

```
In [1]: # Set the seed value for the notebook so the results are reproducible
from numpy.random import seed
seed(1)
```

```
In [2]: # Dependencies
import numpy as np
import pandas as pd
```

```
In [3]: import tensorflow
tensorflow.keras.__version__
```

```
Out[3]: '2.2.4-tf'
```

```
In [4]: df = pd.read_csv("main_data_tree2.csv")
df.head(10)
```

```
Out[4]:
```

	EOG_quintile	percent_EDS	parent_unemployed	county_poc	median_inc_county
0	4	96.0	11.3	50.3	40433
1	4	96.0	8.3	21.6	53419
2	3	96.0	8.6	56.6	59329
3	4	96.0	19.3	74.2	35407
4	4	4.0	9.5	42.7	50112
5	1	4.0	2.8	11.5	64426
6	1	4.0	12.7	52.5	44015
7	1	4.0	4.0	39.1	80169
8	3	4.0	9.2	35.7	57710
9	1	4.0	4.0	39.1	80169

Data Pre-Processing

```
In [5]: X = df.drop("EOG_quintile", axis=1)
y = df["EOG_quintile"]
print(X.shape, y.shape)
```

```
(2492, 4) (2492,)
```

```
In [6]: from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, MinMaxScaler
from tensorflow.keras.utils import to_categorical
```

```
In [7]: X_train, X_test, y_train, y_test = train_test_split(
X, y, random_state=1, stratify=y)
```

```
In [8]: X_scaler = MinMaxScaler().fit(X_train)
X_train_scaled = X_scaler.transform(X_train)
X_test_scaled = X_scaler.transform(X_test)
```

```
In [9]: # Step 1: Label-encode data set
label_encoder = LabelEncoder()
label_encoder.fit(y_train)
encoded_y_train = label_encoder.transform(y_train)
encoded_y_test = label_encoder.transform(y_test)
```

```
In [10]: # Step 2: Convert encoded labels to one-hot-encoding
y_train_categorical = to_categorical(encoded_y_train)
y_test_categorical = to_categorical(encoded_y_test)
```

Create a Deep Learning Model

```
In [11]: from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

```
In [12]: # Create model and add layers
model = Sequential()
model.add(Dense(units=100, activation='relu', input_dim=4))
model.add(Dense(units=100, activation='relu'))
model.add(Dense(units=5, activation='softmax'))
```

WARNING:tensorflow:From C:\Users\petef\Anaconda3\envs\PythonData\lib\site-packages\tensorflow\python\ops\init_ops.py:1251: calling VarianceScaling.__init__ (from tensorflow.python.ops.init_ops) with dtype is deprecated and will be removed in a future version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it to the constructor

```
In [13]: # Compile and fit the model
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

In [14]: `model.summary()`

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 100)	500
dense_1 (Dense)	(None, 100)	10100
dense_2 (Dense)	(None, 5)	505

=====
Total params: 11,105
Trainable params: 11,105
Non-trainable params: 0
=====

```
In [15]: model.fit(  
    X_train_scaled,  
    y_train_categorical,  
    epochs=60,  
    shuffle=True,  
    verbose=2  
)
```

