IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Monika Nyakone Hegedus-Lukacs
20/06/2024

Outline

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

Executive Summary

- This project is about analyzing SpaceX launch records to gain valuable insights.
- Our goal is to predict if the mission outcome will be successful or not.
- Data has been collected from multiple sources and the datasets needed restructuring and data wrangling before the analysis could start.
- Datasets have been analyzed by various means and methods such as Exploratory Data Analysis, Interactive Visual Analytics and Predictive Analysis.
- The machine learning model built in this project can predict the mission outcome with an 83% accuracy.

Introduction

Project background

SpaceX is a private space company founded in 2002. It designs, manufactures and launches advanced rockets and spacecraft. It also manufactures a so-called Falcon 9 rocket whose first stage is capable of landing and reflight. Since the most expensive part of the rocket can be reused, the cost of space access can be significantly decreased.

Project goal

In this project, our goal is to predict if the first stage of Falcon 9 will land successfully or not. If we can predict the mission outcome, we can determine the cost of the mission more precisely, which might help when bidding against SpaceX as an alternate company.

Section 1

Methodology

Methodology

Data Collection

Data Wrangling

Exploratory Data Analysis (EDA)

Interactive Visual Analytics

Predictive Analysis

Executive Summary

Data was collected from a REST API and via web scraping. During data wrangling, irrelevant data was removed, missing values were identified and replaced, string values were converted to numerical ones where needed. While performing Exploratory Data Analysis, various plots were generated and SQL was used to gain more insights. During Visual Analytics, an interactive Folium map and an interactive dashboard were created. Predictive Analysis was performed by building, tuning and evaluating various classification models.

Data Collection

Data was collected from 2 sources:

1) SpaceX REST API:
   https://api.spacexdata.com/v4/launches/past

[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"sh
{"small":"https://images2.imgbox.com/94/f2/NN6Ph45r_o.png","large":"https:/
[],"original":[]},"presskit":null,"webcast":"https://www.youtube.com/watch?
launch.html","wikipedia":"https://en.wikipedia.org/wiki/DemoSat"},"static_f
17T00:00:00.000Z","static_fire_date_unix":1142553600,"net":false,"window":0
failure"}],"details":"Engine failure at 33 seconds and loss of vehicle","cre
["5eb0e4b5b6c3bb0006eeb1e1"],"launchpad":"5e9e4502f5090995de566f86","flight
25T10:30:00+12:00","date_precision":"hour","upcoming":false,"cores":
[{"core":"5e9e289df35918033d3b2623","flight":1,"gridfins":false,"legs":false
ch_library_id":null,"id":"5eb87cd9ffd86e000604b32a"},{"fairings":{"reused":
{"small":"https://images2.imgbox.com/f9/4a/ZboXReNb_o.png","large":"https:/
[],"original":[]},"presskit":null,"webcast":"https://www.youtube.com/watch?
orbit.html","wikipedia":"https://en.wikipedia.org/wiki/DemoSat"},"static_fi
[{"time":301,"altitude":289,"reason":"harmonic oscillation leading to prema
shutdown at T+7 min 30 s, Failed to reach orbit, Failed to recover first sta
["5eb0e4b6b6c3bb0006eeb1e2"],"launchpad":"5e9e4502f5090995de566f86","flight
21T13:10:00+12:00","date_precision":"hour","upcoming":false,"cores":
[{"core":"5e9e289ef35918416a3b2624","flight":1,"gridfins":false,"legs":false
ch_library_id":null,"id":"5eb87cdaffd86e000604b32b"},{"fairings":{"reused":
{"small":"https://images2.imgbox.com/6c/cb/na1tzhHs_o.png","large":"https:/
[],"original":[]},"presskit":null,"webcast":"https://www.youtube.com/watch?
summary","wikipedia":"https://en.wikipedia.org/wiki/Trailblazer_(satellite)
ailures":[{"time":140,"altitude":35,"reason":"residual stage-1 thrust led t
[],"ships":[],"capsules":[],"payloads":["5eb0e4b6b6c3bb0006eeb1e3","5eb0e4b
03T03:34:00.000Z","date_unix":1217734440,"date_local":"2008-08-03T15:34:00+
[{"core":"5e9e289ef3591814873b2625","flight":1,"gridfins":false,"legs":false

2) Wikipedia – web scraping Falcon 9 launch records with BeautifulSoup

[hide] Flight No.	Date and time (UTC)	Version, booster[c]	Launch site	Payload[d]	Payload mass	Orbit	Customer	Launch outcome	Booster landing
195	3 January 2023 14:56[22]	F9 B5 ♺ B1060.15[23]	CCSFS, SLC-40	*Transporter-6*: (115 payloads Smallsat Rideshare)	Unknown[e]	SSO	Various	Success	Success (ground pad)
	Dedicated SmallSat Rideshare mission to Sun-synchronous orbit. It included six space tugs, also known as orbital transfer vehicles (OTV), which are two of D-Orbit's ION Satellite Carriers, Epic Aerospace's Chimera LEO 1, Momentus's Vigoride-5, Skykraft's OTV and Launcher's Orbiter SN1.[24][25] Orbiter SN1 failed shortly after deployment from Falcon and before deploying payloads. One of the payloads, EWS RROCI failed to deploy from Falcon 9 and the satellite re-entered with the upper stage.[26] This was not a SpaceX failure as brokered dispensers and deployers are used on Transporter missions.[27]								
196	10 January 2023 04:50[28]	F9 B5 ♺ B1076.2	CCSFS, SLC-40	OneWeb Flight #16 / SpaceX Flight 2[29][30] (40 satellites)[31]	6,000 kg (13,000 lb)	Polar LEO	OneWeb	Success	Success (ground pad)
	Following the Russian invasion of Ukraine, OneWeb suspended launches on Soyuz rockets.[32] In March 2022, OneWeb announced that they had signed an agreement with SpaceX to resume satellite launches.[33]								
FH 5	15 January 2023 22:56[34]	Falcon Heavy B5 B1070 (core)	KSC, LC-39A	USSF-67 (CBAS-2 & LDPE-3A)[35]	~3,750 kg (8,270 lb)	GEO	USSF	Success	No attempt
		B1064.2 (side) ♺							Success (ground pad)
		B1065.2 (side) ♺							Success (ground pad)

Data Collection – SpaceX API

- Data collection with SpaceX REST calls

```
spacex_url = https://api.spacexdata.com/v4/launches/past

response = requests.get(spacex_url)

data = pd.json_normalize(response.json())

getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)

launch_dict = {'FlightNumber': list(data['flight_number']),'Date':
list(data['date']),'BoosterVersion':BoosterVersion,'PayloadMass':PayloadMass,'
Orbit':Orbit,'LaunchSite':LaunchSite,'Outcome':Outcome,'Flights':Flights,'Grid
Fins':GridFins,'Reused':Reused,'Legs':Legs,'LandingPad':LandingPad,'Block':Blo
ck,'ReusedCount':ReusedCount,'Serial':Serial,'Longitude':
Longitude,'Latitude': Latitude}

launch_data = pd.DataFrame(launch_dict)
```

- GitHub URL of SpaceX API calls:
  [Module-1_jupyter-labs-spacex-data-collection-api.ipynb]

Sending HTTP GET request to SpaceX API and saving response

↓

Decoding response as json and converting it to pandas dataframe

↓

Extracting additional information from API based on IDs (cores, launchpads, ...) into lists

