import pandas as pd

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

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification\_report, confusion\_matrix, accuracy\_score

# Load the dataset

# Assuming the dataset is in a CSV file named 'machine\_data.csv'

df = pd.read\_csv('/data (1).csv')

# Inspect the first few rows of the dataset

print(df.head())

# Check column names and correct 'temp Mode' if needed

print(df.columns) # Print the column names to verify

# Features and target variable

# Use the correct column name for 'temp Mode' (e.g., 'tempMode' if it has no space)

X = df[['footfall', 'tempMode', 'AQ', 'USS', 'CS', 'VOC', 'RP', 'IP', 'Temperature']]

y = df['fail']

# Split the data into training and testing sets (80% train, 20% test)

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

# Initialize the Random Forest Classifier

clf = RandomForestClassifier(n\_estimators=100, random\_state=42)

# Train the classifier

clf.fit(X\_train, y\_train)

# Make predictions on the test set

y\_pred = clf.predict(X\_test)

# Evaluate the model

print("Confusion Matrix:")

print(confusion\_matrix(y\_test, y\_pred))

print("\nClassification Report:")

print(classification\_report(y\_test, y\_pred))

print("\nAccuracy Score:")

print(accuracy\_score(y\_test, y\_pred))

# Feature importance

feature\_importances = pd.DataFrame(clf.feature\_importances\_,

index = X\_train.columns,

columns=['importance']).sort\_values('importance', ascending=False)

print("\nFeature Importances:")

print(feature\_importances)