import pandas as pd

from sklearn.ensemble import RandomForestClassifier

from xgboost import XGBClassifier

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

from sklearn.metrics import classification report

from sklearn.ensemble import VotingClassifier

df pd.read\_csv("child\_learning data.csv")

x= df.drop("EngagementLabel", axis-1)

y df["Engagement Label"]

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

rf\_model RandomForestClassifier(n\_estimators=100, random state=42)

xgb\_model XGBClassifier(use\_label\_encoder False, eval metric'logloss')

hybrid\_model votingClassifier(estimators=[('rf', rf\_model), ('xgb, xgb\_model)], voting'soft')

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

y\_pred hybrid\_model.predict(X\_test)

print(classification report(y test, y pred))

using UnityEngine;

using UnityEngine.UI;

public class AlFeedbackManager: MonoBehaviour

{

public Text feedbackText;

public float playerScore;

void Start()

{

UpdateFeedback(playerScore);

}

public void UpdateFeedback(float score)

{

if (score >= 0.8f)

feedbackText.text = "Great job! Ready for a tougher challenge?";

else if (score >= 0.5f)

feedbackText.text = "You're doing well! Keep going!";

else

feedbackText.text = "Need a hint? Let's try a simpler puzzlel";

}

}