Winning Space Race with Data Science

Monica Joshi
11/01/2023

Outline

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

Executive Summary

Summary of methodologies

-Data collection through API

-Data collection with Web scrapping

-Data wrangling

-Exploratory data analysis with SQL

-Exploratory data analysis with data visualization

-Interactive visual analystics with folium

-Machine learning prediction

Summary of all results

-Exploratory data analysis result

-Interactive analystics in screenshots

-Predictive analystics result from machine learning lab

Introduction

SpaceX is a revolutionary firm that has disrupted the space sector by offering rocket launches, notably the Falcon 9, for as little as 62 million dollars, whilst other suppliers charge up to 165 million dollars every launch. The majority of these savings are due to SpaceX's brilliant innovation to reuse the first stage of the flight by re-landing the rocket to be utilized on the net mission.Repeating this process will reduce the piece even further. The purpose of this project, as a data scientist for a business competing with Space, is to build a machine learning pipeline to forecast the landing outcome of the first stage in the future.This study will be critical in determining the best pricing to compete against SpaceX for a rocket launch.

Hypothesis:

-Identifying all elements influencing the landing outcome

-the relationship between each variables and its effect on the outcome

-the ideal conditions required to enhance the likelihood of a successful landing

Section 1

Methodology

Executive Summary

Data collection methodology: Data is collected using SpaceX REST API and web scrapping from wikipedia.

Perform data wrangling: Than, Data processed using one-hot encoding for classification feature.

Perform exploratory data analysis (EDA) using visualization and SQL

Perform interactive visual analytics using Folium and Plotly Dash

Perform predictive analysis using classification models

Data Collection

The dataset was collected using REST API and Web Scrapping. The source for data was wikipedia.

REST API: Firstly, Get Request was used. Further, response content were decoded as JSON and converted into pandas dataframe. Furthermore, outlier, missing values were detected and data was made clean.

Web Scrapping: BeautifulSoup was used to extract the launch records as HTML, parse table and convert to pandas dataframe, and then analysis was carried out.

Data Collection – SpaceX API

```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"

In [7]: response = requests.get(spacex_url)
```

```
In [12]: # Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

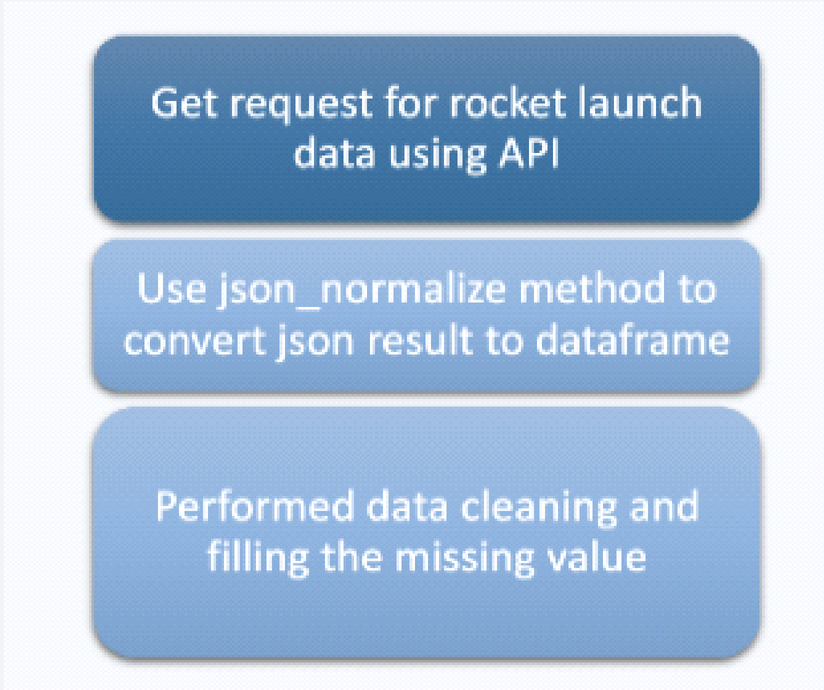
```
In [14]: # Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that h
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the featur
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```



Link: https://github.com/joshimonica/applied-data-science-capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection – Scraping

```python
# use requests.get() method with the provided static_url
# assign the response to a object
data = requests.get(static_url).text
```

Create a `BeautifulSoup` object from the HTML `response`

```python
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data,'html.parser')
```

```python
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
```

Request the Falcon9 Launch Wiki page from url

Create a BeautifulSoup from the HTML response

Extract all column/variable names from the HTML header

link: https://github.com/joshimonica/applied-data-science-capstone/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