Epoch 1/60
1869/1869 - 0s - loss: 1.3764 - acc: 0.4195
Epoch 2/60
1869/1869 - 0s - loss: 1.1821 - acc: 0.5024
Epoch 3/60
1869/1869 - 0s - loss: 1.1004 - acc: 0.5361
Epoch 4/60
1869/1869 - 0s - loss: 1.0611 - acc: 0.5324
Epoch 5/60
1869/1869 - 0s - loss: 1.0375 - acc: 0.5484
Epoch 6/60
1869/1869 - 0s - loss: 1.0341 - acc: 0.5431
Epoch 7/60
1869/1869 - 0s - loss: 1.0138 - acc: 0.5591
Epoch 8/60
1869/1869 - 0s - loss: 1.0153 - acc: 0.5639
Epoch 9/60
1869/1869 - 0s - loss: 1.0121 - acc: 0.5639
Epoch 10/60
1869/1869 - 0s - loss: 1.0058 - acc: 0.5645
Epoch 11/60
1869/1869 - 0s - loss: 1.0018 - acc: 0.5688
Epoch 12/60
1869/1869 - 0s - loss: 1.0015 - acc: 0.5629
Epoch 13/60
1869/1869 - 0s - loss: 1.0000 - acc: 0.5677
Epoch 14/60
1869/1869 - 0s - loss: 0.9971 - acc: 0.5650
Epoch 15/60
1869/1869 - 0s - loss: 0.9965 - acc: 0.5704
Epoch 16/60
1869/1869 - 0s - loss: 0.9948 - acc: 0.5688
Epoch 17/60
1869/1869 - 0s - loss: 0.9917 - acc: 0.5714
Epoch 18/60
1869/1869 - 0s - loss: 0.9913 - acc: 0.5811
Epoch 19/60
1869/1869 - 0s - loss: 0.9914 - acc: 0.5730
Epoch 20/60
1869/1869 - 0s - loss: 0.9891 - acc: 0.5757
Epoch 21/60
1869/1869 - 0s - loss: 0.9898 - acc: 0.5757
Epoch 22/60
1869/1869 - 0s - loss: 0.9873 - acc: 0.5768
Epoch 23/60
1869/1869 - 0s - loss: 0.9845 - acc: 0.5762
Epoch 24/60
1869/1869 - 0s - loss: 0.9895 - acc: 0.5752
Epoch 25/60
1869/1869 - 0s - loss: 0.9813 - acc: 0.5869
Epoch 26/60
1869/1869 - 0s - loss: 0.9824 - acc: 0.5773
Epoch 27/60
1869/1869 - 0s - loss: 0.9823 - acc: 0.5778
Epoch 28/60
1869/1869 - 0s - loss: 0.9836 - acc: 0.5805
Epoch 29/60
1869/1869 - 0s - loss: 0.9806 - acc: 0.5864
Epoch 30/60
1869/1869 - 0s - loss: 0.9845 - acc: 0.5869
Epoch 31/60
1869/1869 - 0s - loss: 0.9832 - acc: 0.5805
Epoch 32/60
1869/1869 - 0s - loss: 0.9831 - acc: 0.5821
Epoch 33/60
1869/1869 - 0s - loss: 0.9776 - acc: 0.5800
Epoch 34/60
1869/1869 - 0s - loss: 0.9762 - acc: 0.5811
Epoch 35/60
1869/1869 - 0s - loss: 0.9757 - acc: 0.5784
Epoch 36/60
1869/1869 - 0s - loss: 0.9762 - acc: 0.5891
Epoch 37/60
1869/1869 - 0s - loss: 0.9735 - acc: 0.6003
Epoch 38/60
1869/1869 - 0s - loss: 0.9815 - acc: 0.5746
Epoch 39/60
1869/1869 - 0s - loss: 0.9726 - acc: 0.5859
Epoch 40/60
1869/1869 - 0s - loss: 0.9786 - acc: 0.5827
Epoch 41/60
1869/1869 - 0s - loss: 0.9718 - acc: 0.5875
Epoch 42/60
1869/1869 - 0s - loss: 0.9781 - acc: 0.5848
Epoch 43/60
1869/1869 - 0s - loss: 0.9759 - acc: 0.5811
Epoch 44/60
1869/1869 - 0s - loss: 0.9715 - acc: 0.5837
Epoch 45/60
1869/1869 - 0s - loss: 0.9693 - acc: 0.5859
Epoch 46/60
1869/1869 - 0s - loss: 0.9723 - acc: 0.5789
Epoch 47/60
1869/1869 - 0s - loss: 0.9726 - acc: 0.5837
Epoch 48/60
1869/1869 - 0s - loss: 0.9710 - acc: 0.5902
Epoch 49/60
1869/1869 - 0s - loss: 0.9696 - acc: 0.5891
Epoch 50/60
1869/1869 - 0s - loss: 0.9707 - acc: 0.5843
Epoch 51/60
1869/1869 - 0s - loss: 0.9706 - acc: 0.5896

```
Epoch 52/60
1869/1869 - 0s - loss: 0.9691 - acc: 0.5843
Epoch 53/60
1869/1869 - 0s - loss: 0.9662 - acc: 0.5891
Epoch 54/60
1869/1869 - 0s - loss: 0.9680 - acc: 0.5902
Epoch 55/60
1869/1869 - 0s - loss: 0.9678 - acc: 0.5821
Epoch 56/60
1869/1869 - 0s - loss: 0.9724 - acc: 0.5827
Epoch 57/60
1869/1869 - 0s - loss: 0.9636 - acc: 0.5886
Epoch 58/60
1869/1869 - 0s - loss: 0.9616 - acc: 0.5923
Epoch 59/60
1869/1869 - 0s - loss: 0.9652 - acc: 0.5832
Epoch 60/60
1869/1869 - 0s - loss: 0.9632 - acc: 0.5869
```

Out[15]: <tensorflow.python.keras.callbacks.History at 0x183c7df5710>

Quantify our Trained Model

```
In [16]: model_loss, model_accuracy = model.evaluate(
        X_test_scaled, y_test_categorical, verbose=2)
print(
    f"Normal Neural Network - Loss: {model_loss}, Accuracy: {model_accuracy}")

623/623 - 0s - loss: 0.9350 - acc: 0.6067
Normal Neural Network - Loss: 0.9350250822583133, Accuracy: 0.6067415475845337
```

Make Predictions

```
In [17]: encoded_predictions = model.predict_classes(X_test_scaled[:40])
        prediction_labels = label_encoder.inverse_transform(encoded_predictions)
```

```
In [18]: print(f"Predicted classes: {prediction_labels}")
        print(f"Actual Labels: {list(y_test[:40])}")
```

```
Predicted classes: [3 3 3 3 2 2 2 3 4 2 2 3 3 1 3 3 2 3 1 3 2 3 3 4 2 2 3 3 2 2 3 2 1 3 3 3 4
 4 3 2]
```

```
Actual Labels: [3, 3, 3, 3, 2, 1, 2, 3, 4, 2, 3, 2, 3, 1, 3, 3, 2, 3, 1, 3, 2, 3, 3, 4, 2, 2, 3, 4, 2, 2, 3, 3, 1, 4, 3, 2, 3, 4, 3, 3]
```

In []: