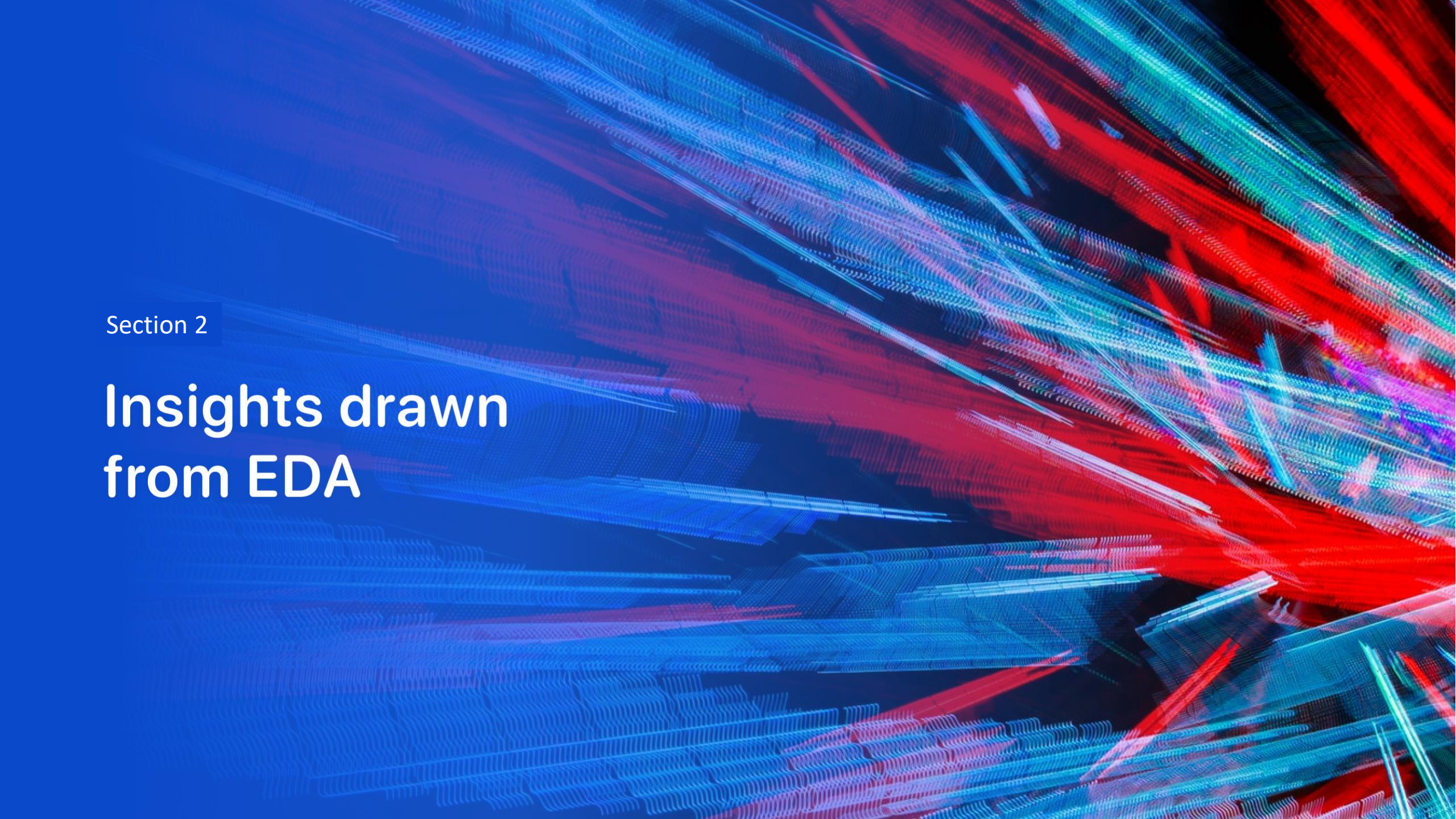
- Python libraries: matplotlib, seaborn
- Aim of this section: identifying the success rate for each launch site and understanding the relationship between success and payload mass
- Two pie charts and a scatter plot are shown
- GitHub URL of the completed Plotly Dash lab (https://github.com/nicholasviv95/IBM_ML_Capstone_project_final_exam/blob/main/NV_lab_3.2_dash.py), for peer-review purpose

Predictive Analysis (Classification)

- Python libraries: numpy, pandas, matplotlib, seaborn, sklearn
- After organizing the data, the database is split into training and test dataset
- Four different Machine Learning classification method:
  - ✓ Logistic Regression
  - ✓ Support Vector Machine
  - ✓ Decision Tree Classifier
  - ✓ K Nearest Neighbors Classifier
- GitHub URL of the completed predictive analysis lab (https://github.com/nicholasviv95/IBM_ML_Capstone_project_final_exam/blob/main/NV_lab_4_predictive_analysis.ipynb), for peer-review purpose

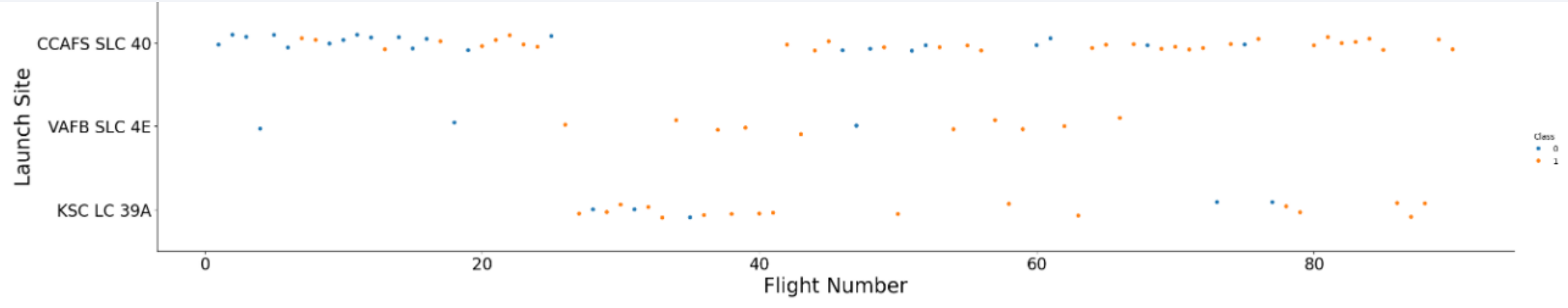
Results

- Exploratory data analysis results: Section 2
- Interactive analytics demo in screenshots: Sections 3-4
- Predictive analysis results: Section 5