- In the dataset, there were several casses where the booster did not land successfully. such as true ocean, true RTLS and true ASDS means the mission has been successful and if all false then mission will fail.
- We need to transform string variables into categorical variables where 1 means the mission has been successful and 0 means the mission was failure.
- https://github.com/joshimonica/applied-data-science-capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()

landing_outcomes
```

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

```
df['Class']=landing_class
df[['Class']].head(8)
```

EDA with Data Visualization

- Scatter graph, bar graph and line graph: Scatter plot show relationship between variables. the relationship is called correlation Bar graphs shows the elationship between numeric and categorical variables and line graph can help us to show global behaviour and make production for unseen data. In this project, flight no. vs payout mass, flight no. vs. launch site, payload vs. launch site, orbit vs. flight no., payload vs orbit type, orbit vspayload mass are example of scatter graph, success rate vs. orbit is example of bar graph and success rate vs year is example of line graph.
- https://github.com/joshimonica/applied-data-science-capstone/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- SQL queries were peformed to gather and understand from dataset:
  - displaying the names of the unique launch sites in the space mission.
  - Display 5 records where launch sites begin with the string CCA
  - display the total payload mass caried by bossters launched by NASA
  - Display average pauload mass caried booster version F9
  - Lsit the data when the first successful la nding outcome in gound pad was achieve.
  - list the total number of successful and failue mission outcomes
  - list the names of the bosster_vesions which have carried the maimum payload mass.