↓

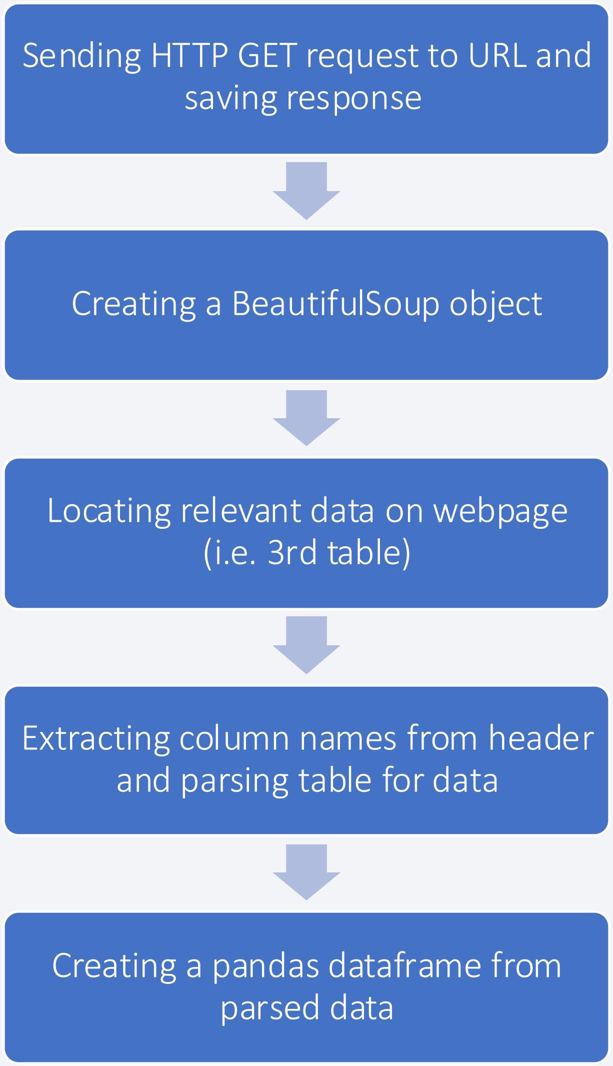
Combining lists into dictionary and then converting it to a dataframe

Data Collection – Web Scraping

- Web scraping process

```
response = requests.get(static_url).text
soup = BeautifulSoup(response, 'html5lib')
html_tables = soup.find_all('table')
first_launch_table = html_tables[2]   # 3rd table
column_names = []
launch_dict = dict.fromkeys(column_names)
# […] parsing table...
df = pd.DataFrame({ key:pd.Series(value) for key,
value in launch_dict.items() })
```

- GitHub URL of Web Scraping notebook:
  [Module-1_jupyter-labs-webscraping.ipynb]



Data Wrangling

- Removing irrelevant data (e.g. Falcon 1)

```
data_falcon9 = launch_data[launch_data['BoosterVersion']!='Falcon 1']
```

- Identifying missing values

```
data_falcon9.isnull().sum()
```

- Replacing missing values with mean value

```
avg_PayloadMass = data_falcon9['PayloadMass'].astype('float').mean(axis=0)
data_falcon9['PayloadMass'].replace(np.nan, avg_PayloadMass, inplace=True)
```

- Converting landing outcomes to classes (1 or 0)

```
for i, outcome in enumerate(landing_outcomes.keys()):
 print(i,outcome)
bad_outcomes = set(landing_outcomes.keys()[[1,3,5,6,7]])
landing_class = [0 if outcome in bad_outcomes else 1 for outcome in df['Outcome']]
df['Class'] = landing_class
```

- GitHub URL of Data Wrangling related notebooks:
  [Module-1_jupyter-labs-spacex-data-collection-api.ipynb]
  [Module-1_jupyter-labs-spacex-data-wrangling.ipynb]

```
FlightNumber       0
Date               0
BoosterVersion     0
PayloadMass        5
Orbit              0
LaunchSite         0
Outcome            0
Flights            0
GridFins           0
Reused             0
Legs               0
LandingPad        26
Block              0
ReusedCount        0
Serial             0
Longitude          0
Latitude           0
dtype: int64
```

```
0 True ASDS
1 None None
2 True RTLS
3 False ASDS
4 True Ocean
5 False Ocean
6 None ASDS
7 False RTLS
```

	Class
0	0
1	0
2	0
3	0
4	0

EDA with Data Visualization

- The following charts have been plotted during EDA:
  - Task 1: Scatter plot displaying FlightNumber vs LaunchSite
  - Task 2: Scatter plot displaying PayloadMass vs LaunchSite
  - Task 3: Bar chart displaying Orbit vs mean success rate
  - Task 4: Scatter plot displaying Orbit vs FlightNumber
  - Task 5: Scatter plot displaying Orbit vs PayloadMass
  - Task 6: Line chart displaying Year vs mean success rate

- GitHub URL of EDA with data visualization notebook: [Module-2_jupyter-labs-eda-dataviz.ipynb]

EDA with SQL

- The following SQL queries have been performed during EDA:
  - Task 1: `%sql SELECT DISTINCT ("Launch_Site") FROM SPACEXTABLE;`
  - Task 2: `%sql SELECT * FROM SPACEXTABLE WHERE "Launch_Site" LIKE ('CCA%') LIMIT 5;`
  - Task 3: `%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE "Customer" = 'NASA (CRS)';`
  - Task 4: `%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE "Booster_Version" LIKE ('F9 v1.1%');`
  - Task 5:
    ```
    %%sql SELECT MIN(DATE) FROM SPACEXTABLE
    WHERE "Mission_Outcome" = 'Success' AND "Landing_Outcome" LIKE ('%ground%');
    ```
  - Task 6:
    ```
    %%sql SELECT "Booster_Version" FROM SPACEXTABLE
    WHERE "Landing_Outcome" LIKE ('%Success%') AND "Landing_Outcome" LIKE ('%drone%')
    AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
    ```
  - Task 7: `%sql SELECT COUNT("Mission_Outcome") FROM SPACEXTABLE;`
  - Task 8:
    ```
    %%sql SELECT "Booster_Version" FROM SPACEXTABLE
    WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
    ```
  - Task 9:
    ```
    %%sql SELECT substr(Date, 6,2) AS Month, "Landing_Outcome", "Booster_Version", "Launch_Site"
    from SPACEXTABLE WHERE "Landing_Outcome" LIKE ('%Failure%') AND "Landing_Outcome" LIKE ('%drone%')
    AND substr(Date,0,5)='2015';
    ```
  - Task 10:
    ```
    %%sql SELECT * FROM (SELECT "Landing_Outcome", COUNT(*) AS QTY FROM SPACEXTABLE
    WHERE Date BETWEEN "2010-06-04" and "2017-03-20" GROUP BY "Landing_Outcome" ) AS TASK_10_TABLE;
    ```
- GitHub URL of EDA with SQL notebook: [Module-2_jupyter-labs-eda-sql.ipynb]

Build an Interactive Map with Folium

- The following map objects have been created and added:
  - Folium Map object (`folium.Map`) to create a map
  - Circle object (`folium.Circle`) to add a highlighted circle area for each launch site
  - Marker object (`folium.map.Marker`) to mark specific locations on the map
  - Polyline object (`folium.PolyLine`) to draw a line between two points

- GitHub URL of Interactive Map with Folium:
  Module-3_jupyter-labs-launch-sites.ipynb

Build a Dashboard with Plotly Dash

- The following plots/graphs and interactions have been added to the dashboard:
  - Dropdown menu (`dcc.Dropdown`) to enable users to select launch sites
  - Pie chart (`px.pie`) to visualize success rate per launch site
  - Range slider (`dcc.RangeSlider`) to enable users to select a payload range
  - Scatter plot (`px.scatter`) to visualize how payload and mission outcomes are correlated

- GitHub URL of Plotly Dash app:
  Module-3_spacex_dash_app.py

Predictive Analysis (Classification)

- Model development process:

Creating a numpy array from column Class → Standardizing the data in dataframe X → Splitting the data into training and test data

Creating a classification model object → Fitting the object to find the best parameters → Calculating the accuracy of the classification model

- GitHub URL of Predictive Analysis lab: [Module-4_jupyter-labs-machine-learning.ipynb]

