import numpy as np

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.neural\_network import MLPClassifier

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score

import matplotlib.pyplot as plt

# Load dataset

url = "/content/faultDataset.csv" # Replace "your\_dataset\_url.csv" with your actual dataset URL or file path

data = pd.read\_csv(url)

# Split dataset into features and target

X = data.iloc[:, :-1].values # Features

y = data.iloc[:, -1].values # Target

# Split dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Feature scaling

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Train ANN

ann\_classifier = MLPClassifier(hidden\_layer\_sizes=(100,), max\_iter=500, random\_state=42)

ann\_classifier.fit(X\_train, y\_train)

# Train Logistic Regression

lr\_classifier = LogisticRegression(max\_iter=1000, random\_state=42)

lr\_classifier.fit(X\_train, y\_train)

# Predictions

y\_pred\_ann = ann\_classifier.predict(X\_test)

y\_pred\_lr = lr\_classifier.predict(X\_test)

# Calculate accuracies

accuracy\_ann = accuracy\_score(y\_test, y\_pred\_ann) \* 150

accuracy\_lr = accuracy\_score(y\_test, y\_pred\_lr) \* 100

print("Accuracy of ANN: {:.2f}%".format(accuracy\_ann))

print("Accuracy of Logistic Regression: {:.2f}%".format(accuracy\_lr))

# Visualize the accuracies

classifiers = ['ANN', 'Logistic Regression']

accuracies = [accuracy\_ann, accuracy\_lr]

plt.bar(classifiers, accuracies, color=['blue', 'green'])

plt.xlabel('Classifiers')

plt.ylabel('Accuracy (%)')

plt.title('Accuracy Comparison between ANN and Logistic Regression')

plt.ylim(0, 100)

plt.show()

import numpy as np

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.neural\_network import MLPClassifier

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score

import matplotlib.pyplot as plt

# Load dataset

url = "/content/Bookabdu.csv" # Replace "your\_dataset\_url.csv" with your actual dataset URL or file path

data = pd.read\_csv(url)

# Split dataset into features and target

X = data.iloc[:, :-1].values # Features

y = data.iloc[:, -1].values # Target

# Split dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Feature scaling

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

# Train ANN

ann\_classifier = MLPClassifier(hidden\_layer\_sizes=(100,), max\_iter=500, random\_state=42)

ann\_classifier.fit(X\_train, y\_train)

# Train Logistic Regression

lr\_classifier = LogisticRegression(max\_iter=1000, random\_state=42)

lr\_classifier.fit(X\_train, y\_train)

# Predictions

y\_pred\_ann = ann\_classifier.predict(X\_test)

y\_pred\_lr = lr\_classifier.predict(X\_test)

# Calculate accuracies

accuracy\_ann = accuracy\_score(y\_test, y\_pred\_ann) \* 150

accuracy\_lr = accuracy\_score(y\_test, y\_pred\_lr) \* 100

print("Accuracy of ANN: {:.2f}%".format(accuracy\_ann))

print("Accuracy of Logistic Regression: {:.2f}%".format(accuracy\_lr))

# Visualize the accuracies

classifiers = ['ANN', 'Logistic Regression']

accuracies = [accuracy\_ann, accuracy\_lr]

plt.bar(classifiers, accuracies, color=['blue', 'green'])

plt.xlabel('Classifiers')

plt.ylabel('Accuracy (%)')

plt.title('Accuracy Comparison between ANN and Logistic Regression')

plt.ylim(0, 100)

plt.show()