Your Uni : am6490, cj2831, hk3354

Your Full name : Arsh Misra, Conor Jones, Flora Kwon

Link to your Public Github repository with Final report:

https://github.com/hyerhinkwon/QMSS5074-Adv-ML.git

Submission Due Date: 03/07/2025

World Happiness Classification Competition

Goals :

- Understand how the models function
- Understand what the parameters control
- Learn from the model experimentation process
- Make a good looking notebook report
- Upload as a personal project on Github

Overall Steps:

1. Load datasets and merge them.
2. Preprocess data using Sklearn Column Transformer/ Write and Save Preprocessor function
3. Fit model on preprocessed data and save preprocessor function and model
4. Generate predictions from X_test data and submit predictions

0. Loading Datasets

Loading the World Happiness 2023 datasets

```
# Import libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split

#Load the dataset
whr_df = pd.read_csv('https://raw.githubusercontent.com/hyerhinkwon/QMSS5074-Adv-

# Inspect the first few rows to understand the structure
whr_df
```

	country	region	happiness_score	gdp_per_capita	social_support	health
0	Finland	Western Europe	7.804	1.888	1.585	
1	Denmark	Western Europe	7.586	1.949	1.548	
2	Iceland	Western Europe	7.530	1.926	1.620	
3	Israel	Middle East and North Africa	7.473	1.833	1.521	
4	Netherlands	Western Europe	7.403	1.942	1.488	
...	...	...	...	...	...	
132	Congo (Kinshasa)	Sub-Saharan Africa	3.207	0.531	0.784	
133	Zimbabwe	Sub-Saharan Africa	3.204	0.758	0.881	
134	Sierra Leone	Sub-Saharan Africa	3.138	0.670	0.540	
135	Lebanon	Middle East and North Africa	2.392	1.417	0.476	
136	Afghanistan	South Asia	1.859	0.645	0.000	

137 rows × 9 columns

Next steps: Generate code with `whr_df` View recommended plots New interactive sheet

```
# Convert the regression target ('happiness_score') into classification labels
# We'll use quartiles to create 5 happiness categories: Very Low, Low, Average, Hig

# Define quartiles
whr_df['happiness_category'] = pd.qcut(whr_df['happiness_score'],
                                       q=5,
                                       labels=['Very Low', 'Low','Average', 'High',

# Select features and target
X = whr_df.drop(columns=['happiness_score', 'happiness_category'])
y = whr_df['happiness_category']

# Split into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_sta

# Convert y_train and y_test to numerical labels
y_train_labels = y_train.astype('category').cat.codes
# y_test_labels = ## Complete in a similar manner as above
y_test_labels = y_test.astype('category').cat.codes
```

```
X_train = X_train.reset_index(drop=True)
X_test = X_test.reset_index(drop=True)
y_train = y_train.reset_index(drop=True)
y_test = y_test.reset_index(drop=True)
y_train_labels = y_train.reset_index(drop=True)
y_test_labels = y_test.reset_index(drop=True)
```

Write in the next cell what the y_train.astype('category').cat.codes line does. What is the difference between y_train_labels and y_train?

Your answer: y_train.astype('category').cat.codes assigns numeric codes to categorical variables. y_train contains the original categorical variables in its original format and y_train_labels contains the numeric codes corresponding to each category in y_train.

Add new data

```
# Truncated and cleaned up region data to merge
countrydata=pd.read_csv('https://raw.githubusercontent.com/hyerhinkwon/QMSS5074-A

countrydata.head()
```

	country_name	population	population_below_poverty_line	hdi	life_expec
0	India	1339180127	21.9	0.623559	
1	Nigeria	190886311	70.0	0.527105	
2	Mexico	129163276	46.2	0.761683	
3	Pakistan	197015955	29.5	0.550354	
4	Bangladesh	164669751	31.5	0.578824	

Next steps: Generate code with countrydata | View recommended plots | New interactive sheet

```
# Merge in new data to X_train and X_test by taking "country" from first table an
# Also check which countries are common in both the datasets, and which type of m
# Hint: Look on the 'how' parameter of merge function of pandas.

# Check common countries.
X_train_common = set(X_train['country']).intersection(set(countrydata['country_na
print(X_train_common)
X_test_common = set(X_test['country']).intersection(set(countrydata['country_name
print(X_test_common)

# Merge
X_train = pd.merge(X_train, countrydata, left_on='country', right_on='country_nam
X_test = pd.merge(X_test, countrydata, left_on='country', right_on='country_name'
```

```
{'Saudi Arabia', 'China', 'Brazil', 'Kosovo', 'Serbia', 'Niger', 'Latvia', 'Ja
{'United Arab Emirates', 'Chad', 'Mozambique', 'Switzerland', 'Tunisia', 'Gua
```

```
X_train.head(1)
```

	country	region	gdp_per_capita	social_support	healthy_life_expectancy
0	Madagascar	Sub-Saharan Africa	0.632	0.779	0.178

Next steps: Generate code with X_train | View recommended plots | New interactive sheet

1. EDA

```
print(X_train.dtypes)
```

```
country                          object
region                           object
gdp_per_capita                  float64
social_support                  float64
healthy_life_expectancy         float64
freedom_to_make_life_choices    float64
generosity                      float64
perceptions_of_corruption       float64
country_name                     object
population                      float64
population_below_poverty_line   float64
hdi                             float64
life_expectancy                 float64
expected_years_of_schooling     float64
mean_years_of_schooling         float64
gni                             float64
dtype: object
```

Describe what you see above?

```
# Your answer:
# The above describes what data each variables have.
# 'object' refers to categorical variables while 'float64' refers to numerical va
```

Find out the number and percentage of missing values in the table per column

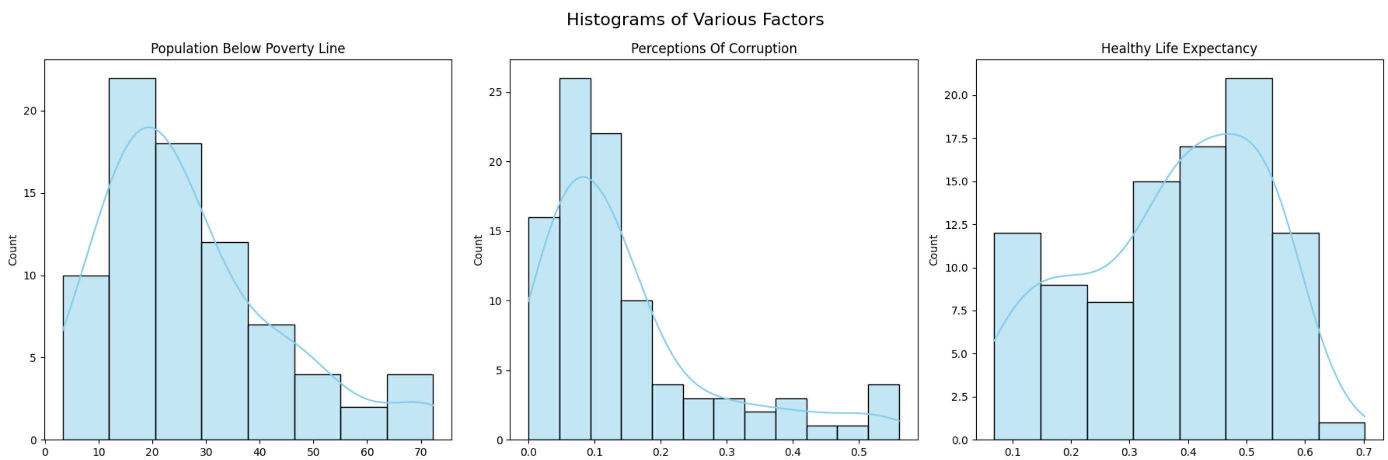
```
# Your code here:
missingvalues_X_train = X_train.isnull().mean()
missingvalues_X_train
```

	0
country	0.000000
region	0.000000
gdp_per_capita	0.000000
social_support	0.000000
healthy_life_expectancy	0.000000
freedom_to_make_life_choices	0.000000
generosity	0.000000
perceptions_of_corruption	0.000000
country_name	0.063158
population	0.063158
population_below_poverty_line	0.168421
hdi	0.063158
life_expectancy	0.073684
expected_years_of_schooling	0.073684
mean_years_of_schooling	0.073684
gni	0.073684

dtype: float64

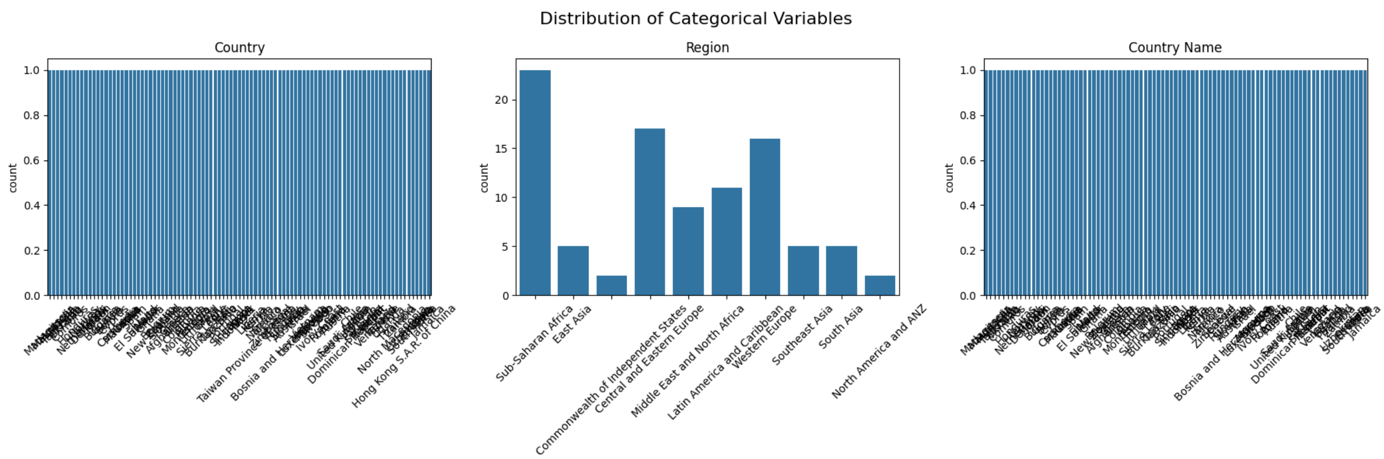
Plot the frequency distribution / histogram of some of the numerical features that you think are important

```
# Your plotting code here:

# Import libraries
import matplotlib.pyplot as plt
import seaborn as sns

# Create the histogram
variables = ['population_below_poverty_line', 'perceptions_of_corruption', 'healt
fig, axes = plt.subplots(1, 3, figsize=(18, 6))
fig.suptitle('Histograms of Various Factors', fontsize=16)
for i, var in enumerate(variables):
    sns.histplot(data=X_train, x=var, kde=True, color='skyblue', edgecolor='black
    axes[i].set_title(var.replace("_", " ").title())
    axes[i].set_xlabel('')

plt.tight_layout()
plt.show()
```



Plot the categorical variables and their distribution

```
# Your plotting code here:
cat_variables = ['country', 'region', 'country_name']
fig, axes = plt.subplots(1, 3, figsize=(18, 6))
fig.suptitle('Distribution of Categorical Variables', fontsize=16)
for i, var in enumerate(cat_variables):
    sns.countplot(data=X_train, x=var, ax=axes[i])
    axes[i].set_title(var.replace("_", " ").title())
    axes[i].set_xlabel('')
    axes[i].tick_params(axis='x', rotation=45)

plt.tight_layout()
plt.show()
```



Perform feature correlation analysis to identify relationships between variables. Use Pearson, Spearman, or Kendall correlation coefficients to analyze feature dependencies.

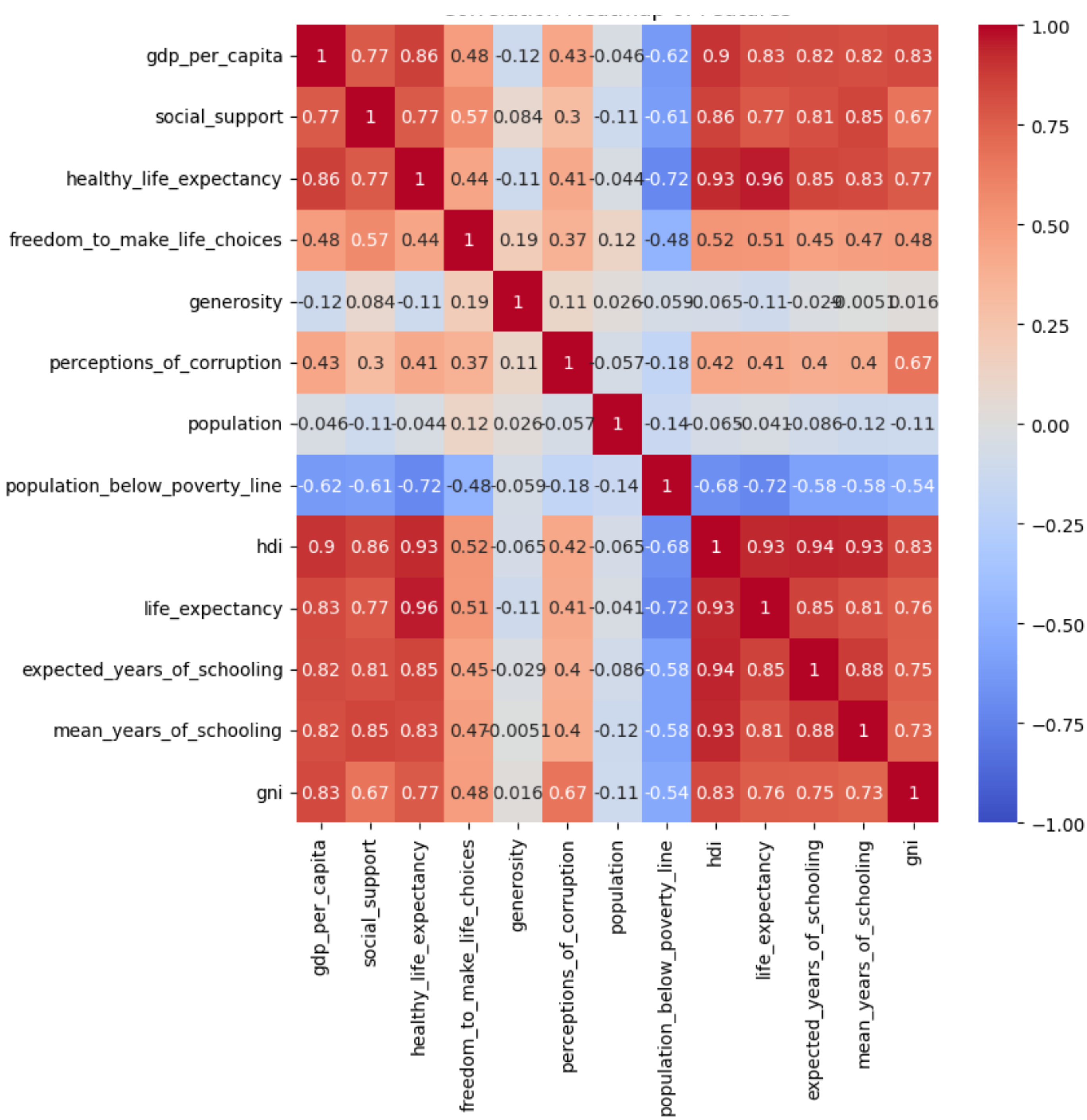
```
# Your code here:

numerical_X_train = X_train.select_dtypes(include='float64')
numerical_X_train.corr(method='pearson')
```

	gdp_per_capita	social_support	healthy_life_expect
gdp_per_capita	1.000000	0.769394	0.86
social_support	0.769394	1.000000	0.77
healthy_life_expectancy	0.860164	0.771294	1.00
freedom_to_make_life_choices	0.483145	0.571646	0.44
generosity	-0.115580	0.084327	-0.11
perceptions_of_corruption	0.432076	0.301807	0.41
population	-0.045583	-0.109651	-0.04
population_below_poverty_line	-0.620790	-0.605304	-0.71
hdi	0.902386	0.855297	0.92
life_expectancy	0.830886	0.774096	0.95
expected_years_of_schooling	0.820940	0.809587	0.84
mean_years_of_schooling	0.815423	0.845490	0.83
gni	0.834177	0.667188	0.76

