

In [1]:

```
# Import Libraries
import pandas as pd
import numpy as np
import tensorflow as tf
from tensorflow.keras import layers
from sklearn.model_selection import train_test_split
import seaborn as sns
from matplotlib import pyplot as plt
```

In [2]:

```
# Load the TensorBoard notebook extension.
%load_ext tensorboard
from datetime import datetime
from packaging import version

print("TensorFlow version: ", tf.__version__)
assert version.parse(tf.__version__).release[0] >= 2, \
    "This notebook requires TensorFlow 2.0 or above."
import tensorboard
tensorboard.__version__
```

TensorFlow version: 2.6.0

Out[2]:

'2.6.0'

In [3]:

```
rooms = {
    "Brayden's Room": 0,
    "Bedroom": 1,
    "Bathroom 1": 2,
    "Kitchen": 3,
    "Dining Room": 4,
    "Living Room": 5,
    "Extra Room": 6
}

rooms_labels = dict()
for key, value in rooms.items():
    rooms_labels[value] = key
```

Inport and clean data

In [4]:

```
ble = pd.read_csv("data/ble_data-3.csv", delimiter=";", quoting = 3)
```

In [5]:

```

ble = ble.fillna(-200)

replacements = dict()
for column in ble.columns:
    replacements[column] = column.replace(':', '-')

    if column != "room":
        ble[column] = pd.to_numeric(ble[column])

ble.rename(columns=replacements, inplace=True)
ble_shuffle = ble.sample(frac=1).reset_index(drop=True)

# Normalize and drop unused columns
drop = []
for column in ble_shuffle.columns:
    if column == "room":
        continue
    if ble_shuffle[column].max() == ble_shuffle[column].min():
        drop.append(column)

ble_cleaned = ble_shuffle.drop(columns=drop)

ble_normalized = ble_cleaned.copy()

# apply normalization techniques
for column in ble_normalized.columns:
    if column == "room":
        ble_normalized[column] = ble_normalized[column]
    else:
        ble_normalized[column] = (ble_normalized[column] - ble_normalized[col

```

In [6]:

```

train_df, test_df = train_test_split(ble_normalized, test_size=0.1)
train_df, val_df = train_test_split(train_df, test_size=0.2)

print(len(train_df), 'train examples')
print(len(val_df), 'validation examples')
print(len(test_df), 'test examples')

```

```

13687 train examples
3422 validation examples
1901 test examples

```

Prepare Model

In [7]:

```

# Create an empty list
feature_cols = []

# Numeric Columns
for header in ble_normalized.columns:
    if header == "room":
        continue

    feature_cols.append(tf.feature_column.numeric_column(header))

print("Feature columns: ", feature_cols, "\n")

feature_layer = tf.keras.layers.DenseFeatures(feature_cols, name='Features')

```

```

Feature columns: [NumericColumn(key='50-de-06-6c-21-08', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='e4-f8-9d-6c-92-2c', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='4d-74-04-a1-d0-66', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='f5-3a-4f-36-13-5f', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='28-11-a5-d6-8b-07', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='77-e2-15-65-85-90', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='d2-f5-db-e4-dc-65', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='12-4e-8f-cf-e8-de', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='57-31-a2-ac-c3-16', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='18-47-3d-38-71-8c', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='dd-da-dc-5f-2a-95', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='64-07-f6-7f-10-cf', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='f3-33-b2-94-89-26', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='30-24-a9-da-84-7a', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='65-38-8a-a5-19-6b', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='cb-4e-fd-a1-6c-35', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='51-5b-0e-35-01-15', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='c0-28-8d-39-44-41', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='50-a9-75-7b-ee-d6', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='bc-14-85-f8-84-91', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='f0-ef-86-f1-cf-22', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='55-3f-98-28-5a-d7', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='db-22-2b-21-04-47', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='64-07-f6-52-a5-e9', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='50-de-06-70-b0-f1', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='40-03-a1-1f-26-ad', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='c0-d3-91-e5-c6-21', shape=(1,), default_value=None, dtype=tf.float

```

```
32, normalizer_fn=None), NumericColumn(key='c8-ff-77-c9-54-f7', shape=(1,), de
fault_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='a4
-c1-38-e3-ce-d8', shape=(1,), default_value=None, dtype=tf.float32, normalizer
_fn=None), NumericColumn(key='64-07-f6-5a-99-bf', shape=(1,), default_value=No
ne, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='fb-56-df-d1-48-3
a', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), Num
ericColumn(key='28-6b-e3-cd-ee-5f', shape=(1,), default_value=None, dtype=tf.f
loat32, normalizer_fn=None), NumericColumn(key='62-c8-31-71-6f-46', shape=(1,)
, default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key
='73-3a-fd-91-e2-fd', shape=(1,), default_value=None, dtype=tf.float32, normal
izer_fn=None), NumericColumn(key='08-66-98-cf-c3-28', shape=(1,), default_valu
e=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='7f-6a-aa-9f-
c6-b0', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None),
NumericColumn(key='1d-91-1d-19-15-30', shape=(1,), default_value=None, dtype=t
f.float32, normalizer_fn=None), NumericColumn(key='2c-41-a1-7e-7c-60', shape=(
1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(
key='f0-b3-ec-34-14-2c', shape=(1,), default_value=None, dtype=tf.float32, nor
malizer_fn=None), NumericColumn(key='68-64-4b-24-37-1d', shape=(1,), default_v
alue=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='40-06-f6-
1a-6e-83', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=Non
e), NumericColumn(key='39-b2-6d-1d-c8-b9', shape=(1,), default_value=None, dty
pe=tf.float32, normalizer_fn=None), NumericColumn(key='08-66-98-bc-f7-48', sha
pe=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericCol
umn(key='e7-e7-a6-0d-97-ab', shape=(1,), default_value=None, dtype=tf.float32,
normalizer_fn=None), NumericColumn(key='39-2b-70-81-47-5d', shape=(1,), default
_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='d2-10-
29-32-35-43', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=
None), NumericColumn(key='05-8d-16-44-19-69', shape=(1,), default_value=None,
dtype=tf.float32, normalizer_fn=None), NumericColumn(key='66-14-58-7c-19-c7',
shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), Numeric
Column(key='0c-75-76-ab-d5-40', shape=(1,), default_value=None, dtype=tf.float
32, normalizer_fn=None), NumericColumn(key='58-8e-81-eb-19-0c', shape=(1,), de
fault_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='73
-3e-8d-15-10-18', shape=(1,), default_value=None, dtype=tf.float32, normalizer
_fn=None), NumericColumn(key='d8-a7-56-df-44-73', shape=(1,), default_value=No
ne, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='68-85-19-55-e5-e
4', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), Num
ericColumn(key='d8-0f-99-78-96-ae', shape=(1,), default_value=None, dtype=tf.f
loat32, normalizer_fn=None), NumericColumn(key='9c-20-7b-85-4c-d8', shape=(1,)
, default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key
='e4-6a-ae-ee-21-85', shape=(1,), default_value=None, dtype=tf.float32, normal
izer_fn=None), NumericColumn(key='ec-81-93-6b-87-8c', shape=(1,), default_valu
e=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='5e-82-77-dc-
6d-48', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None),
NumericColumn(key='52-c2-a2-98-da-d8', shape=(1,), default_value=None, dtype=t
f.float32, normalizer_fn=None), NumericColumn(key='2f-b0-45-ca-d8-aa', shape=(
1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(
key='c0-a9-d2-4f-d8-ae', shape=(1,), default_value=None, dtype=tf.float32, nor
malizer_fn=None), NumericColumn(key='8c-79-f5-e2-22-0f', shape=(1,), default_v
alue=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='f0-b3-ec-
33-c3-ee', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=Non
e), NumericColumn(key='3c-9b-50-54-0a-b7', shape=(1,), default_value=None, dty
pe=tf.float32, normalizer_fn=None), NumericColumn(key='04-2a-9c-4b-cb-a2', sha
```

```
pe=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='28-11-a5-d6-a2-f4', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='50-de-06-9e-2d-c5', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='f9-08-04-b3-5c-9f', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='dd-27-7d-b2-e8-bd', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None), NumericColumn(key='56-a5-db-f0-ea-4e', shape=(1,), default_value=None, dtype=tf.float32, normalizer_fn=None)]
```

