

ML_L7_CA2

SUBMISSION TO: CCT COLLEGE DUBLIN

AUTHOR: KENNY THOMAS

STUDENT NUMBER: SBA20022

Introduction

For purposes of this CA, I was given a dataset 'glass_data.csv', the dataset consists of following columns:

- Id number: 1 to 214
- Ri: refractive index
- Na: Sodium (unit measurement: weight percent in corresponding oxide, as are attributes 4-10)
- Mg: Magnesium
- Al: Aluminium
- Si: Silicon
- K: Potassium
- Ca: Calcium
- Ba: Barium
- Fe: Iron
- Type of glass: (class attribute)
 - 1 building_windows_float_processed
 - 2 building_windows_non_float_processed
 - 3 vehicle_windows_float_processed
 - 4 vehicle_windows_non_float_processed (none in this database)
 - 5 containers
 - 6 tableware
 - 7 headlamps

The objective for me is to conduct an analysis using a neural network with the following dataset is as follows:

- To perform EDA on the dataset and discuss findings and what relevance they have on the glass network I wish to create.
- Pre-processing data for the machine learning models and justification on how the chosen methods should help me.
- Create and implement a Dense Neural Network that will output the glass type classification.
- Test and try to improve the accuracy of the neural network models using various methods and parameters.
- Make a Classification using the Final choice of model and validate the results.

```
In [1]: import numpy as np
import pandas as pd

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow import optimizers

from sklearn import metrics
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split

from sklearn import preprocessing
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier

import matplotlib
from matplotlib import pyplot as plt
matplotlib.use('agg')

import warnings
warnings.filterwarnings('ignore', category=FutureWarning)
warnings.filterwarnings('ignore', category=DeprecationWarning)

from keras.models import Sequential
from keras.layers import Dense
from keras.utils import np_utils

import xgboost as xgb
from xgboost import XGBClassifier
```

Libraries -

I imported multiple libraries for the following:

- Pandas - For creating a data frame.
- Keras and tensorflow - For creating machine learning and Neural Network Models, along with multiple parameters that I would use for my test models.
- Numpy - To convert the data to a numpy array.
- Sklearn - For pre-processing data and checking accuracy, metrics among others.
- Matplotlib - For creating plots.
- xgboost - For Xgboost test and classification
- Warnings - To suppress warnings.

1. Exploratory Data Analysis

Creating a variable to store the column names for EDA analysis

```
In [2]: column_names = ['Id','Ri','Na','Mg','Al','Si','K','Ca','Ba','Fe','Glass Type']

Reading the file and creating a data frame
```

```
In [8]: data = pd.read_csv('glass_data.csv',names = column_names,index_col = 0,header = None)
data.head()
```

```
Out [8]:
```

	Ri	Na	Mg	Al	Si	K	Ca	Ba	Fe	Glass Type
Id										
1	1.52101	13.64	4.49	1.10	71.78	0.08	8.75	0.0	0.0	1
2	1.51761	13.89	3.60	1.36	72.93	0.48	7.83	0.0	0.0	1
3	1.51618	13.53	3.56	1.54	72.99	0.39	7.78	0.0	0.0	1
4	1.51766	13.21	3.69	1.29	72.61	0.57	8.22	0.0	0.0	1
5	1.51742	13.27	3.62	1.24	73.08	0.55	8.07	0.0	0.0	1

I used the ID Index column data as index for the created data frame, this is due to the fact that the ID data serves no purpose for predicting the output target glass types in classification or neural network models that I will create further in my analysis.

Check the shape of data

```
In [4]: data.shape

Out [4]: (214, 10)
```

Check null values and data types within the dataset

```
In [5]: data.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 214 entries, 1 to 214
Data columns (total 10 columns):
 #   Column      Non-Null Count  Dtype
---  ---
 0   Id          214 non-null    float64
 1   Ri          214 non-null    float64
 2   Na          214 non-null    float64
 3   Mg          214 non-null    float64
 4   Al          214 non-null    float64
 5   Si          214 non-null    float64
 6   Ca          214 non-null    float64
 7   Ba          214 non-null    float64
 8   Fe          214 non-null    float64
 9   Glass Type  214 non-null    int64
dtypes: float64(9), int64(1)
memory usage: 18.4 KB
```

Check the data for value spread and distribution of values for mean and min-max between the columns of independent variables

```
In [6]: data.describe()

Out [6]:
```

	Ri	Na	Mg	Al	Si	K	Ca	Ba	Fe	Glass Type
count	214.000000	214.000000	214.000000	214.000000	214.000000	214.000000	214.000000	214.000000	214.000000	214.000000
mean	1.518365	13.407850	2.684533	1.444907	72.850935	0.497056	8.956963	0.175047	0.057009	2.780374
std	0.003037	0.816064	1.442408	0.496207	0.774546	0.652192	1.423153	0.497219	0.067439	2.103739
min	1.511150	10.730000	0.000000	0.296000	69.810000	0.000000	5.430000	0.000000	0.000000	1.000000
25%	1.516523	12.907500	2.115000	1.190000	72.280000	0.122500	8.240000	0.000000	0.000000	1.000000
50%	1.517680	13.300000	3.480000	1.360000	72.790000	0.555000	8.600000	0.000000	0.000000	2.000000
75%	1.519157	13.825000	3.600000	1.630000	73.087500	0.610000	9.172500	0.000000	0.100000	3.000000
max	1.533930	17.380000	4.490000	3.500000	75.410000	6.210000	16.190000	3.150000	0.510000	7.000000

The data values show no negative values in range, while the mean values for silicon and sodium columns show highest. The independent columns hold float values and so it would be good to scale the data to apply a normal distribution of values for my neural network model to learn and predict more accurately.

```
In [7]: data['Glass Type'].unique()

Out [7]: array([1, 2, 3, 5, 6, 7], dtype=int64)
```

Unique values from the target column 'Glass Type' reveal the glass type 4 data does not exist in the dataset, as described in the class assessment brief.

```
In [8]: Glass_types_sum = data['Glass Type'].value_counts()
Glass_types_sum
```

```
Out [8]:
```

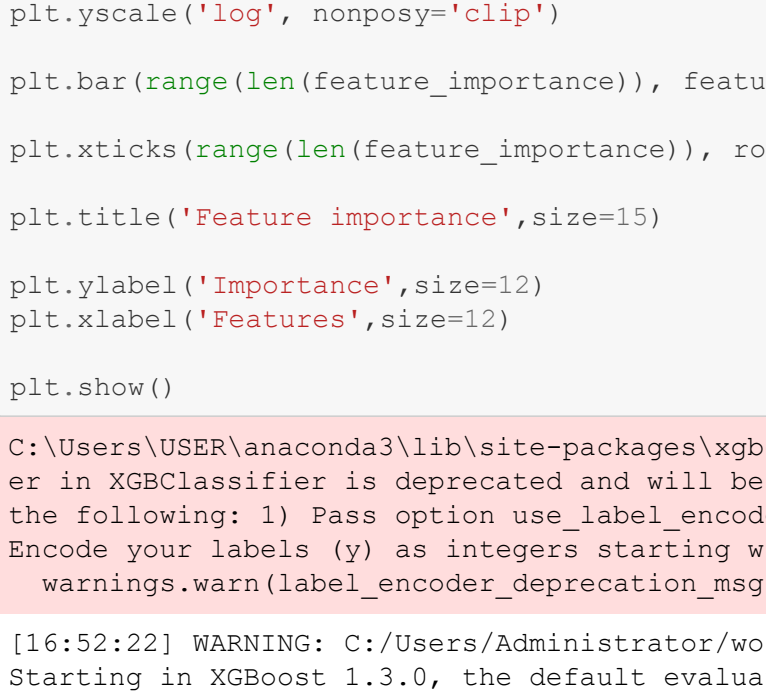
2	76
7	70
3	29
5	17
6	13
1	9

Name: Glass Type, dtype: int64

```
In [9]: data['Glass Type'].value_counts().plot(kind='bar')

plt.xlabel("Glass Types", labelpad=14)
plt.ylabel("Count", labelpad=10)

Out [9]: Text(0, 0.5, 'Count')
```

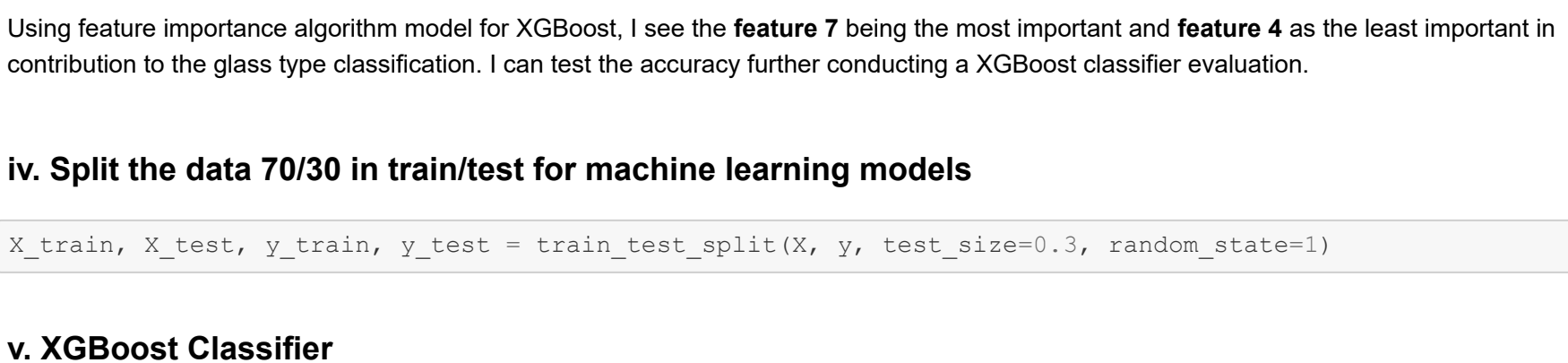


There are six types of glass in the dataset, it may be correct in assuming the data is imbalanced for categorical classification glass types with the most prominent type being 'Type 2' and least being 'Type 6'.

Check Correlation of Glass dataset

```
In [10]: import seaborn as sns

plt.figure(figsize=(12, 6))
corr=data.corr()
sns.heatmap(corr,cmap='viridis',annot=True)
```



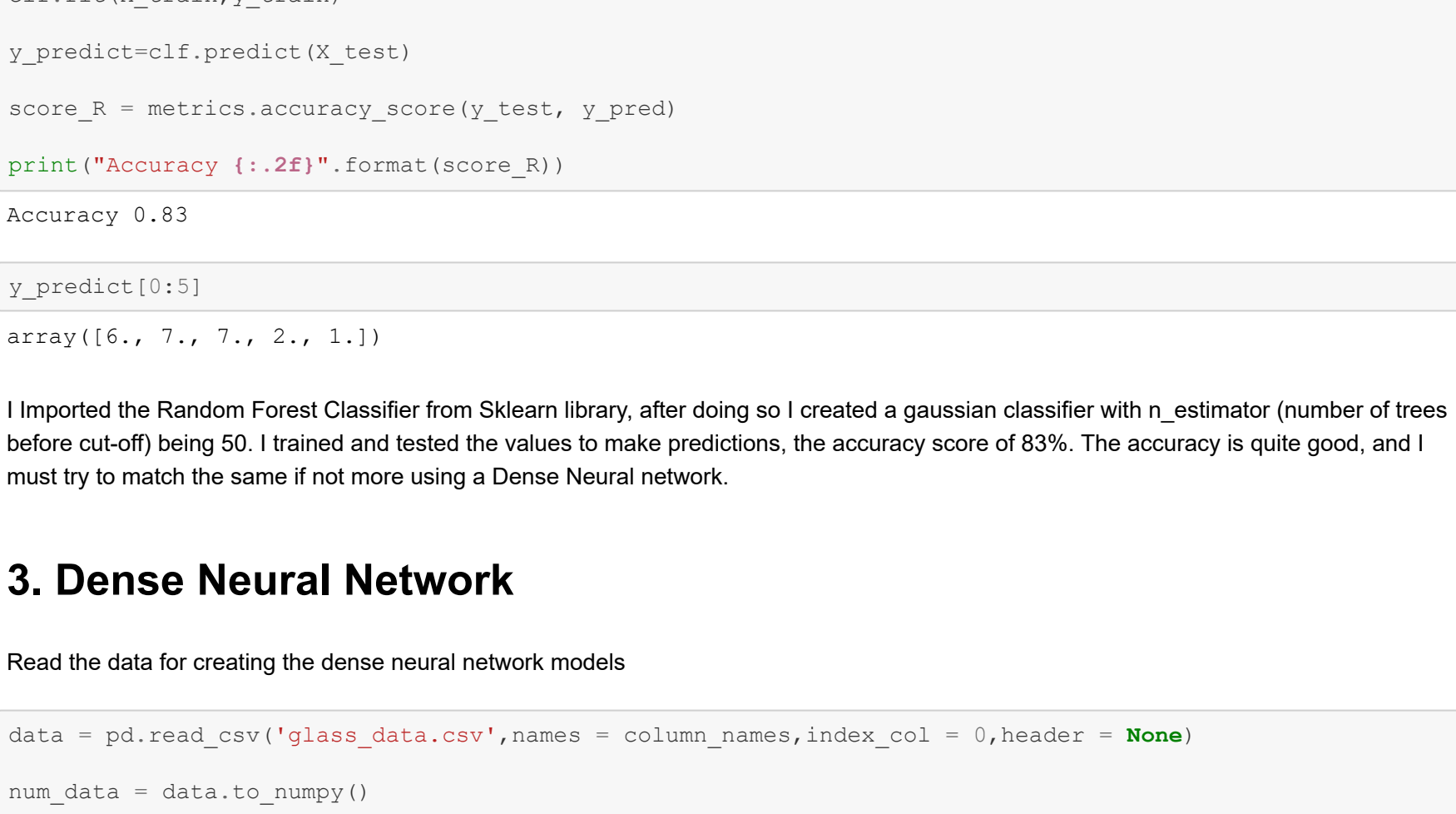
There is high correlation between Calcium and Refractive index showing 81%, also some low correlation among Barium, Aluminium and Potassium. The least correlating feature being Magnesium in the glass dataset excluding the target variable Glass type.

```
In [11]: plt.figure(figsize=(10, 5))

cmap= sns.diverging_palette(150, 275, s=80, l=55, n=5)

heatmap = sns.heatmap(data.corr()[['Glass Type']],
                        sort_values(by='Glass Type', ascending=False),
                        vmin=-1, vmax=1, annot=True, cmap=cmap)

heatmap.set_title('Features correlating to Glass Type', fontdict={'fontsize':15}, pad=16);
```



Sodium, Calcium and Barium is showing moderate to good correlations with the target variable Glass type.

Preliminary report for EDA

The dataset shares float values for all independent variables within the dataset, the target variable consists of glass type values as integers and no data records for glass type 4 form the dataset.

Due the nature of how the nature of how the integers values are assigned to glass data type I will need encode glass type column to individual categorical units and convert them in to a 1 dimensional array for my neural network.

I can verify the method further by cross validating the data with other algorithms, using the target variable with no One-hot or categorical encoding.

- Pre-processing data for the machine learning models and justification on how the chosen methods should help me.
- Create and implement a Dense Neural Network that will output the glass type classification.
- Test and try to improve the accuracy of the model using various methods, parameters in its accuracy.
- Make a Classification using the Final choice of model and validate the results.

2. Preprocess the data to fit the machine learning models and evaluation with algorithms

i. Converting the existing dataframe to numpy for Machine learning model going forward

```
In [12]: data_num = data.to_numpy()

In [13]: X = data_num[:,0:9]
y = data_num[:,9]
```

```
In [14]: y[150:170]

Out [14]: array([3., 3., 3., 3., 3., 3., 3., 3., 3., 3., 3., 3., 5., 5., 5., 5.,
                5., 5., 5.])
```

ii. Scaling the variables

```
In [15]: scaler = MinMaxScaler()

dataset1 = data.values

X = dataset1[:,0:9]
y = dataset1[:,9]

X = scaler.fit_transform(X)
```

iii. Using XGBoost to find the feature importance

```
In [16]: model = XGBClassifier(n_estimators=50)

model.fit(X, y)

feature_importance = model.feature_importances_

plt.figure(figsize=(10, 6))

plt.yscale('log', nonposy='clip')

plt.bar(range(len(feature_importance)), feature_importance, align='center')

plt.xticks(range(len(feature_importance)), rotation='horizontal')

plt.title('Feature importance',size=15)

plt.ylabel('Importance',size=12)
plt.xlabel('Features',size=12)

plt.show()
```

C:\Users\USER\anaconda3\lib\site-packages\xgboost\sklearn.py:888: UserWarning: The use of label encoder in XGBClassifier is deprecated and will be removed in a future release. To remove this warning, do the following: 1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., (num_class - 1). warnings.warn(label_encoder.deprecation_msg, UserWarning)

[16:52:22] WARNING: C:\Users\Administrator\workspace\xgboost\win64_release_1.3.0/src\learner.cc:1061: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'multi:softprob' was changed from 'error' to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the old behavior.



Using feature importance algorithm model for XGBoost, I see the feature 7 being the most important and feature 4 as the least important in contribution to the glass type classification. I can test the accuracy further conducting a XGBoost classifier evaluation.

iv. Split the data 70/30 in train/test for machine learning models

```
In [17]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)
```

v. XGBoost Classifier

```
In [18]: xgb = XGBClassifier(max_depth = 6, n_estimators = 100, learning_rate = 0.07, random_state = 2)

xgb.fit(X_train,y_train)

xgb_pred = xgb.predict(X_test)

y_pred = metrics.accuracy_score(y_test, xgb_pred)
```

[16:52:23] WARNING: C:\Users\Administrator\workspace\xgboost\win64_release_1.3.0/src\learner.cc:1061: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'multi:softprob' was changed from 'error' to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the old behavior.

```
In [19]: print ('The Accuracy for XGBoost classifier :',xgb_accuracy)

The Accuracy for XGBoost classifier : 0.8307692307692308
```

I achieved an accuracy score of 83% approximately after choosing max_depth = 6 as default value and a learning rate of 0.07, it is an impressive accuracy percentage with the help of regularization and tree pruning methods the XGBoost method offers. It is no surprise as a prominent algorithm in the industry for data analysis.

vi. Check accuracy with Random Forest

```
In [20]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)

clf=RandomForestClassifier(n_estimators=50)

clf.fit(X_train,y_train)

y_pred=clf.predict(X_test)

score_R = metrics.accuracy_score(y_test, y_pred)

print("Accuracy {:.2f}".format(score_R))

Accuracy 0.83
```

```
In [21]: y_predict[0:5]

Out [21]: array([6., 7., 7., 2., 1.])
```

I Imported the Random Forest Classifier from Sklearn library, after doing so I created a gaussian classifier with n_estimator (number of trees before cut-off) being 50. I trained and tested the values to make predictions, the accuracy score of 83%. The accuracy is quite good, and I must try to match the same if not more using a Dense Neural network.

3. Dense Neural Network

Read the data for creating the dense neural network models

```
In [22]: data = pd.read_csv('glass_data.csv',names = column_names,index_col = 0,header = None)

num_data = data.to_numpy()

dataset1 = num_data

X_cols = dataset1[:,0:9].astype(float)

y_cols = dataset1[:,9]

preprocessing.scale(X_cols, copy=False)

encoder = preprocessing.LabelEncoder()

encoder.fit(y_cols)

y_train = encoder.transform(y_cols).reshape(-1, 1)

y_enc = np_utils.to_categorical(y_train)

y_enc,X_cols
```

```
Out [22]: (array([[1., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0.],
       [1., 0., 0., 0., 0., 0., 0., 0., 0.],
       ...,
       [0., 0., 0., 0., 0., 1., 1.],
       [0., 0., 0., 0., 0., 1.]], dtype=float32),
 array([[ 0.87286785, -0.26495326, 1.25463857, ..., -0.14576634,
        -0.35287683, -0.5864509 ,
        -0.24933347,  0.59181718,  0.63616803, ..., -0.79373376,
        -0.35287683, -0.5864509 ,
        -0.72131606,  0.14993314,  0.60142249, ..., -0.82894938,
        -0.35287683, -0.5864509 ,
        ...,
        [ 0.75404635,  1.16872135, -1.86551055, ..., -0.36410319,
        2.95320036, -0.5864509 ,
        -0.61239854,  1.19327046, -1.86551055, ..., -0.33593069,
        2.81208731, -0.5864509 ,
        -0.41436305,  1.00913221, -1.86551055, ..., -0.23732695,
        3.01367739, -0.5864509 ]]))
```

Review

Pre-processing the data to fit the ANN models was challenging. I converted the values in target column with one hot encoding and later converting it to categorical, doing so I am able to pass in the values of target 'Glass Type' column without causing an imbalance in the data. Glass type 4 is non-existent in the data frame and also to use categorical_crossentropy as my primary loss function initially, it enables to get the categorical values a higher learning rate.