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

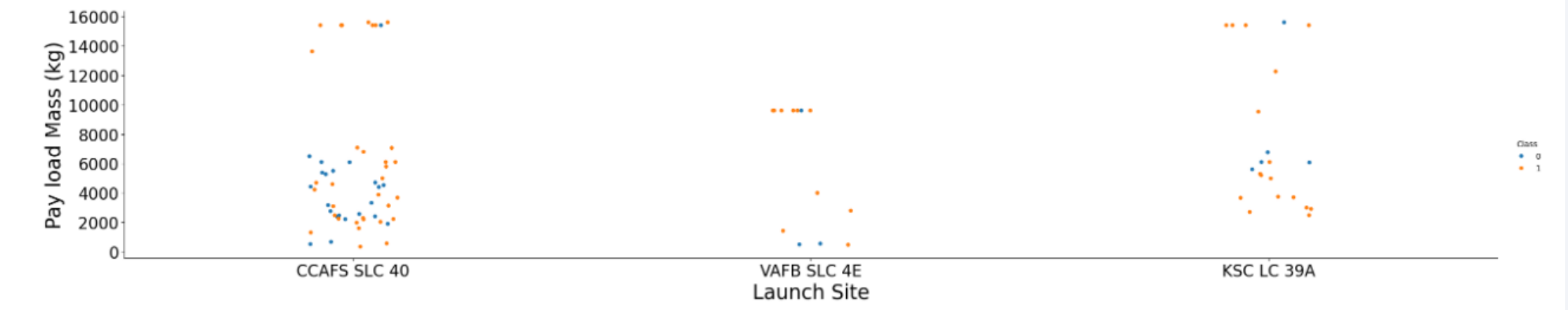
Launch sites Vs Flight Number scatter-plot, data split into failure (blue) and successful (orange) launches



- Three different launch sites have been exploited
- CCAFS SLC 40 is the most frequently chosen launch site, with the majority of failures (early launches) and successes (last launches)
- The frequency of successful launches increases over time (represented by the Flight Number)

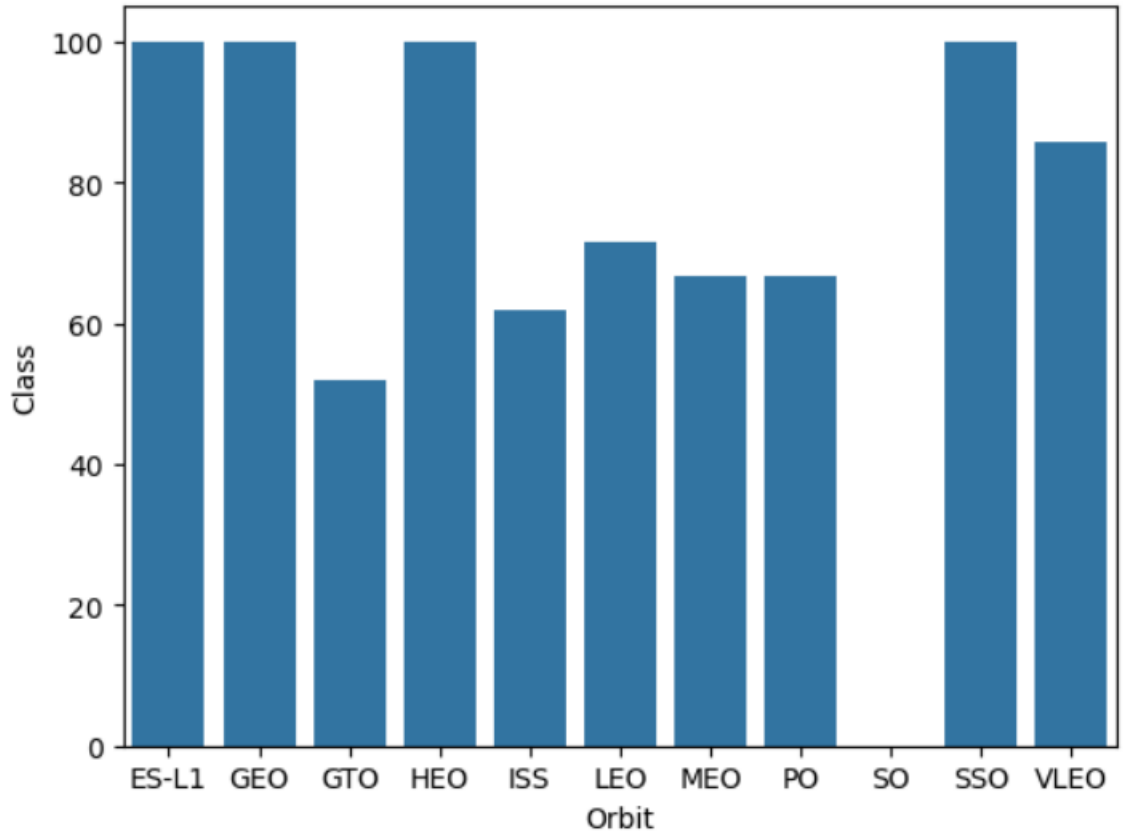
Payload vs. Launch Site

Payload mass Vs Launch sites scatter-plot, data split into failure (blue) and successful (orange) launches