  - https://github.com/joshimonica/applied-data-science-capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Folium map object is map centered on NASA johnson space center at houson Texas.
  - Red circle at NASA johnson space center co-ordinates with label showing its name
  - Red circles at each launch site coodinates woth label showing launch site name
  - the grouping of points in a cluster to display multiple and different information for the same coodinates.
  - markers to show successful and unsuccessful alanding. Green for successful landing and red for unsuccessful landind.
  - markers to show distance between launch site to key location.
  - These objects ae created in order to understand better the problem and the data. we can show easily all launch istes, thier suoundings and the number of usccessful and unsuccessful landings
- https://github.com/joshimonica/applied-data-science-capstone/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_3_lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Dashboard has dopdown, pie chart, rangeslide and scatter plot components.
  - dropdown allows a user to choose the launch sites o all launch sites
  - pie charts shows the total success and total failure fo the launch site chosen with the dropdown componenet.
  - rangeslider allows a user to select a payloa mass in a fixed range
  - scatter chart shows the elationship between two variables, in particular success vs payload mass
- https://github.com/joshimonica/applied-data-science-capstone/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Predictive Analysis (Classification)

- Data preparation
  - load dataset, nomalize data, split data into taining and test sets.
- model preparation
  - Selection of ML algorithm, set parameters fo each algoithm to gridsearchCV, training Gridsearchmodel with taining datasets.
