IBM Data Science Capstone Project

Nikita Patel

Github Repository

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

EXECUTIVE SUMMARY

Objective

Create machine learning model to predict if SpaceX will reuse the 1st Stage from launch

Data Collection and Wrangling

Gather SpaceX Launch Data with API

Transform and clean for meaningful dataset

Exploratory Data Analysis

Explore database

Identify features/attributes:
- Correlated with successful landings
- Determining if 1st stage can be reused

Interactive Visual Analytics and Dashboard

Stakeholder Dashboard and Interactive Map

Identify how to choose optimal launch site

Explore more insights

Predictive Analysis

Machine Learning pipeline to predict if 1st stage lands successfully

Outcome

Create machine learning model to predict if SpaceX will reuse the 1st Stage from launch

INTRODUCTION

BACKGROUND

SpaceX advertises lower cost rocket launches compared to other providers. A SpaceX Falcon9 launch is advertised to cost $62million compared to other providers who have launch costs upwards of $165million. The SpaceX price point per launch is lower due to the ability to reuse the first stage. If we can determine if the first stage will land, we can determine the cost of the launch. This would enable companies like SpaceY to compete with SpaceX.

GOAL

The goal of this project is to create a machine learning pipeline to predict if the first stage will land successfully

KEY QUESTIONS

- What factors determine if the rocket will land successfully?
- The interaction amongst various features that determine the success rate of a successful landing.
- Which machine learning classifier best predicts if the first stage of a launch will land successfully?

METHODOLOGY

Data Collection
SpaceX API
Web-scraping from Wikipedia

Data Wrangling
One-hot encoding of categorical features

Exploratory Data Analysis

SQL
Visualization

Interactive Visual Analytics and Dashboard

Folium Interactive Map
Plotly Dash Dashboard

Predictive Analysis (Classification)

Preprocess
Standardize
Gridsearch for optimal hyperparameters and accuracy
Evaluate classification models

METHODOLOGY

Data Collection

Request data
- Perform get request to SpaceX API

Convert JSON to Dataframe
- API returns in JSON form
- View results using .json() method
- Convert results to pandas dataframe using json.normalize()

Filter and Clean Data
- Filter to show Falcon 9 launch data only
- Check for missing values
- Fill missing values where necessary

Webscraping from Wikipedia
- Use BeautifulSoup to web scrape Falcon 9 launch records
- Extract relevant column names from HTML table header
- Parse HTML table and convert to pandas dataframe

METHODOLOGY

Data Collection

SpaceX API

Use get request to SpaceX API to collect data

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

```
response = requests.get(spacex_url)
```

Clean the requested data

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

Basic data wrangling and formatting

```
# Calculate the mean value of PayloadMass column
avg_PayloadMass = data_falcon9['PayloadMass'].astype('float').mean(axis=0)
avg_PayloadMass

# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan,avg_PayloadMass,inplace=True)
data_falcon9.head()

#data_falcon9.isnull().sum()

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

Link to Notebook:
Data Collection API Notebook

METHODOLOGY

Data Collection

Web-scraping from Wikipedia

Link to Notebook:
[Data Collection with Web Scraping Notebook]

Use HTTP GET method to request the Falcong Launch HTML page
Use BeautifulSoup method

```python
# use requests.get() method with the provided static_url
# assign the response to a object
response = 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(response,'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```python
# Use soup.title attribute
print("Title of the website is : ")
for title in soup.find_all('title'):
    print(title.get_text())
```

```
Title of the website is :
List of Falcon 9 and Falcon Heavy launches - Wikipedia
```

Extract column name from HTML table header

```python
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names

col_finder = first_launch_table.find_all('th')

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

Table was parsed and converted into a pandas dataframe

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

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	NASA (COTS)\nNRO	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	NASA (COTS)	Success	F9 v1.0B0005.1	No attempt\n	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA (CRS)	Success\n	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA (CRS)	Success\n	F9 v1.0B0007.1	No attempt\n	1 March 2013	15:10

METHODOLOGY

Data Wrangling

One-hot encoding of categorical features

Link to Notebook:
[Data Wrangling Notebook]()

Perform Exploratory Data Analysis and Determine Training labels

Number of launches on each site

```
# 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
```

Number and occurrence of each orbit

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

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

Number and occurrence of mission outcome per orbit type

```
# 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
```

Landing outcome label from outcome column:
0 = 1st stage did not land successfully
1 = 1st stage successful landing

METHODOLOGY

Exploratory Data Analysis

SQL

Load SQL extension and establish a connection with the database to perform queries in Jupyter notebook

Explore database with SQL to get more insight. Queries written include:

- Names of the unique launch sites in the space mission
- Date when first successful landing outcome in ground pad acheived
- Total number of successful and failure mission outcomes
- Names of the booster_versions which have carried the maximum payload mass with subquery
- Rank the count of landing outcomes between the date 2010-06-04 and 2017-03-20

Link to Notebook:
[EDA with SQL Notebook]

METHODOLOGY

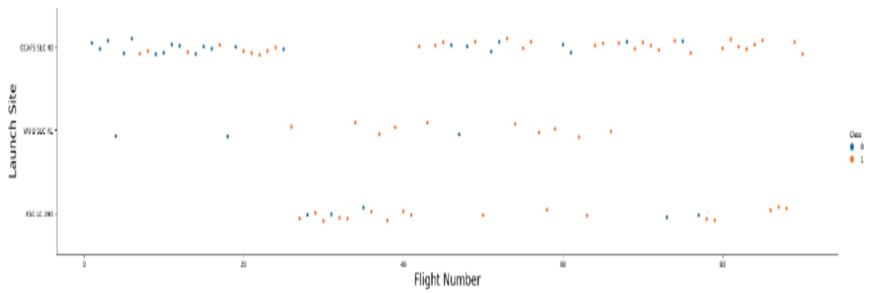
Exploratory Data Analysis

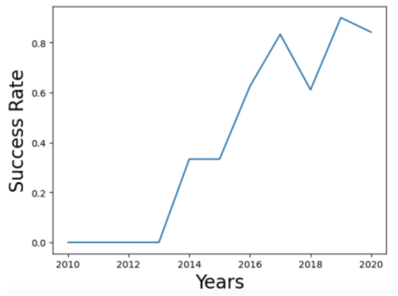
Visualization

Link to Notebook:
[EDA with Data Visualization Notebook]