- Successful launches rate is higher at low Payload Mass (< 8000 kg)
- Failures rate is higher at high Payload Mass (> 8000 kg)
- CCAFS SLC 40 has supported the widest interval in Payload Mass
- VAFB SLC 4E Launches payload mass has been limited to 10000 kg

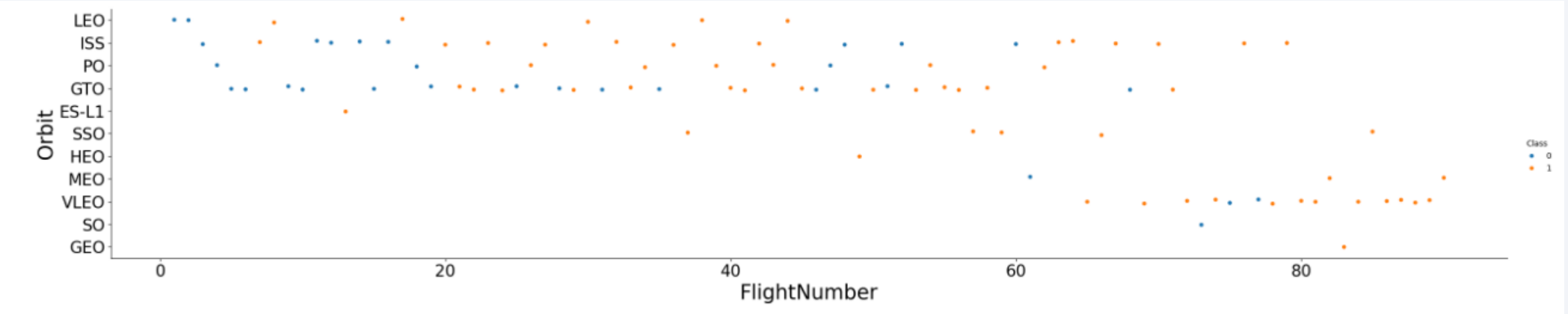
Success Rate vs. Orbit Type



- SO orbit has no successful landings
- All the other orbits have a successful rate higher than 50%
- ES-L1, GEO, HEO, SSO orbits have a 100% success rate landings

Flight Number vs. Orbit Type

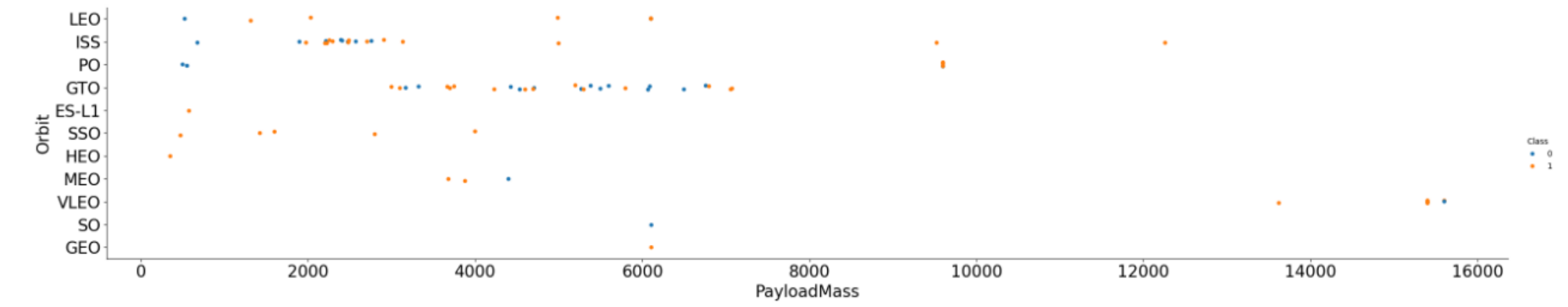
Orbit type Vs Flight Number scatter-plot, data split into failure (blue) and successful (orange) launches



- Different Orbits have been tested by Space X
- The preference has switched from LEO-ISS-PO-GTO to VLEO-SO-GEO
- The success rate increases in time, confirming the validity of the new orbits choice