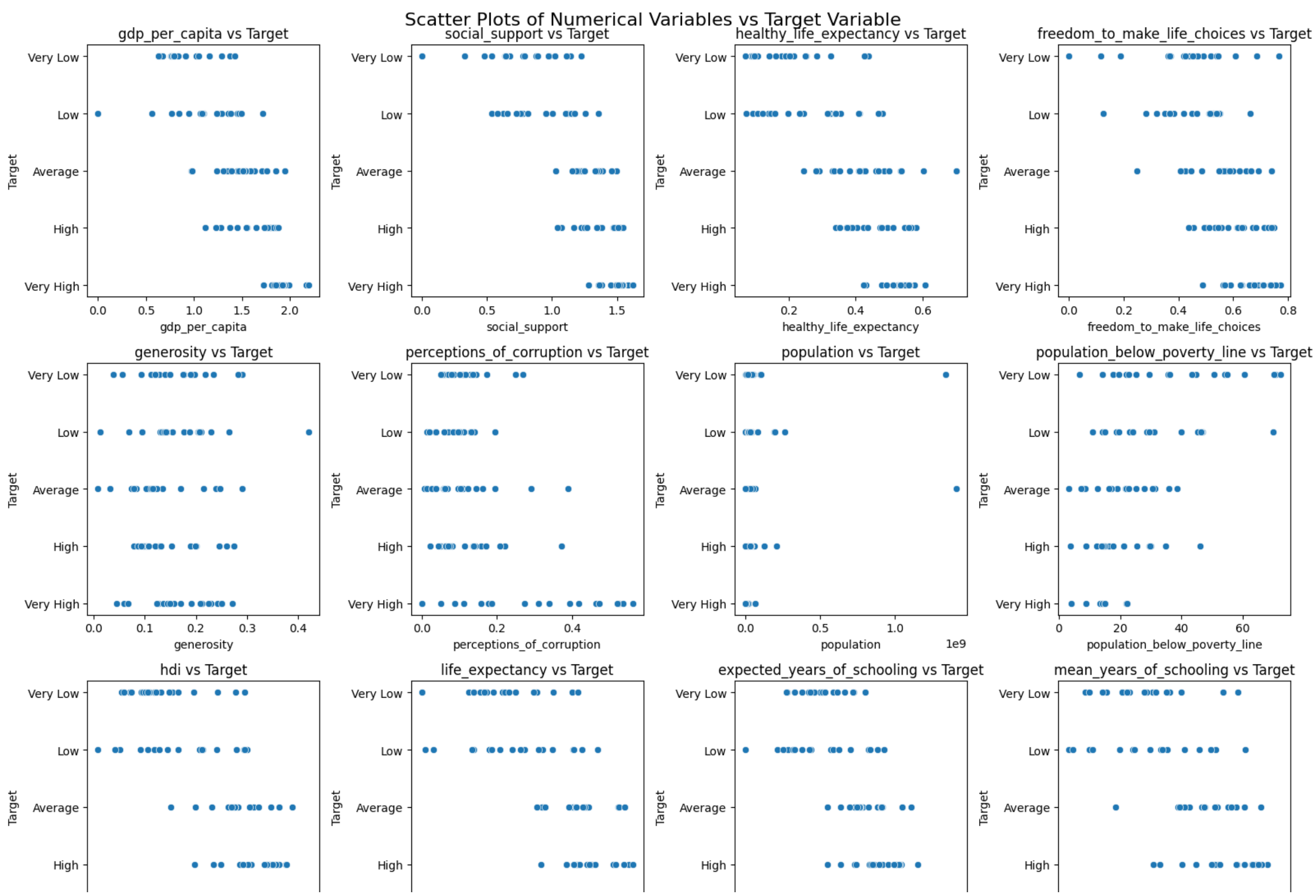
Explore relationships between variables (bivariate, etc), correlation tables, and how they associate with the target variable.

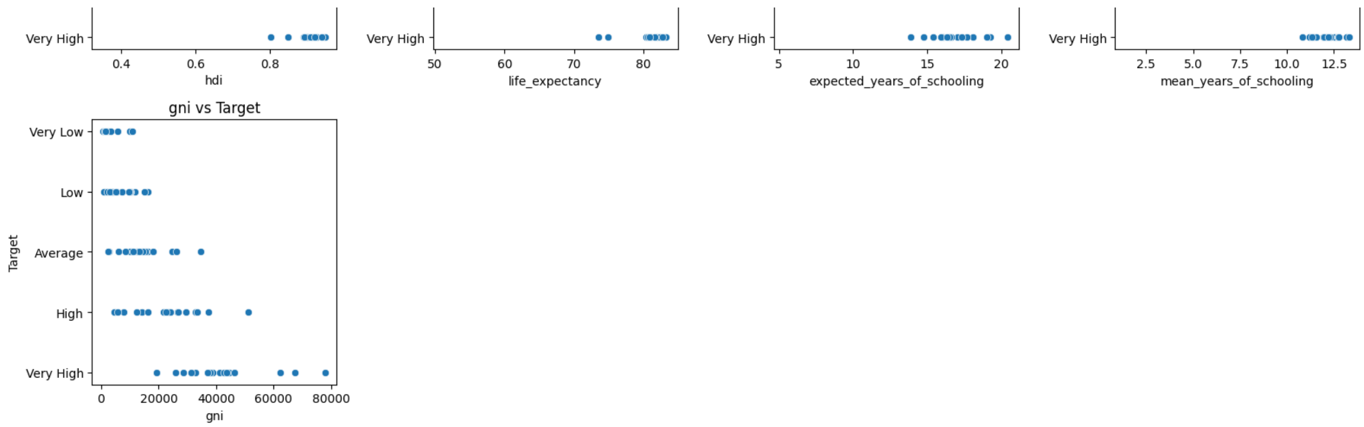
```
# Your plotting code(s) here:

# Calculate correlation.
correlation_matrix = numerical_X_train.corr()
plt.figure(figsize=(8, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Heatmap of Features')
plt.show()
```

Correlation Heatmap of Features


gdp_per_capita 1 0.77 0.86 0.48 -0.12 0.43 -0.046 -0.62 0.9 0.83 0.82 0.82 0.83
social_support 0.77 1 0.77 0.57 0.084 0.3 -0.11 -0.61 0.86 0.77 0.81 0.85 0.67
healthy_life_expectancy 0.86 0.77 1 0.44 -0.11 0.41 -0.044 -0.72 0.93 0.96 0.85 0.83 0.77
freedom_to_make_life_choices 0.48 0.57 0.44 1 0.19 0.37 0.12 -0.48 0.52 0.51 0.45 0.47 0.48
generosity -0.12 0.084 -0.11 0.19 1 0.11 0.026 -0.059 -0.065 -0.11 -0.029 -0.0051 0.016
perceptions_of_corruption 0.43 0.3 0.41 0.37 0.11 1 -0.057 -0.18 0.42 0.41 0.4 0.4 0.67
population -0.046 -0.11 -0.044 0.12 0.026 -0.057 1 -0.14 -0.065 -0.041 -0.086 -0.12 -0.11
population_below_poverty_line -0.62 -0.61 -0.72 -0.48 -0.059 -0.18 -0.14 1 -0.68 -0.72 -0.58 -0.58 -0.54
hdi 0.9 0.86 0.93 0.52 -0.065 0.42 -0.065 -0.68 1 0.93 0.94 0.93 0.83
life_expectancy 0.83 0.77 0.96 0.51 -0.11 0.41 -0.041 -0.72 0.93 1 0.85 0.81 0.76
expected_years_of_schooling 0.82 0.81 0.85 0.45 -0.029 0.4 -0.086 -0.58 0.94 0.85 1 0.88 0.75
mean_years_of_schooling 0.82 0.85 0.83 0.47 -0.0051 0.4 -0.12 -0.58 0.93 0.81 0.88 1 0.73
gni 0.83 0.67 0.77 0.48 0.016 0.67 -0.11 -0.54 0.83 0.76 0.75 0.73 1
gdp_per_capita
social_support
healthy_life_expectancy
freedom_to_make_life_choices
generosity
perceptions_of_corruption
population
population_below_poverty_line
hdi
life_expectancy
expected_years_of_schooling
mean_years_of_schooling
gni
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

```
# How it relates to target feature.
fig, axes = plt.subplots(4, 4, figsize=(16, 16))
fig.suptitle('Scatter Plots of Numerical Variables vs Target Variable', fontsize=

axes = axes.flatten()

# Create scatter plots
num_plots = min(len(numerical_X_train.columns), len(axes))
for i, column in enumerate(numerical_X_train.columns[:num_plots]):
    sns.scatterplot(x=numerical_X_train[column], y=y_train, ax=axes[i])
    axes[i].set_xlabel(column)
    axes[i].set_ylabel('Target')
    axes[i].set_title(f'{column} vs Target')

for j in range(num_plots, len(axes)):
    axes[j].set_visible(False)

plt.tight_layout()
plt.show()
```




Very High
hdi
life_expectancy
expected_years_of_schooling
mean_years_of_schooling
gni vs Target
Very Low
Low
Average
High
Very High
Target
gni

Also, detect outliers using box plots, Z-score analysis, or the IQR method to identify potential data anomalies.

```
# Your code here:

from scipy import stats

# Calculate Z-scores for each numerical X variable
z_scores = numerical_X_train.apply(stats.zscore)

# Identify potential outliers (Z-score > 3 or < -3)
outliers = (z_scores > 3) | (z_scores < -3)

# Print the number of outliers for each variable
print("Number of outliers per variable:")
print(outliers.sum())
```

```
Number of outliers per variable:
gdp_per_capita                   1
social_support                   1
healthy_life_expectancy          0
freedom_to_make_life_choices     1
generosity                       1
perceptions_of_corruption        1
population                       0
population_below_poverty_line    0
hdi                              0
life_expectancy                  0
expected_years_of_schooling      0
mean_years_of_schooling          0
gni                              0
dtype: int64
```

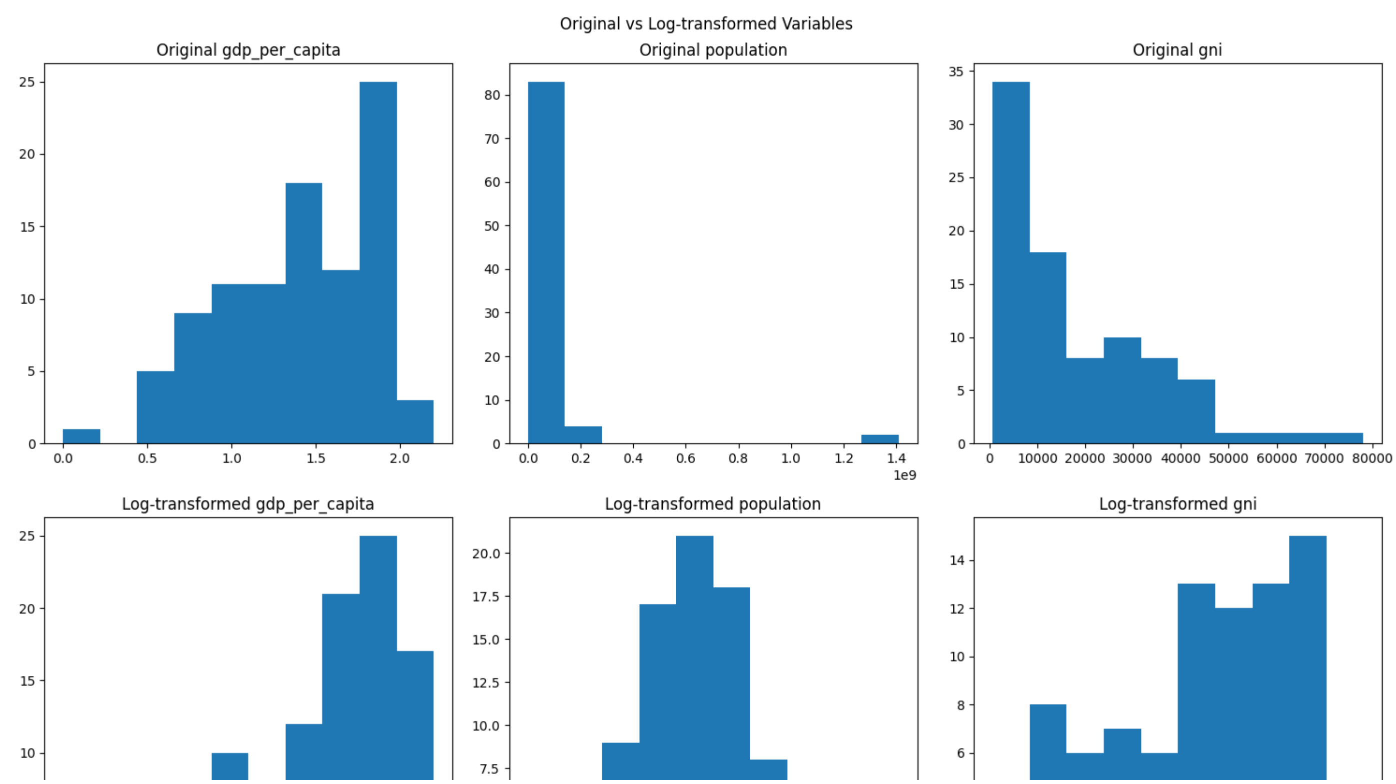
Write what you observed and your General comments on what should be done:

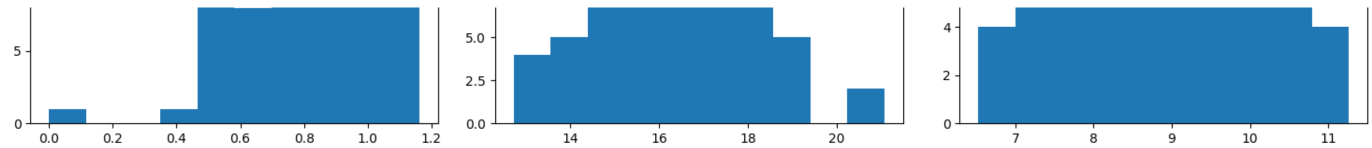
```
# Your comments here:
# Five variables show one outlier each (gdp_per_capita, social_support,
# freedom_to_make_life_choices, generosity, perceptions_of_corruption).
# Using feature engineering we can create new features that bin the data or norma
```

