Project Check-in 5

```
%pip install --upgrade pip
%pip install scikit-lego
%pip install seaborn
%pip install nbstripout
!nbstripout --install
```

```
Requirement already satisfied: pip in c:\users\isaac\appdata\local\
programs\python\python311\lib\site-packages (24.3.1)
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: scikit-lego in c:\users\isaac\appdata\
local\programs\python\python311\lib\site-packages (0.9.1)
Requirement already satisfied: narwhals>=1.0.0 in c:\users\isaac\
appdata\local\programs\python\python311\lib\site-packages (from
scikit-lego) (1.9.3)
Requirement already satisfied: pandas>=1.1.5 in c:\users\isaac\
appdata\local\programs\python\python311\lib\site-packages (from
scikit-lego) (2.1.2)
Requirement already satisfied: scikit-learn>=1.0 in c:\users\isaac\
appdata\local\programs\python\python311\lib\site-packages (from
scikit-lego) (1.3.2)
Requirement already satisfied: numpy<2,>=1.23.2 in c:\users\isaac\
appdata\local\programs\python\python311\lib\site-packages (from
pandas>=1.1.5->scikit-lego) (1.26.1)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\
isaac\appdata\roaming\python\python311\site-packages (from
pandas>=1.1.5->scikit-lego) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\isaac\appdata\
local\programs\python\python311\lib\site-packages (from pandas>=1.1.5-
>scikit-lego) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\isaac\
appdata\local\programs\python\python311\lib\site-packages (from
pandas>=1.1.5->scikit-lego) (2023.3)
Requirement already satisfied: scipy>=1.5.0 in c:\users\isaac\appdata\
local\programs\python\python311\lib\site-packages (from scikit-
learn>=1.0->scikit-lego) (1.11.3)
Requirement already satisfied: joblib>=1.1.1 in c:\users\isaac\
appdata\local\programs\python\python311\lib\site-packages (from
scikit-learn>=1.0->scikit-lego) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\isaac\
appdata\local\programs\python\python311\lib\site-packages (from
scikit-learn>=1.0->scikit-lego) (3.2.0)
Requirement already satisfied: six>=1.5 in c:\users\isaac\appdata\
roaming\python\python311\site-packages (from python-dateutil>=2.8.2-
>pandas>=1.1.5->scikit-lego) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: seaborn in c:\users\isaac\appdata\
local\programs\python\python311\lib\site-packages (0.13.2)
```

```
Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\users\isaac\
appdata\local\programs\python\python311\lib\site-packages (from
seaborn) (1.26.1)
Requirement already satisfied: pandas>=1.2 in c:\users\isaac\appdata\
local\programs\python\python311\lib\site-packages (from seaborn)
(2.1.2)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in c:\users\
isaac\appdata\local\programs\python\python311\lib\site-packages (from
seaborn) (3.8.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\isaac\
appdata\local\programs\python\python311\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (1.1.1)
Requirement already satisfied: cycler>=0.10 in c:\users\isaac\appdata\
local\programs\python\python311\lib\site-packages (from matplotlib!
=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\isaac\
appdata\local\programs\python\python311\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (4.43.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\isaac\
appdata\local\programs\python\python311\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (1.4.5)
Requirement already satisfied: packaging>=20.0 in c:\users\isaac\
appdata\roaming\python\python311\site-packages (from matplotlib!
=3.6.1,>=3.4->seaborn) (23.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\isaac\
appdata\local\programs\python\python311\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (10.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\isaac\
appdata\local\programs\python\python311\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\isaac\
appdata\roaming\python\python311\site-packages (from matplotlib!
=3.6.1,>=3.4->seaborn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\isaac\appdata\
local\programs\python\python311\lib\site-packages (from pandas>=1.2-
>seaborn) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\isaac\
appdata\local\programs\python\python311\lib\site-packages (from
pandas>=1.2->seaborn) (2023.3)
Requirement already satisfied: six>=1.5 in c:\users\isaac\appdata\
roaming\python\python311\site-packages (from python-dateutil>=2.7-
>matplotlib!=3.6.1,>=3.4->seaborn) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: nbstripout in c:\users\isaac\appdata\
local\programs\python\python311\lib\site-packages (0.7.1)
Requirement already satisfied: nbformat in c:\users\isaac\appdata\
local\programs\python\python311\lib\site-packages (from nbstripout)
(5.10.4)
Requirement already satisfied: fastjsonschema>=2.15 in c:\users\isaac\
```

```
appdata\local\programs\python\python311\lib\site-packages (from
nbformat->nbstripout) (2.20.0)
Requirement already satisfied: jsonschema>=2.6 in c:\users\isaac\
appdata\local\programs\python\python311\lib\site-packages (from
nbformat->nbstripout) (4.23.0)
Requirement already satisfied: jupyter-core!=5.0.*,>=4.12 in c:\users\
isaac\appdata\roaming\python\python311\site-packages (from nbformat-
>nbstripout) (5.4.0)
Requirement already satisfied: traitlets>=5.1 in c:\users\isaac\
appdata\roaming\python\python311\site-packages (from nbformat-
>nbstripout) (5.11.2)
Requirement already satisfied: attrs>=22.2.0 in c:\users\isaac\
appdata\local\programs\python\python311\lib\site-packages (from
jsonschema>=2.6->nbformat->nbstripout) (24.2.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in
c:\users\isaac\appdata\local\programs\python\python311\lib\site-
packages (from jsonschema>=2.6->nbformat->nbstripout) (2024.10.1)
Requirement already satisfied: referencing>=0.28.4 in c:\users\isaac\
appdata\local\programs\python\python311\lib\site-packages (from
jsonschema>=2.6->nbformat->nbstripout) (0.35.1)
Requirement already satisfied: rpds-py>=0.7.1 in c:\users\isaac\
appdata\local\programs\python\python311\lib\site-packages (from
jsonschema>=2.6->nbformat->nbstripout) (0.20.0)
Requirement already satisfied: platformdirs>=2.5 in c:\users\isaac\
appdata\roaming\python\python311\site-packages (from jupyter-core!
=5.0.*,>=4.12->nbformat->nbstripout) (3.11.0)
Requirement already satisfied: pywin32>=300 in c:\users\isaac\appdata\
roaming\python\python311\site-packages (from jupyter-core!
=5.0.*,>=4.12->nbformat->nbstripout) (306)
Note: you may need to restart the kernel to use updated packages.
```