In [8]:

```
# Hyperparameters
epochs = 100

# Label
label_name = "room"
shuffle = True

#---Create a sequential model---#
model = tf.keras.Sequential([
    feature_layer,
    # tf.keras.layers.Dense(256 ,activation='relu', name='hidden_layer'),
    tf.keras.layers.Dense(60 ,activation='relu', name='hidden_layer2'),
    tf.keras.layers.Dense(20 ,activation='relu', name='hidden_layer3'),
    tf.keras.layers.Dense(len(rooms), name='output')
])

model.compile(optimizer=tf.keras.optimizers.Adam(),
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

#---Train the Model---#
# Keras TensorBoard callback.
logdir = "logs/fit/" + datetime.now().strftime("%Y%m%d-%H%M%S")
tensorboard_callback = tf.keras.callbacks.TensorBoard(log_dir=logdir)

# Split the datasets into features and label.
train_lbl = np.array([rooms[x] for x in train_df[label_name]])
train_df_ = train_df.drop(columns=[label_name])
train_ft = {name:np.array(value) for name, value in train_df_.items()}

val_lbl = np.array([rooms[x] for x in val_df[label_name]])
val_df_ = val_df.drop(columns=[label_name])
val_ft = {name:np.array(value) for name, value in val_df_.items()}

model.fit(train_ft, train_lbl, epochs=epochs,
          validation_data=(val_ft, val_lbl))
```

2021-11-20 12:06:10 008674: I tensorflow/core/platform/cpu_feature_guard.cc:14

```
2021-11-20 12:06:10.000074: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX2 FMA
```

To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.

```
2021-11-20 12:06:10.025708: I tensorflow/core/profiler/lib/profiler_session.cc:131] Profiler session initializing.
```

```
2021-11-20 12:06:10.025725: I tensorflow/core/profiler/lib/profiler_session.cc:146] Profiler session started.
```

```
2021-11-20 12:06:10.025853: I tensorflow/core/profiler/lib/profiler_session.cc:164] Profiler session tear down.
```

```
2021-11-20 12:06:10.191623: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR Optimization Passes are enabled (registered 2)
Epoch 1/100
```