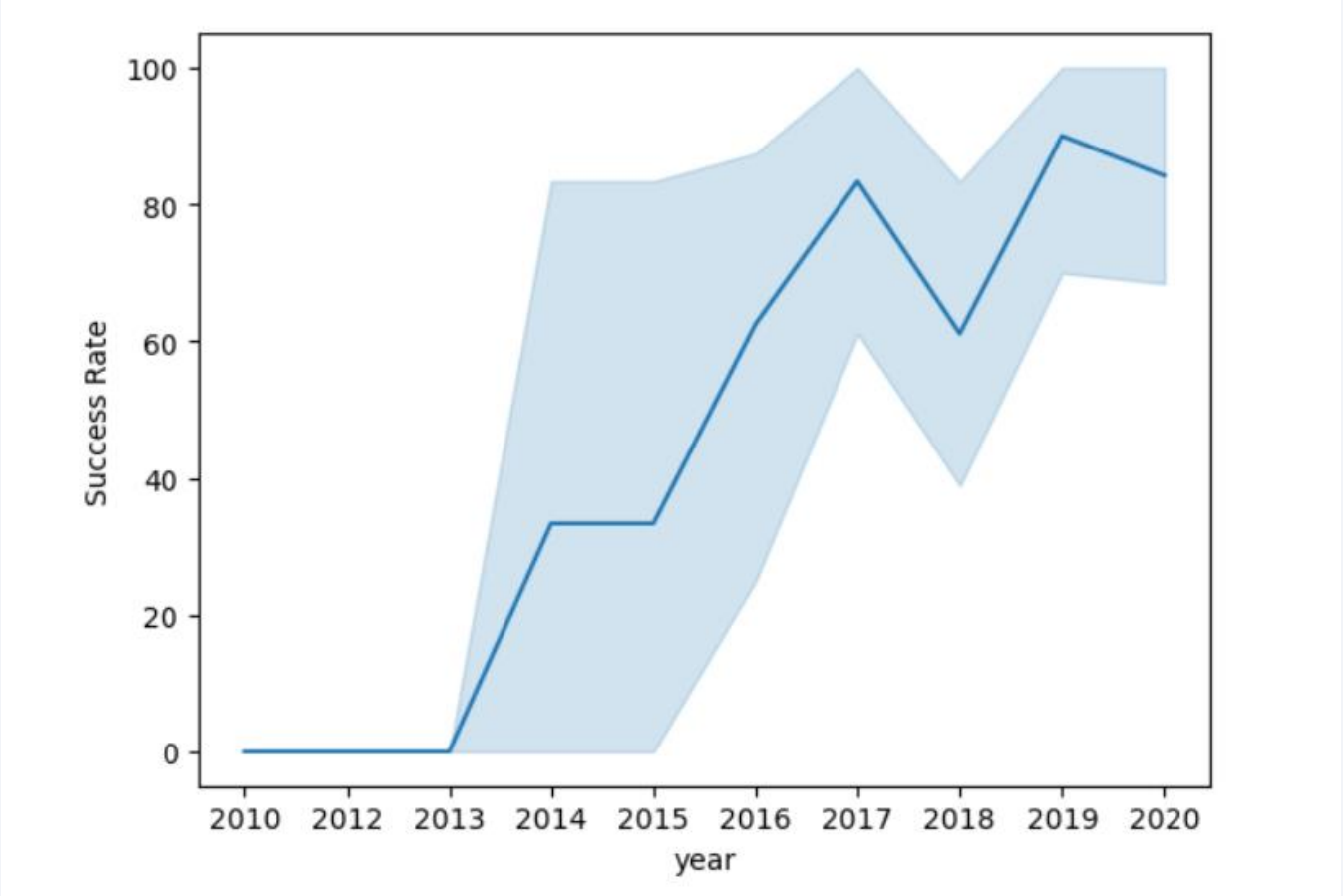
Payload vs. Orbit Type

Orbit type Vs Payloadmass scatter-plot, data split into failure (blue) and successful (orange) launches



- The payload mass seems to be slightly correlated with the orbit choice
- For LEO ISS and SSO orbits, the payload mass is mainly the low range (0-5000 kg)
- For GTO orbit, the payload mass is in the intermediate range (3000-7000 kg)
- For VLEO orbit, the payload mass is in the high range (> 12000 kg)

Launch Success Yearly Trend



- Success yearly trend rate (95% Conf. Level)
- 0% rate before 2013
- Average increase in the success rate since 2013
- Decrease of the rate in 2018
- Success rate presently around 80%

All Launch Site Names

Task 1

Display the names of the unique launch sites in the space mission

In [8]:
```
%sql select DISTINCT LAUNCH_SITE from SPACEXTBL
```

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

Out[8]:

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

- The UNIQUE query is needed to get the unique launch sites name
- Four different sites are identified by the query
- Likely two of the sites conicide
- The sites are: CCAFS LC-40, VAFB SLC-4E, KSC LC-39A

Launch Site Names Begin with ‘CCA’

In [9]:
```
%sql select * from SPACEXTBL where launch_site like 'CCA%' limit 5
```

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

Out[9]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

SELECT query with LIKE ‘CCA’ (to identify the string begining with ‘CCA’) and LIMIT 5 (to limit the result to the first five database rows)

Total Payload Mass

SELECT query including SUM(payload_mass__kg_) to calculate the *total* payload mass and LIKE 'NASA (CRS)' to identify the boosters launched by NASA (CRS)

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

In [10]:
```
%sql select sum(payload_mass__kg_) as sum from SPACEXTBL where customer like 'NASA (CRS)'
```

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

Out[10]:

sum
45596

Average Payload Mass by F9 v1.1

SELECT query with AVG(payload_mass__kg_) to calculate the average payaload mass and the WHERE query to identify the F9 v1.1

Display average payload mass carried by booster version F9 v1.1

In [11]:
```
%sql select avg(payload_mass__kg_) as Average from SPACEXTBL where booster_version like 'F9 v1.1%'
```

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

Out[11]:

Average
2534.6666666666665

First Successful Ground Landing Date

This query identifies the first successful ground landing date, using the SELECT query and the WHERE clause

List the date where the succesful landing outcome in drone ship was acheived.

Hint:Use min function

In [12]:
```
%sql select min(date) as Date from SPACEXTBL where mission_outcome like 'Success'
```

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

Out[12]:

Date
2010-06-04

Successful Drone Ship Landing with Payload between 4000 and 6000

The successful Drone ship landing are listed with a SELECT query. The WHERE clause is used to select the data where the mission outcome is ‘Success’ and the payload mass between 4000 kg and 6000 kg

List the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000

In [13]:
```
%sql select booster_version from SPACEXTBL where (mission_outcome like 'Success') AND (payload_mass__kg_ BETWEEN 4000 AND 6
```

Total Number of Successful and Failure Mission Outcomes

To list the total number of successful and failure mission outcomes a COUNT query has been used, including the COUNT and GROUP BY clauses, in order to divide the launches depending on their outcomes.

