part_3

December 10, 2023

1 Imports and Helper Functions:

[ ]:
```
# Import packages
# DL Packages
import tensorflow as tf
import keras

# Others
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import scipy as sp
import sympy as sym
import seaborn as sns

from sklearn.model_selection import KFold
from sklearn.metrics import confusion_matrix
```

```
2023-12-10 21:34:39.270115: I tensorflow/core/util/port.cc:111] oneDNN custom
operations are on. You may see slightly different numerical results due to
floating-point round-off errors from different computation orders. To turn them
off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2023-12-10 21:34:39.292233: E
tensorflow/compiler/xla/stream_executor/cuda/cuda_dnn.cc:9342] Unable to
register cuDNN factory: Attempting to register factory for plugin cuDNN when one
has already been registered
2023-12-10 21:34:39.292253: E
tensorflow/compiler/xla/stream_executor/cuda/cuda_fft.cc:609] Unable to register
cuFFT factory: Attempting to register factory for plugin cuFFT when one has
already been registered
2023-12-10 21:34:39.292269: E
tensorflow/compiler/xla/stream_executor/cuda/cuda_blas.cc:1518] Unable to
register cuBLAS factory: Attempting to register factory for plugin cuBLAS when
one has already been registered
2023-12-10 21:34:39.296329: I tensorflow/core/platform/cpu_feature_guard.cc:182]
This TensorFlow binary is optimized to use available CPU instructions in
performance-critical operations.
To enable the following instructions: AVX2 AVX512F AVX512_VNNI FMA, in other
```

operations, rebuild TensorFlow with the appropriate compiler flags.

```
[ ]: def plot_confusion_matrix(Y: np.array, pred: np.array, labels=[], savename="",␣
   ↪logscale=False, title="Confustion Matrix"):
        """
        Convenience function for generating a confusion Matrix

        Args:
            Y (np.array): Actual labels for the dataset (n rows, 1 column)
            pred (np.array): Predicted labels for the data (n rows, 1 column)
            labels (list of str): class labels
            savename (str, optional): File to save plot to. If none is given shows␣
   ↪figure.
                                    Defaults to "".

        Returns:
            confusion matrix
        """
        # Figure out predicted class -- infer from Y and pred the number of classes
        if Y.shape[1] > 1:
            Y_labels = np.zeros(Y.shape[0], dtype=int)
            pred_labels = np.zeros_like(Y_labels)
            for i in range(Y.shape[0]):
                Y_labels[i] = np.argmax(Y[i])
                pred_labels[i] = np.argmax(pred[i])
        else:
            Y_labels = Y
            pred_labels = (Y >= 0.5).astype(int)
        cm = confusion_matrix(Y_labels, pred_labels)
        f, ax = plt.subplots()
        if logscale:
            from matplotlib.colors import LogNorm, Normalize
            sns.heatmap(cm, annot=True, fmt='g', ax=ax, cmap='Blues',␣
   ↪norm=LogNorm())
        else:
            sns.heatmap(cm, annot=True, fmt='g', ax=ax, cmap='Blues')
        # labels, title and ticks
        ax.set_xlabel("Predicted labels")
        ax.set_ylabel("True labels")
        ax.set_title(title)
        if not labels:
            labels = np.arange(max(Y.shape[1], 2))
        ax.xaxis.set_ticklabels(labels)
        ax.yaxis.set_ticklabels(labels)

        if savename != "":
            plt.savefig(savename)
```

```
        plt.close(f)
    else:
        f.set_size_inches((8,8))
        plt.tight_layout()
        plt.show()

    return cm
```

2 Examine and Preprocess the Data:

2.1 Examine Data

[ ]:
```
data = pd.read_csv("Final_News_DF_Labeled_ExamDataset.csv")
data
```

[ ]:
```
         LABEL  according  agency  ahead  alabama  amazon  america  american  \
0     politics          0       0      0        0       0        0         0
1     politics          0       0      0        0       0        0         0
2     politics          0       0      0        0       0        0         0
3     politics          0       0      0        0       0        0         0
4     politics          0       0      0        0       0        0         0
…          …          …       …      …        …       …        …         …
1488   science          0       0      0        0       0        0         0
1489   science          0       0      0        0       0        0         0
1490   science          0       0      0        0       0        0         0
1491   science          0       0      0        0       0        0         0
1492   science          0       0      0        0       0        0         0

      announced  appeared  …  wolverines  women  work  working  world  \
0             1         0  …           0      0     0        0      0
1             0         0  …           0      0     0        0      0
2             0         0  …           0      0     0        0      0
3             0         0  …           0      0     0        0      0
4             0         0  …           0      0     0        0      0
…           …         …  …         …      …     …        …      …
1488          0         0  …           0      0     0        0      0
1489          0         0  …           0      0     0        0      0
1490          0         0  …           0      0     0        0      0
1491          0         0  …           0      0     0        0      0
1492          0         0  …           0      0     0        0      0

      wrote  year  years  york  young
0         0     0      0     0      0
1         0     0      1     0      0
2         0     0      0     0      0
3         0     0      0     0      0
4         0     0      0     0      0
```

```
...     ...  ...    ...  ...    ...
1488       0      0       0      0       0
1489       0      0       0      0       0
1490       0      0       0      0       0
1491       0      0       0      0       0
1492       0      0       0      0       0

[1493 rows x 301 columns]
```

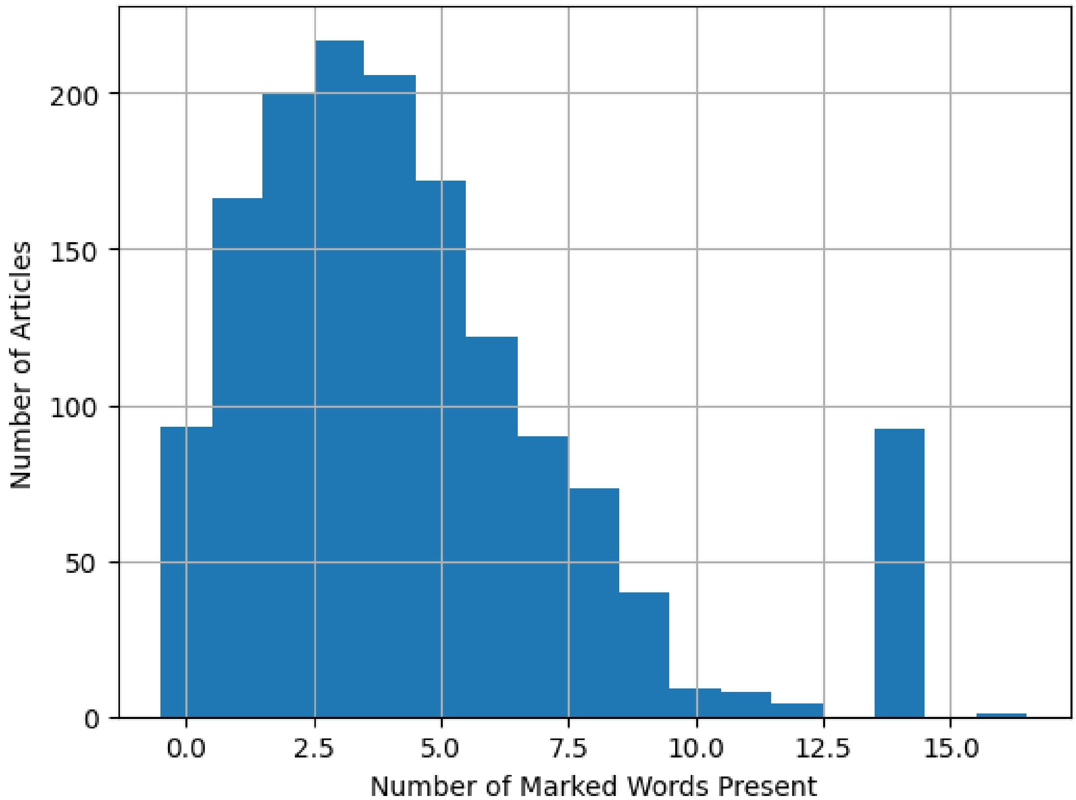
Each entry is labeled by a catergory of article, with the columns representing words that appeared or not

[ ]:
```
data.groupby("LABEL").size()
```

[ ]:
```
LABEL
football    500
politics    497
science     496
dtype: int64
```

[ ]:
```
nwords_present = data[[c for c in data.columns if c != "LABEL"]].sum(axis=1)
print("fewest words present", nwords_present.min())
print("most words present", nwords_present.max())
nwords_present.hist(bins=nwords_present.max()-nwords_present.min(),align="left")
plt.ylabel("Number of Articles")
plt.xlabel("Number of Marked Words Present");
```

```
fewest words present 0
most words present 17
```

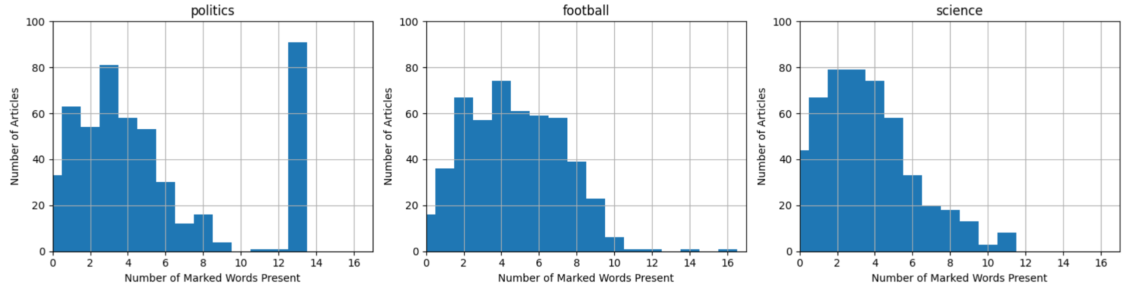


The data is pretty sparse, meaning almost all articles have fewer than 10 marked words.

[ ]:
```
f, ax = plt.subplots(ncols=3)
f.set_size_inches((15,4))

for i, label in enumerate(data.LABEL.unique()):
    nwords_present = data[data.LABEL == label][[c for c in data.columns if c !=␣
↪"LABEL"]].sum(axis=1)
    ax[i].hist(x = nwords_present.values, bins=nwords_present.
↪max()-nwords_present.min(),align="left")
    ax[i].grid()
    ax[i].set_ylabel("Number of Articles")
    ax[i].set_xlabel("Number of Marked Words Present");
    ax[i].set_title(label)
    ax[i].set_xlim(0,17)
    ax[i].set_ylim(0,100)

plt.tight_layout();
```



2.2 Preprocess Data

Turn it into data matrix X and label vector Y. X is already OHE, while Y needs to be turned into OHE.

[ ]:
```
X = data[[c for c in data.columns if c != "LABEL"]].to_numpy()
Y_raw = data["LABEL"].to_numpy()

# Do the OHE
vals = np.unique(Y_raw)
val_map = {}
for i, v in enumerate(vals):
    val_map[v] = i
Y = np.zeros((Y_raw.shape[0], len(val_map)), dtype=int)
for i in range(Y.shape[0]):
    Y[i, val_map[Y_raw[i]]] = 1

print("val_map", val_map)
print("X shape:", X.shape)
print("X:\n", X)
print("Y shape:", Y.shape)
print("Y:\n", Y)
```

```
val_map {'football': 0, 'politics': 1, 'science': 2}
X shape: (1493, 300)
X:
 [[0 0 0 … 0 0 0]
 [0 0 0 … 1 0 0]
 [0 0 0 … 0 0 0]
 …
 [0 0 0 … 0 0 0]
 [0 0 0 … 0 0 0]
 [0 0 0 … 0 0 0]]
Y shape: (1493, 3)
Y:
 [[0 1 0]
 [0 1 0]
```

```
 [0 1 0]
 …
 [0 0 1]
 [0 0 1]
 [0 0 1]]
```

Preprocessing looks good. The data is balanced, with an approximately equal number of entries from each class. As it did not come with a specified train/test split, I'll do my own n-fold cross validation. As there are approximately 500 in each class, I'll do 5 fold to get about 100 of each in each test set.

```
[ ]: splits = KFold(n_splits=5, shuffle=True, random_state=7)
```

Let's look at the balance of the splits to make sure it worked:

```
[ ]: fold = 1
for train, test in splits.split(X, Y_raw):
    print("---------------------------")
    print("FOLD:", fold)
    print("TRAIN -", ", ".join([f"{y[0]}: {y[1]}" for y in zip(*[list(x) for x␣
 ↪in np.unique(Y_raw[train], return_counts=True)])]))
    print("VAL -", ", ".join([f"{y[0]}: {y[1]}" for y in zip(*[list(x) for x in␣
 ↪np.unique(Y_raw[test], return_counts=True)])]))
    fold += 1
```

```
---------------------------
FOLD: 1
TRAIN - football: 400, politics: 391, science: 403
VAL - football: 100, politics: 106, science: 93
---------------------------
FOLD: 2
TRAIN - football: 402, politics: 396, science: 396
VAL - football: 98, politics: 101, science: 100
---------------------------
FOLD: 3
TRAIN - football: 402, politics: 402, science: 390
VAL - football: 98, politics: 95, science: 106
---------------------------
FOLD: 4
TRAIN - football: 393, politics: 404, science: 398
VAL - football: 107, politics: 93, science: 98
---------------------------
FOLD: 5
TRAIN - football: 403, politics: 395, science: 397
VAL - football: 97, politics: 102, science: 99
```

Folds are looking balanced, let's move ahead with making some models after defining some helpers for the folding:

[ ]:
```python
def train_on_folds(model_fn, splits, X, Y, epochs=100, batch_size=2000):

    models = []
    histories = []

    # Train a different model for each fold
    for fold_n, (train, test) in enumerate(splits.split(X, Y)):

        print("------------------")
        print("Training on fold", fold_n)

        X_train = X[train]
        Y_train = Y[train]

        X_test = X[test]
        Y_test = Y[test]

        model = model_fn()
        history = model.fit(X_train, Y_train, epochs=epochs, validation_data =
 ↪(X_test, Y_test))

        models.append(model)
        histories.append(history)

    return models, histories

def pred_on_folds(models, splits, X, Y):

    Y_pred = np.zeros_like(Y, dtype=float)
    for fold_n, (train, test) in enumerate(splits.split(X, Y)):
        preds = models[fold_n].predict(X[test])
        Y_pred[test] = preds

    return Y_pred
```

3 ANN Model:

I honestly don't think there's too much information in the tokenization as is. I think the best we'll be able to do is to essentially find that statistically some combinations of words appear more often in certain types of articles than others. With more data maybe or some transfer learning maybe we could do more.

For the ANN model I will I will include just one small hidden layer before the softmax:

3.1 Define and Train

[ ]:
```
def ann_model(input_size=X.shape[1], output_size=Y.shape[1], optimizer="adam",␣
↪loss="categorical_crossentropy",
              metrics=[keras.metrics.CategoricalAccuracy()]):

    model = keras.models.Sequential([
                keras.layers.Input(input_size),
                keras.layers.Dropout(0.1),
                keras.layers.Dense(2*output_size, activation="sigmoid"),
                keras.layers.Dense(output_size, activation="softmax")
            ])

    model.build(input_size)
    model.compile(optimizer=optimizer, loss=loss, metrics=metrics)

    return model

ann_test = ann_model()
ann_test.summary()
```

```
Model: "sequential"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dropout (Dropout)           (None, 300)               0

 dense (Dense)               (None, 6)                 1806

 dense_1 (Dense)             (None, 3)                 21

=================================================================
Total params: 1827 (7.14 KB)
Trainable params: 1827 (7.14 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
2023-12-10 21:34:41.294099: I
tensorflow/compiler/xla/stream_executor/cuda/cuda_gpu_executor.cc:894]
successful NUMA node read from SysFS had negative value (-1), but there must be
at least one NUMA node, so returning NUMA node zero. See more at
https://github.com/torvalds/linux/blob/v6.0/Documentation/ABI/testing/sysfs-bus-
pci#L344-L355
2023-12-10 21:34:41.297310: I
tensorflow/compiler/xla/stream_executor/cuda/cuda_gpu_executor.cc:894]
successful NUMA node read from SysFS had negative value (-1), but there must be
at least one NUMA node, so returning NUMA node zero. See more at
https://github.com/torvalds/linux/blob/v6.0/Documentation/ABI/testing/sysfs-bus-
pci#L344-L355
```

```
2023-12-10 21:34:41.297403: I
tensorflow/compiler/xla/stream_executor/cuda/cuda_gpu_executor.cc:894]
successful NUMA node read from SysFS had negative value (-1), but there must be
at least one NUMA node, so returning NUMA node zero. See more at
https://github.com/torvalds/linux/blob/v6.0/Documentation/ABI/testing/sysfs-bus-
pci#L344-L355
2023-12-10 21:34:41.298369: I
tensorflow/compiler/xla/stream_executor/cuda/cuda_gpu_executor.cc:894]
successful NUMA node read from SysFS had negative value (-1), but there must be
at least one NUMA node, so returning NUMA node zero. See more at
https://github.com/torvalds/linux/blob/v6.0/Documentation/ABI/testing/sysfs-bus-
pci#L344-L355
2023-12-10 21:34:41.298463: I
tensorflow/compiler/xla/stream_executor/cuda/cuda_gpu_executor.cc:894]
successful NUMA node read from SysFS had negative value (-1), but there must be
at least one NUMA node, so returning NUMA node zero. See more at
https://github.com/torvalds/linux/blob/v6.0/Documentation/ABI/testing/sysfs-bus-
pci#L344-L355
2023-12-10 21:34:41.298520: I
tensorflow/compiler/xla/stream_executor/cuda/cuda_gpu_executor.cc:894]
successful NUMA node read from SysFS had negative value (-1), but there must be
at least one NUMA node, so returning NUMA node zero. See more at
https://github.com/torvalds/linux/blob/v6.0/Documentation/ABI/testing/sysfs-bus-
pci#L344-L355
2023-12-10 21:34:41.346975: I
tensorflow/compiler/xla/stream_executor/cuda/cuda_gpu_executor.cc:894]
successful NUMA node read from SysFS had negative value (-1), but there must be
at least one NUMA node, so returning NUMA node zero. See more at
https://github.com/torvalds/linux/blob/v6.0/Documentation/ABI/testing/sysfs-bus-
pci#L344-L355
2023-12-10 21:34:41.347069: I
tensorflow/compiler/xla/stream_executor/cuda/cuda_gpu_executor.cc:894]
successful NUMA node read from SysFS had negative value (-1), but there must be
at least one NUMA node, so returning NUMA node zero. See more at
https://github.com/torvalds/linux/blob/v6.0/Documentation/ABI/testing/sysfs-bus-
pci#L344-L355
2023-12-10 21:34:41.347136: I
tensorflow/compiler/xla/stream_executor/cuda/cuda_gpu_executor.cc:894]
successful NUMA node read from SysFS had negative value (-1), but there must be
at least one NUMA node, so returning NUMA node zero. See more at
https://github.com/torvalds/linux/blob/v6.0/Documentation/ABI/testing/sysfs-bus-
pci#L344-L355
2023-12-10 21:34:41.347189: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1886] Created device
/job:localhost/replica:0/task:0/device:GPU:0 with 18426 MB memory:  -> device:
0, name: NVIDIA RTX A4500, pci bus id: 0000:01:00.0, compute capability: 8.6
```