WARNING:tensorflow:Layers in a Sequential model should only have a single input tensor, but we receive a <class 'dict'> input: {'50-de-06-6c-21-08': <tf.Tensor 'ExpandDims_21:0' shape=(None, 1) dtype=float32>, 'e4-f8-9d-6c-92-2c': <tf.Tensor 'ExpandDims_60:0' shape=(None, 1) dtype=float32>, '4d-74-04-a1-d0-66': <tf.Tensor 'ExpandDims_19:0' shape=(None, 1) dtype=float32>, 'f5-3a-4f-36-13-5f': <tf.Tensor 'ExpandDims_67:0' shape=(None, 1) dtype=float32>, '28-11-a5-d6-8b-07': <tf.Tensor 'ExpandDims_8:0' shape=(None, 1) dtype=float32>, '77-e2-15-65-85-90': <tf.Tensor 'ExpandDims_41:0' shape=(None, 1) dtype=float32>, 'd2-f5-db-e4-dc-65': <tf.Tensor 'ExpandDims_53:0' shape=(None, 1) dtype=float32>, '12-4e-8f-cf-e8-de': <tf.Tensor 'ExpandDims_5:0' shape=(None, 1) dtype=float32>, '57-31-a2-ac-c3-16': <tf.Tensor 'ExpandDims_28:0' shape=(None, 1) dtype=float32>, '18-47-3d-38-71-8c': <tf.Tensor 'ExpandDims_6:0' shape=(None, 1) dtype=float32>, 'dd-da-dc-5f-2a-95': <tf.Tensor 'ExpandDims_58:0' shape=(None, 1) dtype=float32>, '64-07-f6-7f-10-cf': <tf.Tensor 'ExpandDims_34:0' shape=(None, 1) dtype=float32>, 'f3-33-b2-94-89-26': <tf.Tensor 'ExpandDims_66:0' shape=(None, 1) dtype=float32>, '30-24-a9-da-84-7a': <tf.Tensor 'ExpandDims_13:0' shape=(None, 1) dtype=float32>, '65-38-8a-a5-19-6b': <tf.Tensor 'ExpandDims_35:0' shape=(None, 1) dtype=float32>, 'cb-4e-fd-a1-6c-35': <tf.Tensor 'ExpandDims_51:0' shape=(None, 1) dtype=float32>, '51-5b-0e-35-01-15': <tf.Tensor 'ExpandDims_24:0' shape=(None, 1) dtype=float32>, 'c0-28-8d-39-44-41': <tf.Tensor 'ExpandDims_47:0' shape=(None, 1) dtype=float32>, '50-a9-75-7b-ee-d6': <tf.Tensor 'ExpandDims_20:0' shape=(None, 1) dtype=float32>, 'bc-14-85-f8-84-91': <tf.Tensor 'ExpandDims_46:0' shape=(None, 1) dtype=float32>, 'f0-ef-86-f1-cf-22': <tf.Tensor 'ExpandDims_65:0' shape=(None, 1) dtype=float32>, '55-3f-98-28-5a-d7': <tf.Tensor 'ExpandDims_26:0' shape=(None, 1) dtype=float32>, 'db-22-2b-21-04-47': <tf.Tensor 'ExpandDims_56:0' shape=(None, 1) dtype=float32>, '64-07-f6-52-a5-e9': <tf.Tensor 'ExpandDims_32:0' shape=(None, 1) dtype=float32>, '50-de-06-70-b0-f1': <tf.Tensor 'ExpandDims_22:0' shape=(None, 1) dtype=float32>, '40-03-a1-1f-26-ad': <tf.Tensor 'ExpandDims_17:0' shape=(None, 1) dtype=float32>, 'c0-d3-91-e5-c6-21': <tf.Tensor 'ExpandDims_49:0' shape=(None, 1) dtype=float32>, 'c8-f-f77-c9-54-f7': <tf.Tensor 'ExpandDims_50:0' shape=(None, 1) dtype=float32>, 'a4-c1-38-e3-ce-d8': <tf.Tensor 'ExpandDims_45:0' shape=(None, 1) dtype=float32>, '64-07-f6-5a-99-bf': <tf.Tensor 'ExpandDims_33:0' shape=(None, 1) dtype=float32>, 'fb-56-df-d1-48-3a': <tf.Tensor 'ExpandDims_69:0' shape=(None, 1) dtype=float32>, '28-6b-e3-cd-ee-5f': <tf.Tensor 'ExpandDims_10:0' shape=(None, 1) dtype=float32>, '62-c8-31-71-6f-46': <tf.Tensor 'ExpandDims_31:0' shape=(None, 1) dtype=float32>, '73-3a-fd-91-e2-fd': <tf.Tensor 'ExpandDims_39:0' shape=(None, 1) dtype=float32>, '08-66-98-cf-c3-28': <tf.Tensor 'ExpandDims_3:0' shape=(None, 1) dtype=float32>, '7f-6a-aa-9f-c6-b0': <tf.Tensor 'ExpandDims_42:0' sh

```
ape=(None, 1) dtype=float32>, '1d-91-1d-19-15-30': <tf.Tensor 'ExpandDims_7:0'
shape=(None, 1) dtype=float32>, '2c-41-a1-7e-7c-60': <tf.Tensor 'ExpandDims_11
:0' shape=(None, 1) dtype=float32>, 'f0-b3-ec-34-14-2c': <tf.Tensor 'ExpandDim
s_64:0' shape=(None, 1) dtype=float32>, '68-64-4b-24-37-1d': <tf.Tensor 'Expan
dDims_37:0' shape=(None, 1) dtype=float32>, '40-06-f6-1a-6e-83': <tf.Tensor 'E
xpandDims_18:0' shape=(None, 1) dtype=float32>, '39-b2-6d-1d-c8-b9': <tf.Tenso
r 'ExpandDims_15:0' shape=(None, 1) dtype=float32>, '08-66-98-bc-f7-48': <tf.T
ensor 'ExpandDims_2:0' shape=(None, 1) dtype=float32>, 'e7-e7-a6-0d-97-ab': <t
f.Tensor 'ExpandDims_61:0' shape=(None, 1) dtype=float32>, '39-2b-70-81-47-5d'
: <tf.Tensor 'ExpandDims_14:0' shape=(None, 1) dtype=float32>, 'd2-10-29-32-35
-43': <tf.Tensor 'ExpandDims_52:0' shape=(None, 1) dtype=float32>, '05-8d-16-4
4-19-69': <tf.Tensor 'ExpandDims_1:0' shape=(None, 1) dtype=float32>, '66-14-5
8-7c-19-c7': <tf.Tensor 'ExpandDims_36:0' shape=(None, 1) dtype=float32>, '0c-
75-76-ab-d5-40': <tf.Tensor 'ExpandDims_4:0' shape=(None, 1) dtype=float32>, '
58-8e-81-eb-19-0c': <tf.Tensor 'ExpandDims_29:0' shape=(None, 1) dtype=float32
>, '73-3e-8d-15-10-18': <tf.Tensor 'ExpandDims_40:0' shape=(None, 1) dtype=flo
at32>, 'd8-a7-56-df-44-73': <tf.Tensor 'ExpandDims_55:0' shape=(None, 1) dtype
=float32>, '68-85-19-55-e5-e4': <tf.Tensor 'ExpandDims_38:0' shape=(None, 1) d
type=float32>, 'd8-0f-99-78-96-ae': <tf.Tensor 'ExpandDims_54:0' shape=(None,
1) dtype=float32>, '9c-20-7b-85-4c-d8': <tf.Tensor 'ExpandDims_44:0' shape=(No
ne, 1) dtype=float32>, 'e4-6a-ae-ee-21-85': <tf.Tensor 'ExpandDims_59:0' shape
=(None, 1) dtype=float32>, 'ec-81-93-6b-87-8c': <tf.Tensor 'ExpandDims_62:0' s
hape=(None, 1) dtype=float32>, '5e-82-77-dc-6d-48': <tf.Tensor 'ExpandDims_30:
0' shape=(None, 1) dtype=float32>, '52-c2-a2-98-da-d8': <tf.Tensor 'ExpandDims
_25:0' shape=(None, 1) dtype=float32>, '2f-b0-45-ca-d8-aa': <tf.Tensor 'Expand
Dims_12:0' shape=(None, 1) dtype=float32>, 'c0-a9-d2-4f-d8-ae': <tf.Tensor 'Ex
pandDims_48:0' shape=(None, 1) dtype=float32>, '8c-79-f5-e2-22-0f': <tf.Tensor
'ExpandDims_43:0' shape=(None, 1) dtype=float32>, 'f0-b3-ec-33-c3-ee': <tf.Ten
sor 'ExpandDims_63:0' shape=(None, 1) dtype=float32>, '3c-9b-50-54-0a-b7': <tf
.Tensor 'ExpandDims_16:0' shape=(None, 1) dtype=float32>, '04-2a-9c-4b-cb-a2':
<tf.Tensor 'ExpandDims:0' shape=(None, 1) dtype=float32>, '28-11-a5-d6-a2-f4':
<tf.Tensor 'ExpandDims_9:0' shape=(None, 1) dtype=float32>, '50-de-06-9e-2d-c5
': <tf.Tensor 'ExpandDims_23:0' shape=(None, 1) dtype=float32>, 'f9-08-04-b3-5
c-9f': <tf.Tensor 'ExpandDims_68:0' shape=(None, 1) dtype=float32>, 'dd-27-7d-
b2-e8-bd': <tf.Tensor 'ExpandDims_57:0' shape=(None, 1) dtype=float32>, '56-a5
-db-f0-ea-4e': <tf.Tensor 'ExpandDims_27:0' shape=(None, 1) dtype=float32>}
```