2. Feature Engineering

Apply log transformations to normalize skewed data and improve model stability (If any).

```
# Your code here:

columns_to_log = ['gdp_per_capita', 'population', 'gni']
for col in columns_to_log:
    X_train[f'{col}_log'] = np.log1p(X_train[col])

# Visualize change
fig, axes = plt.subplots(2, 3, figsize=(15, 10))
fig.suptitle('Original vs Log-transformed Variables')

for i, col in enumerate(columns_to_log):
    axes[0, i].hist(X_train[col])
    axes[0, i].set_title(f'Original {col}')
    axes[1, i].hist(X_train[f'{col}_log'])
    axes[1, i].set_title(f'Log-transformed {col}')

plt.tight_layout()
plt.show()
```





Create at least one interaction feature to capture relationship between existing variables, enhancing predictive power.

```
# Your code here:
X_train['freedom_healthy'] = X_train['freedom_to_make_life_choices'] * X_train['h
```

3. Preprocess data using Sklearn Column Transformer/ Write and Save Preprocessor function

```
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer, make_column_transformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler, OneHotEncoder

# Create the preprocessing pipelines for both numeric and categorical data.
numeric_features = X_train.select_dtypes(include=['float64'])
numeric_features=numeric_features.columns.tolist()

numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='median')), ## Is this good enough?
    ('scaler', StandardScaler())]) # You will need to describe why this is being 

categorical_features = ['region', 'country', 'country_name']

# Replacing missing values with Modal value and then one hot encoding.
categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='most_frequent')),
    ('onehot', OneHotEncoder(handle_unknown='ignore'))])

# Final preprocessor object set up with ColumnTransformer
preprocessor = ColumnTransformer(transformers=[('num', numeric_transformer, numer

# Fit your preprocessor object
preprocess = preprocessor.fit(X_train)
```

Describe step-by-step what we are doing above, and why? You are free to change how values are imputed. What change did you make if any, and why?

```
## Your answer :
# For numerical features, the pipeline fills missing values with the median of ea
# with imputation and standardizes the features to have zero mean and unit varian
# We changed the imputation strategy, because constant imputation with zero repla
# assuming that "missing" and "zero" mean the same thing, which may not be the ca
# It specified categorical features, fills missing values with the most frequent 
# connverts categorical variables into binary columns with one-hot encoding.
# Then the piepline combines both numeric and categorical transformers into a sin
# applies the appropriate transformer to each set of features, and fits the prepr
```

```
# Write function to transform data with preprocessor

def preprocessor(data):
    data.drop(['country', 'region'], axis=1)
    preprocessed_data=preprocess.transform(data)
    return preprocessed_data
```

What are the differences between the "preprocessor" object, the "preprocess" object, the "preprocessor" function, and the "preprocessed_data" that is returned finally?

```
## Your Answer :
# The "preprocessor" object defines the preprocessing steps.
# The "preprocess" object is the fitted version of the preprocessor.
# The "preprocessor" function is a wrapper that includes additional steps (droppi
# uses the fitted "preprocess" object. The "preprocessed_data" is the final, tran
```

```
# check shape of X data after preprocessing it using our new function
preprocessor(X_train).shape
```

```
(95, 211)
```

4. Fit model on preprocessed data and save preprocessor function and model

```
## Define a Random Forest Model here, fit it, and score it

from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

model_1 = RandomForestClassifier(random_state=42)

# Fit model
model_1.fit(preprocessor(X_train), y_train)

# Apply same log transformations and interaction variable to X_test
columns_to_log = ['gdp_per_capita', 'population', 'gni']
for col in columns_to_log:
    X_test[f'{col}_log'] = np.log1p(X_test[col])

X_test['freedom_healthy'] = X_test['freedom_to_make_life_choices'] * X_test['heal

# Score the model on training data
train_score = model_1.score(preprocessor(X_train), y_train)
print(f"Training Accuracy: {train_score:.4f}")

# Score the model on testing data
test_score = model_1.score(preprocessor(X_test), y_test)
print(f"Testing Accuracy: {test_score:.4f}")

# Your cell should have a score between 0-1 as output
```

```
Training Accuracy: 1.0000
Testing Accuracy: 0.5476
```

5. Generate predictions from X_test data and compare it with true labels in Y_test

```
#-- Generate predicted values (Model 1)
prediction_labels_1 = model_1.predict(preprocessor(X_test))

## Write code to show model performance by comparing prediction_labels with true
accuracy = accuracy_score(y_test, prediction_labels_1)
print(f"Accuracy: {accuracy:.4f}")
```

```
Accuracy: 0.5476
```

6. Repeat the process with different parameters to improve the accuracy

```
# Train model 2 using same preprocessor (note that you could save a new preproces
from sklearn.ensemble import RandomForestClassifier

## Make a new model with changed parameters to improve the score
model_2 = RandomForestClassifier(
    n_estimators=200,
    max_depth=10,
    max_features='sqrt',
    min_samples_split=5,
    min_samples_leaf=2,
    random_state=42
)
```

What changes did you make, what do the parameters you changed control, and why does it improve performance?

```
## Your answer :
# n_estimators=200 increases the number of trees to 200 to improve model stabilit
# max_depth=10 limits the depth of each tree to prevent overfitting.
# max_features='sqrt' selects a subset of features at each split (square root of 
# to increase diversity among trees and improve overall generalization.
# min_samples_split=5 requires at least 5 samples to split a node, reducing overf
# min_samples_leaf=2 ensures that leaf nodes have at least 2 samples,
# which prevents overly small splits that might capture noise.
```

```
#Evaluate Model 2:

## Write code to show model performance by comparing prediction_labels with true

# Fit Model 2
model_2.fit(preprocessor(X_train), y_train)

#-- Generate predicted y values (Model 2)
prediction_labels_2 = model_2.predict(preprocessor(X_test))

# Evaluate the model's performance using accuracy score
accuracy_2 = accuracy_score(y_test, prediction_labels_2)
print(f"Accuracy of Model 2: {accuracy_2:.4f}")
```

```
Accuracy of Model 2: 0.6190
```

Do you think it is worth making more changes to the parameters? Should we keep trying random values and see what works better? What is an alternative to doing this manually?

```
## Your answer:
# The model performs better but trying random variables manually may not be effic
# We could use systematic methods to optimize hyperparameters with GridSearchCV.
```

```
# Submit a third model using GridSearchCV

from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV
import numpy as np

# Use np.arange to create a sequence of numbers for each parameter's space you th
param_grid = {
    'n_estimators': np.arange(100, 301, 50),
    'max_depth': np.arange(5, 16, 5),
    'min_samples_split': np.arange(2, 11, 3),
    'min_samples_leaf': np.arange(1, 5),
    'max_features': ['sqrt', 'log2']
}

# Read GridSearchCV docs and create an object with RandomForestClassifier as the 
model_3 = GridSearchCV(estimator=RandomForestClassifier(random_state=42),
                         param_grid=param_grid,
                         scoring='accuracy',
                         cv=5,
                         verbose=1)

# Fit the model using GridSearchCV
model_3.fit(preprocessor(X_train), y_train)

# Extract and print the best score and parameters
print("Best mean cross-validation score: {:.4f}".format(model_3.best_score_))
print("Best parameters: {}".format(model_3.best_params_))
```

```
Fitting 5 folds for each of 360 candidates, totalling 1800 fits
Best mean cross-validation score: 0.5895
Best parameters: {'max_depth': 5, 'max_features': 'sqrt', 'min_samples_leaf':
```

```
#Submit Model 3:

#-- Generate predicted values
prediction_labels_3 = model_3.predict(preprocessor(X_test))

## Write code to show model performance by comparing prediction_labels with true
accuracy = accuracy_score(y_test, prediction_labels_3)
print(f"Accuracy: {accuracy:.4f}")
```

```
Accuracy: 0.5952
```

```
# Here are several classic ML architectures you can consider choosing from to exp
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import BaggingClassifier
from sklearn.ensemble import GradientBoostingClassifier

# Trying GradientBoostingClassifier
model_4 = GradientBoostingClassifier(n_estimators=100, learning_rate=0.1, max_dep

# Fit the model to the training data
history = model_4.fit(preprocessor(X_train), y_train)

#-- Generate predicted values
prediction_labels_4 = model_4.predict(preprocessor(X_test))

# Calculate accuracy
accuracy = accuracy_score(y_test, prediction_labels_4)
print(f"Accuracy: {accuracy:.4f}")
```

```
Accuracy: 0.5476
```

```
# Trying KNeighborsClassifier

model_5 = KNeighborsClassifier(n_neighbors=5)

# Fit the model to the training data
model_5.fit(preprocessor(X_train), y_train)

#-- Generate predicted values
prediction_labels_5 = model_5.predict(preprocessor(X_test))

# Calculate accuracy
accuracy = accuracy_score(y_test, prediction_labels_5)
print(f"Accuracy: {accuracy:.4f}")
```

```
Accuracy: 0.5714
```

```
# Trying SVC

model_6 = SVC(kernel='linear', C=0.1, random_state=42)

# Fit the model to the training data
model_6.fit(preprocessor(X_train), y_train)

#-- Generate predicted values
prediction_labels_6 = model_6.predict(preprocessor(X_test))

# Calculate accuracy
accuracy = accuracy_score(y_test, prediction_labels_6)
print(f"Accuracy: {accuracy:.4f}")
```

```
Accuracy: 0.5238
```

Describe what were the parameters you defined in GradientBoostingClassifier, and/or BaggingClassifier, and/or KNNs, and/or SVC? What worked and why?

```
## Your answer:

# With GradientBoostingClassifier, we built 100 sequential decision trees, kept e
# with maximum 3 levels deep, and addds each tree's predictions at a controlled r
# This was intended to produce consistent results across multiple runs.

# With KNeighbors Classifier, we chose k = 5 to prevent the model from being
# too sensitive to noise and losing local patterns. However it is simple and not 

# With SVC, we chose linear kernel for a simple model.
# with a small C parameter for stronger regularization strength and to prevent ov

# GridSearchCV performed the best of the 4.
#However, RandomForrestClassifier in model 2 performed better than the 4 subseque
```

7. Basic Deep Learning

```
# Now experiment with deep learning models:
import keras
from keras.models import Sequential
from keras.layers import Dense, Activation
```

```
from sklearn.preprocessing import LabelBinarizer

# Count features in input data
# The preprocessor function is likely changing the number of features
# We need to get the number of features *after* preprocessing
feature_count = preprocessor(X_train).shape[1] # Get feature count after preproce

num_classes = len(y_train.unique())

# Define a Neural Network Model with 5 layers 128->64->64->32->(?)
# Use Softmax activation in last layer. How many neurons should there be in the l
keras_model = Sequential([
    Dense(128, input_dim=feature_count, activation='relu'), # Use the correct fea
    Dense(64, activation='relu'),
    Dense(64, activation='relu'),
    Dense(32, activation='relu'),
    Dense(num_classes, activation='softmax') #Should be five classes
])

# Compile model
keras_model.compile(loss='categorical_crossentropy', optimizer='sgd', metrics=['a

# Convert y_train to one-hot encoding
lb = LabelBinarizer()
y_train_encoded = lb.fit_transform(y_train)

# Fitting the model to the Training set
history = keras_model.fit(preprocessor(X_train), y_train_encoded, ## Note that ke
               batch_size = 20,
               epochs = 300, validation_split=0.25)

# Save history for plotting later
history_dict = history.history
```

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9464 - loss: 0.2625 - val_
Epoch 203/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.9910 - loss: 0.2556 - val_
Epoch 204/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.9760 - loss: 0.2668 - val_
Epoch 205/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.9910 - loss: 0.2542 - val_
Epoch 206/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.9944 - loss: 0.2546 - val_
Epoch 207/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9860 - loss: 0.2569 - val_
Epoch 208/300
```

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 0.9944 - loss: 0.2209 - val_
Epoch 209/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.9944 - loss: 0.2188 - val_
Epoch 210/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9910 - loss: 0.2247 - val_
Epoch 211/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9760 - loss: 0.2380 - val_
Epoch 212/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.9760 - loss: 0.2259 - val_
Epoch 213/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.9910 - loss: 0.2171 - val_
Epoch 214/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9860 - loss: 0.2156 - val_
Epoch 215/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9910 - loss: 0.2210 - val_
Epoch 216/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9910 - loss: 0.2044 - val_
Epoch 217/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.9944 - loss: 0.1992 - val_
Epoch 218/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9760 - loss: 0.2238 - val_
Epoch 219/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.9760 - loss: 0.2290 - val_
Epoch 220/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.9944 - loss: 0.2010 - val_
Epoch 221/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.9860 - loss: 0.1994 - val_
Epoch 222/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9771 - loss: 0.1995 - val_
Epoch 223/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.9760 - loss: 0.2079 - val_
Epoch 224/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9944 - loss: 0.1883 - val_
Epoch 225/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9944 - loss: 0.1851 - val_
Epoch 226/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.9944 - loss: 0.1905 - val_
Epoch 227/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.9910 - loss: 0.1699 - val_
Epoch 228/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9910 - loss: 0.1630 - val_
Epoch 229/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.9944 - loss: 0.1623 - val_
Epoch 230/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 94ms/step - accuracy: 1.0000 - loss: 0.1853 - val_
Epoch 231/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 91ms/step - accuracy: 1.0000 - loss: 0.1860 - val_
```

Which activations did you use in the middle layers? Why was softmax used in the last layer?

```
## Your answer:
# We used ReLU to allow the network to learn more complex patterns and be more co
# as it simply returns the input for positive values and zero for negative values
# Softmax is appropriate for multi-class classification problems where
# probabilities are across multiple possible output classes. It ensures that:
# All output probabilities range between 0 and 1 and all sum to 1 to form a proba
# The class with the highest probability is the model's prediction
```

Was it a good idea to train for 300 epochs? Should you train a bit more? Why or why not?

```
## Your answer:
# After 150 epochs, it appears that validation loss started to increase, hinting 
# We would recommend training on less epochs.
```

Why is loss='categorical_crossentropy' and optimizer='sgd'? Would you want to change something? Why / Why not?

