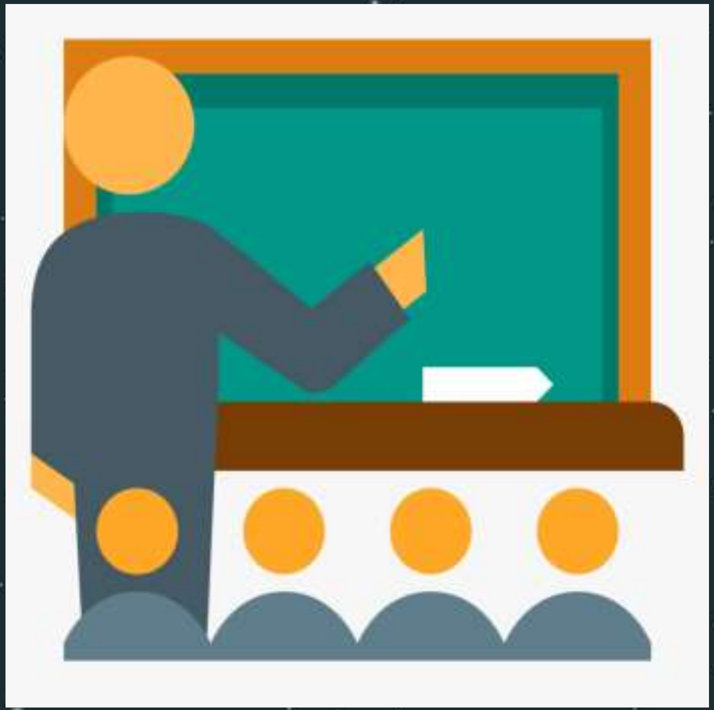
Fábio Souza da Silva

December 10, 2022

Outline



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

Executive Summary

Methodologies

- Data Collection from SpaceX API and Web scraping
- Exploratory Data Analysis (EDA)
- Prediction Machine Learning Tools

Main results

- Exploratory Data Analysis
- Interactive Analysis
- Predictive Analysis

Introduction

SpaceX designs, manufactures and launches advanced rockets and spacecraft. The company was founded in 2002 to revolutionize space technology

Using public information, as well machine learning tools, we are predicted the best place to make launches

Methodology

Data Collection

- SpaceX API
- Web Scrapping Wikipedia

Data Wrangling

- Filtering the data
- Dealing with missing values
- To label the date label based on outcome date after summarizing and analyzing features.

Perfomed exploratory data analysis (EDA) using visualization and SQL
Perfomed interactive visual analysis using Foliu and Plotly Dash
Perfomed predictive analysis using classification models.

Data Collection – SpaceX API

1) To get response from API and covert the results to .json file

```python
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.clo
response.status_code
# Use json_normalize meethod to convert the json result into a dataframe
response = requests.get(static_json_url).json()
df = pd.json_normalize(response)
```

Python

2) Cleaning data

```python
# Call getLaunchSite
getLaunchSite(data)
```

[38] ✓ 50.6s

```python
# Call getPayloadData
getPayloadData(data)
```

[39] ✓ 48.6s

```python
# Call getCoreData
getCoreData(data)
```

[40] ✓ 48.5s

CLICK HERE TO GO TO NOTEBOOK

Data Collection – SpaceX API

3) Converting list to dataframe

Finally lets construct our dataset using the data we have obtained. We we combine the columns into a dictionary.

```
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}
```

4) Filtering the dataframe and converting to csv file

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

✓ 0.1s

Data Collection – Web Scrapping

1) Get a respose from HTML adress

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

[12] ✓ 1.2s Python

```
200
```

2) To create BeautifulSoup Object

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

```python
# Use BeautifulSoup() to create a BeautifulSoup obje
soup = BeautifulSoup(page.text, 'html.parser')
```

[13] ✓ 1.3s Python

3) To find tables of interest

```python
# Use the find_all function in the BeautifulSoup obj
# Assign the result to a list called `html_tables`
html_tables = soup.find_all('table')
```

[15] ✓ 0.9s Python

4) To get the columns info

```python
column_names = []

# Apply find_all() function with `th` element on fir
# Iterate each th element and apply the provided ext
# Append the Non-empty column name (`if name is not
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
```

✓ 0.1s Python

Data Collection – Web Scrapping

5) Creating a dictionary

```python
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 b
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']=[]
```

✓ 0.7s Python

6) Appending data from all Keys, converting a dataframe and saving cs

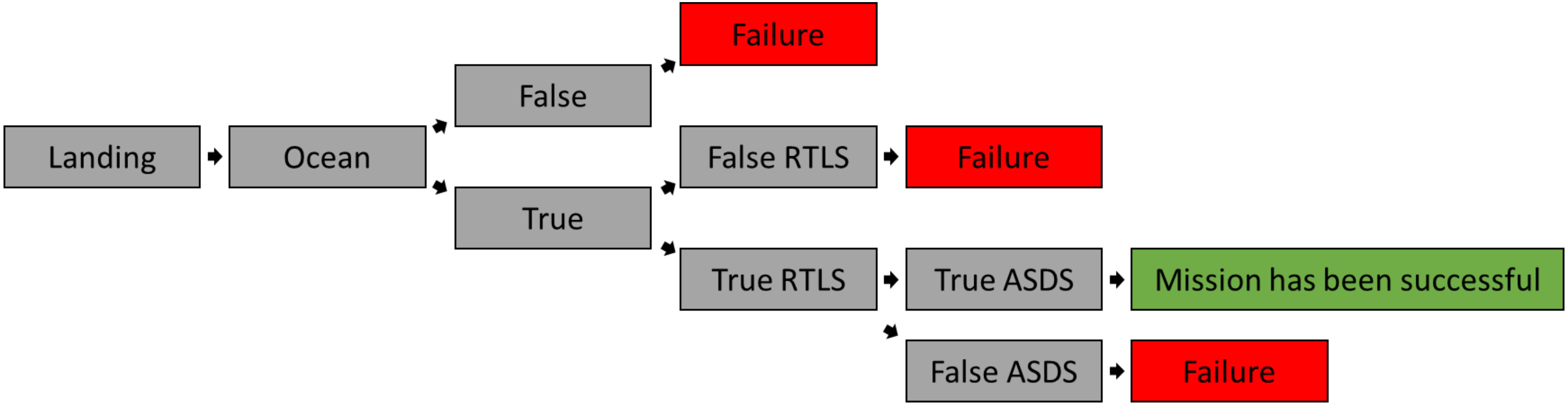
```python
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowhead
    # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch
        if rows.th:
          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