Perform Exploratory Data Analysis and Feature Engineering using Pandas and Matplotlib

Explore relationships between data features and their effect on launch outcomes with different visual plots
- Flight no. Vs launch site Scatter
- Payload vs launch site Scatter
- Success rate of each orbit type bar
- Flight number vs orbit type scatter
- Payload vs orbit type scatter
-Launch success trend line





Feature engineering and One hot encoding of categorical data with get_dummies() for categorical columns

METHODOLOGY

Interactive Visual Analytics and Dashboard

Folium Interactive Map

Mark all launch sites on map and added objects such as markers, circles, lines to mark launch success/failures at each site

Assign launch feature outcomes to class 0 to 1

- 0 = failure
- 1=success

Use clusters to identify which launch sites have a high success rate

Calculated distance between launch site to proximities to understand if:

- Are launch sites in close proximity to railways?
- Are launch sites in close proximity to highways?
- Are launch sites in close proximity to coastline?
- Do launch sites keep certain distance away from cities?

Link to Notebook:
Interactive Visuals with Folium Notebook
NB Viewer to show maps

METHODOLOGY

Interactive Visual Analytics and Dashboard

Plotly Dash Dashboard

Link to Notebook:
[Interactive Dashboard with Plotly Dash Notebook]

Built Plotly interactive dashboard

Setup drop down menu for launch sites

Plot pie chart for total launches filtered by sites drop down menu

Setup Payload Mass (kg) slider

Scatter for relationship with outcome vs payload for different booster versions

Interactive/customizable results by launch site drop down and payload mass (kg) slider

METHODOLOGY

Predictive Analysis (Classification)

Link to Notebook:
[Machine Learning Prediction Notebook]
[NB Viewer to show Notebook]

Load data using numpy and pandas

Standardize and Transform data

Split into training and test sets

Build different machine learning classifiers and tune different hyperparameters using GridsearchCV

Use accuracy as metric for model, improve model using feature engineering and algorithm tuning

Find best performing classification model using accuracy and a confusion matrix

RESULTS

Exploratory Data Analysis (EDA) Insights

SQL
Visualization

Launch Sites Proximities Analysis

Folium Interactive Map

Interactive Analytics Dashboard

Plotly Dash Dashboard

Predictive Analysis (Classification)

GridSearchCV Optimal Machine Learning Model and Hyperarameters

RESULTS

Exploratory Data Analysis (EDA) Insights

SQL
Visualization

Launch Sites Proximities Analysis

Folium Interactive Map

Interactive Analytics Dashboard

Plotly Dash Dashboard

Predictive Analysis (Classification)

GridSearchCV Optimal Machine Learning Model and Hyperarameters

RESULTS

Exploratory Data Analysis (EDA) Insights

SQL

Link to Notebook: [EDA with SQL Notebook]()

All Launch Sites

Use DISTINCT to show unique launch sites from SpaceX data

```
%sql select distinct(LAUNCH_SITE) from SPACEXTBL;
```

```
 * ibm_db_sa://pqd13006:***@54a2f15b-5c0f-46df-8954-
Done.
```

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

RESULTS

Exploratory Data Analysis (EDA) Insights

SQL

Launch Site Names Begin with 'CCA'

Use WHERE to filter for Launch Sites, LIKE to specify launch sites with names beginning with 'CCA' and LIMIT to return 5 records

```
%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5;
```

```
 * ibm_db_sa://pqd13006:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

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

Link to Notebook:
[EDA with SQL Notebook]

RESULTS

Exploratory Data Analysis (EDA) Insights

SQL

Total Payload Mass

Use SUM to apply to Payload Mass data and WHERE to filter for NASA (CRS) customer launches only

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)';

 * ibm_db_sa://pqd13006:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.clogj3sd0tgtu0lqd
Done.
```

1
45596

Total Payload Mass (kg) = 45596kg for boosters launched NASA (CRS)

Link to Notebook:
[EDA with SQL Notebook]

RESULTS

Exploratory Data Analysis (EDA) Insights

SQL

Average Payload Mass for Booster F9 v1.1

Use AVERAGE to apply to Payload Mass data and WHERE to filter for Booster version F9 v1.1

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1';
```

```
 * ibm_db_sa://pqd13006:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.clogj3sd0tgtu0lqde00
Done.
```

1
2928

Average Payload Mass (kg) = 2928kg for F9 v1.1 boosters

Link to Notebook:
[EDA with SQL Notebook]

RESULTS

Exploratory Data Analysis (EDA) Insights

SQL

First Successful landing in ground pad

Use MIN to find earilest date and WHERE to filter for landing outcoming success in ground pad

```
%sql select min(DATE) from SPACEXTBL where LANDING__OUTCOME = 'Success (ground pad)';
```

```
 * ibm_db_sa://pqd13006:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.clogj3sd0tgtu0lqde00.
Done.
```

1
2015-12-22

First Date of Successful ground pad landing = 2015-12-22

Link to Notebook:
[EDA with SQL Notebook]

RESULTS

Exploratory Data Analysis (EDA) Insights

SQL

Boosters with success in drone ship landing with payload mass between 4000 and 6000kg

Select Booster Version data, use WHERE to filter for Payload Mass range between 4000-6000 with AND to determine successful landing in drone ship for Payload Mass between 4000-600