List the total number of successful and failure mission outcomes

In [16]:
```
%sql SELECT mission_outcome, count(*) as Count FROM SPACEXTBL GROUP by mission_outcome ORDER BY mission_outcome
```

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

Out[16]:

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

Boosters Carried Maximum Payload

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [17]:
```
maxm = %sql select max(payload_mass__kg_) from SPACEXTBL
maxv = maxm[0][0]

%sql select booster_version from SPACEXTBL where payload_mass__kg_=(select max(payload_mass__kg_) from SPACEXTBL)
```

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

Out[17]:

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

- First the maximum payload is determined
- Then all the booster versions, where the payload mass is equal to the maximum value are listed

2015 Launch Records

With this query some features (month, landing outcome booster version and launch site) are recovered for the mission that took place in 2015

Note: SQLLite does not support monthnames. So you need to use substr(Date,6,2) for month, substr(Date,9,2) for date, substr(Date,0,5),='2017' for year.

n [18]:

```
%sql select MONTHNAME(DATE) as Month, landing__outcome, booster_version, launch_site from SPACEXTBL where DATE like '2015%'
```

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

A SELECT query is used to count (COUNT FUNCTION) where the Date is between 2010-06-04 and 2017-03-20, Grouping (GROUP BY clause) by landing outcomes and ordering (ORDER BY clause) number of outcomes in descending order (DESC).

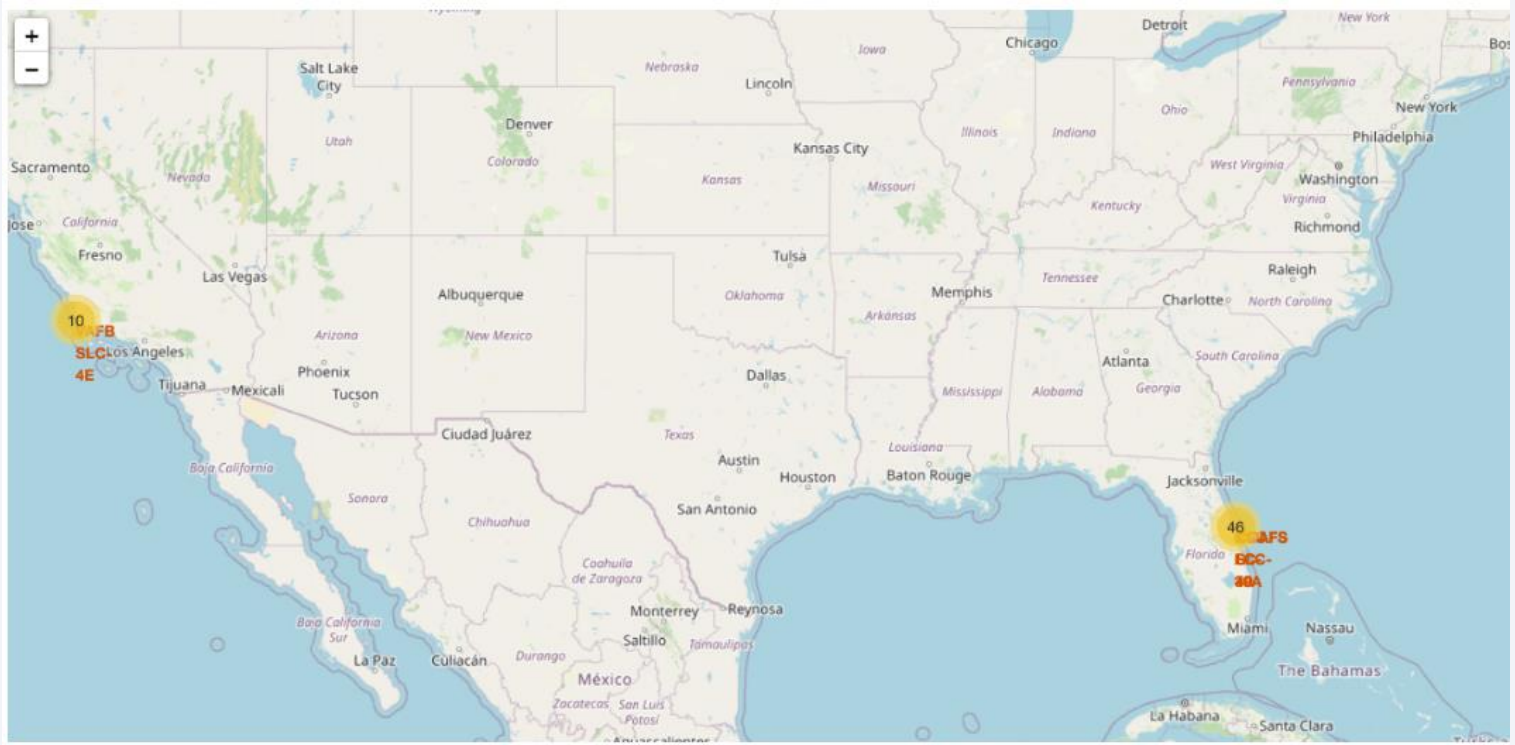
Task 10

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

In [19]:
```
%sql select landing__outcome, count(*) as count from SPACEXTBL where Date >= '2010-06-04' AND Date <= '2017-03-20' GROUP by
```