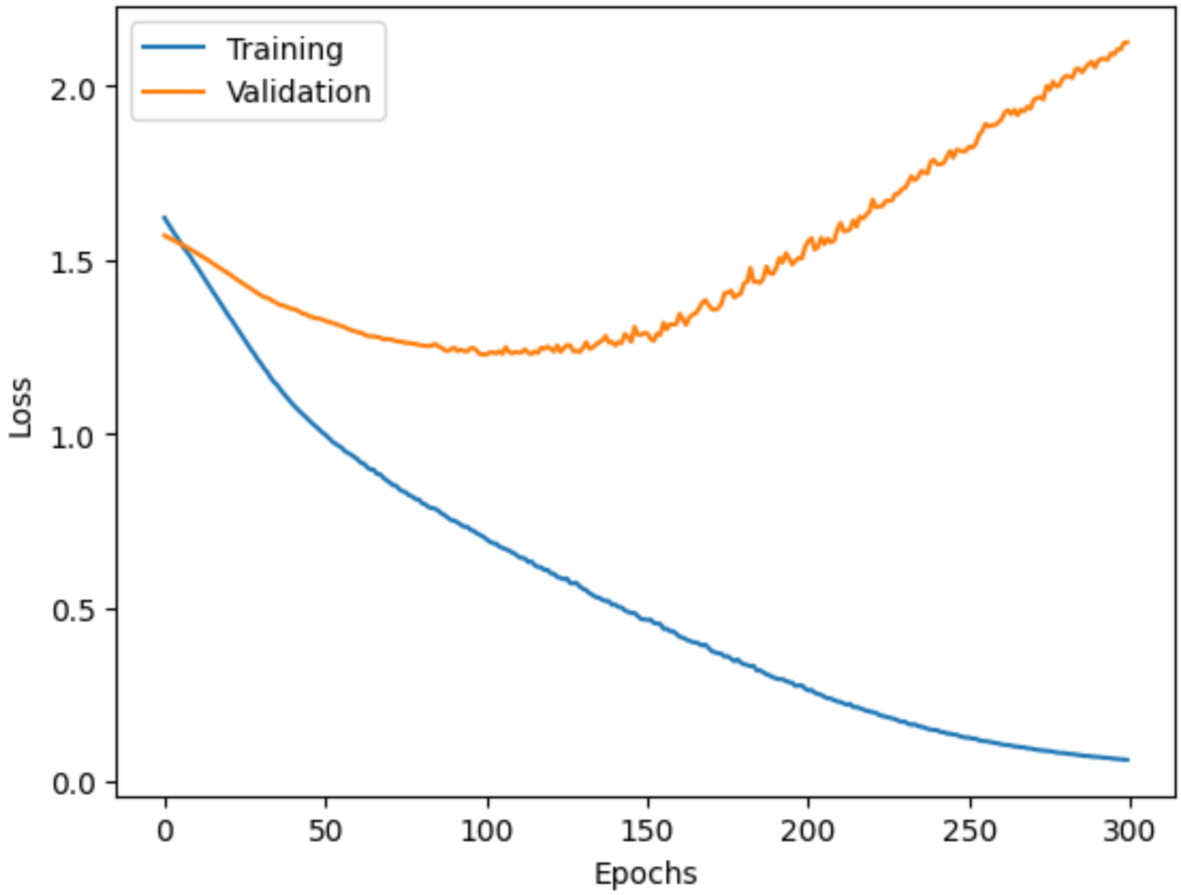
```
## Your answer:
# 'categorical_crossentropy' can be used for multi-class classification problems 
# the classes are mutually exclusive and the target variable is one-hot encoded.
# 'sgd' is appropriate for convex optimization problems but may get stuck in loca
# We could try an adaptive optimizer instead of stochastic gradient descent.
```

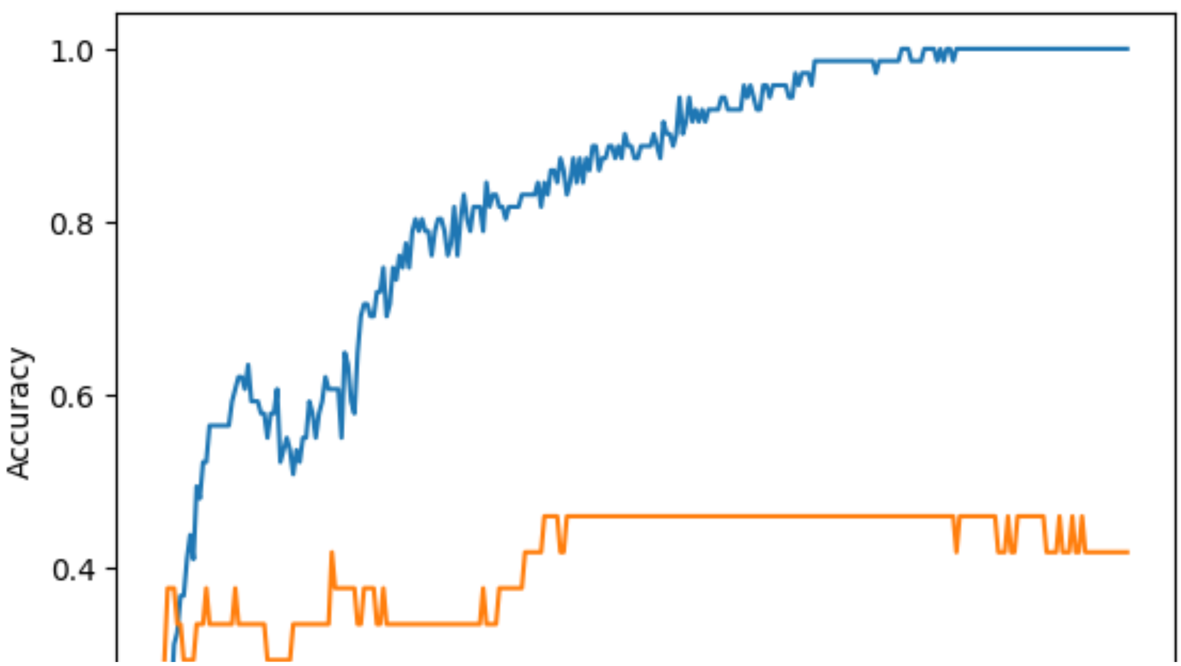
Can you try getting the model's training history out and plotting the curves?

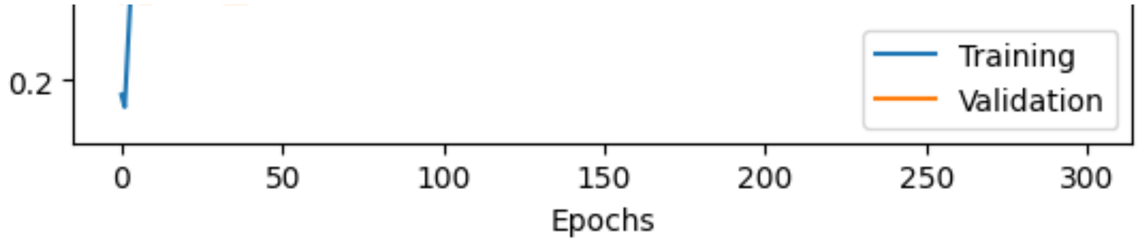
```
## Your code to plot training and validation curves in a single plot (Make change

#plot loss and accuracy at each epoch
plt.figure()
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.legend(['Training', 'Validation'])

plt.figure()
plt.xlabel('Epochs')
```

```
plt.ylabel('Accuracy')
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.legend(['Training', 'Validation'], loc='lower right')
```

```
<matplotlib.legend.Legend at 0x7c757de53f10>
```








```
#-- Generate predicted y values
y_pred = keras_model.predict(preprocessor(X_test))

# Note: Keras predict returns the predicted column index location for classificat
prediction_column_index = np.argmax(y_pred, axis=1)

# extract correct prediction labels
prediction_labels = [y_train.cat.categories[i] for i in prediction_column_index]

## Write code to show model performance by comparing prediction_labels with true
from sklearn.metrics import classification_report
print(classification_report(y_test, prediction_labels))
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 89ms/step
              precision    recall  f1-score   support

     Average       0.11      0.12      0.12         8
        High       0.12      0.12      0.12         8
         Low       0.00      0.00      0.00         8
   Very High       0.00      0.00      0.00         9
    Very Low       0.00      0.00      0.00         9

    accuracy                           0.05        42
   macro avg       0.05      0.05      0.05        42
weighted avg       0.04      0.05      0.05        42
```

Implement regularization techniques such as Dropout and Batch Normalization to improve model generalization and observe change in performance.
Note: Observe the training and testing loss and accuracy.

```
# Your code here:

from keras.layers import Dropout, BatchNormalization

regularized_model = Sequential([
```

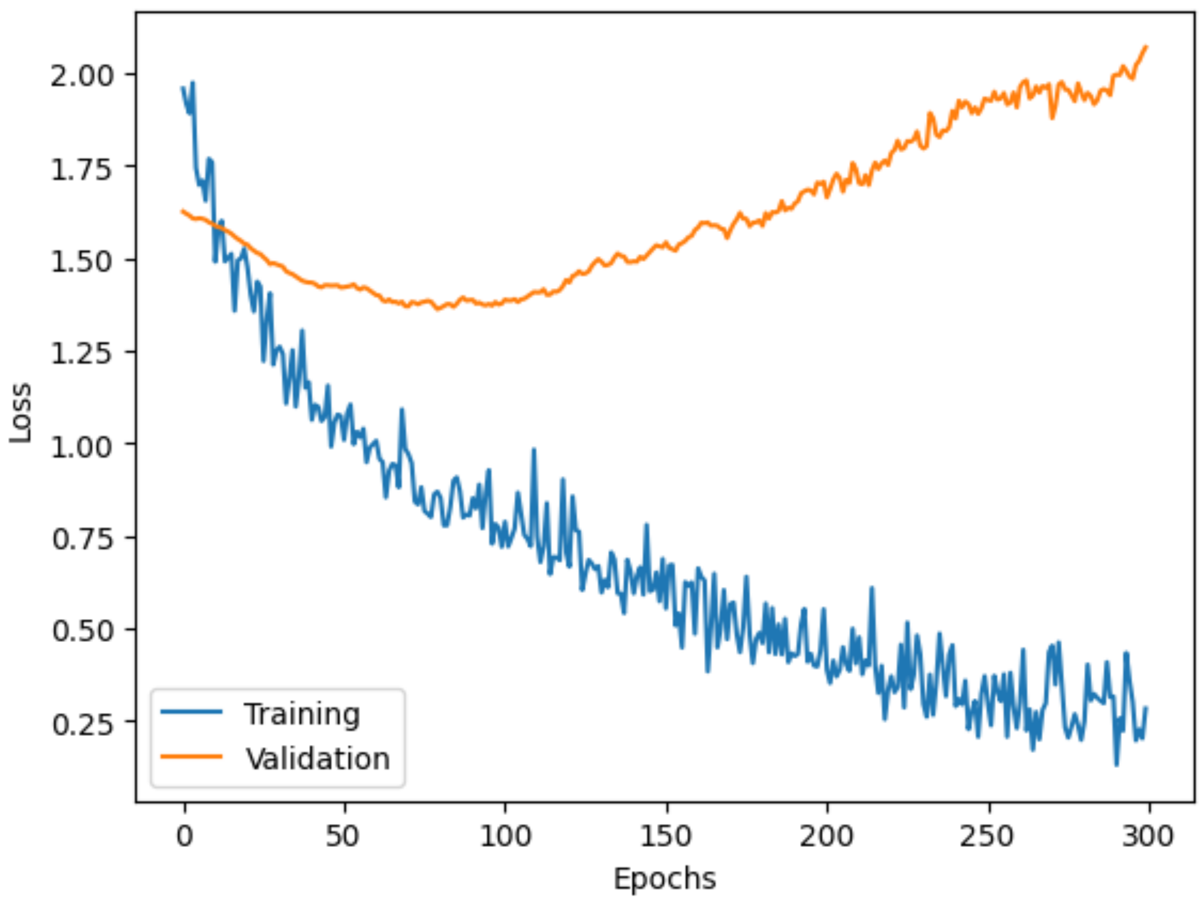
```
        # First layer
        Dense(128, input_dim=feature_count),
        BatchNormalization(),
        Activation('relu'),
        Dropout(0.3),  # 30% dropout

        # Second layer
        Dense(64),
        BatchNormalization(),
        Activation('relu'),
        Dropout(0.25),  # 25% dropout

        # Third layer
        Dense(64),
        BatchNormalization(),
        Activation('relu'),
        Dropout(0.25),  # 25% dropout

        # Fourth layer
        Dense(32),
        BatchNormalization(),
        Activation('relu'),
        Dropout(0.2),  # 20% dropout

        # Output layer
        Dense(num_classes, activation='softmax')
    ])

# Compile the model
regularized_model.compile(optimizer='sgd', loss='categorical_crossentropy', metri

# Fitting the model to the Training set
history = regularized_model.fit(preprocessor(X_train), y_train_encoded,
              batch_size = 20,
              epochs = 300, validation_split=0.25)
```

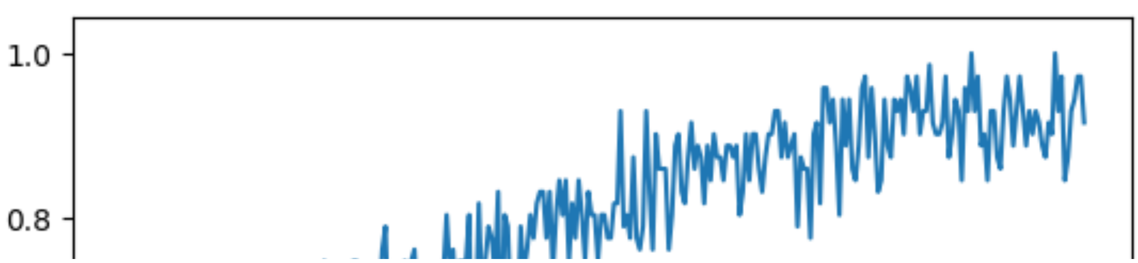
```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.8856 - loss: 0.3486 - val_
Epoch 270/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.8497 - loss: 0.4408 - val_
Epoch 271/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9085 - loss: 0.5408 - val_
Epoch 272/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9468 - loss: 0.2917 - val_
Epoch 273/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.8943 - loss: 0.3920 - val_
Epoch 274/300
```

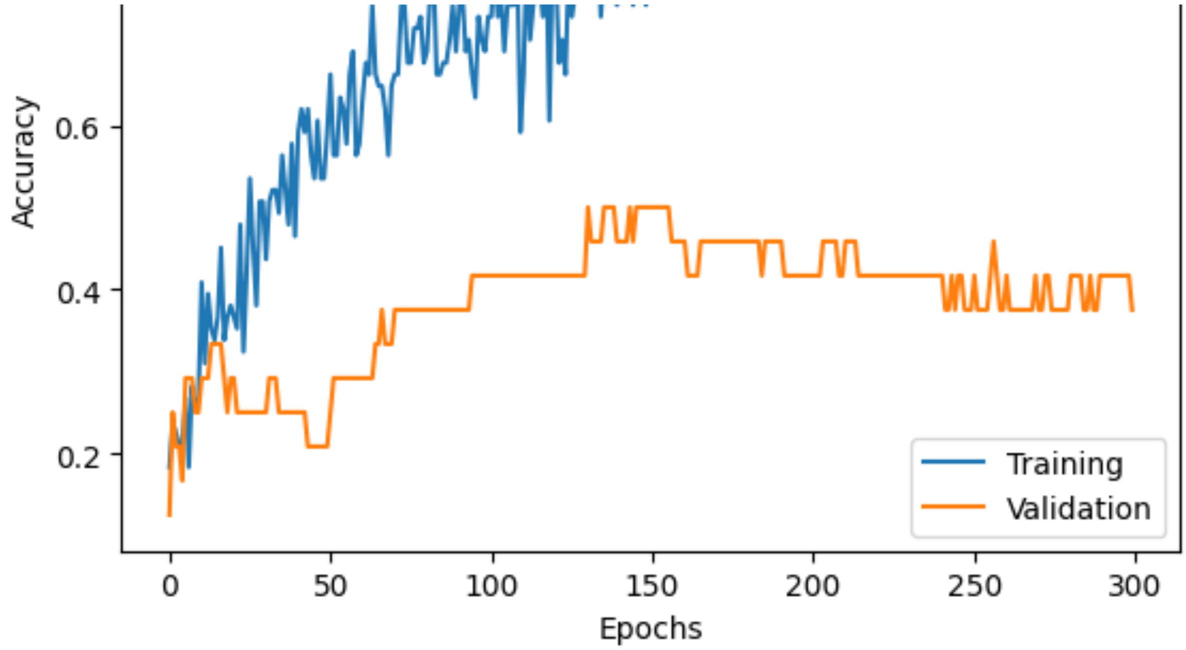
```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.8353 - loss: 0.3749 - val_
Epoch 275/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9052 - loss: 0.2838 - val_
Epoch 276/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.9821 - loss: 0.1791 - val_
Epoch 277/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9558 - loss: 0.2224 - val_
Epoch 278/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.8783 - loss: 0.2777 - val_
Epoch 279/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 94ms/step - accuracy: 0.9435 - loss: 0.2303 - val_
Epoch 280/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 93ms/step - accuracy: 0.9854 - loss: 0.1924 - val_
Epoch 281/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 68ms/step - accuracy: 0.9235 - loss: 0.2650 - val_
Epoch 282/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.8666 - loss: 0.4219 - val_
Epoch 283/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 98ms/step - accuracy: 0.9468 - loss: 0.2559 - val_
Epoch 284/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 94ms/step - accuracy: 0.8906 - loss: 0.3445 - val_
Epoch 285/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.9602 - loss: 0.2622 - val_
Epoch 286/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 74ms/step - accuracy: 0.9162 - loss: 0.3016 - val_
Epoch 287/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 93ms/step - accuracy: 0.8883 - loss: 0.2972 - val_
Epoch 288/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 57ms/step - accuracy: 0.9160 - loss: 0.3157 - val_
Epoch 289/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.9145 - loss: 0.3526 - val_
Epoch 290/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9256 - loss: 0.3020 - val_
Epoch 291/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.1167 - val_
Epoch 292/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.9235 - loss: 0.2393 - val_
Epoch 293/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.9721 - loss: 0.2221 - val_
Epoch 294/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.8664 - loss: 0.3510 - val_
Epoch 295/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9226 - loss: 0.3025 - val_
Epoch 296/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9252 - loss: 0.3077 - val_
Epoch 297/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.9658 - loss: 0.1583 - val_
Epoch 298/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9704 - loss: 0.2290 - val_
```

```
# Plot loss and accuracy at each epoch
plt.figure()
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.legend(['Training', 'Validation'])

plt.figure()
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.legend(['Training', 'Validation'], loc='lower right')
```

```
<matplotlib.legend.Legend at 0x7c750e341f90>
```








```
# Your comments about the change in performance:
# The regularized model shows lower training accuracy.
# The regularized model also should show more consistent improvement in validatio
# The original model's validation dipped initially but then starts increasing (U-
# The regularized model's validation loss remains more consistent.
```