[ ]:
```
models_ann, histories_ann = train_on_folds(ann_model, splits, X, Y, epochs=60)
pred_ann = pred_on_folds(models_ann, splits, X, Y)
```

```
-----------------
Training on fold 0
Epoch 1/60

2023-12-10 21:34:41.854673: I tensorflow/tsl/platform/default/subprocess.cc:304]
Start cannot spawn child process: No such file or directory

 1/38 […] - ETA: 29s - loss: 1.1948 -
categorical_accuracy: 0.4062

2023-12-10 21:34:42.159061: I tensorflow/compiler/xla/service/service.cc:168]
XLA service 0x7f07a02b3f70 initialized for platform CUDA (this does not
guarantee that XLA will be used). Devices:
2023-12-10 21:34:42.159079: I tensorflow/compiler/xla/service/service.cc:176]
StreamExecutor device (0): NVIDIA RTX A4500, Compute Capability 8.6
2023-12-10 21:34:42.162069: I
tensorflow/compiler/mlir/tensorflow/utils/dump_mlir_util.cc:269] disabling MLIR
crash reproducer, set env var `MLIR_CRASH_REPRODUCER_DIRECTORY` to enable.
2023-12-10 21:34:42.170000: I
tensorflow/compiler/xla/stream_executor/cuda/cuda_dnn.cc:442] Loaded cuDNN
version 8700
2023-12-10 21:34:42.209930: I ./tensorflow/compiler/jit/device_compiler.h:186]
Compiled cluster using XLA!  This line is logged at most once for the lifetime
of the process.

38/38 [==============================] - 1s 4ms/step - loss: 1.1236 -
categorical_accuracy: 0.3928 - val_loss: 1.0802 - val_categorical_accuracy:
0.4582
Epoch 2/60
 1/38 […] - ETA: 0s - loss: 1.0965 -
categorical_accuracy: 0.4375Epoch 2/60
38/38 [==============================] - 0s 2ms/step - loss: 1.0772 -
categorical_accuracy: 0.4263 - val_loss: 1.0471 - val_categorical_accuracy:
0.4916
Epoch 3/60
38/38 [==============================] - 0s 2ms/step - loss: 1.0467 -
categorical_accuracy: 0.4590 - val_loss: 1.0227 - val_categorical_accuracy:
0.5084
Epoch 4/60
38/38 [==============================] - 0s 2ms/step - loss: 1.0229 -
categorical_accuracy: 0.5352 - val_loss: 1.0029 - val_categorical_accuracy:
0.5920
Epoch 5/60
38/38 [==============================] - 0s 2ms/step - loss: 1.0018 -
categorical_accuracy: 0.6281 - val_loss: 0.9848 - val_categorical_accuracy:
0.6656
Epoch 6/60
```

```
38/38 [==============================] - 0s 2ms/step - loss: 0.9827 -
categorical_accuracy: 0.6734 - val_loss: 0.9665 - val_categorical_accuracy:
0.6957
Epoch 7/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9658 -
categorical_accuracy: 0.7136 - val_loss: 0.9497 - val_categorical_accuracy:
0.7191
Epoch 8/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9470 -
categorical_accuracy: 0.7136 - val_loss: 0.9323 - val_categorical_accuracy:
0.7090
Epoch 9/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9282 -
categorical_accuracy: 0.7328 - val_loss: 0.9157 - val_categorical_accuracy:
0.7358
Epoch 10/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9156 -
categorical_accuracy: 0.7529 - val_loss: 0.8993 - val_categorical_accuracy:
0.7291
Epoch 11/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8963 -
categorical_accuracy: 0.7697 - val_loss: 0.8824 - val_categorical_accuracy:
0.7458
Epoch 12/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8766 -
categorical_accuracy: 0.7789 - val_loss: 0.8658 - val_categorical_accuracy:
0.7391
Epoch 13/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8606 -
categorical_accuracy: 0.7680 - val_loss: 0.8496 - val_categorical_accuracy:
0.7559
Epoch 14/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8437 -
categorical_accuracy: 0.7714 - val_loss: 0.8339 - val_categorical_accuracy:
0.7625
Epoch 15/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8264 -
categorical_accuracy: 0.7789 - val_loss: 0.8183 - val_categorical_accuracy:
0.7625
Epoch 16/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8077 -
categorical_accuracy: 0.7873 - val_loss: 0.8031 - val_categorical_accuracy:
0.7659
Epoch 17/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7915 -
categorical_accuracy: 0.7864 - val_loss: 0.7879 - val_categorical_accuracy:
0.7692
Epoch 18/60
```

```
38/38 [==============================] - 0s 2ms/step - loss: 0.7758 -
categorical_accuracy: 0.7873 - val_loss: 0.7739 - val_categorical_accuracy:
0.7692
Epoch 19/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7615 -
categorical_accuracy: 0.7839 - val_loss: 0.7596 - val_categorical_accuracy:
0.7726
Epoch 20/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7452 -
categorical_accuracy: 0.7923 - val_loss: 0.7467 - val_categorical_accuracy:
0.7726
Epoch 21/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7349 -
categorical_accuracy: 0.7831 - val_loss: 0.7339 - val_categorical_accuracy:
0.7726
Epoch 22/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7183 -
categorical_accuracy: 0.7982 - val_loss: 0.7214 - val_categorical_accuracy:
0.7759
Epoch 23/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7073 -
categorical_accuracy: 0.7864 - val_loss: 0.7101 - val_categorical_accuracy:
0.7793
Epoch 24/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6942 -
categorical_accuracy: 0.7915 - val_loss: 0.6994 - val_categorical_accuracy:
0.7793
Epoch 25/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6822 -
categorical_accuracy: 0.7956 - val_loss: 0.6891 - val_categorical_accuracy:
0.7893
Epoch 26/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6675 -
categorical_accuracy: 0.7965 - val_loss: 0.6789 - val_categorical_accuracy:
0.7826
Epoch 27/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6568 -
categorical_accuracy: 0.8040 - val_loss: 0.6697 - val_categorical_accuracy:
0.7860
Epoch 28/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6476 -
categorical_accuracy: 0.8007 - val_loss: 0.6604 - val_categorical_accuracy:
0.7893
Epoch 29/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6380 -
categorical_accuracy: 0.8090 - val_loss: 0.6518 - val_categorical_accuracy:
0.7893
Epoch 30/60
```

```
38/38 [==============================] - 0s 2ms/step - loss: 0.6277 -
categorical_accuracy: 0.7931 - val_loss: 0.6440 - val_categorical_accuracy:
0.7893
Epoch 31/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6189 -
categorical_accuracy: 0.8015 - val_loss: 0.6361 - val_categorical_accuracy:
0.7860
Epoch 32/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6057 -
categorical_accuracy: 0.8057 - val_loss: 0.6294 - val_categorical_accuracy:
0.7893
Epoch 33/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6016 -
categorical_accuracy: 0.8090 - val_loss: 0.6222 - val_categorical_accuracy:
0.7893
Epoch 34/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5930 -
categorical_accuracy: 0.8049 - val_loss: 0.6160 - val_categorical_accuracy:
0.7826
Epoch 35/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5790 -
categorical_accuracy: 0.8107 - val_loss: 0.6103 - val_categorical_accuracy:
0.7793
Epoch 36/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5787 -
categorical_accuracy: 0.8065 - val_loss: 0.6043 - val_categorical_accuracy:
0.7860
Epoch 37/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5658 -
categorical_accuracy: 0.8166 - val_loss: 0.5990 - val_categorical_accuracy:
0.7826
Epoch 38/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5575 -
categorical_accuracy: 0.8099 - val_loss: 0.5939 - val_categorical_accuracy:
0.7860
Epoch 39/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5550 -
categorical_accuracy: 0.8116 - val_loss: 0.5897 - val_categorical_accuracy:
0.7860
Epoch 40/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5428 -
categorical_accuracy: 0.8174 - val_loss: 0.5850 - val_categorical_accuracy:
0.7793
Epoch 41/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5434 -
categorical_accuracy: 0.8099 - val_loss: 0.5810 - val_categorical_accuracy:
0.7826
Epoch 42/60
```

```
38/38 [==============================] - 0s 2ms/step - loss: 0.5390 -
categorical_accuracy: 0.8124 - val_loss: 0.5778 - val_categorical_accuracy:
0.7826
Epoch 43/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5282 -
categorical_accuracy: 0.8224 - val_loss: 0.5739 - val_categorical_accuracy:
0.7793
Epoch 44/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5269 -
categorical_accuracy: 0.8183 - val_loss: 0.5704 - val_categorical_accuracy:
0.7793
Epoch 45/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5200 -
categorical_accuracy: 0.8191 - val_loss: 0.5674 - val_categorical_accuracy:
0.7793
Epoch 46/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5165 -
categorical_accuracy: 0.8132 - val_loss: 0.5642 - val_categorical_accuracy:
0.7759
Epoch 47/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5108 -
categorical_accuracy: 0.8191 - val_loss: 0.5621 - val_categorical_accuracy:
0.7759
Epoch 48/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5069 -
categorical_accuracy: 0.8183 - val_loss: 0.5589 - val_categorical_accuracy:
0.7726
Epoch 49/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4981 -
categorical_accuracy: 0.8308 - val_loss: 0.5566 - val_categorical_accuracy:
0.7726
Epoch 50/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4972 -
categorical_accuracy: 0.8141 - val_loss: 0.5546 - val_categorical_accuracy:
0.7726
Epoch 51/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4900 -
categorical_accuracy: 0.8291 - val_loss: 0.5524 - val_categorical_accuracy:
0.7726
Epoch 52/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4929 -
categorical_accuracy: 0.8183 - val_loss: 0.5504 - val_categorical_accuracy:
0.7793
Epoch 53/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4800 -
categorical_accuracy: 0.8266 - val_loss: 0.5483 - val_categorical_accuracy:
0.7793
Epoch 54/60
```

```
38/38 [==============================] - 0s 2ms/step - loss: 0.4771 -
categorical_accuracy: 0.8342 - val_loss: 0.5469 - val_categorical_accuracy:
0.7793
Epoch 55/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4788 -
categorical_accuracy: 0.8208 - val_loss: 0.5458 - val_categorical_accuracy:
0.7826
Epoch 56/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4730 -
categorical_accuracy: 0.8241 - val_loss: 0.5443 - val_categorical_accuracy:
0.7793
Epoch 57/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4725 -
categorical_accuracy: 0.8191 - val_loss: 0.5427 - val_categorical_accuracy:
0.7759
Epoch 58/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4719 -
categorical_accuracy: 0.8208 - val_loss: 0.5417 - val_categorical_accuracy:
0.7826
Epoch 59/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4641 -
categorical_accuracy: 0.8241 - val_loss: 0.5406 - val_categorical_accuracy:
0.7826
Epoch 60/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4645 -
categorical_accuracy: 0.8308 - val_loss: 0.5396 - val_categorical_accuracy:
0.7826
------------------
Training on fold 1
Epoch 1/60
38/38 [==============================] - 0s 3ms/step - loss: 1.1301 -
categorical_accuracy: 0.4253 - val_loss: 1.1003 - val_categorical_accuracy:
0.3545
Epoch 2/60
38/38 [==============================] - 0s 2ms/step - loss: 1.0771 -
categorical_accuracy: 0.3953 - val_loss: 1.0580 - val_categorical_accuracy:
0.4381
Epoch 3/60
38/38 [==============================] - 0s 2ms/step - loss: 1.0390 -
categorical_accuracy: 0.5084 - val_loss: 1.0270 - val_categorical_accuracy:
0.4983
Epoch 4/60
38/38 [==============================] - 0s 2ms/step - loss: 1.0117 -
categorical_accuracy: 0.5863 - val_loss: 1.0030 - val_categorical_accuracy:
0.6120
Epoch 5/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9886 -
categorical_accuracy: 0.6533 - val_loss: 0.9821 - val_categorical_accuracy:
```

```
0.7023
Epoch 6/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9649 -
categorical_accuracy: 0.7136 - val_loss: 0.9637 - val_categorical_accuracy:
0.7258
Epoch 7/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9465 -
categorical_accuracy: 0.7286 - val_loss: 0.9460 - val_categorical_accuracy:
0.7124
Epoch 8/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9277 -
categorical_accuracy: 0.7412 - val_loss: 0.9292 - val_categorical_accuracy:
0.7090
Epoch 9/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9071 -
categorical_accuracy: 0.7521 - val_loss: 0.9129 - val_categorical_accuracy:
0.7157
Epoch 10/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8901 -
categorical_accuracy: 0.7471 - val_loss: 0.8965 - val_categorical_accuracy:
0.7258
Epoch 11/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8704 -
categorical_accuracy: 0.7655 - val_loss: 0.8814 - val_categorical_accuracy:
0.7224
Epoch 12/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8517 -
categorical_accuracy: 0.7680 - val_loss: 0.8660 - val_categorical_accuracy:
0.7224
Epoch 13/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8333 -
categorical_accuracy: 0.7605 - val_loss: 0.8506 - val_categorical_accuracy:
0.7492
Epoch 14/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8202 -
categorical_accuracy: 0.7647 - val_loss: 0.8362 - val_categorical_accuracy:
0.7324
Epoch 15/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8005 -
categorical_accuracy: 0.7764 - val_loss: 0.8224 - val_categorical_accuracy:
0.7525
Epoch 16/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7833 -
categorical_accuracy: 0.7848 - val_loss: 0.8091 - val_categorical_accuracy:
0.7559
Epoch 17/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7671 -
categorical_accuracy: 0.7822 - val_loss: 0.7964 - val_categorical_accuracy:
```

```
0.7592
Epoch 18/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7548 -
categorical_accuracy: 0.7814 - val_loss: 0.7835 - val_categorical_accuracy:
0.7592
Epoch 19/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7402 -
categorical_accuracy: 0.7856 - val_loss: 0.7711 - val_categorical_accuracy:
0.7592
Epoch 20/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7249 -
categorical_accuracy: 0.7923 - val_loss: 0.7601 - val_categorical_accuracy:
0.7559
Epoch 21/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7118 -
categorical_accuracy: 0.7831 - val_loss: 0.7497 - val_categorical_accuracy:
0.7592
Epoch 22/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6991 -
categorical_accuracy: 0.7940 - val_loss: 0.7390 - val_categorical_accuracy:
0.7726
Epoch 23/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6841 -
categorical_accuracy: 0.7940 - val_loss: 0.7287 - val_categorical_accuracy:
0.7659
Epoch 24/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6676 -
categorical_accuracy: 0.7940 - val_loss: 0.7197 - val_categorical_accuracy:
0.7659
Epoch 25/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6632 -
categorical_accuracy: 0.7965 - val_loss: 0.7111 - val_categorical_accuracy:
0.7692
Epoch 26/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6510 -
categorical_accuracy: 0.7982 - val_loss: 0.7025 - val_categorical_accuracy:
0.7692
Epoch 27/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6425 -
categorical_accuracy: 0.8015 - val_loss: 0.6947 - val_categorical_accuracy:
0.7726
Epoch 28/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6247 -
categorical_accuracy: 0.8116 - val_loss: 0.6869 - val_categorical_accuracy:
0.7726
Epoch 29/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6214 -
categorical_accuracy: 0.7931 - val_loss: 0.6790 - val_categorical_accuracy:
```

```
0.7692
Epoch 30/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6102 -
categorical_accuracy: 0.8015 - val_loss: 0.6724 - val_categorical_accuracy:
0.7726
Epoch 31/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6002 -
categorical_accuracy: 0.8099 - val_loss: 0.6663 - val_categorical_accuracy:
0.7726
Epoch 32/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5923 -
categorical_accuracy: 0.8049 - val_loss: 0.6596 - val_categorical_accuracy:
0.7759
Epoch 33/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5881 -
categorical_accuracy: 0.7998 - val_loss: 0.6542 - val_categorical_accuracy:
0.7726
Epoch 34/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5812 -
categorical_accuracy: 0.8074 - val_loss: 0.6491 - val_categorical_accuracy:
0.7659
Epoch 35/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5668 -
categorical_accuracy: 0.8074 - val_loss: 0.6439 - val_categorical_accuracy:
0.7659
Epoch 36/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5638 -
categorical_accuracy: 0.8124 - val_loss: 0.6390 - val_categorical_accuracy:
0.7726
Epoch 37/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5536 -
categorical_accuracy: 0.8141 - val_loss: 0.6339 - val_categorical_accuracy:
0.7692
Epoch 38/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5535 -
categorical_accuracy: 0.8040 - val_loss: 0.6297 - val_categorical_accuracy:
0.7659
Epoch 39/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5475 -
categorical_accuracy: 0.8040 - val_loss: 0.6258 - val_categorical_accuracy:
0.7726
Epoch 40/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5421 -
categorical_accuracy: 0.8082 - val_loss: 0.6225 - val_categorical_accuracy:
0.7692
Epoch 41/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5296 -
categorical_accuracy: 0.8174 - val_loss: 0.6190 - val_categorical_accuracy:
```

```
0.7692
Epoch 42/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5348 -
categorical_accuracy: 0.8074 - val_loss: 0.6159 - val_categorical_accuracy:
0.7726
Epoch 43/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5316 -
categorical_accuracy: 0.8124 - val_loss: 0.6132 - val_categorical_accuracy:
0.7726
Epoch 44/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5197 -
categorical_accuracy: 0.8116 - val_loss: 0.6106 - val_categorical_accuracy:
0.7692
Epoch 45/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5117 -
categorical_accuracy: 0.8141 - val_loss: 0.6085 - val_categorical_accuracy:
0.7592
Epoch 46/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5092 -
categorical_accuracy: 0.8191 - val_loss: 0.6063 - val_categorical_accuracy:
0.7559
Epoch 47/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5047 -
categorical_accuracy: 0.8099 - val_loss: 0.6041 - val_categorical_accuracy:
0.7592
Epoch 48/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5006 -
categorical_accuracy: 0.8141 - val_loss: 0.6015 - val_categorical_accuracy:
0.7592
Epoch 49/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4932 -
categorical_accuracy: 0.8183 - val_loss: 0.5996 - val_categorical_accuracy:
0.7592
Epoch 50/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4896 -
categorical_accuracy: 0.8208 - val_loss: 0.5982 - val_categorical_accuracy:
0.7659
Epoch 51/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4912 -
categorical_accuracy: 0.8124 - val_loss: 0.5967 - val_categorical_accuracy:
0.7625
Epoch 52/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4791 -
categorical_accuracy: 0.8191 - val_loss: 0.5952 - val_categorical_accuracy:
0.7592
Epoch 53/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4766 -
categorical_accuracy: 0.8233 - val_loss: 0.5940 - val_categorical_accuracy:
```

```
0.7592
Epoch 54/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4800 -
categorical_accuracy: 0.8183 - val_loss: 0.5923 - val_categorical_accuracy:
0.7525
Epoch 55/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4795 -
categorical_accuracy: 0.8174 - val_loss: 0.5915 - val_categorical_accuracy:
0.7592
Epoch 56/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4793 -
categorical_accuracy: 0.8241 - val_loss: 0.5904 - val_categorical_accuracy:
0.7559
Epoch 57/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4665 -
categorical_accuracy: 0.8283 - val_loss: 0.5899 - val_categorical_accuracy:
0.7559
Epoch 58/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4633 -
categorical_accuracy: 0.8183 - val_loss: 0.5891 - val_categorical_accuracy:
0.7525
Epoch 59/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4665 -
categorical_accuracy: 0.8241 - val_loss: 0.5878 - val_categorical_accuracy:
0.7525
Epoch 60/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4657 -
categorical_accuracy: 0.8216 - val_loss: 0.5882 - val_categorical_accuracy:
0.7525
-----------------
Training on fold 2
Epoch 1/60
38/38 [==============================] - 0s 3ms/step - loss: 1.0997 -
categorical_accuracy: 0.4153 - val_loss: 1.0848 - val_categorical_accuracy:
0.3813
Epoch 2/60
38/38 [==============================] - 0s 2ms/step - loss: 1.0778 -
categorical_accuracy: 0.4447 - val_loss: 1.0717 - val_categorical_accuracy:
0.5452
Epoch 3/60
38/38 [==============================] - 0s 2ms/step - loss: 1.0608 -
categorical_accuracy: 0.5838 - val_loss: 1.0598 - val_categorical_accuracy:
0.5886
Epoch 4/60
38/38 [==============================] - 0s 2ms/step - loss: 1.0453 -
categorical_accuracy: 0.6080 - val_loss: 1.0480 - val_categorical_accuracy:
0.5318
Epoch 5/60
```

```
38/38 [==============================] - 0s 2ms/step - loss: 1.0294 -
categorical_accuracy: 0.6256 - val_loss: 1.0349 - val_categorical_accuracy:
0.5518
Epoch 6/60
38/38 [==============================] - 0s 2ms/step - loss: 1.0127 -
categorical_accuracy: 0.6675 - val_loss: 1.0218 - val_categorical_accuracy:
0.5552
Epoch 7/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9965 -
categorical_accuracy: 0.6633 - val_loss: 1.0075 - val_categorical_accuracy:
0.5619
Epoch 8/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9780 -
categorical_accuracy: 0.6817 - val_loss: 0.9933 - val_categorical_accuracy:
0.6120
Epoch 9/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9614 -
categorical_accuracy: 0.6918 - val_loss: 0.9779 - val_categorical_accuracy:
0.6120
Epoch 10/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9441 -
categorical_accuracy: 0.7027 - val_loss: 0.9630 - val_categorical_accuracy:
0.6421
Epoch 11/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9218 -
categorical_accuracy: 0.7085 - val_loss: 0.9468 - val_categorical_accuracy:
0.6455
Epoch 12/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9020 -
categorical_accuracy: 0.7152 - val_loss: 0.9313 - val_categorical_accuracy:
0.6455
Epoch 13/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8837 -
categorical_accuracy: 0.7320 - val_loss: 0.9154 - val_categorical_accuracy:
0.6656
Epoch 14/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8609 -
categorical_accuracy: 0.7404 - val_loss: 0.9001 - val_categorical_accuracy:
0.6689
Epoch 15/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8416 -
categorical_accuracy: 0.7521 - val_loss: 0.8843 - val_categorical_accuracy:
0.6856
Epoch 16/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8207 -
categorical_accuracy: 0.7588 - val_loss: 0.8699 - val_categorical_accuracy:
0.6856
Epoch 17/60
```

```
38/38 [==============================] - 0s 2ms/step - loss: 0.8036 -
categorical_accuracy: 0.7546 - val_loss: 0.8551 - val_categorical_accuracy:
0.6890
Epoch 18/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7833 -
categorical_accuracy: 0.7730 - val_loss: 0.8405 - val_categorical_accuracy:
0.7224
Epoch 19/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7634 -
categorical_accuracy: 0.7747 - val_loss: 0.8278 - val_categorical_accuracy:
0.7324
Epoch 20/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7459 -
categorical_accuracy: 0.7814 - val_loss: 0.8152 - val_categorical_accuracy:
0.7324
Epoch 21/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7281 -
categorical_accuracy: 0.7873 - val_loss: 0.8029 - val_categorical_accuracy:
0.7425
Epoch 22/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7126 -
categorical_accuracy: 0.7990 - val_loss: 0.7919 - val_categorical_accuracy:
0.7324
Epoch 23/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7035 -
categorical_accuracy: 0.7948 - val_loss: 0.7806 - val_categorical_accuracy:
0.7391
Epoch 24/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6842 -
categorical_accuracy: 0.7948 - val_loss: 0.7709 - val_categorical_accuracy:
0.7324
Epoch 25/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6687 -
categorical_accuracy: 0.8023 - val_loss: 0.7611 - val_categorical_accuracy:
0.7291
Epoch 26/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6614 -
categorical_accuracy: 0.8032 - val_loss: 0.7529 - val_categorical_accuracy:
0.7358
Epoch 27/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6421 -
categorical_accuracy: 0.8007 - val_loss: 0.7443 - val_categorical_accuracy:
0.7358
Epoch 28/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6325 -
categorical_accuracy: 0.8040 - val_loss: 0.7369 - val_categorical_accuracy:
0.7391
Epoch 29/60
```

```
38/38 [==============================] - 0s 2ms/step - loss: 0.6203 -
categorical_accuracy: 0.8040 - val_loss: 0.7304 - val_categorical_accuracy:
0.7391
Epoch 30/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6012 -
categorical_accuracy: 0.8157 - val_loss: 0.7236 - val_categorical_accuracy:
0.7358
Epoch 31/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5947 -
categorical_accuracy: 0.8183 - val_loss: 0.7177 - val_categorical_accuracy:
0.7358
Epoch 32/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5898 -
categorical_accuracy: 0.8074 - val_loss: 0.7125 - val_categorical_accuracy:
0.7358
Epoch 33/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5765 -
categorical_accuracy: 0.8116 - val_loss: 0.7068 - val_categorical_accuracy:
0.7324
Epoch 34/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5667 -
categorical_accuracy: 0.8124 - val_loss: 0.7025 - val_categorical_accuracy:
0.7358
Epoch 35/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5567 -
categorical_accuracy: 0.8191 - val_loss: 0.6985 - val_categorical_accuracy:
0.7391
Epoch 36/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5494 -
categorical_accuracy: 0.8141 - val_loss: 0.6947 - val_categorical_accuracy:
0.7358
Epoch 37/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5395 -
categorical_accuracy: 0.8224 - val_loss: 0.6914 - val_categorical_accuracy:
0.7358
Epoch 38/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5349 -
categorical_accuracy: 0.8224 - val_loss: 0.6884 - val_categorical_accuracy:
0.7358
Epoch 39/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5249 -
categorical_accuracy: 0.8233 - val_loss: 0.6858 - val_categorical_accuracy:
0.7358
Epoch 40/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5221 -
categorical_accuracy: 0.8166 - val_loss: 0.6829 - val_categorical_accuracy:
0.7358
Epoch 41/60
```

```
38/38 [==============================] - 0s 2ms/step - loss: 0.5168 -
categorical_accuracy: 0.8258 - val_loss: 0.6812 - val_categorical_accuracy:
0.7391
Epoch 42/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5124 -
categorical_accuracy: 0.8183 - val_loss: 0.6795 - val_categorical_accuracy:
0.7425
Epoch 43/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5000 -
categorical_accuracy: 0.8224 - val_loss: 0.6781 - val_categorical_accuracy:
0.7391
Epoch 44/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5029 -
categorical_accuracy: 0.8199 - val_loss: 0.6762 - val_categorical_accuracy:
0.7425
Epoch 45/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4942 -
categorical_accuracy: 0.8233 - val_loss: 0.6747 - val_categorical_accuracy:
0.7425
Epoch 46/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4854 -
categorical_accuracy: 0.8266 - val_loss: 0.6736 - val_categorical_accuracy:
0.7425
Epoch 47/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4875 -
categorical_accuracy: 0.8233 - val_loss: 0.6729 - val_categorical_accuracy:
0.7425
Epoch 48/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4850 -
categorical_accuracy: 0.8191 - val_loss: 0.6720 - val_categorical_accuracy:
0.7391
Epoch 49/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4846 -
categorical_accuracy: 0.8199 - val_loss: 0.6717 - val_categorical_accuracy:
0.7425
Epoch 50/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4747 -
categorical_accuracy: 0.8233 - val_loss: 0.6714 - val_categorical_accuracy:
0.7425
Epoch 51/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4701 -
categorical_accuracy: 0.8275 - val_loss: 0.6711 - val_categorical_accuracy:
0.7425
Epoch 52/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4571 -
categorical_accuracy: 0.8333 - val_loss: 0.6714 - val_categorical_accuracy:
0.7425
Epoch 53/60
```

```
38/38 [==============================] - 0s 2ms/step - loss: 0.4556 -
categorical_accuracy: 0.8333 - val_loss: 0.6710 - val_categorical_accuracy:
0.7425
Epoch 54/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4471 -
categorical_accuracy: 0.8300 - val_loss: 0.6703 - val_categorical_accuracy:
0.7425
Epoch 55/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4541 -
categorical_accuracy: 0.8283 - val_loss: 0.6711 - val_categorical_accuracy:
0.7425
Epoch 56/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4500 -
categorical_accuracy: 0.8317 - val_loss: 0.6715 - val_categorical_accuracy:
0.7458
Epoch 57/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4487 -
categorical_accuracy: 0.8216 - val_loss: 0.6716 - val_categorical_accuracy:
0.7458
Epoch 58/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4381 -
categorical_accuracy: 0.8325 - val_loss: 0.6728 - val_categorical_accuracy:
0.7425
Epoch 59/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4403 -
categorical_accuracy: 0.8342 - val_loss: 0.6734 - val_categorical_accuracy:
0.7425
Epoch 60/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4298 -
categorical_accuracy: 0.8400 - val_loss: 0.6741 - val_categorical_accuracy:
0.7425
-----------------
Training on fold 3
Epoch 1/60
38/38 [==============================] - 0s 3ms/step - loss: 1.3332 -
categorical_accuracy: 0.4190 - val_loss: 1.3275 - val_categorical_accuracy:
0.3121
Epoch 2/60
38/38 [==============================] - 0s 2ms/step - loss: 1.2340 -
categorical_accuracy: 0.3431 - val_loss: 1.2353 - val_categorical_accuracy:
0.3121
Epoch 3/60
38/38 [==============================] - 0s 2ms/step - loss: 1.1558 -
categorical_accuracy: 0.3598 - val_loss: 1.1642 - val_categorical_accuracy:
0.3322
Epoch 4/60
38/38 [==============================] - 0s 2ms/step - loss: 1.0984 -
categorical_accuracy: 0.3874 - val_loss: 1.1106 - val_categorical_accuracy:
```

```
0.3758
Epoch 5/60
38/38 [==============================] - 0s 2ms/step - loss: 1.0537 -
categorical_accuracy: 0.4385 - val_loss: 1.0716 - val_categorical_accuracy:
0.4128
Epoch 6/60
38/38 [==============================] - 0s 2ms/step - loss: 1.0201 -
categorical_accuracy: 0.4887 - val_loss: 1.0417 - val_categorical_accuracy:
0.4396
Epoch 7/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9973 -
categorical_accuracy: 0.5138 - val_loss: 1.0203 - val_categorical_accuracy:
0.4698
Epoch 8/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9768 -
categorical_accuracy: 0.5615 - val_loss: 1.0034 - val_categorical_accuracy:
0.5268
Epoch 9/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9618 -
categorical_accuracy: 0.5925 - val_loss: 0.9893 - val_categorical_accuracy:
0.5906
Epoch 10/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9463 -
categorical_accuracy: 0.6285 - val_loss: 0.9769 - val_categorical_accuracy:
0.6342
Epoch 11/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9336 -
categorical_accuracy: 0.6703 - val_loss: 0.9641 - val_categorical_accuracy:
0.6376
Epoch 12/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9213 -
categorical_accuracy: 0.6879 - val_loss: 0.9533 - val_categorical_accuracy:
0.6644
Epoch 13/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9078 -
categorical_accuracy: 0.7029 - val_loss: 0.9420 - val_categorical_accuracy:
0.6644
Epoch 14/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8930 -
categorical_accuracy: 0.7456 - val_loss: 0.9302 - val_categorical_accuracy:
0.6779
Epoch 15/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8805 -
categorical_accuracy: 0.7456 - val_loss: 0.9191 - val_categorical_accuracy:
0.6812
Epoch 16/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8694 -
categorical_accuracy: 0.7506 - val_loss: 0.9077 - val_categorical_accuracy:
```

```
0.6913
Epoch 17/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8542 -
categorical_accuracy: 0.7665 - val_loss: 0.8959 - val_categorical_accuracy:
0.7148
Epoch 18/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8421 -
categorical_accuracy: 0.7782 - val_loss: 0.8848 - val_categorical_accuracy:
0.7013
Epoch 19/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8264 -
categorical_accuracy: 0.7766 - val_loss: 0.8731 - val_categorical_accuracy:
0.7114
Epoch 20/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8142 -
categorical_accuracy: 0.7833 - val_loss: 0.8618 - val_categorical_accuracy:
0.7114
Epoch 21/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7987 -
categorical_accuracy: 0.7858 - val_loss: 0.8501 - val_categorical_accuracy:
0.7248
Epoch 22/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7864 -
categorical_accuracy: 0.7866 - val_loss: 0.8388 - val_categorical_accuracy:
0.7315
Epoch 23/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7763 -
categorical_accuracy: 0.7874 - val_loss: 0.8275 - val_categorical_accuracy:
0.7315
Epoch 24/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7635 -
categorical_accuracy: 0.7833 - val_loss: 0.8166 - val_categorical_accuracy:
0.7248
Epoch 25/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7513 -
categorical_accuracy: 0.7841 - val_loss: 0.8061 - val_categorical_accuracy:
0.7282
Epoch 26/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7374 -
categorical_accuracy: 0.7883 - val_loss: 0.7953 - val_categorical_accuracy:
0.7282
Epoch 27/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7223 -
categorical_accuracy: 0.7908 - val_loss: 0.7849 - val_categorical_accuracy:
0.7349
Epoch 28/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7139 -
categorical_accuracy: 0.7866 - val_loss: 0.7744 - val_categorical_accuracy:
```

```
0.7383
Epoch 29/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6971 -
categorical_accuracy: 0.7958 - val_loss: 0.7646 - val_categorical_accuracy:
0.7383
Epoch 30/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6867 -
categorical_accuracy: 0.7958 - val_loss: 0.7550 - val_categorical_accuracy:
0.7517
Epoch 31/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6771 -
categorical_accuracy: 0.7933 - val_loss: 0.7461 - val_categorical_accuracy:
0.7517
Epoch 32/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6628 -
categorical_accuracy: 0.8017 - val_loss: 0.7367 - val_categorical_accuracy:
0.7517
Epoch 33/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6497 -
categorical_accuracy: 0.8092 - val_loss: 0.7283 - val_categorical_accuracy:
0.7450
Epoch 34/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6396 -
categorical_accuracy: 0.8142 - val_loss: 0.7203 - val_categorical_accuracy:
0.7416
Epoch 35/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6400 -
categorical_accuracy: 0.8000 - val_loss: 0.7124 - val_categorical_accuracy:
0.7550
Epoch 36/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6180 -
categorical_accuracy: 0.8159 - val_loss: 0.7055 - val_categorical_accuracy:
0.7450
Epoch 37/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6117 -
categorical_accuracy: 0.8142 - val_loss: 0.6983 - val_categorical_accuracy:
0.7450
Epoch 38/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6079 -
categorical_accuracy: 0.8084 - val_loss: 0.6916 - val_categorical_accuracy:
0.7517
Epoch 39/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5982 -
categorical_accuracy: 0.8134 - val_loss: 0.6858 - val_categorical_accuracy:
0.7450
Epoch 40/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5896 -
categorical_accuracy: 0.8134 - val_loss: 0.6800 - val_categorical_accuracy:
```

```
0.7450
Epoch 41/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5731 -
categorical_accuracy: 0.8192 - val_loss: 0.6747 - val_categorical_accuracy:
0.7450
Epoch 42/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5724 -
categorical_accuracy: 0.8192 - val_loss: 0.6693 - val_categorical_accuracy:
0.7416
Epoch 43/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5655 -
categorical_accuracy: 0.8134 - val_loss: 0.6645 - val_categorical_accuracy:
0.7450
Epoch 44/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5590 -
categorical_accuracy: 0.8167 - val_loss: 0.6594 - val_categorical_accuracy:
0.7483
Epoch 45/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5551 -
categorical_accuracy: 0.8151 - val_loss: 0.6550 - val_categorical_accuracy:
0.7483
Epoch 46/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5475 -
categorical_accuracy: 0.8167 - val_loss: 0.6511 - val_categorical_accuracy:
0.7517
Epoch 47/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5458 -
categorical_accuracy: 0.8134 - val_loss: 0.6474 - val_categorical_accuracy:
0.7550
Epoch 48/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5356 -
categorical_accuracy: 0.8167 - val_loss: 0.6436 - val_categorical_accuracy:
0.7517
Epoch 49/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5274 -
categorical_accuracy: 0.8326 - val_loss: 0.6407 - val_categorical_accuracy:
0.7550
Epoch 50/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5201 -
categorical_accuracy: 0.8218 - val_loss: 0.6372 - val_categorical_accuracy:
0.7550
Epoch 51/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5210 -
categorical_accuracy: 0.8201 - val_loss: 0.6344 - val_categorical_accuracy:
0.7517
Epoch 52/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5078 -
categorical_accuracy: 0.8218 - val_loss: 0.6317 - val_categorical_accuracy:
```

```
0.7584
Epoch 53/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4997 -
categorical_accuracy: 0.8285 - val_loss: 0.6297 - val_categorical_accuracy:
0.7550
Epoch 54/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5016 -
categorical_accuracy: 0.8268 - val_loss: 0.6269 - val_categorical_accuracy:
0.7550
Epoch 55/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4956 -
categorical_accuracy: 0.8293 - val_loss: 0.6251 - val_categorical_accuracy:
0.7517
Epoch 56/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4913 -
categorical_accuracy: 0.8251 - val_loss: 0.6238 - val_categorical_accuracy:
0.7550
Epoch 57/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4847 -
categorical_accuracy: 0.8293 - val_loss: 0.6225 - val_categorical_accuracy:
0.7517
Epoch 58/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4815 -
categorical_accuracy: 0.8326 - val_loss: 0.6207 - val_categorical_accuracy:
0.7584
Epoch 59/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4792 -
categorical_accuracy: 0.8243 - val_loss: 0.6198 - val_categorical_accuracy:
0.7584
Epoch 60/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4722 -
categorical_accuracy: 0.8360 - val_loss: 0.6186 - val_categorical_accuracy:
0.7584
------------------
Training on fold 4
Epoch 1/60
38/38 [==============================] - 0s 3ms/step - loss: 1.1388 -
categorical_accuracy: 0.4213 - val_loss: 1.1034 - val_categorical_accuracy:
0.3255
Epoch 2/60
38/38 [==============================] - 0s 2ms/step - loss: 1.0759 -
categorical_accuracy: 0.3874 - val_loss: 1.0548 - val_categorical_accuracy:
0.4060
Epoch 3/60
38/38 [==============================] - 0s 2ms/step - loss: 1.0385 -
categorical_accuracy: 0.4243 - val_loss: 1.0243 - val_categorical_accuracy:
0.4396
Epoch 4/60
```

```
38/38 [==============================] - 0s 2ms/step - loss: 1.0098 -
categorical_accuracy: 0.4971 - val_loss: 1.0013 - val_categorical_accuracy:
0.5235
Epoch 5/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9869 -
categorical_accuracy: 0.5724 - val_loss: 0.9806 - val_categorical_accuracy:
0.5738
Epoch 6/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9666 -
categorical_accuracy: 0.6343 - val_loss: 0.9615 - val_categorical_accuracy:
0.6376
Epoch 7/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9444 -
categorical_accuracy: 0.6577 - val_loss: 0.9436 - val_categorical_accuracy:
0.6510
Epoch 8/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9277 -
categorical_accuracy: 0.6812 - val_loss: 0.9261 - val_categorical_accuracy:
0.6913
Epoch 9/60
38/38 [==============================] - 0s 2ms/step - loss: 0.9084 -
categorical_accuracy: 0.7280 - val_loss: 0.9091 - val_categorical_accuracy:
0.7081
Epoch 10/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8891 -
categorical_accuracy: 0.7264 - val_loss: 0.8925 - val_categorical_accuracy:
0.6913
Epoch 11/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8712 -
categorical_accuracy: 0.7423 - val_loss: 0.8770 - val_categorical_accuracy:
0.6946
Epoch 12/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8539 -
categorical_accuracy: 0.7356 - val_loss: 0.8618 - val_categorical_accuracy:
0.7047
Epoch 13/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8388 -
categorical_accuracy: 0.7548 - val_loss: 0.8466 - val_categorical_accuracy:
0.7248
Epoch 14/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8223 -
categorical_accuracy: 0.7506 - val_loss: 0.8320 - val_categorical_accuracy:
0.7248
Epoch 15/60
38/38 [==============================] - 0s 2ms/step - loss: 0.8056 -
categorical_accuracy: 0.7707 - val_loss: 0.8179 - val_categorical_accuracy:
0.7450
Epoch 16/60
```

```
38/38 [==============================] - 0s 2ms/step - loss: 0.7906 -
categorical_accuracy: 0.7640 - val_loss: 0.8046 - val_categorical_accuracy:
0.7383
Epoch 17/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7790 -
categorical_accuracy: 0.7623 - val_loss: 0.7922 - val_categorical_accuracy:
0.7416
Epoch 18/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7630 -
categorical_accuracy: 0.7741 - val_loss: 0.7796 - val_categorical_accuracy:
0.7450
Epoch 19/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7474 -
categorical_accuracy: 0.7724 - val_loss: 0.7670 - val_categorical_accuracy:
0.7584
Epoch 20/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7349 -
categorical_accuracy: 0.7816 - val_loss: 0.7562 - val_categorical_accuracy:
0.7584
Epoch 21/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7254 -
categorical_accuracy: 0.7849 - val_loss: 0.7455 - val_categorical_accuracy:
0.7517
Epoch 22/60
38/38 [==============================] - 0s 2ms/step - loss: 0.7062 -
categorical_accuracy: 0.7925 - val_loss: 0.7344 - val_categorical_accuracy:
0.7584
Epoch 23/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6965 -
categorical_accuracy: 0.7866 - val_loss: 0.7246 - val_categorical_accuracy:
0.7584
Epoch 24/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6844 -
categorical_accuracy: 0.7925 - val_loss: 0.7157 - val_categorical_accuracy:
0.7517
Epoch 25/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6778 -
categorical_accuracy: 0.7975 - val_loss: 0.7070 - val_categorical_accuracy:
0.7550
Epoch 26/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6635 -
categorical_accuracy: 0.7967 - val_loss: 0.6982 - val_categorical_accuracy:
0.7483
Epoch 27/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6560 -
categorical_accuracy: 0.7950 - val_loss: 0.6904 - val_categorical_accuracy:
0.7483
Epoch 28/60
```