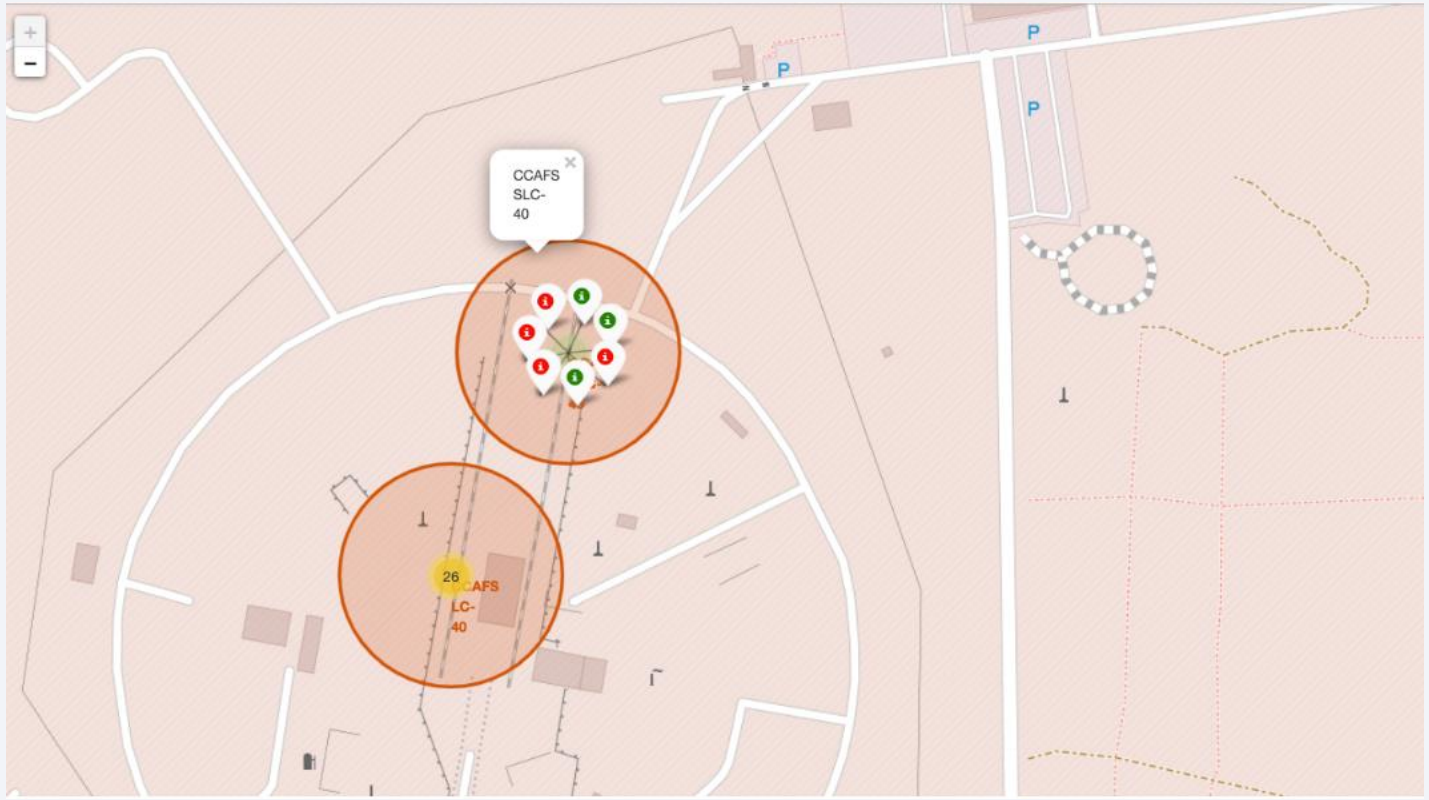

Section 3
Launch Sites
Proximities Analysis

Launch site locations map



- The map shows the launch site locations in the US.
- They are located in California and Florida
- Both of them are located near the ocean and close to the equator