Experiment with different activation functions (ReLU, LeakyReLU, Tanh, Sigmoid) to observe their impact on model performance.

```
# Your code here:

# LeakyReLU activation
from keras.layers import LeakyReLU

leaky_relu = Sequential([
        Dense(128, input_dim=feature_count),
        LeakyReLU(alpha=0.1),
        Dense(64),
        LeakyReLU(alpha=0.1),
        Dense(64),
        LeakyReLU(alpha=0.1),
        Dense(32),
        LeakyReLU(alpha=0.1),
        Dense(num_classes, activation='softmax')
    ])
```

```
# Compile the model
leaky_relu.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['ac

# Fitting the model to the Training set
history = leaky_relu.fit(preprocessor(X_train), y_train_encoded,
               batch_size = 20,
               epochs = 300, validation_split=0.25)

#plot loss and accuracy at each epoch
plt.figure()
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.legend(['Training', 'Validation'])

plt.figure()
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.legend(['Training', 'Validation'], loc='lower right')
```

```
Epoch 1/300
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87: Use
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
/usr/local/lib/python3.11/dist-packages/keras/src/layers/activations/leaky_rel
  warnings.warn(
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 140ms/step - accuracy: 0.4050 - loss: 1.5635 - val
Epoch 2/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 0.4267 - loss: 1.5448 - val_
Epoch 3/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.4532 - loss: 1.5208 - val_
Epoch 4/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.4609 - loss: 1.5101 - val_
Epoch 5/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.5065 - loss: 1.4829 - val_
Epoch 6/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.4936 - loss: 1.4443 - val_
Epoch 7/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.4653 - loss: 1.4234 - val_
Epoch 8/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.4553 - loss: 1.4186 - val_
Epoch 9/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.4269 - loss: 1.4324 - val_
Epoch 10/300
```

```
Epoch 10/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 0.4503 - loss: 1.4057 - val_
Epoch 11/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.4319 - loss: 1.3794 - val_
Epoch 12/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.4480 - loss: 1.3578 - val_
Epoch 13/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.4826 - loss: 1.3217 - val_
Epoch 14/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.5326 - loss: 1.3163 - val_
Epoch 15/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 0.4653 - loss: 1.3147 - val_
Epoch 16/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.4749 - loss: 1.2837 - val_
Epoch 17/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.4349 - loss: 1.3055 - val_
Epoch 18/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.4809 - loss: 1.2728 - val_
Epoch 19/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.4642 - loss: 1.2389 - val_
Epoch 20/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 0.5009 - loss: 1.1720 - val_
Epoch 21/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.5172 - loss: 1.1909 - val_
Epoch 22/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.4399 - loss: 1.2279 - val_
Epoch 23/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.4549 - loss: 1.1981 - val_
Epoch 24/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 86ms/step - accuracy: 0.4538 - loss: 1.1765 - val_
Epoch 25/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 101ms/step - accuracy: 0.5422 - loss: 1.1080 - val
Epoch 26/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 66ms/step - accuracy: 0.5241 - loss: 1.1722 - val_
Epoch 27/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 90ms/step - accuracy: 0.5770 - loss: 1.0939 - val_
Epoch 28/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 73ms/step - accuracy: 0.6016 - loss: 1.0814 - val_
Epoch 29/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 99ms/step - accuracy: 0.5820 - loss: 1.1093 - val_
Epoch 30/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.5833 - loss: 1.0806 - val_
Epoch 31/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step - accuracy: 0.5887 - loss: 1.0493 - val_
Epoch 32/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.5960 - loss: 1.0823 - val_
Epoch 33/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.5737 - loss: 1.0671 - val_
Epoch 34/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.5987 - loss: 1.0296 - val_
Epoch 35/300
```

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.5831 - loss: 1.0029 - val_
Epoch 36/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.5847 - loss: 1.0115 - val_
Epoch 37/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.5458 - loss: 1.0244 - val_
Epoch 38/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 0.5364 - loss: 1.0677 - val_
Epoch 39/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 82ms/step - accuracy: 0.5814 - loss: 0.9986 - val_
Epoch 40/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.5777 - loss: 1.0264 - val
Epoch 41/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 99ms/step - accuracy: 0.5931 - loss: 0.9824 - val_
Epoch 42/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.5727 - loss: 0.9786 - val_
Epoch 43/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 110ms/step - accuracy: 0.5750 - loss: 0.9447 - val
Epoch 44/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 96ms/step - accuracy: 0.6031 - loss: 0.9213 - val_
Epoch 45/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 72ms/step - accuracy: 0.5260 - loss: 1.0332 - val_
Epoch 46/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.5843 - loss: 0.9476 - val_
Epoch 47/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.5504 - loss: 0.9851 - val_
Epoch 48/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.5727 - loss: 0.9573 - val_
Epoch 49/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.6093 - loss: 0.9394 - val_
Epoch 50/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.5860 - loss: 0.9167 - val_
Epoch 51/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.5477 - loss: 0.9538 - val_
Epoch 52/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.6093 - loss: 0.8928 - val_
Epoch 53/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 0.5800 - loss: 0.9450 - val_
Epoch 54/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.5683 - loss: 0.9296 - val_
Epoch 55/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.6112 - loss: 0.8838 - val_
Epoch 56/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.5983 - loss: 0.9194 - val_
Epoch 57/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.6539 - loss: 0.8680 - val_
Epoch 58/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.5639 - loss: 0.9176 - val_
Epoch 59/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.6679 - loss: 0.8650 - val_
Epoch 60/300
```

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.5823 - loss: 0.8919 - val_
Epoch 61/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.5889 - loss: 0.9123 - val_
Epoch 62/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.6162 - loss: 0.8877 - val_
Epoch 63/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 0.6375 - loss: 0.9179 - val_
Epoch 64/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.5562 - loss: 0.9388 - val_
Epoch 65/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.6319 - loss: 0.8789 - val_
Epoch 66/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.6219 - loss: 0.8817 - val_
Epoch 67/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.6375 - loss: 0.8796 - val_
Epoch 68/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 79ms/step - accuracy: 0.6858 - loss: 0.8187 - val_
Epoch 69/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 91ms/step - accuracy: 0.6471 - loss: 0.8665 - val_
Epoch 70/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 99ms/step - accuracy: 0.6892 - loss: 0.8211 - val_
Epoch 71/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 229ms/step - accuracy: 0.6802 - loss: 0.8099 - val
Epoch 72/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 187ms/step - accuracy: 0.6577 - loss: 0.8582 - val
Epoch 73/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 193ms/step - accuracy: 0.7094 - loss: 0.8011 - val
Epoch 74/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 79ms/step - accuracy: 0.6731 - loss: 0.7799 - val_
Epoch 75/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 91ms/step - accuracy: 0.7144 - loss: 0.7892 - val_
Epoch 76/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7050 - loss: 0.8214 - val_
Epoch 77/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.6840 - loss: 0.8070 - val_
Epoch 78/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7117 - loss: 0.8042 - val_
Epoch 79/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.7361 - loss: 0.7648 - val_
Epoch 80/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.6748 - loss: 0.7889 - val_
Epoch 81/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.7467 - loss: 0.7735 - val_
Epoch 82/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.6867 - loss: 0.8043 - val_
Epoch 83/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7077 - loss: 0.7749 - val_
Epoch 84/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.7067 - loss: 0.8110 - val_
Epoch 85/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.6830 - loss: 0.8206 - val_
```

```
Epoch 86/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.7263 - loss: 0.7789 - val_
Epoch 87/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.7130 - loss: 0.7783 - val_
Epoch 88/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.7046 - loss: 0.7732 - val_
Epoch 89/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7640 - loss: 0.7481 - val_
Epoch 90/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7286 - loss: 0.7743 - val_
Epoch 91/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.6907 - loss: 0.7671 - val_
Epoch 92/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7123 - loss: 0.7452 - val_
Epoch 93/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.7063 - loss: 0.7549 - val_
Epoch 94/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.6773 - loss: 0.7628 - val_
Epoch 95/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.7207 - loss: 0.7151 - val_
Epoch 96/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7523 - loss: 0.7311 - val_
Epoch 97/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.7830 - loss: 0.7002 - val_
Epoch 98/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.7749 - loss: 0.7341 - val_
Epoch 99/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.6913 - loss: 0.7766 - val_
Epoch 100/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7413 - loss: 0.6930 - val_
Epoch 101/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 0.7530 - loss: 0.6882 - val_
Epoch 102/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 0.7615 - loss: 0.7227 - val_
Epoch 103/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.7436 - loss: 0.7029 - val_
Epoch 104/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.7838 - loss: 0.7318 - val_
Epoch 105/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.7645 - loss: 0.7057 - val_
Epoch 106/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.7292 - loss: 0.7093 - val_
Epoch 107/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.7432 - loss: 0.6957 - val_
Epoch 108/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.7945 - loss: 0.6895 - val_
Epoch 109/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.7682 - loss: 0.7070 - val_
Epoch 110/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.8472 - loss: 0.6254 - val_
```

```
Epoch 111/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.7961 - loss: 0.6815 - val_
Epoch 112/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.7832 - loss: 0.6575 - val_
Epoch 113/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.7715 - loss: 0.6640 - val_
Epoch 114/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.8038 - loss: 0.6359 - val_
Epoch 115/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.7999 - loss: 0.6544 - val_
Epoch 116/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7828 - loss: 0.6604 - val_
Epoch 117/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.8001 - loss: 0.6589 - val_
Epoch 118/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 80ms/step - accuracy: 0.8311 - loss: 0.5912 - val_
Epoch 119/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 66ms/step - accuracy: 0.8434 - loss: 0.5899 - val_
Epoch 120/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 64ms/step - accuracy: 0.7722 - loss: 0.6405 - val_
Epoch 121/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 95ms/step - accuracy: 0.8061 - loss: 0.6280 - val_
Epoch 122/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 65ms/step - accuracy: 0.7761 - loss: 0.6481 - val_
Epoch 123/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 94ms/step - accuracy: 0.8351 - loss: 0.5982 - val_
Epoch 124/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 96ms/step - accuracy: 0.8161 - loss: 0.5780 - val_
Epoch 125/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 97ms/step - accuracy: 0.8011 - loss: 0.6415 - val_
Epoch 126/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 96ms/step - accuracy: 0.7968 - loss: 0.6214 - val_
Epoch 127/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.8430 - loss: 0.5762 - val_
Epoch 128/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.8557 - loss: 0.5696 - val_
Epoch 129/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.8207 - loss: 0.6091 - val_
Epoch 130/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.8430 - loss: 0.5561 - val_
Epoch 131/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.8130 - loss: 0.5887 - val_
Epoch 132/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.8747 - loss: 0.5583 - val_
Epoch 133/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.8553 - loss: 0.5787 - val_
Epoch 134/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.8364 - loss: 0.5463 - val_
Epoch 135/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.8530 - loss: 0.5308 - val_
Epoch 136/300
```

```
Epoch 136/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.8824 - loss: 0.5429 - val_
Epoch 137/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.8680 - loss: 0.5252 - val_
Epoch 138/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.8987 - loss: 0.5277 - val_
Epoch 139/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.8910 - loss: 0.5340 - val_
Epoch 140/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.8264 - loss: 0.5429 - val_
Epoch 141/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.8660 - loss: 0.5177 - val_
Epoch 142/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.8993 - loss: 0.5152 - val_
Epoch 143/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.8876 - loss: 0.5025 - val_
Epoch 144/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.8826 - loss: 0.5099 - val_
Epoch 145/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.8633 - loss: 0.4989 - val_
Epoch 146/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.8716 - loss: 0.5013 - val_
Epoch 147/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.8243 - loss: 0.5211 - val_
Epoch 148/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9033 - loss: 0.4762 - val_
Epoch 149/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.9049 - loss: 0.4804 - val_
Epoch 150/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.9083 - loss: 0.4958 - val_
Epoch 151/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.9212 - loss: 0.4836 - val_
Epoch 152/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9066 - loss: 0.4590 - val_
Epoch 153/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 0.9083 - loss: 0.4834 - val_
Epoch 154/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9195 - loss: 0.4950 - val_
Epoch 155/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.9012 - loss: 0.4778 - val_
Epoch 156/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.9045 - loss: 0.4736 - val_
Epoch 157/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9345 - loss: 0.4464 - val_
Epoch 158/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.8995 - loss: 0.4659 - val_
Epoch 159/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9162 - loss: 0.4470 - val_
Epoch 160/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9229 - loss: 0.4552 - val_
Epoch 161/300
```