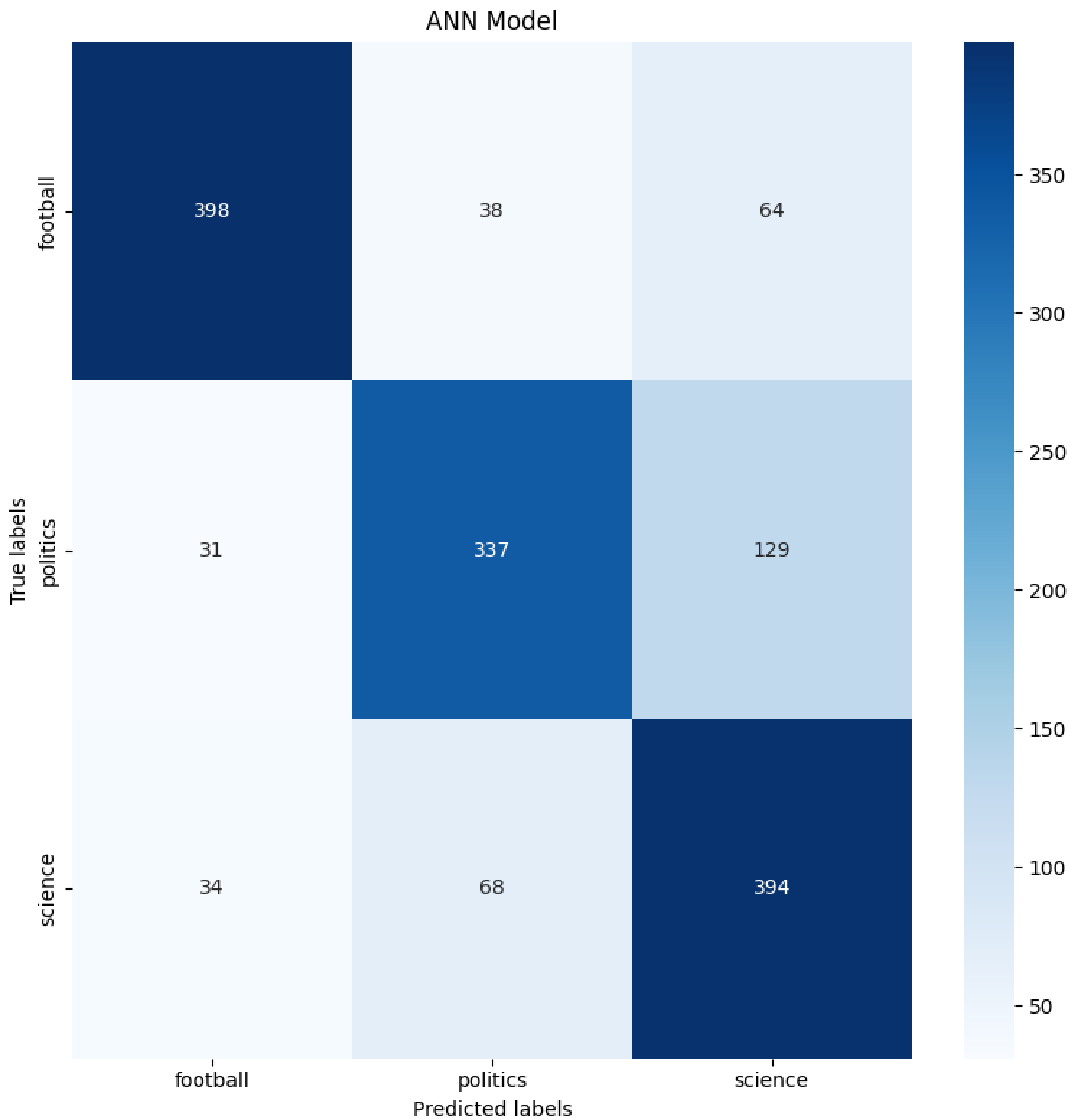
```
38/38 [==============================] - 0s 2ms/step - loss: 0.6459 -
categorical_accuracy: 0.7941 - val_loss: 0.6828 - val_categorical_accuracy:
0.7517
Epoch 29/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6315 -
categorical_accuracy: 0.7933 - val_loss: 0.6759 - val_categorical_accuracy:
0.7550
Epoch 30/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6268 -
categorical_accuracy: 0.7925 - val_loss: 0.6689 - val_categorical_accuracy:
0.7517
Epoch 31/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6185 -
categorical_accuracy: 0.7967 - val_loss: 0.6631 - val_categorical_accuracy:
0.7550
Epoch 32/60
38/38 [==============================] - 0s 2ms/step - loss: 0.6040 -
categorical_accuracy: 0.7967 - val_loss: 0.6573 - val_categorical_accuracy:
0.7550
Epoch 33/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5946 -
categorical_accuracy: 0.8033 - val_loss: 0.6520 - val_categorical_accuracy:
0.7550
Epoch 34/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5930 -
categorical_accuracy: 0.7975 - val_loss: 0.6461 - val_categorical_accuracy:
0.7550
Epoch 35/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5755 -
categorical_accuracy: 0.8100 - val_loss: 0.6415 - val_categorical_accuracy:
0.7550
Epoch 36/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5720 -
categorical_accuracy: 0.8025 - val_loss: 0.6367 - val_categorical_accuracy:
0.7584
Epoch 37/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5646 -
categorical_accuracy: 0.7992 - val_loss: 0.6325 - val_categorical_accuracy:
0.7584
Epoch 38/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5607 -
categorical_accuracy: 0.7967 - val_loss: 0.6280 - val_categorical_accuracy:
0.7584
Epoch 39/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5508 -
categorical_accuracy: 0.8008 - val_loss: 0.6243 - val_categorical_accuracy:
0.7584
Epoch 40/60
```

```
38/38 [==============================] - 0s 2ms/step - loss: 0.5478 -
categorical_accuracy: 0.8059 - val_loss: 0.6208 - val_categorical_accuracy:
0.7584
Epoch 41/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5479 -
categorical_accuracy: 0.8075 - val_loss: 0.6178 - val_categorical_accuracy:
0.7550
Epoch 42/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5314 -
categorical_accuracy: 0.8109 - val_loss: 0.6144 - val_categorical_accuracy:
0.7550
Epoch 43/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5309 -
categorical_accuracy: 0.8126 - val_loss: 0.6122 - val_categorical_accuracy:
0.7517
Epoch 44/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5283 -
categorical_accuracy: 0.8067 - val_loss: 0.6093 - val_categorical_accuracy:
0.7550
Epoch 45/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5190 -
categorical_accuracy: 0.8084 - val_loss: 0.6081 - val_categorical_accuracy:
0.7483
Epoch 46/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5167 -
categorical_accuracy: 0.8042 - val_loss: 0.6058 - val_categorical_accuracy:
0.7483
Epoch 47/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5115 -
categorical_accuracy: 0.8151 - val_loss: 0.6036 - val_categorical_accuracy:
0.7483
Epoch 48/60
38/38 [==============================] - 0s 2ms/step - loss: 0.5056 -
categorical_accuracy: 0.8176 - val_loss: 0.6017 - val_categorical_accuracy:
0.7450
Epoch 49/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4990 -
categorical_accuracy: 0.8176 - val_loss: 0.6003 - val_categorical_accuracy:
0.7483
Epoch 50/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4905 -
categorical_accuracy: 0.8234 - val_loss: 0.5987 - val_categorical_accuracy:
0.7517
Epoch 51/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4924 -
categorical_accuracy: 0.8209 - val_loss: 0.5980 - val_categorical_accuracy:
0.7483
Epoch 52/60
```

```
38/38 [==============================] - 0s 2ms/step - loss: 0.4909 -
categorical_accuracy: 0.8100 - val_loss: 0.5966 - val_categorical_accuracy:
0.7517
Epoch 53/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4844 -
categorical_accuracy: 0.8167 - val_loss: 0.5953 - val_categorical_accuracy:
0.7483
Epoch 54/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4864 -
categorical_accuracy: 0.8218 - val_loss: 0.5940 - val_categorical_accuracy:
0.7483
Epoch 55/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4769 -
categorical_accuracy: 0.8126 - val_loss: 0.5936 - val_categorical_accuracy:
0.7450
Epoch 56/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4794 -
categorical_accuracy: 0.8201 - val_loss: 0.5924 - val_categorical_accuracy:
0.7483
Epoch 57/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4721 -
categorical_accuracy: 0.8201 - val_loss: 0.5923 - val_categorical_accuracy:
0.7450
Epoch 58/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4711 -
categorical_accuracy: 0.8259 - val_loss: 0.5919 - val_categorical_accuracy:
0.7450
Epoch 59/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4659 -
categorical_accuracy: 0.8268 - val_loss: 0.5917 - val_categorical_accuracy:
0.7450
Epoch 60/60
38/38 [==============================] - 0s 2ms/step - loss: 0.4564 -
categorical_accuracy: 0.8301 - val_loss: 0.5914 - val_categorical_accuracy:
0.7450
10/10 [==============================] - 0s 718us/step
10/10 [==============================] - 0s 846us/step
10/10 [==============================] - 0s 704us/step
10/10 [==============================] - 0s 860us/step
10/10 [==============================] - 0s 687us/step
```