- model evaluation
  - Get best hyperparameters fo ach type of model, compute accuracy for each model with test dataset, plot confusion matrix.
- model comparison
  - Comparison of moels according to their accuracy, the model with the best accuracy will be chosen.
- https://github.com/joshimonica/applied-data-science-capstone/blob/main/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

Exploratory data analysis results

Interactive analytics demo in screenshots

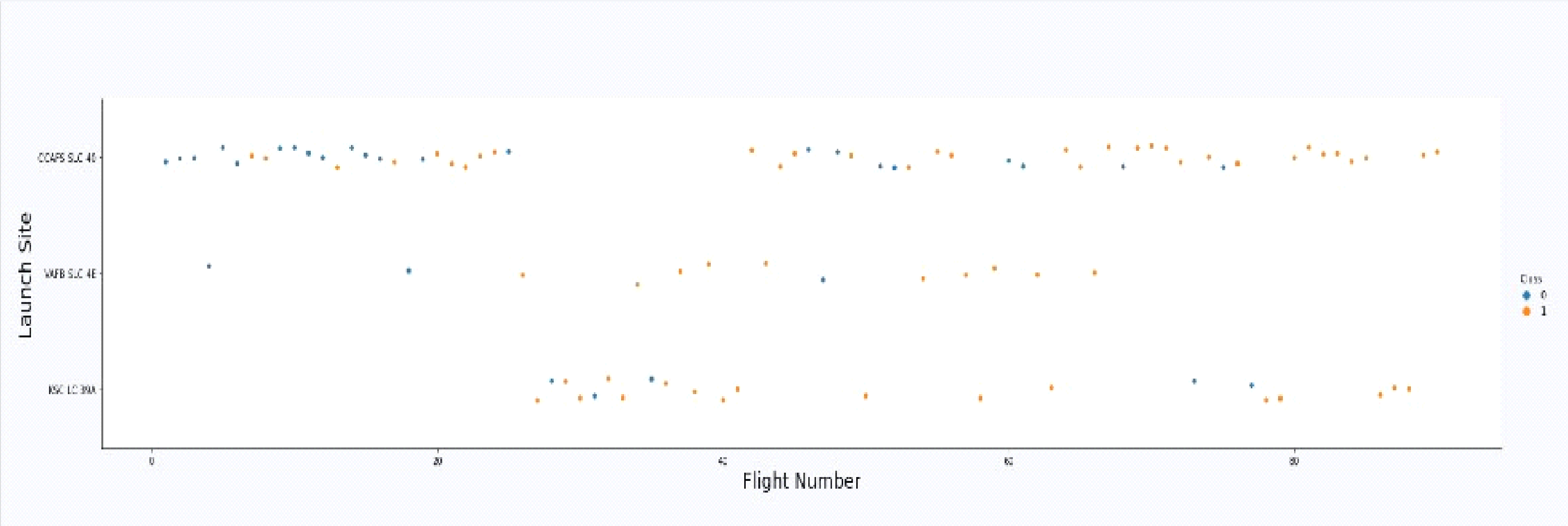
Predictive analysis results

Section 2

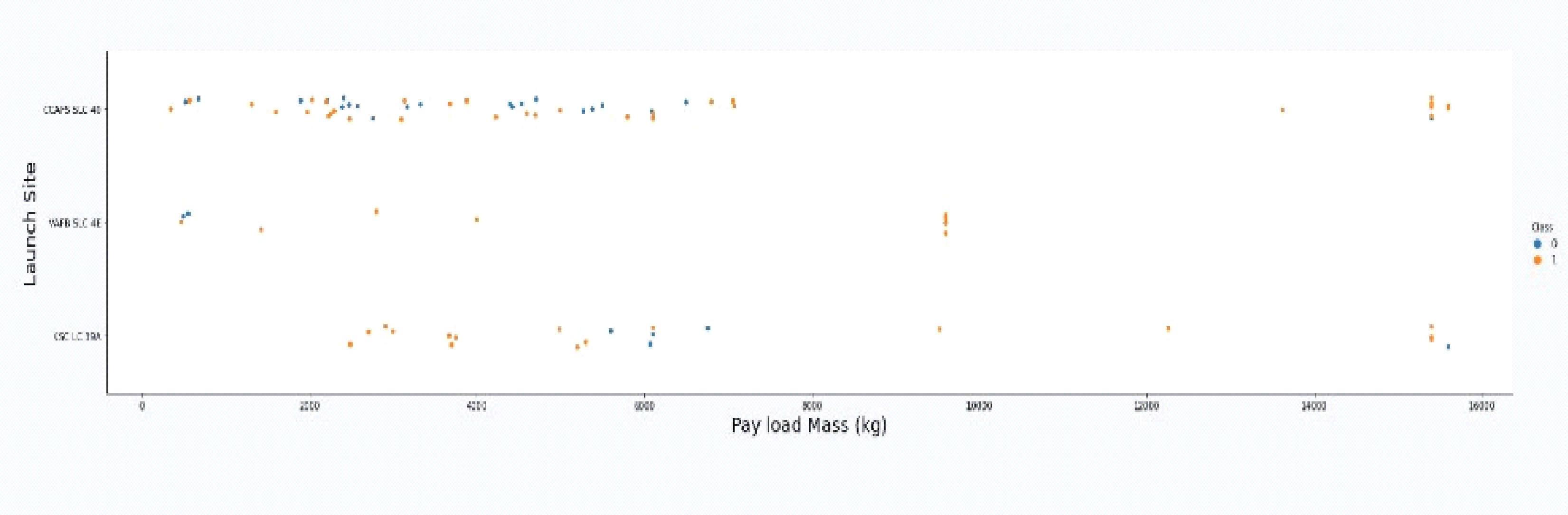
Insights drawn from EDA

Flight Number vs. Launch Site

Success rate is increasing as shown in graph

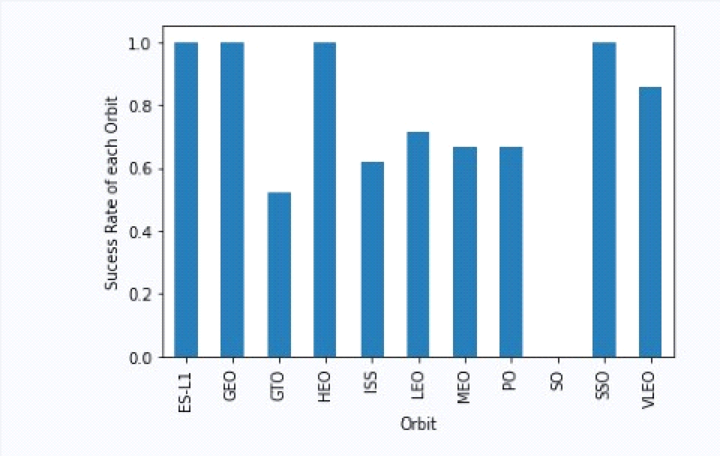


Payload vs. Launch Site



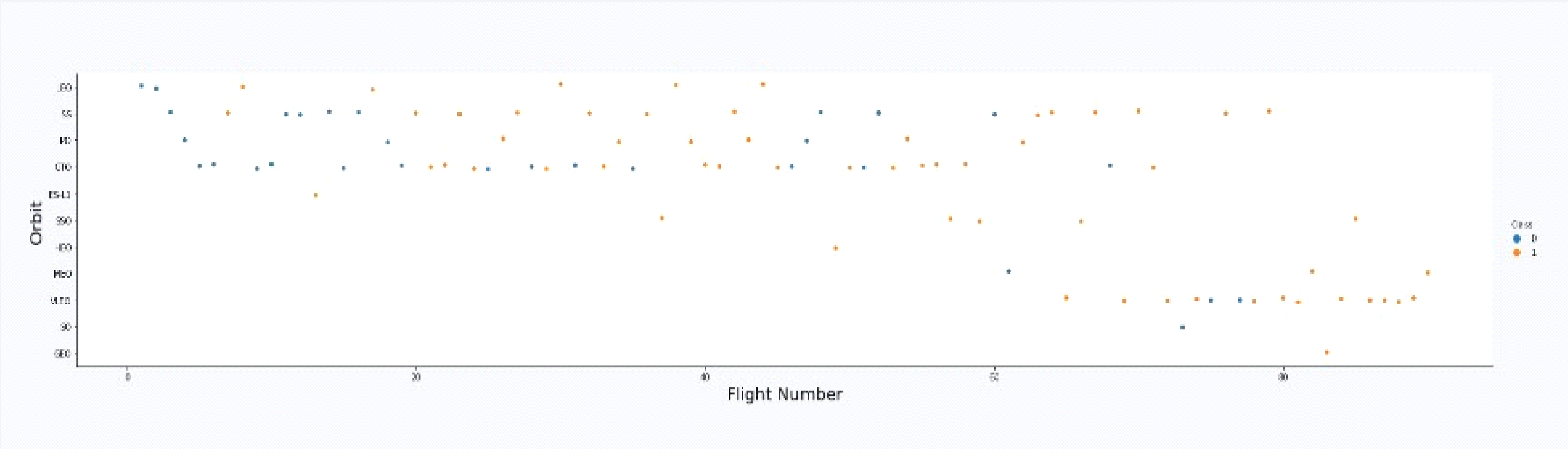
Depending on the launch site, a heavier payload may be a consideration fo a successful landing. On the other hand, a too heavy payload can make a landing failure.

Success Rate vs. Orbit Type



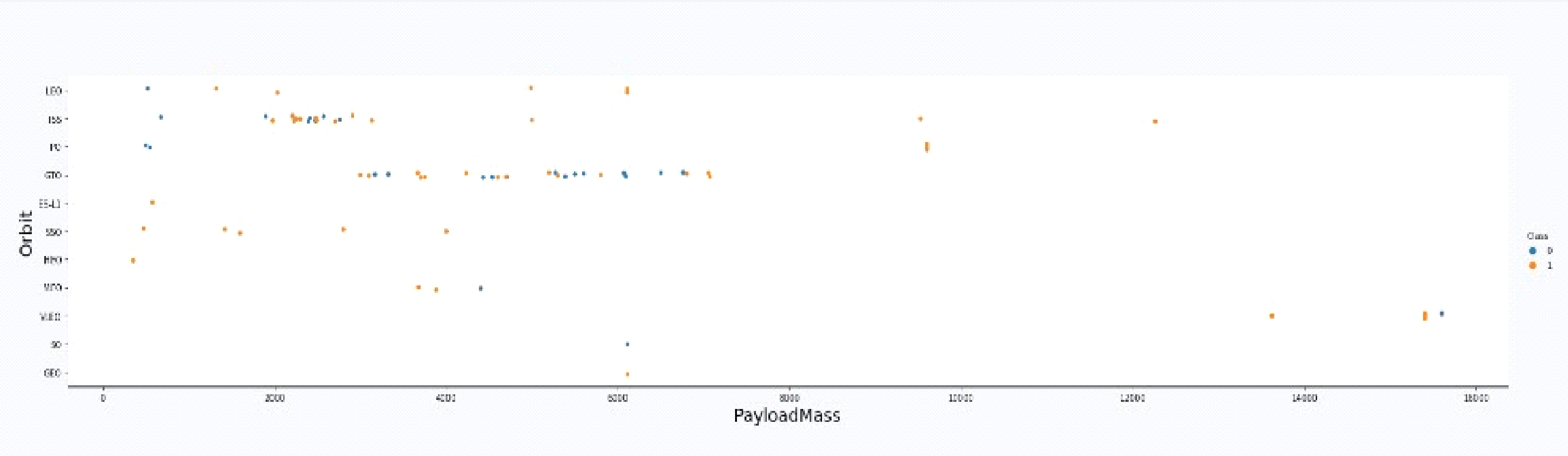
It is observed, success rate for different orbit types. Among them, ES-L1, GEO and HEO, SSO have the better success rate.

Flight Number vs. Orbit Type



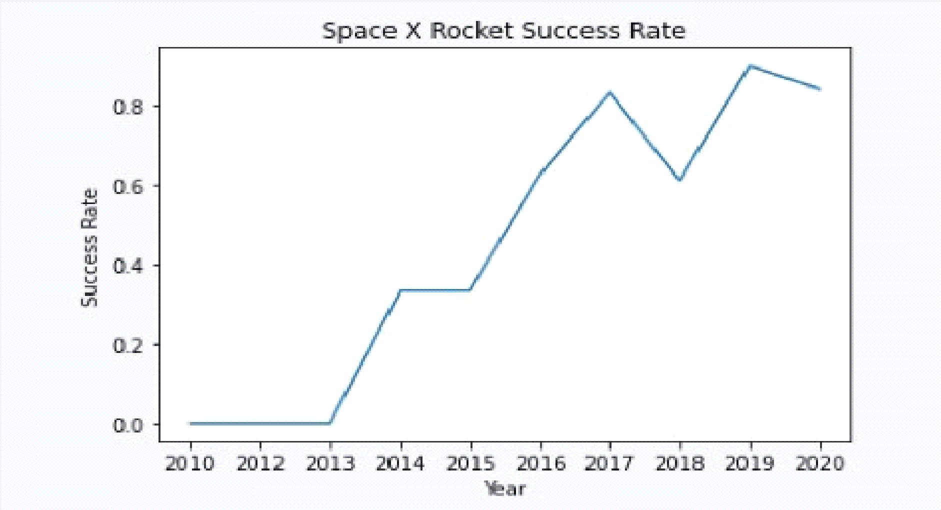
- Here, success rate increases with the number of flight for the LEO orbit. For some orbits like GTO, thee is no relation between the success rate and number of flight. But we can suppose that the high success rate of some orbits like SSO or HEO is due to the knowlegde learned during former launches for other obits.

Payload vs. Orbit Type



- it is seen that the weight of payload plays important role in success ate of launches in certain obits. Heavier payloads improve the sucess rate for theLEO orbit.

Launch Success Yearly Trend



- there is an increment in the spaceX rocket success rate since 2013

All Launch Site Names

SQL Query