Later I will need to evaluate the learning rates and epochs/batch sizes for the models to check the models fit.

I will explore multiple variations using activation functions, loss functions, number of epochs and batch sizes to improve the model and finding meaningful results.

ANN TEST MODELS

Model 1

Model creation parameters

- Sequential Model**
 - I will start by creating 9 input layers, along with 3 hidden dense layers.
- Test size = 0.3** a 70/30 split for train and testing the dataset at the beginning.
- random_state = 0** set as 0 initially and will evaluate as I progress through the models.
- kernel_initializer = 'normal'** for normal distribution of weights through the neural layers.
- Activation = 'relu'** for linear activation initially, serves a costless method for achieving a good learning rate without impeding the slope or causing to vanishing gradient.
- Final Activation = 'softmax'** for classification of glass types and predicting the probabilities with last layer of logits.
- Optimizer = 'adam'** to reduce the loss and adjust weights in model training and fast convergence with each iteration.


```
Epoch 395/400
15/15 - 0s - loss: 0.0991 - accuracy: 0.9799
Epoch 396/400
15/15 - 0s - loss: 0.0983 - accuracy: 0.9799
Epoch 397/400
15/15 - 0s - loss: 0.0991 - accuracy: 0.9732
Epoch 398/400
15/15 - 0s - loss: 0.0984 - accuracy: 0.9799
Epoch 399/400
15/15 - 0s - loss: 0.0985 - accuracy: 0.9799
Epoch 400/400
15/15 - 0s - loss: 0.0975 - accuracy: 0.9799
3/3 [=====] - 0s 2ms/step - loss: 1.4907 - accuracy: 0.6769

Out[23]: [1.4906502962112427, 0.6769230961799622]

In [24]: _, train_acc = trial_model_1.evaluate(X_train, y_train, verbose=0)
         _, test_acc = trial_model_1.evaluate(X_test, y_test, verbose=0)

         print ('Train:%.3f, Test:%.3f' % (train_acc, test_acc))

Train:0.980, Test:0.677
```


Train and test Accuracy Scores from multiple model test runs

1. Train:0.649, Test:0.558
2. Train:0.649, Test:0.558
3. Train:0.626, Test:0.465
4. Train:0.620, Test:0.488

Change in Epochs and batch size

1. Train:0.883, Test:0.605
2. Train:0.973, Test:0.708
3. Train:0.966, Test:0.769
4. Train:0.966, Test:0.708

Initial test run shows a training accuracy between 62% - 64% and testing accuracy between 46% - 55%. I will try to improve the model further to reduce under or over fitting the data and tune the model further.

Notes

1. After changing the values of epochs and batch size I am getting high accuracy in training and low testing accuracy, this could be a case of overfitting the data.
2. There could be discriminant values among the test data that may impact the learning rate, I will try to introduce random_state, custom learning rates with higher epochs/batch sizes to achieve the predictions.
3. The loss does seem to reduce over time through the number of epochs but the accuracy gap between the training and testing data proves its not a good fit and may need to try other parameters or methods to get the accuracy higher and reduce loss in errors.

Model 2

- **Sequential Model**
- I have added 9 input layers, along with 3 hidden dense layers of 12,7 and 6.
- **Test size = 0.3** a 70/30 split for training and testing the dataset at the beginning.
- **random_state = 2** set as 2 to evaluate the randomness between the values with various magnitudes and to not be skewed to one side while the model learns through the training set.
- **kernel_initializer = 'normal'** for normal distribution of weights through the neural layers.
- **Activation = 'relu'** for linear activation initially, serves a costless method for achieving a good learning rate without impeding the slope or causing to vanishing gradient.
- **Final Activation = 'softmax'** for classification of glass types and predicting the probabilities with last layer of logits.
- **Optimizer = adam** to reduce the loss and adjust weights in model training and fast convergence with each iteration.
- **Batch size = 80, Epochs=1200**
- **Loss = 'MSE'** - As Mean squared error is set to try and leverage the loss over the number of epochs by squaring the values and adding the errors to obtain the mean among the errors.

2/2 [=====]	-	0s	4ms/step	-	loss: 0.0853	-	accuracy: 0.6849
2/2 [=====]	-	0s	2ms/step	-	loss: 0.0827	-	accuracy: 0.6974
Epoch 413/1200							
2/2 [=====]	-	0s	3ms/step	-	loss: 0.0851	-	accuracy: 0.6769
2/2 [=====]	-	0s	6ms/step	-	loss: 0.0816	-	accuracy: 0.7061
Epoch 415/1200							
2/2 [=====]	-	0s	3ms/step	-	loss: 0.0847	-	accuracy: 0.6769
2/2 [=====]	-	0s	5ms/step	-	loss: 0.0854	-	accuracy: 0.6727
Epoch 416/1200							
2/2 [=====]	-	0s	7ms/step	-	loss: 0.0846	-	accuracy: 0.6852
2/2 [=====]	-	0s	6ms/step	-	loss: 0.0856	-	accuracy: 0.6769
Epoch 418/1200							
2/2 [=====]	-	0s	4ms/step	-	loss: 0.0833	-	accuracy: 0.6811
Epoch 420/1200							
2/2 [=====]	-	0s	3ms/step	-	loss: 0.0853	-	accuracy: 0.6686
2/2 [=====]	-	0s	3ms/step	-	loss: 0.0853	-	accuracy: 0.6855
2/2 [=====]	-	0s	4ms/step	-	loss: 0.0839	-	accuracy: 0.6855
0.0852 - accuracy: 0.6730							
2/2 [=====]	-	ETA: 0s -	loss: 0.0858 - accuracy: 0.65 -	0s	3ms/step	-	loss: 0.0828 - accuracy: 0.6852
2/2 [=====]	-	0s	7ms/step	-	loss: 0.0861	-	accuracy: 0.6689
Epoch 426/1200							
2/2 [=====]	-	0s	5ms/step	-	loss: 0.0842	-	accuracy: 0.6772
2/2 [=====]	-	0s	4ms/step	-	loss: 0.0870	-	accuracy: 0.6814
Epoch 428/1200							
2/2 [=====]	-	0s	6ms/step	-	loss: 0.0832	-	accuracy: 0.6980
2/2 [=====]	-	0s	4ms/step	-	loss: 0.0851	-	accuracy: 0.6980
Epoch 429/1200							
2/2 [=====]	-	0s	5ms/step	-	loss: 0.0895	-	accuracy: 0.6689
Epoch 431/1200							
2/2 [=====]	-	0s	5ms/step	-	loss: 0.0878	-	accuracy: 0.6647
2/2 [=====]	-	0s	4ms/step	-	loss: 0.0840	-	accuracy: 0.6730
Epoch 433/1200							
2/2 [=====]	-	0s	10ms/step	-	loss: 0.0835	-	accuracy: 0.6897
Epoch 434/1200							
2/2 [=====]	-	0s	5ms/step	-	loss: 0.0848	-	accuracy: 0.6727
Epoch 435/1200							
2/2 [=====]	-	0s	5ms/step	-	loss: 0.0841	-	accuracy: 0.6897
Epoch 436/1200							
2/2 [=====]	-	0s	4ms/step	-	loss: 0.0844	-	accuracy: 0.6855
2/2 [=====]	-	0s	3ms/step	-	loss: 0.0801	-	accuracy: 0.7105
Epoch 438/1200							
2/2 [=====]	-	0s	5ms/step	-	loss: 0.0829	-	accuracy: 0.6727
Epoch 439/1200							
2/2 [=====]	-	0s	4ms/step	-	loss: 0.0859	-	accuracy: 0.6686
Epoch 440/1200							
2/2 [=====]	-	0s	4ms/step	-	loss: 0.0884	-	accuracy: 0.6727
Epoch 441/1200							
2/2 [=====]	-	0s	2ms/step	-	loss: 0.0840	-	accuracy: 0.6769
2/2 [=====]	-	0s	3ms/step	-	loss: 0.0856	-	accuracy: 0.6644
Epoch 443/1200							
2/2 [=====]	-	0s	7ms/step	-	loss: 0.0851	-	accuracy: 0.6602
Epoch 444/1200							
2/2 [=====]	-	0s	3ms/step	-	loss: 0.0831	-	accuracy: 0.6811
Epoch 445/1200							
0.0828 - accuracy: 0.6852							
2/2 [=====]	-	ETA: 0s -	loss: 0.0810 - accuracy: 0.70 -	0s	7ms/step	-	loss: 0.0828 - accuracy: 0.6852
Epoch 446/1200							
2/2 [=====]	-	0s	4ms/step	-	loss: 0.0856	-	accuracy: 0.6686
Epoch 447/1200							
2/2 [=====]	-	0s	6ms/step	-	loss: 0.0807	-	accuracy: 0.6894
Epoch 448/1200							
2/2 [=====]	-	0s	3ms/step	-	loss: 0.0829	-	accuracy: 0.6852
Epoch 449/1200							
2/2 [=====]	-	0s	3ms/step	-	loss: 0.0847	-	accuracy: 0.6644
2/2 [=====]	-	0s	8ms/step	-			

Bpoch 832/1200	2/2	=====	-	0s	7ms/step	-	loss: 0.0711	-	accuracy: 0.7249
Bpoch 833/1200	2/2	=====	-	0s	2ms/step	-	loss: 0.0715	-	accuracy: 0.7207
Bpoch 834/1200	2/2	=====	-	0s	5ms/step	-	loss: 0.0685	-	accuracy: 0.7374
Bpoch 835/1200	2/2	=====	-	0s	5ms/step	-	loss: 0.0743	-	accuracy: 0.6957
Bpoch 837/1200	2/2	=====	-	0s	3ms/step	-	loss: 0.0688	-	accuracy: 0.7374
Bpoch 838/1200	2/2	=====	-	0s	7ms/step	-	loss: 0.0724	-	accuracy: 0.7124
Bpoch 839/1200	2/2	=====	-	0s	8ms/step	-	loss: 0.0659	-	accuracy: 0.7457
Bpoch 840/1200	2/2	=====	-	0s	2ms/step	-	loss: 0.0688	-	accuracy: 0.7249
Bpoch 841/1200	2/2	=====	-	0s	5ms/step	-	loss: 0.0699	-	accuracy: 0.7249
Bpoch 842/1200	2/2	=====	-	0s	6ms/step	-	loss: 0.0712	-	accuracy: 0.7249
Bpoch 843/1200	2/2	=====	-	0s	4ms/step	-	loss: 0.0694	-	accuracy: 0.7207
Bpoch 844/1200	2/2	=====	-	0s	6ms/step	-	loss: 0.0693	-	accuracy: 0.7207
Bpoch 845/1200	2/2	=====	-	0s	6ms/step	-	loss: 0.0722	-	accuracy: 0.7124
Bpoch 847/1200	2/2	=====	-	0s	4ms/step	-	loss: 0.0688	-	accuracy: 0.7249
Bpoch 848/1200	2/2	=====	-	0s	6ms/step	-	loss: 0.0688	-	accuracy: 0.7374
Bpoch 849/1200	2/2	=====	-	0s	3ms/step	-	loss: 0.0708	-	accuracy: 0.7166
Bpoch 850/1200	2/2	=====	-	0s	5ms/step	-	loss: 0.0671	-	accuracy: 0.7416
Bpoch 851/1200	2/2	=====	-	0s	5ms/step	-	loss: 0.0693	-	accuracy: 0.7124
Bpoch 852/1200	2/2	=====	-	0s	4ms/step	-	loss: 0.0679	-	accuracy: 0.7374
Bpoch 853/1200	2/2	=====	-	0s	4ms/step	-	loss: 0.0704	-	accuracy: 0.7207
Bpoch 855/1200	2/2	=====	-	0s	4ms/step	-	loss: 0.0671	-	accuracy: 0.7291
Bpoch 857/1200	2/2	=====	-	0s	6ms/step	-	loss: 0.0672	-	accuracy: 0.7332
Bpoch 859/1200	2/2	=====	-	0s	4ms/step	-	loss: 0.0715	-	accuracy: 0.7082
Bpoch 860/1200	2/2	=====	-	0s	3ms/step	-	loss: 0.0664	-	accuracy: 0.7374
Bpoch 861/1200	2/2	=====	-	0s	6ms/step	-	loss: 0.0668	-	accuracy: 0.7374
Bpoch 862/1200	2/2	=====	-	0s	3ms/step	-	loss: 0.0681	-	accuracy: 0.7249
Bpoch 863/1200	2/2	=====	-	0s	4ms/step	-	loss: 0.0692	-	accuracy: 0.7207
Bpoch 864/1200	2/2	=====	-	0s	5ms/step	-	loss: 0.0690	-	accuracy: 0.7207
Bpoch 865/1200	2/2	=====	-	0s	6ms/step	-	loss: 0.0680	-	accuracy: 0.7291
Bpoch 866/1200	2/2	=====	-	0s	4ms/step	-	loss: 0.0681	-	accuracy: 0.7291
Bpoch 867/1200	2/2	=====	-	0s	8ms/step	-	loss: 0.0685	-	accuracy: 0.7249
Bpoch 868/1200	2/2	=====	-	0s	6ms/step	-	loss: 0.0698	-	accuracy: 0.7166
Bpoch 869/1200	2/2	=====	-	0s	5ms/step	-	loss: 0.0669	-	accuracy: 0.7416
Bpoch 870/1200	2/2	=====	-	0s	4ms/step	-	loss: 0.0706	-	accuracy: 0.7041
Bpoch 871/1200	2/2	=====	-	0s	6ms/step	-	loss: 0.0721	-	accuracy: 0.7082
Bpoch 872/1200	2/2	=====	-	0s	4ms/step	-	loss: 0.0713	-	accuracy: 0.7166
Bpoch 873/1200	2/2	=====	-	0s	6ms/step	-	loss: 0.0701	-	accuracy: 0.7124
Bpoch 874/1200	2/2	=====	-	0s	3ms/step	-	loss: 0.0716	-	accuracy: 0.7082
Bpoch 875/1200	2/2	=====	-	0s	3ms/step	-	loss: 0.0657	-	accuracy: 0.7416
Bpoch 876/1200	2/2	=====	-	0s	4ms/step	-	loss: 0.0683	-	accuracy: 0.7166
Bpoch 877/1200	2/2	=====	-	0s	4ms/step	-	loss: 0.0685	-	accuracy: 0.7249
Bpoch 878/1200	2/2	=====	-	0s	7ms/step	-	loss: 0.0663	-	accuracy: 0.7374
Bpoch 879/1200	2/2	=====	-	0s	4ms/step	-	loss: 0.0671	-	accuracy: 0.7332
Bpoch 880/1200	2/2	=====	-	0s	4ms/step	-	loss: 0.0673	-	accuracy: 0.7332

```
2/2 [=====] - 0s 4ms/step - loss: 0.0640 - accuracy: 0.7422
Epoch 1098/1200
2/2 [=====] - 0s 6ms/step - loss: 0.0654 - accuracy: 0.7380
Epoch 1099/1200
2/2 [=====] - 0s 4ms/step - loss: 0.0636 - accuracy: 0.7463
Epoch 1100/1200
2/2 [=====] - 0s 5ms/step - loss: 0.0629 - accuracy: 0.7422
Epoch 1101/1200
```

```

2/2 [=====] - 0s 5ms/step - loss: 0.0645 - accuracy: 0.7380
Epoch 1102/1200
2/2 [=====] - 0s 4ms/step - loss: 0.0621 - accuracy: 0.7422
Epoch 1103/1200
2/2 [=====] - 0s 6ms/step - loss: 0.0635 - accuracy: 0.7380
Epoch 1104/1200
2/2 [=====] - 0s 4ms/step - loss: 0.0636 - accuracy: 0.7380
Epoch 1105/1200
2/2 [=====] - 0s 4ms/step - loss: 0.0634 - accuracy: 0.7380

```



2/2 [-----]
Epoch 1119/1200

- ```
2/2 [=====] -
Epoch 1120/1200
2/2 [=====] -
Epoch 1121/1200
2/2 [=====] -
Epoch 1122/1200
2/2 [=====] -
Epoch 1123/1200
```

2/2 (=====)  
Epoch 1124/1200

- ```

2/2 [=====] - 0s 6ms/step - loss: 0.0
Epoch 1125/1200
2/2 [=====] - 0s 3ms/step - loss: 0.0
Epoch 1126/1200
2/2 [=====] - 0s 3ms/step - loss: 0.0
Epoch 1127/1200
2/2 [=====] - 0s 3ms/step - loss: 0.0
Epoch 1128/1200

```

```
2/2 [=====]
Epoch 1129/1200
```

- ```

2/2 [=====] - 0s 3ms/step - loss: 0.0
Epoch 1130/1200
2/2 [=====] - 0s 3ms/step - loss: 0.0
Epoch 1131/1200
2/2 [=====] - 0s 7ms/step - loss: 0.0
Epoch 1132/1200
2/2 [=====] - 0s 3ms/step - loss: 0.0
Epoch 1133/1200
2/2 [=====] - 0s 3ms/step - loss: 0.0

```

## Epoch

```

2/2 [=====] - 0s 5ms/step - loss: 0.0613 - accuracy: 0.7422
Epoch 1135/1200
2/2 [=====] - 0s 3ms/step - loss: 0.0601 - accuracy: 0.7505
Epoch 1136/1200
2/2 [=====] - 0s 4ms/step - loss: 0.0630 - accuracy: 0.7380
Epoch 1137/1200
2/2 [=====] - 0s 7ms/step - loss: 0.0643 - accuracy: 0.7297
Epoch 1138/1200
2/2 [=====] - 0s 3ms/step - loss: 0.0633 - accuracy: 0.7338

```

```
Epoch 1139/1200
2/2 [=====] - 0s 5ms/step - loss: 0.0634 - accuracy: 0.7150
Epoch 1140/1200
2/2 [=====] - 0s 5ms/step - loss: 0.0634 - accuracy: 0.7297
Epoch 1141/1200
2/2 [=====] - 0s 6ms/step - loss: 0.0612 - accuracy: 0.7422
Epoch 1142/1200
2/2 [=====] - 0s 3ms/step - loss: 0.0654 - accuracy: 0.7172
Epoch 1143/1200
2/2 [=====] - 0s 5ms/step - loss: 0.0621 - accuracy: 0.7425
```