Consider rewriting this model with the Functional API.

```
WARNING:tensorflow:Layers in a Sequential model should only have a single input
tensor, but we receive a <class 'dict'> input: {'50-de-06-6c-21-08': <tf.Ten
sor 'ExpandDims_21:0' shape=(None, 1) dtype=float32>, 'e4-f8-9d-6c-92-2c': <t
f.Tensor 'ExpandDims_60:0' shape=(None, 1) dtype=float32>, '4d-74-04-a1-d0-66':
<tf.Tensor 'ExpandDims_19:0' shape=(None, 1) dtype=float32>, 'f5-3a-4f-36-13-5
f': <tf.Tensor 'ExpandDims_67:0' shape=(None, 1) dtype=float32>, '28-11-a5-d6-
8b-07': <tf.Tensor 'ExpandDims_8:0' shape=(None, 1) dtype=float32>, '77-e2-15-
65-85-90': <tf.Tensor 'ExpandDims_41:0' shape=(None, 1) dtype=float32>, 'd2-f5
-db-e4-dc-65': <tf.Tensor 'ExpandDims_53:0' shape=(None, 1) dtype=float32>, '1
2-4e-8f-cf-e8-de': <tf.Tensor 'ExpandDims_5:0' shape=(None, 1) dtype=float32>,
'57-31-a2-ac-c3-16': <tf.Tensor 'ExpandDims_28:0' shape=(None, 1) dtype=float3
2>, '18-47-3d-38-71-8c': <tf.Tensor 'ExpandDims_6:0' shape=(None, 1) dtype=flo
at32>, 'dd-da-dc-5f-2a-95': <tf.Tensor 'ExpandDims_58:0' shape=(None, 1) dtype
=float32>, '64-07-f6-7f-10-cf': <tf.Tensor 'ExpandDims_34:0' shape=(None, 1) d
type=float32>, 'f3-33-b2-94-89-26': <tf.Tensor 'ExpandDims_66:0' shape=(None,
1) dtype=float32>, '30-24-a9-da-84-7a': <tf.Tensor 'ExpandDims_13:0' shape=(No
```

```

ne, 1) dtype=float32>, '65-38-8a-a5-19-6b': <tf.Tensor 'ExpandDims_35:0' shape
=(None, 1) dtype=float32>, 'cb-4e-fd-a1-6c-35': <tf.Tensor 'ExpandDims_51:0' s
hape=(None, 1) dtype=float32>, '51-5b-0e-35-01-15': <tf.Tensor 'ExpandDims_24:
0' shape=(None, 1) dtype=float32>, 'c0-28-8d-39-44-41': <tf.Tensor 'ExpandDims
_47:0' shape=(None, 1) dtype=float32>, '50-a9-75-7b-ee-d6': <tf.Tensor 'Expand
Dims_20:0' shape=(None, 1) dtype=float32>, 'bc-14-85-f8-84-91': <tf.Tensor 'Ex
pandDims_46:0' shape=(None, 1) dtype=float32>, 'f0-ef-86-f1-cf-22': <tf.Tensor
'ExpandDims_65:0' shape=(None, 1) dtype=float32>, '55-3f-98-28-5a-d7': <tf.Ten
sor 'ExpandDims_26:0' shape=(None, 1) dtype=float32>, 'db-22-2b-21-04-47': <tf
.Tensor 'ExpandDims_56:0' shape=(None, 1) dtype=float32>, '64-07-f6-52-a5-e9':
<tf.Tensor 'ExpandDims_32:0' shape=(None, 1) dtype=float32>, '50-de-06-70-b0-f
1': <tf.Tensor 'ExpandDims_22:0' shape=(None, 1) dtype=float32>, '40-03-a1-1f-
26-ad': <tf.Tensor 'ExpandDims_17:0' shape=(None, 1) dtype=float32>, 'c0-d3-91
-e5-c6-21': <tf.Tensor 'ExpandDims_49:0' shape=(None, 1) dtype=float32>, 'c8-f
f-77-c9-54-f7': <tf.Tensor 'ExpandDims_50:0' shape=(None, 1) dtype=float32>, '
a4-c1-38-e3-ce-d8': <tf.Tensor 'ExpandDims_45:0' shape=(None, 1) dtype=float32
>, '64-07-f6-5a-99-bf': <tf.Tensor 'ExpandDims_33:0' shape=(None, 1) dtype=flo
at32>, 'fb-56-df-d1-48-3a': <tf.Tensor 'ExpandDims_69:0' shape=(None, 1) dtype
=float32>, '28-6b-e3-cd-ee-5f': <tf.Tensor 'ExpandDims_10:0' shape=(None, 1) d
type=float32>, '62-c8-31-71-6f-46': <tf.Tensor 'ExpandDims_31:0' shape=(None,
1) dtype=float32>, '73-3a-fd-91-e2-fd': <tf.Tensor 'ExpandDims_39:0' shape=(No
ne, 1) dtype=float32>, '08-66-98-cf-c3-28': <tf.Tensor 'ExpandDims_3:0' shape=
(None, 1) dtype=float32>, '7f-6a-aa-9f-c6-b0': <tf.Tensor 'ExpandDims_42:0' sh
ape=(None, 1) dtype=float32>, '1d-91-1d-19-15-30': <tf.Tensor 'ExpandDims_7:0'
shape=(None, 1) dtype=float32>, '2c-41-a1-7e-7c-60': <tf.Tensor 'ExpandDims_11
:0' shape=(None, 1) dtype=float32>, 'f0-b3-ec-34-14-2c': <tf.Tensor 'ExpandDim
s_64:0' shape=(None, 1) dtype=float32>, '68-64-4b-24-37-1d': <tf.Tensor 'Expan
dDims_37:0' shape=(None, 1) dtype=float32>, '40-06-f6-1a-6e-83': <tf.Tensor 'E
xpandDims_18:0' shape=(None, 1) dtype=float32>, '39-b2-6d-1d-c8-b9': <tf.Tenso
r 'ExpandDims_15:0' shape=(None, 1) dtype=float32>, '08-66-98-bc-f7-48': <tf.T
ensor 'ExpandDims_2:0' shape=(None, 1) dtype=float32>, 'e7-e7-a6-0d-97-ab': <t
f.Tensor 'ExpandDims_61:0' shape=(None, 1) dtype=float32>, '39-2b-70-81-47-5d'
: <tf.Tensor 'ExpandDims_14:0' shape=(None, 1) dtype=float32>, 'd2-10-29-32-35
-43': <tf.Tensor 'ExpandDims_52:0' shape=(None, 1) dtype=float32>, '05-8d-16-4
4-19-69': <tf.Tensor 'ExpandDims_1:0' shape=(None, 1) dtype=float32>, '66-14-5
8-7c-19-c7': <tf.Tensor 'ExpandDims_36:0' shape=(None, 1) dtype=float32>, '0c-
75-76-ab-d5-40': <tf.Tensor 'ExpandDims_4:0' shape=(None, 1) dtype=float32>, '
58-8e-81-eb-19-0c': <tf.Tensor 'ExpandDims_29:0' shape=(None, 1) dtype=float32
>, '73-3e-8d-15-10-18': <tf.Tensor 'ExpandDims_40:0' shape=(None, 1) dtype=flo
at32>, 'd8-a7-56-df-44-73': <tf.Tensor 'ExpandDims_55:0' shape=(None, 1) dtype
=float32>, '68-85-19-55-e5-e4': <tf.Tensor 'ExpandDims_38:0' shape=(None, 1) d
type=float32>, 'd8-0f-99-78-96-ae': <tf.Tensor 'ExpandDims_54:0' shape=(None,
1) dtype=float32>, '9c-20-7b-85-4c-d8': <tf.Tensor 'ExpandDims_44:0' shape=(No
ne, 1) dtype=float32>, 'e4-6a-ae-ee-21-85': <tf.Tensor 'ExpandDims_59:0' shape
=(None, 1) dtype=float32>, 'ec-81-93-6b-87-8c': <tf.Tensor 'ExpandDims_62:0' s
hape=(None, 1) dtype=float32>, '5e-82-77-dc-6d-48': <tf.Tensor 'ExpandDims_30:
0' shape=(None, 1) dtype=float32>, '52-c2-a2-98-da-d8': <tf.Tensor 'ExpandDims
_25:0' shape=(None, 1) dtype=float32>, '2f-b0-45-ca-d8-aa': <tf.Tensor 'Expand
Dims_12:0' shape=(None, 1) dtype=float32>, 'c0-a9-d2-4f-d8-ae': <tf.Tensor 'Ex
pandDims_48:0' shape=(None, 1) dtype=float32>, '8c-79-f5-e2-22-0f': <tf.Tensor
'ExpandDims_43:0' shape=(None, 1) dtype=float32>, 'f0-b3-ec-33-c3-ee': <tf.Ten
sor 'ExpandDims_63:0' shape=(None, 1) dtype=float32>, '3c-9b-50-54-0a-b7': <t
f.Tensor 'ExpandDims_16:0' shape=(None, 1) dtype=float32>, '04-2a-9c-4b-cb-a2':

```



```
<tf.Tensor 'ExpandDims:0' shape=(None, 1) dtype=float32>, '28-11-a5-d6-a2-f4':
<tf.Tensor 'ExpandDims_9:0' shape=(None, 1) dtype=float32>, '50-de-06-9e-2d-c5
': <tf.Tensor 'ExpandDims_23:0' shape=(None, 1) dtype=float32>, 'f9-08-04-b3-5
c-9f': <tf.Tensor 'ExpandDims_68:0' shape=(None, 1) dtype=float32>, 'dd-27-7d-
b2-e8-bd': <tf.Tensor 'ExpandDims_57:0' shape=(None, 1) dtype=float32>, '56-a5
-db-f0-ea-4e': <tf.Tensor 'ExpandDims_27:0' shape=(None, 1) dtype=float32>}
```