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from sklearn.preprocessing import StandardScaler
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
import keras
df = pd.read_csv("./dataset.csv")

# Step 1: Clean Data
# Remove duplicates
df_cleaned = df.drop(columns='Unnamed:
0').drop_duplicates(subset=['track_id','album_name','artists','track_n
ame'])

# Remove columns with every row unique. Also dropping artist and album
```

```
because it would be too much one-hot encoding
df_cleaned.drop(columns=['track_id', 'track_name',
'artists','album_name'], inplace=True)
df_cleaned.dropna(axis=0,inplace=True)
df_cleaned.reset_index(drop=True, inplace=True)

#The columns with object datatype will be categorical
columns = df_cleaned.select_dtypes(include=['int64',
'float64']).columns.tolist()
df_cleaned = df_cleaned[columns]

y = df_cleaned["popularity"]
y = np.array([float(i) for i in y])
print(y)
scaler = StandardScaler() # Scale the data so that the variances for
each feature can be similarly weighted
df_cleaned = scaler.fit_transform(df_cleaned)
df_cleaned = pd.DataFrame(df_cleaned, columns=columns)
```

```
[73. 55. 57. ... 22. 41. 22.]
```

```
df_cleaned
```

```
       popularity  duration_ms  danceability    energy       key
loudness  \
0        1.933926     0.013495      0.644260 -0.675976 -1.203286
0.335731
1        1.059320    -0.704151     -0.804604 -1.825609 -1.203286 -
1.673094
2        1.156499    -0.162163     -0.702731 -1.073476 -1.484194 -
0.236523
3        1.836747    -0.240899     -1.676186 -2.240257 -1.484194 -
1.918236
4        2.371228    -0.268168      0.316001 -0.746123 -0.922377 -
0.226373
...           ...          ...           ...       ...       ...
...
89736   -0.592712     1.379913     -2.208191 -1.556712 -0.079651 -
1.511837
89737   -0.544123     1.379922     -2.196872 -2.016565 -1.484194 -
1.880508
89738   -0.544123     0.374726      0.378257 -1.190388 -1.484194 -
0.458875
89739    0.379072     0.484751      0.140553 -0.500608  0.482166 -
0.457725
89740   -0.544123     0.112302     -0.204684 -0.574653 -1.203286 -
0.326536