```python
headings = []
for key,values in dict(launch_dict).items():
    if key not in headings:
        headings.append(key)
    if values is None:
        del launch_dict[key]

def pad_dict_list(dict_list, padel):
    lmax = 0
    for lname in dict_list.keys():
        lmax = max(lmax, len(dict_list[lname]))
    for lname in dict_list.keys():
        ll = len(dict_list[lname])
        if  ll < lmax:
            dict_list[lname] += [padel] * (lmax - ll)
    return dict_list

pad_dict_list(launch_dict,0)

df = pd.DataFrame.from_dict(launch_dict)
df.head()
df.to_csv('spacex_web_scraped.csv', index=False)
```

✓ 0.1s

Data Wrangling

Regarding the data analysis process, there are several cases in which the booster did not land sucessfully.



Data Wrangling

1) Number of lauches

```
1  # Apply value_counts() on column LaunchSite
2  df["LaunchSite"].value_counts()
[34]  ✓ 0.1s
...  CCAFS SLC 40    55
     KSC LC 39A      22
     VAFB SLC 4E     13
     Name: LaunchSite, dtype: int64
```

2) Number accourence of Eath orbit

```
1  # Apply value_counts on Orbit column
2  df["Orbit"].value_counts("Orbit")
[35]  ✓ 0.1s
...  GTO      0.300000
     ISS      0.233333
     VLEO     0.155556
     PO       0.100000
     LEO      0.077778
     SSO      0.055556
     MEO      0.033333
     HEO      0.011111
     SO       0.011111
     ES-L1    0.011111
     GEO      0.011111
     Name: Orbit, dtype: float64
```

3) Number mission outcome

```
1  # landing_outcomes = values on Outcome column
2  landing_outcomes = df["Outcome"].value_counts()
3  landing_outcomes
[44]  ✓ 0.9s
...  True ASDS      41
     None None      19
     True RTLS      14
     False ASDS      6
     True Ocean      5
     None ASDS       2
     False Ocean     2
     False RTLS      1
     Name: Outcome, dtype: int64
```

4) Creating outcome label