Epoch 1  
2/2 [==



```
[128]: X_train,X_test, y_train, y_test = train_test_split(X_cols, y_enc, test_size=0.3, random_state = 5)
trial_model_3 = Sequential()
trial_model_3.add(Dense(24, input_dim=9, kernel_initializer='normal', activation='relu'))
trial_model_3.add(Dense(12, kernel_initializer='normal', activation='relu'))
trial_model_3.compile(loss='MSE', optimizer='adam', metrics=['accuracy'])
History = trial_model_3.fit(X_train, y_train, epochs=600, batch_size=50, verbose=1)
trial_model_3.evaluate(X_test, y_test)

Epoch 2/600 - 0s 1ms/step - loss: 0.1386 - accuracy: 0.3145
Epoch 3/600 - 0s 2ms/step - loss: 0.1387 - accuracy: 0.3179
Epoch 4/600 - 0s 2ms/step - loss: 0.1385 - accuracy: 0.3721
Epoch 5/600 - 0s 2ms/step - loss: 0.1384 - accuracy: 0.4063
Epoch 6/600 - 0s 2ms/step - loss: 0.1382 - accuracy: 0.3796
Epoch 7/600 - 0s 2ms/step - loss: 0.1384 - accuracy: 0.3721
Epoch 8/600 - 0s 3ms/step - loss: 0.1379 - accuracy: 0.3479
Epoch 9/600 - 0s 2ms/step - loss: 0.1374 - accuracy: 0.3929
Epoch 10/600 - 0s 2ms/step - loss: 0.1371 - accuracy: 0.3779
Epoch 11/600 - 0s 4ms/step - loss: 0.1368 - accuracy: 0.3829
Epoch 12/600 - 0s 3ms/step - loss: 0.1362 - accuracy: 0.4129
Epoch 13/600 - 0s 2ms/step - loss: 0.1361 - accuracy: 0.3479
Epoch 14/600 - 0s 2ms/step - loss: 0.1356 - accuracy: 0.3579
Epoch 15/600 - 0s 2ms/step - loss: 0.1352 - accuracy: 0.3829
Epoch 16/600 - 0s 3ms/step - loss: 0.1344 - accuracy: 0.3554
Epoch 17/600 - 0s 2ms/step - loss: 0.1337 - accuracy: 0.3929
Epoch 18/600 - 0s 2ms/step - loss: 0.1329 - accuracy: 0.3779
Epoch 19/600 - 0s 2ms/step - loss: 0.1320 - accuracy: 0.3879
Epoch 20/600 - 0s 2ms/step - loss: 0.1313 - accuracy: 0.3754
Epoch 21/600 - 0s 2ms/step - loss: 0.1302 - accuracy: 0.3654
Epoch 22/600 - 0s 4ms/step - loss: 0.1288 - accuracy: 0.3379
Epoch 23/600 - 0s 4ms/step - loss: 0.1270 - accuracy: 0.3963
Epoch 24/600 - 0s 1ms/step - loss: 0.1257 - accuracy: 0.3880
Epoch 25/600 - 0s 2ms/step - loss: 0.1240 - accuracy: 0.4364
Epoch 26/600 - 0s 2ms/step - loss: 0.1223 - accuracy: 0.4673
Epoch 27/600 - 0s 2ms/step - loss: 0.1217 - accuracy: 0.4582
Epoch 28/600 - 0s 3ms/step - loss: 0.1192 - accuracy: 0.4899
Epoch 29/600 - 0s 2ms/step - loss: 0.1181 - accuracy: 0.4743
Epoch 30/600 - 0s 2ms/step - loss: 0.1163 - accuracy: 0.4741
Epoch 31/600 - 0s 2ms/step - loss: 0.1148 - accuracy: 0.5116
Epoch 32/600 - 0s 2ms/step - loss: 0.1137 - accuracy: 0.4541
Epoch 33/600 - 0s 2ms/step - loss: 0.1121 - accuracy: 0.4891
Epoch 34/600 - 0s 2ms/step - loss: 0.1118 - accuracy: 0.4766
Epoch 35/600 - 0s 4ms/step - loss: 0.1095 - accuracy: 0.4891
Epoch 36/600 - 0s 4ms/step - loss: 0.1097 - accuracy: 0.4841
Epoch 37/600 - 0s 3ms/step - loss: 0.1085 - accuracy: 0.4966
Epoch 38/600 - 0s 2ms/step - loss: 0.1065 - accuracy: 0.4966
Epoch 39/600 - 0s 2ms/step - loss: 0.1072 - accuracy: 0.4966
Epoch 40/600 - 0s 2ms/step - loss: 0.1072 - accuracy: 0.4991
Epoch 41/600 - 0s 5ms/step - loss: 0.1064 - accuracy: 0.4916
Epoch 42/600 - 0s 2ms/step - loss: 0.1041 - accuracy: 0.4616
Epoch 43/600 - 0s 3ms/step - loss: 0.1027 - accuracy: 0.4966
Epoch 44/600 - 0s 2ms/step - loss: 0.1018 - accuracy: 0.5008
Epoch 45/600 - 0s 2ms/step - loss: 0.0985 - accuracy: 0.5000
Epoch 46/600 - 0s 2ms/step - loss: 0.1013 - accuracy: 0.5058
Epoch 47/600 - 0s 3ms/step - loss: 0.0996 - accuracy: 0.5242
Epoch 48/600 - 0s 2ms/step - loss: 0.0968 - accuracy: 0.5450
Epoch 49/600 - 0s 2ms/step - loss: 0.0991 - accuracy: 0.5117
Epoch 50/600 - 0s 2ms/step - loss: 0.0988 - accuracy: 0.5585
Epoch 51/600 - 0s 2ms/step - loss: 0.1008 - accuracy: 0.5852
Epoch 52/600 - 0s 2ms/step - loss: 0.0959 - accuracy: 0.6053
Epoch 53/600 - 0s 2ms/step - loss: 0.0954 - accuracy: 0.6246
Epoch 54/600 - 0s 4ms/step - loss: 0.0941 - accuracy: 0.6412
Epoch 55/600 - 0s 3ms/step - loss: 0.0932 - accuracy: 0.6329
Epoch 56/600 - 0s 3ms/step - loss: 0.0932 - accuracy: 0.6346
Epoch 57/600 - 0s 2ms/step - loss: 0.0919 - accuracy: 0.6738
Epoch 58/600 - 0s 2ms/step - loss: 0.0909 - accuracy: 0.6438
Epoch 59/600 - 0s 3ms/step - loss: 0.0935 - accuracy: 0.6321
Epoch 60/600 - 0s 3ms/step - loss: 0.0923 - accuracy: 0.6496
Epoch 61/600 - 0s 2ms/step - loss: 0.0889 - accuracy: 0.6721
Epoch 62/600 - 0s 5ms/step - loss: 0.0900 - accuracy: 0.6546
Epoch 63/600 - 0s 2ms/step - loss: 0.0894 - accuracy: 0.6446
Epoch 64/600 - 0s 2ms/step - loss: 0.0887 - accuracy: 0.6513
Epoch 65/600 - 0s 3ms/step - loss: 0.0873 - accuracy: 0.6538
Epoch 66/600 - 0s 3ms/step - loss: 0.0880 - accuracy: 0.6279
Epoch 67/600 - 0s 3ms/step - loss: 0.0894 - accuracy: 0.6254
Epoch 68/600 - 0s 3ms/step - loss: 0.0879 - accuracy: 0.6646
Epoch 69/600 - 0s 2ms/step - loss: 0.0903 - accuracy: 0.6238
Epoch 70/600 - 0s 2ms/step - loss: 0.0861 - accuracy: 0.6371
Epoch 71/600 - 0s 5ms/step - loss: 0.0851 - accuracy: 0.6646
Epoch 72/600 - 0s 5ms/step - loss: 0.0828 - accuracy: 0.6621
Epoch 73/600 - 0s 2ms/step - loss: 0.0868 - accuracy: 0.6421
Epoch 74/600 - 0s 2ms/step - loss: 0.0842 - accuracy: 0.6696
Epoch 75/600 - 0s 3ms/step - loss: 0.0855 - accuracy: 0.6530
Epoch 76/600 - 0s 3ms/step - loss: 0.0867 - accuracy: 0.6480
Epoch 77/600 - 0s 3ms/step - loss: 0.0791 - accuracy: 0.6705
Epoch 78/600 - 0s 2ms/step - loss: 0.0831 - accuracy: 0.6480
Epoch 79/600 - 0s 4ms/step - loss: 0.0794 - accuracy: 0.6721
Epoch 80/600 - 0s 2ms/step - loss: 0.0775 - accuracy: 0.6855
Epoch 81/600 - 0s 2ms/step - loss: 0.0785 - accuracy: 0.6380
Epoch 82/600 - 0s 2ms/step - loss: 0.0840 - accuracy: 0.6371
Epoch 83/600 - 0s 3ms/step - loss: 0.0826 - accuracy: 0.6796
Epoch 84/600 - 0s 2ms/step - loss: 0.0767 - accuracy: 0.6796
Epoch 85/600 - 0s 2ms/step - loss: 0.0776 - accuracy: 0.6521
Epoch 86/600 - 0s 2ms/step - loss: 0.0751 - accuracy: 0.6596
Epoch 87/600 - 0s 2ms/step - loss: 0.0788 - accuracy: 0.6605
Epoch 88/600 - 0s 2ms/step - loss: 0.0764 - accuracy: 0.6805
Epoch 89/600 - 0s 3ms/step - loss: 0.0774 - accuracy: 0.6597
Epoch 90/600 - 0s 2ms/step - loss: 0.0731 - accuracy: 0.6981
Epoch 91/600 - 0s 4ms/step - loss: 0.0772 - accuracy: 0.6764
Epoch 92/600 - 0s 3ms/step - loss: 0.0739 - accuracy: 0.6781
Epoch 93/600 - 0s 3ms/step - loss: 0.0758 - accuracy: 0.6656
Epoch 94/600 - 0s 3ms/step - loss: 0.0776 - accuracy: 0.6681
Epoch 95/600 - 0s 3ms/step - loss: 0.0758 - accuracy: 0.6684
Epoch 96/600 - 0s 2ms/step - loss: 0.0749 - accuracy: 0.6739
Epoch 97/600 - 0s 5ms/step - loss: 0.0718 - accuracy: 0.6989
Epoch 98/600 - 0s 3ms/step - loss: 0.0750 - accuracy: 0.6873
Epoch 99/600 - 0s 3ms/step - loss: 0.0716 - accuracy: 0.6948
Epoch 100/600 - 0s 3ms/step - loss: 0.0753 - accuracy: 0.6748
Epoch 101/600 - 0s 2ms/step - loss: 0.0704 - accuracy: 0.7073
Epoch 102/600 - 0s 4ms/step - loss: 0.0710 - accuracy: 0.6873
Epoch 103/600 - 0s 2ms/step - loss: 0.0750 - accuracy: 0.6623
Epoch 104/600 - 0s 4ms/step - loss: 0.0730 - accuracy: 0.6798
Epoch 105/600 - 0s 3ms/step - loss: 0.0705 - accuracy: 0.6923
Epoch 106/600 - 0s 2ms/step - loss: 0.0723 - accuracy: 0.7006
Epoch 107/600 - 0s 2ms/step - loss: 0.0743 - accuracy: 0.6673
Epoch 108/600 - 0s 4ms/step - loss: 0.0673 - accuracy: 0.7148
Epoch 109/600 - 0s 3ms/step - loss: 0.0692 - accuracy: 0.6898
Epoch 110/600 - 0s 3ms/step - loss: 0.0728 - accuracy: 0.6598
Epoch 111/600 - 0s 2ms/step - loss: 0.0676 - accuracy: 0.7376
Epoch 112/600 - 0s 4ms/step - loss: 0.0714 - accuracy: 0.7015
Epoch 113/600 - 0s 3ms/step - loss: 0.0704 - accuracy: 0.7015
Epoch 114/600 - 0s 4ms/step - loss: 0.0694 - accuracy: 0.7031
Epoch 115/600 - 0s 2ms/step - loss: 0.0723 - accuracy: 0.6781
Epoch 116/600 - 0s 3ms/step - loss: 0.0711 - accuracy: 0.6815
Epoch 117/600 - 0s 2ms/step - loss: 0.0723 - accuracy: 0.6781
Epoch 118/600 - 0s 3ms/step - loss: 0.0727 - accuracy: 0.6856
Epoch 119/600 - 0s 4ms/step - loss: 0.0726 - accuracy: 0.6706
Epoch 120/600 - 0s 3ms/step - loss: 0.0697 - accuracy: 0.6940
Epoch 121/600 - 0s 2ms/step - loss: 0.0679 - accuracy: 0.7015
Epoch 122/600 - 0s 3ms/step - loss: 0.0739 - accuracy: 0.6690
Epoch 123/600 - 0s 2ms/step - loss: 0.0655 - accuracy: 0.7165
Epoch 124/600 - 0s 3ms/step - loss: 0.0681 - accuracy: 0.7140
Epoch 125/600 - 0s 2ms/step - loss: 0.0663 - accuracy: 0.7065
Epoch 126/600 - 0s 3ms/step - loss: 0.0663 - accuracy: 0.7065
Epoch 127/600 - 0s 3ms/step - loss: 0.0715 - accuracy: 0.6665
Epoch 128/600 - 0s 3ms/step - loss: 0.0683 - accuracy: 0.6965
Epoch 129/600 - 0s 3ms/step - loss: 0.0624 - accuracy: 0.7240
Epoch 130/600 - 0s 3ms/step - loss: 0.0638 - accuracy: 0.6948
Epoch 131/600 - 0s 3ms/step - loss: 0.0656 - accuracy: 0.6990
Epoch 132/600 - 0s 4ms/step - loss: 0.0703 - accuracy: 0.6790
Epoch 133/600 - 0s 3ms/step - loss: 0.0655 - accuracy: 0.7198
Epoch 134/600 - 0s 3ms/step - loss: 0.0635 - accuracy: 0.7148
Epoch 135/600 - 0s 3ms/step - loss: 0.0663 - accuracy: 0.7157
Epoch 136/600 - 0s 4ms/step - loss: 0.0683 - accuracy: 0.7032
Epoch 137/600 - 0s 3ms/step - loss: 0.0659 - accuracy: 0.7141
Epoch 138/600 - 0s 4ms/step - loss: 0.0669 - accuracy: 0.7274
Epoch 139/600 - 0s 2ms/step - loss: 0.0638 - accuracy: 0.7324
Epoch 140/600 - 0s 3ms/step - loss: 0.0656 - accuracy: 0.7441
Epoch 141/600 - 0s 2ms/step - loss: 0.0660 - accuracy: 0.7341
Epoch 142/600 - 0s 3ms/step - loss: 0.0649 - accuracy: 0.7166
Epoch 143/600 - 0s 4ms/step - loss: 0.0646 - accuracy: 0.7450
Epoch 144/600 - 0s 2ms/step - loss: 0.0639 - accuracy: 0.7450
Epoch 145/600 - 0s 2ms/step - loss: 0.0655 - accuracy: 0.7400
Epoch 146/600 - 0s 2ms/step - loss: 0.0696 - accuracy: 0.7125
Epoch 147/600 - 0s 2ms/step - loss: 0.0667 - accuracy: 0.7450
Epoch 148/600 - 0s 3ms/step - loss: 0.0608 - accuracy: 0.7758
Epoch 149/600 - 0s 2ms/step - loss: 0.0650 - accuracy: 0.7442
Epoch 150/600 - 0s 2ms/step - loss: 0.0650 - accuracy: 0.7592
Epoch 151/600 - 0s 3ms/step - loss: 0.0661 - accuracy: 0.7592
Epoch 152/600 - 0s 3ms/step - loss: 0.0635 - accuracy: 0.7567
Epoch 153/600 - 0s 3ms/step - loss: 0.0593 - accuracy: 0.7801
Epoch 154/600 - 0s 2ms/step - loss: 0.0654 - accuracy: 0.7601
Epoch 155/600 - 0s 2ms/step - loss: 0.0636 - accuracy: 0.7742
Epoch 156/600 - 0s 4ms/step - loss: 0.0632 - accuracy: 0.7601
Epoch 157/600 - 0s 3ms/step - loss: 0.0618 - accuracy: 0.7576
Epoch 158/600 - 0s 2ms/step - loss: 0.0643 - accuracy: 0.7693
Epoch 159/600 - 0s 3ms/step - loss: 0.0666 - accuracy: 0.7493
Epoch 160/600 - 0s 2ms/step - loss: 0.0639 - accuracy: 0.7343
Epoch 161/600 - 0s 3ms/step - loss: 0.0652 - accuracy: 0.7643
Epoch 162/600 - 0s 4ms/step - loss: 0.0653 - accuracy: 0.7718
Epoch 163/600 - 0s 3ms/step - loss: 0.0558 - accuracy: 0.8018
Epoch 164/600 - 0s 2ms/step - loss: 0.0626 - accuracy: 0.7718
Epoch 165/600 - 0s 5ms/step - loss: 0.0594 - accuracy: 0.7768
Epoch 166/600 - 0s 2ms/step - loss: 0.0617 - accuracy: 0.7743
Epoch 167/600 - 0s 3ms/step - loss: 0.0644 - accuracy: 0.7651
Epoch 168/600 - 0s 4ms/step - loss: 0.0625 - accuracy: 0.7776
Epoch 169/600 - 0s 2ms/step - loss: 0.0560 - accuracy: 0.8126
Epoch 170/600 - 0s 2ms/step - loss: 0.0642 - accuracy: 0.7676
Epoch 171/600 - 0s 5ms/step - loss: 0.0572 - accuracy: 0.7901
Epoch 172/600 - 0s 3ms/step - loss: 0.0583 - accuracy: 0.7951
Epoch 173/600 - 0s 4ms/step - loss: 0.0542 - accuracy: 0.8260
Epoch 174/600 - 0s 2ms/step - loss: 0.0616 - accuracy: 0.7810
Epoch 175/600 - 0s 2ms/step - loss: 0.0571 - accuracy: 0.7960
Epoch 176/600 - 0s 2ms/step - loss: 0.0573 - accuracy: 0.8010
Epoch 177/600 - 0s 3ms/step - loss: 0.0596 - accuracy: 0.7835
Epoch 178/600 - 0s 3ms/step - loss: 0.0617 - accuracy: 0.7860
Epoch 179/600 - 0s 2ms/step - loss: 0.0611 - accuracy: 0.7860
Epoch 180/600 - 0s 4ms/step - loss: 0.0585 - accuracy: 0.7910
Epoch 181/600 - 0s 3ms/step - loss: 0.0618 - accuracy: 0.7760
Epoch 182/600 - 0s 2ms/step - loss: 0.0560 - accuracy: 0.8043
Epoch 183/600 - 0s 4ms/step - loss: 0.0562 - accuracy: 0.8143
Epoch 184/600 - 0s 2ms/step - loss: 0.0536 - accuracy: 0.7818
Epoch 185/600 - 0s 3ms/step - loss: 0.0591 - accuracy: 0.7943
Epoch 186/600 - 0s 2ms/step - loss: 0.0595 - accuracy: 0.7793
Epoch 187/600 - 0s 4ms/step - loss: 0.0548 - accuracy: 0.8068
Epoch 188/600 - 0s 2ms/step - loss: 0.0584 - accuracy: 0.7918
Epoch 189/600 - 0s 4ms/step - loss: 0.0628 - accuracy: 0.7768
Epoch 190/600 - 0s 2ms/step - loss: 0.0539 - accuracy: 0.8168
Epoch 191/600 - 0s 1ms/step - loss: 0.0502 - accuracy: 0.8293
Epoch 192/600 - 0s 2ms/step - loss: 0.0591 - accuracy: 0.7868
Epoch 193/600 - 0s 4ms/step - loss: 0.0522 - accuracy: 0.8193
Epoch 194/600 - 0s 2ms/step - loss: 0.0539 - accuracy: 0.8068
Epoch 195/600 - 0s 4ms/step - loss: 0.0578 - accuracy: 0.7818
Epoch 196/600 - 0s 2ms/step - loss: 0.0542 - accuracy: 0.8018
Epoch 197/600 - 0s 3ms/step - loss: 0.0552 - accuracy: 0.8043
Epoch 198/600 - 0s 2ms/step - loss: 0.0563 - accuracy: 0.7827
Epoch 199/600 - 0s 2ms/step - loss: 0.0548 - accuracy: 0.8060
Epoch 200/600 - 0s 2ms/step - loss: 0.0538 - accuracy: 0.8185
Epoch 201/600 - 0s 4ms/step - loss: 0.0500 - accuracy: 0.8260
Epoch 202/600 - 0s 2ms/step - loss: 0.0505 - accuracy: 0.8385
Epoch 203/600 - 0s 4ms/step - loss: 0.0614 - accuracy: 0.7835
Epoch 204/600 - 0s 3ms/step - loss: 0.0568 - accuracy: 0.7935
Epoch 205/600 - 0s 3ms/step - loss: 0.0519 - accuracy: 0.8260
Epoch 206/600 - 0s 2ms/step - loss: 0.0557 - accuracy: 0.8069
Epoch 207/600 - 0s 4ms/step - loss: 0.0537 - accuracy: 0.8094
Epoch 208/600 - 0s 2ms/step - loss: 0.0452 - accuracy: 0.8544
Epoch 209/600 - 0s 2ms/step - loss: 0.0526 - accuracy: 0.8219
Epoch 210/600 - 0s 4ms/step - loss: 0.0529 - accuracy: 0.8194
Epoch 211/600 - 0s 2ms/step - loss: 0.0566 - accuracy: 0.7869
Epoch 212/600 - 0s 2ms/step - loss: 0.0532 - accuracy: 0.8069
Epoch 213/600 - 0s 2ms/step - loss: 0.0552 - accuracy: 0.8244
Epoch 214/600 - 0s 3ms/step - loss: 0.0541 - accuracy: 0.8194
Epoch 215/600 - 0s 3ms/step - loss: 0.0510 - accuracy: 0.8294
Epoch 216/600 - 0s 2ms/step - loss: 0.0567 - accuracy: 0.8044
Epoch 217/600 - 0s 3ms/step - loss: 0.0537 - accuracy: 0.8144
Epoch 218/600 - 0s 3ms/step - loss: 0.0523 - accuracy: 0.8303
Epoch 219/600 - 0s 4ms/step - loss: 0.0533 - accuracy: 0.8153
Epoch 220/600 - 0s 4ms/step - loss: 0.0550 - accuracy: 0.8128
Epoch 221/600 - 0s 3ms/step - loss: 0.0564 - accuracy: 0.8003
Epoch 222/600 - 0s 3ms/step - loss: 0.0545 - accuracy: 0.8128
Epoch 223/600 - 0s 3ms/step - loss: 0.0545 - accuracy: 0.8128
Epoch 224/600 - 0s 5ms/step - loss: 0.0474 - accuracy: 0.8353
Epoch 225/600 - 0s 3ms/step - loss: 0.0488 - accuracy: 0.8353
Epoch 226/600 - 0s 3ms/step - loss: 0.0485 - accuracy: 0.8328
Epoch 227/600 - 0s 2ms/step - loss: 0.0526 - accuracy: 0.8273
Epoch 228/600 - 0s 2ms/step - loss: 0.0506 - accuracy: 0.8253
Epoch 229/600 - 0s 2ms/step - loss: 0.0523 - accuracy: 0.8303
Epoch 230/600 - 0s 3ms/step - loss: 0.0510 - accuracy: 0.8278
Epoch 231/600 - 0s 2ms/step - loss: 0.0489 - accuracy: 0.8303
Epoch 232/600 - 0s 3ms/step - loss: 0.0486 - accuracy: 0.8328
Epoch 233/600 - 0s 2ms/step - loss: 0.0508 - accuracy: 0.8228
Epoch 234/600 - 0s 3ms/step - loss: 0.0492 - accuracy: 0.8169
Epoch 235/600 - 0s 3ms/step - loss: 0.0473 - accuracy: 0.8344
Epoch 236/600 - 0s 2ms/step - loss: 0.0493 - accuracy: 0.8219
Epoch 237/600 - 0s 3ms/step - loss: 0.0488 - accuracy: 0.8119
Epoch 238/600 - 0s 3ms/step - accuracy: 0.8094
Epoch 239/600 - 0s 2ms/step - loss: 0.0490 - accuracy: 0.8253
Epoch 240/600 - 0s 2ms/step - loss: 0.0454 - accuracy: 0.8453
Epoch 241/600 - 0s 4ms/step - loss: 0.0511 - accuracy: 0.8128
Epoch 242/600 - 0s 2ms/step - loss: 0.0470 - accuracy: 0.8294
Epoch 243/600 - 0s 2ms/step - loss: 0.0482 - accuracy: 0.8269
Epoch 244/600 - 0s 5ms/step - loss: 0.0517 - accuracy: 0.8069
Epoch 245/600 - 0s 3ms/step - loss: 0.0504 - accuracy: 0.8069
Epoch 246/600 - 0s 4ms/step - loss: 0.0450 - accuracy: 0.8094
Epoch 247/600 - 0s 8ms/step - loss: 0.0436 - accuracy: 0.8394
Epoch 248/600 - 0s 2ms/step - loss: 0.0475 - accuracy: 0.8094
Epoch 249/600 - 0s 5ms/step - loss: 0.0490 - accuracy: 0.8244
Epoch 250/600 - 0s 3ms/step - loss: 0.0435 - accuracy: 0.8394
Epoch 251/600 - 0s 3ms/step - loss: 0.0548 - accuracy: 0.7735
Epoch 252/600 - 0s 3ms/step - loss: 0.0482 - accuracy: 0.8035
Epoch 253/600 - 0s 3ms/step - loss: 0.0455 - accuracy: 0.8160
Epoch 254/600 - 0s 5ms/step - loss: 0.0474 - accuracy: 0.8244
Epoch 255/600 - 0s 4ms/step - loss: 0.0502 - accuracy: 0.8019
Epoch 256/600 - 0s 3ms/step - loss: 0.0447 - accuracy: 0.8369
Epoch 257/600 - 0s 2ms/step - loss: 0.0483 - accuracy: 0.8069
Epoch 258/600 - 0s 4ms/step - loss: 0.0434 - accuracy: 0.8314
Epoch 259/600 - 0s 3ms/step - loss: 0.0483 - accuracy: 0.8110
Epoch 260/600 - 0s 2ms/step - loss: 0.0489 - accuracy: 0.8194
Epoch 261/600 - 0s 3ms/step - loss: 0.0452 - accuracy: 0.8319
Epoch 262/600 - 0s 2ms/step - loss: 0.0471 - accuracy: 0.8219
Epoch 263/600 - 0s 5ms/step - loss: 0.0471 - accuracy: 0.8019
Epoch 264/600 - 0s 3ms/step - loss: 0.0478 - accuracy: 0.8019
Epoch 265/600 - 0s 4ms/step - loss: 0.0482 - accuracy: 0.7985
Epoch 266/600 - 0s 2ms/step - loss: 0.0426 - accuracy: 0.8469
Epoch 267/600 - 0s 3ms/step - loss: 0.0477 - accuracy: 0.8019
Epoch 268/600 - 0s 2ms/step - loss: 0.0427 - accuracy: 0.8269
Epoch 269/600 - 0s 3ms/step - loss: 0.0472 - accuracy: 0.8119
Epoch 270/600 - 0s 5ms/step - loss: 0.0465 - accuracy: 0.8169
Epoch 271/600 - 0s 3ms/step - loss: 0.0459 - accuracy: 0.8219
Epoch 272/600 - 0s 3ms/step - loss: 0.0449 - accuracy: 0.8203
Epoch 273/600 - 0s 2ms/step - loss: 0.0471 - accuracy: 0.8278
Epoch 274/600 - 0s 4ms/step - loss: 0.0378 - accuracy: 0.8703
Epoch 275/600 - 0s 2ms/step - loss: 0.0440 - accuracy: 0.8278
Epoch 276/600 - 0s 4ms/step - loss: 0.0466 - accuracy: 0.8178
Epoch 277/600 - 0s 2ms/step - loss: 0.0498 - accuracy: 0.8019
Epoch 278/600 - 0s 4ms/step - loss: 0.0467 - accuracy: 0.7985
Epoch 279/600 - 0s 4ms/step - loss: 0.0402 - accuracy: 0.8348
Epoch 280/600 - 0s 2ms/step - loss: 0.0442 - accuracy: 0.8328
Epoch 281/600 - 0s 3ms/step - loss: 0.0500 - accuracy: 0.8028
Epoch 282/600 - 0s 3ms/step - loss: 0.0442 - accuracy: 0.8353
Epoch 283/600 - 0s 2ms/step - loss: 0.0488 - accuracy: 0.8203
Epoch 284/600 - 0s 5ms/step - loss: 0.0440 - accuracy: 0.8203
Epoch 285/600 - 0s 2ms/step - loss: 0.0452 - accuracy: 0.8261
Epoch 286/600 - 0s 3ms/step - loss: 0.0418 - accuracy: 0.8587
Epoch 287/600 - 0s 3ms/step - loss: 0.0370 - accuracy: 0.8537
Epoch 288/600 - 0s 3ms/step - loss: 0.0361 - accuracy: 0.8634
Epoch 289/600 - 0s 3ms/step - loss: 0.0415 - accuracy: 0.8528
Epoch 290/600 - 0s 3ms/step - loss: 0.0362 - accuracy: 0.8653
Epoch 291/600 - 0s 2ms/step - loss: 0.0401 - accuracy: 0.8603
Epoch 292/600 - 0s 2ms/step - loss: 0.0372 - accuracy: 0.8678
Epoch 293/600 - 0s 4ms/step - loss: 0.0381 - accuracy: 0.8528
Epoch 294/600 - 0s 4ms/step - loss: 0.0397 - accuracy: 0.8453
Epoch 295/600 - 0s 4ms/step - loss: 0.0367 - accuracy: 0.8603
Epoch 296/600 - 0s 5ms/step - loss: 0.0417 - accuracy: 0.8353
Epoch 297/600 - 0s 4ms/step - loss: 0.0417 - accuracy: 0.8353
Epoch 298/600 - 0s 2ms/step - loss: 0.0365 - accuracy: 0.8728
Epoch 299/600 - 0s 3ms/step - loss: 0.0370 - accuracy: 0.8728
Epoch 300/600 - 0s 2ms/step - loss: 0.0389 - accuracy: 0.8478
Epoch 301/600 - 0s 3ms/step - loss: 0.0381 - accuracy: 0.8478
Epoch 302/600 - 0s 2ms/step - loss: 0.0397 - accuracy: 0.8478
Epoch 303/600 - 0s 3ms/step - loss: 0.0382 - accuracy: 0.8428
Epoch 304/600 - 0s 3ms/step - loss: 0.0367 - accuracy: 0.8553
Epoch 305/600 - 0s 3ms/step - loss: 0.0369 - accuracy: 0.8553
Epoch 306/600 - 0s 3ms/step - loss: 0.0415 - accuracy: 0.8328
Epoch 307/600 - 0s 4ms/step - loss: 0.0347 - accuracy: 0.8678
Epoch 308/600 - 0s 3ms/step - loss: 0.0386 - accuracy: 0.8353
Epoch 309/600 - 0s 3ms/step - loss: 0.0392 - accuracy: 0.8378
Epoch 310/600 - 0s 4ms/step - loss: 0.0436 - accuracy: 0.8153
Epoch 311/600 - 0s 2ms/step - loss: 0.0352 - accuracy: 0.8528
Epoch 312/600 - 0s 3ms/step - loss: 0.0383 - accuracy: 0.8403
Epoch 313/600 - 0s 3ms/step - loss: 0.0404 - accuracy: 0.8278
Epoch 314/600 - 0s 3ms/step - loss: 0.0354 - accuracy: 0.8653
Epoch 315/600 - 0s 2ms/step - loss: 0.0408 - accuracy: 0.8278
Epoch 316/600 - 0s 3ms/step - loss: 0.0404 - accuracy: 0.8286
Epoch 317/600 - 0s 3ms/step - loss: 0.0411 - accuracy: 0.8361
```



```
3/3 [=====] - ETA: 0s - loss: 0.0508 - accuracy: 0.78 - 0s 6ms/step - loss: 0.0300 - accuracy: 0.8395
Epoch 416/600
3/3 [=====] - 0s 5ms/step - loss: 0.0317 - accuracy: 0.8870
Epoch 417/600
3/3 [=====] - 0s 6ms/step - loss: 0.0359 - accuracy: 0.8670
Epoch 418/600
3/3 [=====] - 0s 5ms/step - loss: 0.0380 - accuracy: 0.8287
Epoch 419/600
3/3 [=====] - 0s 5ms/step - loss: 0.0364 - accuracy: 0.8595
Epoch 420/600
3/3 [=====] - 0s 3ms/step - loss: 0.0362 - accuracy: 0.8570
Epoch 421/600
3/3 [=====] - 0s 3ms/step - loss: 0.0342 - accuracy: 0.8620
Epoch 422/600
3/3 [=====] - 0s 3ms/step - loss: 0.0343 - accuracy: 0.8645
Epoch 423/600
3/3 [=====] - 0s 5ms/step - loss: 0.0329 - accuracy: 0.8545
Epoch 424/600
3/3 [=====] - 0s 2ms/step - loss: 0.0316 - accuracy: 0.8745
Epoch 425/600
3/3 [=====] - 0s 6ms/step - loss: 0.0328 - accuracy: 0.8595
Epoch 426/600
3/3 [=====] - 0s 6ms/step - loss: 0.0336 - accuracy: 0.8475
Epoch 427/600
3/3 [=====] - 0s 3ms/step - loss: 0.0354 - accuracy: 0.8779
Epoch 428/600
3/3 [=====] - 0s 2ms/step - loss: 0.0359 - accuracy: 0.8604
Epoch 429/600
3/3 [=====] - 0s 2ms/step - loss: 0.0328 - accuracy: 0.8704
Epoch 430/600
3/3 [=====] - 0s 2ms/step - loss: 0.0342 - accuracy: 0.8579
Epoch 431/600
3/3 [=====] - 0s 2ms/step - loss: 0.0329 - accuracy: 0.8604
Epoch 432/600
3/3 [=====] - 0s 5ms/step - loss: 0.0324 - accuracy: 0.8754
Epoch 433/600
3/3 [=====] - 0s 5ms/step - loss: 0.0349 - accuracy: 0.8520
Epoch 434/600
3/3 [=====] - 0s 3ms/step - loss: 0.0326 - accuracy: 0.8704
Epoch 435/600
3/3 [=====] - 0s 2ms/step - loss: 0.0326 - accuracy: 0.8754
Epoch 436/600
3/3 [=====] - 0s 2ms/step - loss: 0.0346 - accuracy: 0.8579
Epoch 437/600
3/3 [=====] - 0s 3ms/step - loss: 0.0321 - accuracy: 0.8754
Epoch 438/600
3/3 [=====] - 0s 3ms/step - loss: 0.0303 - accuracy: 0.8795
Epoch 439/600
3/3 [=====] - 0s 5ms/step - loss: 0.0334 - accuracy: 0.8779
Epoch 440/600
3/3 [=====] - 0s 4ms/step - loss: 0.0352 - accuracy: 0.8637
Epoch 441/600
3/3 [=====] - 0s 2ms/step - loss: 0.0328 - accuracy: 0.8612
Epoch 442/600
3/3 [=====] - 0s 2ms/step - loss: 0.0331 - accuracy: 0.8612
Epoch 443/600
3/3 [=====] - 0s 3ms/step - loss: 0.0320 - accuracy: 0.8812
Epoch 444/600
3/3 [=====] - 0s 6ms/step - loss: 0.0314 - accuracy: 0.8762
Epoch 445/600
3/3 [=====] - 0s 4ms/step - loss: 0.0356 - accuracy: 0.8587
Epoch 446/600
3/3 [=====] - 0s 2ms/step - loss: 0.0313 - accuracy: 0.8837
Epoch 447/600
3/3 [=====] - 0s 1ms/step - loss: 0.0316 - accuracy: 0.8762
Epoch 448/600
3/3 [=====] - 0s 2ms/step - loss: 0.0326 - accuracy: 0.9037
Epoch 449/600
3/3 [=====] - 0s 2ms/step - loss: 0.0316 - accuracy: 0.8737
Epoch 450/600
3/3 [=====] - 0s 2ms/step - loss: 0.0326 - accuracy: 0.8737
Epoch 451/600
3/3 [=====] - 0s 5ms/step - loss: 0.0276 - accuracy: 0.8962
Epoch 452/600
3/3 [=====] - 0s 20ms/step - loss: 0.0325 - accuracy: 0.8737
Epoch 453/600
3/3 [=====] - 0s 4ms/step - loss: 0.0308 - accuracy: 0.8862
Epoch 454/600
3/3 [=====] - 0s 3ms/step - loss: 0.0282 - accuracy: 0.8787
Epoch 455/600
3/3 [=====] - 0s 3ms/step - loss: 0.0382 - accuracy: 0.8337
Epoch 456/600
3/3 [=====] - 0s 3ms/step - loss: 0.0326 - accuracy: 0.8787
Epoch 457/600
3/3 [=====] - 0s 2ms/step - loss: 0.0325 - accuracy: 0.8662
Epoch 458/600
3/3 [=====] - 0s 3ms/step - loss: 0.0300 - accuracy: 0.8587
Epoch 459/600
3/3 [=====] - 0s 2ms/step - loss: 0.0327 - accuracy: 0.8737
Epoch 460/600
3/3 [=====] - 0s 2ms/step - loss: 0.0328 - accuracy: 0.8587
Epoch 461/600
3/3 [=====] - 0s 2ms/step - loss: 0.0319 - accuracy: 0.8612
Epoch 462/600
3/3 [=====] - 0s 2ms/step - loss: 0.0292 - accuracy: 0.8937
Epoch 463/600
3/3 [=====] - 0s 2ms/step - loss: 0.0328 - accuracy: 0.8712
Epoch 464/600
3/3 [=====] - 0s 1ms/step - loss: 0.0269 - accuracy: 0.8962
Epoch 465/600
3/3 [=====] - 0s 2ms/step - loss: 0.0285 - accuracy: 0.8887
Epoch 466/600
3/3 [=====] - 0s 2ms/step - loss: 0.0369 - accuracy: 0.8487
Epoch 467/600
3/3 [=====] - 0s 1ms/step - loss: 0.0288 - accuracy: 0.8887
Epoch 468/600
3/3 [=====] - 0s 2ms/step - loss: 0.0293 - accuracy: 0.8737
Epoch 469/600
3/3 [=====] - 0s 2ms/step - loss: 0.0277 - accuracy: 0.8937
Epoch 470/600
3/3 [=====] - 0s 2ms/step - loss: 0.0324 - accuracy: 0.8637
Epoch 471/600
3/3 [=====] - 0s 2ms/step - loss: 0.0331 - accuracy: 0.8637
Epoch 472/600
3/3 [=====] - 0s 2ms/step - loss: 0.0288 - accuracy: 0.8962
Epoch 473/600
3/3 [=====] - 0s 3ms/step - loss: 0.0308 - accuracy: 0.8737
Epoch 474/600
3/3 [=====] - 0s 2ms/step - loss: 0.0328 - accuracy: 0.8787
Epoch 475/600
3/3 [=====] - 0s 2ms/step - loss: 0.0334 - accuracy: 0.8637
Epoch 476/600
3/3 [=====] - 0s 2ms/step - loss: 0.0294 - accuracy: 0.8762
Epoch 477/600
3/3 [=====] - 0s 2ms/step - loss: 0.0296 - accuracy: 0.8787
Epoch 478/600
3/3 [=====] - 0s 2ms/step - loss: 0.0291 - accuracy: 0.8871
Epoch 479/600
3/3 [=====] - 0s 2ms/step - loss: 0.0287 - accuracy: 0.8812
Epoch 480/600
3/3 [=====] - 0s 2ms/step - loss: 0.0310 - accuracy: 0.8737
Epoch 481/600
3/3 [=====] - 0s 1ms/step - loss: 0.0302 - accuracy: 0.8687
Epoch 482/600
3/3 [=====] - 0s 2ms/step - loss: 0.0296 - accuracy: 0.8812
Epoch 483/600
3/3 [=====] - 0s 2ms/step - loss: 0.0282 - accuracy: 0.8787
Epoch 484/600
3/3 [=====] - 0s 1ms/step - loss: 0.0281 - accuracy: 0.8612
Epoch 485/600
3/3 [=====] - 0s 2ms/step - loss: 0.0275 - accuracy: 0.8871
Epoch 486/600
3/3 [=====] - 0s 2ms/step - loss: 0.0281 - accuracy: 0.8821
Epoch 487/600
3/3 [=====] - 0s 2ms/step - loss: 0.0281 - accuracy: 0.8921
Epoch 488/600
3/3 [=====] - 0s 2ms/step - loss: 0.0264 - accuracy: 0.9021
Epoch 489/600
3/3 [=====] - 0s 1ms/step - loss: 0.0280 - accuracy: 0.8846
Epoch 490/600
3/3 [=====] - 0s 2ms/step - loss: 0.0307 - accuracy: 0.8737
Epoch 491/600
3/3 [=====] - 0s 1ms/step - loss: 0.0303 - accuracy: 0.8696
Epoch 492/600
3/3 [=====] - 0s 2ms/step - loss: 0.0310 - accuracy: 0.8646
Epoch 493/600
3/3 [=====] - 0s 1ms/step - loss: 0.0325 - accuracy: 0.8646
Epoch 494/600
3/3 [=====] - 0s 2ms/step - loss: 0.0308 - accuracy: 0.8712
Epoch 495/600
3/3 [=====] - 0s 1ms/step - loss: 0.0293 - accuracy: 0.8821
Epoch 496/600
3/3 [=====] - 0s 1ms/step - loss: 0.0270 - accuracy: 0.8846
Epoch 497/600
3/3 [=====] - 0s 2ms/step - loss: 0.0230 - accuracy: 0.8671
Epoch 498/600
3/3 [=====] - 0s 2ms/step - loss: 0.0283 - accuracy: 0.8821
Epoch 499/600
3/3 [=====] - 0s 2ms/step - loss: 0.0276 - accuracy: 0.8921
Epoch 500/600
3/3 [=====] - 0s 2ms/step - loss: 0.0307 - accuracy: 0.8721
Epoch 501/600
3/3 [=====] - 0s 3ms/step - loss: 0.0286 - accuracy: 0.8896
Epoch 502/600
3/3 [=====] - 0s 1ms/step - loss: 0.0346 - accuracy: 0.8521
Epoch 503/600
3/3 [=====] - 0s 2ms/step - loss: 0.0296 - accuracy: 0.8821
Epoch 504/600
3/3 [=====] - 0s 3ms/step - loss: 0.0300 - accuracy: 0.8746
Epoch 505/600
3/3 [=====] - 0s 2ms/step - loss: 0.0322 - accuracy: 0.8721
Epoch 506/600
3/3 [=====] - 0s 2ms/step - loss: 0.0322 - accuracy: 0.8621
Epoch 507/600
3/3 [=====] - 0s 2ms/step - loss: 0.0304 - accuracy: 0.8796
Epoch 508/600
3/3 [=====] - 0s 2ms/step - loss: 0.0220 - accuracy: 0.8671
Epoch 509/600
3/3 [=====] - 0s 3ms/step - loss: 0.0302 - accuracy: 0.8771
Epoch 510/600
3/3 [=====] - 0s 2ms/step - loss: 0.0302 - accuracy: 0.8696
Epoch 511/600
3/3 [=====] - 0s 4ms/step - loss: 0.0268 - accuracy: 0.8971
Epoch 512/600
3/3 [=====] - 0s 2ms/step - loss: 0.0287 - accuracy: 0.8671
Epoch 513/600
3/3 [=====] - 0s 2ms/step - loss: 0.0268 - accuracy: 0.8671
Epoch 514/600
3/3 [=====] - 0s 2ms/step - loss: 0.0304 - accuracy: 0.8646
Epoch 515/600
3/3 [=====] - 0s 2ms/step - loss: 0.0297 - accuracy: 0.8771
Epoch 516/600
3/3 [=====] - 0s 2ms/step - loss: 0.0295 - accuracy: 0.8796
Epoch 517/600
3/3 [=====] - 0s 2ms/step - loss: 0.0220 - accuracy: 0.8521
Epoch 518/600
3/3 [=====] - 0s 1ms/step - loss: 0.0299 - accuracy: 0.8746
Epoch 519/600
3/3 [=====] - 0s 3ms/step - loss: 0.0251 - accuracy: 0.8946
Epoch 520/600
3/3 [=====] - 0s 2ms/step - loss: 0.0289 - accuracy: 0.8746
Epoch 521/600
3/3 [=====] - 0s 2ms/step - loss: 0.0245 - accuracy: 0.8996
Epoch 522/600
3/3 [=====] - 0s 2ms/step - loss: 0.0281 - accuracy: 0.8771
Epoch 523/600
3/3 [=====] - 0s 2ms/step - loss: 0.0274 - accuracy: 0.8771
Epoch 524/600
3/3 [=====] - 0s 2ms/step - loss: 0.0288 - accuracy: 0.8921
Epoch 525/600
3/3 [=====] - 0s 2ms/step - loss: 0.0308 - accuracy: 0.8696
Epoch 526/600
3/3 [=====] - 0s 2ms/step - loss: 0.0266 - accuracy: 0.8896
Epoch 527/600
3/3 [=====] - 0s 2ms/step - loss: 0.0272 - accuracy: 0.8871
Epoch 528/600
3/3 [=====] - 0s 2ms/step - loss: 0.0258 - accuracy: 0.8946
Epoch 529/600
3/3 [=====] - 0s 2ms/step - loss: 0.0258 - accuracy: 0.8846
Epoch 530/600
3/3 [=====] - 0s 2ms/step - loss: 0.0272 - accuracy: 0.8946
Epoch 531/600
3/3 [=====] - 0s 2ms/step - loss: 0.0277 - accuracy: 0.8821
Epoch 532/600
3/3 [=====] - 0s 2ms/step - loss: 0.0277 - accuracy: 0.8821
Epoch 533/600
3/3 [=====] - 0s 1ms/step - loss: 0.0266 - accuracy: 0.8896
Epoch 534/600
3/3 [=====] - 0s 2ms/step - loss: 0.0272 - accuracy: 0.8871
Epoch 535/600
3/3 [=====] - 0s 1ms/step - loss: 0.0266 - accuracy: 0.8937
Epoch 536/600
3/3 [=====] - 0s 2ms/step - loss: 0.0272 - accuracy: 0.8946
Epoch 537/600
3/3 [=====] - 0s 2ms/step - loss: 0.0264 - accuracy: 0.8887
Epoch 538/600
3/3 [=====] - 0s 1ms/step - loss: 0.0233 - accuracy: 0.9138
Epoch 539/600
3/3 [=====] - 0s 1ms/step - loss: 0.0289 - accuracy: 0.8846
Epoch 540/600
3/3 [=====] - 0s 2ms/step - loss: 0.0276 - accuracy: 0.8855
Epoch 541/600
3/3 [=====] - 0s 2ms/step - loss: 0.0281 - accuracy: 0.8721
Epoch 542/600
3/3 [=====] - 0s 1ms/step - loss: 0.0268 - accuracy: 0.8913
Epoch 543/600
3/3 [=====] - 0s 2ms/step - loss: 0.0263 - accuracy: 0.9013
Epoch 544/600
3/3 [=====] - 0s 2ms/step - loss: 0.0245 - accuracy: 0.9047
Epoch 545/600
3/3 [=====] - 0s 1ms/step - loss: 0.0274 - accuracy: 0.8913
Epoch 546/600
3/3 [=====] - 0s 3ms/step - loss: 0.0245 - accuracy: 0.8871
Epoch 547/600
3/3 [=====] - 0s 2ms/step - loss: 0.0292 - accuracy: 0.8938
Epoch 548/600
3/3 [=====] - 0s 3ms/step - loss: 0.0261 - accuracy: 0.8838
Epoch 549/600
3/3 [=====] - 0s 2ms/step - loss: 0.0225 - accuracy: 0.9138
Epoch 550/600
3/3 [=====] - 0s 3ms/step - loss: 0.0249 - accuracy: 0.9072
Epoch 551/600
3/3 [=====] - 0s 2ms/step - loss: 0.0277 - accuracy: 0.9080
Epoch 552/600
3/3 [=====] - 0s 2ms/step - loss: 0.0273 - accuracy: 0.8905
Epoch 553/600
3/3 [=====] - 0s 2ms/step - loss: 0.0266 - accuracy: 0.9080
Epoch 554/600
3/3 [=====] - 0s 2ms/step - loss: 0.0298 - accuracy: 0.8722
Epoch 555/600
3/3 [=====] - 0s 2ms/step - loss: 0.0289 - accuracy: 0.8880
Epoch 556/600
3/3 [=====] - 0s 2ms/step - loss: 0.0267 - accuracy: 0.9080
Epoch 557/600
3/3 [=====] - 0s 2ms/step - loss: 0.0269 - accuracy: 0.9105
Epoch 558/600
3/3 [=====] - 0s 2ms/step - loss: 0.0273 - accuracy: 0.8905
Epoch 559/600
3/3 [=====] - 0s 2ms/step - loss: 0.0249 - accuracy: 0.9180
Epoch 560/600
3/3 [=====] - 0s 2ms/step - loss: 0.0248 - accuracy: 0.9147
Epoch 561/600
3/3 [=====] - 0s 2ms/step - loss: 0.0242 - accuracy: 0.9022
Epoch 562/600
3/3 [=====] - 0s 2ms/step - loss: 0.0266 - accuracy: 0.9130
Epoch 563/600
3/3 [=====] - 0s 2ms/step - loss: 0.0286 - accuracy: 0.8880
Epoch 564/600
3/3 [=====] - 0s 2ms/step - loss: 0.0257 - accuracy: 0.9005
Epoch 565/600
3/3 [=====] - 0s 2ms/step - loss: 0.0289 - accuracy: 0.8905
Epoch 566/600
3/3 [=====] - 0s 2ms/step - loss: 0.0251 - accuracy: 0.9105
Epoch 567/600
3/3 [=====] - 0s 2ms/step - loss: 0.0268 - accuracy: 0.8980
Epoch 568/600
3/3 [=====] - 0s 2ms/step - loss: 0.0249 - accuracy: 0.9080
Epoch 569/600
3/3 [=====] - 0s 1ms/step - loss: 0.0275 - accuracy: 0.8905
Epoch 570/600
3/3 [=====] - 0s 2ms/step - loss: 0.0264 - accuracy: 0.8980
Epoch 571/600
3/3 [=====] - 0s 2ms/step - loss: 0.0252 - accuracy: 0.9055
Epoch 572/600
3/3 [=====] - 0s 2ms/step - loss: 0.0209 - accuracy: 0.9280
Epoch 573/600
3/3 [=====] - 0s 2ms/step - loss: 0.0232 - accuracy: 0.9080
Epoch 574/600
3/3 [=====] - 0s 3ms/step - loss: 0.0244 - accuracy: 0.9214
Epoch 575/600
3/3 [=====] - 0s 3ms/step - loss: 0.0253 - accuracy: 0.9005
Epoch 576/600
3/3 [=====] - 0s 3ms/step - loss: 0.0242 - accuracy: 0.9005
Epoch 577/600
3/3 [=====] - 0s 2ms/step - loss: 0.0212 - accuracy: 0.9339
Epoch 578/600
3/3 [=====] - 0s 3ms/step - loss: 0.0274 - accuracy: 0.8897
Epoch 579/600
3/3 [=====] - 0s 2ms/step - loss: 0.0257 - accuracy: 0.9139
Epoch 580/600
3/3 [=====] - 0s 3ms/step - loss: 0.0264 - accuracy: 0.9114
Epoch 581/600
3/3 [=====] - 0s 2ms/step - loss: 0.0395 - accuracy: 0.7692
```

```
Out[28]: [0.059900734573602676, 0.7692307829856873]
```

```
In [29]: _, train_acc = trial_model_3.evaluate(X_train, y_train, verbose=0)
_, test_acc = trial_model_3.evaluate(X_test, y_test, verbose=0)
```

```
print('Train: %.3f, Test: %.3f' % (train_acc, test_acc))
```

```
Train: 0.913, Test: 0.769
```

```
In [30]: print(History1.history.keys())
```

```
plt.plot(History1.history['accuracy'])
```

```
plt.title('Models Accuracy')
```