           mode  speechiness  acousticness  instrumentalness  liveness
\
```

```
0     -1.324600     0.490464     -0.875177          -0.535478  0.723666
1      0.754945    -0.098361      1.760797          -0.535464 -0.595072
2      0.754945    -0.280217     -0.349638          -0.535481 -0.512971
3      0.754945    -0.451480      1.704637          -0.535263 -0.436002
4      0.754945    -0.307584      0.415912          -0.535481 -0.687948
...         ...          ...           ...                ...       ...
89736  0.754945    -0.399395      0.921352           2.330077 -0.670501
89737 -1.324600    -0.417934      1.967702           2.478296 -0.574547
89738 -1.324600    -0.401160      1.592317          -0.535481 -0.682816
89739  0.754945    -0.509744      0.155803          -0.535481  0.272114
89740 -1.324600    -0.131908      1.042540          -0.535481 -0.655108


        valence     tempo  time_signature
0      0.934036 -1.133609        0.226215
1     -0.770280 -1.479854        0.226215
2     -1.329508 -1.518271        0.226215
3     -1.242010  1.981637       -1.979187
4     -1.150708 -0.070037        0.226215
...         ...       ...             ...
89736 -1.657057  0.130711        2.431616
89737 -1.652872 -1.222528        0.226215
89738  1.040556  0.342649        0.226215
89739 -0.214856  0.461584        0.226215
89740  0.907406 -1.423110        0.226215

[89741 rows x 14 columns]
```

1. We evaluated our NN based on the mean squared error between predicted and true values for popularity on our validation dataset. We also graphed predicted values vs true values for popularity.

2. We trained our NN using the built-in learning framework for keras, which utilizes batch gradient descent. Learning rate of 0.001 is recommended for Adam optimizer. Adam will adaptively adjust the learning rate based on an exponentially weighted history of the gradients, so we have "momentum" built-in to our learning rate.

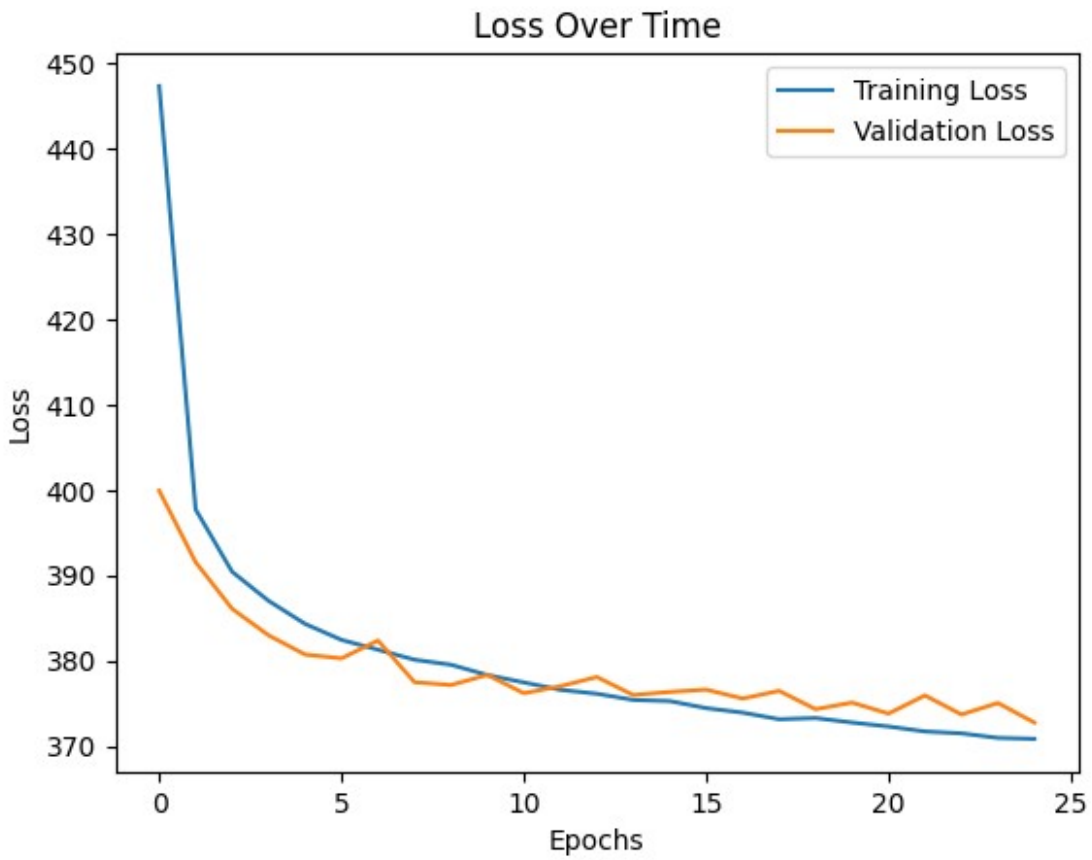
```
X = df_cleaned.drop(columns="popularity")
print(X.shape)
print(y.shape)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)

model = keras.Sequential([
    keras.layers.Input(shape=(13,)),
    keras.layers.Dense(64, activation='relu'),  # Fully connected
layer with 64 units
    keras.layers.Dense(64, activation='relu'),  # Fully connected
layer with 64 units
    keras.layers.Dense(1)  # Output layer for 10 classes
])