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9339 - loss: 0.4090 - val_
Epoch 162/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9485 - loss: 0.4070 - val_
Epoch 163/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.9279 - loss: 0.4166 - val_
Epoch 164/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.8962 - loss: 0.4431 - val_
Epoch 165/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9468 - loss: 0.4258 - val_
Epoch 166/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9241 - loss: 0.4103 - val_
Epoch 167/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9112 - loss: 0.4072 - val_
Epoch 168/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.9385 - loss: 0.4143 - val_
Epoch 169/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 93ms/step - accuracy: 0.9435 - loss: 0.3827 - val_
Epoch 170/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 94ms/step - accuracy: 0.9475 - loss: 0.3784 - val_
Epoch 171/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 93ms/step - accuracy: 0.9441 - loss: 0.3627 - val_
Epoch 172/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 93ms/step - accuracy: 0.9235 - loss: 0.4027 - val_
Epoch 173/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 80ms/step - accuracy: 0.9341 - loss: 0.3936 - val_
Epoch 174/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 97ms/step - accuracy: 0.9441 - loss: 0.3834 - val_
Epoch 175/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9508 - loss: 0.3690 - val_
Epoch 176/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.9391 - loss: 0.3624 - val_
Epoch 177/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9475 - loss: 0.3467 - val_
Epoch 178/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9375 - loss: 0.3834 - val_
Epoch 179/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9475 - loss: 0.3552 - val_
Epoch 180/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.9475 - loss: 0.3529 - val_
Epoch 181/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.9325 - loss: 0.3502 - val_
Epoch 182/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9475 - loss: 0.3446 - val_
Epoch 183/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9631 - loss: 0.3360 - val_
Epoch 184/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.9621 - loss: 0.3406 - val_
Epoch 185/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.9887 - loss: 0.3179 - val_
Epoch 186/300
```

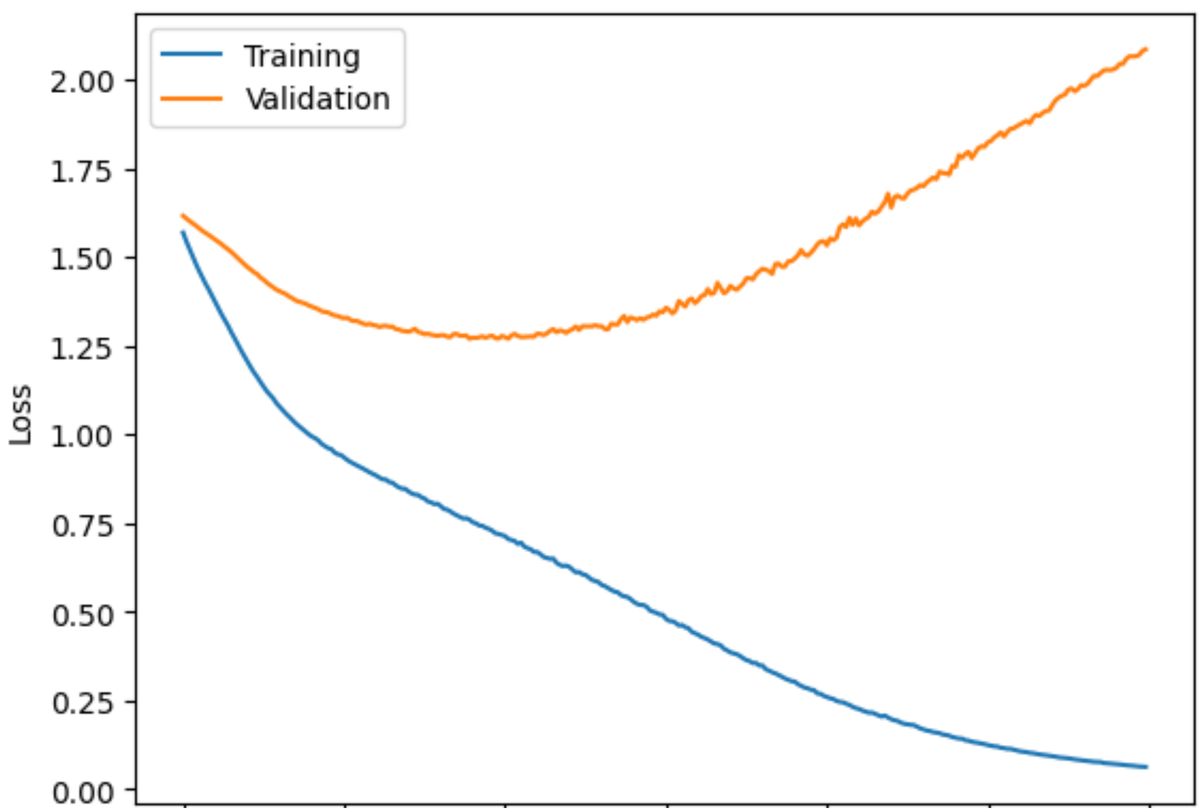
```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.9771 - loss: 0.3423 - val_
Epoch 187/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9760 - loss: 0.2992 - val_
Epoch 188/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9804 - loss: 0.3182 - val_
Epoch 189/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.9721 - loss: 0.2970 - val_
Epoch 190/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.9854 - loss: 0.3086 - val_
Epoch 191/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.9804 - loss: 0.2871 - val_
Epoch 192/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 1.0000 - loss: 0.3060 - val_
Epoch 193/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9854 - loss: 0.3055 - val_
Epoch 194/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.9944 - loss: 0.2814 - val_
Epoch 195/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9821 - loss: 0.2788 - val_
Epoch 196/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.9944 - loss: 0.2668 - val_
Epoch 197/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9944 - loss: 0.2819 - val_
Epoch 198/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9944 - loss: 0.2616 - val_
Epoch 199/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.2524 - val_
Epoch 200/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.9910 - loss: 0.2341 - val_
Epoch 201/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 1.0000 - loss: 0.2427 - val_
Epoch 202/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9944 - loss: 0.2850 - val_
Epoch 203/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.2668 - val_
Epoch 204/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9860 - loss: 0.2193 - val_
Epoch 205/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.2534 - val_
Epoch 206/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9760 - loss: 0.2416 - val_
Epoch 207/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.9944 - loss: 0.2405 - val_
Epoch 208/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 1.0000 - loss: 0.2271 - val_
Epoch 209/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 1.0000 - loss: 0.2356 - val_
Epoch 210/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 1.0000 - loss: 0.2351 - val_
Epoch 211/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 1.0000 - loss: 0.2274 - val_
```

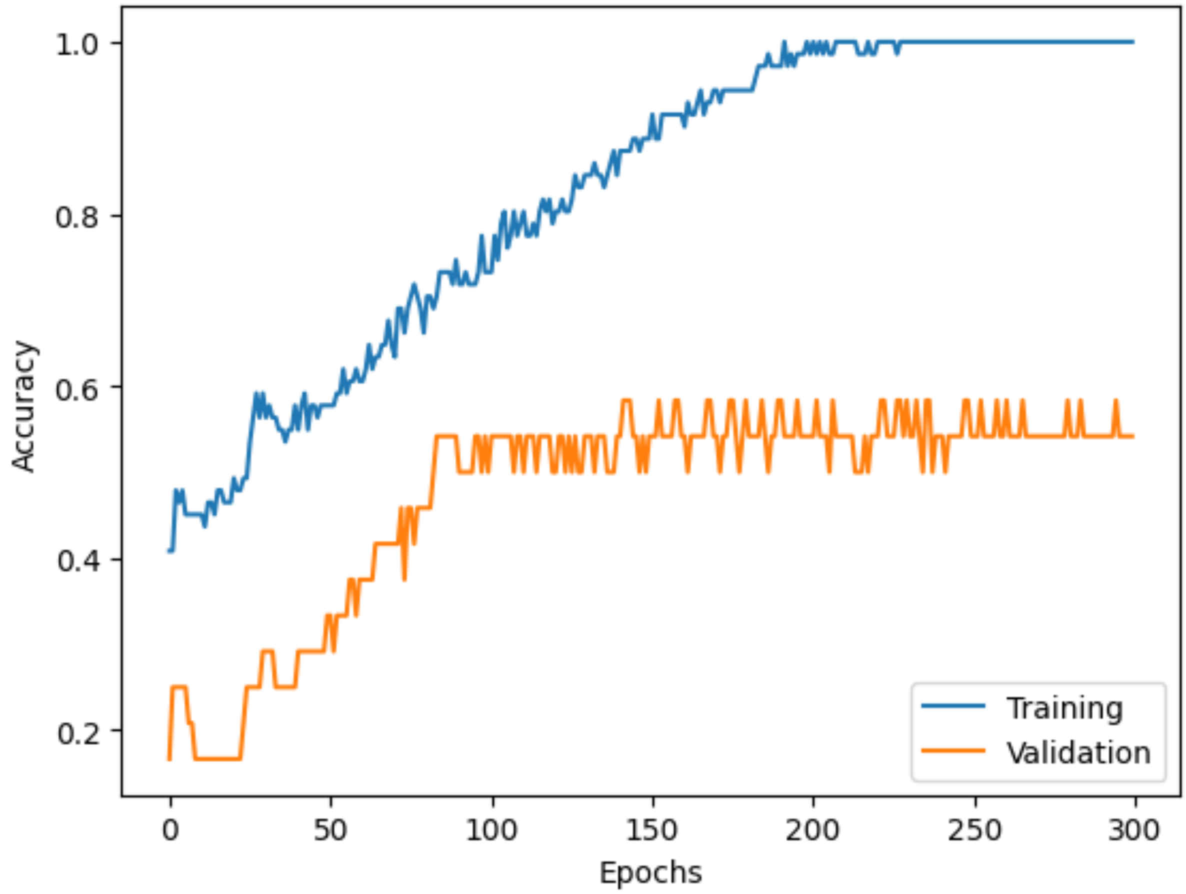
```
Epoch 212/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.1983 - val_
Epoch 213/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.2205 - val_
Epoch 214/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 1.0000 - loss: 0.2217 - val_
Epoch 215/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 68ms/step - accuracy: 0.9760 - loss: 0.2158 - val_
Epoch 216/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.9760 - loss: 0.2212 - val_
Epoch 217/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 91ms/step - accuracy: 0.9910 - loss: 0.1979 - val_
Epoch 218/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 66ms/step - accuracy: 1.0000 - loss: 0.1928 - val_
Epoch 219/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 96ms/step - accuracy: 0.9860 - loss: 0.2079 - val_
Epoch 220/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 102ms/step - accuracy: 0.9910 - loss: 0.2072 - val
Epoch 221/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 1.0000 - loss: 0.1880 - val_
Epoch 222/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 92ms/step - accuracy: 1.0000 - loss: 0.1802 - val_
Epoch 223/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 1.0000 - loss: 0.1952 - val_
Epoch 224/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 87ms/step - accuracy: 1.0000 - loss: 0.1921 - val_
Epoch 225/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 96ms/step - accuracy: 1.0000 - loss: 0.1649 - val_
Epoch 226/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 70ms/step - accuracy: 1.0000 - loss: 0.1801 - val_
Epoch 227/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 91ms/step - accuracy: 0.9944 - loss: 0.1565 - val_
Epoch 228/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 94ms/step - accuracy: 1.0000 - loss: 0.1815 - val_
Epoch 229/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 98ms/step - accuracy: 1.0000 - loss: 0.1806 - val_
Epoch 230/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 98ms/step - accuracy: 1.0000 - loss: 0.1696 - val_
Epoch 231/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 1.0000 - loss: 0.1615 - val_
Epoch 232/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 1.0000 - loss: 0.1642 - val_
Epoch 233/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.1492 - val_
Epoch 234/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 1.0000 - loss: 0.1618 - val_
Epoch 235/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.1571 - val_
Epoch 236/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.1607 - val_
```

```
Epoch 237/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 1.0000 - loss: 0.1639 - val_
Epoch 238/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 1.0000 - loss: 0.1631 - val_
Epoch 239/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 1.0000 - loss: 0.1393 - val_
Epoch 240/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 1.0000 - loss: 0.1397 - val_
Epoch 241/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.1293 - val_
Epoch 242/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 1.0000 - loss: 0.1470 - val_
Epoch 243/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 1.0000 - loss: 0.1468 - val_
Epoch 244/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.1388 - val_
Epoch 245/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 1.0000 - loss: 0.1418 - val_
Epoch 246/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.1190 - val_
Epoch 247/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.1312 - val_
Epoch 248/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 1.0000 - loss: 0.1327 - val_
Epoch 249/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.1416 - val_
Epoch 250/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 1.0000 - loss: 0.1416 - val_
Epoch 251/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 1.0000 - loss: 0.1192 - val_
Epoch 252/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 1.0000 - loss: 0.1211 - val_
Epoch 253/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 1.0000 - loss: 0.1229 - val_
Epoch 254/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 1.0000 - loss: 0.1206 - val_
Epoch 255/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 1.0000 - loss: 0.1309 - val_
Epoch 256/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.1032 - val_
Epoch 257/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step - accuracy: 1.0000 - loss: 0.1032 - val_
Epoch 258/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.1087 - val_
Epoch 259/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 70ms/step - accuracy: 1.0000 - loss: 0.1085 - val_
Epoch 260/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 68ms/step - accuracy: 1.0000 - loss: 0.1036 - val_
Epoch 261/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 90ms/step - accuracy: 1.0000 - loss: 0.1146 - val_
Epoch 262/300
```