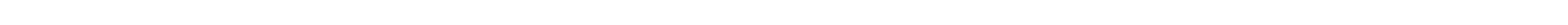
```
plt.ylabel('Accuracy')
```

```
plt.xlabel('epoch')
```

```
plt.legend(['Accuracy'], loc='upper right')
```

```
plt.show()
```

```
dict_keys(['loss', 'accuracy'])
```



**Train and test accuracy scores**

1. Train:0.859, Test:0.815

2. Train:0.879, Test:0.785

3. Train:0.913, Test:0.815

**Changing random\_state = 2**

1. Train:0.872, Test:0.769

**Changing the split size = 0.2**

1. Train:0.889, Test:0.698

2. Train:0.889, Test:0.744

**Changing the split size = 0.3| batch size = 50| epochs = 600**

1. Train:0.889, Test:0.744

2. Train:0.906, Test:0.754

**Notes**

Using MSE as loss function may be limited in how much a testing accuracy can be improved and I can see the accuracy in training the model climbs quickly in training the dataset. I may need to use categorical\_crossentropy for categorical features in target variable being trained.

- The learning rate was set as default 0.01 using adam optimizer

- Due to the randomness in learning between the independent variable values the testing set suffers from a limited predictability using Mean Squared Error as loss function.

**Model 4**











|                 |   |                                                               |                                   |
|-----------------|---|---------------------------------------------------------------|-----------------------------------|
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6687 - accuracy: 0.7025 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.7136 - accuracy: 0.7016 |
| Bpoch 825/2200  | - | 0s 3ms/step                                                   | - loss: 0.6928 - accuracy: 0.7090 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.7222 - accuracy: 0.6942 |
| Bpoch 827/2200  | - | 0s 4ms/step                                                   | - loss: 0.6893 - accuracy: 0.7164 |
| 2/2 [=====]     | - | 0s 2ms/step                                                   | - loss: 0.6962 - accuracy: 0.7090 |
| Bpoch 828/2200  | - | 0s 3ms/step                                                   | - loss: 0.6875 - accuracy: 0.7090 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6772 - accuracy: 0.7127 |
| Bpoch 830/2200  | - | 0s 4ms/step                                                   | - loss: 0.6794 - accuracy: 0.7164 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6900 - accuracy: 0.7164 |
| Bpoch 831/2200  | - | 0s 3ms/step                                                   | - loss: 0.6872 - accuracy: 0.7164 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6973 - accuracy: 0.7201 |
| Bpoch 832/2200  | - | 0s 3ms/step                                                   | - loss: 0.6900 - accuracy: 0.7164 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6872 - accuracy: 0.7164 |
| Bpoch 834/2200  | - | 0s 3ms/step                                                   | - loss: 0.6894 - accuracy: 0.7312 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6986 - accuracy: 0.7164 |
| Bpoch 835/2200  | - | 0s 4ms/step                                                   | - loss: 0.6773 - accuracy: 0.7201 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6934 - accuracy: 0.6979 |
| Bpoch 837/2200  | - | 0s 5ms/step                                                   | - loss: 0.7094 - accuracy: 0.6979 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6588 - accuracy: 0.7164 |
| Bpoch 840/2200  | - | 0s 4ms/step                                                   | - loss: 0.6820 - accuracy: 0.7127 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6975 - accuracy: 0.7090 |
| Bpoch 842/2200  | - | 0s 4ms/step                                                   | - loss: 0.6925 - accuracy: 0.7164 |
| 2/2 [=====]     | - | 0s 2ms/step                                                   | - loss: 0.6900 - accuracy: 0.7127 |
| Bpoch 844/2200  | - | 0s 3ms/step                                                   | - loss: 0.6873 - accuracy: 0.7090 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6956 - accuracy: 0.7164 |
| Bpoch 845/2200  | - | 0s 3ms/step                                                   | - loss: 0.7015 - accuracy: 0.6979 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6860 - accuracy: 0.7275 |
| Bpoch 847/2200  | - | 0s 3ms/step                                                   | - loss: 0.6932 - accuracy: 0.7238 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6911 - accuracy: 0.7090 |
| Bpoch 848/2200  | - | 0s 3ms/step                                                   | - loss: 0.6939 - accuracy: 0.7164 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6906 - accuracy: 0.7164 |
| Bpoch 850/2200  | - | 0s 3ms/step                                                   | - loss: 0.6850 - accuracy: 0.7090 |
| 2/2 [=====]     | - | 0s 6ms/step                                                   | - loss: 0.6889 - accuracy: 0.7053 |
| Bpoch 852/2200  | - | 0s 3ms/step                                                   | - loss: 0.6733 - accuracy: 0.7127 |
| 2/2 [=====]     | - | 0s 2ms/step                                                   | - loss: 0.6697 - accuracy: 0.7275 |
| Bpoch 854/2200  | - | 0s 2ms/step                                                   | - loss: 0.6834 - accuracy: 0.7016 |
| 2/2 [=====]     | - | 0s 5ms/step                                                   | - loss: 0.6432 - accuracy: 0.7386 |
| Bpoch 857/2200  | - | 0s 4ms/step                                                   | - loss: 0.6912 - accuracy: 0.7127 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6933 - accuracy: 0.7090 |
| Bpoch 859/2200  | - | 0s 7ms/step                                                   | - loss: 0.6784 - accuracy: 0.7164 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6615 - accuracy: 0.7238 |
| Bpoch 861/2200  | - | 0s 3ms/step                                                   | - loss: 0.6704 - accuracy: 0.7127 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6822 - accuracy: 0.7090 |
| Bpoch 862/2200  | - | 0s 3ms/step                                                   | - loss: 0.6572 - accuracy: 0.7238 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6822 - accuracy: 0.7090 |
| Bpoch 864/2200  | - | 0s 4ms/step                                                   | - loss: 0.6795 - accuracy: 0.7127 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6836 - accuracy: 0.7090 |
| Bpoch 865/2200  | - | 0s 3ms/step                                                   | - loss: 0.6723 - accuracy: 0.7275 |
| 2/2 [=====]     | - | 0s 5ms/step                                                   | - loss: 0.7062 - accuracy: 0.6979 |
| Bpoch 867/2200  | - | 0s 3ms/step                                                   | - loss: 0.6532 - accuracy: 0.7201 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6880 - accuracy: 0.7090 |
| Bpoch 869/2200  | - | 0s 6ms/step                                                   | - loss: 0.6859 - accuracy: 0.7090 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6693 - accuracy: 0.7164 |
| Bpoch 872/2200  | - | 0s 3ms/step                                                   | - loss: 0.6432 - accuracy: 0.7090 |
| 2/2 [=====]     | - | 0s 2ms/step                                                   | - loss: 0.6718 - accuracy: 0.7127 |
| Bpoch 874/2200  | - | 0s 3ms/step                                                   | - loss: 0.6955 - accuracy: 0.7053 |
| 2/2 [=====]     | - | 0s 2ms/step                                                   | - loss: 0.6771 - accuracy: 0.7238 |
| Bpoch 875/2200  | - | 0s 5ms/step                                                   | - loss: 0.6682 - accuracy: 0.7164 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6711 - accuracy: 0.7127 |
| Bpoch 877/2200  | - | 0s 3ms/step                                                   | - loss: 0.6665 - accuracy: 0.7238 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6458 - accuracy: 0.7386 |
| Bpoch 879/2200  | - | 0s 7ms/step                                                   | - loss: 0.6985 - accuracy: 0.6904 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6669 - accuracy: 0.7127 |
| Bpoch 881/2200  | - | 0s 5ms/step                                                   | - loss: 0.6759 - accuracy: 0.7164 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6653 - accuracy: 0.7127 |
| Bpoch 884/2200  | - | 0s 3ms/step                                                   | - loss: 0.6479 - accuracy: 0.7388 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6750 - accuracy: 0.7277 |
| Bpoch 886/2200  | - | 0s 2ms/step                                                   | - loss: 0.6739 - accuracy: 0.7203 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6726 - accuracy: 0.7166 |
| Bpoch 887/2200  | - | 0s 3ms/step                                                   | - loss: 0.6611 - accuracy: 0.7351 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6466 - accuracy: 0.7314 |
| Bpoch 889/2200  | - | 0s 3ms/step                                                   | - loss: 0.6865 - accuracy: 0.7055 |
| 2/2 [=====]     | - | 0s 5ms/step                                                   | - loss: 0.6633 - accuracy: 0.7240 |
| Bpoch 891/2200  | - | 0s 5ms/step                                                   | - loss: 0.6783 - accuracy: 0.7240 |
| 2/2 [=====]     | - | 0s 5ms/step                                                   | - loss: 0.6539 - accuracy: 0.7166 |
| Bpoch 893/2200  | - | 0s 6ms/step                                                   | - loss: 0.6763 - accuracy: 0.7129 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6886 - accuracy: 0.7092 |
| Bpoch 894/2200  | - | 0s 3ms/step                                                   | - loss: 0.6931 - accuracy: 0.7351 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6863 - accuracy: 0.7240 |
| Bpoch 896/2200  | - | 0s 4ms/step                                                   | - loss: 0.6563 - accuracy: 0.7240 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6674 - accuracy: 0.7203 |
| Bpoch 898/2200  | - | 0s 3ms/step                                                   | - loss: 0.6872 - accuracy: 0.7016 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6764 - accuracy: 0.7055 |
| Bpoch 899/2200  | - | 0s 4ms/step                                                   | - loss: 0.6600 - accuracy: 0.7277 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6587 - accuracy: 0.7277 |
| Bpoch 901/2200  | - | 0s 3ms/step                                                   | - loss: 0.6561 - accuracy: 0.7129 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6891 - accuracy: 0.7166 |
| Bpoch 904/2200  | - | 0s 4ms/step                                                   | - loss: 0.6596 - accuracy: 0.7203 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6626 - accuracy: 0.7240 |
| Bpoch 906/2200  | - | 0s 3ms/step                                                   | - loss: 0.6837 - accuracy: 0.7018 |
| 2/2 [=====]     | - | 0s 2ms/step                                                   | - loss: 0.6812 - accuracy: 0.7092 |
| Bpoch 908/2200  | - | 0s 4ms/step                                                   | - loss: 0.6708 - accuracy: 0.7129 |
| 2/2 [=====]     | - | 0s 2ms/step                                                   | - loss: 0.6469 - accuracy: 0.7277 |
| Bpoch 909/2200  | - | 0s 3ms/step                                                   | - loss: 0.6919 - accuracy: 0.7092 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6704 - accuracy: 0.7166 |
| Bpoch 911/2200  | - | 0s 3ms/step                                                   | - loss: 0.6630 - accuracy: 0.7166 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6584 - accuracy: 0.7203 |
| Bpoch 913/2200  | - | 0s 3ms/step                                                   | - loss: 0.6461 - accuracy: 0.7314 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6313 - accuracy: 0.7351 |
| Bpoch 916/2200  | - | 0s 4ms/step                                                   | - loss: 0.6931 - accuracy: 0.7018 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6670 - accuracy: 0.7240 |
| Bpoch 918/2200  | - | 0s 3ms/step                                                   | - loss: 0.6648 - accuracy: 0.7092 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6775 - accuracy: 0.7018 |
| Bpoch 920/2200  | - | 0s 2ms/step                                                   | - loss: 0.6564 - accuracy: 0.7277 |
| 2/2 [=====]     | - | 0s 2ms/step                                                   | - loss: 0.6699 - accuracy: 0.7129 |
| Bpoch 921/2200  | - | 0s 3ms/step                                                   | - loss: 0.6439 - accuracy: 0.7240 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6582 - accuracy: 0.7203 |
| Bpoch 924/2200  | - | 0s 4ms/step                                                   | - loss: 0.6642 - accuracy: 0.7166 |
| 2/2 [=====]     | - | 0s 2ms/step                                                   | - loss: 0.6398 - accuracy: 0.7388 |
| Bpoch 926/2200  | - | 0s 2ms/step                                                   | - loss: 0.6879 - accuracy: 0.6981 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6591 - accuracy: 0.7129 |
| Bpoch 928/2200  | - | 0s 3ms/step                                                   | - loss: 0.6931 - accuracy: 0.6943 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6693 - accuracy: 0.7203 |
| Bpoch 930/2200  | - | 0s 4ms/step                                                   | - loss: 0.6529 - accuracy: 0.7203 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6479 - accuracy: 0.7314 |
| Bpoch 933/2200  | - | 0s 3ms/step                                                   | - loss: 0.6386 - accuracy: 0.7388 |
| 2/2 [=====]     | - | 0s 2ms/step                                                   | - loss: 0.6437 - accuracy: 0.7351 |
| Bpoch 935/2200  | - | 0s 3ms/step                                                   | - loss: 0.6587 - accuracy: 0.7092 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6699 - accuracy: 0.7168 |
| Bpoch 937/2200  | - | 0s 2ms/step                                                   | - loss: 0.6501 - accuracy: 0.7316 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6308 - accuracy: 0.7427 |
| Bpoch 938/2200  | - | 0s 2ms/step                                                   | - loss: 0.6558 - accuracy: 0.7316 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6540 - accuracy: 0.7316 |
| Bpoch 941/2200  | - | 0s 3ms/step                                                   | - loss: 0.6646 - accuracy: 0.7168 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6608 - accuracy: 0.7429 |
| Bpoch 943/2200  | - | 0s 3ms/step                                                   | - loss: 0.6608 - accuracy: 0.7429 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6532 - accuracy: 0.7355 |
| Bpoch 947/2200  | - | 0s 3ms/step                                                   | - loss: 0.6751 - accuracy: 0.7207 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6694 - accuracy: 0.7281 |
| Bpoch 948/2200  | - | 0s 3ms/step                                                   | - loss: 0.6667 - accuracy: 0.7207 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6573 - accuracy: 0.7355 |
| Bpoch 950/2200  | - | 0s 3ms/step                                                   | - loss: 0.6790 - accuracy: 0.7133 |
| 2/2 [=====]     | - | 0s 5ms/step                                                   | - loss: 0.6484 - accuracy: 0.7355 |
| Bpoch 952/2200  | - | ETA: 0s - loss: 0.6553 - accuracy: 0.73 - 0s 5ms/step - loss: |                                   |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6420 - accuracy: 0.7429 |
| Bpoch 954/2200  | - | 0s 5ms/step                                                   | - loss: 0.6533 - accuracy: 0.7394 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6570 - accuracy: 0.7394 |
| Bpoch 956/2200  | - | 0s 3ms/step                                                   | - loss: 0.6678 - accuracy: 0.7283 |
| 2/2 [=====]     | - | 0s 2ms/step                                                   | - loss: 0.6660 - accuracy: 0.7357 |
| Bpoch 959/2200  | - | 0s 4ms/step                                                   | - loss: 0.6486 - accuracy: 0.7320 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6835 - accuracy: 0.7209 |
| Bpoch 961/2200  | - | 0s 3ms/step                                                   | - loss: 0.6539 - accuracy: 0.7468 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6367 - accuracy: 0.7468 |
| Bpoch 963/2200  | - | 0s 4ms/step                                                   | - loss: 0.6417 - accuracy: 0.7468 |
| 2/2 [=====]     | - | 0s 2ms/step                                                   | - loss: 0.6431 - accuracy: 0.7320 |
| Bpoch 964/2200  | - | 0s 3ms/step                                                   | - loss: 0.6471 - accuracy: 0.7394 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6534 - accuracy: 0.7357 |
| Bpoch 965/2200  | - | 0s 3ms/step                                                   | - loss: 0.6381 - accuracy: 0.7283 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6406 - accuracy: 0.7431 |
| Bpoch 968/2200  | - | 0s 4ms/step                                                   | - loss: 0.6560 - accuracy: 0.7431 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6547 - accuracy: 0.7357 |
| Bpoch 971/2200  | - | 0s 4ms/step                                                   | - loss: 0.6637 - accuracy: 0.7431 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6294 - accuracy: 0.7431 |
| Bpoch 973/2200  | - | 0s 4ms/step                                                   | - loss: 0.6587 - accuracy: 0.7357 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6797 - accuracy: 0.7283 |
| Bpoch 975/2200  | - | 0s 4ms/step                                                   | - loss: 0.6319 - accuracy: 0.7431 |
| 2/2 [=====]     | - | 0s 2ms/step                                                   | - loss: 0.6475 - accuracy: 0.7468 |
| Bpoch 977/2200  | - | 0s 3ms/step                                                   | - loss: 0.6360 - accuracy: 0.7431 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6316 - accuracy: 0.7505 |
| Bpoch 978/2200  | - | 0s 4ms/step                                                   | - loss: 0.6575 - accuracy: 0.7283 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6416 - accuracy: 0.7283 |
| Bpoch 980/2200  | - | 0s 5ms/step                                                   | - loss: 0.6654 - accuracy: 0.7283 |
| 2/2 [=====]     | - | 0s 6ms/step                                                   | - loss: 0.6511 - accuracy: 0.7320 |
| Bpoch 983/2200  | - | 0s 4ms/step                                                   | - loss: 0.6616 - accuracy: 0.7283 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6601 - accuracy: 0.7433 |
| Bpoch 987/2200  | - | 0s 3ms/step                                                   | - loss: 0.6407 - accuracy: 0.7433 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6276 - accuracy: 0.7544 |
| Bpoch 988/2200  | - | 0s 3ms/step                                                   | - loss: 0.6437 - accuracy: 0.7581 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6276 - accuracy: 0.7544 |
| Bpoch 990/2200  | - | 0s 2ms/step                                                   | - loss: 0.6466 - accuracy: 0.7433 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6400 - accuracy: 0.7470 |
| Bpoch 992/2200  | - | 0s 5ms/step                                                   | - loss: 0.6502 - accuracy: 0.7396 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6342 - accuracy: 0.7544 |
| Bpoch 993/2200  | - | 0s 3ms/step                                                   | - loss: 0.6482 - accuracy: 0.7396 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6428 - accuracy: 0.7433 |
| Bpoch 995/2200  | - | 0s 3ms/step                                                   | - loss: 0.6305 - accuracy: 0.7507 |
| 2/2 [=====]     | - | 0s 2ms/step                                                   | - loss: 0.6510 - accuracy: 0.7507 |
| Bpoch 997/2200  | - | 0s 3ms/step                                                   | - loss: 0.6635 - accuracy: 0.7433 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6107 - accuracy: 0.7618 |
| Bpoch 1000/2200 | - | 0s 3ms/step                                                   | - loss: 0.6604 - accuracy: 0.7285 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6702 - accuracy: 0.7211 |
| Bpoch 1002/2200 | - | 0s 3ms/step                                                   | - loss: 0.6384 - accuracy: 0.7396 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6284 - accuracy: 0.7581 |
| Bpoch 1004/2200 | - | 0s 3ms/step                                                   | - loss: 0.6412 - accuracy: 0.7544 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6608 - accuracy: 0.7359 |
| Bpoch 1007/2200 | - | 0s 3ms/step                                                   | - loss: 0.6584 - accuracy: 0.7248 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6513 - accuracy: 0.7322 |
| Bpoch 1008/2200 | - | 0s 4ms/step                                                   | - loss: 0.6345 - accuracy: 0.7507 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6494 - accuracy: 0.7432 |
| Bpoch 1010/2200 | - | 0s 4ms/step                                                   | - loss: 0.6648 - accuracy: 0.7322 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6416 - accuracy: 0.7470 |
| Bpoch 1012/2200 | - | 0s 3ms/step                                                   | - loss: 0.6345 - accuracy: 0.7544 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6150 - accuracy: 0.7507 |
| Bpoch 1014/2200 | - | 0s 4ms/step                                                   | - loss: 0.6257 - accuracy: 0.7322 |
| 2/2 [=====]     | - | 0s 2ms/step                                                   | - loss: 0.6464 - accuracy: 0.7285 |
| Bpoch 1017/2200 | - | 0s 4ms/step                                                   | - loss: 0.6275 - accuracy: 0.7433 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6467 - accuracy: 0.7433 |
| Bpoch 1019/2200 | - | 0s 3ms/step                                                   | - loss: 0.6321 - accuracy: 0.7544 |
| 2/2 [=====]     | - | 0s 2ms/step                                                   | - loss: 0.6275 - accuracy: 0.7359 |
| Bpoch 1022/2200 | - | 0s 3ms/step                                                   | - loss: 0.6439 - accuracy: 0.7361 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6549 - accuracy: 0.7398 |
| Bpoch 1024/2200 | - | 0s 3ms/step                                                   | - loss: 0.6304 - accuracy: 0.7509 |
| 2/2 [=====]     | - | 0s 3ms/step                                                   | - loss: 0.6527 - accuracy: 0.7361 |
| Bpoch 1027/2200 | - | 0s 3ms/step                                                   | - loss: 0.6436 - accuracy: 0.7472 |
| 2/2 [=====]     | - | 0s 4ms/step                                                   | - loss: 0.6367 - accuracy: 0.7435 |
| Bpoch 1029/2200 | - | 0s 3ms/step                                                   | - loss: 0.6488 - accuracy: 0.7435 |
| 2/2 [=====]     | - | 0s 2ms/step                                                   | - loss: 0.6195 - accuracy: 0.7583 |
| Bpoch 1030/2200 | - | 0s 2ms/step                                                   | - loss: 0.6389 - accuracy: 0.7472 |
| 2/              |   |                                                               |                                   |