3.2 Confusion Matrix

[ ]:
```python
plot_confusion_matrix(Y, pred_ann, labels=list(vals), logscale=False,␣
 ↪title="ANN Model")
```




```
[ ]: array([[398,  38,  64],
            [ 31, 337, 129],
            [ 34,  68, 394]])
```

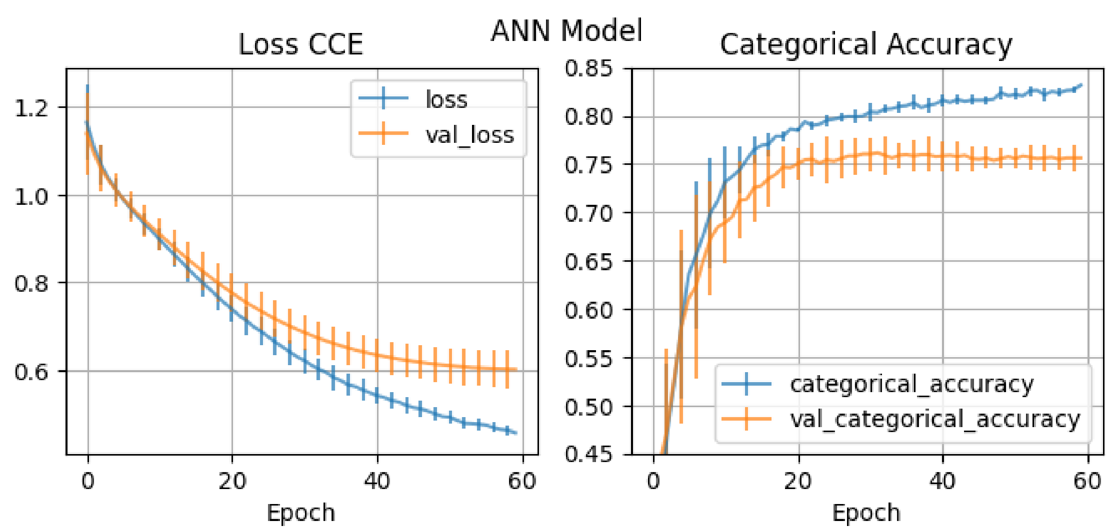
3.3 Training History

```
[ ]: def get_mean_std(histories, key):
         vals = np.array([h.history[key] for h in histories])
         return np.mean(vals, axis=0), np.std(vals, axis=0)

     def plot_key(ax, x_vals, histories, key, errorevery=1):
```

```
    mean, std = get_mean_std(histories, key)
    ax.errorbar(x_vals, mean, std, label=key, errorevery=errorevery, alpha=0.7)


f, ax = plt.subplots(ncols=2)
f.set_size_inches(8,3)
epochs = np.arange(len(histories_ann[0].history["loss"]))
every_n = 2
plot_key(ax[0], epochs, histories_ann, "loss", errorevery=every_n)
plot_key(ax[0], epochs, histories_ann, "val_loss", errorevery=every_n)
plot_key(ax[1], epochs, histories_ann, "categorical_accuracy",␣
 ↪errorevery=every_n)
plot_key(ax[1], epochs, histories_ann, "val_categorical_accuracy",␣
 ↪errorevery=every_n)
ax[0].set_title("Loss CCE")
ax[0].set_xlabel("Epoch")
ax[0].legend()
ax[0].grid()
plt.suptitle("ANN Model")
ax[1].set_xlabel("Epoch")
ax[1].set_title("Categorical Accuracy")
ax[1].grid()
ax[1].set_ylim(0.45, 0.85)
ax[1].legend();
```



4 CNN Model:

4.1 Define and Train

[ ]:
```
def cnn_model(embedding_size=32, input_size=X.shape[1], output_size=Y.shape[1],␣
↪optimizer="adam", loss="categorical_crossentropy",
              metrics=[keras.metrics.CategoricalAccuracy()]):

    model = keras.models.Sequential([
                # This initial dense layer serves as an embedding layer
                keras.layers.Input(input_size),
                keras.layers.Dropout(0.1),
                keras.layers.Dense(embedding_size),
                keras.layers.Reshape((embedding_size, 1)),
                # keras.layers.Reshape((input_size, 1)),

                # Now begin the convolutional part
                keras.layers.Conv1D(filters=embedding_size, kernel_size=3,␣
↪strides=1, padding="same", activation="relu"),
                keras.layers.MaxPool1D(pool_size=2),
                keras.layers.Conv1D(filters=embedding_size/4, kernel_size=3,␣
↪strides=1, padding="same", activation="relu"),
                keras.layers.MaxPool1D(pool_size=2),
                keras.layers.Flatten(),
                keras.layers.Dense(output_size, activation="softmax")
            ])

    model.build(input_size)
    model.compile(optimizer=optimizer, loss=loss, metrics=metrics)

    return model

cnn_test = cnn_model()
cnn_test.summary()
```

```
Model: "sequential_6"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dropout_6 (Dropout)         (None, 300)               0

 dense_12 (Dense)            (None, 32)                9632

 reshape (Reshape)           (None, 32, 1)             0

 conv1d (Conv1D)             (None, 32, 32)            128

 max_pooling1d (MaxPooling1  (None, 16, 32)            0
```

```
 D)

 conv1d_1 (Conv1D)           (None, 16, 8)             776

 max_pooling1d_1 (MaxPoolin  (None, 8, 8)              0
 g1D)

 flatten (Flatten)           (None, 64)                0

 dense_13 (Dense)            (None, 3)                 195

=================================================================
Total params: 10731 (41.92 KB)
Trainable params: 10731 (41.92 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dropout_6 (Dropout)         (None, 300)               0

 dense_12 (Dense)            (None, 32)                9632

 reshape (Reshape)           (None, 32, 1)             0

 conv1d (Conv1D)             (None, 32, 32)            128

 max_pooling1d (MaxPooling1  (None, 16, 32)            0
 D)

 conv1d_1 (Conv1D)           (None, 16, 8)             776

 max_pooling1d_1 (MaxPoolin  (None, 8, 8)              0
 g1D)

 flatten (Flatten)           (None, 64)                0

 dense_13 (Dense)            (None, 3)                 195

=================================================================
Total params: 10731 (41.92 KB)
Trainable params: 10731 (41.92 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

[ ]:
```
models_cnn, histories_cnn = train_on_folds(cnn_model, splits, X, Y, epochs=7)
pred_cnn = pred_on_folds(models_cnn, splits, X, Y)
```

