IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mariano Fanlo
Dec 2021

Outline

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

Executive Summary

This presentation shows the how a dataset containing information on SpaceX launches was processed utilising different methods to predict successful first stage recovery of rockets.

The results of this work are available on GitHub

Introduction

Project background and context

Falcon 9 first stage successful landing prediction

Problems you want to find answers

What factors affect rocket landing success rate

Correlation between those factors

What is required for Space X to achieve best results

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
- Perform data wrangling
- 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 - Methodology

- This stage involved the collection and processing of data from SpaceX REST API.
- The above-mentioned API includes data about launches, rockets utilised, payload delivered, launch specifications and landing outcome.
- The ultimate goal was to predict the likelihood of successful launches
- An alternative approach involved web scraping data from Wikipedia with BeautifulSoup

Data Collection – SpaceX API

1. Obtain response from API

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

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

2. Convert response into .json

```
In [11]:
# Use json_normalize meethod to convert the json result into a dataframe
response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```

3. Custom functions to clean data

```
# Call getBoosterVersion
getBoosterVersion(data)

# Call getPayloadData
getPayloadData(data)

# Call getLaunchSite
getLaunchSite(data)

# Call getCoreData
getCoreData(data)
```

4. List to DF

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

5. Filter DF and export into .csv

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df.loc[df['BoosterVersion']!="Falcon 1"]

data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```

GitHub URL: https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/0b14ee90-401d-4736-945a-a149f34ac7b6/view?access_token=20cfd8fd0219c8408b3917e2a43687fb3b28be2764998437ecafcc8c87cde160

Data Collection - Scraping

1. Obtain response from HTML

```
# use requests.get() method with the provided static_url
# assign the response to a object
page = requests.get(static_url)
page.status_code
```

2. Create BeautifulSoup object

```
soup = BeautifulSoup(page.text, 'html.parser')
```

3. Find tables

```
html_tables = soup.find_all('table')
```

4. Obtain col names

```
column_names = []
temp = soup.find_all('th')
for x in range(len(temp)):
    try:
     name = extract_column_from_header(temp[x])
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
     pass
```

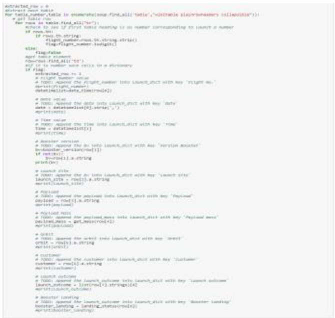
5. Create Dictionary with obtained col names

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6. Append data



7. Convert Dictionary to DF and then DF to csv

```
df=pd.DataFrame(launch_dict)
```

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

GitHub URL: https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/41b02990-b323-429e-a657-69992f711ca7/view?access_token=ee298bc49a1e237ea439fe7f33eb783ac20dcbfdae3e46897be29e56887143d4

Data Wrangling

- Calculate number of launches at each site

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

- Calculate number and occurrence of each orbit

```
# Apply value_counts on Orbit column
df["Orbit"].value_counts("Orbit")
```

- Number and occurrence of mission outcome per orbit type

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

- Create landing outcome label from Outcome Column

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []
for key,value in df["Outcome"].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

- Workout success rate for every landing in dataset

```
In [13]: df["Class"].mean()
Out[13]: 0.6666666666666666
```

- Export data set in .csv format