|                 |             |   |    |          |                                   |
|-----------------|-------------|---|----|----------|-----------------------------------|
| Epoch 1248/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.6000 - accuracy: 0.7511 |
| Epoch 1249/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.6190 - accuracy: 0.7363 |
| Epoch 1250/2200 | 2/2 [=====] | - | 0s | 2ms/step | - loss: 0.6118 - accuracy: 0.7437 |
| Epoch 1251/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.5944 - accuracy: 0.7659 |
| Epoch 1252/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.5933 - accuracy: 0.7511 |
| Epoch 1253/2200 | 2/2 [=====] | - | 0s | 4ms/step | - loss: 0.6083 - accuracy: 0.7474 |
| Epoch 1254/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.5915 - accuracy: 0.7585 |
| Epoch 1255/2200 | 2/2 [=====] | - | 0s | 4ms/step | - loss: 0.5948 - accuracy: 0.7585 |
| Epoch 1256/2200 | 2/2 [=====] | - | 0s | 5ms/step | - loss: 0.5806 - accuracy: 0.7437 |
| Epoch 1257/2200 | 2/2 [=====] | - | 0s | 4ms/step | - loss: 0.6261 - accuracy: 0.7400 |
| Epoch 1258/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.5990 - accuracy: 0.7511 |
| Epoch 1259/2200 | 2/2 [=====] | - | 0s | 7ms/step | - loss: 0.6049 - accuracy: 0.7474 |
| Epoch 1260/2200 | 2/2 [=====] | - | 0s | 4ms/step | - loss: 0.5979 - accuracy: 0.7474 |
| Epoch 1261/2200 | 2/2 [=====] | - | 0s | 2ms/step | - loss: 0.6049 - accuracy: 0.7511 |
| Epoch 1262/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.5981 - accuracy: 0.7474 |
| Epoch 1263/2200 | 2/2 [=====] | - | 0s | 4ms/step | - loss: 0.5993 - accuracy: 0.7622 |
| Epoch 1264/2200 | 2/2 [=====] | - | 0s | 5ms/step | - loss: 0.5792 - accuracy: 0.7622 |
| Epoch 1265/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.5921 - accuracy: 0.7659 |
| Epoch 1266/2200 | 2/2 [=====] | - | 0s | 4ms/step | - loss: 0.5831 - accuracy: 0.7585 |
| Epoch 1267/2200 | 2/2 [=====] | - | 0s | 5ms/step | - loss: 0.6001 - accuracy: 0.7437 |
| Epoch 1268/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.5907 - accuracy: 0.7437 |
| Epoch 1269/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.5645 - accuracy: 0.7696 |
| Epoch 1270/2200 | 2/2 [=====] | - | 0s | 4ms/step | - loss: 0.5928 - accuracy: 0.7585 |
| Epoch 1271/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.5993 - accuracy: 0.7474 |
| Epoch 1272/2200 | 2/2 [=====] | - | 0s | 2ms/step | - loss: 0.6163 - accuracy: 0.7400 |
| Epoch 1273/2200 | 2/2 [=====] | - | 0s | 2ms/step | - loss: 0.5899 - accuracy: 0.7511 |
| Epoch 1274/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.5963 - accuracy: 0.7511 |
| Epoch 1275/2200 | 2/2 [=====] | - | 0s | 4ms/step | - loss: 0.5935 - accuracy: 0.7548 |
| Epoch 1276/2200 | 2/2 [=====] | - | 0s | 4ms/step | - loss: 0.5777 - accuracy: 0.7622 |
| Epoch 1277/2200 | 2/2 [=====] | - | 0s | 2ms/step | - loss: 0.5681 - accuracy: 0.7585 |
| Epoch 1278/2200 | 2/2 [=====] | - | 0s | 4ms/step | - loss: 0.5947 - accuracy: 0.7585 |
| Epoch 1279/2200 | 2/2 [=====] | - | 0s | 2ms/step | - loss: 0.5834 - accuracy: 0.7585 |
| Epoch 1280/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.5756 - accuracy: 0.7733 |
| Epoch 1281/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.5957 - accuracy: 0.7585 |
| Epoch 1282/2200 | 2/2 [=====] | - | 0s | 2ms/step | - loss: 0.6009 - accuracy: 0.7437 |
| Epoch 1283/2200 | 2/2 [=====] | - | 0s | 2ms/step | - loss: 0.5799 - accuracy: 0.7622 |
| Epoch 1284/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.5644 - accuracy: 0.7659 |
| Epoch 1285/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.5907 - accuracy: 0.7474 |
| Epoch 1286/2200 | 2/2 [=====] | - | 0s | 2ms/step | - loss: 0.6006 - accuracy: 0.7474 |
| Epoch 1287/2200 | 2/2 [=====] | - | 0s | 4ms/step | - loss: 0.5880 - accuracy: 0.7585 |
| Epoch 1288/2200 | 2/2 [=====] | - | 0s | 2ms/step | - loss: 0.5917 - accuracy: 0.7400 |
| Epoch 1289/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.5859 - accuracy: 0.7585 |
| Epoch 1290/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.6173 - accuracy: 0.7474 |
| Epoch 1291/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.5796 - accuracy: 0.7659 |
| Epoch 1292/2200 | 2/2 [=====] | - | 0s | 2ms/step | - loss: 0.5922 - accuracy: 0.7474 |
| Epoch 1293/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.5791 - accuracy: 0.7548 |
| Epoch 1294/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.5927 - accuracy: 0.7474 |
| Epoch 1295/2200 | 2/2 [=====] | - | 0s | 2ms/step | - loss: 0.5724 - accuracy: 0.7622 |
| Epoch 1296/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.5951 - accuracy: 0.7437 |
| Epoch 1297/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.5763 - accuracy: 0.7622 |
| Epoch 1298/2200 | 2/2 [=====] | - | 0s | 5ms/step | - loss: 0.5607 - accuracy: 0.7733 |
| Epoch 1299/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.5530 - accuracy: 0.7733 |
| Epoch 1300/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.6015 - accuracy: 0.7437 |
| Epoch 1301/2200 | 2/2 [=====] | - | 0s | 3ms/step | - loss: 0.5881 - accuracy: 0.7585 |
| Epoch 1302/2200 | 2/2 [=====] | - | 0s | 3ms      |                                   |



|                 |                 |                                                   |                                                               |
|-----------------|-----------------|---------------------------------------------------|---------------------------------------------------------------|
| Epoch 1666/2200 |                 | 0.401 - accuracy: 0.7895                          | ETA: 0s - loss: 0.5474 - accuracy: 0.77 - 0s 4ms/step - loss: |
| 2/2 [=====]     | Epoch 1667/2200 | - 0s 3ms/step - loss: 0.5177 - accuracy: 0.7891   |                                                               |
| 2/2 [=====]     | Epoch 1668/2200 | - 0s 4ms/step - loss: 0.5177 - accuracy: 0.7894   |                                                               |
| 2/2 [=====]     | Epoch 1669/2200 | - 0s 4ms/step - loss: 0.5131 - accuracy: 0.7928   |                                                               |
| 2/2 [=====]     | Epoch 1670/2200 | - 0s 5ms/step - loss: 0.5337 - accuracy: 0.7780   |                                                               |
| 2/2 [=====]     | Epoch 1671/2200 | - 0s 3ms/step - loss: 0.5097 - accuracy: 0.7894   |                                                               |
| 2/2 [=====]     | Epoch 1672/2200 | - 0s 3ms/step - loss: 0.5010 - accuracy: 0.8004   |                                                               |
| 2/2 [=====]     | Epoch 1673/2200 | - 0s 6ms/step - loss: 0.5227 - accuracy: 0.7708   |                                                               |
| 2/2 [=====]     | Epoch 1674/2200 | - 0s 4ms/step - loss: 0.5032 - accuracy: 0.7930   |                                                               |
| 2/2 [=====]     | Epoch 1675/2200 | - 0s 4ms/step - loss: 0.5240 - accuracy: 0.7782   |                                                               |
| 2/2 [=====]     | Epoch 1676/2200 | - 0s 3ms/step - loss: 0.5251 - accuracy: 0.8004   |                                                               |
| 2/2 [=====]     | Epoch 1677/2200 | - 0s 5ms/step - loss: 0.5096 - accuracy: 0.7930   |                                                               |
| 2/2 [=====]     | Epoch 1678/2200 | - 0s 3ms/step - loss: 0.5263 - accuracy: 0.7782   |                                                               |
| 2/2 [=====]     | Epoch 1679/2200 | - 0s 6ms/step - loss: 0.5201 - accuracy: 0.7819   |                                                               |
| 2/2 [=====]     | Epoch 1680/2200 | - 0s 4ms/step - loss: 0.5006 - accuracy: 0.7930   |                                                               |
| 2/2 [=====]     | Epoch 1681/2200 | - 0s 4ms/step - loss: 0.5013 - accuracy: 0.8043   |                                                               |
| 2/2 [=====]     | Epoch 1682/2200 | - 0s 4ms/step - loss: 0.5223 - accuracy: 0.7932   |                                                               |
| 2/2 [=====]     | Epoch 1683/2200 | - 0s 2ms/step - loss: 0.5129 - accuracy: 0.8006   |                                                               |
| 2/2 [=====]     | Epoch 1684/2200 | - 0s 4ms/step - loss: 0.5138 - accuracy: 0.7895   |                                                               |
| 2/2 [=====]     | Epoch 1685/2200 | - 0s 3ms/step - loss: 0.4981 - accuracy: 0.7932   |                                                               |
| 2/2 [=====]     | Epoch 1686/2200 | - 0s 5ms/step - loss: 0.4952 - accuracy: 0.8117   |                                                               |
| 2/2 [=====]     | Epoch 1687/2200 | - 0s 3ms/step - loss: 0.5094 - accuracy: 0.7969   |                                                               |
| 2/2 [=====]     | Epoch 1688/2200 | - 0s 3ms/step - loss: 0.4895 - accuracy: 0.8080   |                                                               |
| 2/2 [=====]     | Epoch 1689/2200 | - 0s 3ms/step - loss: 0.5218 - accuracy: 0.7895   |                                                               |
| 2/2 [=====]     | Epoch 1690/2200 | - 0s 3ms/step - loss: 0.5065 - accuracy: 0.8043   |                                                               |
| 2/2 [=====]     | Epoch 1691/2200 | - 0s 2ms/step - loss: 0.4995 - accuracy: 0.8043   |                                                               |
| 2/2 [=====]     | Epoch 1692/2200 | - 0s 3ms/step - loss: 0.4901 - accuracy: 0.8080   |                                                               |
| 2/2 [=====]     | Epoch 1693/2200 | - 0s 3ms/step - loss: 0.4917 - accuracy: 0.8043   |                                                               |
| 2/2 [=====]     | Epoch 1694/2200 | - 0s 3ms/step - loss: 0.5212 - accuracy: 0.7895   |                                                               |
| 2/2 [=====]     | Epoch 1695/2200 | - 0s 2ms/step - loss: 0.5389 - accuracy: 0.7784   |                                                               |
| 2/2 [=====]     | Epoch 1696/2200 | - 0s 4ms/step - loss: 0.5093 - accuracy: 0.8006   |                                                               |
| 2/2 [=====]     | Epoch 1697/2200 | - 0s 2ms/step - loss: 0.4968 - accuracy: 0.8043   |                                                               |
| 2/2 [=====]     | Epoch 1698/2200 | - 0s 2ms/step - loss: 0.5146 - accuracy: 0.7858   |                                                               |
| 2/2 [=====]     | Epoch 1699/2200 | - 0s 2ms/step - loss: 0.5010 - accuracy: 0.8043   |                                                               |
| 2/2 [=====]     | Epoch 1700/2200 | - 0s 5ms/step - loss: 0.5033 - accuracy: 0.8043   |                                                               |
| 2/2 [=====]     | Epoch 1701/2200 | - 0s 2ms/step - loss: 0.5209 - accuracy: 0.7932   |                                                               |
| 2/2 [=====]     | Epoch 1702/2200 | - 0s 4ms/step - loss: 0.5110 - accuracy: 0.8006   |                                                               |
| 2/2 [=====]     | Epoch 1703/2200 | - 0s 3ms/step - loss: 0.5190 - accuracy: 0.7784   |                                                               |
| 2/2 [=====]     | Epoch 1704/2200 | - 0s 3ms/step - loss: 0.5210 - accuracy: 0.8006   |                                                               |
| 2/2 [=====]     | Epoch 1705/2200 | - 0s 3ms/step - loss: 0.5282 - accuracy: 0.7858   |                                                               |
| 2/2 [=====]     | Epoch 1706/2200 | - 0s 3ms/step - loss: 0.5162 - accuracy: 0.8006   |                                                               |
| 2/2 [=====]     | Epoch 1707/2200 | - 0s 3ms/step - loss: 0.5084 - accuracy: 0.7858   |                                                               |
| 2/2 [=====]     | Epoch 1708/2200 | - 0s 3ms/step - loss: 0.4905 - accuracy: 0.8080   |                                                               |
| 2/2 [=====]     | Epoch 1709/2200 | - 0s 3ms/step - loss: 0.5094 - accuracy: 0.7969   |                                                               |
| 2/2 [=====]     | Epoch 1710/2200 | - 0s 4ms/step - loss: 0.5265 - accuracy: 0.7747   |                                                               |
| 2/2 [=====]     | Epoch 1711/2200 | - 0s 3ms/step - loss: 0.4900 - accuracy: 0.7969   |                                                               |
| 2/2 [=====]     | Epoch 1712/2200 | - 0s 3ms/step - loss: 0.4846 - accuracy: 0.8043   |                                                               |
| 2/2 [=====]     | Epoch 1713/2200 | - 0s 3ms/step - loss: 0.5241 - accuracy: 0.7821   |                                                               |
| 2/2 [=====]     | Epoch 1714/2200 | - 0s 3ms/step - loss: 0.5219 - accuracy: 0.7784   |                                                               |
| 2/2 [=====]     | Epoch 1715/2200 | - 0s 3ms/step - loss: 0.5095 - accuracy: 0.7895   |                                                               |
| 2/2 [=====]     | Epoch 1716/2200 | - 0s 3ms/step - loss: 0.5123 - accuracy: 0.7932   |                                                               |
| 2/2 [=====]     | Epoch 1717/2200 | - 0s 3ms/step - loss: 0.5058 - accuracy: 0.8006   |                                                               |
| 2/2 [=====]     | Epoch 1718/2200 | - 0s 3ms/step - loss: 0.4962 - accuracy: 0.8006   |                                                               |
| 2/2 [=====]     | Epoch 1719/2200 | - 0s 2ms/step - loss: 0.4984 - accuracy: 0.8006   |                                                               |
| 2/2 [=====]     | Epoch 1720/2200 | - 0s 3ms/step - loss: 0.5165 - accuracy: 0.7932   |                                                               |
| 2/2 [=====]     | Epoch 1721/2200 | - 0s 2ms/step - loss: 0.5038 - accuracy: 0.7932   |                                                               |
| 2/2 [=====]     | Epoch 1722/2200 | - 0s 2ms/step - loss: 0.5022 - accuracy: 0.7969   |                                                               |
| 2/2 [=====]     | Epoch 1723/2200 | - 0s 2ms/step - loss: 0.5193 - accuracy: 0.7747   |                                                               |
| 2/2 [=====]     | Epoch 1724/2200 | - 0s 4ms/step - loss: 0.4929 - accuracy: 0.7932   |                                                               |
| 2/2 [=====]     | Epoch 1725/2200 | - 0s 2ms/step - loss: 0.5113 - accuracy: 0.7821   |                                                               |
| 2/2 [=====]     | Epoch 1726/2200 | - 0s 3ms/step - loss: 0.4952 - accuracy: 0.8006</ |                                                               |