```
1  for i,outcome in enumerate(landing_outcomes.keys()):
2      print(i,outcome)
[37]  ✓ 0.1s
...  0 True ASDS
     1 None None
     2 True RTLS
     3 False ASDS
     4 True Ocean
     5 None ASDS
     6 False Ocean
     7 False RTLS
```

[CLICK HERE TO GO TO NOTEBOOK]

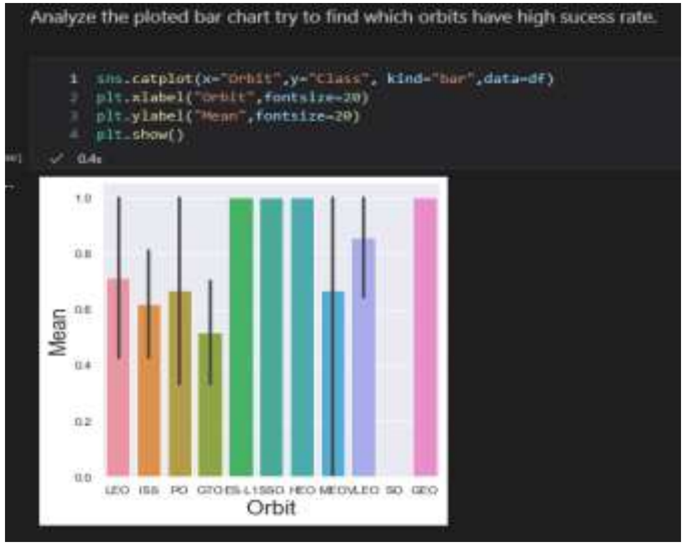
CLICK HERE TO GO TO NOTEBOOK

EDA with Data Visualization

Bar Plots

Mean vs orbit

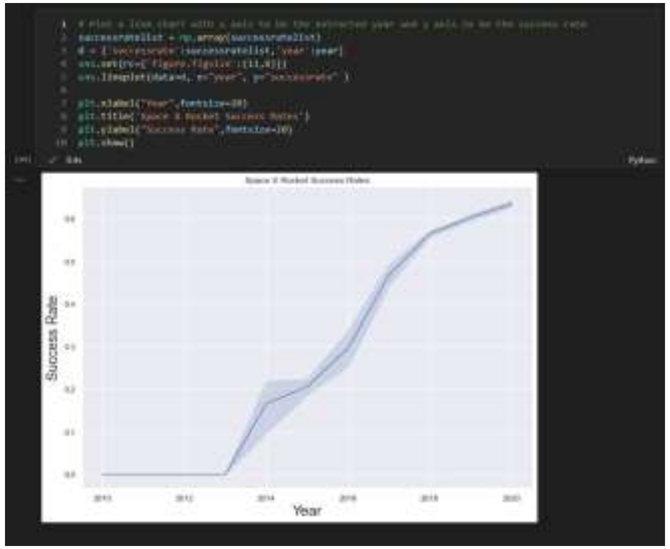
A bar diagram makes it easy to compare sets of data between diferente groups.



Line Plots

Success rate vs Year

Line graphs are usefull in that They show data variables and trends during the time.



Scatter Plots

Scatter plots showed the relationship between two variables (correlaton).

Flight Number vs Payload Mass
Flight Number vs Launch Site
Payload vs Launch Site
Orbit vs Flight Number
Payload vs Orbit Type
Orbit vs Payload Mass

EDA with SQL

SQL queries performed:

- Displaying the names of the unique sites in the space mission;
- Display 5 records Where launch sites begin with ‘CCA’;
- Display the total payload mass carried by boosters launched by NASA;
- Display the average of the amount paid by booster version F9;
- Lsit of the dates of the first sucessfull landing outcome;
- List of the names of the boosters which have sucess in drone ships and have payloads between 4000 and 6000;
- The total of numbers of sucessfull and failure mission outcomes;
- List of the names of the booster_version which have carried the maximum pauloadmass;
- Listing the records which will display the sucessfull landing_outcomes in the groumd pda, booster versions, and launch site for months inthe year 2015 and
- Ranking the Count of sucessfull landing_outcomes betwwen the date 2010-06-04 and 2017-03-20

CLICK HERE TO GO
TO NOTEBOOK

Build na Interactive Map with Follium

To visualize the Launch Data into na interactive map:

- We took the Lattitude and Longitude Coordinates at each launch site and added a Circle Marker Around each launch site with a label the name of the launch site;
- We assigned the dataframe launch outcomes and converted to classes 0 and 1 with Green and Red and
- The red circles at each launch site coordinates with label showing launch site name.

The objects are created to understand better the data problem. Also, we can easily show all launch sites and the sucessfull and unsucessfull landings.