Color Coded Launch sites results

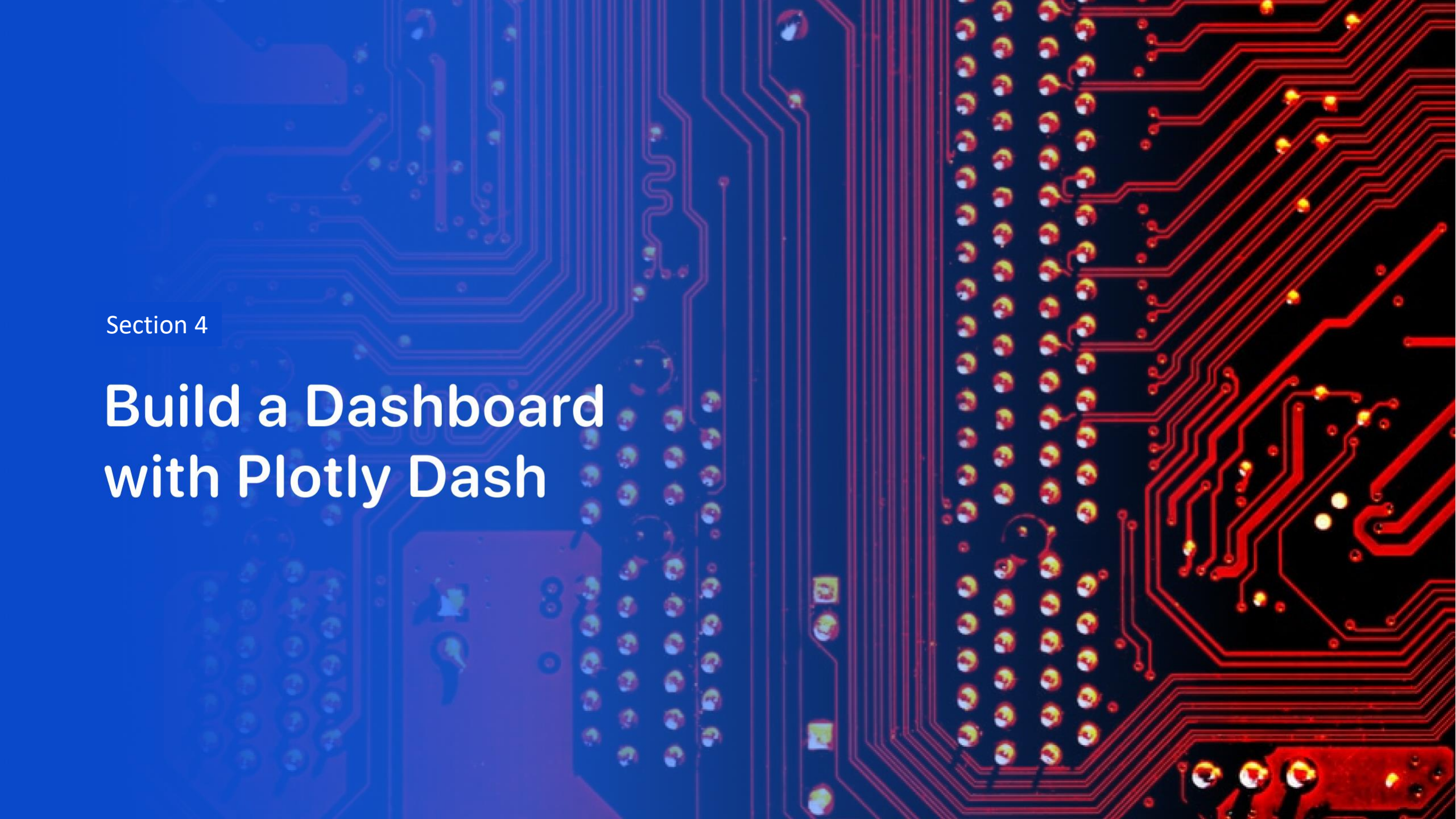


- The map is referred to the CCAFS SLC-40 launch site
- For each launch site, the outcome of launches is shown

Key transportation maps

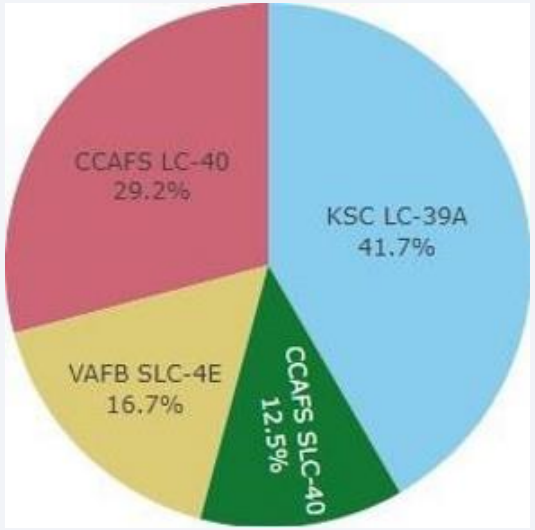
- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed

- Launch sites are relatively close to coast and to infrastructures for transportation of human and material resources to the launch site


Section 4
Build a Dashboard with Plotly Dash

Successful launches location Piechart

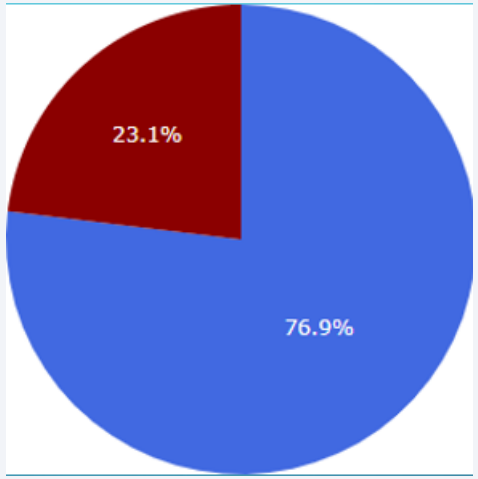
- Launch success count for all sites, in a piechart



- The CCAFS and KSC launch sites have the majority of success launches count (slightly above 40% each)
- The VAFB has a lower percentage of success launches count (about 17%)

Highest successful rate launch site

- Piechart for the launch site with highest launch success ratio



- KSC LC-39A has the highest successful rate of booster landing (76.9%) in blue
- The unsuccess rate (red) is about (23.1%)

Success Vs Payloadmass (booster version category)

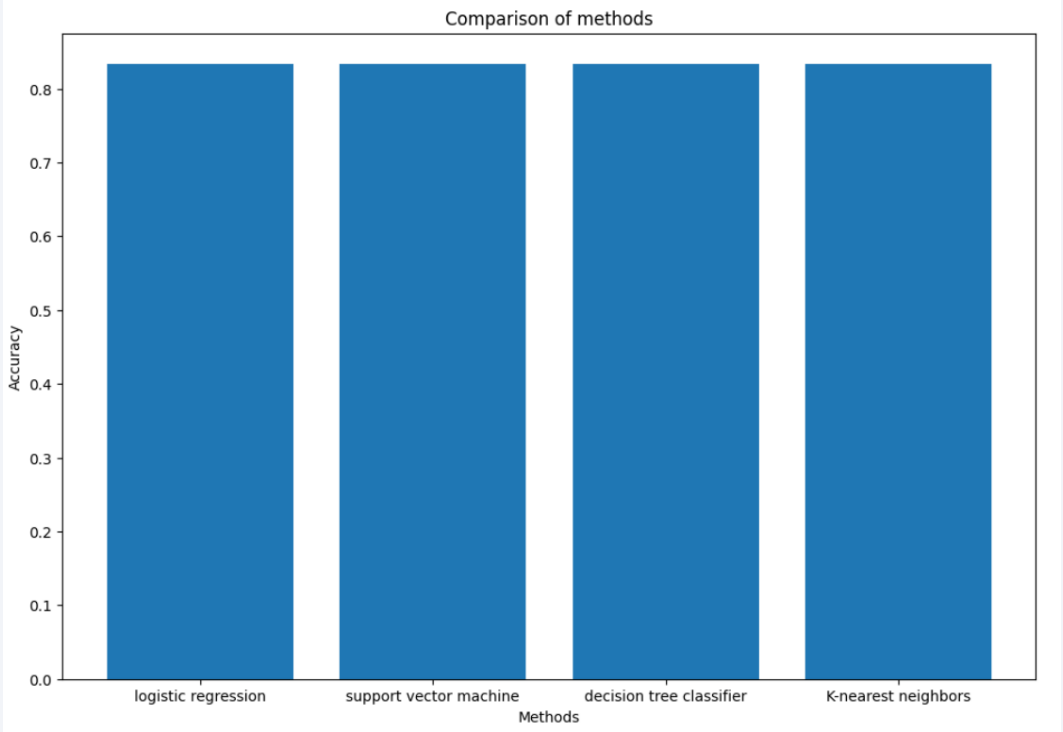
- Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