2/2 [=====] - 0s 2ms/step - loss: 0.4315 - accuracy: 0.8158  
Epoch 2088/2200  
2/2 [=====] - 0s 4ms/step - loss: 0.4240 - accuracy: 0.8158  
Epoch 2089/2200  
2/2 [=====] - 0s 4ms/step - loss: 0.4433 - accuracy: 0.8010  
Epoch 2090/2200  
2/2 [=====] - 0s 4ms/step - loss: 0.4267 - accuracy: 0.8121  
Epoch 2091/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4482 - accuracy: 0.8047  
Epoch 2092/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4443 - accuracy: 0.8014  
Epoch 2093/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4381 - accuracy: 0.8084  
Epoch 2094/2200  
2/2 [=====] - 0s 2ms/step - loss: 0.4342 - accuracy: 0.8084  
Epoch 2095/2200  
2/2 [=====] - 0s 2ms/step - loss: 0.4342 - accuracy: 0.8084  
Epoch 2096/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4454 - accuracy: 0.8047  
Epoch 2097/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4584 - accuracy: 0.7973  
Epoch 2098/2200  
2/2 [=====] - 0s 2ms/step - loss: 0.4453 - accuracy: 0.7936  
Epoch 2099/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4164 - accuracy: 0.8306  
Epoch 2100/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4376 - accuracy: 0.8047  
Epoch 2101/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4247 - accuracy: 0.8158  
Epoch 2102/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4038 - accuracy: 0.8232  
Epoch 2103/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4218 - accuracy: 0.8158  
Epoch 2104/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.428 - accuracy: 0.7973  
Epoch 2105/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4465 - accuracy: 0.8047  
Epoch 2106/2200  
2/2 [=====] - 0s 2ms/step - loss: 0.4380 - accuracy: 0.8084  
Epoch 2107/2200  
2/2 [=====] - 0s 2ms/step - loss: 0.4208 - accuracy: 0.8195  
Epoch 2108/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4362 - accuracy: 0.7973  
Epoch 2109/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4560 - accuracy: 0.8010  
Epoch 2110/2200  
2/2 [=====] - 0s 2ms/step - loss: 0.4290 - accuracy: 0.8121  
Epoch 2111/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4237 - accuracy: 0.8084  
Epoch 2112/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4365 - accuracy: 0.8010  
Epoch 2113/2200  
2/2 [=====] - 0s 2ms/step - loss: 0.4195 - accuracy: 0.8195  
Epoch 2114/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4145 - accuracy: 0.8047  
Epoch 2115/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4549 - accuracy: 0.8010  
Epoch 2116/2200  
2/2 [=====] - 0s 4ms/step - loss: 0.4317 - accuracy: 0.8084  
Epoch 2117/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4381 - accuracy: 0.8158  
Epoch 2118/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4213 - accuracy: 0.8084  
Epoch 2119/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4288 - accuracy: 0.8049  
Epoch 2120/2200  
2/2 [=====] - 0s 2ms/step - loss: 0.4021 - accuracy: 0.8345  
Epoch 2121/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4233 - accuracy: 0.8086  
Epoch 2122/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4304 - accuracy: 0.8160  
Epoch 2123/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4476 - accuracy: 0.8271  
Epoch 2124/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4006 - accuracy: 0.8308  
Epoch 2125/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4369 - accuracy: 0.8160  
Epoch 2126/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4350 - accuracy: 0.8160  
Epoch 2127/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4114 - accuracy: 0.8271  
Epoch 2128/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4237 - accuracy: 0.8234  
Epoch 2129/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4457 - accuracy: 0.8012  
Epoch 2130/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4176 - accuracy: 0.8197  
Epoch 2131/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4254 - accuracy: 0.8086  
Epoch 2132/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4179 - accuracy: 0.8308  
Epoch 2133/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4337 - accuracy: 0.8012  
Epoch 2134/2200  
2/2 [=====] - 0s 2ms/step - loss: 0.4231 - accuracy: 0.8197  
Epoch 2135/2200  
2/2 [=====] - 0s 2ms/step - loss: 0.4601 - accuracy: 0.7938  
Epoch 2137/2200  
2/2 [=====] - 0s 2ms/step - loss: 0.4202 - accuracy: 0.8123  
Epoch 2138/2200  
2/2 [=====] - 0s 2ms/step - loss: 0.4203 - accuracy: 0.8197  
Epoch 2139/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4334 - accuracy: 0.8123  
Epoch 2140/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4275 - accuracy: 0.8086  
Epoch 2141/2200  
2/2 [=====] - 0s 2ms/step - loss: 0.4075 - accuracy: 0.8271  
Epoch 2142/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4294 - accuracy: 0.8160  
Epoch 2143/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4188 - accuracy: 0.8197  
Epoch 2144/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.3976 - accuracy: 0.8382  
Epoch 2145/2200  
2/2 [=====] - 0s 2ms/step - loss: 0.4237 - accuracy: 0.8345  
Epoch 2146/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4456 - accuracy: 0.8012  
Epoch 2147/2200  
2/2 [=====] - 0s 2ms/step - loss: 0.4177 - accuracy: 0.8160  
Epoch 2148/2200  
2/2 [=====] - 0s 5ms/step - loss: 0.4257 - accuracy: 0.8160  
Epoch 2149/2200  
2/2 [=====] - 0s 4ms/step - loss: 0.4249 - accuracy: 0.8123  
Epoch 2150/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4297 - accuracy: 0.7936  
Epoch 2151/2200  
2/2 [=====] - 0s 4ms/step - loss: 0.4350 - accuracy: 0.8121  
Epoch 2152/2200  
2/2 [=====] - 0s 4ms/step - loss: 0.4462 - accuracy: 0.8047  
Epoch 2153/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4018 - accuracy: 0.8306  
Epoch 2154/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4365 - accuracy: 0.8047  
Epoch 2155/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4437 - accuracy: 0.8010  
Epoch 2156/2200  
2/2 [=====] - 0s 4ms/step - loss: 0.4196 - accuracy: 0.8084  
Epoch 2157/2200  
2/2 [=====] - 0s 4ms/step - loss: 0.4392 - accuracy: 0.7973  
Epoch 2158/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4215 - accuracy: 0.8047  
Epoch 2159/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4410 - accuracy: 0.7936  
Epoch 2160/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4301 - accuracy: 0.8047  
Epoch 2161/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4455 - accuracy: 0.7973  
Epoch 2162/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4221 - accuracy: 0.8195  
Epoch 2163/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4360 - accuracy: 0.8084  
Epoch 2164/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4264 - accuracy: 0.8121  
Epoch 2165/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4406 - accuracy: 0.7973  
Epoch 2166/2200  
2/2 [=====] - 0s 4ms/step - loss: 0.4101 - accuracy: 0.8232  
Epoch 2167/2200  
2/2 [=====] - 0s 4ms/step - loss: 0.4275 - accuracy: 0.8084  
Epoch 2168/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4252 - accuracy: 0.8047  
Epoch 2169/2200  
2/2 [=====] - 0s 4ms/step - loss: 0.4054 - accuracy: 0.8121  
Epoch 2170/2200  
2/2 [=====] - 0s 4ms/step - loss: 0.4251 - accuracy: 0.8084  
Epoch 2171/2200  
2/2 [=====] - 0s 2ms/step - loss: 0.4484 - accuracy: 0.8010  
Epoch 2172/2200  
2/2 [=====] - ETA: 0s - loss: 0.4348 - accuracy: 0.80 - 0s 3ms/step - loss: 0.4284 - accuracy: 0.8047  
Epoch 2173/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4461 - accuracy: 0.8047  
Epoch 2174/2200  
2/2 [=====] - 0s 2ms/step - loss: 0.4223 - accuracy: 0.8123  
Epoch 2175/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4232 - accuracy: 0.8012  
Epoch 2176/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4133 - accuracy: 0.8236  
Epoch 2177/2200  
2/2 [=====] - 0s 2ms/step - loss: 0.4275 - accuracy: 0.8199  
Epoch 2178/2200  
2/2 [=====] - 0s 4ms/step - loss: 0.4106 - accuracy: 0.8310  
Epoch 2179/2200  
2/2 [=====] - 0s 2ms/step - loss: 0.4275 - accuracy: 0.8088  
Epoch 2180/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4443 - accuracy: 0.8273  
Epoch 2181/2200  
2/2 [=====] - 0s 4ms/step - loss: 0.4446 - accuracy: 0.8012  
Epoch 2182/2200  
2/2 [=====] - 0s 2ms/step - loss: 0.4167 - accuracy: 0.8236  
Epoch 2183/2200  
2/2 [=====] - 0s 2ms/step - loss: 0.4272 - accuracy: 0.8125  
Epoch 2184/2200  
2/2 [=====] - 0s 2ms/step - loss: 0.4366 - accuracy: 0.8088  
Epoch 2185/2200  
2/2 [=====] - 0s 4ms/step - loss: 0.4271 - accuracy: 0.8088  
Epoch 2186/2200  
2/2 [=====] - 0s 4ms/step - loss: 0.4443 - accuracy: 0.8125  
Epoch 2187/2200  
2/2 [=====] - 0s 4ms/step - loss: 0.4121 - accuracy: 0.8162  
Epoch 2188/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4453 - accuracy: 0.8014  
Epoch 2189/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4295 - accuracy: 0.8049  
Epoch 2190/2200  
2/2 [=====] - 0s 4ms/step - loss: 0.4298 - accuracy: 0.8088  
Epoch 2191/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4406 - accuracy: 0.8051  
Epoch 2192/2200  
2/2 [=====] - 0s 4ms/step - loss: 0.4103 - accuracy: 0.8273  
Epoch 2193/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4314 - accuracy: 0.8125  
Epoch 2194/2200  
2/2 [=====] - 0s 4ms/step - loss: 0.4125 - accuracy: 0.8236  
Epoch 2195/2200  
2/2 [=====] - 0s 4ms/step - loss: 0.4201 - accuracy: 0.8236  
Epoch 2196/2200  
2/2 [=====] - 0s 5ms/step - loss: 0.4239 - accuracy: 0.8162  
Epoch 2197/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4411 - accuracy: 0.8088  
Epoch 2198/2200  
2/2 [=====] - 0s 5ms/step - loss: 0.4212 - accuracy: 0.8162  
Epoch 2199/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4108 - accuracy: 0.7977  
Epoch 2200/2200  
2/2 [=====] - 0s 3ms/step - loss: 0.4255 - accuracy: 0.8162  
2/2 [=====] - 0s 3ms/step - loss: 0.7264 - accuracy: 0.8372

Out[31]: [0.7263950705528259, 0.8372092843055725]

```
In [32]: _, train_acc = trial_model_4.evaluate(X_train, y_train, verbose=0)
_, test_acc = trial_model_4.evaluate(X_test, y_test, verbose=0)

print ('Train:%.3f, Test:%.3f' % (train_acc, test_acc))

Train:0.819, Test:0.837
```

### Accuracy scores

Results by changing values of Test size, epochs, activation and loss functions.

1. Train 0.899, Test 0.708
2. Train 0.760, Test 0.814
3. Train 0.760, Test 0.837

Change in epochs = 2200

1. Train 0.754, Test 0.860
2. Train 0.784, Test 0.837

Change in batch size = 100

1. Train 0.789, Test 0.814
2. Train 0.772, Test 0.814

Change in learning rate = 0.03, batch size= 90, epochs = 2200, random\_state= 4

We see improvements here in this stage!

1. Train 0.830, Test 0.767
2. Train 0.854, Test 0.860
3. Train 0.848, Test 0.814
4. Train 0.854, Test 0.837
5. Train 0.825, Test 0.837
6. Train 0.877, Test 0.837
7. Train 0.842, Test 0.814
8. Train 0.848, Test 0.837
9. Train 0.890, Test 0.837
10. Train 0.889, Test 0.814
11. Train 0.871, Test 0.860

Although the testing accuracy is showing high, the difference in percentage between train and test is varying for each run. I will improve the model further to see what the best model configuration for both test and train accuracy is.

## Model 5 - Final

1. Read the data in, preprocess and scale for the Final Dense Neural Network model

```
In [33]: column_names = ['id','Rt','Na','Mg','Al','Si','K','Ca','Ba','Fe','Glass_Type']

data = pd.read_csv('glass_data.csv',names = column_names,index_col = 0,header = None)

num_data = data.to_numpy()

dataset1 = num_data

X_cols = dataset1[:,0:9].astype(float)

y_cols = dataset1[:,9]

preprocessing.scale(X_cols, copy=False)

encoder = preprocessing.LabelEncoder()

encoder.fit(y_cols)

y_train = encoder.transform(y_cols).reshape(-1, 1)

y_enc = np_utils.to_categorical(y_train)

In [34]: y_enc.shape, X_cols.shape

Out[34]: ((214, 6), (214, 9))
```

1. The data frame created initially is the same format previously used for EDA and wish to keep the originality by presenting the same layout for data preparation and pre-processing for my final model. The index ID is used as the ID column in data frame.

2. The data after being read in is converted to Numpy array.

3. The target variable is separated from the independent variables for X and y variables.

4. I then scale the X\_cols data to fit the model in a min-max scale.

5. The y\_cols data is then label encoded to the respective values, doing so I know the data is in sequence and may trick the model in learning phase, due to the trickery that may happen further down in training the data I convert the labelled y\_cols data to categorical values and retain the classification using softmax as my final activation in the model.

```
In [35]: X_train, X_test, y_train, y_test = train_test_split(X_cols, y_enc, test_size=0.2, random_state=4)

In [36]: X_train.shape, y_train.shape

Out[36]: ((171, 9), (171, 6))

In [37]: X_test.shape, y_test.shape

Out[37]: ((43, 9), (43, 6))
```

### No.5 - ANN Final Model

```
In [38]: final_model = Sequential()

final_model.add(Dense(25, input_dim=9, kernel_initializer='normal', activation='relu'))

final_model.add(Dense(12, kernel_initializer='normal', activation='relu'))

final_model.add(Dense(6, kernel_initializer='normal', activation='softmax'))

final_model.compile(loss='categorical_crossentropy', optimizer=keras.optimizers.Adam(0.0003), metrics=[
'accuracy'])

final_model.summary()

Model: "sequential_4"

Layer (type) Output Shape Param #

dense_12 (Dense) (None, 25) 250
dense_13 (Dense) (None, 12) 312
dense_14 (Dense) (None, 6) 78
Total params: 640
Trainable params: 640
Non-trainable params: 0
```

Parameters used for the final model

- loss - I used categorical\_crossentropy as my loss function, to evaluate and learn the best probability from the label encoded vector of classification glass types. I would use sparse categorical cross entropy had I not encoded the target glass type labels and left them as integers.
- optimizer - I used Adam as my optimiser, additionally adding custom learning rate of 0.003 to improve models network weights. By leveraging the decaying average of previous squared gradients, it becomes efficient and adaptive throughout the train phase.
- epoch - set to 1900 for a close drop off before the model starts to over fit.
- metrics - The accuracy metric to check the model fit percentage wise, showing us a predictability outcome.
- Batch Size - I used the batch gradient descent using the total number of samples 214 from the dataset, works well with layers
- validation separation
- Layers - I have added 9 input layers, along with 3 hidden dense layers of 25,12 and 6 as standard input and outputs.
- Test size = a 70/30 split for training and testing the dataset at the beginning.
- random\_state = 4 to evaluate the randomness between the values with various magnitudes and to not be skewed to one side while the model learns through the training set.
- kernel\_initializer = 'normal' for normal distribution of weights through the neural layers.