```
Epoch 262/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 70ms/step - accuracy: 1.0000 - loss: 0.1094 - val_
Epoch 263/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 92ms/step - accuracy: 1.0000 - loss: 0.1094 - val_
Epoch 264/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 92ms/step - accuracy: 1.0000 - loss: 0.1119 - val_
Epoch 265/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 99ms/step - accuracy: 1.0000 - loss: 0.1021 - val_
Epoch 266/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 97ms/step - accuracy: 1.0000 - loss: 0.1027 - val_
Epoch 267/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step - accuracy: 1.0000 - loss: 0.1035 - val_
Epoch 268/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.0943 - val_
Epoch 269/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.0896 - val_
Epoch 270/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.0955 - val_
Epoch 271/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 1.0000 - loss: 0.0893 - val_
Epoch 272/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.0880 - val_
Epoch 273/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 1.0000 - loss: 0.0930 - val_
Epoch 274/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.0820 - val_
Epoch 275/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.0847 - val_
Epoch 276/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 1.0000 - loss: 0.0821 - val_
Epoch 277/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.0845 - val_
Epoch 278/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.0851 - val_
Epoch 279/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.0798 - val_
Epoch 280/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 1.0000 - loss: 0.0817 - val_
Epoch 281/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.0715 - val_
Epoch 282/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 1.0000 - loss: 0.0735 - val_
Epoch 283/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.0777 - val_
Epoch 284/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 1.0000 - loss: 0.0793 - val_
Epoch 285/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.0735 - val_
Epoch 286/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 1.0000 - loss: 0.0709 - val_
Epoch 287/300
```

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.0819 - val_
Epoch 288/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 1.0000 - loss: 0.0752 - val_
Epoch 289/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.0737 - val_
Epoch 290/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 1.0000 - loss: 0.0683 - val_
Epoch 291/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.0685 - val_
Epoch 292/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 1.0000 - loss: 0.0653 - val_
Epoch 293/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.0600 - val_
Epoch 294/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.0646 - val_
Epoch 295/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.0688 - val_
Epoch 296/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 1.0000 - loss: 0.0694 - val_
Epoch 297/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.0620 - val_
Epoch 298/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.0598 - val_
Epoch 299/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.0629 - val_
Epoch 300/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 1.0000 - loss: 0.0607 - val_
<matplotlib.legend.Legend at 0x7c750fa364d0>
```






```
# Tanh activation
tanh_model = Sequential([
        Dense(128, input_dim=feature_count, activation='tanh'),
        Dense(64, activation='tanh'),
        Dense(64, activation='tanh'),
        Dense(32, activation='tanh'),
        Dense(num_classes, activation='softmax')
    ])

# Compile the model
tanh_model.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=['ac

# Fitting the model to the Training set
history = tanh_model.fit(preprocessor(X_train), y_train_encoded,
               batch_size = 20,
               epochs = 300, validation_split=0.25)
```

```
#plot loss and accuracy at each epoch
plt.figure()
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.legend(['Training', 'Validation'])

plt.figure()
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.legend(['Training', 'Validation'], loc='lower right')
```

```
Epoch 1/300
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87: User
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 139ms/step - accuracy: 0.3961 - loss: 1.4355 - val_
Epoch 2/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 88ms/step - accuracy: 0.5115 - loss: 1.2801 - val_a
Epoch 3/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 65ms/step - accuracy: 0.5328 - loss: 1.1920 - val_a
Epoch 4/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 250ms/step - accuracy: 0.6214 - loss: 1.1128 - val_
Epoch 5/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 95ms/step - accuracy: 0.4655 - loss: 1.1481 - val_a
Epoch 6/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 101ms/step - accuracy: 0.5504 - loss: 1.0486 - val_
Epoch 7/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 97ms/step - accuracy: 0.5537 - loss: 1.0494 - val_a
Epoch 8/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 106ms/step - accuracy: 0.5677 - loss: 1.0305 - val_
Epoch 9/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 66ms/step - accuracy: 0.5860 - loss: 1.0106 - val_a
Epoch 10/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.5543 - loss: 0.9929 - val_a
Epoch 11/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.6575 - loss: 0.9527 - val_a
Epoch 12/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.6152 - loss: 0.9782 - val_a
Epoch 13/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 0.6631 - loss: 0.9133 - val_a
Epoch 14/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.6665 - loss: 0.9656 - val_a
Epoch 15/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.6392 - loss: 0.8983 - val_a
Epoch 16/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.6617 - loss: 0.9189 - val_a
```

```
Epoch 17/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.6690 - loss: 0.8761 - val_a
Epoch 18/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.6663 - loss: 0.9026 - val_a
Epoch 19/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 0.7659 - loss: 0.8487 - val_a
Epoch 20/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.7286 - loss: 0.8688 - val_a
Epoch 21/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.7859 - loss: 0.8030 - val_a
Epoch 22/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 0.7376 - loss: 0.8393 - val_a
Epoch 23/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.7569 - loss: 0.8051 - val_a
Epoch 24/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.7832 - loss: 0.8262 - val_a
Epoch 25/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 79ms/step - accuracy: 0.8059 - loss: 0.8025 - val_a
Epoch 26/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 82ms/step - accuracy: 0.7432 - loss: 0.8303 - val_a
Epoch 27/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.7676 - loss: 0.7839 - val_a
Epoch 28/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 107ms/step - accuracy: 0.8072 - loss: 0.7629 - val_
Epoch 29/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 107ms/step - accuracy: 0.8422 - loss: 0.7210 - val_
Epoch 30/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 87ms/step - accuracy: 0.8018 - loss: 0.7684 - val_a
Epoch 31/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 81ms/step - accuracy: 0.8284 - loss: 0.7364 - val_a
Epoch 32/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.8157 - loss: 0.7312 - val_a
Epoch 33/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.8434 - loss: 0.7514 - val_a
Epoch 34/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.8264 - loss: 0.7305 - val_a
Epoch 35/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.8541 - loss: 0.6523 - val_a
Epoch 36/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.8437 - loss: 0.7306 - val_a
Epoch 37/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.8274 - loss: 0.7051 - val_a
Epoch 38/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.8503 - loss: 0.6651 - val_a
Epoch 39/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.8687 - loss: 0.6677 - val_a
Epoch 40/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.8866 - loss: 0.6922 - val_a
Epoch 41/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step - accuracy: 0.8443 - loss: 0.6643 - val_a
```

```
Epoch 42/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.8470 - loss: 0.6553 - val_a
Epoch 43/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.8437 - loss: 0.6716 - val_a
Epoch 44/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 96ms/step - accuracy: 0.8737 - loss: 0.6065 - val_a
Epoch 45/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 95ms/step - accuracy: 0.9139 - loss: 0.6250 - val_a
Epoch 46/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 95ms/step - accuracy: 0.8939 - loss: 0.6365 - val_a
Epoch 47/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 82ms/step - accuracy: 0.8899 - loss: 0.6174 - val_a
Epoch 48/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 101ms/step - accuracy: 0.8906 - loss: 0.6664 - val_
Epoch 49/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.8899 - loss: 0.6124 - val_a
Epoch 50/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.8956 - loss: 0.6181 - val_a
Epoch 51/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.9116 - loss: 0.5746 - val_a
Epoch 52/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.8872 - loss: 0.5841 - val_a
Epoch 53/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.8893 - loss: 0.5780 - val_a
Epoch 54/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.8956 - loss: 0.6059 - val_a
Epoch 55/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9395 - loss: 0.5500 - val_a
Epoch 56/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.8966 - loss: 0.5798 - val_a
Epoch 57/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 0.9229 - loss: 0.5529 - val_a
Epoch 58/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.8716 - loss: 0.5526 - val_a
Epoch 59/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.9072 - loss: 0.5483 - val_a
Epoch 60/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 0.9006 - loss: 0.5402 - val_a
Epoch 61/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.9189 - loss: 0.5427 - val_a
Epoch 62/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.9156 - loss: 0.5324 - val_a
Epoch 63/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9468 - loss: 0.5215 - val_a
Epoch 64/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9508 - loss: 0.5262 - val_a
Epoch 65/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step - accuracy: 0.9708 - loss: 0.4913 - val_a
Epoch 66/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.9341 - loss: 0.5100 - val_a
Epoch 67/300
```

```
Epoch 67/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9491 - loss: 0.5097 - val_a
Epoch 68/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.9575 - loss: 0.5154 - val_a
Epoch 69/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 57ms/step - accuracy: 0.9235 - loss: 0.5242 - val_a
Epoch 70/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.9441 - loss: 0.4937 - val_a
Epoch 71/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9475 - loss: 0.4871 - val_a
Epoch 72/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.9531 - loss: 0.4931 - val_a
Epoch 73/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.9381 - loss: 0.4995 - val_a
Epoch 74/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.9614 - loss: 0.4855 - val_a
Epoch 75/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9325 - loss: 0.4596 - val_a
Epoch 76/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.9291 - loss: 0.4701 - val_a
Epoch 77/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.9341 - loss: 0.4729 - val_a
Epoch 78/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9714 - loss: 0.4358 - val_a
Epoch 79/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 0.9508 - loss: 0.4345 - val_a
Epoch 80/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9741 - loss: 0.3947 - val_a
Epoch 81/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.9631 - loss: 0.4289 - val_a
Epoch 82/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.9531 - loss: 0.4228 - val_a
Epoch 83/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9631 - loss: 0.4366 - val_a
Epoch 84/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.9525 - loss: 0.3735 - val_a
Epoch 85/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.9714 - loss: 0.3882 - val_a
Epoch 86/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step - accuracy: 0.9491 - loss: 0.3836 - val_a
Epoch 87/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.9541 - loss: 0.3768 - val_a
Epoch 88/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9621 - loss: 0.4098 - val_a
Epoch 89/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 0.9531 - loss: 0.3852 - val_a
Epoch 90/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 97ms/step - accuracy: 0.9441 - loss: 0.3876 - val_a
Epoch 91/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 69ms/step - accuracy: 0.9821 - loss: 0.3626 - val_a
Epoch 92/300
```

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 69ms/step - accuracy: 0.9675 - loss: 0.3660 - val_a
Epoch 93/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 92ms/step - accuracy: 0.9631 - loss: 0.3704 - val_a
Epoch 94/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 97ms/step - accuracy: 0.9591 - loss: 0.3511 - val_a
Epoch 95/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 102ms/step - accuracy: 0.9681 - loss: 0.3541 - val_
Epoch 96/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 78ms/step - accuracy: 0.9681 - loss: 0.3645 - val_a
Epoch 97/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 72ms/step - accuracy: 0.9481 - loss: 0.3693 - val_a
Epoch 98/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9625 - loss: 0.3486 - val_a
Epoch 99/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9621 - loss: 0.3312 - val_a
Epoch 100/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.9564 - loss: 0.3402 - val_a
Epoch 101/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.9508 - loss: 0.3590 - val_a
Epoch 102/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9664 - loss: 0.3168 - val_a
Epoch 103/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.9704 - loss: 0.3312 - val_a
Epoch 104/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.9381 - loss: 0.3339 - val_a
Epoch 105/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step - accuracy: 0.9681 - loss: 0.2923 - val_a
Epoch 106/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.9464 - loss: 0.3107 - val_a
Epoch 107/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.9631 - loss: 0.2996 - val_a
Epoch 108/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.9681 - loss: 0.3052 - val_a
Epoch 109/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.9521 - loss: 0.3166 - val_a
Epoch 110/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 0.9714 - loss: 0.2772 - val_a
Epoch 111/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9481 - loss: 0.2979 - val_a
Epoch 112/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.9671 - loss: 0.2935 - val_a
Epoch 113/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.9614 - loss: 0.2749 - val_a
Epoch 114/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.9481 - loss: 0.2924 - val_a
Epoch 115/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9564 - loss: 0.2739 - val_a
Epoch 116/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9704 - loss: 0.2770 - val_a
Epoch 117/300
```

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.9860 - loss: 0.2708 - val_a
Epoch 118/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9671 - loss: 0.2794 - val_a
Epoch 119/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.9771 - loss: 0.2702 - val_a
Epoch 120/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - accuracy: 0.9760 - loss: 0.2624 - val_a
Epoch 121/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9854 - loss: 0.2522 - val_a
Epoch 122/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.9944 - loss: 0.2359 - val_a
Epoch 123/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9760 - loss: 0.2551 - val_a
Epoch 124/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9887 - loss: 0.2371 - val_a
Epoch 125/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.9621 - loss: 0.2477 - val_a
Epoch 126/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.9910 - loss: 0.2252 - val_a
Epoch 127/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - accuracy: 0.9910 - loss: 0.2361 - val_a
Epoch 128/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9860 - loss: 0.2367 - val_a
Epoch 129/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 0.9760 - loss: 0.2434 - val_a
Epoch 130/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9860 - loss: 0.2418 - val_a
Epoch 131/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.9860 - loss: 0.2299 - val_a
Epoch 132/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9910 - loss: 0.2180 - val_a
Epoch 133/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step - accuracy: 0.9760 - loss: 0.2220 - val_a
Epoch 134/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9910 - loss: 0.2147 - val_a
Epoch 135/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.9760 - loss: 0.2162 - val_a
Epoch 136/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9760 - loss: 0.2122 - val_a
Epoch 137/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.9860 - loss: 0.2075 - val_a
Epoch 138/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 81ms/step - accuracy: 1.0000 - loss: 0.2043 - val_a
Epoch 139/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 95ms/step - accuracy: 0.9760 - loss: 0.2134 - val_a
Epoch 140/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 70ms/step - accuracy: 0.9760 - loss: 0.2044 - val_a
Epoch 141/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 94ms/step - accuracy: 0.9860 - loss: 0.1949 - val_a
Epoch 142/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 72ms/step - accuracy: 1.0000 - loss: 0.1949 - val_a
```

```
Epoch 143/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 99ms/step - accuracy: 1.0000 - loss: 0.1982 - val_a
Epoch 144/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 107ms/step - accuracy: 1.0000 - loss: 0.1847 - val_
Epoch 145/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 80ms/step - accuracy: 1.0000 - loss: 0.1841 - val_a
Epoch 146/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 58ms/step - accuracy: 1.0000 - loss: 0.1823 - val_a
Epoch 147/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 1.0000 - loss: 0.1813 - val_a
Epoch 148/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.1854 - val_a
Epoch 149/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 1.0000 - loss: 0.1631 - val_a
Epoch 150/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 1.0000 - loss: 0.1659 - val_a
Epoch 151/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.1762 - val_a
Epoch 152/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.1793 - val_a
Epoch 153/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.1561 - val_a
Epoch 154/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 1.0000 - loss: 0.1598 - val_a
Epoch 155/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 1.0000 - loss: 0.1623 - val_a
Epoch 156/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 1.0000 - loss: 0.1547 - val_a
Epoch 157/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.1571 - val_a
Epoch 158/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.1535 - val_a
Epoch 159/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.1576 - val_a
Epoch 160/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.1426 - val_a
Epoch 161/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.1497 - val_a
Epoch 162/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.1466 - val_a
Epoch 163/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.1416 - val_a
Epoch 164/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.1424 - val_a
Epoch 165/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.1477 - val_a
Epoch 166/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.1323 - val_a
Epoch 167/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 1.0000 - loss: 0.1462 - val_a
```

```
Epoch 168/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.1279 - val_a
Epoch 169/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.1332 - val_a
Epoch 170/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 1.0000 - loss: 0.1351 - val_a
Epoch 171/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.1324 - val_a
Epoch 172/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.1334 - val_a
Epoch 173/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - accuracy: 1.0000 - loss: 0.1310 - val_a
Epoch 174/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step - accuracy: 1.0000 - loss: 0.1369 - val_a
Epoch 175/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 1.0000 - loss: 0.1221 - val_a
Epoch 176/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.1142 - val_a
Epoch 177/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - accuracy: 1.0000 - loss: 0.1067 - val_a
Epoch 178/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.1150 - val_a
Epoch 179/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.1152 - val_a
Epoch 180/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 1.0000 - loss: 0.1203 - val_a
Epoch 181/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.1174 - val_a
Epoch 182/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.1114 - val_a
Epoch 183/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.1069 - val_a
Epoch 184/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 1.0000 - loss: 0.1047 - val_a
Epoch 185/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 1.0000 - loss: 0.1006 - val_a
Epoch 186/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 97ms/step - accuracy: 1.0000 - loss: 0.1069 - val_a
Epoch 187/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 96ms/step - accuracy: 1.0000 - loss: 0.1007 - val_a
Epoch 188/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 100ms/step - accuracy: 1.0000 - loss: 0.1024 - val_
Epoch 189/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 97ms/step - accuracy: 1.0000 - loss: 0.1053 - val_a
Epoch 190/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 76ms/step - accuracy: 1.0000 - loss: 0.0979 - val_a
Epoch 191/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 99ms/step - accuracy: 1.0000 - loss: 0.1002 - val_a
Epoch 192/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.0985 - val_a
Epoch 193/300
```