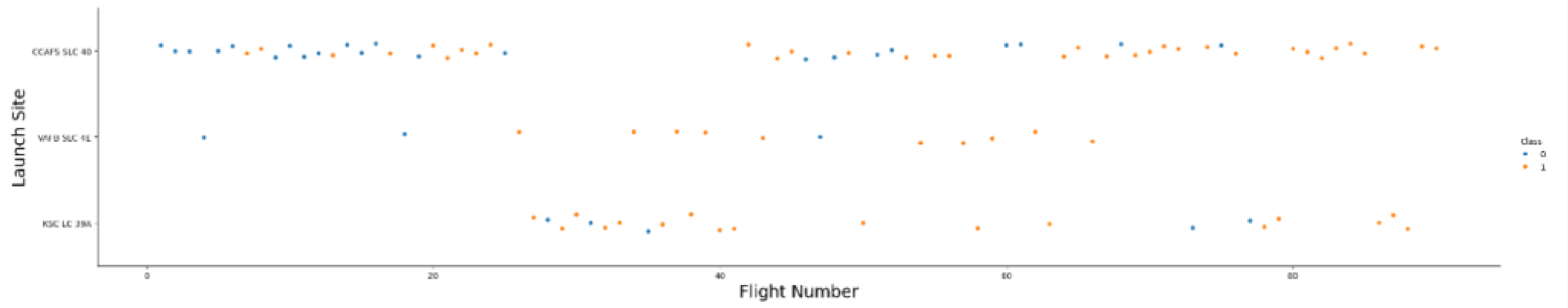
Results

- Exploratory data analysis results are introduced in
  - Section 2: Insights drawn from EDA
- Interactive analytics demo in screenshots is presented in
  - Section 3: Launch Sites Proximities Analysis
  - Section 4: Build a Dashboard with Plotly Dash
- Predictive analysis results are introduced in
  - Section 5: Predictive Analysis (Classification)

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site



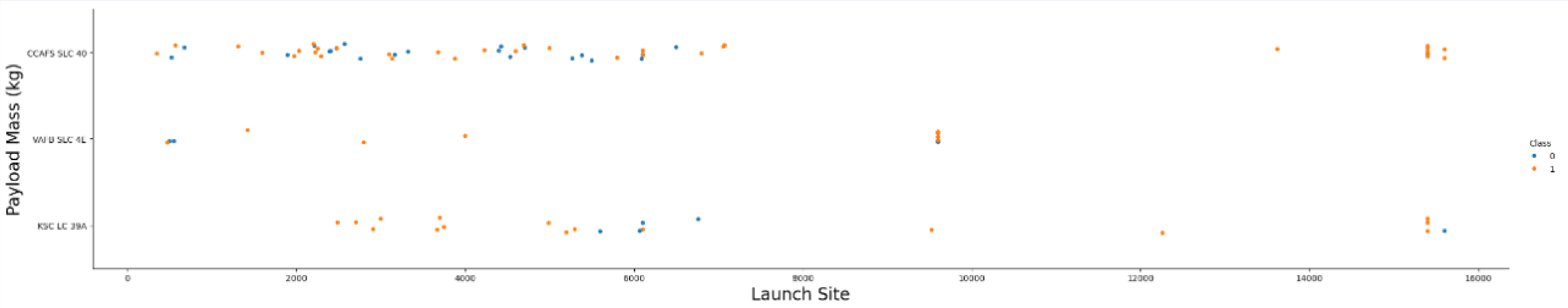

```
sns.catplot(y="LaunchSite", x="FlightNumber",
hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```

Scatter plot of Flight Number vs. Launch Site

Explanations:

- The most flights are launched from the site called CCAFS SLC 40.
- The launch sites VAFB SLC 4E and KSC LC 39A have got a similar success rate.

Payload vs. Launch Site



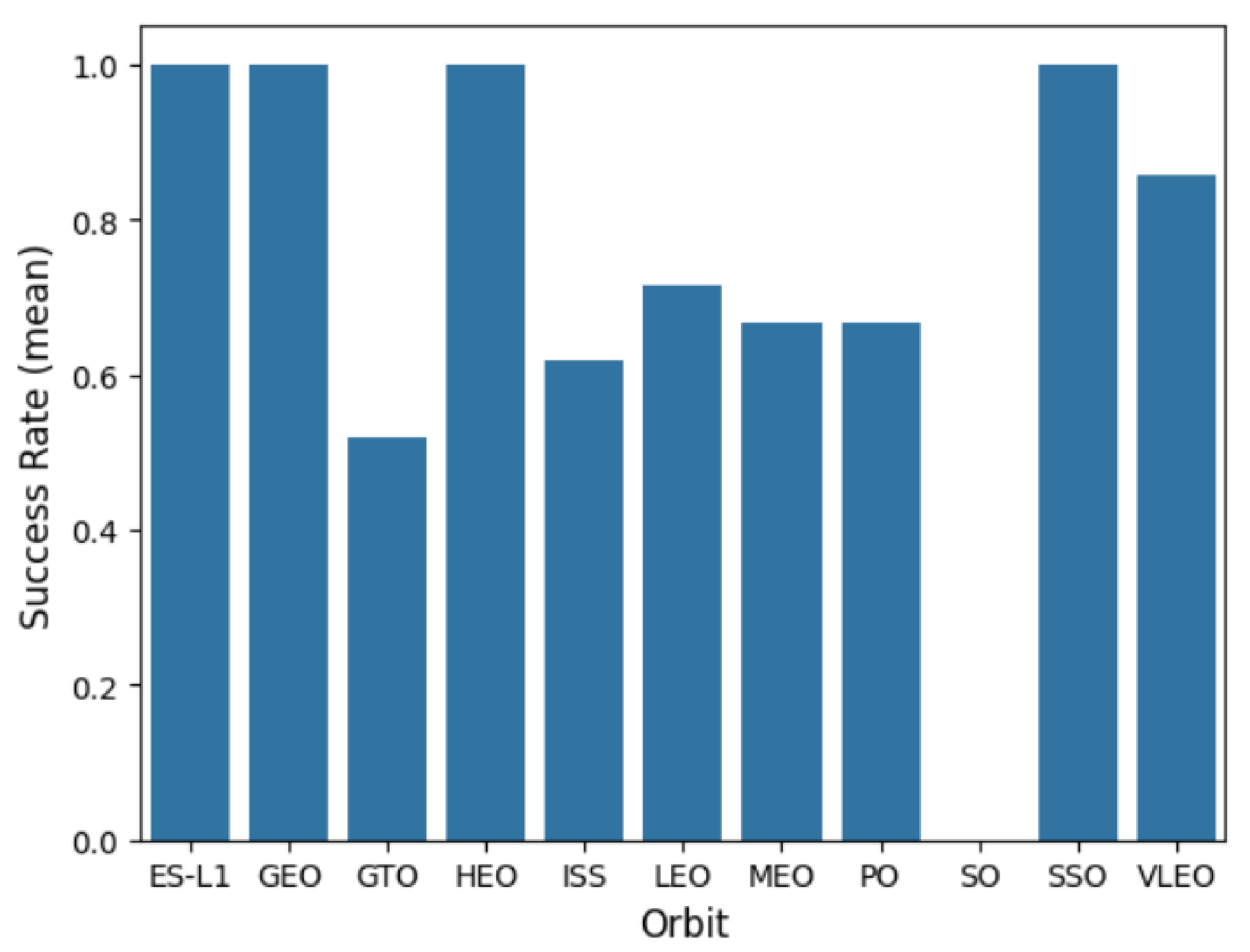