```
df.to_csv("dataset_part\_2.csv", index=False)
```

GitHub URL: https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/8a62dd57-54ab-4ddf-8030-d76ed716385a/view?access_token=efc9ffa1f004dae56c894b6deaa04084c270ffbf9437f6f884128e1096db2274

EDA with Data Visualization

Scatter Plot

- Flight Number vs. Payload Mass
- Flight Number vs. Launch Site
- Payload vs. Launch Site
- Flight Number and Orbit
- Orbit vs Playload Mass

This type of graph shows the relationship between two variables to understand how much one is affected by the other. In other terms, how those variables correlate within each other.

Bar Graph

Orbit vs. Mean

This type of chart is ideal to compare different groups or track anything over time. In this case, it helped us to visualize which orbits had the best success rates.

Line Graph

Success rate per year

Ideal to show trend, this graph helped us to visualize how success rate has evolved over time.

GitHub URL: https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/053739ea-3446-430f-a7a2-1b422f620ff9/view?access_token=d50e1389f57dc68f974d21e2a7a2143c49c7e37440f3fdc03385158bb2020cc2

EDA with SQL

This stage involved executing SQL queries to obtain information from the dataset.

Below there is a summary of the ran queries:

- Display 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 carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was acheived
- List the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
- List total number of successful and failure mission outcomes
- List names of the booster_versions which have carried the maximum payload mass.
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Rank the count of successful landing_outcomes between the date 2010 06 04 and 2017 03 20 in descending order.

GitHub URL: https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/7ff0d763-1777-40ec-bf96-88f772649059/view?access_token=c0430bc5d41df828a8555fcf3efc2ee16755a0b42ecefb7d154a8227f01656d0

Build an Interactive Map with Folium

Mark all launch sites on a map

Indicated each launch site with a circle marker and a label with their corresponding names, based on the coordinates.

Mark the success/failed launches for each site on the map

Assigned red and green markers on the map based on outcome (0 failure – 1 success respectively).

Calculate the distances between a launch site to its proximities

Calculated the distance from the launch site to several points