Consider rewriting this model with the Functional API.

```
408/428 [=====>..] - ETA: 0s - loss: 1.0751 - accuracy:
0.6133WARNING:tensorflow:Layers in a Sequential model should only have a singl
e input tensor, but we receive a <class 'dict'> input: {'50-de-06-6c-21-08': <
tf.Tensor 'ExpandDims_21:0' shape=(None, 1) dtype=float32>, 'e4-f8-9d-6c-92-2c
': <tf.Tensor 'ExpandDims_60:0' shape=(None, 1) dtype=float32>, '4d-74-04-a1-d
0-66': <tf.Tensor 'ExpandDims_19:0' shape=(None, 1) dtype=float32>, 'f5-3a-4f-
36-13-5f': <tf.Tensor 'ExpandDims_67:0' shape=(None, 1) dtype=float32>, '28-11
-a5-d6-8b-07': <tf.Tensor 'ExpandDims_8:0' shape=(None, 1) dtype=float32>, '77
-e2-15-65-85-90': <tf.Tensor 'ExpandDims_41:0' shape=(None, 1) dtype=float32>,
'd2-f5-db-e4-dc-65': <tf.Tensor 'ExpandDims_53:0' shape=(None, 1) dtype=float3
2>, '12-4e-8f-cf-e8-de': <tf.Tensor 'ExpandDims_5:0' shape=(None, 1) dtype=flo
at32>, '57-31-a2-ac-c3-16': <tf.Tensor 'ExpandDims_28:0' shape=(None, 1) dtype
=float32>, '18-47-3d-38-71-8c': <tf.Tensor 'ExpandDims_6:0' shape=(None, 1) dt
ype=float32>, 'dd-da-dc-5f-2a-95': <tf.Tensor 'ExpandDims_58:0' shape=(None, 1
) dtype=float32>, '64-07-f6-7f-10-cf': <tf.Tensor 'ExpandDims_34:0' shape=(Non
e, 1) dtype=float32>, 'f3-33-b2-94-89-26': <tf.Tensor 'ExpandDims_66:0' shape=
(None, 1) dtype=float32>, '30-24-a9-da-84-7a': <tf.Tensor 'ExpandDims_13:0' sh
ape=(None, 1) dtype=float32>, '65-38-8a-a5-19-6b': <tf.Tensor 'ExpandDims_35:0
' shape=(None, 1) dtype=float32>, 'cb-4e-fd-a1-6c-35': <tf.Tensor 'ExpandDims_
51:0' shape=(None, 1) dtype=float32>, '51-5b-0e-35-01-15': <tf.Tensor 'ExpandD
ims_24:0' shape=(None, 1) dtype=float32>, 'c0-28-8d-39-44-41': <tf.Tensor 'Exp
andDims_47:0' shape=(None, 1) dtype=float32>, '50-a9-75-7b-ee-d6': <tf.Tensor
'ExpandDims_20:0' shape=(None, 1) dtype=float32>, 'bc-14-85-f8-84-91': <tf.Ten
sor 'ExpandDims_46:0' shape=(None, 1) dtype=float32>, 'f0-ef-86-f1-cf-22': <tf
.Tensor 'ExpandDims_65:0' shape=(None, 1) dtype=float32>, '55-3f-98-28-5a-d7':
<tf.Tensor 'ExpandDims_26:0' shape=(None, 1) dtype=float32>, 'db-22-2b-21-04-4
7': <tf.Tensor 'ExpandDims_56:0' shape=(None, 1) dtype=float32>, '64-07-f6-52-
a5-e9': <tf.Tensor 'ExpandDims_32:0' shape=(None, 1) dtype=float32>, '50-de-06
-70-b0-f1': <tf.Tensor 'ExpandDims_22:0' shape=(None, 1) dtype=float32>, '40-0
3-a1-1f-26-ad': <tf.Tensor 'ExpandDims_17:0' shape=(None, 1) dtype=float32>, '
c0-d3-91-e5-c6-21': <tf.Tensor 'ExpandDims_49:0' shape=(None, 1) dtype=float32
>, 'c8-ff-77-c9-54-f7': <tf.Tensor 'ExpandDims_50:0' shape=(None, 1) dtype=flo
at32>, 'a4-c1-38-e3-ce-d8': <tf.Tensor 'ExpandDims_45:0' shape=(None, 1) dtype
=float32>, '64-07-f6-5a-99-bf': <tf.Tensor 'ExpandDims_33:0' shape=(None, 1) d
type=float32>, 'fb-56-df-d1-48-3a': <tf.Tensor 'ExpandDims_69:0' shape=(None,
1) dtype=float32>, '28-6b-e3-cd-ee-5f': <tf.Tensor 'ExpandDims_10:0' shape=(No
ne, 1) dtype=float32>, '62-c8-31-71-6f-46': <tf.Tensor 'ExpandDims_31:0' shape
=(None, 1) dtype=float32>, '73-3a-fd-91-e2-fd': <tf.Tensor 'ExpandDims_39:0' s
hape=(None, 1) dtype=float32>, '08-66-98-cf-c3-28': <tf.Tensor 'ExpandDims_3:0
' shape=(None, 1) dtype=float32>, '7f-6a-aa-9f-c6-b0': <tf.Tensor 'ExpandDims_
42:0' shape=(None, 1) dtype=float32>, '1d-91-1d-19-15-30': <tf.Tensor 'ExpandD
ims_7:0' shape=(None, 1) dtype=float32>, '2c-41-a1-7e-7c-60': <tf.Tensor 'Expa
ndDims_11:0' shape=(None, 1) dtype=float32>, 'f0-b3-ec-34-14-2c': <tf.Tensor '
ExpandDims_64:0' shape=(None, 1) dtype=float32>, '68-64-4b-24-37-1d': <tf.Tens
or 'ExpandDims_37:0' shape=(None, 1) dtype=float32>, '40-06-f6-1a-6e-83': <tf.
Tensor 'ExpandDims_18:0' shape=(None, 1) dtype=float32>, '39-b2-6d-1d-c8-b9':
```

```

<tf.Tensor 'ExpandDims_15:0' shape=(None, 1) dtype=float32>, '08-66-98-bc-f7-4
8': <tf.Tensor 'ExpandDims_2:0' shape=(None, 1) dtype=float32>, 'e7-e7-a6-0d-9
7-ab': <tf.Tensor 'ExpandDims_61:0' shape=(None, 1) dtype=float32>, '39-2b-70-
81-47-5d': <tf.Tensor 'ExpandDims_14:0' shape=(None, 1) dtype=float32>, 'd2-10
-29-32-35-43': <tf.Tensor 'ExpandDims_52:0' shape=(None, 1) dtype=float32>, '0
5-8d-16-44-19-69': <tf.Tensor 'ExpandDims_1:0' shape=(None, 1) dtype=float32>,
'66-14-58-7c-19-c7': <tf.Tensor 'ExpandDims_36:0' shape=(None, 1) dtype=float3
2>, '0c-75-76-ab-d5-40': <tf.Tensor 'ExpandDims_4:0' shape=(None, 1) dtype=flo
at32>, '58-8e-81-eb-19-0c': <tf.Tensor 'ExpandDims_29:0' shape=(None, 1) dtype
=float32>, '73-3e-8d-15-10-18': <tf.Tensor 'ExpandDims_40:0' shape=(None, 1) d
type=float32>, 'd8-a7-56-df-44-73': <tf.Tensor 'ExpandDims_55:0' shape=(None,
1) dtype=float32>, '68-85-19-55-e5-e4': <tf.Tensor 'ExpandDims_38:0' shape=(No
ne, 1) dtype=float32>, 'd8-0f-99-78-96-ae': <tf.Tensor 'ExpandDims_54:0' shape
=(None, 1) dtype=float32>, '9c-20-7b-85-4c-d8': <tf.Tensor 'ExpandDims_44:0' s
hape=(None, 1) dtype=float32>, 'e4-6a-ae-ee-21-85': <tf.Tensor 'ExpandDims_59:
0' shape=(None, 1) dtype=float32>, 'ec-81-93-6b-87-8c': <tf.Tensor 'ExpandDims
_62:0' shape=(None, 1) dtype=float32>, '5e-82-77-dc-6d-48': <tf.Tensor 'Expand
Dims_30:0' shape=(None, 1) dtype=float32>, '52-c2-a2-98-da-d8': <tf.Tensor 'Ex
pandDims_25:0' shape=(None, 1) dtype=float32>, '2f-b0-45-ca-d8-aa': <tf.Tensor
'ExpandDims_12:0' shape=(None, 1) dtype=float32>, 'c0-a9-d2-4f-d8-ae': <tf.Ten
sor 'ExpandDims_48:0' shape=(None, 1) dtype=float32>, '8c-79-f5-e2-22-0f': <tf
.Tensor 'ExpandDims_43:0' shape=(None, 1) dtype=float32>, 'f0-b3-ec-33-c3-ee':
<tf.Tensor 'ExpandDims_63:0' shape=(None, 1) dtype=float32>, '3c-9b-50-54-0a-b
7': <tf.Tensor 'ExpandDims_16:0' shape=(None, 1) dtype=float32>, '04-2a-9c-4b-
cb-a2': <tf.Tensor 'ExpandDims:0' shape=(None, 1) dtype=float32>, '28-11-a5-d6
-a2-f4': <tf.Tensor 'ExpandDims_9:0' shape=(None, 1) dtype=float32>, '50-de-06
-9e-2d-c5': <tf.Tensor 'ExpandDims_23:0' shape=(None, 1) dtype=float32>, 'f9-0
8-04-b3-5c-9f': <tf.Tensor 'ExpandDims_68:0' shape=(None, 1) dtype=float32>, '
dd-27-7d-b2-e8-bd': <tf.Tensor 'ExpandDims_57:0' shape=(None, 1) dtype=float32
>, '56-a5-db-f0-ea-4e': <tf.Tensor 'ExpandDims_27:0' shape=(None, 1) dtype=flo
at32>}