# Compile the model
model.compile(optimizer=keras.optimizers.Adam(learning_rate=0.001),
              loss='mean_squared_error',
              metrics=['root_mean_squared_error'])

# Train the model
history = model.fit(X_train, y_train, epochs=25,
validation_data=(X_test, y_test))
# Plot the training and validation loss
plt.plot(history.history['loss'], label='Training Loss')
if 'val_loss' in history.history:
    plt.plot(history.history['val_loss'], label='Validation Loss')

plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.title('Loss Over Time')
plt.legend()
plt.show()

# Generate predictions
predictions = model.predict(X_test).flatten()

# Create a DataFrame to compare predictions with real values
comparison_df = pd.DataFrame({
    'Real Value': y_test,
    'Predicted Value': predictions
})
plt.xlabel("True Value")
plt.ylabel("Predicted Value")
plt.scatter(y_test, predictions)
# Display a sample of the table
print(comparison_df.sample(10))  # Show 10 random samples
```

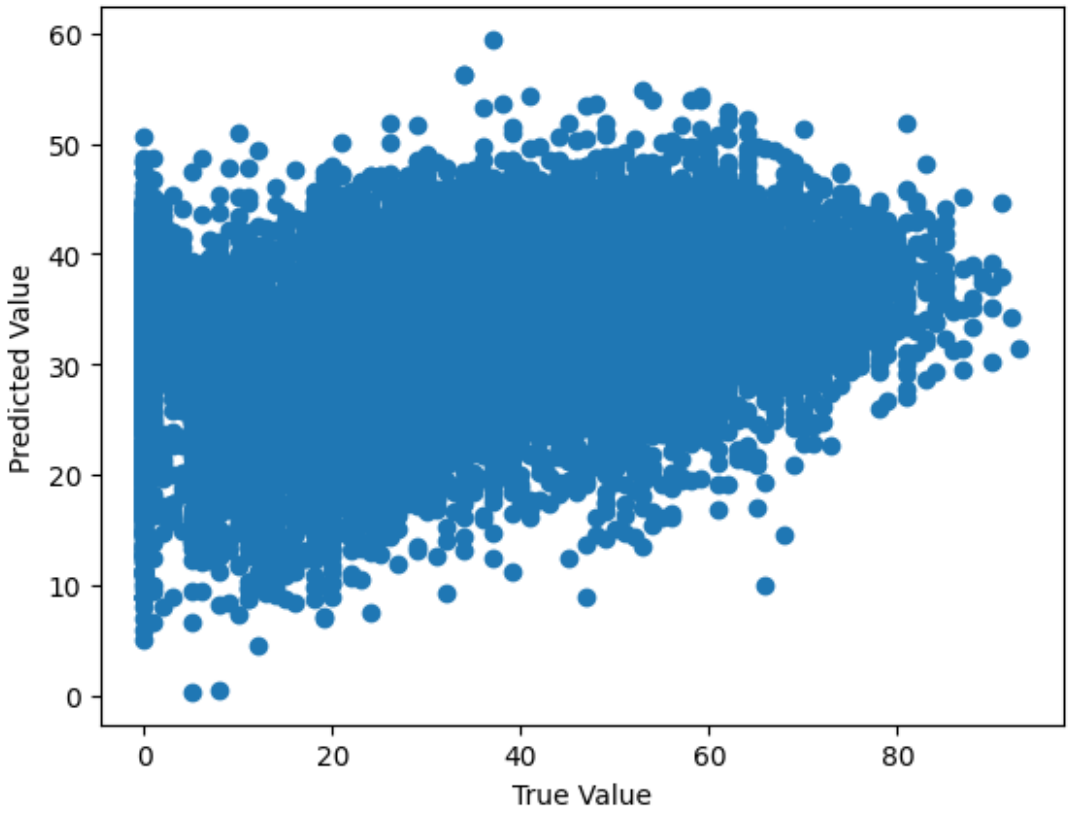
```
(89741, 13)
(89741,)
Epoch 1/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 6s 2ms/step - loss: 560.9721 -
root_mean_squared_error: 23.4052 - val_loss: 399.9182 -
```