```
%sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ between 4000 and 6000 and LANDING__OUTCOME = 'Success (drone
```

```
 * ibm_db_sa://pqd13006:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.clogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

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

Link to Notebook:
[EDA with SQL Notebook]()

RESULTS

Exploratory Data Analysis (EDA) Insights

SQL

Boosters Carried Maximum Payload

Select Booster version data, use WHERE to filter for payload mass and a subquery to select the MAX payload

```
%sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

```
 * ibm_db_sa://pqd13006:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.clogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/I
Done.
```

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

Link to Notebook:
[EDA with SQL Notebook]

RESULTS

Exploratory Data Analysis (EDA) Insights

SQL

2015 Launch Records for Failed landing in drone ship, booster versions and launch site names

Select Landing outcome, booster version and launch site data using WHERE,LIKE,AND and BETWEEN to filter failed landing outcomes, their booster versions and launch site names for the year 2015

```
%sql select LANDING__OUTCOME,BOOSTER_VERSION,LAUNCH_SITE from SPACEXTBL where DATE like '2015%' and LANDING__OUTCOME = 'Failure
```

```
 * ibm_db_sa://pqd13006:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.clogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

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

Link to Notebook:
EDA with SQL Notebook

RESULTS

Exploratory Data Analysis (EDA) Insights

SQL

Rank count of landing outcomes between 2010-06-04 and 2017-03-20

Select COUNT of Landing outcome data with WHERE and BETWEEN to filter between dates 2010-06-04 and 2017-03-20.
Apply GROUP BY and ORDER BY to group data by landing outcome and to sort the output values in descending order

```
%sql select LANDING__OUTCOME,count(LANDING__OUTCOME) from SPACEXTBL where DATE between '2010-06-04' and '2017-03-20' GROUP BY L
```

```
 * ibm_db_sa://pqd13006:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

landing__outcome	2
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

Link to Notebook:
[EDA with SQL Notebook]

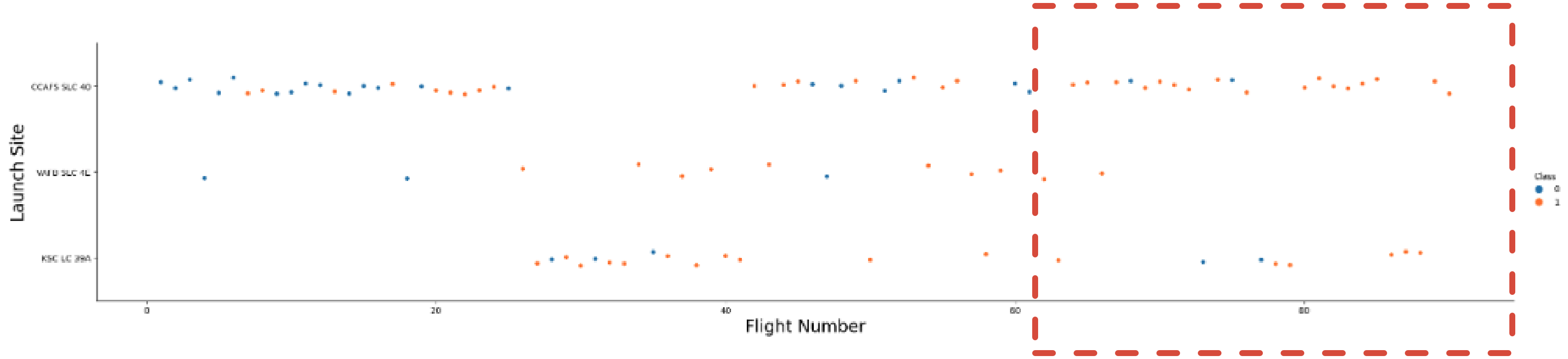
RESULTS

Exploratory Data Analysis (EDA) Insights

Visualization

Flight Number vs Launch Site

As the flight number increases at a launch site, the higher the success rate at the launch site



Link to Notebook:
[EDA with Data Visualization Notebook]()

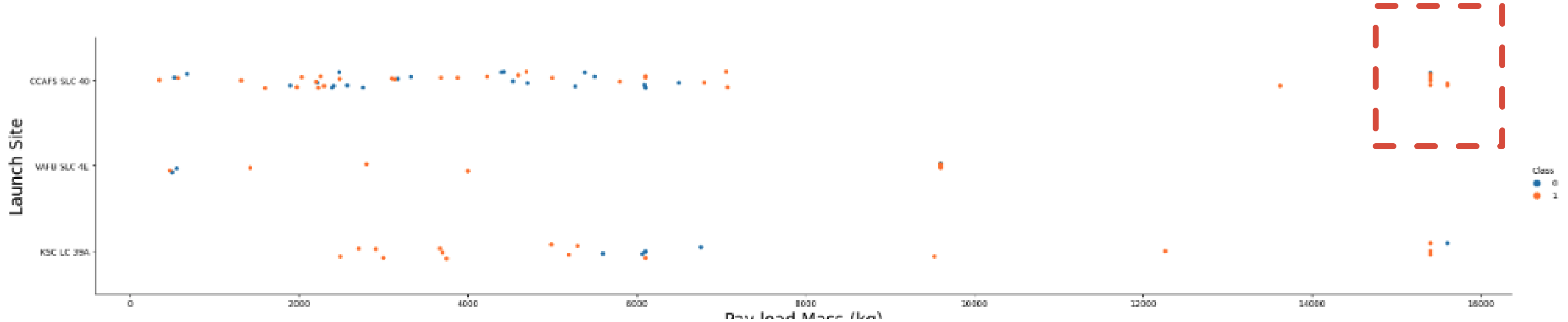
RESULTS

Exploratory Data Analysis (EDA) Insights

Visualization

Payload vs Launch Site

The higher the payload mass for the launch site CCAFS-SLC 40 the higher the success rate



Link to Notebook:
[EDA with Data Visualization Notebook]

RESULTS

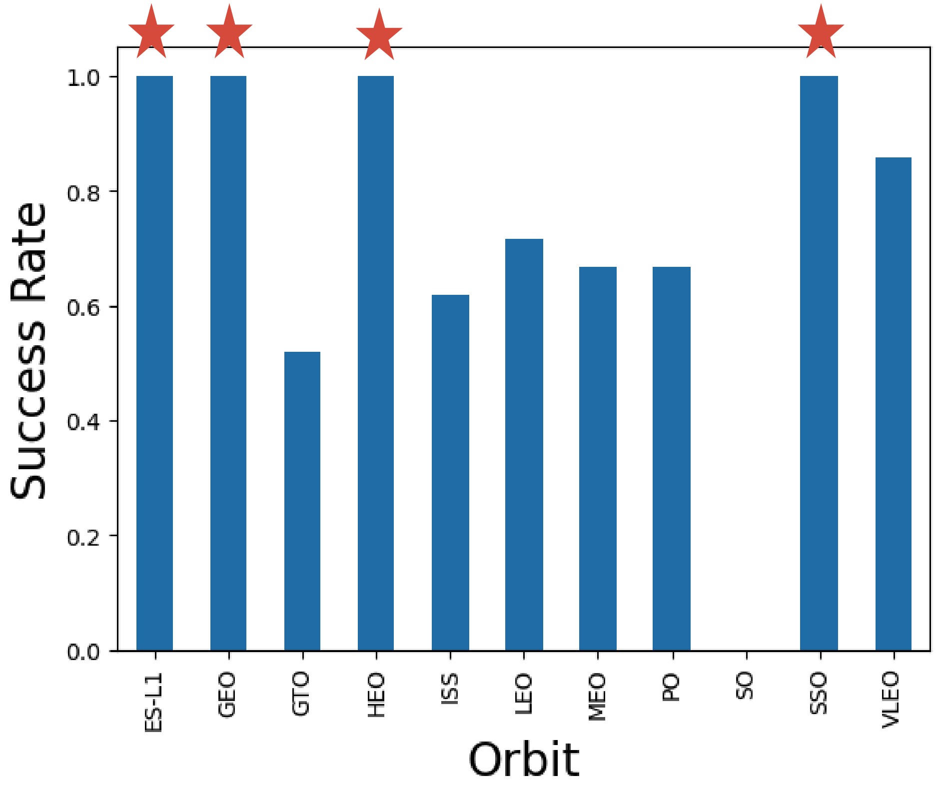
Exploratory Data Analysis (EDA) Insights

Visualization

Link to Notebook:
[EDA with Data Visualization Notebook]

Success Rate vs Orbit Type

ES-L1, GEO, HEO and SSO have the highest success rate



RESULTS

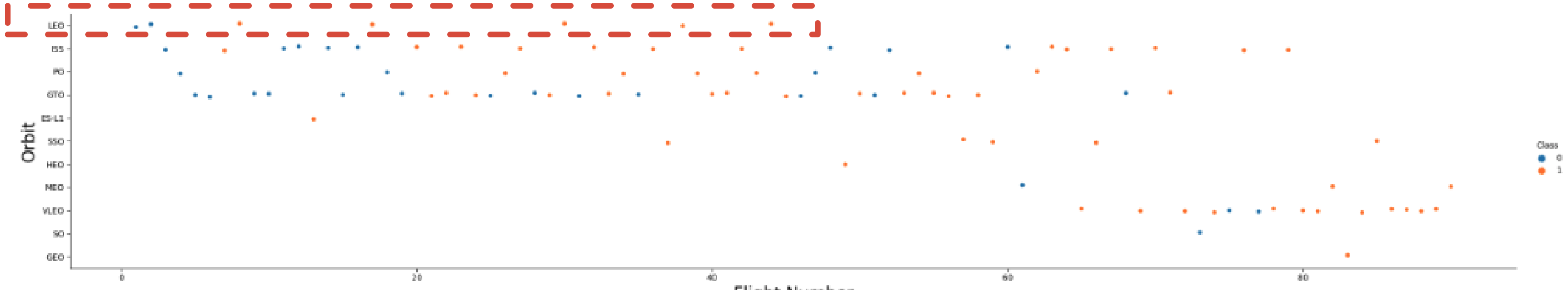
Exploratory Data Analysis (EDA) Insights

Visualization