```

Consider rewriting this model with the Functional API.

```

428/428 [=====] - 3s 4ms/step - loss: 1.0629 - accuracy: 0.6153 - val_loss: 0.7561 - val_accuracy: 0.6783

```

Epoch 2/100

```

428/428 [=====] - 1s 3ms/step - loss: 0.7257 - accuracy: 0.6862 - val_loss: 0.7087 - val_accuracy: 0.6847

```

Epoch 3/100

```

428/428 [=====] - 1s 3ms/step - loss: 0.6976 - accuracy: 0.6938 - val_loss: 0.6975 - val_accuracy: 0.6973

```

Epoch 4/100

```

428/428 [=====] - 1s 3ms/step - loss: 0.6863 - accuracy: 0.6974 - val_loss: 0.6940 - val_accuracy: 0.6937

```

Epoch 5/100

```

428/428 [=====] - 1s 3ms/step - loss: 0.6771 - accuracy: 0.7031 - val_loss: 0.6748 - val_accuracy: 0.7075

```

Epoch 6/100

```

428/428 [=====] - 1s 3ms/step - loss: 0.6666 - accuracy: 0.7018 - val_loss: 0.6644 - val_accuracy: 0.7069

```

Epoch 7/100

```

428/428 [=====] - 1s 3ms/step - loss: 0.6574 - accuracy: 0.7050 - val_loss: 0.6592 - val_accuracy: 0.6911

```

Epoch 8/100

```
428/428 [=====] - 1s 3ms/step - loss: 0.6458 - accuracy: 0.7045 - val_loss: 0.6496 - val_accuracy: 0.7040
Epoch 9/100
428/428 [=====] - 1s 3ms/step - loss: 0.6323 - accuracy: 0.7140 - val_loss: 0.6309 - val_accuracy: 0.7081
Epoch 10/100
428/428 [=====] - 1s 3ms/step - loss: 0.6143 - accuracy: 0.7173 - val_loss: 0.6175 - val_accuracy: 0.6984
Epoch 11/100
428/428 [=====] - 2s 4ms/step - loss: 0.5970 - accuracy: 0.7271 - val_loss: 0.6069 - val_accuracy: 0.7431
Epoch 12/100
428/428 [=====] - 2s 4ms/step - loss: 0.5772 - accuracy: 0.7389 - val_loss: 0.5747 - val_accuracy: 0.7522
Epoch 13/100
428/428 [=====] - 1s 3ms/step - loss: 0.5583 - accuracy: 0.7503 - val_loss: 0.5636 - val_accuracy: 0.7648
Epoch 14/100
428/428 [=====] - 1s 3ms/step - loss: 0.5396 - accuracy: 0.7623 - val_loss: 0.5401 - val_accuracy: 0.7741
Epoch 15/100
428/428 [=====] - 1s 3ms/step - loss: 0.5239 - accuracy: 0.7683 - val_loss: 0.5314 - val_accuracy: 0.7420
Epoch 16/100
428/428 [=====] - 1s 3ms/step - loss: 0.5065 - accuracy: 0.7825 - val_loss: 0.5122 - val_accuracy: 0.7858
Epoch 17/100
428/428 [=====] - 1s 3ms/step - loss: 0.4935 - accuracy: 0.7948 - val_loss: 0.5011 - val_accuracy: 0.8063
Epoch 18/100
428/428 [=====] - 1s 3ms/step - loss: 0.4765 - accuracy: 0.8059 - val_loss: 0.4802 - val_accuracy: 0.8369
Epoch 19/100
428/428 [=====] - 1s 3ms/step - loss: 0.4613 - accuracy: 0.8162 - val_loss: 0.4670 - val_accuracy: 0.8144
Epoch 20/100
428/428 [=====] - 1s 3ms/step - loss: 0.4471 - accuracy: 0.8241 - val_loss: 0.4554 - val_accuracy: 0.8442
Epoch 21/100
428/428 [=====] - 1s 3ms/step - loss: 0.4325 - accuracy: 0.8309 - val_loss: 0.4354 - val_accuracy: 0.8501
Epoch 22/100
428/428 [=====] - 1s 3ms/step - loss: 0.4190 - accuracy: 0.8395 - val_loss: 0.4202 - val_accuracy: 0.8352
Epoch 23/100
428/428 [=====] - 1s 3ms/step - loss: 0.4060 - accuracy: 0.8431 - val_loss: 0.4154 - val_accuracy: 0.8404
Epoch 24/100
428/428 [=====] - 1s 3ms/step - loss: 0.3944 - accuracy: 0.8507 - val_loss: 0.4039 - val_accuracy: 0.8466
Epoch 25/100
428/428 [=====] - 1s 3ms/step - loss: 0.3817 - accuracy: 0.8537 - val_loss: 0.3861 - val_accuracy: 0.8589
```