```
val_root_mean_squared_error: 19.9979
Epoch 2/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 401.8094 -
root_mean_squared_error: 20.0450 - val_loss: 391.5348 -
val_root_mean_squared_error: 19.7871
Epoch 3/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 389.2230 -
root_mean_squared_error: 19.7282 - val_loss: 386.0774 -
val_root_mean_squared_error: 19.6490
Epoch 4/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 389.5697 -
root_mean_squared_error: 19.7366 - val_loss: 382.9676 -
val_root_mean_squared_error: 19.5695
Epoch 5/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 382.9096 -
root_mean_squared_error: 19.5680 - val_loss: 380.7141 -
val_root_mean_squared_error: 19.5120
Epoch 6/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 380.7110 -
root_mean_squared_error: 19.5117 - val_loss: 380.2666 -
val_root_mean_squared_error: 19.5006
Epoch 7/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 5s 2ms/step - loss: 382.1404 -
root_mean_squared_error: 19.5480 - val_loss: 382.3564 -
val_root_mean_squared_error: 19.5542
Epoch 8/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 380.3947 -
root_mean_squared_error: 19.5035 - val_loss: 377.4754 -
val_root_mean_squared_error: 19.4288
Epoch 9/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 381.5798 -
root_mean_squared_error: 19.5337 - val_loss: 377.1416 -
val_root_mean_squared_error: 19.4203
Epoch 10/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 377.1526 -
root_mean_squared_error: 19.4203 - val_loss: 378.3247 -
val_root_mean_squared_error: 19.4507
Epoch 11/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 376.1411 -
root_mean_squared_error: 19.3936 - val_loss: 376.1795 -
val_root_mean_squared_error: 19.3956
Epoch 12/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 377.7834 -
root_mean_squared_error: 19.4364 - val_loss: 376.9943 -
val_root_mean_squared_error: 19.4165
Epoch 13/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 5s 2ms/step - loss: 379.1269 -
root_mean_squared_error: 19.4708 - val_loss: 378.0804 -
val_root_mean_squared_error: 19.4445
```

```
Epoch 14/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 374.2245 -
root_mean_squared_error: 19.3447 - val_loss: 375.9724 -
val_root_mean_squared_error: 19.3903
Epoch 15/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 372.5235 -
root_mean_squared_error: 19.3007 - val_loss: 376.3322 -
val_root_mean_squared_error: 19.3995
Epoch 16/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 372.8218 -
root_mean_squared_error: 19.3085 - val_loss: 376.6034 -
val_root_mean_squared_error: 19.4065
Epoch 17/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 372.9012 -
root_mean_squared_error: 19.3103 - val_loss: 375.5573 -
val_root_mean_squared_error: 19.3796
Epoch 18/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 373.1988 -
root_mean_squared_error: 19.3178 - val_loss: 376.4624 -
val_root_mean_squared_error: 19.4028
Epoch 19/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 375.0405 -
root_mean_squared_error: 19.3653 - val_loss: 374.3104 -
val_root_mean_squared_error: 19.3472
Epoch 20/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 370.5438 -
root_mean_squared_error: 19.2478 - val_loss: 375.0804 -
val_root_mean_squared_error: 19.3673
Epoch 21/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 372.3239 -
root_mean_squared_error: 19.2954 - val_loss: 373.7919 -
val_root_mean_squared_error: 19.3339
Epoch 22/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 371.0252 -
root_mean_squared_error: 19.2618 - val_loss: 375.9265 -
val_root_mean_squared_error: 19.3890
Epoch 23/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 374.2224 -
root_mean_squared_error: 19.3441 - val_loss: 373.6843 -
val_root_mean_squared_error: 19.3311
Epoch 24/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 369.6274 -
root_mean_squared_error: 19.2255 - val_loss: 375.0359 -
val_root_mean_squared_error: 19.3661
Epoch 25/25
2244/2244 ━━━━━━━━━━━━━━━━━━━━ 4s 2ms/step - loss: 371.2413 -
root_mean_squared_error: 19.2675 - val_loss: 372.7372 -
val_root_mean_squared_error: 19.3066
```




```
561/561 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step
       Real Value  Predicted Value
836          45.0        45.661858
8675         22.0        26.295349
7622         70.0        36.315182
137          23.0        26.052212
1595         57.0        33.894871
13905        48.0        33.942513
316          54.0        32.518490
12455         0.0        24.308922
10499        48.0        26.395571
488          46.0        45.782616
```


Predicted Value
True Value
0
10
20
30
40
50
60
0
20
40
60
80