Flight Number vs Orbit Type

LEO Orbit success shows relationship with number of flights

GTO orbit success shows no relationship with flight number



Link to Notebook:
[EDA with Data Visualization Notebook]()

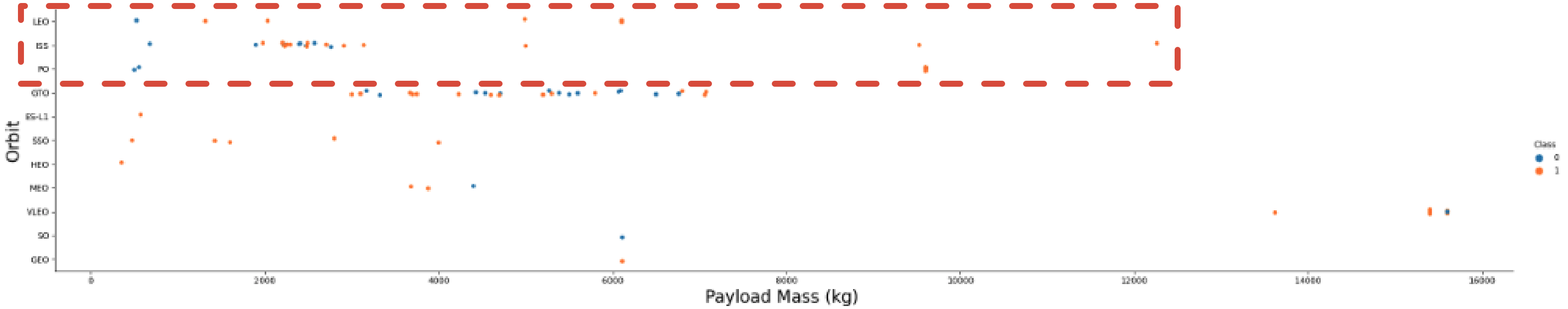
RESULTS

Exploratory Data Analysis (EDA) Insights

Visualization

Payload vs Orbit Type

For Polar, LEO and ISS Orbits, heavier payloads have more successful landings



Link to Notebook:
[EDA with Data Visualization Notebook]()

RESULTS

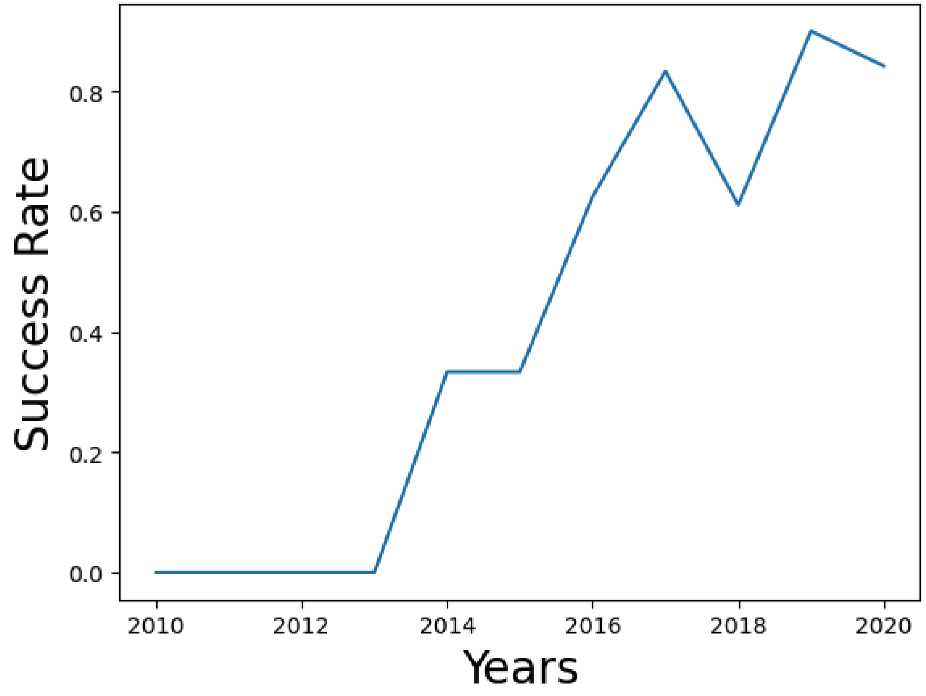
Exploratory Data Analysis (EDA) Insights

Visualization

Link to Notebook:
[EDA with Data Visualization Notebook]

Launch Success Yearly Trend

Success rates increased from 2013 to 2020



RESULTS

Exploratory Data Analysis (EDA) Insights

Visualization

Link to Notebook:
[EDA with Data Visualization Notebook]

Features Engineering

Features identified to be used in success prediction in future model

- Flight Number
- Payload Mass (kg)
- Orbit
- Launch Site
- Flights
- Grid Fins
- Reused
- Legs
- Landing Pad
- Block
- Reused Count
- Serial

RESULTS

Exploratory Data Analysis (EDA) Insights

SQL
Visualization

Launch Sites Proximities Analysis

Folium Interactive Map

Interactive Analytics Dashboard

Plotly Dash Dashboard

Predictive Analysis (Classification)

GridSearchCV Optimal Machine Learning Model and Hyperparameters

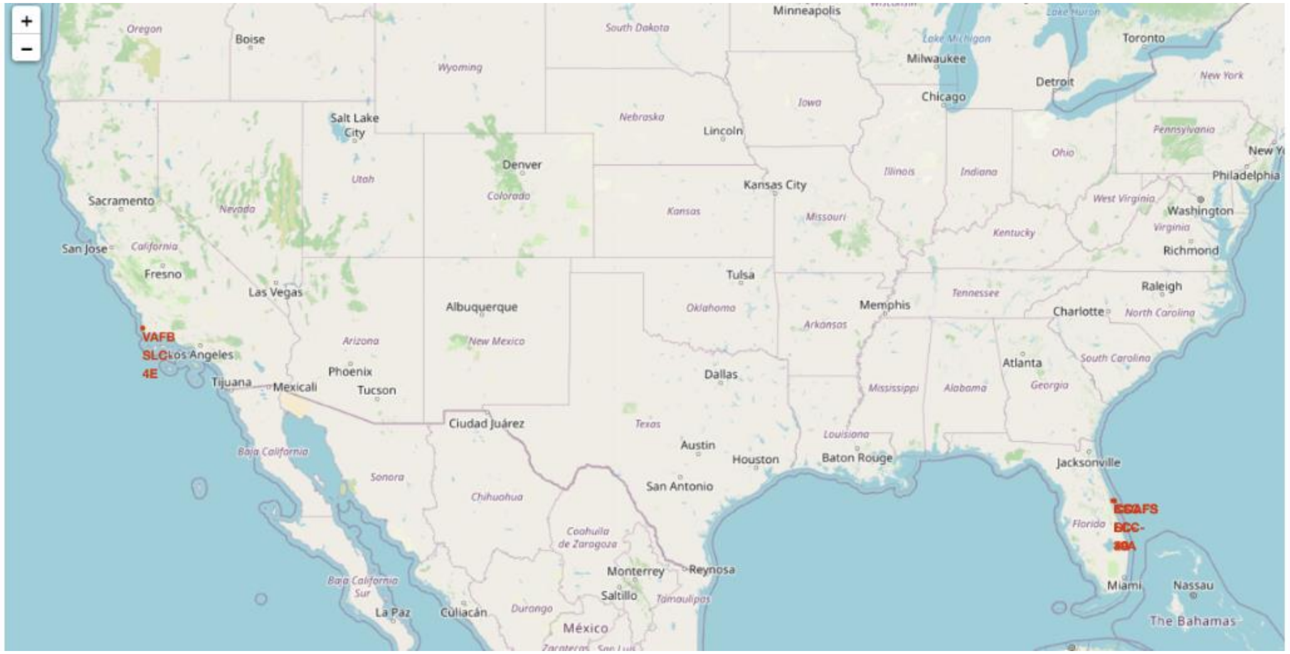
RESULTS

Launch Sites Proximities Analysis

Folium Interactive Map

All Launch Sites on Map

All launch sites for SpaceX are located in the USA
Launch sites are in California and Florida