```
SELECT DISTINCT "LAUNCH_SITE" FROM SPACEXTBL
```

Results

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

DISTINCT is used in query to remove duplicate launch site

Launch Site Names Begin with 'CCA'

SQL Query

```
SELECT * FROM SPACEXTBL WHERE "LAUNCH_SITE" LIKE '%CCA%' LIMIT 5
```

Results

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
08-12-2010	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
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)

- WHERE and LIKE is used for filtering launch sites that contains the substing CCA. LIMIT is set to 5 that only shows 5 records from filtering

Total Payload Mass

SQL Query

```
SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "CUSTOMER" = 'NASA (CRS)'
```

Results

SUM("PAYLOAD_MASS__KG_")
45596

By this query we get the sum of all payload mass where the customersconsidered as NASA

Average Payload Mass by F9 v1.1

SQL Query	Results
`SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "BOOSTER_VERSION" LIKE '%F9 v1.1%'`	AVG("PAYLOAD_MASS__KG_") 2534.6666666666665

- Average of all payload mass is calculated by considering the booster version contains the substing F9

First Successful Ground Landing Date

SQL Query	Results
`SELECT MIN("DATE") FROM SPACEXTBL WHERE "Landing _Outcome" LIKE '%Success%'`	MIN("DATE") 01-05-2017

- oldest successful landing was selected at first. the WHERE query filter the dataset in orer to keep only records whee landing was successful. using MIN function, we were able to select the record with oldest date.

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL Query

```
%sql SELECT "BOOSTER_VERSION" FROM SPACEXTBL WHERE "LANDING _OUTCOME" = 'Success (drone ship)' \
AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000;
```

Results

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

- The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 is shown in result. WHERE and AND query filter the dataset.

Total Number of Successful and Failure Mission Outcomes

SQL Query

```
%sql SELECT (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Success%') AS SUCCESS, \
(SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE "MISSION_OUTCOME" LIKE '%Failure%') AS FAILURE
```

Results

SUCCESS	FAILURE
100	1

- using the first SELECT, we have showed the subqueries that retun results. the first subquery counts the successful mission. the second subquery counts the unsuccessful mission. The WHERE query followed by LIKE query filters mission outcome. The COUNT query counts recods filtered.

Boosters Carried Maximum Payload

SQL Query

```
%sql SELECT DISTINCT "BOOSTER_VERSION" FROM SPACEXTBL \
WHERE "PAYLOAD_MASS__KG_" = (SELECT max("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

Results

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

- we used a subquery to filter data by eturning on heaviest payload mass with MAX. the main query uses subquery reults and returns unique booster version with the heaviest payload mass.

2015 Launch Records

SQL Query

```
%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'
```

Results

MONTH	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

- This query returns month, booster version, launch site where landing was successful and landing date took place in 2015. substr function process date in order to take month or year. Substr(DATE,4,2) shows month. substr(DATE,7,4) shows year

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

SQL Query

```
%sql SELECT "LANDING _OUTCOME", COUNT("LANDING _OUTCOME") FROM SPACEXTBL\
WHERE "DATE" >= '04-06-2010' and "DATE" <= '20-03-2017' and "LANDING _OUTCOME" LIKE '%Success%'\
GROUP BY "LANDING _OUTCOME" \
ORDER BY COUNT("LANDING _OUTCOME") DESC ;
```

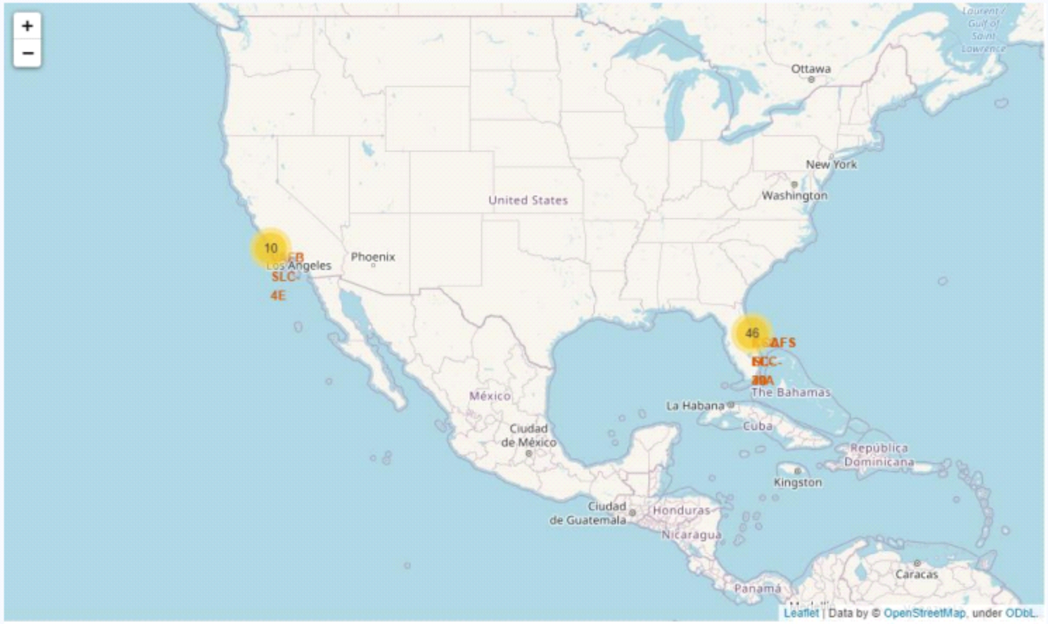
Results

Landing _Outcome	COUNT("LANDING _OUTCOME")
Success	20
Success (drone ship)	8
Success (ground pad)	6