Epoch 26/100
428/428 [=====] - 1s 3ms/step - loss: 0.3734 - accuracy: 0.8565 - val_loss: 0.3759 - val_accuracy: 0.8618
Epoch 27/100
428/428 [=====] - 2s 4ms/step - loss: 0.3638 - accuracy: 0.8591 - val_loss: 0.3680 - val_accuracy: 0.8708
Epoch 28/100
428/428 [=====] - 2s 4ms/step - loss: 0.3588 - accuracy: 0.8594 - val_loss: 0.3658 - val_accuracy: 0.8501
Epoch 29/100
428/428 [=====] - 1s 3ms/step - loss: 0.3464 - accuracy: 0.8648 - val_loss: 0.3537 - val_accuracy: 0.8703
Epoch 30/100
428/428 [=====] - 2s 4ms/step - loss: 0.3409 - accuracy: 0.8692 - val_loss: 0.3461 - val_accuracy: 0.8802
Epoch 31/100
428/428 [=====] - 1s 3ms/step - loss: 0.3346 - accuracy: 0.8678 - val_loss: 0.3381 - val_accuracy: 0.8758
Epoch 32/100
428/428 [=====] - 2s 4ms/step - loss: 0.3298 - accuracy: 0.8691 - val_loss: 0.3310 - val_accuracy: 0.8767
Epoch 33/100
428/428 [=====] - 1s 3ms/step - loss: 0.3231 - accuracy: 0.8717 - val_loss: 0.3260 - val_accuracy: 0.8761
Epoch 34/100
428/428 [=====] - 1s 3ms/step - loss: 0.3192 - accuracy: 0.8721 - val_loss: 0.3229 - val_accuracy: 0.8746
Epoch 35/100
428/428 [=====] - 1s 3ms/step - loss: 0.3119 - accuracy: 0.8746 - val_loss: 0.3255 - val_accuracy: 0.8697
Epoch 36/100
428/428 [=====] - 1s 3ms/step - loss: 0.3075 - accuracy: 0.8774 - val_loss: 0.3255 - val_accuracy: 0.8732
Epoch 37/100
428/428 [=====] - 1s 3ms/step - loss: 0.3043 - accuracy: 0.8759 - val_loss: 0.3130 - val_accuracy: 0.8758
Epoch 38/100
428/428 [=====] - 1s 3ms/step - loss: 0.3002 - accuracy: 0.8778 - val_loss: 0.3119 - val_accuracy: 0.8726
Epoch 39/100
428/428 [=====] - 1s 3ms/step - loss: 0.2976 - accuracy: 0.8784 - val_loss: 0.3063 - val_accuracy: 0.8793
Epoch 40/100
428/428 [=====] - 1s 3ms/step - loss: 0.2932 - accuracy: 0.8775 - val_loss: 0.3105 - val_accuracy: 0.8869
Epoch 41/100
428/428 [=====] - 1s 3ms/step - loss: 0.2888 - accuracy: 0.8816 - val_loss: 0.3012 - val_accuracy: 0.8767
Epoch 42/100
428/428 [=====] - 1s 3ms/step - loss: 0.2849 - accuracy: 0.8827 - val_loss: 0.3082 - val_accuracy: 0.8703
Epoch 43/100
428/428 [=====] - 1s 3ms/step - loss: 0.2843 - accuracy:

```
cy: 0.8816 - val_loss: 0.2926 - val_accuracy: 0.8843
Epoch 44/100
428/428 [=====] - 1s 3ms/step - loss: 0.2800 - accuracy: 0.8827 - val_loss: 0.2925 - val_accuracy: 0.8749
Epoch 45/100
428/428 [=====] - 1s 3ms/step - loss: 0.2783 - accuracy: 0.8817 - val_loss: 0.2865 - val_accuracy: 0.8887
Epoch 46/100
428/428 [=====] - 1s 3ms/step - loss: 0.2748 - accuracy: 0.8838 - val_loss: 0.2826 - val_accuracy: 0.8887
Epoch 47/100
428/428 [=====] - 1s 3ms/step - loss: 0.2714 - accuracy: 0.8862 - val_loss: 0.2880 - val_accuracy: 0.8822
Epoch 48/100
428/428 [=====] - 1s 3ms/step - loss: 0.2727 - accuracy: 0.8864 - val_loss: 0.2786 - val_accuracy: 0.8837
Epoch 49/100
428/428 [=====] - 1s 3ms/step - loss: 0.2693 - accuracy: 0.8878 - val_loss: 0.2892 - val_accuracy: 0.8834
Epoch 50/100
428/428 [=====] - 1s 3ms/step - loss: 0.2675 - accuracy: 0.8873 - val_loss: 0.2821 - val_accuracy: 0.8904
Epoch 51/100
428/428 [=====] - 1s 3ms/step - loss: 0.2686 - accuracy: 0.8847 - val_loss: 0.2721 - val_accuracy: 0.8948
Epoch 52/100
428/428 [=====] - 1s 3ms/step - loss: 0.2619 - accuracy: 0.8892 - val_loss: 0.2703 - val_accuracy: 0.8890
Epoch 53/100
428/428 [=====] - 1s 3ms/step - loss: 0.2618 - accuracy: 0.8899 - val_loss: 0.2711 - val_accuracy: 0.8878
Epoch 54/100
428/428 [=====] - 1s 3ms/step - loss: 0.2595 - accuracy: 0.8900 - val_loss: 0.2780 - val_accuracy: 0.8942
Epoch 55/100
428/428 [=====] - 1s 3ms/step - loss: 0.2573 - accuracy: 0.8887 - val_loss: 0.2637 - val_accuracy: 0.8957
Epoch 56/100
428/428 [=====] - 1s 3ms/step - loss: 0.2550 - accuracy: 0.8910 - val_loss: 0.2796 - val_accuracy: 0.8875
Epoch 57/100
428/428 [=====] - 1s 3ms/step - loss: 0.2514 - accuracy: 0.8925 - val_loss: 0.2633 - val_accuracy: 0.8892
Epoch 58/100
428/428 [=====] - 1s 3ms/step - loss: 0.2522 - accuracy: 0.8912 - val_loss: 0.2624 - val_accuracy: 0.8860
Epoch 59/100
428/428 [=====] - 1s 3ms/step - loss: 0.2503 - accuracy: 0.8932 - val_loss: 0.2874 - val_accuracy: 0.8755
Epoch 60/100
428/428 [=====] - 1s 3ms/step - loss: 0.2475 - accuracy: 0.8963 - val_loss: 0.2796 - val_accuracy: 0.8776
Epoch 61/100
```

```
428/428 [=====] - 1s 3ms/step - loss: 0.2448 - accuracy: 0.8965 - val_loss: 0.2573 - val_accuracy: 0.8960
Epoch 62/100
428/428 [=====] - 1s 3ms/step - loss: 0.2470 - accuracy: 0.8946 - val_loss: 0.2556 - val_accuracy: 0.8895
Epoch 63/100
428/428 [=====] - 1s 3ms/step - loss: 0.2457 - accuracy: 0.8921 - val_loss: 0.2619 - val_accuracy: 0.8869
Epoch 64/100
428/428 [=====] - 1s 3ms/step - loss: 0.2434 - accuracy: 0.8938 - val_loss: 0.2653 - val_accuracy: 0.8822
Epoch 65/100
428/428 [=====] - 1s 3ms/step - loss: 0.2396 - accuracy: 0.8976 - val_loss: 0.2503 - val_accuracy: 0.8971
Epoch 66/100
428/428 [=====] - 1s 3ms/step - loss: 0.2414 - accuracy: 0.8967 - val_loss: 0.2608 - val_accuracy: 0.8939
Epoch 67/100
428/428 [=====] - 1s 3ms/step - loss: 0.2395 - accuracy: 0.8954 - val_loss: 0.2591 - val_accuracy: 0.8957
Epoch 68/100
428/428 [=====] - 1s 3ms/step - loss: 0.2370 - accuracy: 0.8965 - val_loss: 0.2564 - val_accuracy: 0.8945
Epoch 69/100
428/428 [=====] - 1s 3ms/step - loss: 0.2368 - accuracy: 0.8981 - val_loss: 0.2491 - val_accuracy: 0.8948
Epoch 70/100
428/428 [=====] - 1s 3ms/step - loss: 0.2369 - accuracy: 0.8966 - val_loss: 0.2478 - val_accuracy: 0.8922
Epoch 71/100
428/428 [=====] - 1s 3ms/step - loss: 0.2342 - accuracy: 0.8997 - val_loss: 0.2563 - val_accuracy: 0.8942
Epoch 72/100
428/428 [=====] - 1s 3ms/step - loss: 0.2333 - accuracy: 0.9001 - val_loss: 0.2501 - val_accuracy: 0.9009
Epoch 73/100
428/428 [=====] - 1s 3ms/step - loss: 0.2323 - accuracy: 0.9010 - val_loss: 0.2381 - val_accuracy: 0.9006
Epoch 74/100
428/428 [=====] - 1s 3ms/step - loss: 0.2308 - accuracy: 0.9001 - val_loss: 0.2466 - val_accuracy: 0.8963
Epoch 75/100
428/428 [=====] - 1s 3ms/step - loss: 0.2316 - accuracy: 0.8998 - val_loss: 0.2405 - val_accuracy: 0.8960
Epoch 76/100
428/428 [=====] - 1s 3ms/step - loss: 0.2266 - accuracy: 0.9032 - val_loss: 0.2440 - val_accuracy: 0.9001
Epoch 77/100
428/428 [=====] - 1s 3ms/step - loss: 0.2306 - accuracy: 0.9006 - val_loss: 0.2375 - val_accuracy: 0.9021
Epoch 78/100
428/428 [=====] - 1s 3ms/step - loss: 0.2271 - accuracy: 0.9012 - val_loss: 0.2550 - val_accuracy: 0.8998
```