Launch sites are fairly close to the equator
All launch sites are in close proximity to the coast



Link to Notebook:
[Interactive Visuals with Folium Notebook]
[NB Viewer to show maps]

RESULTS

Launch Sites Proximities Analysis

Folium Interactive Map

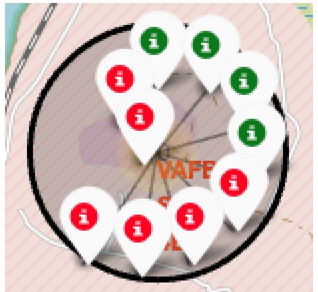
Link to Notebook:
[Interactive Visuals with Folium Notebook]
[NB Viewer to show maps]

All Launch Sites on Map

Launches are clustered by site

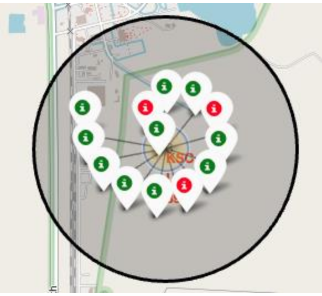
California

VAFB SLC-4E

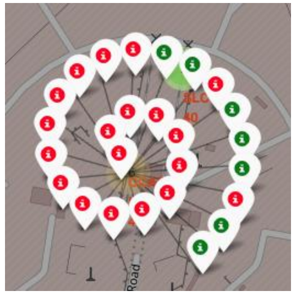


Florida

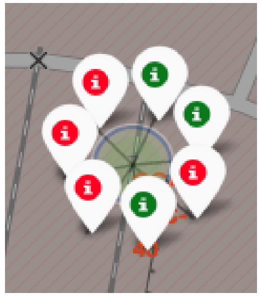
KSC LC-39A



CCAFS LC-40



CCAFS SLC-40



Green: Successful Launches
Red: Unsuccessful Launches

RESULTS

Launch Sites Proximities Analysis

Folium Interactive Map

Link to Notebook:
[Interactive Visuals with Folium Notebook]
[NB Viewer to show maps]

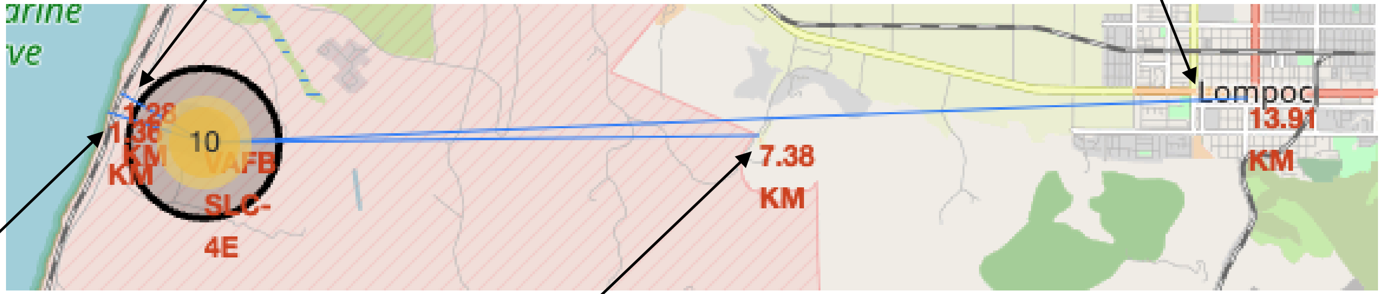
Launch Sites Distance to Landmarks

California

VAFB SLC-4E

1.28km to railway

13.91km to city



1.36km to coastline

7.38km to highway

Are launch sites in close proximity to railways? Yes
Are launch sites in close proximity to highways? No
Are launch sites in close proximity to coastline? Yes
Do launch sites keep a certain distance from cities? Yes

RESULTS

Exploratory Data Analysis (EDA) Insights

SQL
Visualization

Launch Sites Proximities Analysis

Folium Interactive Map

Interactive Analytics Dashboard

Plotly Dash Dashboard

Predictive Analysis (Classification)

GridSearchCV Optimal Machine Learning Model and Hyperparameters

RESULTS

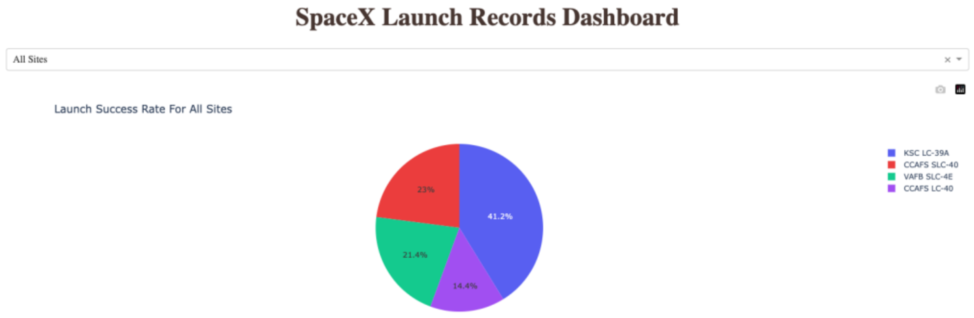
Interactive Dashboard Analytics