- This query returns landing outcomes and their count where mission was successful and date is between 2010 and 2017. the GROUP BY results landing outcome an ORDER BY COUNT DESC shows results in decreasing order.


Section 3
Launch Sites
Proximities Analysis

Folium Map- Ground station



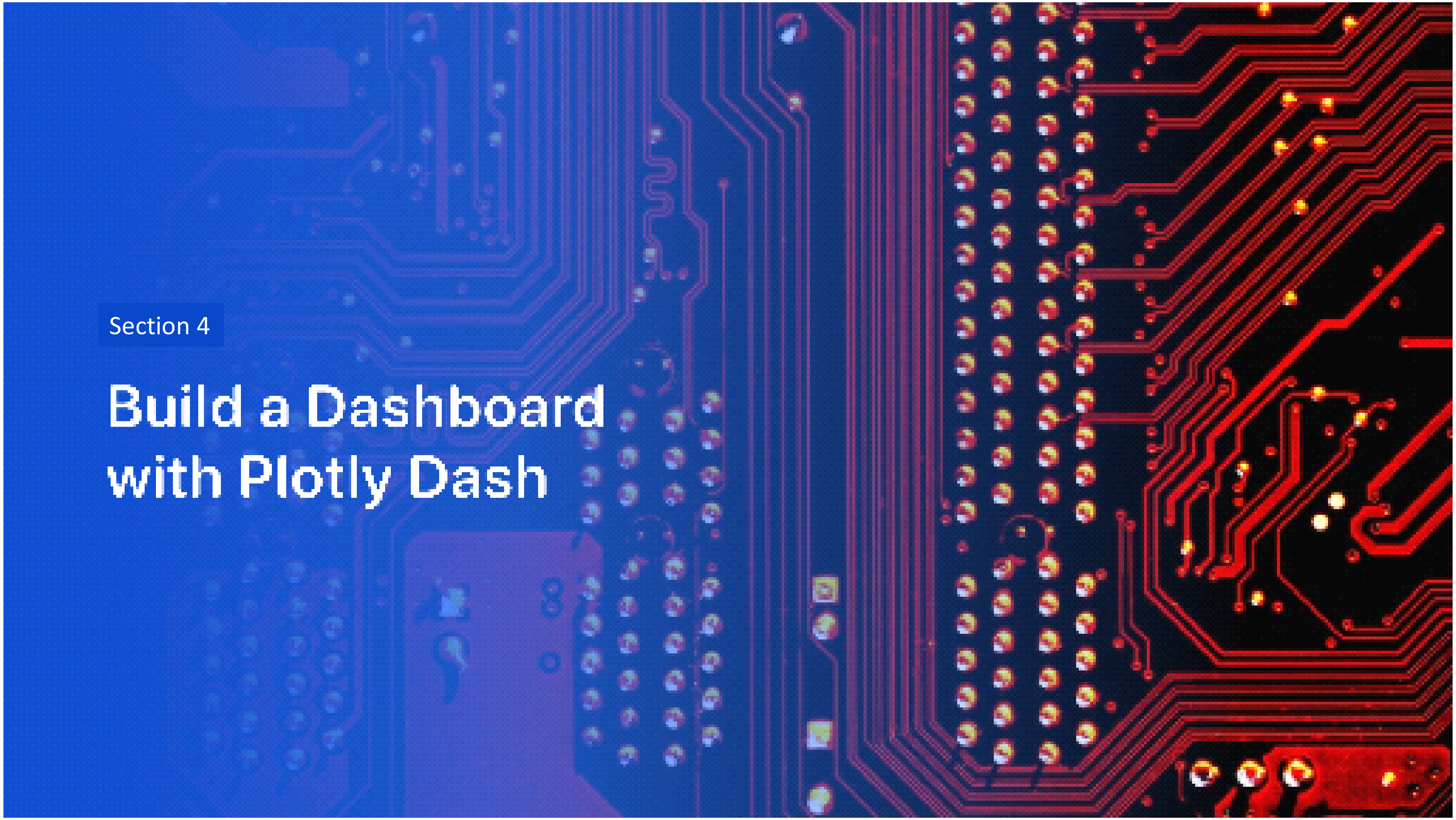
From the map, we can say that SpaceX station is situated on coastal of united states.