```
-----------------
Training on fold 0
Epoch 1/7
38/38 [==============================] - 1s 4ms/step - loss: 1.0716 -
categorical_accuracy: 0.4806 - val_loss: 1.0217 - val_categorical_accuracy:
0.4716
Epoch 2/7
38/38 [==============================] - 0s 2ms/step - loss: 0.9793 -
categorical_accuracy: 0.5637 - val_loss: 0.8982 - val_categorical_accuracy:
0.6388
Epoch 3/7
38/38 [==============================] - 0s 2ms/step - loss: 0.7943 -
categorical_accuracy: 0.7178 - val_loss: 0.6874 - val_categorical_accuracy:
0.7090
Epoch 4/7
38/38 [==============================] - 0s 2ms/step - loss: 0.5879 -
categorical_accuracy: 0.7789 - val_loss: 0.5886 - val_categorical_accuracy:
0.7525
Epoch 5/7
38/38 [==============================] - 0s 2ms/step - loss: 0.4864 -
categorical_accuracy: 0.8032 - val_loss: 0.5689 - val_categorical_accuracy:
0.7692
Epoch 6/7
38/38 [==============================] - 0s 2ms/step - loss: 0.4524 -
categorical_accuracy: 0.8157 - val_loss: 0.5634 - val_categorical_accuracy:
0.7692
Epoch 7/7
38/38 [==============================] - 0s 2ms/step - loss: 0.4110 -
categorical_accuracy: 0.8291 - val_loss: 0.5798 - val_categorical_accuracy:
0.7692
-----------------
Training on fold 1
Epoch 1/7
38/38 [==============================] - 1s 5ms/step - loss: 1.0837 -
categorical_accuracy: 0.4702 - val_loss: 1.0573 - val_categorical_accuracy:
0.4415
Epoch 2/7
38/38 [==============================] - 0s 2ms/step - loss: 1.0045 -
categorical_accuracy: 0.4749 - val_loss: 0.9365 - val_categorical_accuracy:
0.6154
Epoch 3/7
38/38 [==============================] - 0s 2ms/step - loss: 0.8240 -
categorical_accuracy: 0.6742 - val_loss: 0.7435 - val_categorical_accuracy:
0.7124
Epoch 4/7
38/38 [==============================] - 0s 2ms/step - loss: 0.5955 -
categorical_accuracy: 0.7663 - val_loss: 0.6195 - val_categorical_accuracy:
0.7625
```

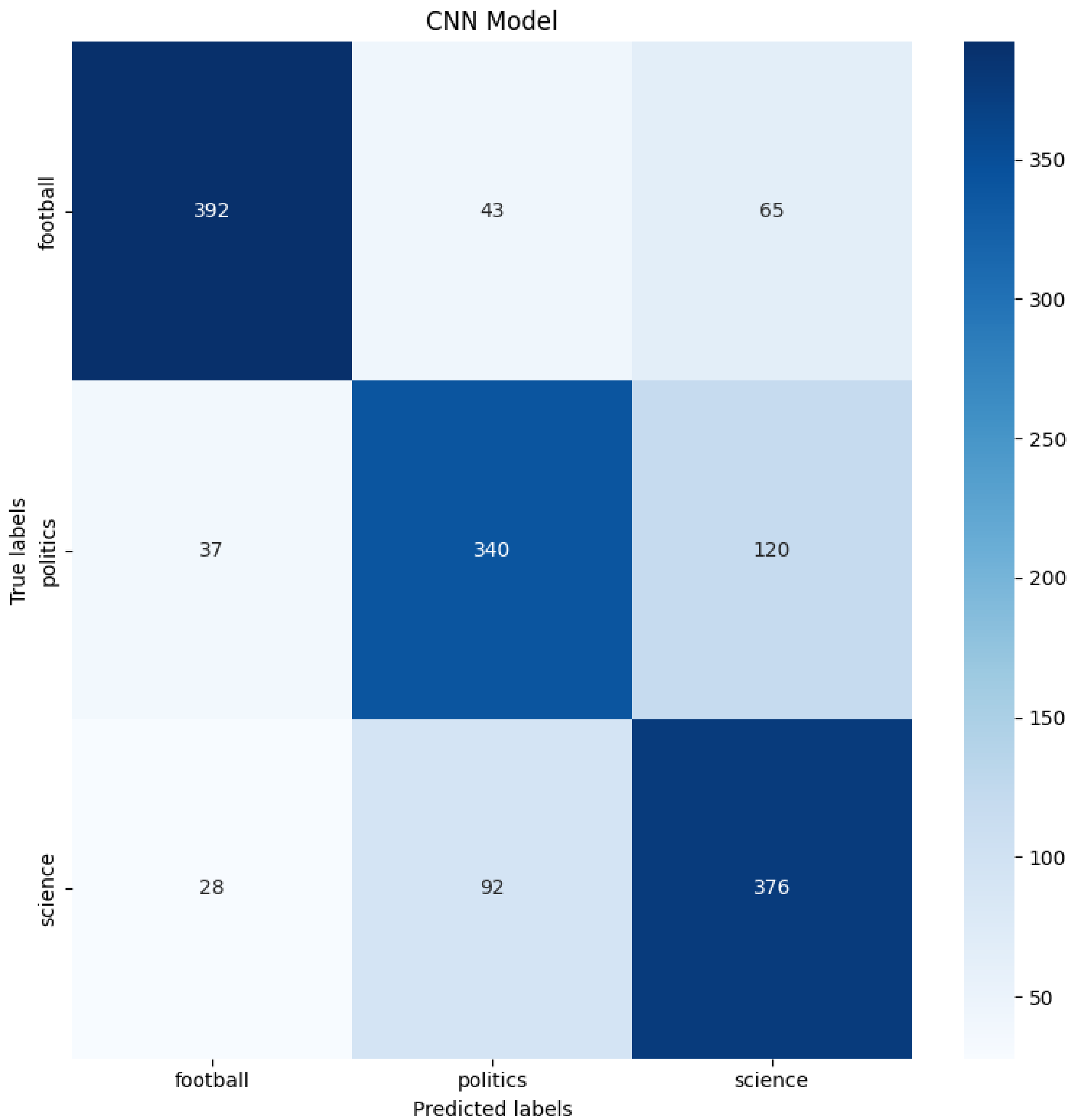
```
Epoch 5/7
38/38 [==============================] - 0s 2ms/step - loss: 0.4830 -
categorical_accuracy: 0.8090 - val_loss: 0.6051 - val_categorical_accuracy:
0.7458
Epoch 6/7
38/38 [==============================] - 0s 2ms/step - loss: 0.4275 -
categorical_accuracy: 0.8141 - val_loss: 0.6028 - val_categorical_accuracy:
0.7726
Epoch 7/7
38/38 [==============================] - 0s 2ms/step - loss: 0.3842 -
categorical_accuracy: 0.8350 - val_loss: 0.6294 - val_categorical_accuracy:
0.7559
-----------------
Training on fold 2
Epoch 1/7
38/38 [==============================] - 1s 4ms/step - loss: 1.0833 -
categorical_accuracy: 0.5050 - val_loss: 1.0613 - val_categorical_accuracy:
0.4749
Epoch 2/7
38/38 [==============================] - 0s 2ms/step - loss: 1.0083 -
categorical_accuracy: 0.5553 - val_loss: 0.9609 - val_categorical_accuracy:
0.5786
Epoch 3/7
38/38 [==============================] - 0s 2ms/step - loss: 0.8379 -
categorical_accuracy: 0.6943 - val_loss: 0.8075 - val_categorical_accuracy:
0.6722
Epoch 4/7
38/38 [==============================] - 0s 2ms/step - loss: 0.6386 -
categorical_accuracy: 0.7714 - val_loss: 0.7182 - val_categorical_accuracy:
0.6923
Epoch 5/7
38/38 [==============================] - 0s 2ms/step - loss: 0.5051 -
categorical_accuracy: 0.7906 - val_loss: 0.7082 - val_categorical_accuracy:
0.7157
Epoch 6/7
38/38 [==============================] - 0s 2ms/step - loss: 0.4498 -
categorical_accuracy: 0.7998 - val_loss: 0.7380 - val_categorical_accuracy:
0.6957
Epoch 7/7
38/38 [==============================] - 0s 2ms/step - loss: 0.3984 -
categorical_accuracy: 0.8308 - val_loss: 0.7740 - val_categorical_accuracy:
0.6957
-----------------
Training on fold 3
Epoch 1/7
38/38 [==============================] - 1s 4ms/step - loss: 1.0665 -
categorical_accuracy: 0.5027 - val_loss: 1.0495 - val_categorical_accuracy:
0.5134
```

```
Epoch 2/7
38/38 [==============================] - 0s 2ms/step - loss: 0.9707 -
categorical_accuracy: 0.6109 - val_loss: 0.9466 - val_categorical_accuracy:
0.6376
Epoch 3/7
38/38 [==============================] - 0s 2ms/step - loss: 0.7999 -
categorical_accuracy: 0.7180 - val_loss: 0.7459 - val_categorical_accuracy:
0.6946
Epoch 4/7
38/38 [==============================] - 0s 2ms/step - loss: 0.5918 -
categorical_accuracy: 0.7682 - val_loss: 0.6167 - val_categorical_accuracy:
0.7517
Epoch 5/7
38/38 [==============================] - 0s 2ms/step - loss: 0.4852 -
categorical_accuracy: 0.7967 - val_loss: 0.6034 - val_categorical_accuracy:
0.7550
Epoch 6/7
38/38 [==============================] - 0s 2ms/step - loss: 0.4235 -
categorical_accuracy: 0.8259 - val_loss: 0.6121 - val_categorical_accuracy:
0.7483
Epoch 7/7
38/38 [==============================] - 0s 2ms/step - loss: 0.3857 -
categorical_accuracy: 0.8435 - val_loss: 0.6360 - val_categorical_accuracy:
0.7550
-----------------
Training on fold 4
Epoch 1/7
38/38 [==============================] - 1s 4ms/step - loss: 1.0719 -
categorical_accuracy: 0.5111 - val_loss: 1.0312 - val_categorical_accuracy:
0.4899
Epoch 2/7
38/38 [==============================] - 0s 2ms/step - loss: 0.9963 -
categorical_accuracy: 0.5138 - val_loss: 0.9474 - val_categorical_accuracy:
0.5738
Epoch 3/7
38/38 [==============================] - 0s 2ms/step - loss: 0.8770 -
categorical_accuracy: 0.6577 - val_loss: 0.8310 - val_categorical_accuracy:
0.6779
Epoch 4/7
38/38 [==============================] - 0s 2ms/step - loss: 0.7003 -
categorical_accuracy: 0.7381 - val_loss: 0.6889 - val_categorical_accuracy:
0.7047
Epoch 5/7
38/38 [==============================] - 0s 2ms/step - loss: 0.5488 -
categorical_accuracy: 0.7816 - val_loss: 0.6197 - val_categorical_accuracy:
0.7114
Epoch 6/7
38/38 [==============================] - 0s 2ms/step - loss: 0.4576 -
```

```
categorical_accuracy: 0.8117 - val_loss: 0.6251 - val_categorical_accuracy:
0.7383
Epoch 7/7
38/38 [==============================] - 0s 2ms/step - loss: 0.3956 -
categorical_accuracy: 0.8377 - val_loss: 0.6272 - val_categorical_accuracy:
0.7349
10/10 [==============================] - 0s 971us/step
10/10 [==============================] - 0s 971us/step
10/10 [==============================] - 0s 963us/step
10/10 [==============================] - 0s 948us/step
10/10 [==============================] - 0s 966us/step
```

4.2 Confusion Matrix

[ ]:
```
plot_confusion_matrix(Y, pred_cnn, labels=list(vals), logscale=False, title="CNN Model")
```

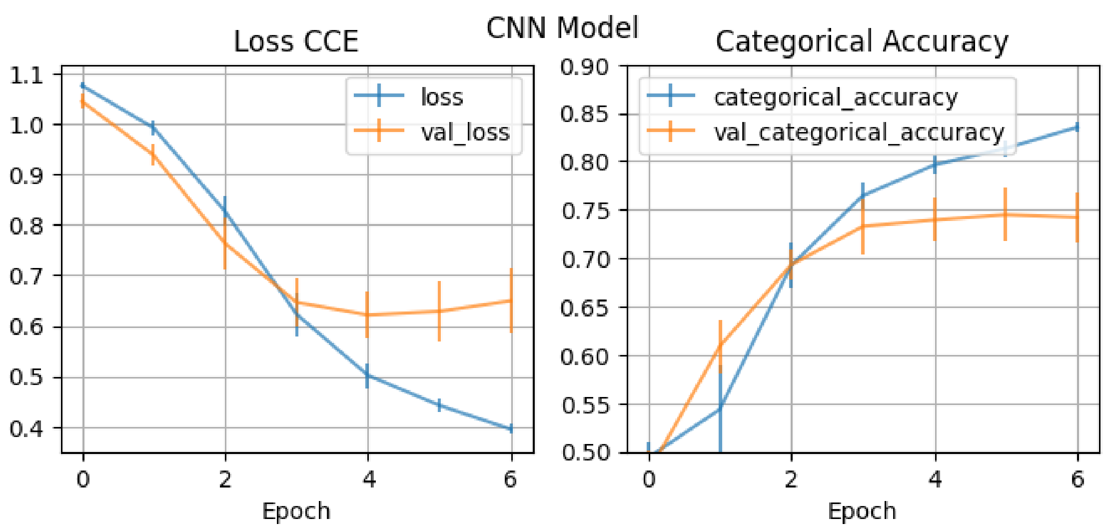



```
[ ]: array([[392,  43,  65],
            [ 37, 340, 120],
            [ 28,  92, 376]])
```

4.3 Training History

```
[ ]: f, ax = plt.subplots(ncols=2)
     f.set_size_inches(8,3)
     epochs = np.arange(len(histories_cnn[0].history["loss"]))
     plot_key(ax[0], epochs, histories_cnn, "loss")
     plot_key(ax[0], epochs, histories_cnn, "val_loss")
```

```
plot_key(ax[1], epochs, histories_cnn, "categorical_accuracy")
plot_key(ax[1], epochs, histories_cnn, "val_categorical_accuracy")
ax[0].set_title("Loss CCE")
ax[0].set_xlabel("Epoch")
ax[0].legend()
ax[0].grid()
ax[1].set_xlabel("Epoch")
ax[1].set_title("Categorical Accuracy")
ax[1].set_ylim(0.5, 0.9)
plt.suptitle("CNN Model")
ax[1].grid()
ax[1].legend();
```



5 LSTM Model

5.1 Define and Train

For the LSTM, I'm going to try treating the binary vector as a time series:

```
[ ]: num_words = X.shape[1]
X_seq = []
for i in range(X.shape[0]):
    X_seq += [np.nonzero(X[i])[0]+2]

max_len = max([len(x) for x in X_seq])

for i in range(X.shape[0]):
    X_seq[i] = np.pad(X_seq[i], (0, max_len - len(X_seq[i])), 'constant',␣
 ↪constant_values=(0, 0))
```

```
X_seq = np.array(X_seq)
X_seq, X_seq.shape
```

[ ]:
```
(array([[  9,  30,  95, …,   0,   0,   0],
        [ 63, 117, 165, …,   0,   0,   0],
        [231,   0,   0, …,   0,   0,   0],
        …,
        [ 79,  81, 178, …,   0,   0,   0],
        [ 82,  93, 111, …,   0,   0,   0],
        [235,   0,   0, …,   0,   0,   0]]),
 (1493, 15))
```

Because there isn't really a temporal relationship here, and coming before is the same as coming after, I will use a bidirectional LSTM.

[ ]:
```
def lstm_model(lstm_size=8, input_size=X.shape[1], output_size=Y.shape[1],␣
 ↪optimizer="adam", loss="categorical_crossentropy",
              metrics=[keras.metrics.CategoricalAccuracy()]):

    model = keras.models.Sequential([
                # keras.layers.Input((max_len, 1))
                keras.layers.Embedding(input_dim=num_words+2,␣
 ↪output_dim=lstm_size, mask_zero=True),
                keras.layers.Reshape((max_len, lstm_size)),
                keras.layers.Dropout(0.1),
                # keras.layers.Dense(lstm_size),,
                keras.layers.Bidirectional(keras.layers.LSTM(lstm_size,␣
 ↪return_sequences=False)),
                # keras.layers.Dense(lstm_size/4, activation="relu"),
                keras.layers.Dense(output_size, activation="softmax")
            ])

    model.build(input_size)
    model.compile(optimizer=optimizer, loss=loss, metrics=metrics)

    return model

lstm_test = lstm_model()
lstm_test.summary()
```

```
Model: "sequential_12"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, None, 8)           2416


 reshape_6 (Reshape)         (None, 15, 8)             0
```

```
 dropout_12 (Dropout)        (None, 15, 8)             0

 bidirectional (Bidirection  (None, 16)                1088
 al)

 dense_24 (Dense)            (None, 3)                 51

=================================================================
Total params: 3555 (13.89 KB)
Trainable params: 3555 (13.89 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, None, 8)           2416

 reshape_6 (Reshape)         (None, 15, 8)             0

 dropout_12 (Dropout)        (None, 15, 8)             0

 bidirectional (Bidirection  (None, 16)                1088
 al)

 dense_24 (Dense)            (None, 3)                 51

=================================================================
Total params: 3555 (13.89 KB)
Trainable params: 3555 (13.89 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```