```
sns.catplot(x="PayloadMass", y="LaunchSite",
hue="Class", data=df, aspect = 5)
plt.xlabel("Launch Site",fontsize=20)
plt.ylabel("Payload Mass (kg)",fontsize=20)
plt.show()
```

Scatter plot of Payload vs. Launch Site

Explanations:

- Rockets with heavy payload are launched both from CCAFS SLC 40 and KSC LC 39A but not from VAFB SLC 4E.

Success Rate vs. Orbit Type



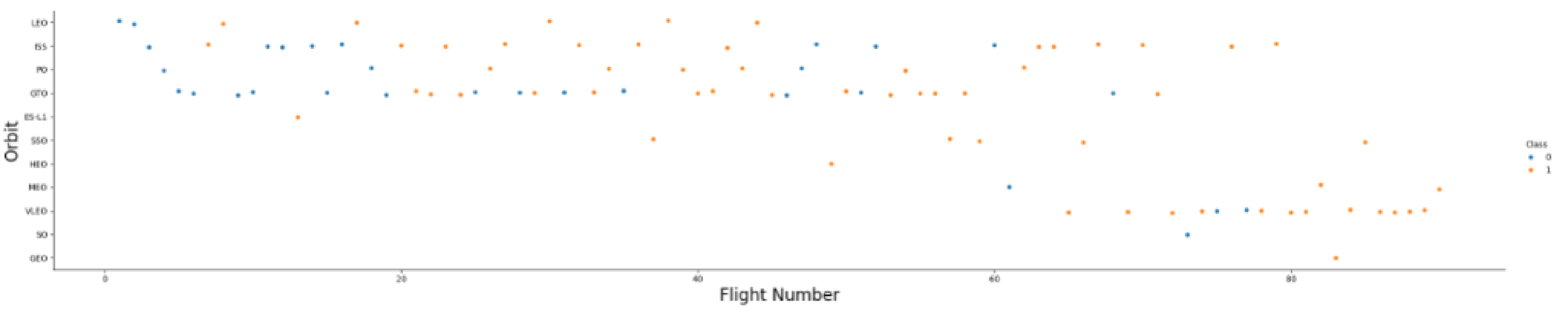

```
df_task3 =
df.groupby(['Orbit']).mean('Class')
sns.barplot(x='Orbit', y='Class',
data=df_task3)
plt.xlabel("Orbit", fontsize=12)
plt.ylabel("Success Rate (mean)",
fontsize=12)
plt.show()
```

Bar chart displaying the success rate of each orbit type

Explanations:

- The orbits ES-L1, GEO, HEO and SSO have got the highest success rate.
- The worst success rate belongs to the orbit GTO.

Flight Number vs. Orbit Type



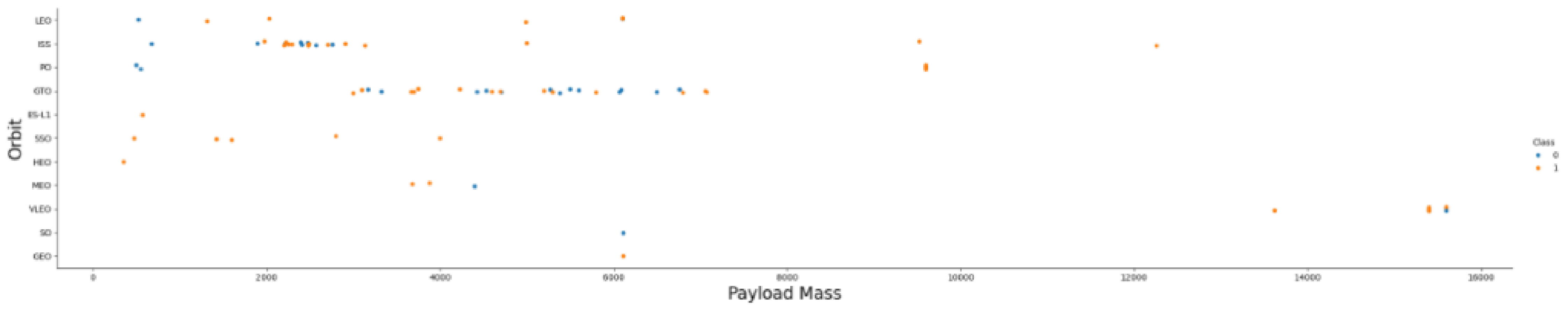

```
sns.catplot(y="Orbit", x="FlightNumber",
hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

Scatter plot of Flight number vs. Orbit type

Explanations:

- The success rate seems to be correlated to the number of flights on the orbit LEO.
- No such relationship is observed on the other orbits.

Payload vs. Orbit Type



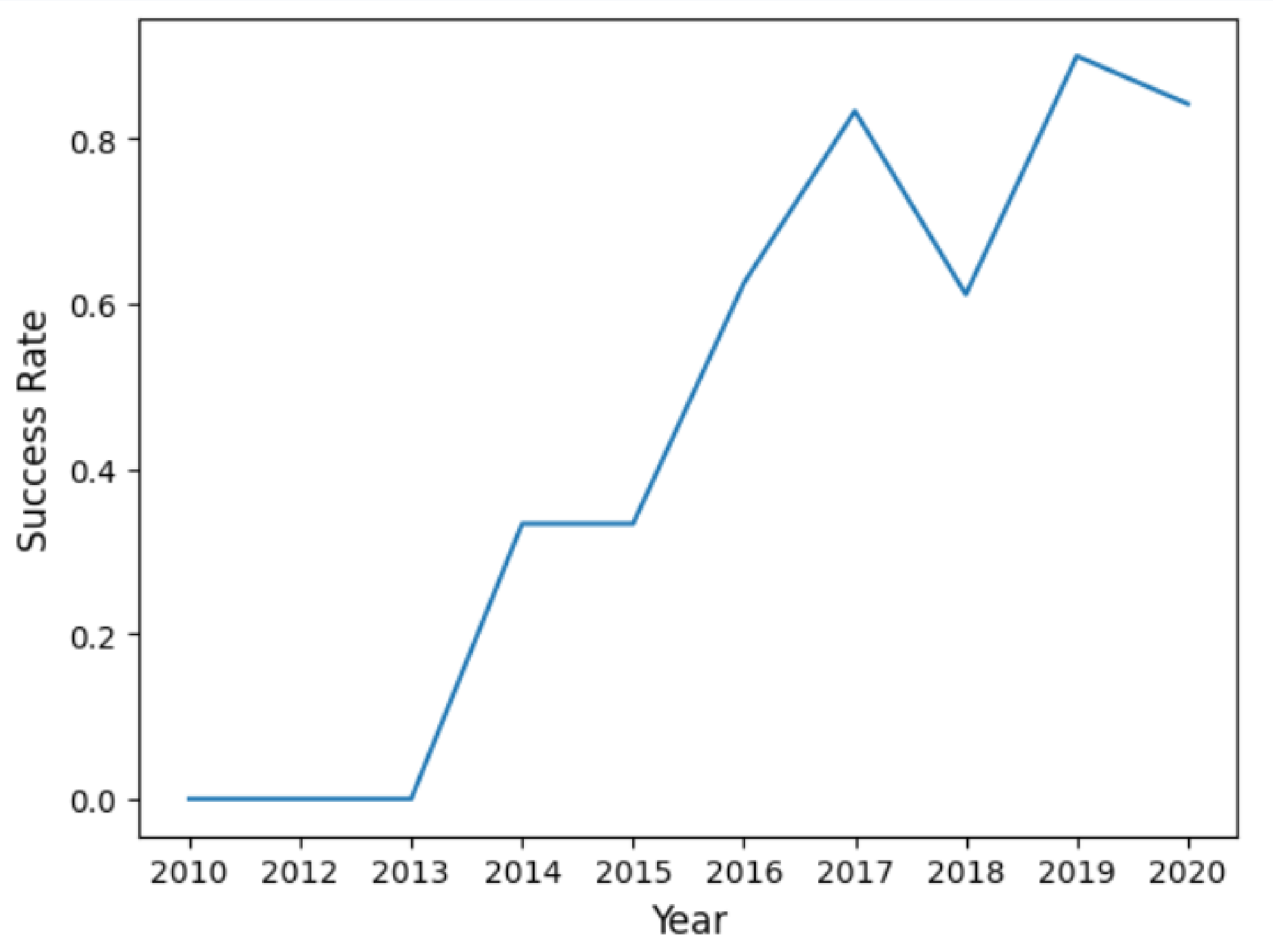

```
sns.catplot(y="Orbit", x="PayloadMass",
hue="Class", data=df, aspect = 5)
plt.xlabel("Payload Mass",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

Scatter plot of Payload vs. Orbit type

Explanations:

- Rockets with the heaviest payload are on the orbit VLEO.
- The success rate and the payload seems to be positively correlated on the orbits LEO, ISS and PO.

Launch Success Yearly Trend