Folium Map-colored labeled marker

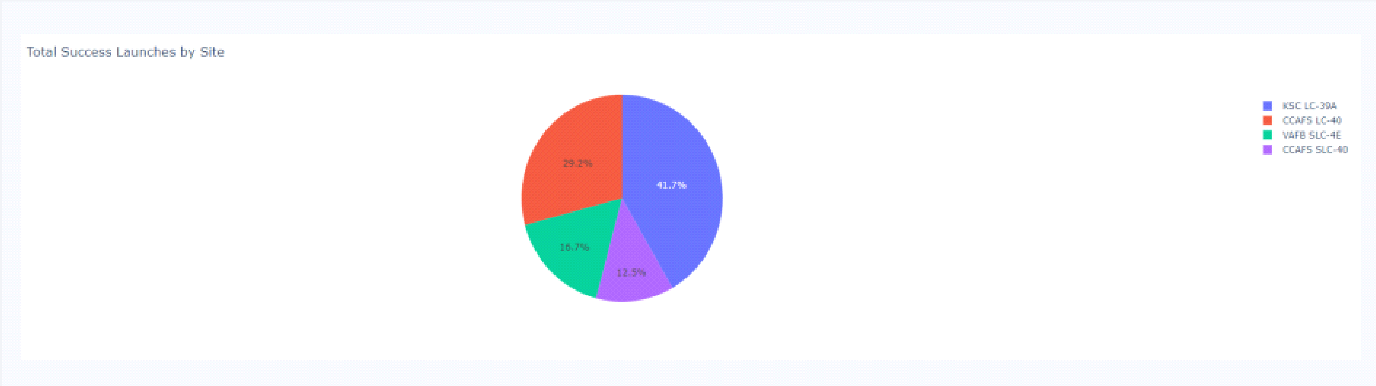
- Green represent success and red represent failure. Here we can see that KSC LC-39A has highest successful launching rate.

Folium Map – Distances between CCAFS SLC-40 and its proximities

- Is CCAFS SLC-40 in close proximity to railways ? Yes
- Is CCAFS SLC-40 in close proximity to highways ? Yes
- Is CCAFS SLC-40 in close proximity to coastline ? Yes
- Do CCAFS SLC-40 keeps certain distance away from cities ? No

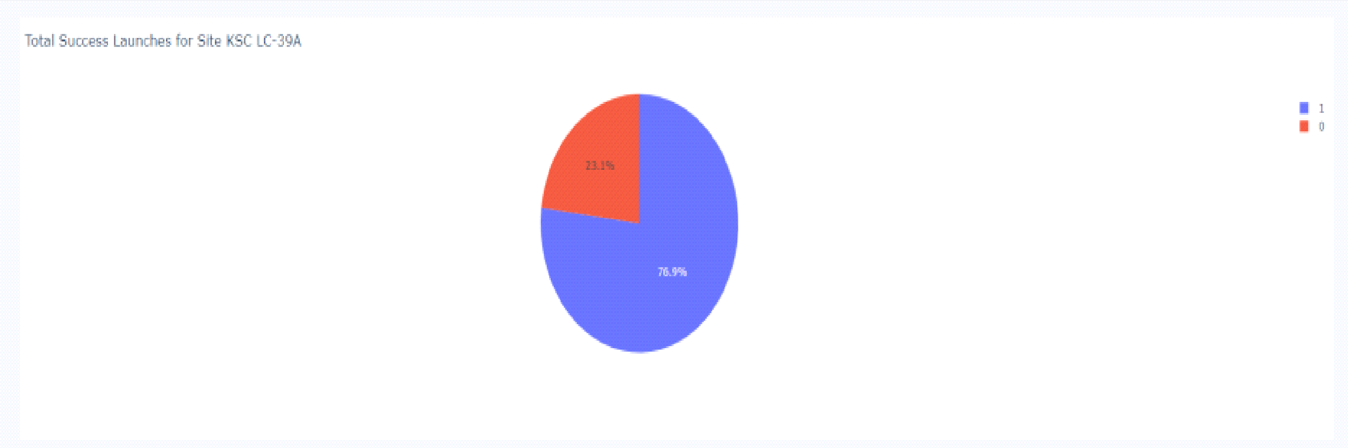

Section 4
Build a Dashboard
with Plotly Dash

Dashboard- total site success



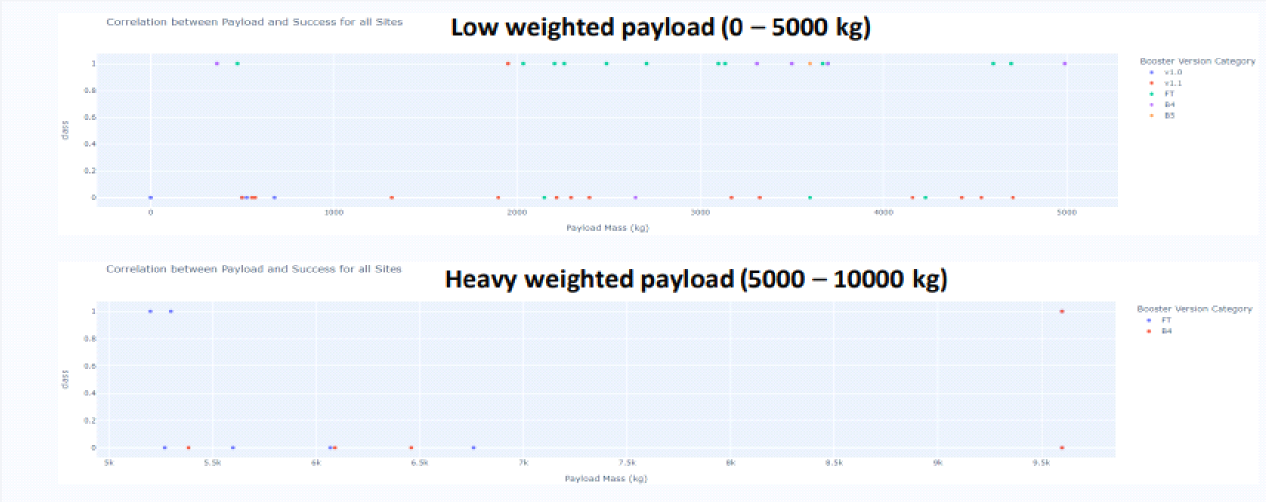
- We see that KSC LC-39A has the best success rate of launches.