[ ]:
```
models_lstm, histories_lstm = train_on_folds(lstm_model, splits, X_seq, Y,␣
 ↪epochs=15)
pred_lstm = pred_on_folds(models_lstm, splits, X_seq, Y)
```

```
-----------------
Training on fold 0
Epoch 1/15
38/38 [==============================] - 3s 43ms/step - loss: 1.0935 -
categorical_accuracy: 0.3978 - val_loss: 1.0831 - val_categorical_accuracy:
0.4247
Epoch 2/15
38/38 [==============================] - 1s 18ms/step - loss: 1.0668 -
categorical_accuracy: 0.4581 - val_loss: 1.0270 - val_categorical_accuracy:
0.5886
Epoch 3/15
```

```
38/38 [==============================] - 1s 15ms/step - loss: 0.9870 -
categorical_accuracy: 0.6106 - val_loss: 0.8895 - val_categorical_accuracy:
0.6120
Epoch 4/15
38/38 [==============================] - 1s 16ms/step - loss: 0.7888 -
categorical_accuracy: 0.6415 - val_loss: 0.7094 - val_categorical_accuracy:
0.7023
Epoch 5/15
38/38 [==============================] - 0s 8ms/step - loss: 0.6717 -
categorical_accuracy: 0.7320 - val_loss: 0.6699 - val_categorical_accuracy:
0.7124
Epoch 6/15
38/38 [==============================] - 0s 6ms/step - loss: 0.6180 -
categorical_accuracy: 0.7521 - val_loss: 0.6682 - val_categorical_accuracy:
0.7090
Epoch 7/15
38/38 [==============================] - 0s 6ms/step - loss: 0.5900 -
categorical_accuracy: 0.7605 - val_loss: 0.7014 - val_categorical_accuracy:
0.6756
Epoch 8/15
38/38 [==============================] - 0s 8ms/step - loss: 0.5660 -
categorical_accuracy: 0.7889 - val_loss: 0.6597 - val_categorical_accuracy:
0.7224
Epoch 9/15
38/38 [==============================] - 0s 5ms/step - loss: 0.5456 -
categorical_accuracy: 0.7806 - val_loss: 0.6392 - val_categorical_accuracy:
0.7358
Epoch 10/15
38/38 [==============================] - 0s 5ms/step - loss: 0.5178 -
categorical_accuracy: 0.7948 - val_loss: 0.6418 - val_categorical_accuracy:
0.7090
Epoch 11/15
38/38 [==============================] - 0s 4ms/step - loss: 0.4967 -
categorical_accuracy: 0.8049 - val_loss: 0.6238 - val_categorical_accuracy:
0.7492
Epoch 12/15
38/38 [==============================] - 0s 5ms/step - loss: 0.4681 -
categorical_accuracy: 0.8241 - val_loss: 0.6163 - val_categorical_accuracy:
0.7391
Epoch 13/15
38/38 [==============================] - 0s 5ms/step - loss: 0.4563 -
categorical_accuracy: 0.8367 - val_loss: 0.6083 - val_categorical_accuracy:
0.7559
Epoch 14/15
38/38 [==============================] - 0s 5ms/step - loss: 0.4330 -
categorical_accuracy: 0.8375 - val_loss: 0.6091 - val_categorical_accuracy:
0.7559
Epoch 15/15
```

```
38/38 [==============================] - 0s 6ms/step - loss: 0.4343 -
categorical_accuracy: 0.8384 - val_loss: 0.6060 - val_categorical_accuracy:
0.7492
-----------------
Training on fold 1
Epoch 1/15
38/38 [==============================] - 3s 40ms/step - loss: 1.0950 -
categorical_accuracy: 0.4441 - val_loss: 1.0899 - val_categorical_accuracy:
0.4314
Epoch 2/15
38/38 [==============================] - 0s 12ms/step - loss: 1.0795 -
categorical_accuracy: 0.5394 - val_loss: 1.0634 - val_categorical_accuracy:
0.5719
Epoch 3/15
38/38 [==============================] - 0s 11ms/step - loss: 1.0282 -
categorical_accuracy: 0.6139 - val_loss: 0.9772 - val_categorical_accuracy:
0.5987
Epoch 4/15
38/38 [==============================] - 0s 9ms/step - loss: 0.8711 -
categorical_accuracy: 0.6549 - val_loss: 0.7833 - val_categorical_accuracy:
0.6555
Epoch 5/15
38/38 [==============================] - 0s 7ms/step - loss: 0.6753 -
categorical_accuracy: 0.6943 - val_loss: 0.7222 - val_categorical_accuracy:
0.6890
Epoch 6/15
38/38 [==============================] - 0s 6ms/step - loss: 0.5894 -
categorical_accuracy: 0.7764 - val_loss: 0.6851 - val_categorical_accuracy:
0.7057
Epoch 7/15
38/38 [==============================] - 0s 6ms/step - loss: 0.5537 -
categorical_accuracy: 0.7764 - val_loss: 0.6780 - val_categorical_accuracy:
0.7224
Epoch 8/15
38/38 [==============================] - 0s 4ms/step - loss: 0.5116 -
categorical_accuracy: 0.7906 - val_loss: 0.6594 - val_categorical_accuracy:
0.7191
Epoch 9/15
38/38 [==============================] - 0s 4ms/step - loss: 0.4788 -
categorical_accuracy: 0.8141 - val_loss: 0.7015 - val_categorical_accuracy:
0.6789
Epoch 10/15
38/38 [==============================] - 0s 7ms/step - loss: 0.4693 -
categorical_accuracy: 0.8124 - val_loss: 0.6594 - val_categorical_accuracy:
0.7224
Epoch 11/15
38/38 [==============================] - 0s 3ms/step - loss: 0.4477 -
categorical_accuracy: 0.8233 - val_loss: 0.6572 - val_categorical_accuracy:
```

```
0.7291
Epoch 12/15
38/38 [==============================] - 0s 5ms/step - loss: 0.4263 -
categorical_accuracy: 0.8308 - val_loss: 0.6881 - val_categorical_accuracy:
0.7157
Epoch 13/15
38/38 [==============================] - 0s 4ms/step - loss: 0.4092 -
categorical_accuracy: 0.8367 - val_loss: 0.6949 - val_categorical_accuracy:
0.7124
Epoch 14/15
38/38 [==============================] - 0s 5ms/step - loss: 0.3954 -
categorical_accuracy: 0.8367 - val_loss: 0.6885 - val_categorical_accuracy:
0.7291
Epoch 15/15
38/38 [==============================] - 0s 3ms/step - loss: 0.3884 -
categorical_accuracy: 0.8476 - val_loss: 0.7094 - val_categorical_accuracy:
0.7224
-----------------
Training on fold 2
Epoch 1/15
38/38 [==============================] - 3s 39ms/step - loss: 1.0950 -
categorical_accuracy: 0.4595 - val_loss: 1.0887 - val_categorical_accuracy:
0.5819
Epoch 2/15
38/38 [==============================] - 1s 18ms/step - loss: 1.0714 -
categorical_accuracy: 0.5988 - val_loss: 1.0496 - val_categorical_accuracy:
0.6355
Epoch 3/15
38/38 [==============================] - 0s 5ms/step - loss: 0.9726 -
categorical_accuracy: 0.5980 - val_loss: 0.8936 - val_categorical_accuracy:
0.6355
Epoch 4/15
38/38 [==============================] - 0s 5ms/step - loss: 0.7354 -
categorical_accuracy: 0.6616 - val_loss: 0.7618 - val_categorical_accuracy:
0.6856
Epoch 5/15
38/38 [==============================] - 0s 11ms/step - loss: 0.6308 -
categorical_accuracy: 0.7404 - val_loss: 0.7512 - val_categorical_accuracy:
0.6957
Epoch 6/15
38/38 [==============================] - 0s 6ms/step - loss: 0.5891 -
categorical_accuracy: 0.7471 - val_loss: 0.7398 - val_categorical_accuracy:
0.6890
Epoch 7/15
38/38 [==============================] - 0s 7ms/step - loss: 0.5448 -
categorical_accuracy: 0.7806 - val_loss: 0.7436 - val_categorical_accuracy:
0.6957
Epoch 8/15
```

```
38/38 [==============================] - 0s 7ms/step - loss: 0.5159 -
categorical_accuracy: 0.7973 - val_loss: 0.7362 - val_categorical_accuracy:
0.6990
Epoch 9/15
38/38 [==============================] - 0s 4ms/step - loss: 0.4897 -
categorical_accuracy: 0.8090 - val_loss: 0.7399 - val_categorical_accuracy:
0.6990
Epoch 10/15
38/38 [==============================] - 0s 6ms/step - loss: 0.4741 -
categorical_accuracy: 0.8082 - val_loss: 0.7470 - val_categorical_accuracy:
0.7090
Epoch 11/15
38/38 [==============================] - 0s 4ms/step - loss: 0.4504 -
categorical_accuracy: 0.8191 - val_loss: 0.7514 - val_categorical_accuracy:
0.7324
Epoch 12/15
38/38 [==============================] - 0s 3ms/step - loss: 0.4333 -
categorical_accuracy: 0.8183 - val_loss: 0.7588 - val_categorical_accuracy:
0.7124
Epoch 13/15
38/38 [==============================] - 0s 3ms/step - loss: 0.4151 -
categorical_accuracy: 0.8317 - val_loss: 0.7707 - val_categorical_accuracy:
0.7191
Epoch 14/15
38/38 [==============================] - 0s 5ms/step - loss: 0.4127 -
categorical_accuracy: 0.8384 - val_loss: 0.7913 - val_categorical_accuracy:
0.7324
Epoch 15/15
38/38 [==============================] - 0s 3ms/step - loss: 0.3981 -
categorical_accuracy: 0.8484 - val_loss: 0.7739 - val_categorical_accuracy:
0.7191
-----------------
Training on fold 3
Epoch 1/15
38/38 [==============================] - 3s 41ms/step - loss: 1.0927 -
categorical_accuracy: 0.5335 - val_loss: 1.0875 - val_categorical_accuracy:
0.5403
Epoch 2/15
38/38 [==============================] - 1s 16ms/step - loss: 1.0679 -
categorical_accuracy: 0.5741 - val_loss: 1.0538 - val_categorical_accuracy:
0.5604
Epoch 3/15
38/38 [==============================] - 0s 8ms/step - loss: 0.9831 -
categorical_accuracy: 0.5833 - val_loss: 0.9382 - val_categorical_accuracy:
0.6107
Epoch 4/15
38/38 [==============================] - 0s 6ms/step - loss: 0.8120 -
categorical_accuracy: 0.6795 - val_loss: 0.7949 - val_categorical_accuracy:
```

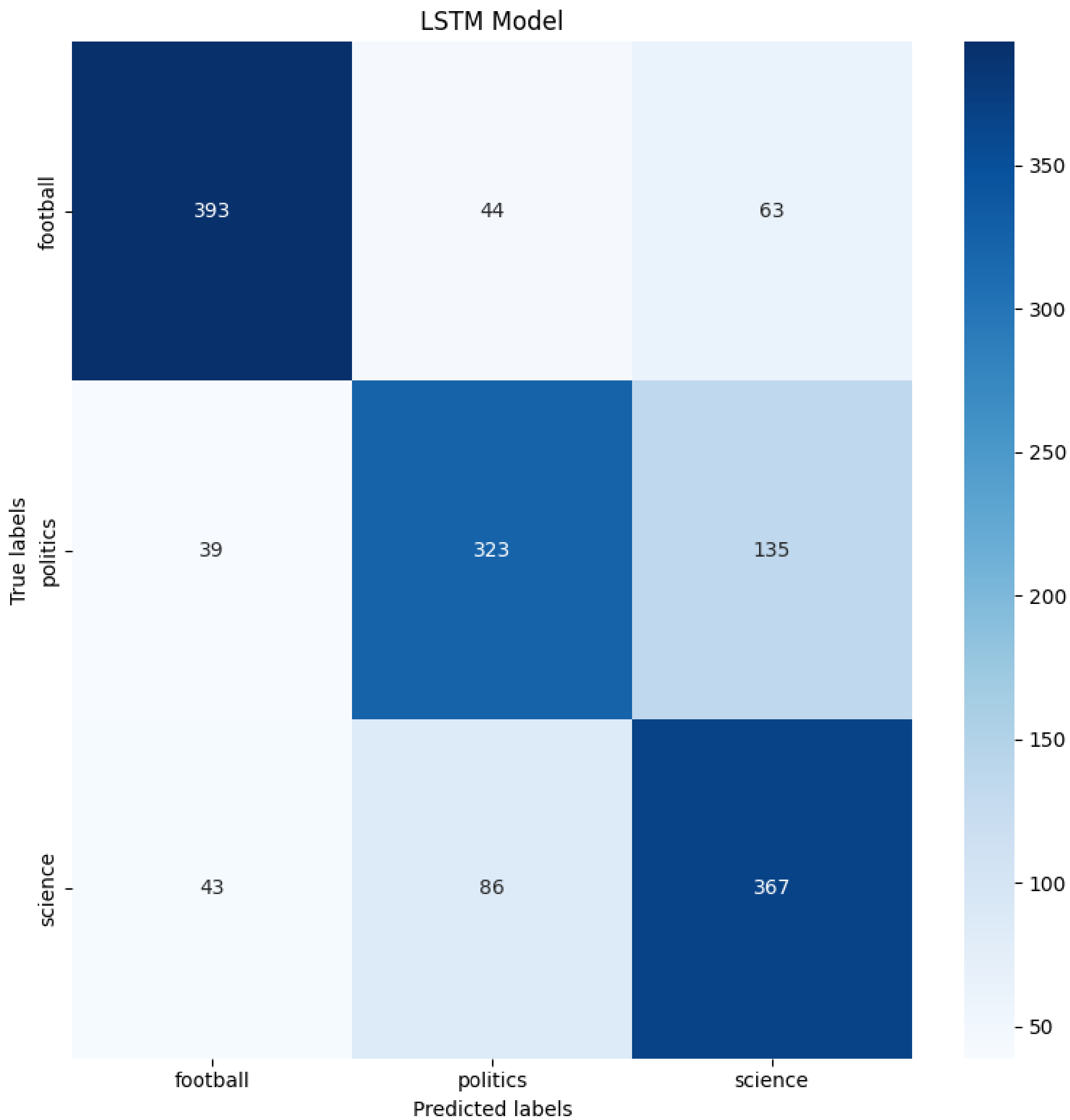
```
0.6208
Epoch 5/15
38/38 [==============================] - 0s 8ms/step - loss: 0.6720 -
categorical_accuracy: 0.7314 - val_loss: 0.7145 - val_categorical_accuracy:
0.6577
Epoch 6/15
38/38 [==============================] - 0s 7ms/step - loss: 0.6038 -
categorical_accuracy: 0.7632 - val_loss: 0.6989 - val_categorical_accuracy:
0.6711
Epoch 7/15
38/38 [==============================] - 0s 7ms/step - loss: 0.5622 -
categorical_accuracy: 0.7824 - val_loss: 0.6839 - val_categorical_accuracy:
0.6913
Epoch 8/15
38/38 [==============================] - 0s 5ms/step - loss: 0.5301 -
categorical_accuracy: 0.8050 - val_loss: 0.6828 - val_categorical_accuracy:
0.6846
Epoch 9/15
38/38 [==============================] - 0s 6ms/step - loss: 0.4998 -
categorical_accuracy: 0.8218 - val_loss: 0.6853 - val_categorical_accuracy:
0.6846
Epoch 10/15
38/38 [==============================] - 0s 5ms/step - loss: 0.4849 -
categorical_accuracy: 0.8251 - val_loss: 0.6696 - val_categorical_accuracy:
0.7081
Epoch 11/15
38/38 [==============================] - 0s 4ms/step - loss: 0.4517 -
categorical_accuracy: 0.8460 - val_loss: 0.6846 - val_categorical_accuracy:
0.6980
Epoch 12/15
38/38 [==============================] - 0s 9ms/step - loss: 0.4450 -
categorical_accuracy: 0.8435 - val_loss: 0.6829 - val_categorical_accuracy:
0.6980
Epoch 13/15
38/38 [==============================] - 0s 3ms/step - loss: 0.4357 -
categorical_accuracy: 0.8460 - val_loss: 0.6844 - val_categorical_accuracy:
0.7013
Epoch 14/15
38/38 [==============================] - 0s 3ms/step - loss: 0.4156 -
categorical_accuracy: 0.8477 - val_loss: 0.7081 - val_categorical_accuracy:
0.7081
Epoch 15/15
38/38 [==============================] - 0s 3ms/step - loss: 0.4026 -
categorical_accuracy: 0.8577 - val_loss: 0.6871 - val_categorical_accuracy:
0.7181
-----------------
Training on fold 4
Epoch 1/15
```

```
38/38 [==============================] - 3s 38ms/step - loss: 1.0951 -
categorical_accuracy: 0.4930 - val_loss: 1.0872 - val_categorical_accuracy:
0.5604
Epoch 2/15
38/38 [==============================] - 1s 16ms/step - loss: 1.0716 -
categorical_accuracy: 0.5439 - val_loss: 1.0479 - val_categorical_accuracy:
0.5772
Epoch 3/15
38/38 [==============================] - 0s 12ms/step - loss: 0.9835 -
categorical_accuracy: 0.6092 - val_loss: 0.8949 - val_categorical_accuracy:
0.6376
Epoch 4/15
38/38 [==============================] - 0s 8ms/step - loss: 0.7734 -
categorical_accuracy: 0.6703 - val_loss: 0.7443 - val_categorical_accuracy:
0.7047
Epoch 5/15
38/38 [==============================] - 0s 10ms/step - loss: 0.6889 -
categorical_accuracy: 0.7163 - val_loss: 0.7144 - val_categorical_accuracy:
0.7215
Epoch 6/15
38/38 [==============================] - 0s 5ms/step - loss: 0.6502 -
categorical_accuracy: 0.7498 - val_loss: 0.7220 - val_categorical_accuracy:
0.7282
Epoch 7/15
38/38 [==============================] - 0s 5ms/step - loss: 0.5981 -
categorical_accuracy: 0.7665 - val_loss: 0.7227 - val_categorical_accuracy:
0.7282
Epoch 8/15
38/38 [==============================] - 0s 4ms/step - loss: 0.5556 -
categorical_accuracy: 0.7958 - val_loss: 0.6772 - val_categorical_accuracy:
0.7248
Epoch 9/15
38/38 [==============================] - 0s 5ms/step - loss: 0.5191 -
categorical_accuracy: 0.7958 - val_loss: 0.6788 - val_categorical_accuracy:
0.7349
Epoch 10/15
38/38 [==============================] - 0s 4ms/step - loss: 0.4852 -
categorical_accuracy: 0.8176 - val_loss: 0.6760 - val_categorical_accuracy:
0.7248
Epoch 11/15
38/38 [==============================] - 0s 5ms/step - loss: 0.4645 -
categorical_accuracy: 0.8167 - val_loss: 0.6624 - val_categorical_accuracy:
0.7349
Epoch 12/15
38/38 [==============================] - 0s 4ms/step - loss: 0.4356 -
categorical_accuracy: 0.8335 - val_loss: 0.6776 - val_categorical_accuracy:
0.7282
Epoch 13/15
```

```
38/38 [==============================] - 0s 4ms/step - loss: 0.4255 -
categorical_accuracy: 0.8343 - val_loss: 0.6845 - val_categorical_accuracy:
0.7315
Epoch 14/15
38/38 [==============================] - 0s 4ms/step - loss: 0.4050 -
categorical_accuracy: 0.8427 - val_loss: 0.6803 - val_categorical_accuracy:
0.7349
Epoch 15/15
38/38 [==============================] - 0s 4ms/step - loss: 0.3972 -
categorical_accuracy: 0.8410 - val_loss: 0.6980 - val_categorical_accuracy:
0.7181
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
```

5.2 Confusion Matrix

[ ]:
```
plot_confusion_matrix(Y, pred_lstm, labels=list(vals), logscale=False,
 ↪title="LSTM Model")
```

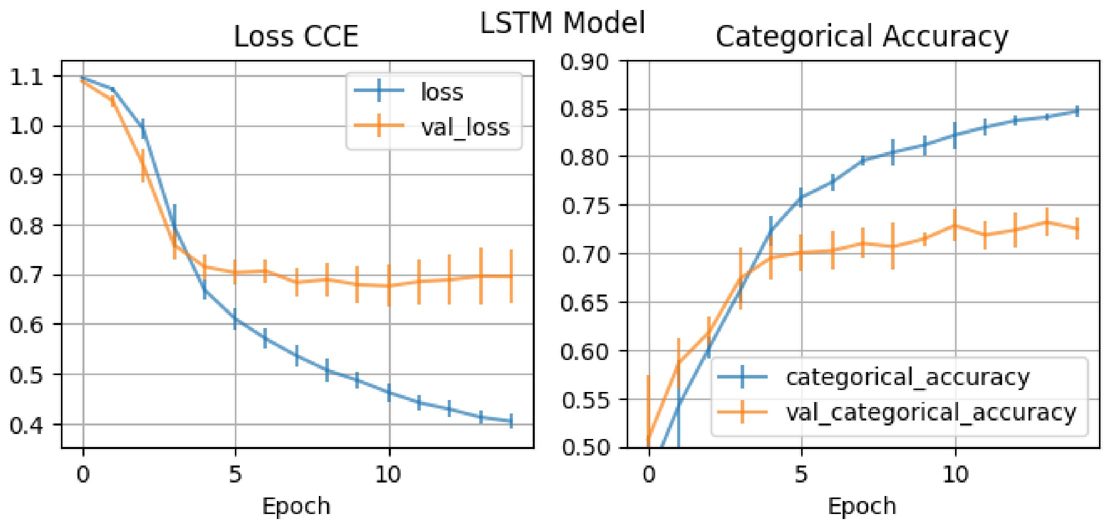



```
[ ]: array([[393,  44,  63],
            [ 39, 323, 135],
            [ 43,  86, 367]])
```

5.3 Training History

```
[ ]: f, ax = plt.subplots(ncols=2)
     f.set_size_inches(8,3)
     epochs = np.arange(len(histories_lstm[0].history["loss"]))
     plot_key(ax[0], epochs, histories_lstm, "loss")
     plot_key(ax[0], epochs, histories_lstm, "val_loss")
```

```
plot_key(ax[1], epochs, histories_lstm, "categorical_accuracy")
plot_key(ax[1], epochs, histories_lstm, "val_categorical_accuracy")
ax[0].set_title("Loss CCE")
ax[0].set_xlabel("Epoch")
ax[0].grid()
ax[0].legend()
ax[1].set_xlabel("Epoch")
ax[1].set_title("Categorical Accuracy")
ax[1].grid()
ax[1].set_ylim(0.5, 0.9)
plt.suptitle("LSTM Model")
ax[1].legend();
```