```
sns.lineplot(x=np.unique(df['Date']),
y=df.groupby('Date')['Class'].mean())
plt.xlabel("Year",fontsize=12)
plt.ylabel("Success Rate",fontsize=12)
plt.show()
```

Line chart displaying yearly average success rate

Explanations:

- The success rate and year are positively correlated.

All Launch Site Names

Finding the names of the unique launch sites:

```
%sql SELECT DISTINCT ("Launch_Site") FROM SPACEXTABLE;
```

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

Explanation:

There are 4 unique launch sites in the dataset.

Launch Site Names Begin with 'CCA'

Finding 5 records where launch sites begin with `CCA`:

```
%%sql SELECT * FROM SPACEXTABLE
WHERE "Launch_Site" LIKE ('CCA%') LIMIT 5;
```

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

Explanation:

Records have been filtered to contain only the rows where launch sites begin with "CCA", from which only 5 records are displayed.

Total Payload Mass

Calculating the total payload carried by boosters from NASA:

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

SUM(PAYLOAD_MASS__KG_)
45596

Explanation:

The sum of all values in Payload_Mass is displayed as a result.

Average Payload Mass by F9 v1.1

Calculating the average payload mass carried by booster version F9 v1.1:

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

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

Explanation:

Records have been filtered to contain only the rows where the Booster Version is F9 v1.1 and the average of the values is displayed as a result.

First Successful Ground Landing Date

Finding the dates of the first successful landing outcome on ground pad:

```
%%sql
SELECT MIN(DATE) FROM SPACEXTABLE
WHERE "Mission_Outcome" = 'Success'
AND "Landing_Outcome" LIKE ('%ground%');
```

MIN(DATE)
2015-12-22

Explanations:

Records have been filtered to contain only the rows where the mission was successful and the rocket has landed on a ground pad. The first date is displayed as a result.

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
%%sql
SELECT "Booster_Version" FROM SPACEXTABLE
WHERE "Landing_Outcome" LIKE ('%Success%')
AND "Landing_Outcome" LIKE ('%drone%')
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Explanation:

Records have been filtered to contain only the rows where:

- the mission was successful,
- and the rocket has landed on a drone ship,
- and the payload mass is between 4000 and 6000 kg.

The booster versions are displayed as a result.

Total Number of Successful and Failure Mission Outcomes

Calculating the total number of successful and failure mission outcomes:

```
%sql SELECT COUNT("Mission_Outcome") FROM SPACEXTABLE;
```

COUNT("Mission_Outcome")
101

Explanation:

The values in Mission_Outcome have been counted and this number is displayed as a result.

Boosters Carried Maximum Payload

Listing the names of the booster which have carried the maximum payload mass:

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

Explanation:

Records have been filtered to contain only the rows where the rockets have carried the heaviest payload mass.

The booster versions are displayed as a result.

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

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

```
%%sql
SELECT substr(Date, 6,2) AS Month, "Landing_Outcome", "Booster_Version", "Launch_Site" from
SPACEXTABLE
WHERE "Landing_Outcome" LIKE ('%Failure%')
AND "Landing_Outcome" LIKE ('%drone%')
AND substr(Date,0,5)='2015';
```

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

Explanation:

Records have been filtered to contain only the rows where the landing outcome on a drone ship was a failure in 2015.

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

Ranking 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:

```
%%sql
SELECT * FROM (
SELECT "Landing_Outcome", COUNT(*) AS QTY FROM SPACEXTABLE
WHERE Date BETWEEN "2010-06-04" and "2017-03-20"
GROUP BY "Landing_Outcome"
) AS TASK_10_TABLE;
```

Landing_Outcome	QTY
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

Explanation:

Records have been filtered to contain only the rows between the dates 2010-06-04 and 2017-03-20, and the numbers of different landing outcomes are displayed.

Section 3

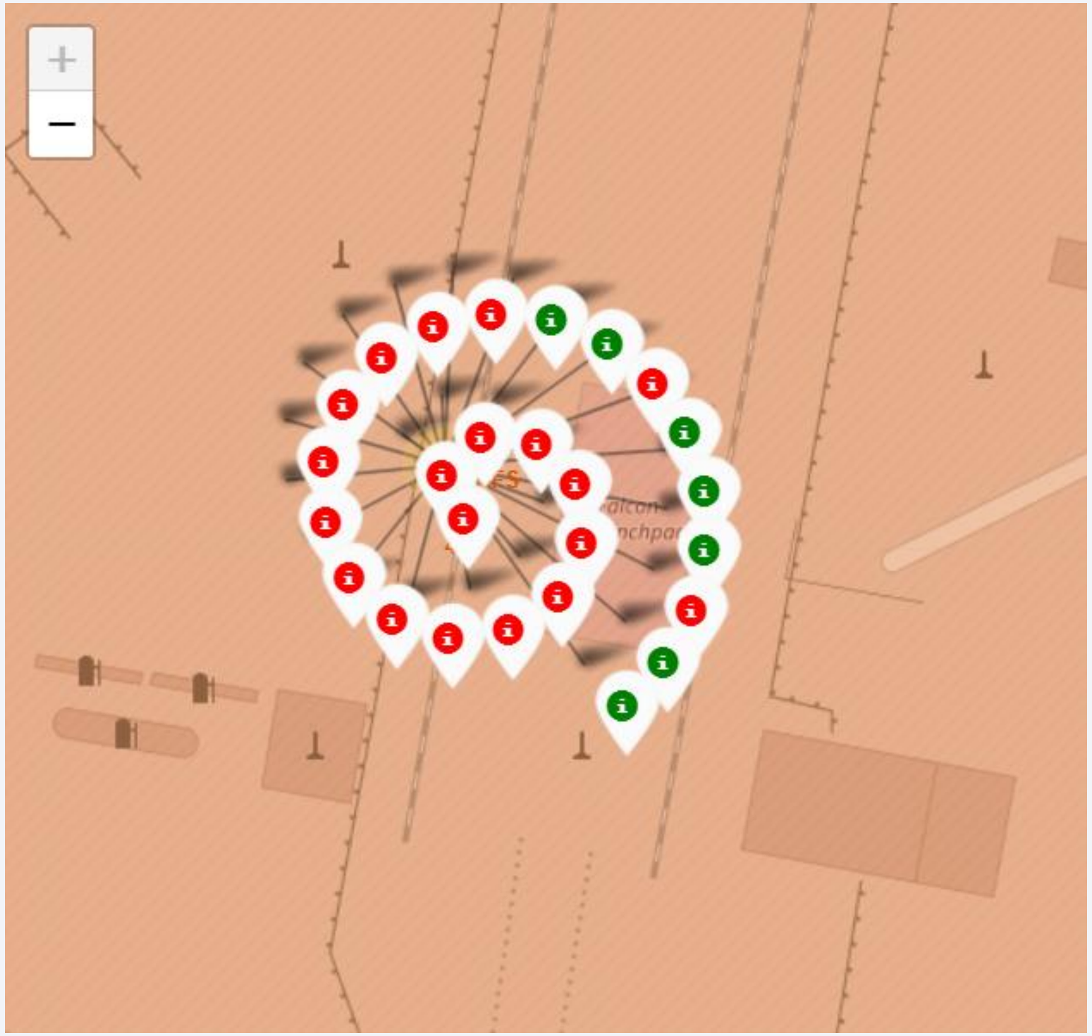
Launch Sites Proximities Analysis

Marking All Launch Sites on a Folium Map

SpaceX launch sites are:

- close to the equator
- close to the ocean
- close to transportation infrastructure

Marking Successful/Failed Launches on Each Site

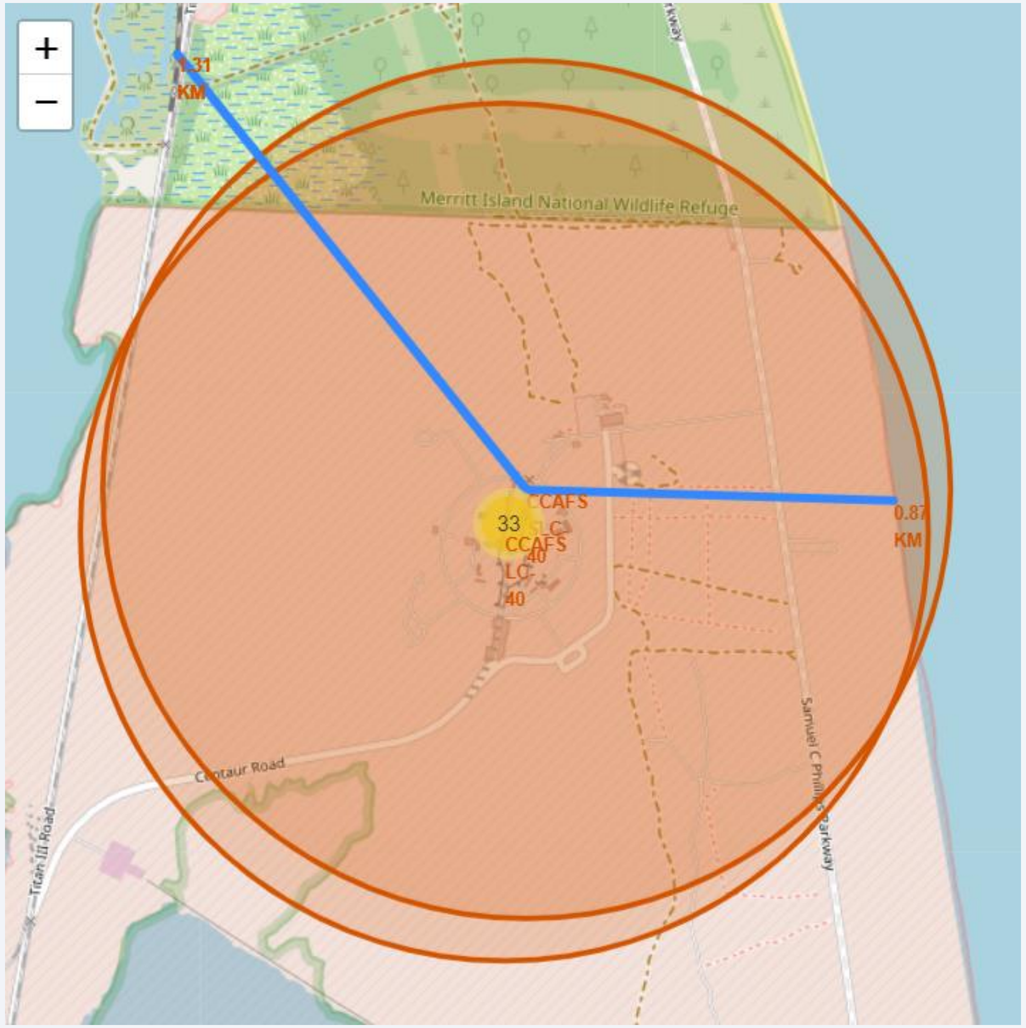


By adding colour-labelled markers to the site map, the launch outcomes become clearly visible on each site.

For example:

- most of the missions on the launch site CCAFS LC-40 ended with a failure,
- whereas, most of the missions on the launch site KSC LC-39A were successful.

Calculating Distances between a Launch Site and its Proximities



By adding distance markers to the site map, the distance of launch sites from their proximities become clearly visible.

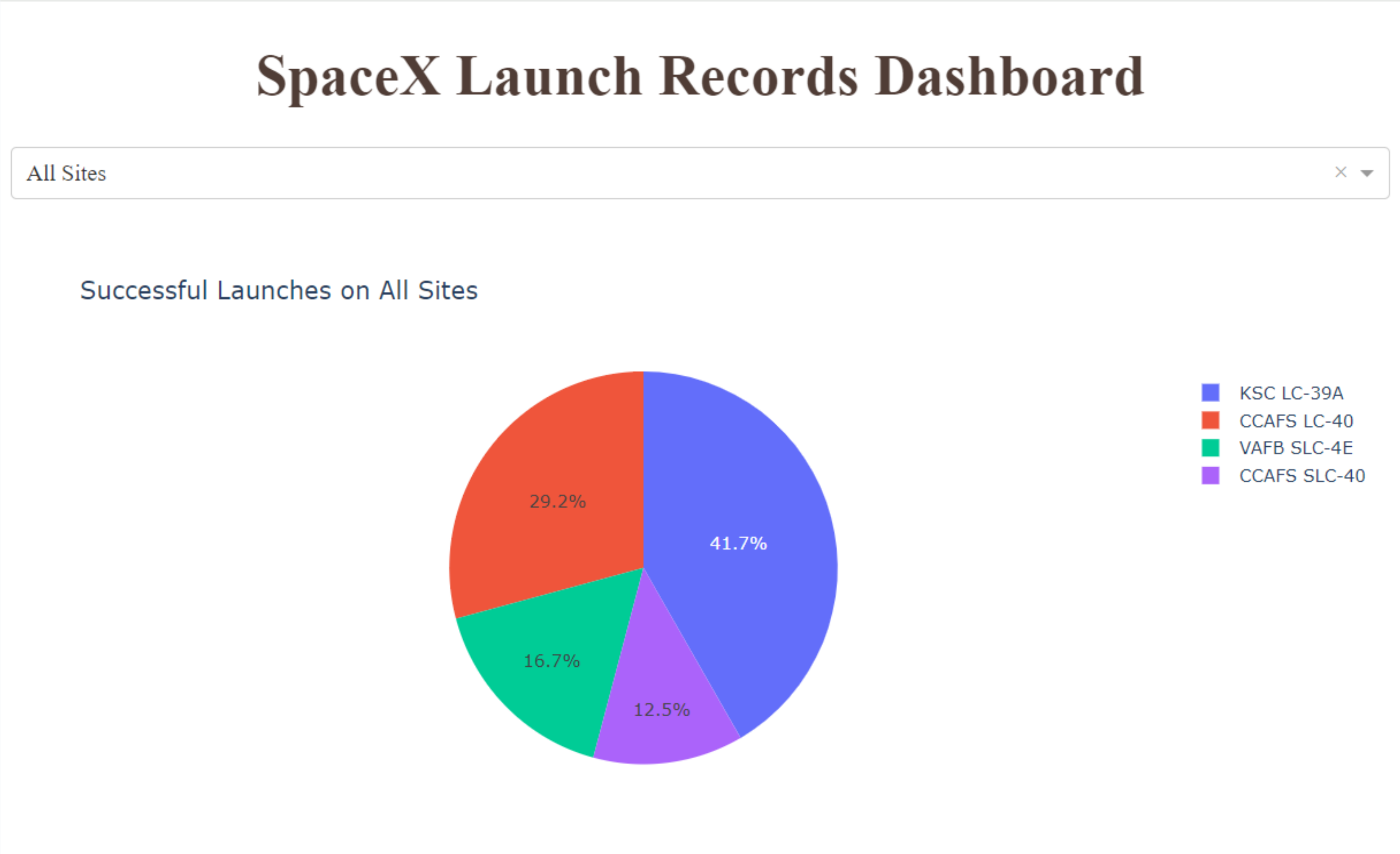
For example, the launch site CCAFS SLC-40 is