GitHub URL: https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/2644955a-6699-4756-aef3-8b2c9f6d6d05/view?access_token=d0bcfdc79fd56bacca3f10d8e6da27ad26ec8bcb49272ac9c15b2e846c09058f

Predictive Analysis (Classification)

This stage involved the following steps:

1. Building the model

Load DF and transform the data
Split the data into training and testing data using the function train_test_split. The training data is divided into validation data, a second set used for training data.
We count the samples
Select machine learning algorithm and set parameters
Fit datasets into the GridSearchCV objects and train dataset

2. Evaluation

Assess each model based on their accuracy.
Obtained tuned hyperparameters
Plot Confusion Matrix

3. Improvement

Tuning algorithms

4. Best Performing Classification

Selection based on the corresponding accuracy score.
The notebook contains a dictionary named parameters

GitHub URL: https://dataplatform.cloud.ibm.com/analytics/notebooks/v2/f0ee8ef8-dbb6-4834-9135-24d50967c215/view?access_token=488124002c52541ab95d8ea92d8f0e689345df2a085f93cfef0902ace60df307

Results

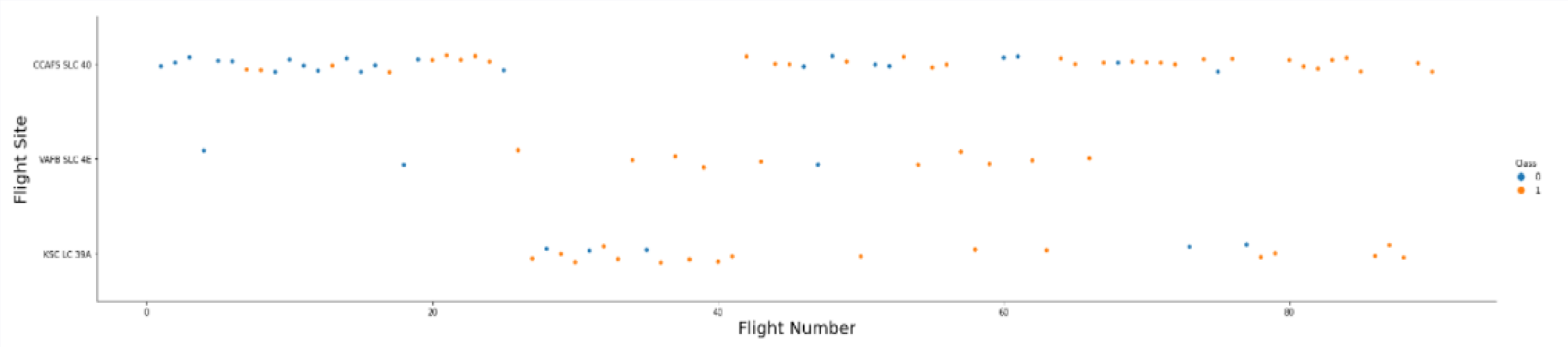
- Exploratory data analysis results
  Improvement in the success rate from 2013 to 2020.