### Model Fit







|                |     |       |   |    |          |   |              |   |                  |
|----------------|-----|-------|---|----|----------|---|--------------|---|------------------|
| Epoch 421/1900 | 1/1 | ===== | - | 0s | 4ms/step | - | loss: 1.0054 | - | accuracy: 0.6667 |
| Epoch 422/1900 | 1/1 | ===== | - | 0s | 4ms/step | - | loss: 1.0043 | - | accuracy: 0.6667 |
| Epoch 423/1900 | 1/1 | ===== | - | 0s | 4ms/step | - | loss: 1.0032 | - | accuracy: 0.6667 |
| Epoch 424/1900 | 1/1 | ===== | - | 0s | 3ms/step | - | loss: 1.0021 | - | accuracy: 0.6667 |
| Epoch 425/1900 | 1/1 | ===== | - | 0s | 3ms/step | - | loss: 1.0010 | - | accuracy: 0.6667 |
| Epoch 426/1900 | 1/1 | ===== | - | 0s | 4ms/step | - | loss: 0.9999 | - | accuracy: 0.6667 |
| Epoch 427/1900 | 1/1 | ===== | - | 0s | 4ms/step | - | loss: 0.9989 | - | accuracy: 0.6667 |
| Epoch 428/1900 | 1/1 | ===== | - | 0s | 3ms/step | - | loss: 0.9978 | - | accuracy: 0.6667 |
| Epoch 429/1900 | 1/1 | ===== | - | 0s | 4ms/step | - | loss: 0.9967 | - | accuracy: 0.6667 |
| Epoch 430/1900 | 1/1 | ===== | - | 0s | 4ms/step | - | loss: 0.9957 | - | accuracy: 0.6725 |
| Epoch 431/1900 | 1/1 | ===== | - | 0s | 7ms/step | - | loss: 0.9946 | - | accuracy: 0.6725 |
| Epoch 432/1900 | 1/1 | ===== | - | 0s | 3ms/step | - | loss: 0.9936 | - | accuracy: 0.6725 |
| Epoch 433/1900 | 1/1 | ===== | - | 0s | 3ms/step | - | loss: 0.9925 | - | accuracy: 0.6725 |
| Epoch 434/1900 | 1/1 | ===== | - | 0s | 5ms/step | - | loss: 0.9915 | - | accuracy: 0.6725 |
| Epoch 435/1900 | 1/1 | ===== | - | 0s | 3ms/step | - | loss: 0.9904 | - | accuracy: 0.6725 |
| Epoch 436/1900 | 1/1 | ===== | - | 0s | 3ms/step | - | loss: 0.9894 | - | accuracy: 0.6784 |
| Epoch 437/1900 | 1/1 | ===== | - | 0s | 3ms/step | - | loss: 0.9884 | - | accuracy: 0.6784 |
| Epoch 438/1900 | 1/1 | ===== | - | 0s | 4ms/step | - | loss: 0.9873 | - | accuracy: 0.6784 |
| Epoch 439/1900 | 1/1 | ===== | - | 0s | 5ms/step | - | loss: 0.9863 | - | accuracy: 0.6784 |
| Epoch 440/1900 | 1/1 | ===== | - | 0s | 4ms/step | - | loss: 0.9853 | - | accuracy: 0.6784 |
| Epoch 441/1900 | 1/1 | ===== | - | 0s | 3ms/step | - | loss: 0.9843 | - | accuracy: 0.6784 |
| Epoch 442/1900 | 1/1 | ===== | - | 0s | 4ms/step | - | loss: 0.9833 | - | accuracy: 0.6784 |
| Epoch 443/1900 | 1/1 | ===== | - | 0s | 4ms/step | - | loss: 0.9823 | - | accuracy: 0.6784 |
| Epoch 444/1900 | 1/1 | ===== | - | 0s | 4ms/step | - | loss: 0.9813 | - | accuracy: 0.6784 |
| Epoch 445/1900 | 1/1 | ===== | - | 0s | 4ms/step | - | loss: 0.9803 | - | accuracy: 0.6784 |
| Epoch 446/1900 | 1/1 | ===== | - | 0s | 8ms/step | - | loss: 0.9793 | - | accuracy: 0.6784 |
| Epoch 447/1900 | 1/1 | ===== | - | 0s | 4ms/step | - | loss: 0.9783 | - | accuracy: 0.6784 |
| Epoch 448/1900 | 1/1 | ===== | - | 0s | 4ms/step | - | loss: 0.9773 | - | accuracy: 0.6784 |
| Epoch 449/1900 | 1/1 | ===== | - | 0s | 6ms/step | - | loss: 0.9763 | - | accuracy: 0.6784 |
| Epoch 450/1900 | 1/1 | ===== | - | 0s | 5ms/step | - | loss: 0.9753 | - | accuracy: 0.6784 |
| Epoch 451/1900 | 1/1 | ===== | - | 0s | 5ms/step | - | loss: 0.9743 | - | accuracy: 0.6784 |
| Epoch 452/1900 | 1/1 | ===== | - | 0s | 4ms/step | - | loss: 0.9734 | - | accuracy: 0.6784 |
| Epoch 453/1900 | 1/1 | ===== | - | 0s | 3ms/step | - | loss: 0.9724 | - | accuracy: 0.6784 |
| Epoch 454/1900 | 1/1 | ===== | - | 0s | 3ms/step | - | loss: 0.9714 | - | accuracy: 0.6784 |
| Epoch 455/1900 | 1/1 | ===== | - | 0s | 3ms/step | - | loss: 0.9705 | - | accuracy: 0.6784 |
| Epoch 456/1900 | 1/1 | ===== | - | 0s | 4ms/step | - | loss: 0.9695 | - | accuracy: 0.6784 |
| Epoch 457/1900 | 1/1 | ===== | - | 0s | 3ms/step | - | loss: 0.9686 | - | accuracy: 0.6842 |
| Epoch 458/1900 | 1/1 | ===== | - | 0s | 3ms/step | - | loss: 0.9676 | - | accuracy: 0.6842 |
| Epoch 459/1900 | 1/1 | ===== | - | 0s | 4ms/step | - | loss: 0.9667 | - | accuracy: 0.6842 |
| Epoch 460/1900 | 1/1 | ===== | - | 0s | 4ms/step | - | loss: 0.9657 | - | accuracy: 0.6842 |
| Epoch 461/1900 | 1/1 | ===== | - | 0s | 3ms/step | - | loss: 0.9647 | - | accuracy: 0.6842 |
| Epoch 462/1900 | 1/1 | ===== | - | 0s | 3ms/step | - | loss: 0.9639 | - | accuracy: 0.6842 |
| Epoch 463/1900 | 1/1 | ===== | - | 0s | 3ms/step | - | loss: 0.9629 | - | accuracy: 0.6842 |
| Epoch 464/1900 | 1/1 | ===== | - | 0s | 5ms/step | - | loss: 0.9620 | - | accuracy: 0.6842 |
| Epoch 465/1900 | 1/1 | ===== | - | 0s | 5        |   |              |   |                  |



|                |  |   |    |          |   |              |   |                  |
|----------------|--|---|----|----------|---|--------------|---|------------------|
| Epoch 844/1900 |  | - | 0s | 4ms/step | - | loss: 0.7554 | - | accuracy: 0.6959 |
| Epoch 845/1900 |  | - | 0s | 4ms/step | - | loss: 0.7551 | - | accuracy: 0.6959 |
| 1/1 [=====]    |  | - | 0s | 3ms/step | - | loss: 0.7551 | - | accuracy: 0.6959 |
| Epoch 846/1900 |  | - | 0s | 3ms/step | - | loss: 0.7548 | - | accuracy: 0.6959 |
| Epoch 847/1900 |  | - | 0s | 3ms/step | - | loss: 0.7545 | - | accuracy: 0.6959 |
| 1/1 [=====]    |  | - | 0s | 3ms/step | - | loss: 0.7541 | - | accuracy: 0.6959 |
| Epoch 849/1900 |  | - | 0s | 3ms/step | - | loss: 0.7538 | - | accuracy: 0.6959 |
| Epoch 850/1900 |  | - | 0s | 2ms/step | - | loss: 0.7535 | - | accuracy: 0.6959 |
| 1/1 [=====]    |  | - | 0s | 3ms/step | - | loss: 0.7532 | - | accuracy: 0.6959 |
| Epoch 852/1900 |  | - | 0s | 4ms/step | - | loss: 0.7529 | - | accuracy: 0.6959 |
| 1/1 [=====]    |  | - | 0s | 4ms/step | - | loss: 0.7526 | - | accuracy: 0.6959 |
| Epoch 854/1900 |  | - | 0s | 3ms/step | - | loss: 0.7522 | - | accuracy: 0.6959 |
| Epoch 855/1900 |  | - | 0s | 5ms/step | - | loss: 0.7519 | - | accuracy: 0.6959 |
| 1/1 [=====]    |  | - | 0s | 3ms/step | - | loss: 0.7516 | - | accuracy: 0.6959 |
| Epoch 857/1900 |  | - | 0s | 3ms/step | - | loss: 0.7513 | - | accuracy: 0.6959 |
| 1/1 [=====]    |  | - | 0s | 3ms/step | - | loss: 0.7510 | - | accuracy: 0.6959 |
| Epoch 859/1900 |  | - | 0s | 3ms/step | - | loss: 0.7507 | - | accuracy: 0.6959 |
| Epoch 860/1900 |  | - | 0s | 3ms/step | - | loss: 0.7504 | - | accuracy: 0.6959 |
| Epoch 861/1900 |  | - | 0s | 3ms/step | - | loss: 0.7501 | - | accuracy: 0.6959 |
| Epoch 862/1900 |  | - | 0s | 4ms/step | - | loss: 0.7498 | - | accuracy: 0.6959 |
| 1/1 [=====]    |  | - | 0s | 3ms/step | - | loss: 0.7494 | - | accuracy: 0.6959 |
| Epoch 864/1900 |  | - | 0s | 4ms/step | - | loss: 0.7491 | - | accuracy: 0.6959 |
| 1/1 [=====]    |  | - | 0s | 4ms/step | - | loss: 0.7488 | - | accuracy: 0.6959 |
| Epoch 866/1900 |  | - | 0s | 3ms/step | - | loss: 0.7485 | - | accuracy: 0.6959 |
| Epoch 867/1900 |  | - | 0s | 4ms/step | - | loss: 0.7482 | - | accuracy: 0.6959 |
| 1/1 [=====]    |  | - | 0s | 3ms/step | - | loss: 0.7479 | - | accuracy: 0.7018 |
| Epoch 869/1900 |  | - | 0s | 3ms/step | - | loss: 0.7476 | - | accuracy: 0.7018 |
| 1/1 [=====]    |  | - | 0s | 3ms/step | - | loss: 0.7473 | - | accuracy: 0.7018 |
| Epoch 871/1900 |  | - | 0s | 4ms/step | - | loss: 0.7470 | - | accuracy: 0.7018 |
| Epoch 872/1900 |  | - | 0s | 4ms/step | - | loss: 0.7467 | - | accuracy: 0.7018 |
| 1/1 [=====]    |  | - | 0s | 4ms/step | - | loss: 0.7464 | - | accuracy: 0.7018 |
| Epoch 874/1900 |  | - | 0s | 4ms/step | - | loss: 0.7461 | - | accuracy: 0.7018 |
| 1/1 [=====]    |  | - | 0s | 4ms/step | - | loss: 0.7458 | - | accuracy: 0.7018 |
| Epoch 876/1900 |  | - | 0s | 3ms/step | - | loss: 0.7455 | - | accuracy: 0.7018 |
| Epoch 877/1900 |  | - | 0s | 3ms/step | - | loss: 0.7452 | - | accuracy: 0.7018 |
| Epoch 878/1900 |  | - | 0s | 3ms/step | - | loss: 0.7449 | - | accuracy: 0.7018 |
| Epoch 879/1900 |  | - | 0s | 3ms/step | - | loss: 0.7446 | - | accuracy: 0.7018 |
| 1/1 [=====]    |  | - | 0s | 3ms/step | - | loss: 0.7443 | - | accuracy: 0.7018 |
| Epoch 881/1900 |  | - | 0s | 3ms/step | - | loss: 0.7440 | - | accuracy: 0.7018 |
| Epoch 882/1900 |  | - | 0s | 3ms/step | - | loss: 0.7437 | - | accuracy: 0.7018 |
| 1/1 [=====]    |  | - | 0s | 3ms/step | - | loss: 0.7434 | - | accuracy: 0.7018 |
| Epoch 883/1900 |  | - | 0s | 3ms/step | - | loss: 0.7431 | - | accuracy: 0.7018 |
| Epoch 884/1900 |  | - | 0s | 3ms/step | - | loss: 0.7428 | - | accuracy: 0.7018 |
| 1/1 [=====]    |  | - | 0s | 3ms/step | - | loss: 0.7425 | - | accuracy: 0.7018 |
| Epoch 886/1900 |  | - | 0s | 4ms/step | - | loss: 0.7422 | - | accuracy: 0.7018 |
| Epoch 888/1900 |  | - | 0s | 3ms/step | - | loss: 0.7419 | - | accuracy: 0.7018 |
| Epoch 889/1900 |  | - | 0s | 3ms/step | - | loss: 0.7416 | - | accuracy: 0.7018 |
| 1/1 [=====]    |  | - | 0s | 3ms/step | - | loss: 0.7413 | - | accuracy: 0.7018 |
| Epoch 891/1900 |  | - | 0s | 4ms/step | - | loss: 0.7410 | - | accuracy: 0.7018 |
| 1/1 [=====]    |  | - | 0s | 3ms/step | - | loss: 0.7407 | - | accuracy: 0.7018 |
| Epoch 893/1900 |  | - | 0s | 3ms/step | - | loss: 0.7404 | - | accuracy: 0.7018 |



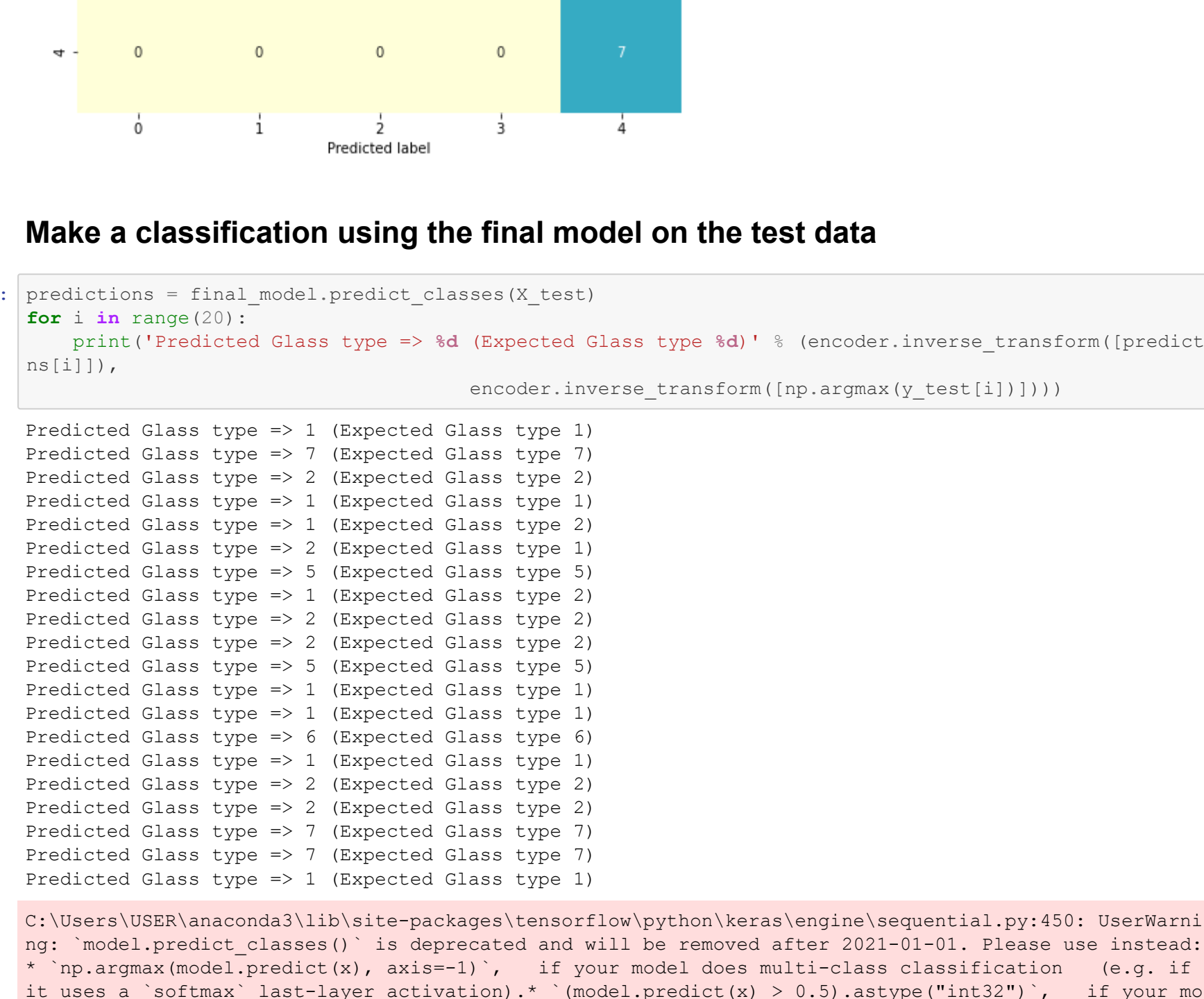




[illegible]

|     |    |    |
|-----|----|----|
| not | 3  | 13 |
| yes | 10 | 10 |

|       |   |   |   |   |   |
|-------|---|---|---|---|---|
| True  | 0 | 0 | 2 | 0 | 1 |
| False | 1 | 0 | 0 | 1 | 0 |



```
del does binary classification (e.g. if it uses a 'sigmoid' las
warnings.warn('model.predict_classes()' is deprecated and '
```

The classification reflects the accuracy achieved of the final model; we can see most values predicted to be accurate to the percentage of the model accuracy.

The model predictions are taken from the shuffled dataset values of the training and testing datapoints model and then reverted back to make predictions, this is the reason why the data points are different to the order of the datapoints in the original index of the dataset.

### Review of accuracy ranges from the Final Model and others

The range of Accuracy from my final model is between 78% - 88% and I can see the Training and Testing difference in percentage is most close to each other. Due to the stochastic nature of the model the results can vary for each test run.

Challenges during the training phase various models I tried where to get the testing and training accuracy score as close as possible and reduce the sum squared error substantially in loss.

begin to see results as I introduced random state values and learning rate step count for the model, to lower the impact in skewed learning from an imbalanced scale of rate/magnitude of values while training. Doing so I was able to lower the loss and increase the prediction accuracy of the testing set data towards the Final model.

The distribution of weights where also a challenge as I had introduced hyperparameters like custom learning rate and trialling the model's accuracy in test each time to fit the model until the epochs and batch sizes where optimal fit.

A few glass types where dominant in the dataset, also the samples provided in the dataset where small to work with, by label-encoding an

transforming the target data to categorical values I opted to use categorical cross entropy as my loss function mostly. If the values were to remain as integers in target I estimated the learning outcome would be of somewhat of an advanced method that could bring up the accuracy score between the training and testing test.

In many of my model training runs I had managed to reduce the loss substantially without overfitting the data and getting an agreeable accuracy level to make predictions as shown above in the predicted classifications.

## References

1. Aarshay, J. March 1, 2016. Complete Guide to Parameter Tuning in XGBoost with codes in Python. [Online]. [23 May 2021]. Available from: <https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/>
2. Analyticsvidhya. October 6, 2020. How to Improve Class Imbalance using Class Weights in Machine Learning. [Online]. [22 May 2021]. Available from: <https://www.analyticsvidhya.com/blog/2020/10/improve-class-imbalance-class-weights/>

1. Brownlee, J. April 8, 2020. 4 Types of Classification Tasks in Machine Learning. [Online]. [21 May 2021]. Available from: <https://machinelearningmastery.com/types-of-classification-in-machine-learning/>

2. Bitor, S. May 26, 2020. How to Create a Seaborn Correlation Heatmap in Python? [Online]. [22 May 2021]. Available from: <https://medium.com/@szabo.bitor/how-to-create-a-seaborn-correlation-heatmap-in-python-834c0686b88e>

3. Geeksforgeeks. 05 Sep, 2020. Python Keras() kerasutils.categorical(). [Online]. [21 May 2021]. Available from: [https://www.geeksforgeeks.org/python-keras-keras-utils-to\\_categorical/](https://www.geeksforgeeks.org/python-keras-keras-utils-to_categorical/)

4. Petlarian. 'No date'. Categorical crossentropy. [Online]. [23 May 2021]. Available from: <https://petlarian.com/knowledge/>

center/documentation/multilabel-view/build-an-ai-model/loss-functions/categorical-crossentropy

5. Roy, B. April 6, 2020. All about Feature Scaling. [Online]. [21 May 2021]. Available from: <https://towardsdatascience.com/all-about-feature-scaling-bcc0ad75c335>

6. Sachin, K. Dec 16, 2018. Testing the model's accuracy in ML. [Online]. [23 May 2021]. Available from: <https://sachinkr375.medium.com/testing-the-models-accuracy-in-ml-8335e4d443ef>

7. Satyam, K. Jun 9, 2020. Overview of various Optimizers in Neural Networks. [Online]. [22 May 2021]. Available from:

[3] Shivam K. No 18. 2019. Understanding a Classification Report For Your Machine Learning Model. [Online]. [22 May 2021]. Available from: <https://www.analyticsvidhya.com/blog/2019/05/understaing-a-classification-report-in-neural-networks-17-c16b2d0a0>

[4] <https://medium.com/@kohishivam5522/understanding-a-classification-report-for-your-machine-learning-model-88815e2c397>

[5] Stackoverflow. 'No date'. How to interpret loss and accuracy for a machine learning model [closed]. [Online]. [21 May 2021]. Available from: <https://stackoverflow.com/questions/34518656/how-to-interpret-loss-and-accuracy-for-a-machine-learning-model>

[6] Stackoverflow. 'No date'. Issues using Keras np\_utilato. [Online]. [21 May 2021]. Available from: <https://stackoverflow.com/questions/41494625/issues-using-keras-np-util-to-categorical>

[7] Sunny S., Jul 30, 2018. Label Encoder vs One Hot Encoder in Machine Learning. [Online]. [22 May 2021]. Available from: <https://www.analyticsvidhya.com/blog/2018/07/label-encoder-vs-one-hot-encoder-in-machine-learning/>

7. Uniqtech. Jan 30, 2018. Understand the Softmax Function in Minutes. [Online]. [22 May 2021]. Available from: <https://medium.com/data-science-bootcamp/understand-the-softmax-function-in-minutes-f3a50641e86d>