- 0.87 km from the coastline
- 1.31 km from the NASA Railroad

Section 4

Build a Dashboard with Plotly Dash

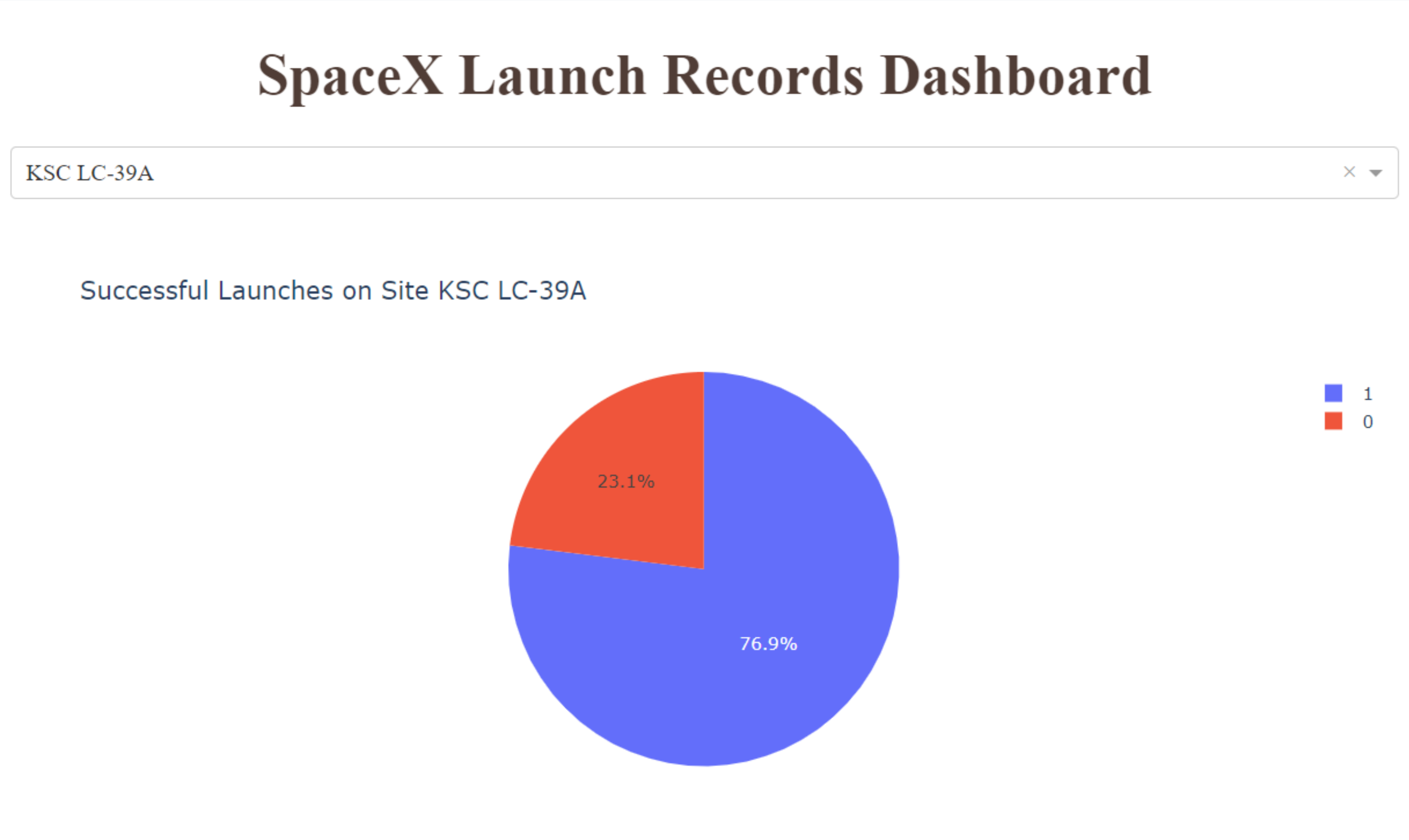
Successful Launches on All Sites



The most successful launches have been performed on the launch site called

Kennedy Space Center's Launchpad 39A.

Successful Launches on Site KSC LC-39A



The launch site KSC LC-39A has got the highest launch success ratio:

- only 23.1% of launches failed
- whereas 76.9% of launches was successful

Payload vs Launch Outcome on All Sites

The scatter plot displaying the Payload vs. Launch Outcome on all sites shows:

- the booster version *FT* has got the highest success rate
- the highest ratio of successful outcomes is between the payload range of 2000 and 5000 kg.

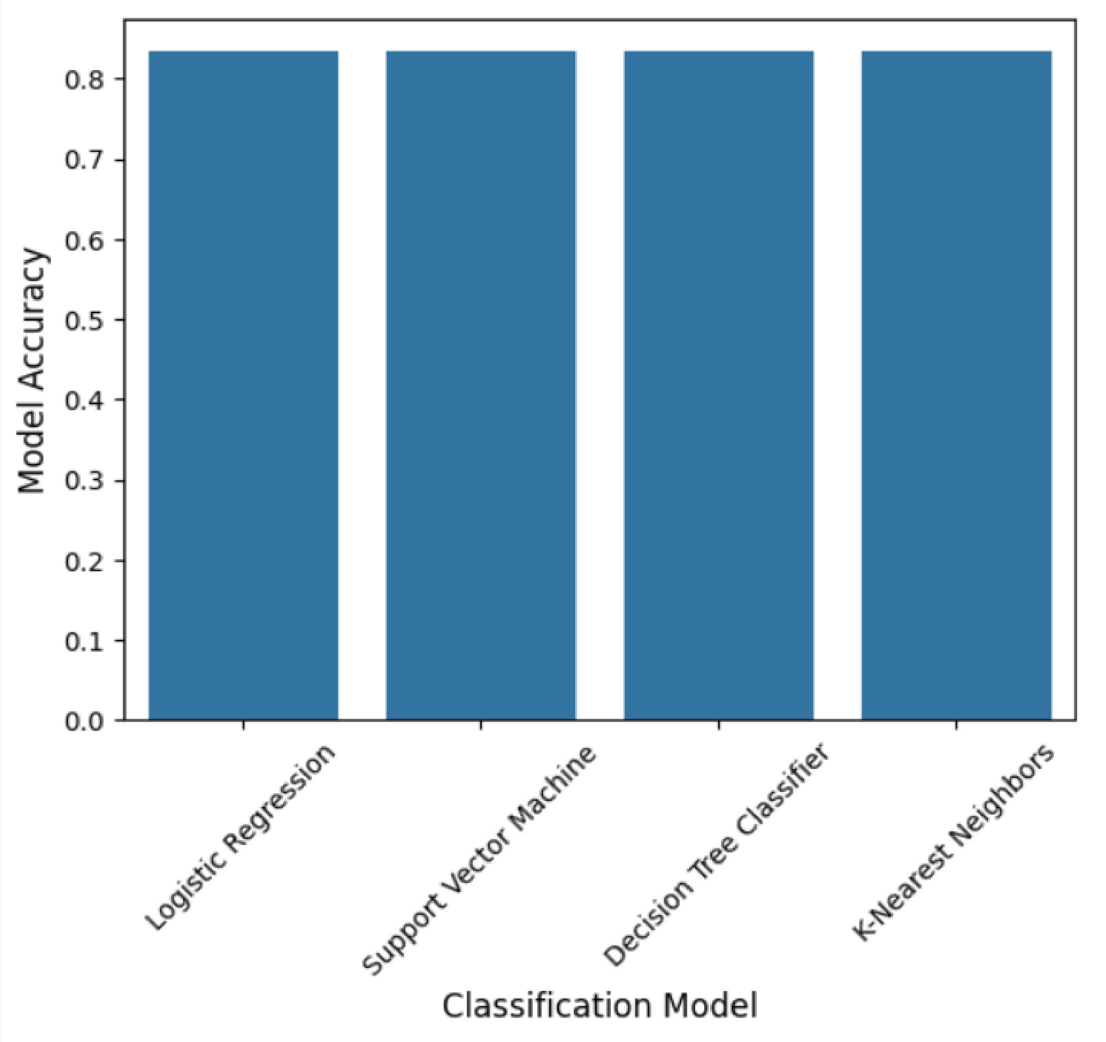
Section 5

Predictive Analysis (Classification)

Classification Accuracy

The classification models built in this lab have got the same accuracy:

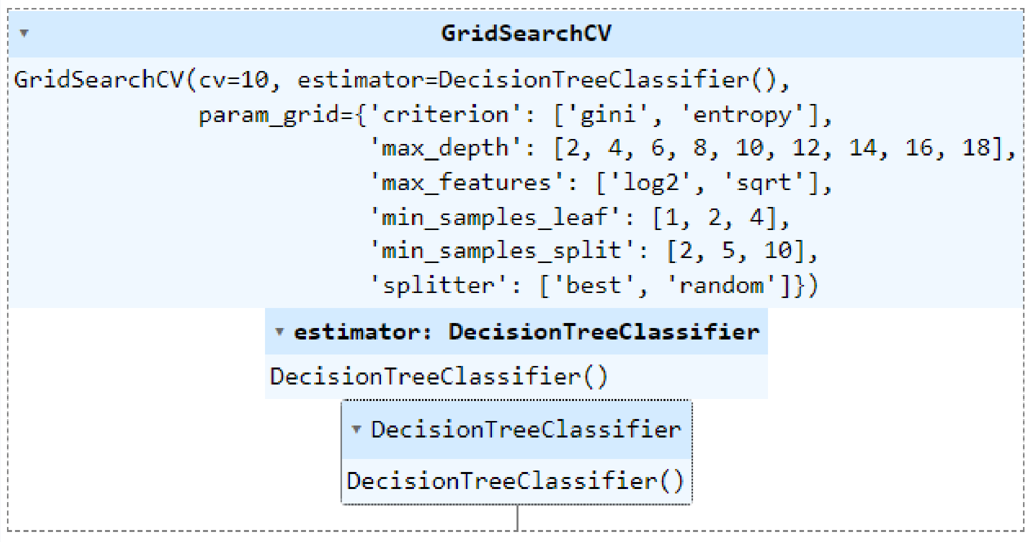
```
Logistic Regression accuracy: 0.8333333333333334
Support Vector Machine accuracy: 0.8333333333333334
Decision Tree Classifier accuracy: 0.8333333333333334
K-Nearest Neighbors accuracy: 0.8333333333333334
```



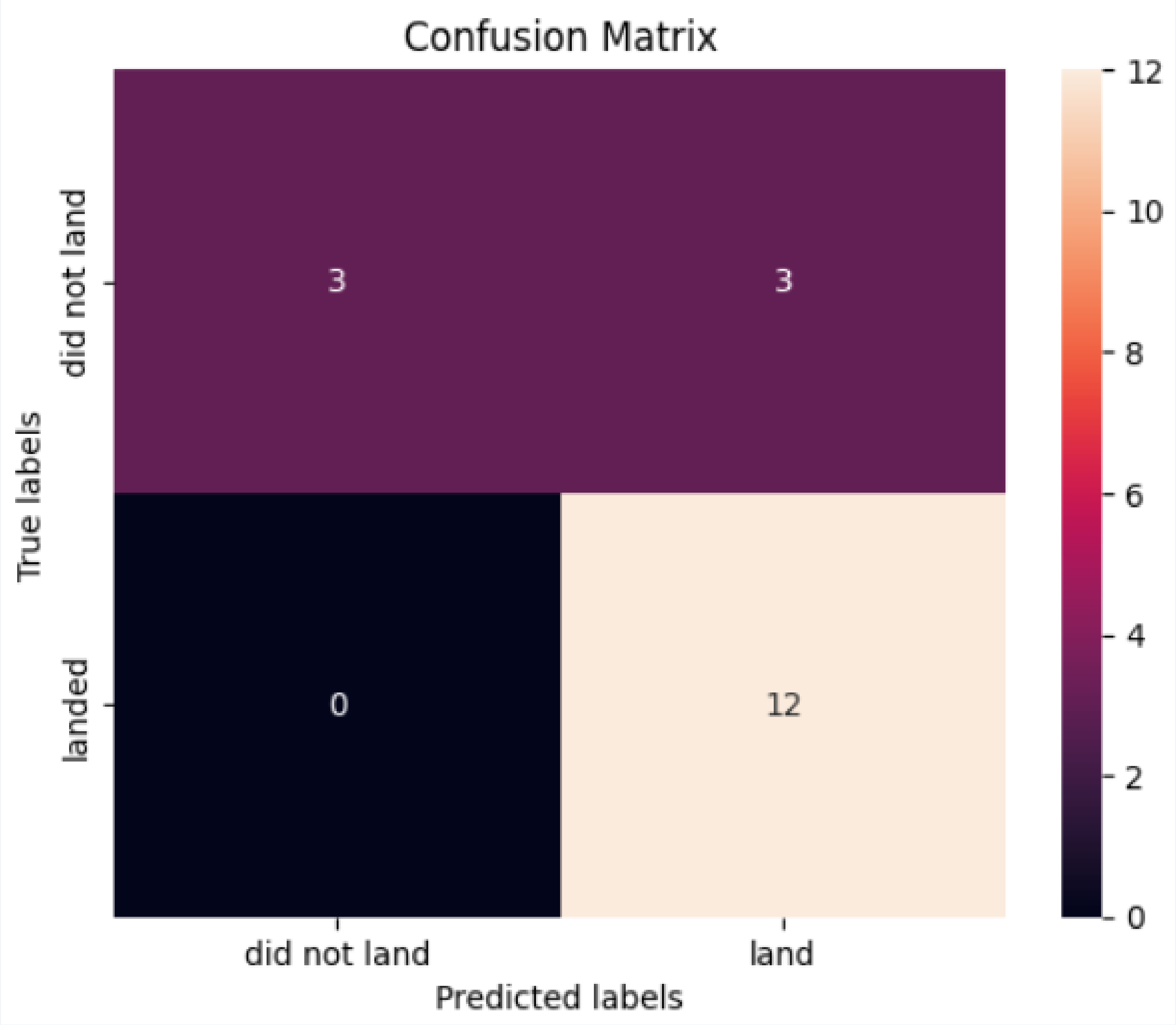
Confusion Matrix