- In this example the maximum payload mass is fixed at 6000 kg.
- Class represents the success (1) or failure (0) of the launch
- The booster version are represented by the different bullets color

Section 5

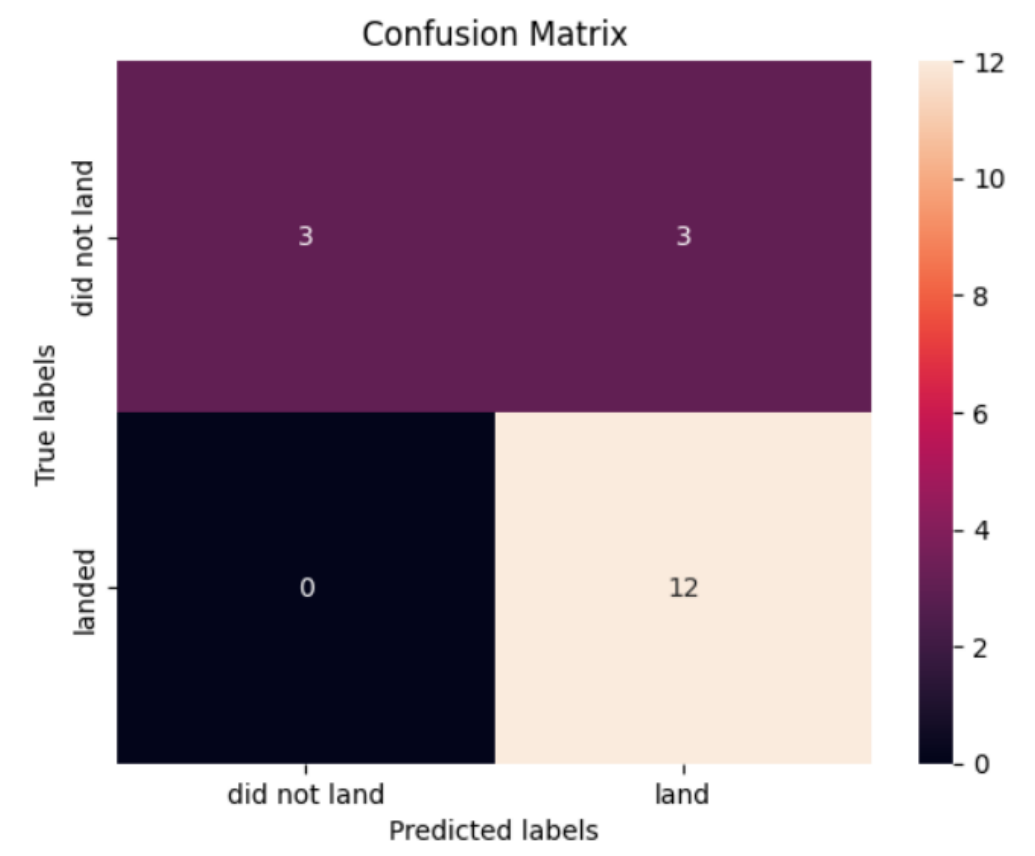
Predictive Analysis (Classification)

Classification Accuracy



- Four different comparison methods have been tested: logistic regression, support vector machine, decision tree classifiers, K-nearest neighbors
- All the four methods return the same accuracy level: 83 %
- At the moment the small size of the sample analyzed (18 elements) makes difficult to state whether a method performs better then the others

Confusion Matrix



- The confusion matrix is the same for the four methods analyzed
- The right predictions are in the top left - bottom right diagonal
- The wrong predictions are in the bottom left – top right diagonal
- The model over-predicted successful landings in 3 cases of 18.
- The outcome of the other 15 cases are correctly predicted

Conclusions

- Goal: developing a Machine Learning model to describe the Falcon 9 first stage landing (and recovery)
- After collecting and analyzing the data, four different ML models have been applied: logistic regression, support vector machine, decision tree classifiers, K-nearest neighbors
- Probability accuracy: 83%

→ The rocket first stage recovery is confirmed as a promising method for future space missions. The predictive classification ML methods tested show a (relatively) high prediction reliability

→ More data are required to select the best performing ML method.

Appendix [Jupyter Notebook links]

- https://github.com/nicholasviv95/IBM_ML_Capstone_project_final_exam/blob/main/NV_lab_1.0_data_collection.ipynb
- https://github.com/nicholasviv95/IBM_ML_Capstone_project_final_exam/blob/main/NV_lab_1.1_webscraping.ipynb
- https://github.com/nicholasviv95/IBM_ML_Capstone_project_final_exam/blob/main/NV_lab_1.2_data_wrangling.ipynb
- https://github.com/nicholasviv95/IBM_ML_Capstone_project_final_exam/blob/main/NV_lab_2.1_eda.ipynb
- https://github.com/nicholasviv95/IBM_ML_Capstone_project_final_exam/blob/main/NV_lab_2.2_eda_sql.ipynb
- https://github.com/nicholasviv95/IBM_ML_Capstone_project_final_exam/blob/main/NV_lab_3.1_Folium.ipynb
- https://github.com/nicholasviv95/IBM_ML_Capstone_project_final_exam/blob/main/NV_lab_3.2_dash.py
- https://github.com/nicholasviv95/IBM_ML_Capstone_project_final_exam/blob/main/NV_lab_4_predictive_analysis.ipynb

Thank you!