Plotly Dash Dashboard

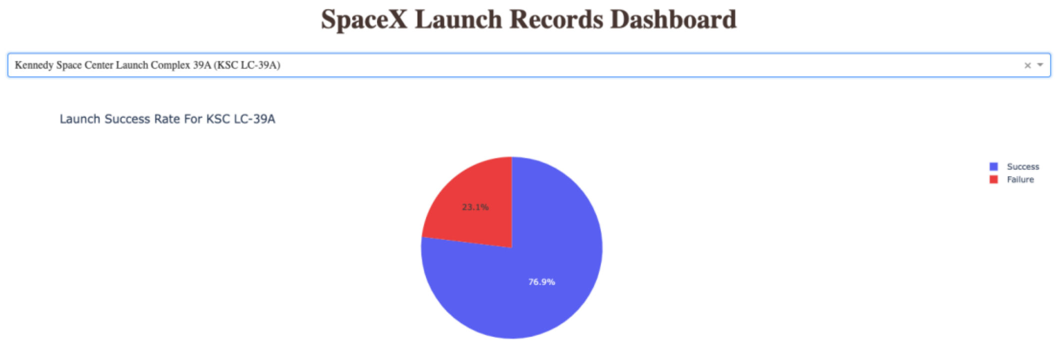
Link to Notebook:
[Interactive Dashboard with Plotly Dash Notebook]

Launch Sites Successes

KSC LC-39A has the largest successful launches at 41.2%



KSC LC-39A has the highest launch success rate at 76.9% success



RESULTS

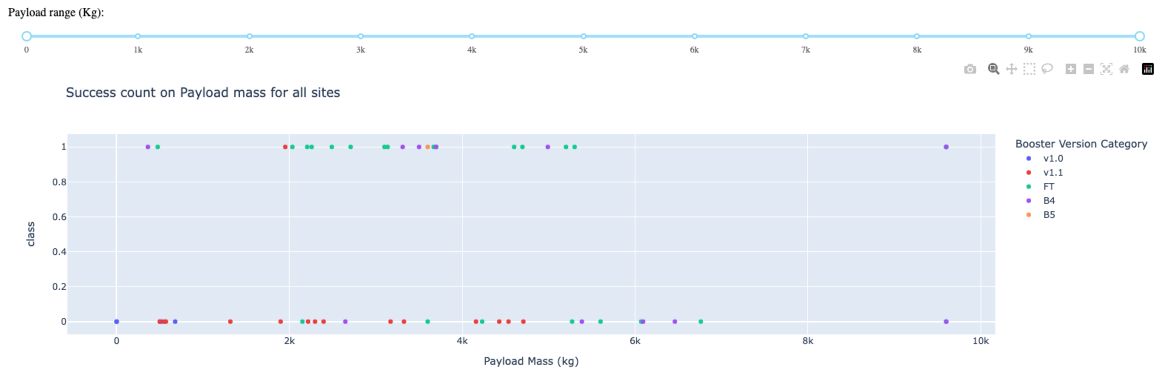
Interactive Dashboard Analytics

Plotly Dash Dashboard

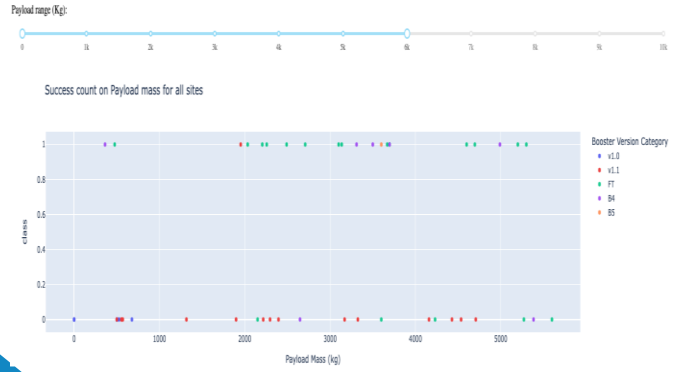
Link to Notebook:
[Interactive Dashboard with Plotly Dash Notebook]

Launch Sites Distance to Landmarks

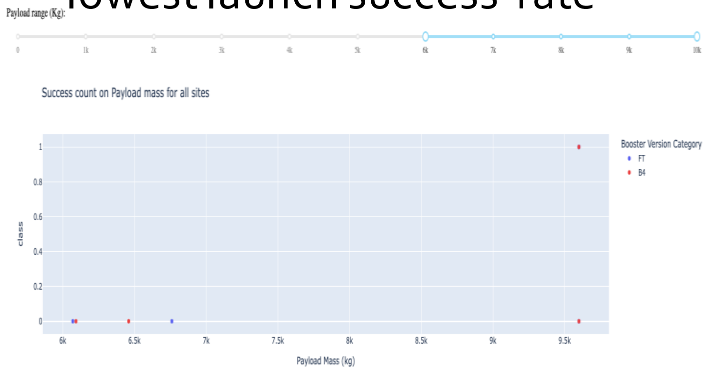
Success Count of Payload mass for all sites



Payload Range 0-6000kg has the highest launch success rate



Payload Range 6000-10000kg has the lowest launch success rate



RESULTS

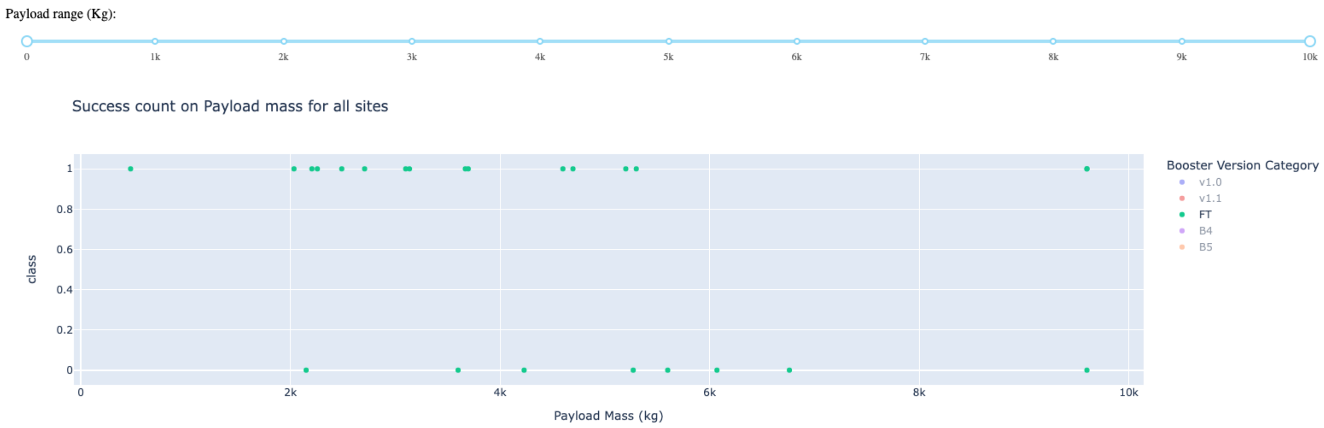
Interactive Dashboard Analytics

Plotly Dash Dashboard

Link to Notebook:
[Interactive Dashboard with Plotly Dash Notebook]()

Launch Sites Distance to Landmarks

F9 Booster version FT has the highest launch success rate



RESULTS

Exploratory Data Analysis (EDA) Insights

SQL
Visualization

Launch Sites Proximities Analysis

Folium Interactive Map

Interactive Analytics Dashboard

Plotly Dash Dashboard

Predictive Analysis (Classification)

GridSearchCV Optimal Machine Learning Model and Hyperarameters

RESULTS

Predictive Analysis (Classification)

GridSearchCV Optimal Machine Learning Model and Hyper-parameters

Link to Notebook:
[Machine Learning Prediction Notebook]
[NB Viewer to show Notebook]

Best Performing Method

The decision tree classifier model has the highest classification accuracy

```
#Scores for each test method
print('Logistic RegressionAccuracy on test data: ',logreg_cv.score(X_test,Y_test))
print('Support Vector Machine Accuracy on test data: ',svm_cv.score(X_test,Y_test))
print('Decision Tree Accuracy on test data: ',tree_cv.score(X_test,Y_test))
print('K Nearest Neighour Accuracy on test data: ',knn_cv.score(X_test,Y_test))

print('The Decision Tree method performs best on test data')
```

```
Logistic RegressionAccuracy on test data:  0.8333333333333334
Support Vector Machine Accuracy on test data:  0.8333333333333334
Decision Tree Accuracy on test data:  0.8888888888888888
K Nearest Neighour Accuracy on test data:  0.8333333333333334
The Decision Tree method performs best on test data
```

Decision tree classifier model best parameters are:

{'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'best'}

RESULTS

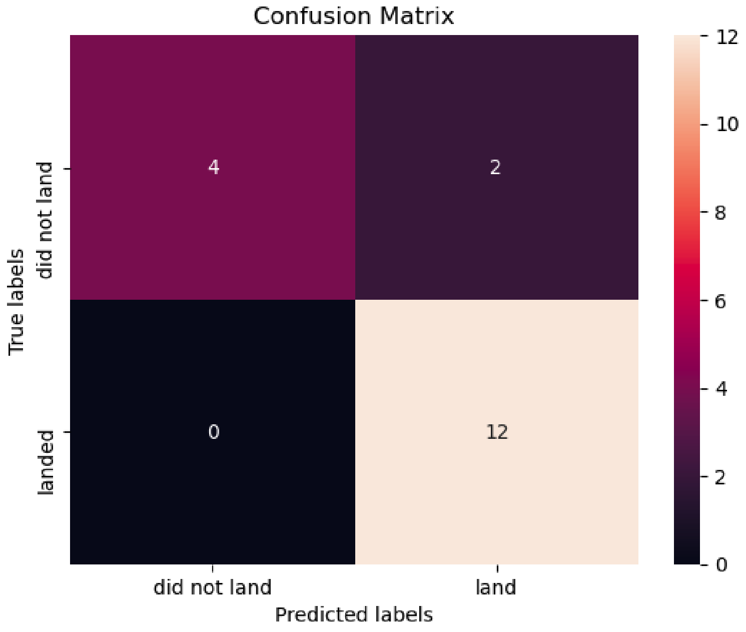
Predictive Analysis (Classification)

GridSearchCV Optimal Machine Learning Model and Hyper-parameters

Link to Notebook:
[Machine Learning Prediction Notebook]
[NB Viewer to show Notebook]

Decision Tree Confusion Matrix

Decision Tree Classifier Confusion Matrix was able to:
- Distinguish between different classes
- Provided improvement on false positives observed with other classifier models tested



CONCLUSION

What factors determine if the rocket will land successfully?

- Number of flights
- Orbit
- Payload
- Booster version
- Launch site

Interactions amongst various features that determine the success rate of a successful landing?

- As the flight number increases at a launch site, the higher the success rate at the launch site
- Orbits
  - ES-L1, GEO, HEO and SSO: highest success rate
  - Polar, LEO and ISS: heavier payloads have more successful landings
- Success rates increased from 2013 to 2020
- KSC LC-39A has the highest launch site success rate
- Payload Range 0-6000kg has the highest launch success rate
- F9 Booster version FT has the highest launch success rate

Which machine learning classifier best predicts if the first stage of a launch will land successfully?

- Decision tree classifier is best machine learning algorithm for this task

THANK YOU