6 With GLOVE

[ ]:
```
import os
path_to_glove_file = "../assignment_3_keras_tf/glove.6B.50d.txt"

embeddings_index = {}
with open(path_to_glove_file) as f:
    for line in f:
        word, coefs = line.split(maxsplit=1)
        coefs = np.fromstring(coefs, "f", sep=" ")
        embeddings_index[word] = coefs

print("Found %s word vectors." % len(embeddings_index))

num_words = X.shape[1]
```

```
num_tokens = num_words + 2
embedding_dim = 50
hits = 0
misses = 0

# Prepare embedding matrix
words = [c for c in data.columns if c != "LABEL"]
embedding_matrix = np.zeros((num_tokens, embedding_dim))
for i, word in enumerate(words):
    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None and i < num_tokens:
        # Words not found in embedding index will be all-zeros.
        # This includes the representation for "padding" and "OOV"
        embedding_matrix[i] = embedding_vector
        hits += 1
    else:
        misses += 1
        print(word, "was a miss")
print("Converted %d words (%d misses)" % (hits, misses))
```

```
Found 400000 word vectors.
informationen was a miss
lesen was a miss
openai was a miss
Converted 297 words (3 misses)
```

[ ]:
```
embedding_matrix.shape
```

[ ]:
```
(302, 50)
```

6.1 Make Input Data:

make vectors containing all the words in the sequence

6.2 Define and Train

[ ]:
```
def model_glove(embedding_size=50, output_size=Y.shape[1], optimizer="adam",␣
↪loss="categorical_crossentropy",
              metrics=[keras.metrics.CategoricalAccuracy()]):

    model = keras.models.Sequential([
                # This initial dense layer serves as an embedding layer
                keras.layers.Embedding(input_dim=num_words+2,␣
↪output_dim=embedding_size,
                                          mask_zero=True,
                                          embeddings_initializer=keras.
↪initializers.Constant(embedding_matrix),
                                          trainable=False),
```

```
                # LSTM
                keras.layers.Reshape((max_len, embedding_size)),
                keras.layers.Bidirectional(keras.layers.LSTM(16,␣
↪return_sequences=False)),

                # CNN
                # keras.layers.Reshape((embedding_size, max_len)),
                # keras.layers.Reshape((max_len, embedding_size)),
                # keras.layers.Conv1D(8, kernel_size=3),
                # keras.layers.MaxPool1D(2),
                # keras.layers.Conv1D(4, kernel_size=3),
                # keras.layers.MaxPool1D(2),
                # keras.layers.Flatten(),

                keras.layers.Dense(output_size, activation="softmax")
            ])


    model.build(num_words + 2)
    model.compile(optimizer=optimizer, loss=loss, metrics=metrics)

    return model

cnn_test_glove = model_glove()
cnn_test_glove.summary()
```

```
Model: "sequential_18"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_6 (Embedding)     (None, None, 50)          15100

 reshape_12 (Reshape)        (None, 15, 50)            0

 bidirectional_6 (Bidirecti  (None, 32)                8576
 onal)

 dense_30 (Dense)            (None, 3)                 99

=================================================================
Total params: 23775 (92.87 KB)
Trainable params: 8675 (33.89 KB)
Non-trainable params: 15100 (58.98 KB)
_________________________________________________________________
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_6 (Embedding)     (None, None, 50)          15100
```

```
 reshape_12 (Reshape)        (None, 15, 50)            0

 bidirectional_6 (Bidirecti  (None, 32)                8576
 onal)

 dense_30 (Dense)            (None, 3)                 99

=================================================================
Total params: 23775 (92.87 KB)
Trainable params: 8675 (33.89 KB)
Non-trainable params: 15100 (58.98 KB)
_________________________________________________________________
```

[ ]:
```
models_glove, histories_glove = train_on_folds(model_glove, splits, X_seq, Y,␣
 ↪epochs=40)
pred_lstm = pred_on_folds(models_glove, splits, X_seq, Y)
```

```
-----------------
Training on fold 0
Epoch 1/40
38/38 [==============================] - 2s 11ms/step - loss: 1.0943 -
categorical_accuracy: 0.3987 - val_loss: 1.0433 - val_categorical_accuracy:
0.4181
Epoch 2/40
38/38 [==============================] - 0s 3ms/step - loss: 1.0264 -
categorical_accuracy: 0.4757 - val_loss: 0.9918 - val_categorical_accuracy:
0.5217
Epoch 3/40
38/38 [==============================] - 0s 3ms/step - loss: 0.9790 -
categorical_accuracy: 0.5117 - val_loss: 0.9444 - val_categorical_accuracy:
0.5284
Epoch 4/40
38/38 [==============================] - 0s 3ms/step - loss: 0.9324 -
categorical_accuracy: 0.5452 - val_loss: 0.9118 - val_categorical_accuracy:
0.5385
Epoch 5/40
38/38 [==============================] - 0s 3ms/step - loss: 0.8917 -
categorical_accuracy: 0.5704 - val_loss: 0.8876 - val_categorical_accuracy:
0.5819
Epoch 6/40
38/38 [==============================] - 0s 3ms/step - loss: 0.8668 -
categorical_accuracy: 0.5821 - val_loss: 0.8622 - val_categorical_accuracy:
0.5853
Epoch 7/40
38/38 [==============================] - 0s 3ms/step - loss: 0.8345 -
categorical_accuracy: 0.6122 - val_loss: 0.8502 - val_categorical_accuracy:
0.6020
```

```
Epoch 8/40
38/38 [==============================] - 0s 3ms/step - loss: 0.7988 -
categorical_accuracy: 0.6491 - val_loss: 0.8352 - val_categorical_accuracy:
0.5853
Epoch 9/40
38/38 [==============================] - 0s 3ms/step - loss: 0.7846 -
categorical_accuracy: 0.6516 - val_loss: 0.8154 - val_categorical_accuracy:
0.6187
Epoch 10/40
38/38 [==============================] - 0s 3ms/step - loss: 0.7457 -
categorical_accuracy: 0.6759 - val_loss: 0.8102 - val_categorical_accuracy:
0.6455
Epoch 11/40
38/38 [==============================] - 0s 3ms/step - loss: 0.7250 -
categorical_accuracy: 0.6968 - val_loss: 0.7985 - val_categorical_accuracy:
0.6254
Epoch 12/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6972 -
categorical_accuracy: 0.7111 - val_loss: 0.7863 - val_categorical_accuracy:
0.6221
Epoch 13/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6823 -
categorical_accuracy: 0.7152 - val_loss: 0.7966 - val_categorical_accuracy:
0.6187
Epoch 14/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6727 -
categorical_accuracy: 0.7136 - val_loss: 0.7800 - val_categorical_accuracy:
0.6154
Epoch 15/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6588 -
categorical_accuracy: 0.7219 - val_loss: 0.7723 - val_categorical_accuracy:
0.6455
Epoch 16/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6362 -
categorical_accuracy: 0.7337 - val_loss: 0.7761 - val_categorical_accuracy:
0.6421
Epoch 17/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6233 -
categorical_accuracy: 0.7370 - val_loss: 0.7677 - val_categorical_accuracy:
0.6455
Epoch 18/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6021 -
categorical_accuracy: 0.7538 - val_loss: 0.7746 - val_categorical_accuracy:
0.6187
Epoch 19/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5914 -
categorical_accuracy: 0.7588 - val_loss: 0.7742 - val_categorical_accuracy:
0.6388
```

```
Epoch 20/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5682 -
categorical_accuracy: 0.7755 - val_loss: 0.7708 - val_categorical_accuracy:
0.6187
Epoch 21/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5516 -
categorical_accuracy: 0.7839 - val_loss: 0.7718 - val_categorical_accuracy:
0.6388
Epoch 22/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5402 -
categorical_accuracy: 0.7772 - val_loss: 0.7589 - val_categorical_accuracy:
0.6656
Epoch 23/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5291 -
categorical_accuracy: 0.7982 - val_loss: 0.7931 - val_categorical_accuracy:
0.6355
Epoch 24/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5166 -
categorical_accuracy: 0.8015 - val_loss: 0.8079 - val_categorical_accuracy:
0.6221
Epoch 25/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4960 -
categorical_accuracy: 0.8099 - val_loss: 0.7821 - val_categorical_accuracy:
0.6321
Epoch 26/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4828 -
categorical_accuracy: 0.8157 - val_loss: 0.7877 - val_categorical_accuracy:
0.6321
Epoch 27/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4851 -
categorical_accuracy: 0.8124 - val_loss: 0.7985 - val_categorical_accuracy:
0.6288
Epoch 28/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4584 -
categorical_accuracy: 0.8250 - val_loss: 0.7789 - val_categorical_accuracy:
0.6355
Epoch 29/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4435 -
categorical_accuracy: 0.8325 - val_loss: 0.7750 - val_categorical_accuracy:
0.6589
Epoch 30/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4681 -
categorical_accuracy: 0.8107 - val_loss: 0.7592 - val_categorical_accuracy:
0.6522
Epoch 31/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4256 -
categorical_accuracy: 0.8392 - val_loss: 0.7907 - val_categorical_accuracy:
0.6522
```

```
Epoch 32/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4146 -
categorical_accuracy: 0.8417 - val_loss: 0.7928 - val_categorical_accuracy:
0.6522
Epoch 33/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3981 -
categorical_accuracy: 0.8492 - val_loss: 0.8099 - val_categorical_accuracy:
0.6488
Epoch 34/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3925 -
categorical_accuracy: 0.8526 - val_loss: 0.8067 - val_categorical_accuracy:
0.6589
Epoch 35/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3831 -
categorical_accuracy: 0.8576 - val_loss: 0.8283 - val_categorical_accuracy:
0.6522
Epoch 36/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3815 -
categorical_accuracy: 0.8559 - val_loss: 0.8082 - val_categorical_accuracy:
0.6589
Epoch 37/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3733 -
categorical_accuracy: 0.8551 - val_loss: 0.7770 - val_categorical_accuracy:
0.6656
Epoch 38/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3699 -
categorical_accuracy: 0.8576 - val_loss: 0.8451 - val_categorical_accuracy:
0.6455
Epoch 39/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3652 -
categorical_accuracy: 0.8593 - val_loss: 0.8011 - val_categorical_accuracy:
0.6522
Epoch 40/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3590 -
categorical_accuracy: 0.8559 - val_loss: 0.8179 - val_categorical_accuracy:
0.6589
-----------------
Training on fold 1
Epoch 1/40
38/38 [==============================] - 2s 11ms/step - loss: 1.0639 -
categorical_accuracy: 0.4869 - val_loss: 1.0278 - val_categorical_accuracy:
0.4950
Epoch 2/40
38/38 [==============================] - 0s 3ms/step - loss: 0.9963 -
categorical_accuracy: 0.5151 - val_loss: 0.9898 - val_categorical_accuracy:
0.5017
Epoch 3/40
38/38 [==============================] - 0s 3ms/step - loss: 0.9460 -
```

```
categorical_accuracy: 0.5486 - val_loss: 0.9301 - val_categorical_accuracy:
0.5619
Epoch 4/40
38/38 [==============================] - 0s 3ms/step - loss: 0.9003 -
categorical_accuracy: 0.5754 - val_loss: 0.8872 - val_categorical_accuracy:
0.5652
Epoch 5/40
38/38 [==============================] - 0s 3ms/step - loss: 0.8578 -
categorical_accuracy: 0.5997 - val_loss: 0.8521 - val_categorical_accuracy:
0.6254
Epoch 6/40
38/38 [==============================] - 0s 3ms/step - loss: 0.8180 -
categorical_accuracy: 0.6340 - val_loss: 0.8204 - val_categorical_accuracy:
0.6087
Epoch 7/40
38/38 [==============================] - 0s 3ms/step - loss: 0.7975 -
categorical_accuracy: 0.6449 - val_loss: 0.8017 - val_categorical_accuracy:
0.6187
Epoch 8/40
38/38 [==============================] - 0s 3ms/step - loss: 0.7568 -
categorical_accuracy: 0.6625 - val_loss: 0.7905 - val_categorical_accuracy:
0.6321
Epoch 9/40
38/38 [==============================] - 0s 3ms/step - loss: 0.7286 -
categorical_accuracy: 0.6868 - val_loss: 0.7990 - val_categorical_accuracy:
0.6087
Epoch 10/40
38/38 [==============================] - 0s 3ms/step - loss: 0.7166 -
categorical_accuracy: 0.6734 - val_loss: 0.7748 - val_categorical_accuracy:
0.6488
Epoch 11/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6965 -
categorical_accuracy: 0.6809 - val_loss: 0.7716 - val_categorical_accuracy:
0.6455
Epoch 12/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6693 -
categorical_accuracy: 0.7094 - val_loss: 0.7698 - val_categorical_accuracy:
0.6555
Epoch 13/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6590 -
categorical_accuracy: 0.7085 - val_loss: 0.7699 - val_categorical_accuracy:
0.6589
Epoch 14/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6444 -
categorical_accuracy: 0.7119 - val_loss: 0.7542 - val_categorical_accuracy:
0.6488
Epoch 15/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6149 -
```

```
categorical_accuracy: 0.7429 - val_loss: 0.7513 - val_categorical_accuracy:
0.6823
Epoch 16/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6099 -
categorical_accuracy: 0.7395 - val_loss: 0.7473 - val_categorical_accuracy:
0.6823
Epoch 17/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5921 -
categorical_accuracy: 0.7479 - val_loss: 0.7809 - val_categorical_accuracy:
0.6488
Epoch 18/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5761 -
categorical_accuracy: 0.7538 - val_loss: 0.7403 - val_categorical_accuracy:
0.6689
Epoch 19/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5705 -
categorical_accuracy: 0.7538 - val_loss: 0.7802 - val_categorical_accuracy:
0.6789
Epoch 20/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5519 -
categorical_accuracy: 0.7680 - val_loss: 0.7488 - val_categorical_accuracy:
0.6957
Epoch 21/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5328 -
categorical_accuracy: 0.7781 - val_loss: 0.7272 - val_categorical_accuracy:
0.7057
Epoch 22/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5226 -
categorical_accuracy: 0.7898 - val_loss: 0.7350 - val_categorical_accuracy:
0.7124
Epoch 23/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5095 -
categorical_accuracy: 0.7881 - val_loss: 0.7340 - val_categorical_accuracy:
0.6722
Epoch 24/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5007 -
categorical_accuracy: 0.7915 - val_loss: 0.7442 - val_categorical_accuracy:
0.7157
Epoch 25/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5012 -
categorical_accuracy: 0.7940 - val_loss: 0.7651 - val_categorical_accuracy:
0.6890
Epoch 26/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4777 -
categorical_accuracy: 0.8124 - val_loss: 0.7464 - val_categorical_accuracy:
0.7023
Epoch 27/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4638 -
```

```
categorical_accuracy: 0.8132 - val_loss: 0.7560 - val_categorical_accuracy:
0.7090
Epoch 28/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4554 -
categorical_accuracy: 0.8166 - val_loss: 0.7501 - val_categorical_accuracy:
0.7057
Epoch 29/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4517 -
categorical_accuracy: 0.8258 - val_loss: 0.7588 - val_categorical_accuracy:
0.7023
Epoch 30/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4310 -
categorical_accuracy: 0.8266 - val_loss: 0.7965 - val_categorical_accuracy:
0.6990
Epoch 31/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4315 -
categorical_accuracy: 0.8275 - val_loss: 0.7748 - val_categorical_accuracy:
0.6856
Epoch 32/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4338 -
categorical_accuracy: 0.8275 - val_loss: 0.7969 - val_categorical_accuracy:
0.6957
Epoch 33/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4144 -
categorical_accuracy: 0.8308 - val_loss: 0.7611 - val_categorical_accuracy:
0.6957
Epoch 34/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4288 -
categorical_accuracy: 0.8224 - val_loss: 0.7738 - val_categorical_accuracy:
0.7258
Epoch 35/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3849 -
categorical_accuracy: 0.8476 - val_loss: 0.7975 - val_categorical_accuracy:
0.7124
Epoch 36/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3875 -
categorical_accuracy: 0.8467 - val_loss: 0.8309 - val_categorical_accuracy:
0.7023
Epoch 37/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3971 -
categorical_accuracy: 0.8375 - val_loss: 0.8115 - val_categorical_accuracy:
0.6923
Epoch 38/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3716 -
categorical_accuracy: 0.8543 - val_loss: 0.7941 - val_categorical_accuracy:
0.6957
Epoch 39/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3614 -
```

```
categorical_accuracy: 0.8576 - val_loss: 0.8198 - val_categorical_accuracy:
0.7057
Epoch 40/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3682 -
categorical_accuracy: 0.8601 - val_loss: 0.8169 - val_categorical_accuracy:
0.7124
-----------------
Training on fold 2
Epoch 1/40
38/38 [==============================] - 2s 11ms/step - loss: 1.0952 -
categorical_accuracy: 0.4635 - val_loss: 1.0464 - val_categorical_accuracy:
0.4816
Epoch 2/40
38/38 [==============================] - 0s 3ms/step - loss: 1.0170 -
categorical_accuracy: 0.4992 - val_loss: 1.0070 - val_categorical_accuracy:
0.4716
Epoch 3/40
38/38 [==============================] - 0s 3ms/step - loss: 0.9768 -
categorical_accuracy: 0.5285 - val_loss: 0.9617 - val_categorical_accuracy:
0.5318
Epoch 4/40
38/38 [==============================] - 0s 3ms/step - loss: 0.9336 -
categorical_accuracy: 0.5536 - val_loss: 0.9352 - val_categorical_accuracy:
0.5385
Epoch 5/40
38/38 [==============================] - 0s 3ms/step - loss: 0.8972 -
categorical_accuracy: 0.5628 - val_loss: 0.9035 - val_categorical_accuracy:
0.5719
Epoch 6/40
38/38 [==============================] - 0s 3ms/step - loss: 0.8615 -
categorical_accuracy: 0.5871 - val_loss: 0.8760 - val_categorical_accuracy:
0.5853
Epoch 7/40
38/38 [==============================] - 0s 3ms/step - loss: 0.8292 -
categorical_accuracy: 0.6181 - val_loss: 0.8713 - val_categorical_accuracy:
0.5853
Epoch 8/40
38/38 [==============================] - 0s 3ms/step - loss: 0.7989 -
categorical_accuracy: 0.6399 - val_loss: 0.8471 - val_categorical_accuracy:
0.5953
Epoch 9/40
38/38 [==============================] - 0s 3ms/step - loss: 0.7688 -
categorical_accuracy: 0.6558 - val_loss: 0.8163 - val_categorical_accuracy:
0.5853
Epoch 10/40
38/38 [==============================] - 0s 3ms/step - loss: 0.7267 -
categorical_accuracy: 0.6834 - val_loss: 0.8177 - val_categorical_accuracy:
0.5853
```

```
Epoch 11/40
38/38 [==============================] - 0s 3ms/step - loss: 0.7012 -
categorical_accuracy: 0.6843 - val_loss: 0.7827 - val_categorical_accuracy:
0.6087
Epoch 12/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6651 -
categorical_accuracy: 0.7069 - val_loss: 0.7578 - val_categorical_accuracy:
0.6823
Epoch 13/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6837 -
categorical_accuracy: 0.6968 - val_loss: 0.7802 - val_categorical_accuracy:
0.6288
Epoch 14/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6170 -
categorical_accuracy: 0.7245 - val_loss: 0.7572 - val_categorical_accuracy:
0.6923
Epoch 15/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5942 -
categorical_accuracy: 0.7404 - val_loss: 0.7937 - val_categorical_accuracy:
0.6187
Epoch 16/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5763 -
categorical_accuracy: 0.7513 - val_loss: 0.7740 - val_categorical_accuracy:
0.6388
Epoch 17/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5601 -
categorical_accuracy: 0.7647 - val_loss: 0.8028 - val_categorical_accuracy:
0.6254
Epoch 18/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5546 -
categorical_accuracy: 0.7479 - val_loss: 0.7827 - val_categorical_accuracy:
0.6555
Epoch 19/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5327 -
categorical_accuracy: 0.7697 - val_loss: 0.7647 - val_categorical_accuracy:
0.6656
Epoch 20/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5170 -
categorical_accuracy: 0.7722 - val_loss: 0.7852 - val_categorical_accuracy:
0.6756
Epoch 21/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4958 -
categorical_accuracy: 0.7797 - val_loss: 0.8553 - val_categorical_accuracy:
0.6288
Epoch 22/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4927 -
categorical_accuracy: 0.7864 - val_loss: 0.8228 - val_categorical_accuracy:
0.6288
```

```
Epoch 23/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4639 -
categorical_accuracy: 0.7998 - val_loss: 0.8205 - val_categorical_accuracy:
0.6321
Epoch 24/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4450 -
categorical_accuracy: 0.8099 - val_loss: 0.8206 - val_categorical_accuracy:
0.6722
Epoch 25/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4318 -
categorical_accuracy: 0.8241 - val_loss: 0.8705 - val_categorical_accuracy:
0.6455
Epoch 26/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4193 -
categorical_accuracy: 0.8266 - val_loss: 0.8616 - val_categorical_accuracy:
0.6756
Epoch 27/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4146 -
categorical_accuracy: 0.8208 - val_loss: 0.8706 - val_categorical_accuracy:
0.6689
Epoch 28/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4102 -
categorical_accuracy: 0.8325 - val_loss: 0.8604 - val_categorical_accuracy:
0.7023
Epoch 29/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3915 -
categorical_accuracy: 0.8434 - val_loss: 0.9178 - val_categorical_accuracy:
0.6455
Epoch 30/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3746 -
categorical_accuracy: 0.8476 - val_loss: 0.9022 - val_categorical_accuracy:
0.6789
Epoch 31/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3775 -
categorical_accuracy: 0.8409 - val_loss: 0.8867 - val_categorical_accuracy:
0.6689
Epoch 32/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3763 -
categorical_accuracy: 0.8384 - val_loss: 0.9215 - val_categorical_accuracy:
0.6421
Epoch 33/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3614 -
categorical_accuracy: 0.8451 - val_loss: 0.9256 - val_categorical_accuracy:
0.6321
Epoch 34/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3464 -
categorical_accuracy: 0.8559 - val_loss: 0.9269 - val_categorical_accuracy:
0.6522
```

```
Epoch 35/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3458 -
categorical_accuracy: 0.8585 - val_loss: 0.9276 - val_categorical_accuracy:
0.6555
Epoch 36/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3248 -
categorical_accuracy: 0.8626 - val_loss: 0.9696 - val_categorical_accuracy:
0.6823
Epoch 37/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3331 -
categorical_accuracy: 0.8626 - val_loss: 0.9504 - val_categorical_accuracy:
0.6589
Epoch 38/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3217 -
categorical_accuracy: 0.8626 - val_loss: 1.0103 - val_categorical_accuracy:
0.6455
Epoch 39/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3115 -
categorical_accuracy: 0.8660 - val_loss: 0.9682 - val_categorical_accuracy:
0.6455
Epoch 40/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3047 -
categorical_accuracy: 0.8744 - val_loss: 1.0375 - val_categorical_accuracy:
0.6722
-----------------
Training on fold 3
Epoch 1/40
38/38 [==============================] - 2s 11ms/step - loss: 1.0936 -
categorical_accuracy: 0.4398 - val_loss: 1.0691 - val_categorical_accuracy:
0.4161
Epoch 2/40
38/38 [==============================] - 0s 3ms/step - loss: 1.0342 -
categorical_accuracy: 0.4669 - val_loss: 1.0301 - val_categorical_accuracy:
0.4832
Epoch 3/40
38/38 [==============================] - 0s 3ms/step - loss: 0.9644 -
categorical_accuracy: 0.5289 - val_loss: 0.9531 - val_categorical_accuracy:
0.5503
Epoch 4/40
38/38 [==============================] - 0s 3ms/step - loss: 0.8916 -
categorical_accuracy: 0.5623 - val_loss: 0.9045 - val_categorical_accuracy:
0.5805
Epoch 5/40
38/38 [==============================] - 0s 3ms/step - loss: 0.8451 -
categorical_accuracy: 0.5983 - val_loss: 0.9260 - val_categorical_accuracy:
0.5369
Epoch 6/40
38/38 [==============================] - 0s 3ms/step - loss: 0.8109 -
```

```
categorical_accuracy: 0.6134 - val_loss: 0.8624 - val_categorical_accuracy:
0.6040
Epoch 7/40
38/38 [==============================] - 0s 3ms/step - loss: 0.7725 -
categorical_accuracy: 0.6435 - val_loss: 0.8604 - val_categorical_accuracy:
0.5906
Epoch 8/40
38/38 [==============================] - 0s 3ms/step - loss: 0.7458 -
categorical_accuracy: 0.6460 - val_loss: 0.8349 - val_categorical_accuracy:
0.6208
Epoch 9/40
38/38 [==============================] - 0s 3ms/step - loss: 0.7136 -
categorical_accuracy: 0.6653 - val_loss: 0.8286 - val_categorical_accuracy:
0.6309
Epoch 10/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6900 -
categorical_accuracy: 0.6854 - val_loss: 0.8533 - val_categorical_accuracy:
0.6443
Epoch 11/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6585 -
categorical_accuracy: 0.7088 - val_loss: 0.8221 - val_categorical_accuracy:
0.6577
Epoch 12/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6307 -
categorical_accuracy: 0.7264 - val_loss: 0.8143 - val_categorical_accuracy:
0.6443
Epoch 13/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6062 -
categorical_accuracy: 0.7389 - val_loss: 0.8346 - val_categorical_accuracy:
0.6510
Epoch 14/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5859 -
categorical_accuracy: 0.7414 - val_loss: 0.8264 - val_categorical_accuracy:
0.6611
Epoch 15/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5786 -
categorical_accuracy: 0.7414 - val_loss: 0.8218 - val_categorical_accuracy:
0.6644
Epoch 16/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5621 -
categorical_accuracy: 0.7565 - val_loss: 0.8271 - val_categorical_accuracy:
0.6544
Epoch 17/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5289 -
categorical_accuracy: 0.7699 - val_loss: 0.8228 - val_categorical_accuracy:
0.6577
Epoch 18/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5257 -
```

```
categorical_accuracy: 0.7741 - val_loss: 0.8653 - val_categorical_accuracy:
0.6443
Epoch 19/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5040 -
categorical_accuracy: 0.7958 - val_loss: 0.9156 - val_categorical_accuracy:
0.6510
Epoch 20/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5246 -
categorical_accuracy: 0.7816 - val_loss: 0.8569 - val_categorical_accuracy:
0.6477
Epoch 21/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4989 -
categorical_accuracy: 0.7874 - val_loss: 0.8422 - val_categorical_accuracy:
0.6711
Epoch 22/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4875 -
categorical_accuracy: 0.7891 - val_loss: 0.8465 - val_categorical_accuracy:
0.6611
Epoch 23/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4743 -
categorical_accuracy: 0.7874 - val_loss: 0.8405 - val_categorical_accuracy:
0.6611
Epoch 24/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4490 -
categorical_accuracy: 0.8201 - val_loss: 0.8377 - val_categorical_accuracy:
0.6678
Epoch 25/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4362 -
categorical_accuracy: 0.8192 - val_loss: 0.9388 - val_categorical_accuracy:
0.6577
Epoch 26/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4272 -
categorical_accuracy: 0.8259 - val_loss: 0.8596 - val_categorical_accuracy:
0.6745
Epoch 27/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4414 -
categorical_accuracy: 0.8126 - val_loss: 0.8851 - val_categorical_accuracy:
0.6443
Epoch 28/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4020 -
categorical_accuracy: 0.8335 - val_loss: 0.8952 - val_categorical_accuracy:
0.6544
Epoch 29/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3923 -
categorical_accuracy: 0.8377 - val_loss: 0.8866 - val_categorical_accuracy:
0.6611
Epoch 30/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3753 -
```

```
categorical_accuracy: 0.8452 - val_loss: 0.9433 - val_categorical_accuracy:
0.6577
Epoch 31/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3749 -
categorical_accuracy: 0.8536 - val_loss: 0.9031 - val_categorical_accuracy:
0.6510
Epoch 32/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3645 -
categorical_accuracy: 0.8527 - val_loss: 0.9290 - val_categorical_accuracy:
0.6745
Epoch 33/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3557 -
categorical_accuracy: 0.8544 - val_loss: 0.9401 - val_categorical_accuracy:
0.6745
Epoch 34/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3449 -
categorical_accuracy: 0.8669 - val_loss: 0.9416 - val_categorical_accuracy:
0.6611
Epoch 35/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3434 -
categorical_accuracy: 0.8619 - val_loss: 0.9482 - val_categorical_accuracy:
0.6577
Epoch 36/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3237 -
categorical_accuracy: 0.8636 - val_loss: 1.0010 - val_categorical_accuracy:
0.6678
Epoch 37/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3197 -
categorical_accuracy: 0.8711 - val_loss: 0.9656 - val_categorical_accuracy:
0.6644
Epoch 38/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3143 -
categorical_accuracy: 0.8787 - val_loss: 0.9764 - val_categorical_accuracy:
0.6644
Epoch 39/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3116 -
categorical_accuracy: 0.8745 - val_loss: 1.0210 - val_categorical_accuracy:
0.6644
Epoch 40/40
38/38 [==============================] - 0s 3ms/step - loss: 0.3067 -
categorical_accuracy: 0.8703 - val_loss: 0.9962 - val_categorical_accuracy:
0.6611
-----------------
Training on fold 4
Epoch 1/40
38/38 [==============================] - 2s 11ms/step - loss: 1.0911 -
categorical_accuracy: 0.4421 - val_loss: 1.0523 - val_categorical_accuracy:
0.4262
```

