

Project Development Phase Model Performance Test

Date	10 November 2022
Team ID	PNT2022TMID10046
Project Name	Project - Crude Oil Price Prediction
Maximum Marks	10 Marks

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Model Summary	-	<p>Model Building</p> <p>Importing The Model Building Libraries</p> <pre>In []: from tensorflow.keras.models import Sequential from tensorflow.keras.layers import Dense from tensorflow.keras.layers import LSTM</pre> <p>Initializing The Model</p> <pre>In []: model = Sequential()</pre> <p>Adding LSTM Layers</p> <pre>In []: model.add(LSTM(50, return_sequences=True, input_shape=(10,1))) model.add(LSTM(50, return_sequences=True)) model.add(LSTM(50))</pre> <p>Adding Output Layers</p> <pre>In []: model.add(Dense(1))</pre> <pre>In []: model.summary()</pre> <pre>Model: "sequential_1" Layer (type) Output Shape Param # ----- lstm_3 (LSTM) (None, 10, 50) 10400 lstm_4 (LSTM) (None, 10, 50) 20200 lstm_5 (LSTM) (None, 50) 20200 dense_1 (Dense) (None, 1) 51 ----- Total params: 50,851 Trainable params: 50,851 Non-trainable params: 0</pre> <p>Configure The Learning Process</p> <pre>In []: model.compile(loss='mean_squared_error', optimizer='adam')</pre> <p>Train The Model</p> <pre>In []: model.fit(X_train, y_train, validation_data = (X_test, ytest), epochs=50, batch_size=64, verbose=1)</pre>

2.	Accuracy	<div>Training Accuracy - mean_squared_error : 0.005620035171812541</div> <div>Validation Accuracy – mean_squared_error : 0.013970680158126952</div>	<div><div>Train The Model</div><div><div>In []:</div><div>model.fit(X_train, y_train, validation_data = (X_test, ytest), epochs=50, batch_size=64, verbose=1)</div></div><div><div>Epoch 1/50</div><div>84/84 [=====] - 9s 42ms/step - loss: 0.0018 - val_loss: 9.8627e-04</div><div>Epoch 2/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 1.2840e-04 - val_loss: 7.8349e-04</div><div>Epoch 3/50</div><div>84/84 [=====] - 2s 27ms/step - loss: 1.2931e-04 - val_loss: 7.6868e-04</div><div>Epoch 4/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 1.2753e-04 - val_loss: 7.8465e-04</div><div>Epoch 5/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 1.3176e-04 - val_loss: 8.3704e-04</div><div>Epoch 6/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 1.3144e-04 - val_loss: 7.7508e-04</div><div>Epoch 7/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 1.2721e-04 - val_loss: 8.3512e-04</div><div>Epoch 8/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 1.2070e-04 - val_loss: 7.8521e-04</div><div>Epoch 9/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 1.1302e-04 - val_loss: 0.0011</div><div>Epoch 10/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 1.1513e-04 - val_loss: 9.1192e-04</div><div>Epoch 11/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 1.1818e-04 - val_loss: 6.5657e-04</div><div>Epoch 12/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 1.0612e-04 - val_loss: 6.5551e-04</div><div>Epoch 13/50</div><div>84/84 [=====] - 3s 31ms/step - loss: 1.0874e-04 - val_loss: 7.8740e-04</div><div>Epoch 14/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 1.1205e-04 - val_loss: 7.6408e-04</div><div>Epoch 15/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 1.0192e-04 - val_loss: 8.0062e-04</div><div>Epoch 16/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 1.0737e-04 - val_loss: 5.5239e-04</div><div>Epoch 17/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 4.9269e-05 - val_loss: 2.0577e-04</div><div>Epoch 18/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 4.7191e-05 - val_loss: 3.8797e-04</div><div>Epoch 19/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 4.9790e-05 - val_loss: 1.9337e-04</div><div>Epoch 20/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 4.4857e-05 - val_loss: 2.7048e-04</div><div>Epoch 21/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 4.0470e-05 - val_loss: 2.1115e-04</div><div>Epoch 22/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 4.2266e-05 - val_loss: 1.9419e-04</div><div>Epoch 23/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 4.4702e-05 - val_loss: 1.9286e-04</div><div>Epoch 24/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 4.0846e-05 - val_loss: 1.8738e-04</div><div>Epoch 25/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 3.8158e-05 - val_loss: 2.5169e-04</div><div>Epoch 26/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 3.9268e-05 - val_loss: 2.2022e-04</div><div>Epoch 27/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 4.0808e-05 - val_loss: 4.9809e-04</div><div>Epoch 28/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 3.5735e-05 - val_loss: 2.5024e-04</div><div>Epoch 29/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 3.4770e-05 - val_loss: 2.2390e-04</div><div>Epoch 30/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 3.3905e-05 - val_loss: 2.2329e-04</div><div>Epoch 31/50</div><div>84/84 [=====] - 2s 26ms/step - loss: 3.5018e-05 - val_loss: 2.1140e-04</div><div>Epoch 32/50</div><div>84/84 [=====] - 3s 39ms/step - loss: 3.4990e-05 - val_loss: 1.9518e-04</div></div><div><div>Out[]:</div></div><div><div>Model Evaluation</div><div><div>In []:</div><div>train_predict = model.predict(X_train) test_predict = model.predict(X_test) train_predict.shape, test_predict.shape</div></div><div><div>167/167 [=====] - 2s 5ms/step</div><div>90/90 [=====] - 0s 5ms/step</div></div><div><div>Out[]:</div><div>((5329, 1), (2865, 1))</div></div><div><div>In []:</div><div>import math from sklearn.metrics import mean_squared_error</div></div><div><div>In []:</div><div>math.sqrt(mean_squared_error(y_train, train_predict))</div></div><div><div>Out[]:</div><div>0.005620035171812541</div></div><div><div>In []:</div><div>math.sqrt(mean_squared_error(ytest, test_predict))</div></div><div><div>Out[]:</div><div>0.013970680158126952</div></div></div></div>
3.	Confidence Score (Only Yolo Projects)	<div>Class Detected -</div> <div>Confidence Score -</div>	<div>-</div>