CLICK HERE TO GO TO NOTEBOOK

Build a Dashboard with Plotly Dash

The dashboard has a dropdrow, pie chart and scatters plot:

- The dropdown allows you to choose the launch site;
- The pie chart shows the total of sucessfull and unsucessfull launches sites selected from the dropdowns.

Scatter Graph The relatioship with Outcome and Payload Mass for the diferente Booster Version:

- It shows the relatinship between two variables;7
- It’s best to show yoy a non-linear pattem;
- The range of data flow, maximum and minimum value can be determined and
- Observation and Reading are starightforward.

CLICK HERE TO GO TO NOTEBOOK

Predictive Analysis (Classification)

Data Preparation

- Load our dataset into NumPy and Pandas;
- Data transformation and
- Split our data into training and test data sets.

Model Evaluation

- Check the accuracy each model and
- Get tuned hyperoarameters for each type of algorithm.

Model Preparation

- Check how many test samples we have;
- Decide which type of machine learning algorithms to GridSeachCV and
- Training GridSeachCV.

Improving the model and finding the best classification model

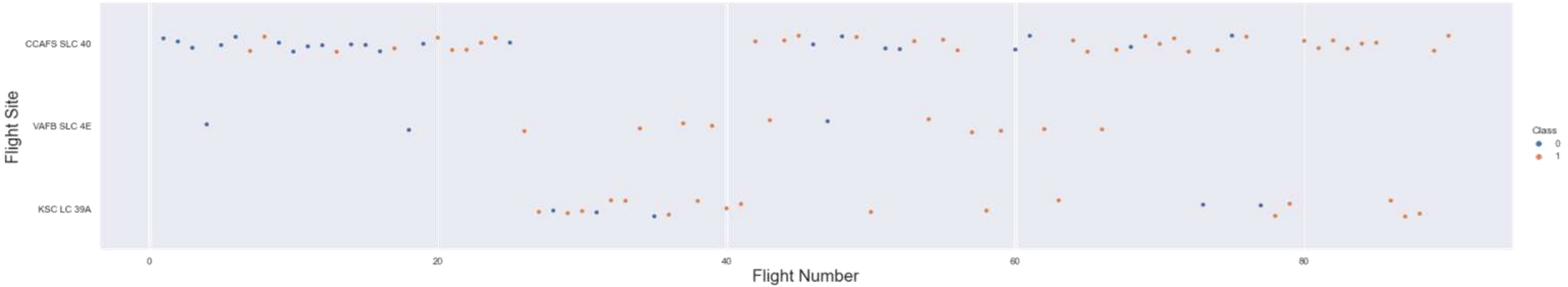
- Feature Engineering;
- Algorithm Tuning;
- Compare betwwen methods and the model with the best accuracy score wins the best performing model.

CLICK HERE TO GO TO NOTEBOOK

Results

- Exploratory Data Analysis Results
- Interactive Analytics Demo in Screenshots
- Predictive Analysis Results

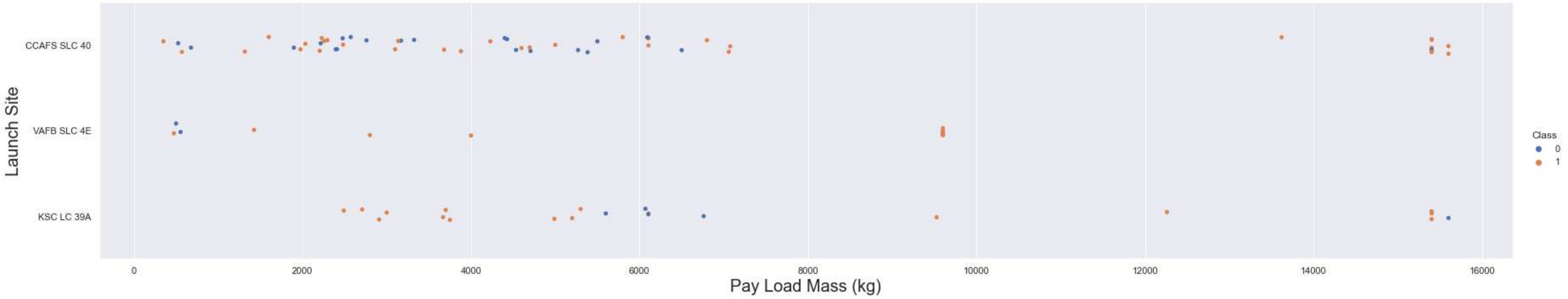
Flight Number vs Launch Site



It was observed, for each site, the sucess rate is increasing.

CLICK HERE TO GO
TO [NOTEBOOK]

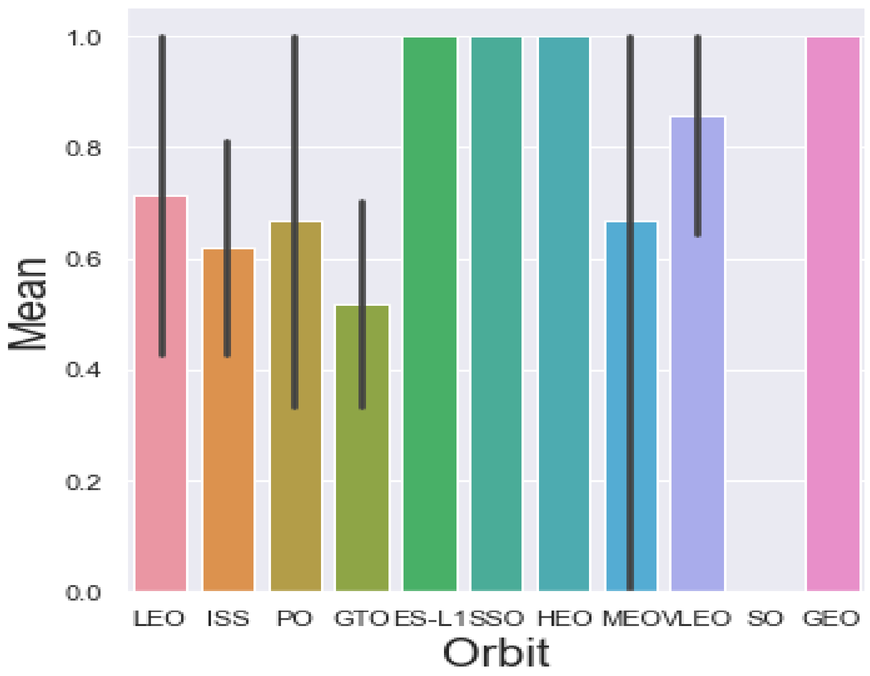
Payload vs Launch Site

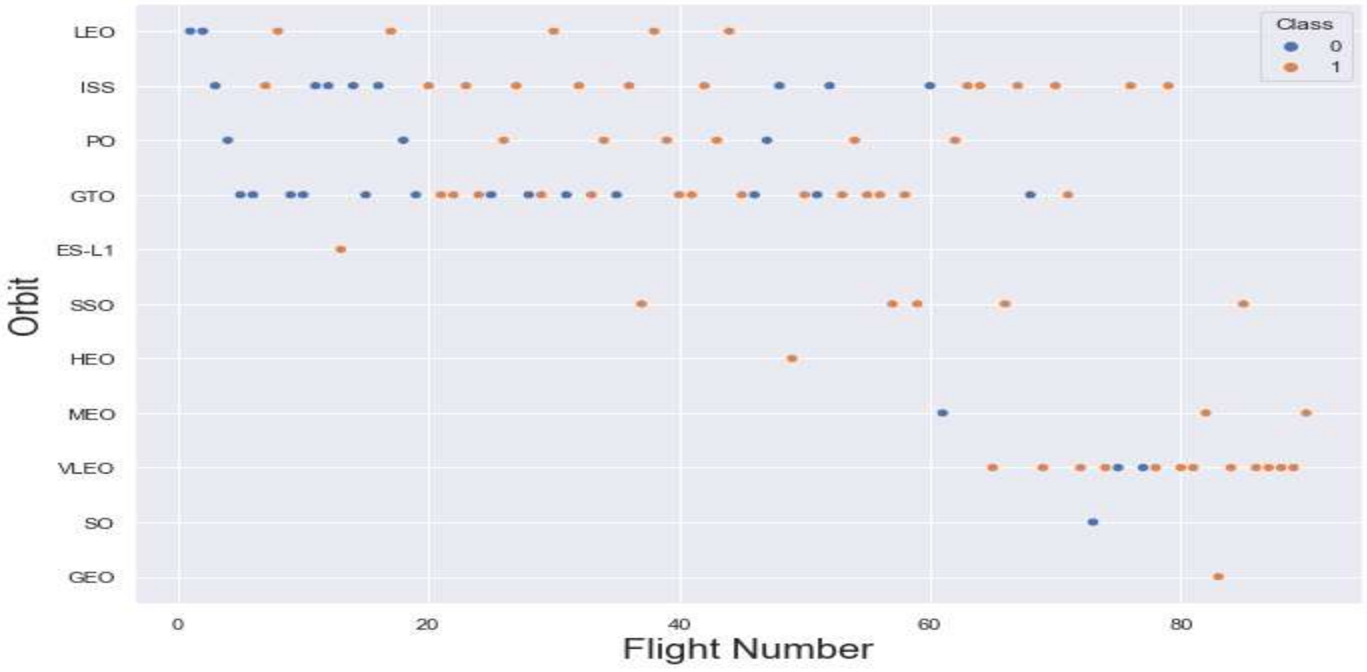


The heave impacts the launch site, therefore a heavier payload may be a consideration for a sucessfull landing.

CLICK HERE TO GO TO NOTEBOOK

Success Rate vs Orbit Type





We found that the sucess rate increases with the number of flights for the LEO orbit. On the Other hand, in Other orbits, like GTO, it's not related to sucess.

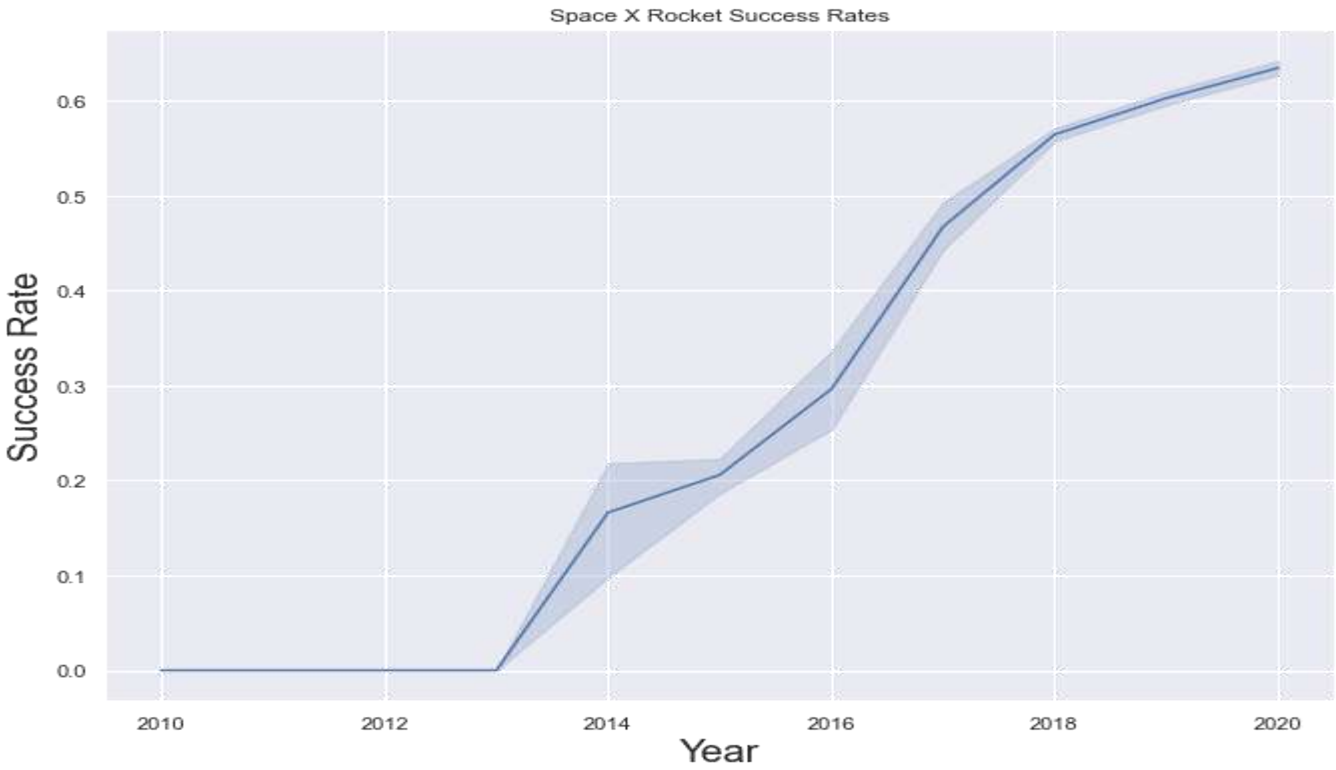
CLICK HERE TO GO TO NOTEBOOK

Payload vs Orbit Type

Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO orbits.

CLICK HERE TO GO
TO NOTEBOOK

Launch Sucess Yearly Trend



Since 2013, the sucess rate has beeen increasing.

CLICK HERE TO GO TO [NOTEBOOK]

All Launch Site Names

```
%sql select DISTINCT launch_site from SpaceX
```

Python

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

To use DISTINCT in a query remove all duplicated values .

CLICK HERE TO GO TO NOTEBOOK

Launch Site Names Begin with 'CCA'

```python
1  %sql select * from SpaceX WHERE launch_site LIKE 'CCA%' limit 5
```

Python

DATE	time_utc_	booster_version	launch_site	payload	payload_mass_kg_	orbit	customer
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
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
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)

To use WHERE follwed by LIKE allows get the launchs that contain subststing 'CCA', and getting Only the 5 rows using LIMIT 5

CLICK HERE TO GO TO NOTEBOOK

Total Payload Mass

```
%sql select SUM(payload_mass__kg_) from SpaceX where Customer = 'NASA (CRS)'
```

Python

1
45596

The query gives the sum of all payload Where customer is equal to NASA.

CLICK HERE TO GO TO NOTEBOOK

Average Payload Mass by F9 v1.1

```
%sql select AVG(payload_mass__kg_) from SpaceX where Booster_Version = 'F9 v1.1'
```

1
2928

The WHERE clause filters the dataset to Only perform calculations on Booster_version F9 v1.1.

CLICK HERE TO GO TO NOTEBOOK

First Sucessfull Ground Landing Date

```
%sql select MIN(DATE) from SpaceX where landing__outcome = 'Success (ground pad)'
```

Python

1
2015-12-22

The WHERE clause filters the dataset to Only perform calculations on Landing_Outcome = Sucess (ground pad) and get first day.

CLICK HERE TO GO
TO NOTEBOOK

Sucessfull Drone Ship Landing with Payload betwenn 4000 and 6000

```python
%sql select DISTINCT booster_version from SpaceX where landing__outcome = 'Success
```

Python

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

The query returns the booster version Where landing was sucessfull and payload was between 4k and 6k.

CLICK HERE TO GO TO NOTEBOOK

Total Number of Sucessfull and Failure Mission Outcomes