The confusion matrix of the `DecisionTreeClassifier` model shows that

- it predicts well the positive outcomes,
- however, the number of false positives is relatively high



```
GridSearchCV
GridSearchCV(cv=10, estimator=DecisionTreeClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [2, 4, 6, 8, 10, 12, 14, 16, 18],
                         'max_features': ['log2', 'sqrt'],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [2, 5, 10],
                         'splitter': ['best', 'random']})
estimator: DecisionTreeClassifier
DecisionTreeClassifier()
DecisionTreeClassifier
DecisionTreeClassifier()
```



Conclusions

- Success rate and time are positively correlated.
- The highest success rate belongs to
  - the launch site KSC LC-39A with a success ratio of 76.9%
  - the orbits ES-L1, GEO, HEO and SSO
  - the booster version FT
  - the payload range between 2000 and 5000 kg.
- The average payload mass carried by boosters is 6138 kg.
- Rockets with heavy payload are launched both from CCAFS SLC 40 and KSC LC 39A but not from VAFB SLC 4E.
- The classification models built in this lab can predict the mission outcome with an 83% accuracy.

Appendix

Datasets:

- Spacex.xlsx
- spacex_web_scraped.csv
- dataset_part_1.csv
- dataset_part_2.csv
- dataset_part_3.csv
- spacex_launch_dash.xlsx

Thank you!