Dashboard-Total success launches for Site KSC LC-39A



- We see that KSC LC-39A has achieved a 76.9% success rate while getting a 23.1% failure rate.

Dashboard – Payload mass vs Outcome for all sites with different payload mass selected



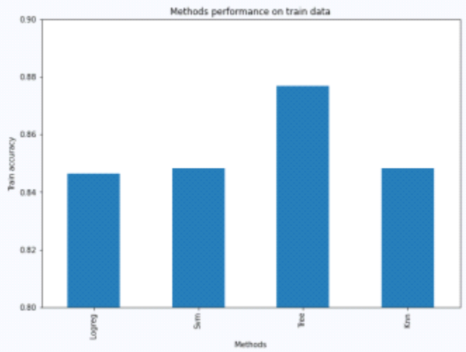
- Low weighted payloads have a better success rate than the heavy weighted payloads.

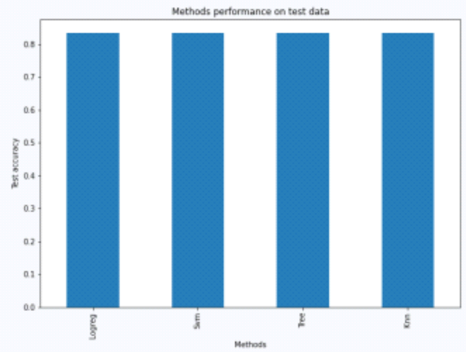
Section 5

Predictive Analysis (Classification)

Classification Accuracy

	Accuracy Train	Accuracy Test
Tree	0.876786	0.833333
Knn	0.848214	0.833333
Svm	0.848214	0.833333
Logreg	0.846429	0.833333



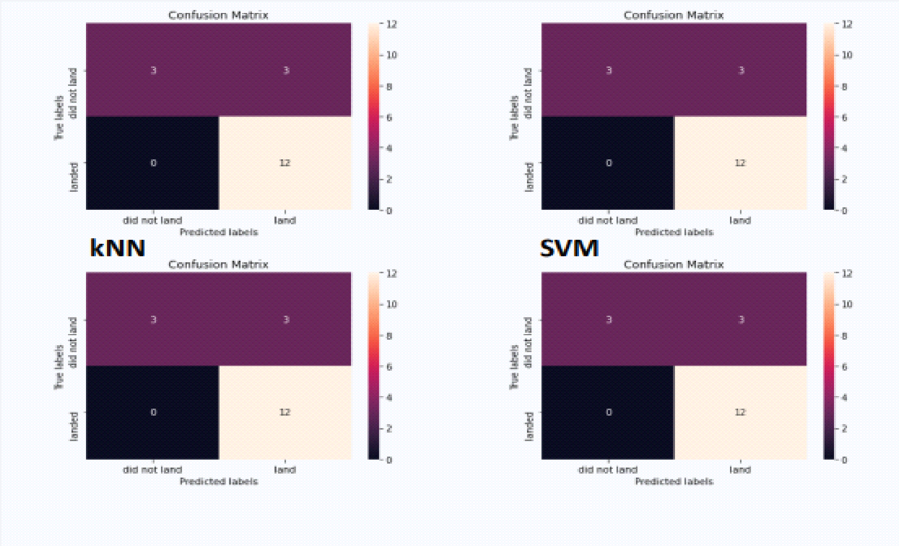


Decision tree best parameters

```
tuned hyperparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf':
4, 'min_samples_split': 2, 'splitter': 'random'}
```

- For accuracy test, all methods performed similar. We could get more test data to decide between
- them. But if we really need to choose one right now, we would take the decision tree

Confusion Matrix



		Actual values 1	Actual values 0
Predicted values	1	TP	FP
	0	FN	TN

- As the test accuracy are all equal, the confusion matrices are also identical. The main problem of these models are false positives.

Conclusions

- The success of a mission can be explained by several factors such as the launch site, the orbit and especially the number of previous launches. Indeed, we can assume that there has been a gain in knowledge between launches that allowed to go from a launch failure to a success.
- The orbits with the best success rates are GEO, HEO, SSO, ES-L1.
- Depending on the orbits, the payload mass can be a criterion to take into account for the success of a mission. Some orbits require a light or heavy payload mass. But generally low weighted payloads perform better than the heavy weighted payloads.
- With the current data, we cannot explain why some launch sites are better than others (KSC LC-39A is the best launch site). To get an answer to this problem, we could obtain atmospheric or other relevant data.
- For this dataset, we choose the Decision Tree Algorithm as the best model even if the test accuracy between all the models used is identical. We choose Decision Tree Algorithm because it has a better train accuracy


Thank you!