```
%sql SELECT mission_outcome, COUNT(*) FROM SpaceX GROUP BY mission_outcome
```

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Using the function COUNT works out the amount.

CLICK HERE TO GO TO NOTEBOOK

Boosters Carried Maximum Payload

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

Using the word SELECT in the query means that it will show values in the /booster_Version colum from Space.

CLICK HERE TO GO TO NOTEBOOK

2015 Launch Records

DATE	landing__outcome	booster_version	launch_site
2015-01-10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

DATE LIKE puts the value of 2015.

CLICK HERE TO GO
TO NOTEBOOK

Ranking Landing Outcomes Betwenn 2010-06-04 and 2017-03-20

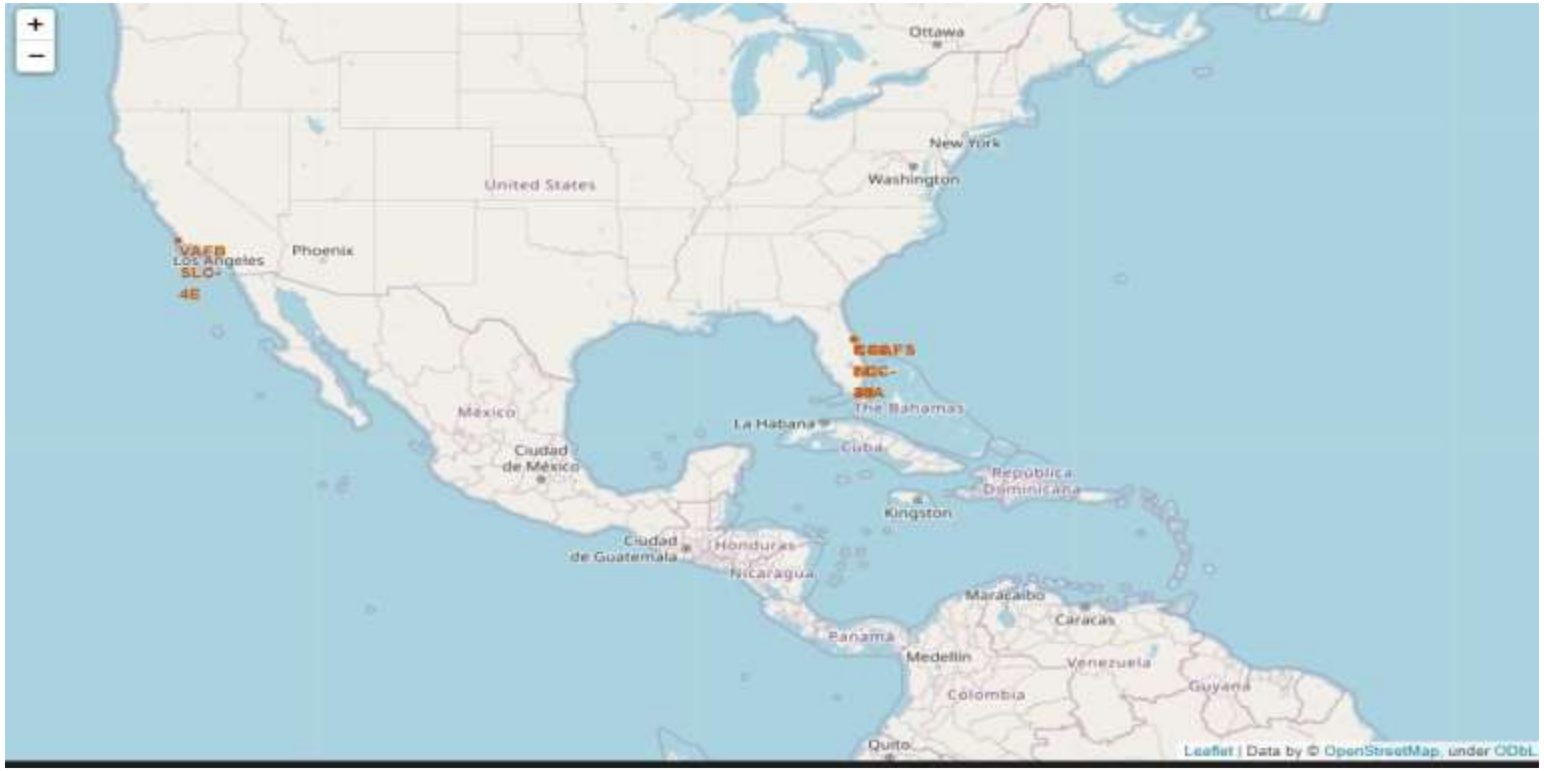
DATE	time_utc_	booster_version	launch_site	payload	payload_mass_kg_	orbit	customer	mission_outcome	landing_outcome
2016-06-15	14:29:00	F9 FT B1024	CCAFS LC-40	ABS-2A Eutelsat 117 West B	3600	GTO	ABS Eutelsat	Success	Failure (drone ship)
2016-03-04	23:35:00	F9 FT B1020	CCAFS LC-40	SES-9	5271	GTO	SES	Success	Failure (drone ship)
2016-01-17	18:42:00	F9 v1.1 B1017	VAFB SLC-4E	Jason-3	553	LEO	NASA (LSP) NOAA CNES	Success	Failure (drone ship)
2015-04-14	20:10:00	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
