Space X Falcon 9 First Stage Landing Prediction

Assignment: Machine Learning Prediction

Estimated time needed: **60** minutes

Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore in our ventures in the first stage will not, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch. In this lab you will create a machine learning prediction to predict if the first stage will land given the data from the preceding labs.

Several examples of an unsuccessful landing are shown here:

Most unsuccessful landings are planned. Space X performs a controlled landing in the oceans.

Objectives

Perform exploratory Data Analysis and determine Training Labels

- Create a training data from files data
- Find the best Machine Learning model
- Optimize Machine Learning Model

Exercises that accompanied the ETL (Extraction Transform and Load) Assignment:

- Find the number and frequency of each orbit in the dataset

Import Libraries and Define Auxiliary Functions

```
!pip install numpy
!pip install pandas
!pip install matplotlib
!pip install seaborn==0.9.0
```

We will import the following libraries for the lab

```
# Pandas is a software library written for the Python programming language for data manipulation and analysis.
import pandas as pd
# NumPy is the fundamental package for scientific computing with Python. It contains among other things: a powerful N-dimensional array object, sophisticated (broadcasting) functions,
# tools for integrating C/C++ and Fortran code, useful linear algebra, Fourier transform, and random number capabilities
# Matplotlib is a plotting library for python and pyplot gives us a MatLab like plotting framework. We will use this in our plotter function to plot data.
import numpy as np
import matplotlib.pyplot as plt
# Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics
import seaborn as sns
# Preprocessing allows us to standardize our data
from sklearn.preprocessing import preprocessing
# Allow the use of display to format output
from IPython.display import display, HTML
# Supressing warnings
import warnings
warnings.filterwarnings('ignore')
# Scikit-learn is a free software machine learning library for the Python programming language.
# It features various classification, regression and clustering algorithms
# including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.linear_regression import LinearRegression
# The Keras library provides a wide variety of building blocks for deep learning.
# We will use Keras to create a feature extraction model
from keras.models import Sequential
from keras.layers import Dense
# Scikit-learn also provides some utilities
from sklearn.utils import shuffle
# Matplotlib is a library for plotting
from matplotlib import rcParams
# Preserve aspect ratio when resizing images
rcParams['figure.autolayout'] = True
# Ignore warnings
import warnings
warnings.filterwarnings("ignore")
# Jupyter extension to render matplotlib plots inline
%matplotlib inline
# Jupyter extension to load variables from one notebook to another
%store -r
# Jupyter extension to show all outputs in a cell instead of just the last one
%load_ext jupyternotebooks.ipynb.ipyrun
# Jupyter extension to time execution of a cell
%load_ext jupyternotebooks.ipynb.ipytime
# Jupyter extension to suppress output of a cell
%load_ext jupyternotebooks.ipynb.ipysilent
```

```
[notice] A new release of pip available: 22.3.1 → 23.0.1
[notice] To update, run: python3.10 -m pip install --upgrade pip
Collecting numpy
  Downloading numpy-1.24.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (17.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17.3/17.3 MB 12.1 MB/s eta 0:00:00
Collecting pandas
  Downloading pandas-1.5.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (12.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.2/12.2 MB 12.1 MB/s eta 0:00:00
Collecting matplotlib
  Downloading matplotlib-3.7.1-cp310-cp310-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (9.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 9.2/9.2 MB 12.1 MB/s eta 0:00:00
Collecting seaborn==0.9.0
  Downloading seaborn-0.9.0-tar.gz (1.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.0/1.0 MB 12.1 MB/s eta 0:00:00
  Preparing metadata (setup.py) ...
Collecting scipy>=0.14.0
  Downloading scipy-1.10.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (34.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 34.4/34.4 MB 12.1 MB/s eta 0:00:00
Collecting matplotlib-inline
  Downloading matplotlib_inline-0.1.6-py3.10.egg
Collecting jupyternotebooks
  Downloading jupyternotebooks-0.0.0-py3.10.egg
Collecting jupyternotebooks.ipynb.ipysilent
  Downloading jupyternotebooks.ipynb.ipysilent-0.0.0-py3.10.egg
Collecting jupyternotebooks.ipynb.ipytime
  Downloading jupyternotebooks.ipynb.ipytime-0.0.0-py3.10.egg
Collecting pytz
  Downloading pytz-2023.3-py2.py3.egg
Collecting dateutil
  Downloading dateutil-2.8.2-py2.py3.egg
Collecting contourpy>=1.0.1
  Downloading contourpy-1.0.7-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (300 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 300.2/300.2 kB 12.1 MB/s eta 0:00:00
Collecting cycler>=0.10
  Downloading cycler-0.11.0-py3.10.egg
Collecting fonttools>=4.22.0
  Downloading fonttools-4.39.3-py3.10.egg
Collecting kiwisolver>=1.0.1
  Downloading kiwisolver-1.4.4-cp310-cp310-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (1.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 12.1 MB/s eta 0:00:00
Collecting packaging>=20.0
  Downloading packaging-23.0-py3.10.egg
Collecting pillow>=6.2.0
  Downloading pillow-9.5.0-cp310-cp310-manylinux_2_28_x86_64.whl (3.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.4/3.4 MB 12.1 MB/s eta 0:00:00
Collecting pyparsing>=2.3.1
  Downloading pyparsing-3.0.9-py3.10.egg
Collecting six>=1.5
  Downloading six-1.16.0-py2.py3.egg
Collecting numpy
  Downloading numpy-1.24.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (17.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17.3/17.3 MB 12.1 MB/s eta 0:00:00
Collecting pandas
  Downloading pandas-1.5.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (12.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.2/12.2 MB 12.1 MB/s eta 0:00:00
Collecting matplotlib
  Downloading matplotlib-3.7.1-cp310-cp310-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (9.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 9.2/9.2 MB 12.1 MB/s eta 0:00:00
Collecting seaborn==0.9.0
  Downloading seaborn-0.9.0-tar.gz (1.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.0/1.0 MB 12.1 MB/s eta 0:00:00
  Preparing metadata (setup.py) ...
Collecting scipy>=0.14.0
  Downloading scipy-1.10.1-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (34.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 34.4/34.4 MB 12.1 MB/s eta 0:00:00
Collecting matplotlib-inline
  Downloading matplotlib_inline-0.1.6-py3.10.egg
Collecting jupyternotebooks
  Downloading jupyternotebooks-0.0.0-py3.10.egg
Collecting jupyternotebooks.ipynb.ipysilent
  Downloading jupyternotebooks.ipynb.ipysilent-0.0.0-py3.10.egg
Collecting jupyternotebooks.ipynb.ipytime
  Downloading jupyternotebooks.ipynb.ipytime-0.0.0-py3.10.egg
Collecting pytz
  Downloading pytz-2023.3-py2.py3.egg
Collecting dateutil
  Downloading dateutil-2.8.2-py2.py3.egg
Collecting contourpy>=1.0.1
  Downloading contourpy-1.0.7-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (300 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 300.2/300.2 kB 12.1 MB/s eta 0:00:00
Collecting cycler>=0.10
  Downloading cycler-0.11.0-py3.10.egg
Collecting fonttools>=4.22.0
  Downloading fonttools-4.39.3-py3.10.egg
Collecting kiwisolver>=1.0.1
  Downloading kiwisolver-1.4.4-cp310-cp310-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (1.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 12.1 MB/s eta 0:00:00
Collecting packaging>=20.0
  Downloading packaging-23.0-py3.10.egg
Collecting pillow>=6.2.0
  Downloading pillow-9.5.0-cp310-cp310-manylinux_2_28_x86_64.whl (3.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.4/3.4 MB 12.1 MB/s eta 0:00:00
Collecting pyparsing>=2.3.1
  Downloading pyparsing-3.0.9-py3.10.egg
Collecting six>=1.5
  Downloading six-1.16.0-py2.py3.egg
```

Load the dataframe

Load the data

```
# Load dataset
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv")
# Preview data
df.head(5)
df = df.drop(df[df['Block'] == 5].index)
df[['Block', 'ReusedCount']].value_counts()
```

data/dataset0

FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude
0	2010-06-04	Falcon 9	6104.959412	LEO	CCSFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366
1	2012-05-22	Falcon 9	525.000000	LEO	CCSFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366
2	2013-03-01	Falcon 9	677.000000	ISS	CCSFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366
3	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829
4	2013-12-03	Falcon 9	3170.000000	GTO	CCSFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366

```
# Load dataset
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv")
# Preview data
df.head(5)
df = df.drop(df[df['Block'] == 5].index)
df[['Block', 'ReusedCount']].value_counts()
```

	FlightNumber	PayloadMass	Flights	Block	ReusedCount	Orbit_ES-L1	Orbit_GEO	Orbit_HEO	Orbit_ISS	...	Serial_B1056	Serial_B1058	Serial_B1059	Serial_B1060	Serial_B1062
0	1.0	0.710402	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
1	2.0	0.044732	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
2	3.0	0.271888	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
3	4.0	0.060267	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0
4	5.0	0.213445	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0

5 rows × 83 columns

TASK 1

Create a NumPy array from the column `Class` in `data`, by applying the method `.to_numpy()` then assign it to the variable `Y` (make sure to capital `Y` instead of lowercase `y`). The output type should be integer.

```
# Convert pandas df to np
Y = df['Class'].to_numpy()
# Preview first 5 values
print("First 5 Class values:", Y[:5])
```

```
First 5 Class values: [0 0 0 0 0]
```

TASK 2

Standardize the data in `X` then reassign it to the variable `X` using the transform provided below.

```
# Standard scaler
X = preprocessing.StandardScaler().fit(X).transform(X.astype(float))
# Preview first 5 rows of standardized features
X[:5]
```

```
array([[-1.71291154, -1.94814463, -0.65393146, -0.65393146,
        -0.41731474, -0.18732414, -0.38888889, -0.26666667,
        -0.08164966, -0.08164966, -0.08164966, -0.08164966
```