- Predictive analysis results
  The Tree is the best method with a score of ~0.89

Section 2

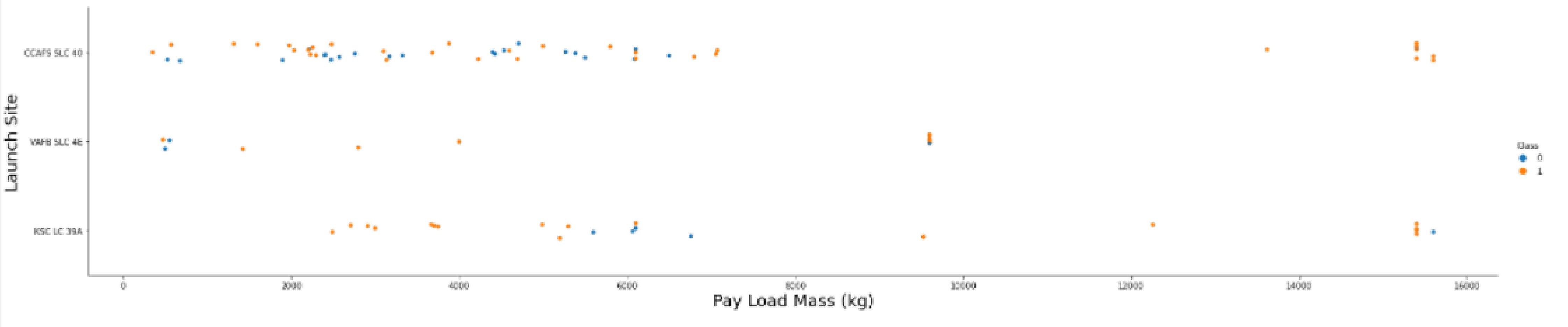
Insights drawn from EDA

Flight Number vs. Launch Site



- There is a positive relationship between number of flights and the success rate at each launchsite
- KSC LC-39A and VAFB SLC 4E launch sites have a higher proportion of successful (Class 1) launches

Payload vs. Launch Site

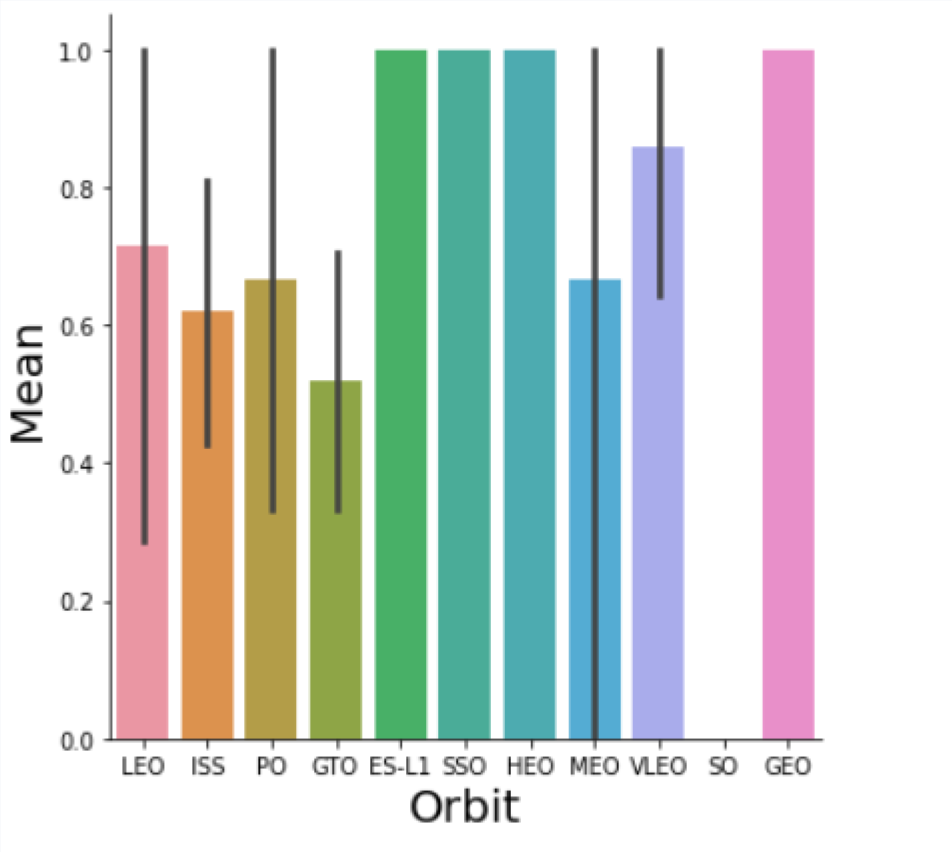


- Positive correlation between Pay Load Mass and success rate.

Success Rate vs. Orbit Type

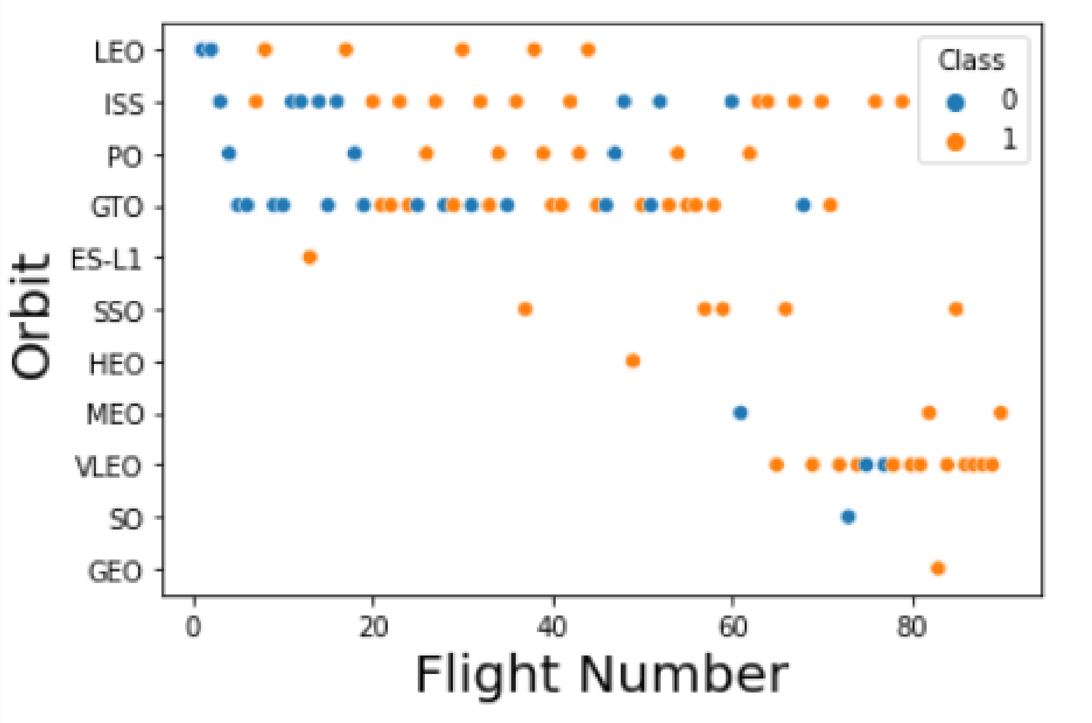
Success rate by orbit:

- ES-L1, GEO, HEO and SSO have 100%
- VLEO has >80%
- LEO has ~70%
- ISS, PO and MEO have >60%
- GTO has >50%
- SO has 0%



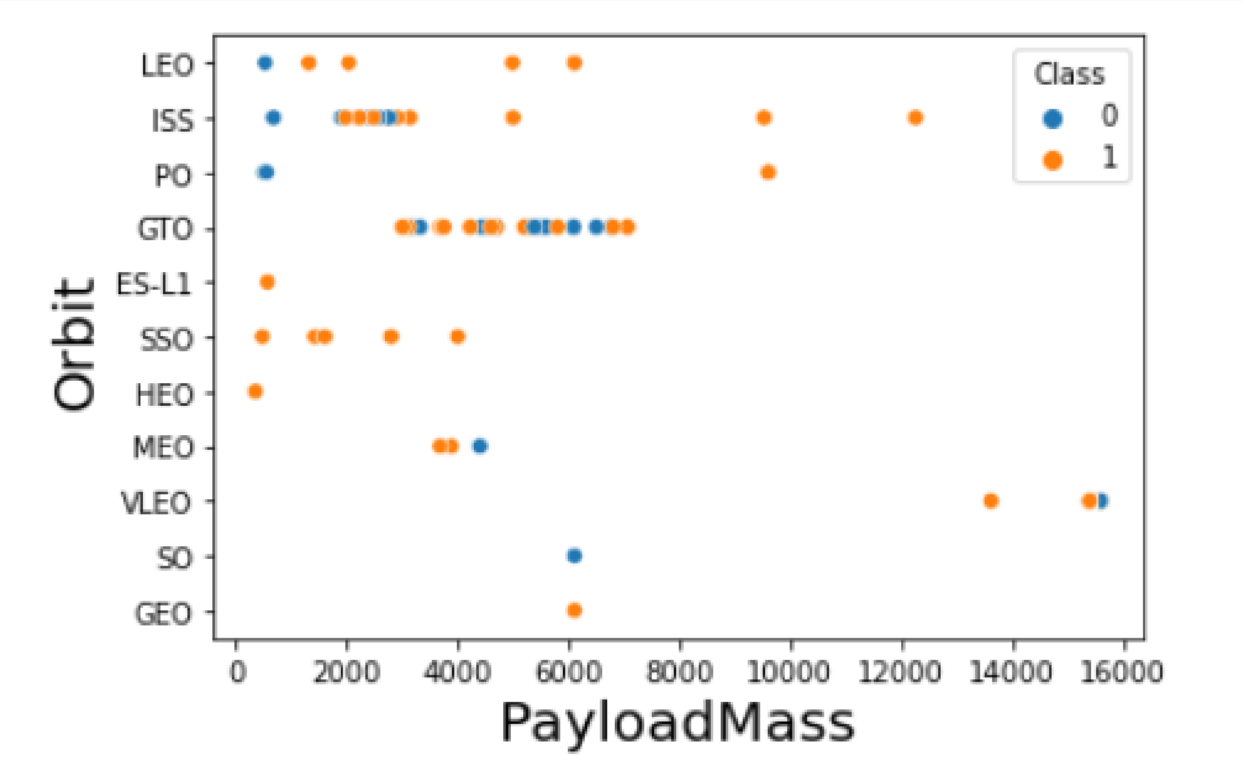
Flight Number vs. Orbit Type

- No apparent correlation between orbit and number of flights regarding success rate



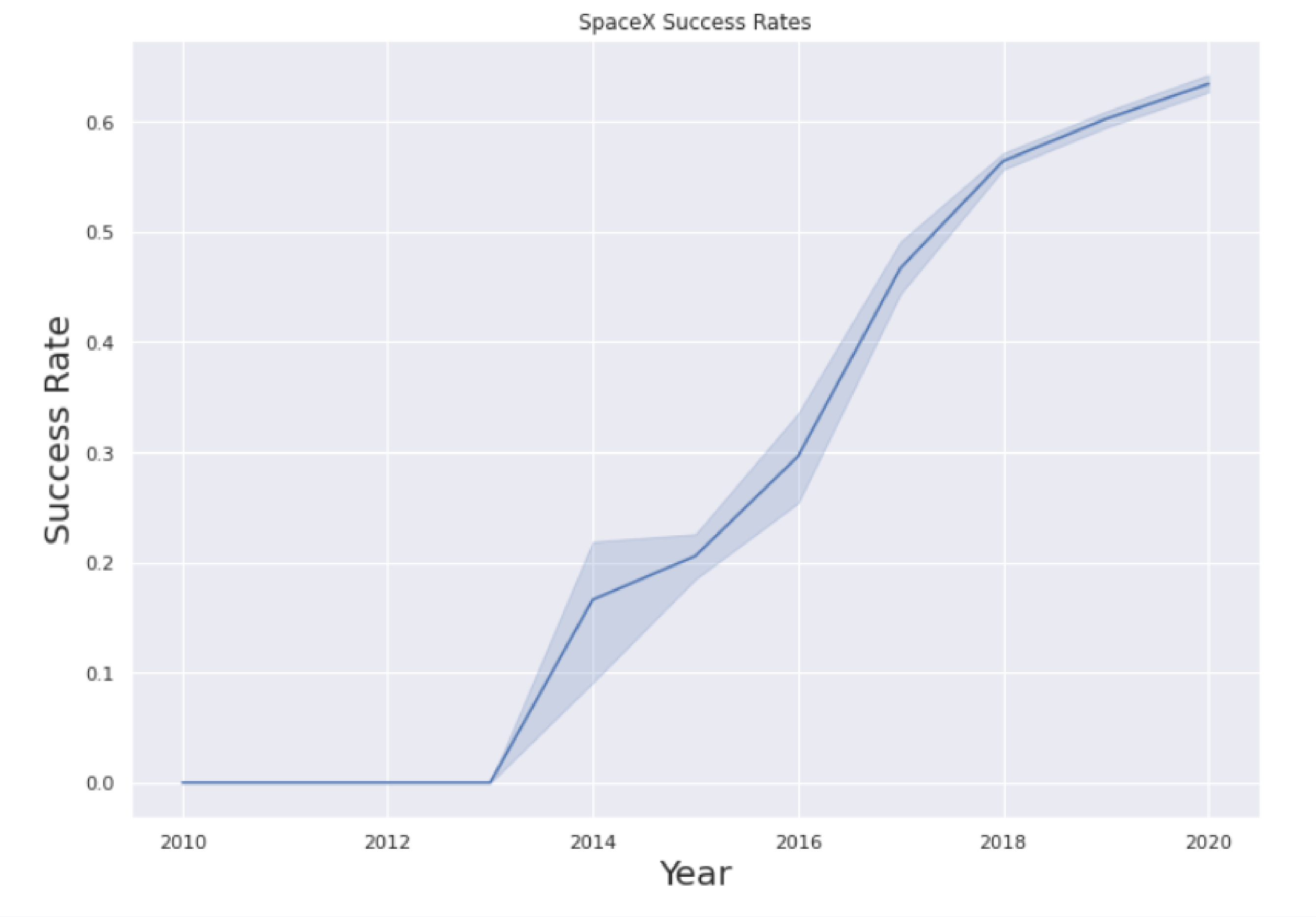
Payload vs. Orbit Type

- Positive correlation between the payload weight and success rates particularly for PO, LEO and ISS.
- Unclear correlation for GTO.



Launch Success Yearly Trend

- Upward trend in success rate since 2013



All Launch Site Names

- We have identified 4 unique launch sites

```
%sql SELECT DISTINCT(launch_site) FROM SPACEX
```

```
 * ibm_db_sa://bzm16693:***@2f3279a5-73d1-4859-88f0-a6c3e
Done.
```

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

Launch Site Names Begin with 'CCA'

```
%sql SELECT * FROM SPACEX 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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

Total Payload Mass

```
%%sql SELECT SUM(payload_mass__kg_) as T0talPayLoadMass_kg FROM SPACEX
WHERE CUSTOMER='NASA (CRS)'
```

t0talpayloadmass_kg
45596

Average Payload Mass by F9 v1.1

```
%%sql SELECT AVG(payload_mass__kg_) AVE_PayLoadMass_kg FROM SPACEX
WHERE booster_version LIKE 'F9 v1.1%'
```

ave_payloadmass_kg
2534

First Successful Ground Landing Date

```
%%sql SELECT MIN(DATE) AS FIRST_SUCCESSFUL_LANDING_OUTCOME_IN_GROUND_PAD_DATE FROM SPACEX
WHERE landing__outcome = 'Success (ground pad)'
```

first_successful_landing_outcome_in_ground_pad_date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql SELECT booster_version, landing__outcome,  payload_mass__kg_ FROM SPACEX
WHERE landing__outcome = 'Success (drone ship)'
AND payload_mass__kg_ BETWEEN 4000 AND 6000
```

booster_version	landing__outcome	payload_mass__kg_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT mission_outcome, COUNT(mission_outcome) AS count FROM SPACEX GROUP BY mission_outcome
```

mission_outcome	COUNT
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

```
%%sql SELECT DISTINCT(booster_version) FROM SPACEX
WHERE payload_mass__kg_ = (SELECT MAX(payload_mass__kg_) FROM SPACEX)
```

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

```
%%sql SELECT landing__outcome, booster_version, launch_site  FROM SPACEX
WHERE landing__outcome = 'Failure (drone ship)'
AND YEAR(DATE) = 2015
```

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

```
%%sql SELECT landing__outcome, COUNT(landing__outcome) AS COUNT
FROM SPACEX
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY landing__outcome
ORDER BY COUNT DESC
```

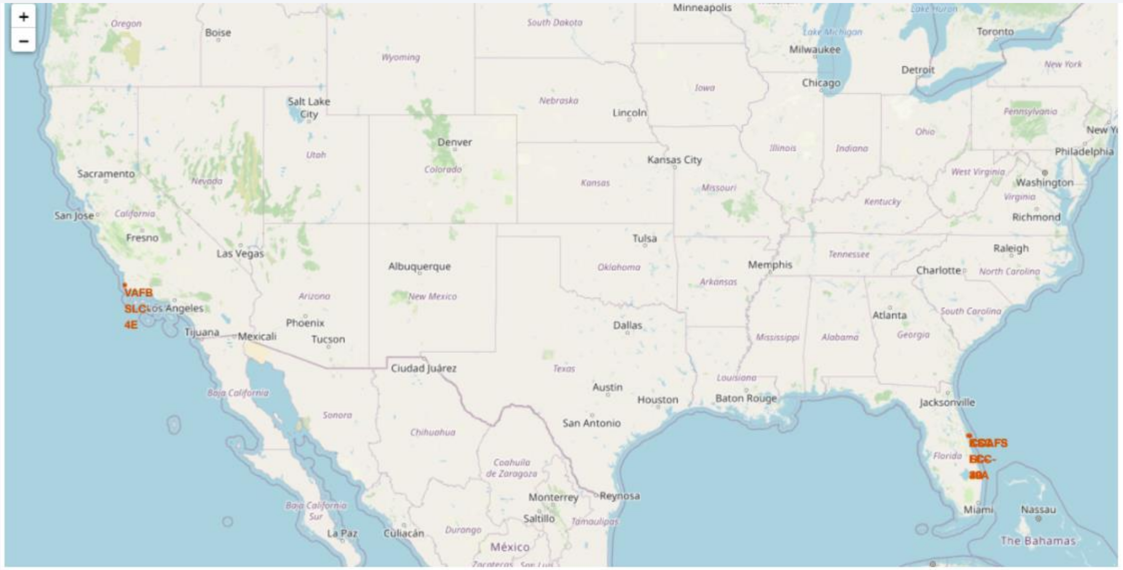
landing__outcome	COUNT
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 4