2015-01-10	09:47:00	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
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)
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)

Function WHERE filters landing_outcome and LIKE (Sucessor Failure; AND (DATE between) DESC means its arranging the dataset into descending order.

CLICK HERE TO GO
TO NOTEBOOK

Launch Sites Analysis

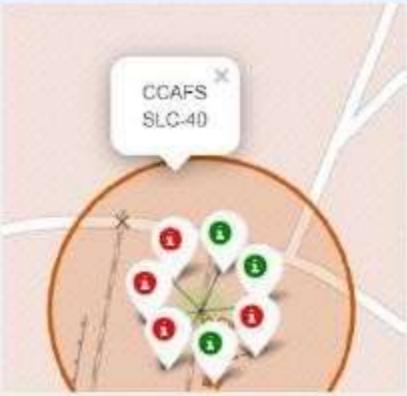
SpaceX in the US

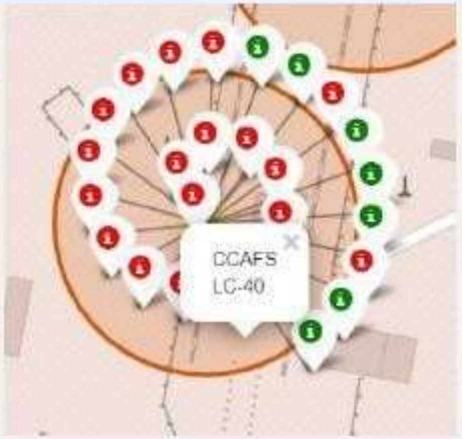


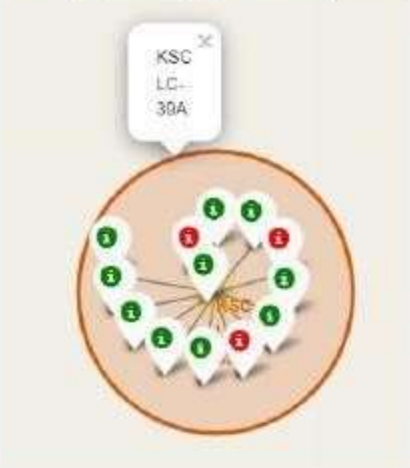
SpaceX in the US.

CLICK HERE TO GO TO NOTEBOOK

Color Label Launch Outcomes







Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

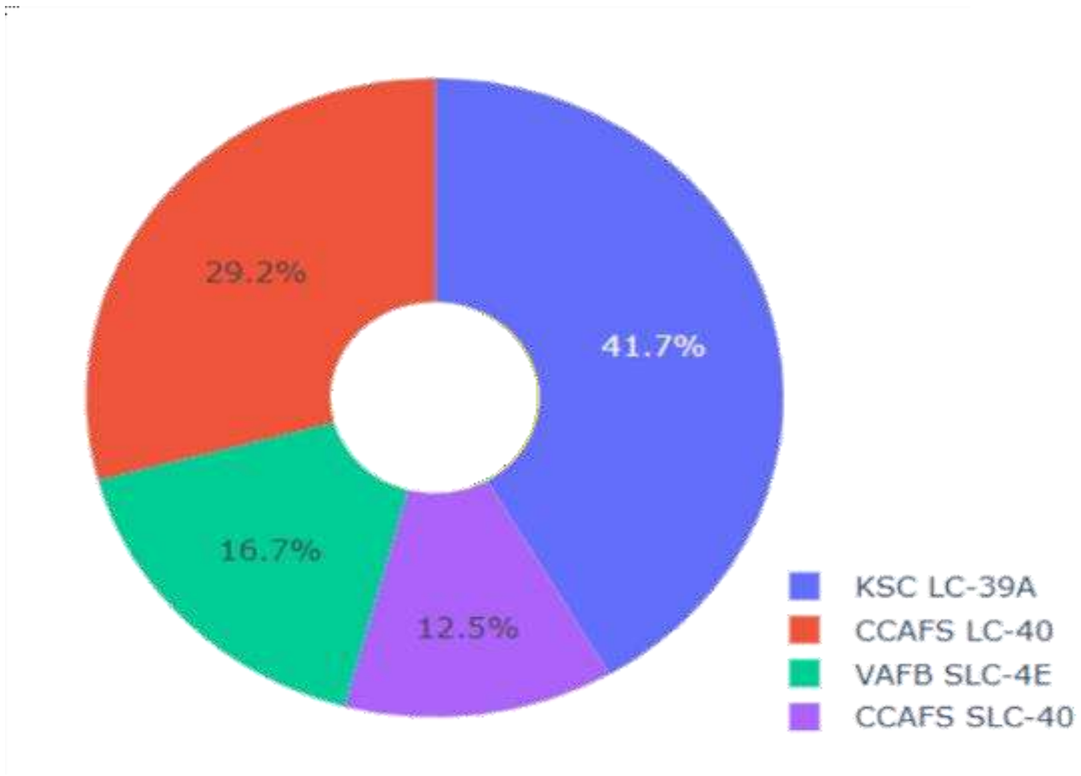
CLICK HERE TO GO TO NOTEBOOK

Color Label Launch

CLICK HERE TO GO
TO NOTEBOOK

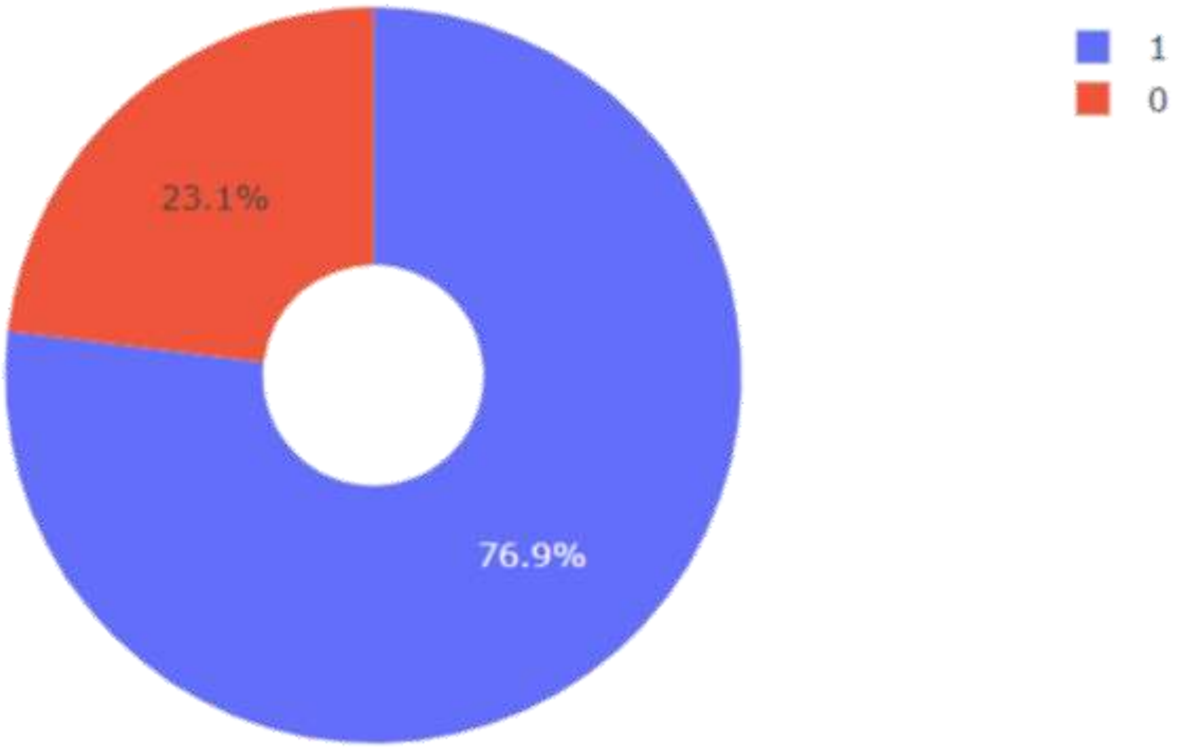
Build a Dashboard with Ploty Dash

Pie Chart Showing the Sucess Percentage



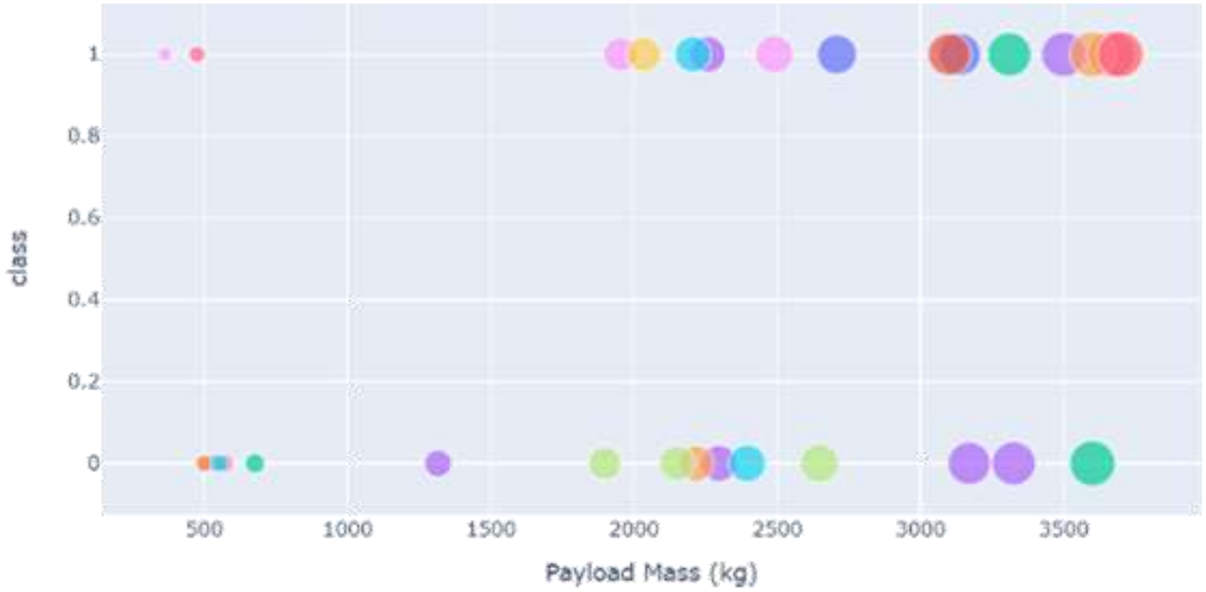
KSC had most sucessfull launches.

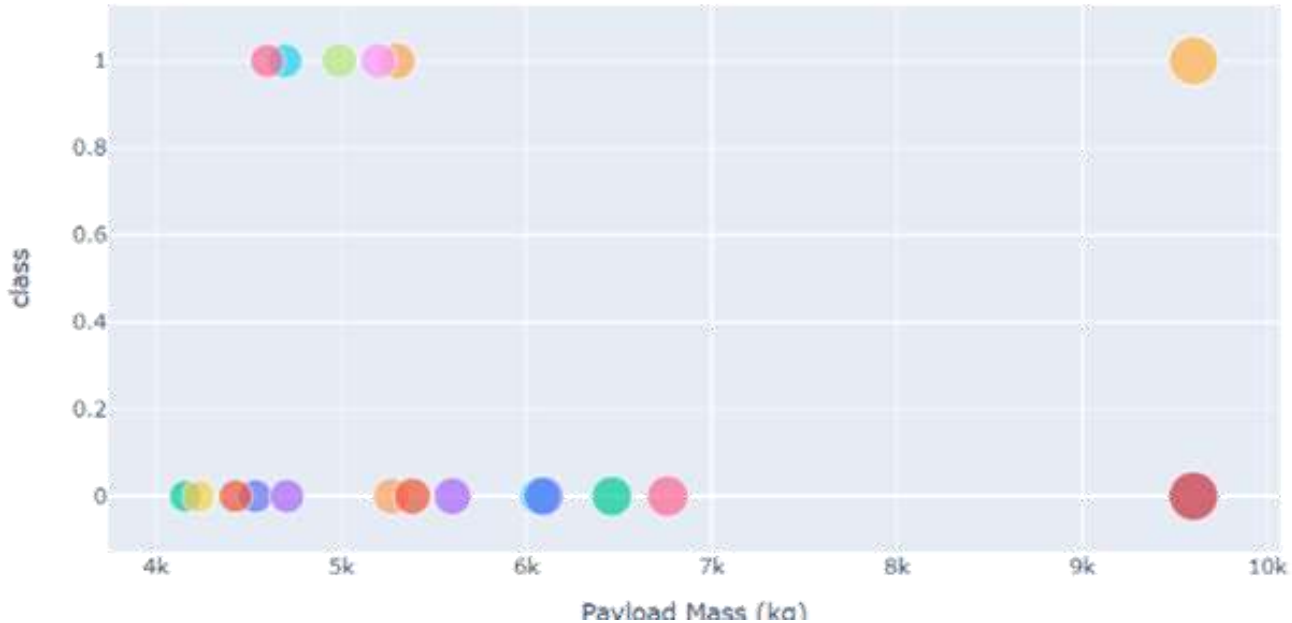
Pie Chart Showing the Sucess Ratio



KSC LS-39[a] achieved a 76,9% of sucess.

Scatter plot of Payload vs Launch Outcomes





The sucess rates for low weighted payload us highter than heavy payloads.

Predictive Analysis (Classification)

Classification Accuracy