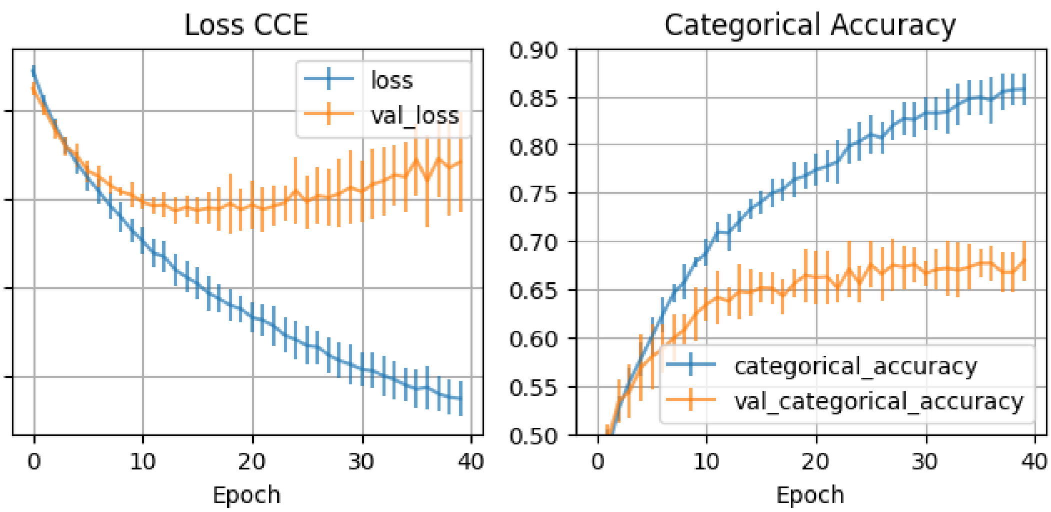
```
Epoch 2/40
38/38 [==============================] - 0s 3ms/step - loss: 1.0240 -
categorical_accuracy: 0.4469 - val_loss: 0.9964 - val_categorical_accuracy:
0.4866
Epoch 3/40
38/38 [==============================] - 0s 3ms/step - loss: 0.9670 -
categorical_accuracy: 0.5146 - val_loss: 0.9942 - val_categorical_accuracy:
0.4966
Epoch 4/40
38/38 [==============================] - 0s 3ms/step - loss: 0.9417 -
categorical_accuracy: 0.5381 - val_loss: 0.9479 - val_categorical_accuracy:
0.5000
Epoch 5/40
38/38 [==============================] - 0s 3ms/step - loss: 0.9117 -
categorical_accuracy: 0.5573 - val_loss: 0.9185 - val_categorical_accuracy:
0.5268
Epoch 6/40
38/38 [==============================] - 0s 3ms/step - loss: 0.8760 -
categorical_accuracy: 0.5866 - val_loss: 0.8947 - val_categorical_accuracy:
0.5168
Epoch 7/40
38/38 [==============================] - 0s 3ms/step - loss: 0.8520 -
categorical_accuracy: 0.5992 - val_loss: 0.8663 - val_categorical_accuracy:
0.5436
Epoch 8/40
38/38 [==============================] - 0s 3ms/step - loss: 0.8271 -
categorical_accuracy: 0.6318 - val_loss: 0.8478 - val_categorical_accuracy:
0.5638
Epoch 9/40
38/38 [==============================] - 0s 3ms/step - loss: 0.8035 -
categorical_accuracy: 0.6301 - val_loss: 0.8226 - val_categorical_accuracy:
0.5973
Epoch 10/40
38/38 [==============================] - 0s 3ms/step - loss: 0.7688 -
categorical_accuracy: 0.6728 - val_loss: 0.7879 - val_categorical_accuracy:
0.5973
Epoch 11/40
38/38 [==============================] - 0s 3ms/step - loss: 0.7453 -
categorical_accuracy: 0.6678 - val_loss: 0.7891 - val_categorical_accuracy:
0.6342
Epoch 12/40
38/38 [==============================] - 0s 3ms/step - loss: 0.7247 -
categorical_accuracy: 0.6937 - val_loss: 0.7891 - val_categorical_accuracy:
0.6040
Epoch 13/40
38/38 [==============================] - 0s 3ms/step - loss: 0.7139 -
categorical_accuracy: 0.6828 - val_loss: 0.7499 - val_categorical_accuracy:
0.6309
```

```
Epoch 14/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6795 -
categorical_accuracy: 0.7121 - val_loss: 0.7435 - val_categorical_accuracy:
0.6208
Epoch 15/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6697 -
categorical_accuracy: 0.7188 - val_loss: 0.7675 - val_categorical_accuracy:
0.6208
Epoch 16/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6604 -
categorical_accuracy: 0.7222 - val_loss: 0.7423 - val_categorical_accuracy:
0.6409
Epoch 17/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6335 -
categorical_accuracy: 0.7305 - val_loss: 0.7206 - val_categorical_accuracy:
0.6779
Epoch 18/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6145 -
categorical_accuracy: 0.7397 - val_loss: 0.7236 - val_categorical_accuracy:
0.6275
Epoch 19/40
38/38 [==============================] - 0s 3ms/step - loss: 0.6020 -
categorical_accuracy: 0.7414 - val_loss: 0.7095 - val_categorical_accuracy:
0.6443
Epoch 20/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5961 -
categorical_accuracy: 0.7406 - val_loss: 0.7163 - val_categorical_accuracy:
0.6846
Epoch 21/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5834 -
categorical_accuracy: 0.7406 - val_loss: 0.7337 - val_categorical_accuracy:
0.6678
Epoch 22/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5873 -
categorical_accuracy: 0.7456 - val_loss: 0.7169 - val_categorical_accuracy:
0.6477
Epoch 23/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5891 -
categorical_accuracy: 0.7389 - val_loss: 0.7298 - val_categorical_accuracy:
0.6544
Epoch 24/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5499 -
categorical_accuracy: 0.7682 - val_loss: 0.7441 - val_categorical_accuracy:
0.6779
Epoch 25/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5475 -
categorical_accuracy: 0.7690 - val_loss: 0.7384 - val_categorical_accuracy:
0.6544
```

```
Epoch 26/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5392 -
categorical_accuracy: 0.7715 - val_loss: 0.7080 - val_categorical_accuracy:
0.6946
Epoch 27/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5248 -
categorical_accuracy: 0.7749 - val_loss: 0.7238 - val_categorical_accuracy:
0.6779
Epoch 28/40
38/38 [==============================] - 0s 3ms/step - loss: 0.5127 -
categorical_accuracy: 0.7900 - val_loss: 0.7297 - val_categorical_accuracy:
0.6779
Epoch 29/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4970 -
categorical_accuracy: 0.7950 - val_loss: 0.7198 - val_categorical_accuracy:
0.6980
Epoch 30/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4854 -
categorical_accuracy: 0.7992 - val_loss: 0.7269 - val_categorical_accuracy:
0.6913
Epoch 31/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4714 -
categorical_accuracy: 0.8025 - val_loss: 0.7209 - val_categorical_accuracy:
0.6745
Epoch 32/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4752 -
categorical_accuracy: 0.8017 - val_loss: 0.7230 - val_categorical_accuracy:
0.6879
Epoch 33/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4780 -
categorical_accuracy: 0.7900 - val_loss: 0.7652 - val_categorical_accuracy:
0.7081
Epoch 34/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4528 -
categorical_accuracy: 0.8109 - val_loss: 0.8192 - val_categorical_accuracy:
0.6510
Epoch 35/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4463 -
categorical_accuracy: 0.8126 - val_loss: 0.7423 - val_categorical_accuracy:
0.6879
Epoch 36/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4406 -
categorical_accuracy: 0.8151 - val_loss: 0.8310 - val_categorical_accuracy:
0.6745
Epoch 37/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4541 -
categorical_accuracy: 0.8033 - val_loss: 0.6970 - val_categorical_accuracy:
0.7047
```

```
Epoch 38/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4281 -
categorical_accuracy: 0.8201 - val_loss: 0.8289 - val_categorical_accuracy:
0.6879
Epoch 39/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4159 -
categorical_accuracy: 0.8268 - val_loss: 0.7369 - val_categorical_accuracy:
0.6711
Epoch 40/40
38/38 [==============================] - 0s 3ms/step - loss: 0.4117 -
categorical_accuracy: 0.8259 - val_loss: 0.7459 - val_categorical_accuracy:
0.6946
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
10/10 [==============================] - 0s 1ms/step
```

[ ]:
```
f, ax = plt.subplots(ncols=2)
f.set_size_inches(8,3)
epochs = np.arange(len(histories_glove[0].history["loss"]))
plot_key(ax[0], epochs, histories_glove, "loss")
plot_key(ax[0], epochs, histories_glove, "val_loss")
plot_key(ax[1], epochs, histories_glove, "categorical_accuracy")
plot_key(ax[1], epochs, histories_glove, "val_categorical_accuracy")
ax[0].set_title("Loss CCE")
ax[0].set_xlabel("Epoch")
ax[0].legend()
ax[0].grid()
ax[1].set_xlabel("Epoch")
ax[1].set_title("Categorical Accuracy")
ax[1].grid()
ax[1].set_ylim(0.5, 0.9)
ax[1].legend();
```


Loss CCE
loss
val_loss
1.0
0.8
0.6
0.4
0
10
20
30
40
Epoch
Categorical Accuracy
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
categorical_accuracy
val_categorical_accuracy
0
10
20
30
40
Epoch