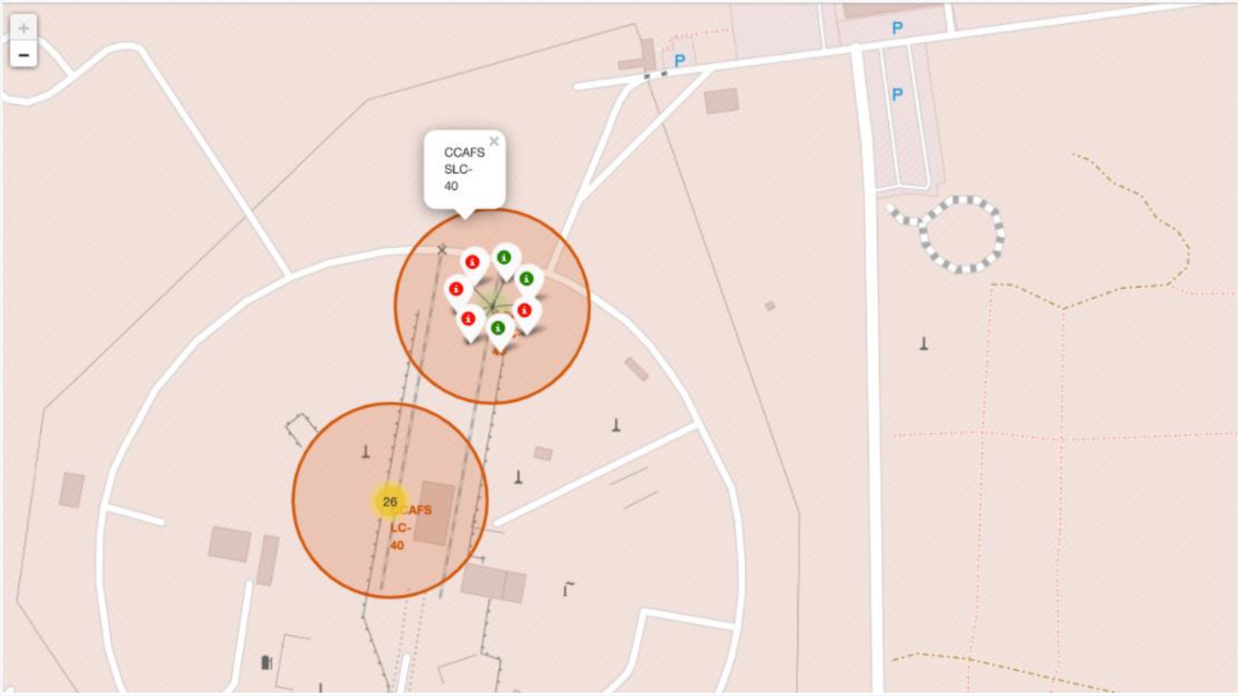
Launch Sites Proximities Analysis

<Folium Map Screenshot 1>

- Launch sites in both coastlines of the USA, in California and Florida.



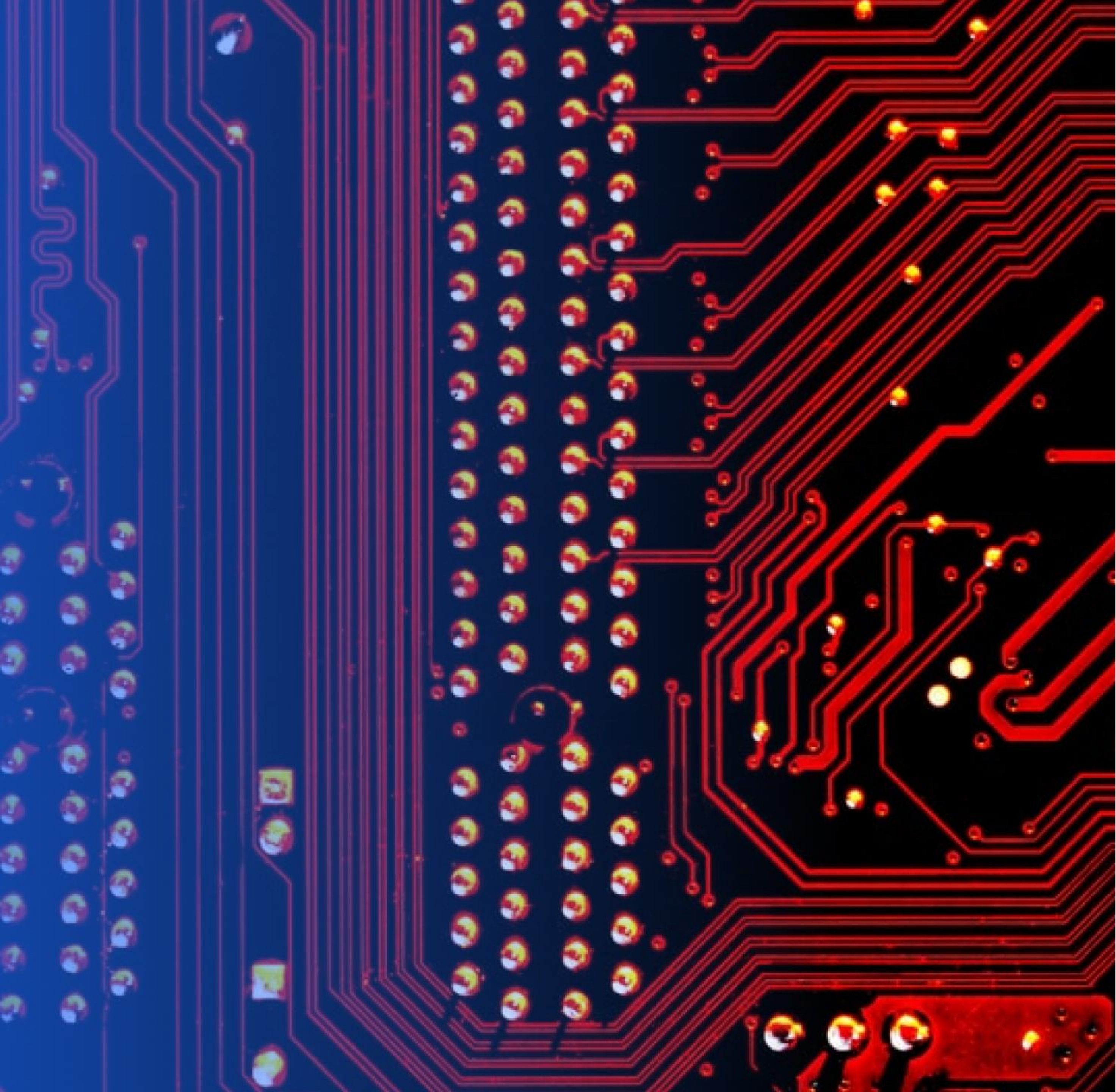
<Folium Map Screenshot 2>



<Folium Map Screenshot 3>

Section 5

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.

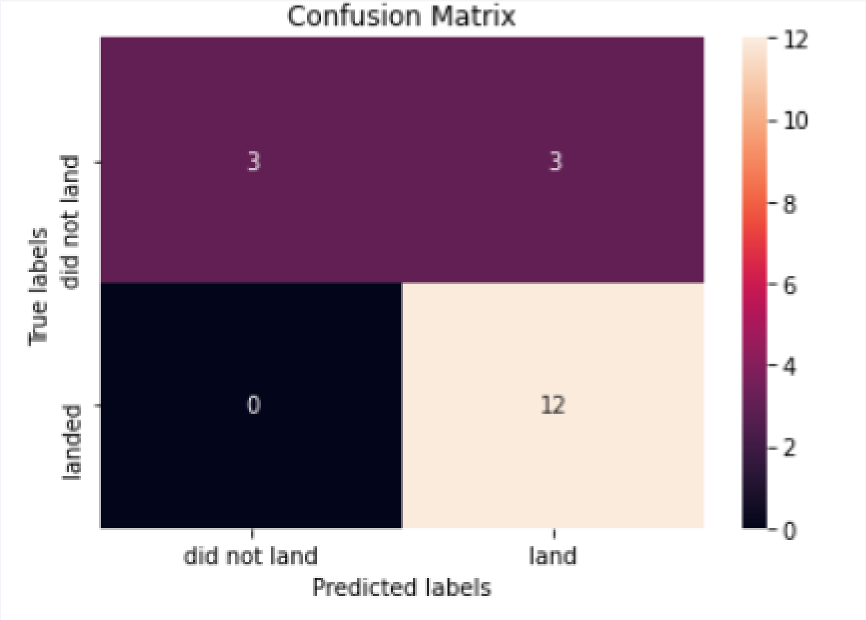
Section 6

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

Confusion Matrix



- Consistent confusion matrix across all models
- False positives (which lead to over prediction) are a major issue

Conclusions

- The Decision Tree model is the best classifier algorithm
- Upward trend in success rate across the period reviewed
- KSC LC 39A is deemed as the site with highest success rate
- Orbits GEO, HEO, SSO and ES-L1 are deemed as having the highest success rate

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

Thank you!