```python
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
```

[31] ✓ 0.1s Python

```python
tree_cv=GridSearchCV(tree, param_grid=parameters, cv=10)
tree_cv.fit(X_train,Y_train)
```

[32] ✓ 6.2s Python

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

```python
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

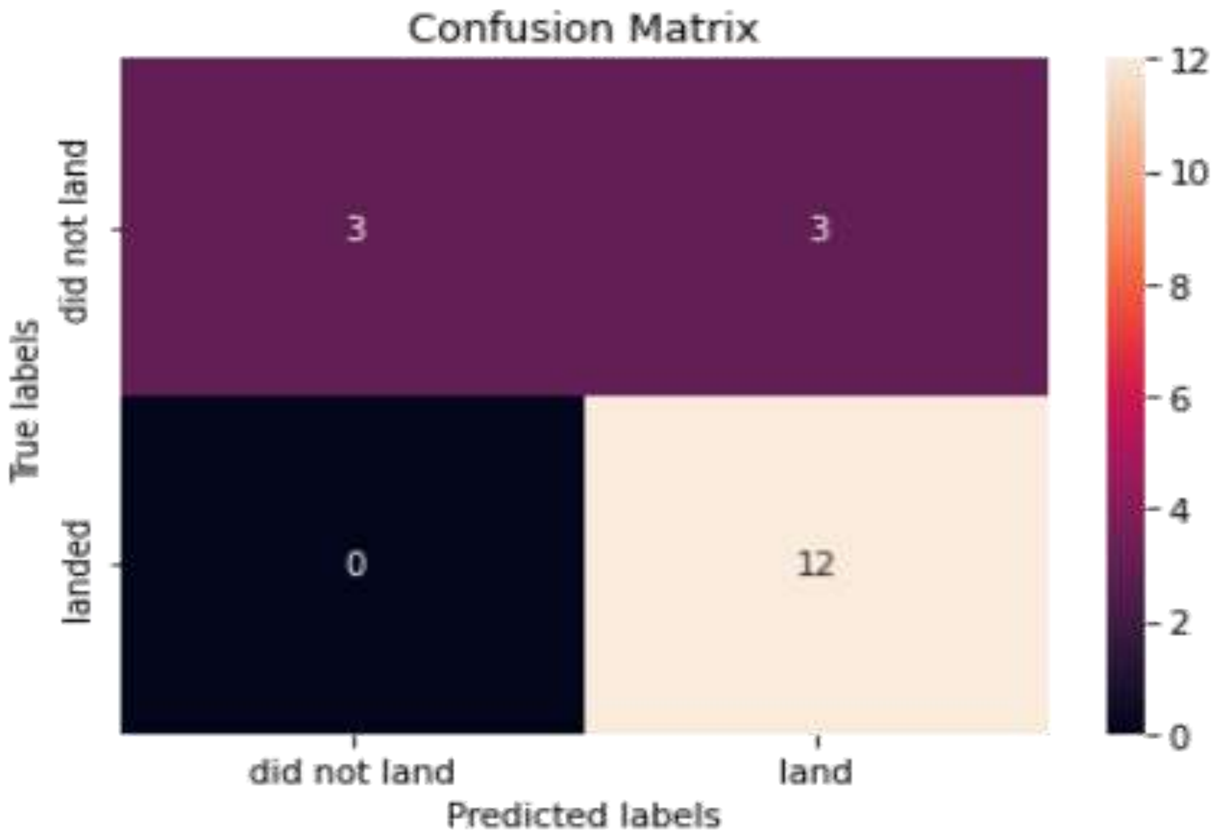
[33] ✓ 0.7s Python

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 1,
'min_samples_split': 2, 'splitter': 'random'}
accuracy : 0.8767857142857143
```

The decision tree was the best model based in the classification accuracy.

CLICK HERE TO GO
TO NOTEBOOK

Confusion Matrix



The confusion matrix for the decison tree classifier shows that the best can distinguish betwwen the classes.

CLICK HERE TO GO TO NOTEBOOK

Conclusion

- ✓ The success rates for SpaceX launches is directly proportional time in years;
- ✓ Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate;
- ✓ The Decision tree classifier is the best machine learning algorithm for this task and
- ✓ SpaceX's successful launches are directly linked to years of improvement

Thank you

IBM®

: