. You're right. The error occurs because the `predictor` variable is not in scope for the `evaluate\_model()` function. I'll modify the code to pass the scaler as a parameter.

```python

import numpy as np

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

def evaluate\_model(y\_true, y\_pred, scaler):

"""

Calculate and print model evaluation metrics.

Parameters:

y\_true (array-like): True target values

y\_pred (array-like): Predicted target values

scaler (MinMaxScaler): Scaler used to inverse transform the data

"""

# Inverse transform the scaled predictions and true values

y\_true\_original = scaler.inverse\_transform(y\_true.reshape(-1, 1)).flatten()

y\_pred\_original = scaler.inverse\_transform(y\_pred.reshape(-1, 1)).flatten()

# Calculate metrics

mae = mean\_absolute\_error(y\_true\_original, y\_pred\_original)

r2 = r2\_score(y\_true\_original, y\_pred\_original)

mse = mean\_squared\_error(y\_true\_original, y\_pred\_original)

rmse = np.sqrt(mse)

# Print evaluation metrics

print("\nModel Evaluation Metrics:")

print(f"Mean Absolute Error (MAE): {mae:.4f}")

print(f"Root Mean Squared Error (RMSE): {rmse:.4f}")

print(f"Mean Squared Error (MSE): {mse:.4f}")

print(f"R-squared (R2) Score: {r2:.4f}")

def main():

CSV\_PATH = 'lottery\_data.csv'

predictor = DateIncorporatedLotteryPredictor()

try:

data = predictor.load\_data(CSV\_PATH)

X, y = predictor.prepare\_data(data, window\_size=10)

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

predictor.build\_lstm\_model(input\_shape=(X.shape[1], 2))

history = predictor.train(X\_train, y\_train, epochs=50, batch\_size=32)

predictor.plot\_training\_history(history)

# Make predictions on the test set

y\_pred = predictor.model.predict(X\_test)

# Evaluate the model

evaluate\_model(y\_test, y\_pred, predictor.scaler)

recent\_data = X\_test[-1][:, 0]

recent\_dates = X\_test[-1][:, 1]

predicted\_number = predictor.predict(recent\_data, recent\_dates)

print(f"Predicted Lottery Number: {predicted\_number}")

except Exception as e:

print(f"An error occurred: {e}")

if \_\_name\_\_ == "\_\_main\_\_":

main()

```

The key changes are:

1. Modified `evaluate\_model()` to take the scaler as a parameter

2. Passed `predictor.scaler` when calling `evaluate\_model()`

This should resolve the "predictor is not defined" error. The function now receives the scaler directly, making it more flexible and independent of the global scope.

Make sure you have the following imports at the top of your script:

```python

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

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

```

Would you like me to explain anything further about the evaluation metrics or the code?