Epoch 79/100
428/428 [=====] - 1s 3ms/step - loss: 0.2262 - accuracy: 0.9026 - val_loss: 0.2386 - val_accuracy: 0.9027
Epoch 80/100
428/428 [=====] - 1s 3ms/step - loss: 0.2245 - accuracy: 0.9036 - val_loss: 0.2318 - val_accuracy: 0.9056
Epoch 81/100
428/428 [=====] - 1s 3ms/step - loss: 0.2220 - accuracy: 0.9036 - val_loss: 0.2379 - val_accuracy: 0.8998
Epoch 82/100
428/428 [=====] - 1s 3ms/step - loss: 0.2231 - accuracy: 0.9033 - val_loss: 0.2472 - val_accuracy: 0.8971
Epoch 83/100
428/428 [=====] - 1s 3ms/step - loss: 0.2238 - accuracy: 0.9025 - val_loss: 0.2439 - val_accuracy: 0.8936
Epoch 84/100
428/428 [=====] - 1s 3ms/step - loss: 0.2201 - accuracy: 0.9066 - val_loss: 0.2455 - val_accuracy: 0.8948
Epoch 85/100
428/428 [=====] - 1s 3ms/step - loss: 0.2229 - accuracy: 0.9003 - val_loss: 0.2364 - val_accuracy: 0.8936
Epoch 86/100
428/428 [=====] - 1s 3ms/step - loss: 0.2192 - accuracy: 0.9059 - val_loss: 0.2450 - val_accuracy: 0.8963
Epoch 87/100
428/428 [=====] - 1s 3ms/step - loss: 0.2187 - accuracy: 0.9044 - val_loss: 0.2358 - val_accuracy: 0.8995
Epoch 88/100
428/428 [=====] - 1s 3ms/step - loss: 0.2183 - accuracy: 0.9055 - val_loss: 0.2551 - val_accuracy: 0.8895
Epoch 89/100
428/428 [=====] - 1s 3ms/step - loss: 0.2181 - accuracy: 0.9040 - val_loss: 0.2344 - val_accuracy: 0.8989
Epoch 90/100
428/428 [=====] - 1s 3ms/step - loss: 0.2176 - accuracy: 0.9044 - val_loss: 0.2311 - val_accuracy: 0.8992
Epoch 91/100
428/428 [=====] - 1s 3ms/step - loss: 0.2133 - accuracy: 0.9073 - val_loss: 0.2309 - val_accuracy: 0.8992
Epoch 92/100
428/428 [=====] - 1s 3ms/step - loss: 0.2151 - accuracy: 0.9070 - val_loss: 0.2315 - val_accuracy: 0.8983
Epoch 93/100
428/428 [=====] - 1s 3ms/step - loss: 0.2144 - accuracy: 0.9044 - val_loss: 0.2251 - val_accuracy: 0.9100
Epoch 94/100
428/428 [=====] - 1s 3ms/step - loss: 0.2128 - accuracy: 0.9071 - val_loss: 0.2239 - val_accuracy: 0.8989
Epoch 95/100
428/428 [=====] - 1s 3ms/step - loss: 0.2127 - accuracy: 0.9073 - val_loss: 0.2289 - val_accuracy: 0.8998
Epoch 96/100
428/428 [=====] - 1s 3ms/step - loss: 0.2123 - accuracy:

```

cy: 0.9084 - val_loss: 0.2279 - val_accuracy: 0.9012
Epoch 97/100
428/428 [=====] - 1s 3ms/step - loss: 0.2115 - accuracy: 0.9086 - val_loss: 0.2285 - val_accuracy: 0.9077
Epoch 98/100
428/428 [=====] - 1s 3ms/step - loss: 0.2122 - accuracy: 0.9087 - val_loss: 0.2226 - val_accuracy: 0.9077
Epoch 99/100
428/428 [=====] - 1s 3ms/step - loss: 0.2084 - accuracy: 0.9080 - val_loss: 0.2293 - val_accuracy: 0.9047
Epoch 100/100
428/428 [=====] - 1s 3ms/step - loss: 0.2102 - accuracy: 0.9098 - val_loss: 0.2345 - val_accuracy: 0.8936

```

Out[8]: <keras.callbacks.History at 0x1695cc850>

```

In [9]: test_lbl = np.array([rooms[x] for x in test_df[label_name]])
        test_df_ = test_df.drop(columns=[label_name])
        test_ft = {key:np.array(value) for key, value in test_df_.items()}

        model.evaluate(test_ft, test_lbl)

```

```

60/60 [=====] - 0s 2ms/step - loss: 0.2511 - accuracy: 0.8927

```

Out[9]: [0.25112712383270264, 0.8926880359649658]

```

In [16]: # Convert to softmax as probability distribution
        probability_model = tf.keras.Sequential([model,
                                                  tf.keras.layers.Softmax()])

```

```

In [ ]: # Save the entire model as a SavedModel.
        !mkdir -p saved_model
        model.save('saved_model/my_model')

```


In [17]:

```

def predict(data):
    converted = convert(data)
    predicted = probability_model.predict(converted)
    arg_max = np.argmax(predicted)
    room = rooms_labels[arg_max]
    prob = predicted[0][arg_max]
    return room, prob

def convert(data):
    new_data = dict()
    for column in train_df.columns:
        val = np.array([data.get(column, -200)])
        # Normalize
        new_data[column] = (val - ble_cleaned[column].min()) / (ble_cleaned[c
    return new_data

```

In []:

```

import socket
import json
import traceback

def main():

    host = '10.0.0.132' #Server ip
    port = 4004

    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    s.bind((host, port))

    print("Server Started")
    try:
        while True:
            data, addr = s.recvfrom(10240)
            try:
                data_loaded = json.loads(data) #data loaded
            except json.decoder.JSONDecodeError:
                print("Json error with {}".format(data))
            label, prob = predict(data_loaded)
            predicted_data = {"room": label, "prob": str(prob)}
            message = json.dumps(predicted_data) #data serialized
            s.sendto(message.encode(), addr)
        except KeyboardInterrupt as ki:
            pass
        except Exception as e:
            traceback.print_exc()
        finally:
            print("Ending")
            s.close()

    main()

```

Server Started