Name: Pulkit Arora

Group: 3CO10

Roll no : 102103267

Kaggle Team Name: 102103267\_Pulkit\_Arora

Kaggle ID: <https://www.kaggle.com/pulkittarora>

Code using Random Forest Regression

#Import Libraries

import pandas as pd

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

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean\_squared\_error as mse, r2\_score as r2

# Load training dataset

train\_data = pd.read\_parquet('/kaggle/input/2024ucs654labeval1007/Lab Eval/train\_data.parquet').sample(frac=0.6, random\_state=42)

# Load test dataset

test\_data = pd.read\_parquet('/kaggle/input/2024ucs654labeval1007/Lab Eval/test\_data.parquet')

# Reset index for both dataframes

test\_data.reset\_index(inplace=True)

train\_data.reset\_index(inplace=True)

# Preprocess data

test\_ids = test\_data['id']

test\_data = test\_data.drop('data\_type', axis=1)

train\_data = train\_data.drop(['id', 'era', 'data\_type', 'T0', 'T1', 'T2', 'T3', 'T4', 'T5', 'T6', 'T8', 'T9'], axis=1)

# Convert numeric columns to categorical

num\_cols = train\_data.select\_dtypes(include=['int8']).columns

train\_data[num\_cols] = train\_data[num\_cols].astype('category')

# Select features and target variable

features = train\_data.drop('T7', axis=1)

target = train\_data['T7']

# Split data into training and testing sets

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

# Initialize Random Forest Regressor

rf\_model = RandomForestRegressor(random\_state=42, n\_jobs=-1)

# Train the model

rf\_model.fit(X\_train, y\_train)

# Make predictions on the testing set

y\_pred\_test = rf\_model.predict(X\_test)

# Evaluate the model

mean\_error = mse(y\_test, y\_pred\_test)

r\_squared = r2(y\_test, y\_pred\_test)

print("Mean Squared Error:", mean\_error)

print("R-squared:", r\_squared)

# Make predictions on the test data

pred\_test = rf\_model.predict(test\_data)

# Create submission dataframe

submission = pd.DataFrame({'ID': test\_ids, 'Target': pred\_test})

# Save submission file

submission.to\_csv('submission.csv', index=False)

submission.head()