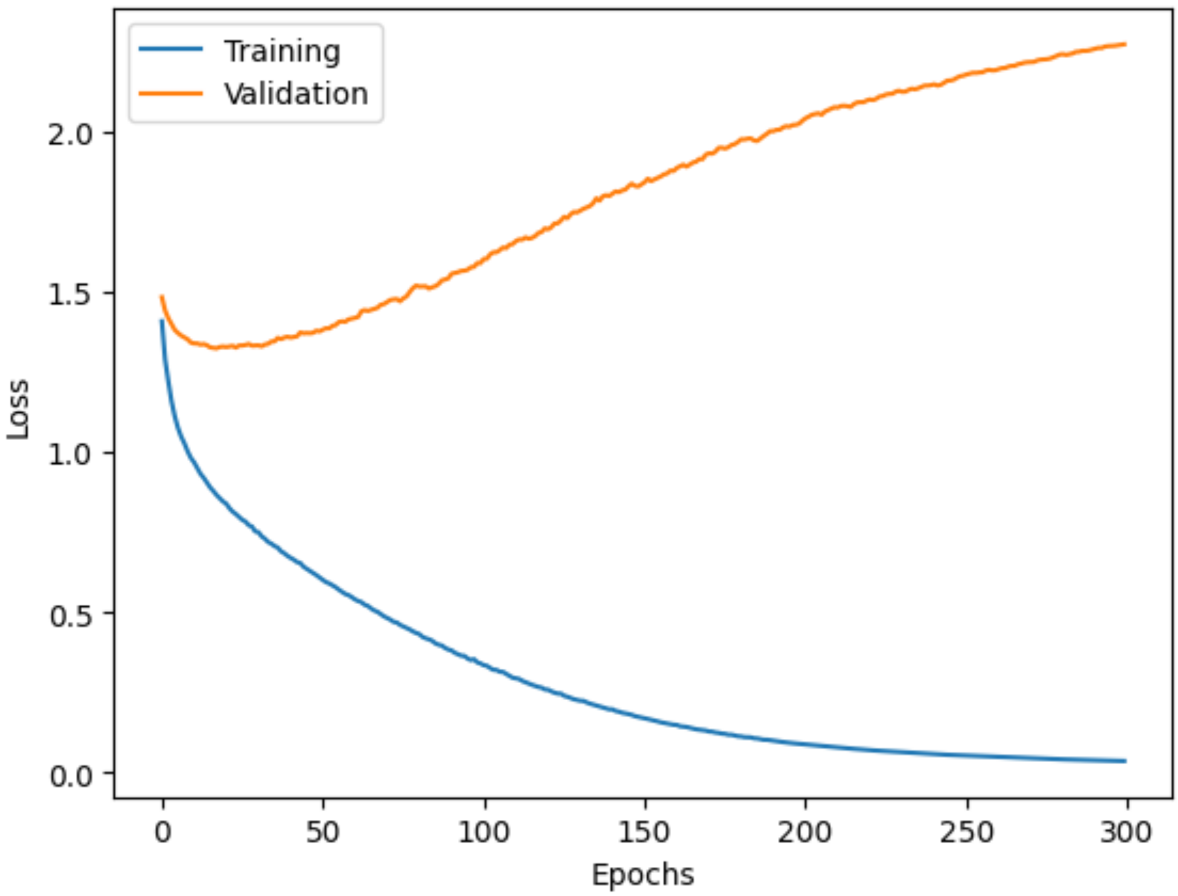
```
Epoch 193/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0915 - val_a
Epoch 194/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.0915 - val_a
Epoch 195/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 1.0000 - loss: 0.0901 - val_a
Epoch 196/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0915 - val_a
Epoch 197/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0922 - val_a
Epoch 198/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.0915 - val_a
Epoch 199/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - accuracy: 1.0000 - loss: 0.0919 - val_a
Epoch 200/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0933 - val_a
Epoch 201/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.0864 - val_a
Epoch 202/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0850 - val_a
Epoch 203/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0864 - val_a
Epoch 204/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - accuracy: 1.0000 - loss: 0.0916 - val_a
Epoch 205/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.0813 - val_a
Epoch 206/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.0791 - val_a
Epoch 207/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.0812 - val_a
Epoch 208/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 1.0000 - loss: 0.0759 - val_a
Epoch 209/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0809 - val_a
Epoch 210/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.0823 - val_a
Epoch 211/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.0765 - val_a
Epoch 212/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.0722 - val_a
Epoch 213/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0816 - val_a
Epoch 214/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0794 - val_a
Epoch 215/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - accuracy: 1.0000 - loss: 0.0724 - val_a
Epoch 216/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.0799 - val_a
Epoch 217/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0700 - val_a
Epoch 218/300
```

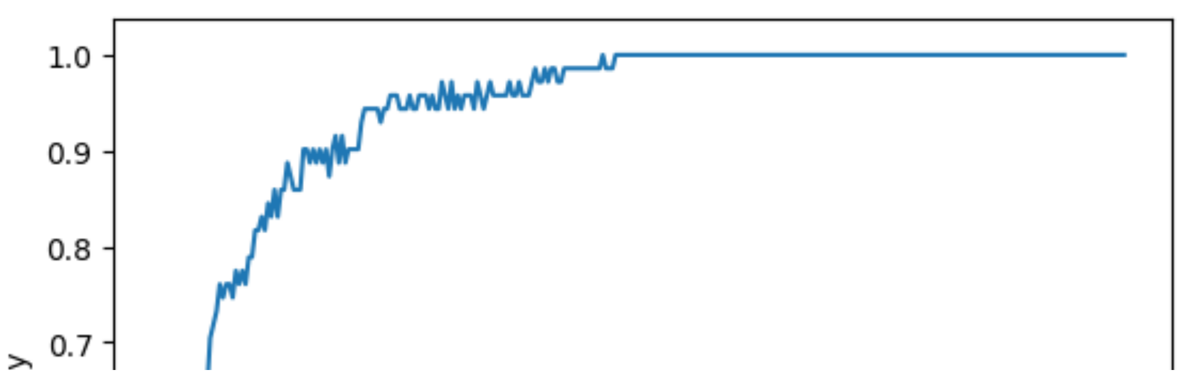
```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.0669 - val_a
Epoch 219/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 1.0000 - loss: 0.0704 - val_a
Epoch 220/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - accuracy: 1.0000 - loss: 0.0716 - val_a
Epoch 221/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 1.0000 - loss: 0.0686 - val_a
Epoch 222/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0666 - val_a
Epoch 223/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 1.0000 - loss: 0.0673 - val_a
Epoch 224/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0655 - val_a
Epoch 225/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0713 - val_a
Epoch 226/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.0673 - val_a
Epoch 227/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.0687 - val_a
Epoch 228/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 1.0000 - loss: 0.0643 - val_a
Epoch 229/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.0627 - val_a
Epoch 230/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0648 - val_a
Epoch 231/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0649 - val_a
Epoch 232/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 94ms/step - accuracy: 1.0000 - loss: 0.0632 - val_a
Epoch 233/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 71ms/step - accuracy: 1.0000 - loss: 0.0622 - val_a
Epoch 234/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 95ms/step - accuracy: 1.0000 - loss: 0.0641 - val_a
Epoch 235/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 97ms/step - accuracy: 1.0000 - loss: 0.0581 - val_a
Epoch 236/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 97ms/step - accuracy: 1.0000 - loss: 0.0588 - val_a
Epoch 237/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 105ms/step - accuracy: 1.0000 - loss: 0.0608 - val_
Epoch 238/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 99ms/step - accuracy: 1.0000 - loss: 0.0593 - val_a
Epoch 239/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 106ms/step - accuracy: 1.0000 - loss: 0.0603 - val_
Epoch 240/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 66ms/step - accuracy: 1.0000 - loss: 0.0573 - val_a
Epoch 241/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.0558 - val_a
Epoch 242/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.0558 - val_a
Epoch 243/300
```

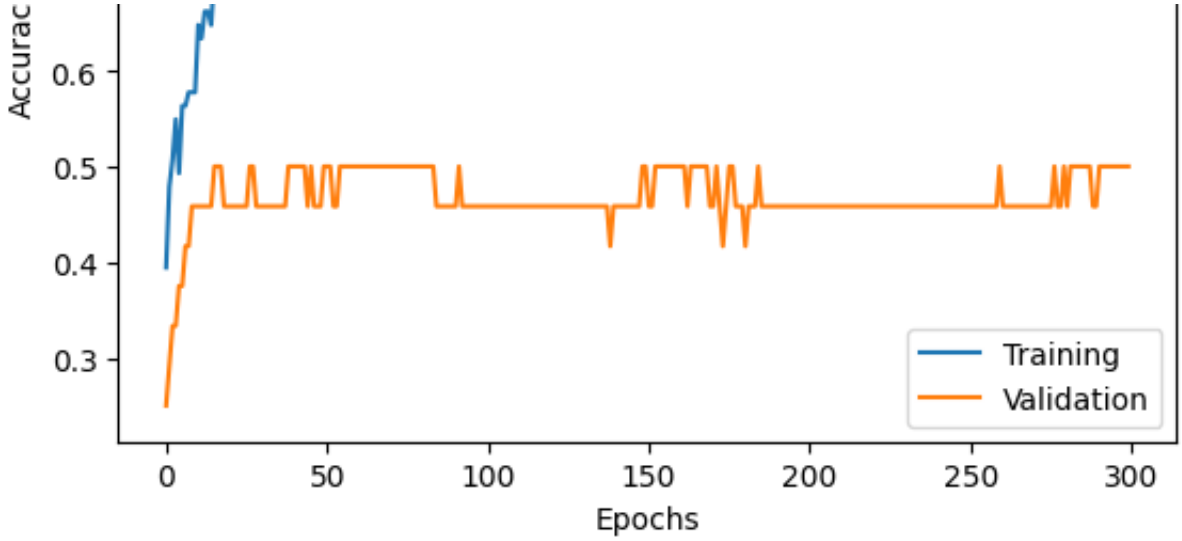
```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 1.0000 - loss: 0.0568 - val_a
Epoch 244/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0579 - val_a
Epoch 245/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.0574 - val_a
Epoch 246/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.0510 - val_a
Epoch 247/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.0516 - val_a
Epoch 248/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0550 - val_a
Epoch 249/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.0520 - val_a
Epoch 250/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.0513 - val_a
Epoch 251/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.0501 - val_a
Epoch 252/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0545 - val_a
Epoch 253/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 1.0000 - loss: 0.0511 - val_a
Epoch 254/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.0485 - val_a
Epoch 255/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step - accuracy: 1.0000 - loss: 0.0479 - val_a
Epoch 256/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0499 - val_a
Epoch 257/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 1.0000 - loss: 0.0477 - val_a
Epoch 258/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - accuracy: 1.0000 - loss: 0.0461 - val_a
Epoch 259/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0469 - val_a
Epoch 260/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.0480 - val_a
Epoch 261/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0463 - val_a
Epoch 262/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - accuracy: 1.0000 - loss: 0.0475 - val_a
Epoch 263/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0453 - val_a
Epoch 264/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0452 - val_a
Epoch 265/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.0467 - val_a
Epoch 266/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 1.0000 - loss: 0.0453 - val_a
Epoch 267/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.0453 - val_a
Epoch 268/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 1.0000 - loss: 0.0447 - val_a
```

```
Epoch 269/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.0442 - val_a
Epoch 270/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.0425 - val_a
Epoch 271/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.0418 - val_a
Epoch 272/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - accuracy: 1.0000 - loss: 0.0433 - val_a
Epoch 273/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.0454 - val_a
Epoch 274/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 1.0000 - loss: 0.0448 - val_a
Epoch 275/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 1.0000 - loss: 0.0403 - val_a
Epoch 276/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0388 - val_a
Epoch 277/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0428 - val_a
Epoch 278/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 1.0000 - loss: 0.0425 - val_a
Epoch 279/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.0411 - val_a
Epoch 280/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 1.0000 - loss: 0.0422 - val_a
Epoch 281/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 101ms/step - accuracy: 1.0000 - loss: 0.0398 - val_
Epoch 282/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 99ms/step - accuracy: 1.0000 - loss: 0.0408 - val_a
Epoch 283/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 89ms/step - accuracy: 1.0000 - loss: 0.0408 - val_a
Epoch 284/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 93ms/step - accuracy: 1.0000 - loss: 0.0404 - val_a
Epoch 285/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 94ms/step - accuracy: 1.0000 - loss: 0.0410 - val_a
Epoch 286/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 100ms/step - accuracy: 1.0000 - loss: 0.0402 - val_
Epoch 287/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 82ms/step - accuracy: 1.0000 - loss: 0.0392 - val_a
Epoch 288/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 59ms/step - accuracy: 1.0000 - loss: 0.0380 - val_a
Epoch 289/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - accuracy: 1.0000 - loss: 0.0393 - val_a
Epoch 290/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.0370 - val_a
Epoch 291/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.0368 - val_a
Epoch 292/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0380 - val_a
Epoch 293/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.0352 - val_a
```

```
Epoch 294/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.0349 - val_a
Epoch 295/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.0390 - val_a
Epoch 296/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 1.0000 - loss: 0.0371 - val_a
Epoch 297/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 1.0000 - loss: 0.0348 - val_a
Epoch 298/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 1.0000 - loss: 0.0365 - val_a
Epoch 299/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 1.0000 - loss: 0.0353 - val_a
Epoch 300/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 1.0000 - loss: 0.0355 - val_a
<matplotlib.legend.Legend at 0x7c7510474ad0>
```








```
# Sigmoid activation
sigmoid_model = Sequential([
        Dense(128, input_dim=feature_count, activation='sigmoid'),
        Dense(64, activation='sigmoid'),
        Dense(64, activation='sigmoid'),
        Dense(32, activation='sigmoid'),
        Dense(num_classes, activation='softmax')
    ])

# Compile the model
sigmoid_model.compile(optimizer='sgd', loss='categorical_crossentropy', metrics=[

# Fitting the model to the Training set
history = sigmoid_model.fit(preprocessor(X_train), y_train_encoded,
               batch_size = 20,
               epochs = 300, validation_split=0.25)

#plot loss and accuracy at each epoch
plt.figure()
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.legend(['Training', 'Validation'])

plt.figure()
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
```

```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.legend(['Training', 'Validation'], loc='lower right')
```

```
Epoch 1/300
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87: User
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 139ms/step - accuracy: 0.2451 - loss: 1.6709 - val_
Epoch 2/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2035 - loss: 1.6453 - val_a
Epoch 3/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2268 - loss: 1.6655 - val_a
Epoch 4/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.1851 - loss: 1.7040 - val_a
Epoch 5/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.2318 - loss: 1.6359 - val_a
Epoch 6/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.1918 - loss: 1.6314 - val_a
Epoch 7/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.1885 - loss: 1.6454 - val_a
Epoch 8/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2268 - loss: 1.6232 - val_a
Epoch 9/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.2101 - loss: 1.6267 - val_a
Epoch 10/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2401 - loss: 1.5982 - val_a
Epoch 11/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2135 - loss: 1.6141 - val_a
Epoch 12/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2085 - loss: 1.6136 - val_a
Epoch 13/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2318 - loss: 1.6266 - val_a
Epoch 14/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.2301 - loss: 1.5969 - val_a
Epoch 15/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.1868 - loss: 1.6097 - val_a
Epoch 16/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2351 - loss: 1.6105 - val_a
Epoch 17/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2085 - loss: 1.6298 - val_a
Epoch 18/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.2435 - loss: 1.6388 - val_a
Epoch 19/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2568 - loss: 1.6049 - val_a
Epoch 20/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.2535 - loss: 1.5969 - val_a
Epoch 21/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2518 - loss: 1.6054 - val_a
Epoch 22/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.2035 - loss: 1.6126 - val_a
```

```
Epoch 23/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 70ms/step - accuracy: 0.2485 - loss: 1.6050 - val_a
Epoch 24/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 94ms/step - accuracy: 0.2168 - loss: 1.6075 - val_a
Epoch 25/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 65ms/step - accuracy: 0.2235 - loss: 1.5937 - val_a
Epoch 26/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 64ms/step - accuracy: 0.1901 - loss: 1.6114 - val_a
Epoch 27/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 68ms/step - accuracy: 0.1801 - loss: 1.6168 - val_a
Epoch 28/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 91ms/step - accuracy: 0.2901 - loss: 1.5980 - val_a
Epoch 29/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 66ms/step - accuracy: 0.2285 - loss: 1.5975 - val_a
Epoch 30/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.1751 - loss: 1.6115 - val_a
Epoch 31/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 100ms/step - accuracy: 0.1885 - loss: 1.6115 - val_
Epoch 32/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 96ms/step - accuracy: 0.2318 - loss: 1.5980 - val_a
Epoch 33/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2135 - loss: 1.5975 - val_a
Epoch 34/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.1885 - loss: 1.6122 - val_a
Epoch 35/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2301 - loss: 1.5957 - val_a
Epoch 36/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2335 - loss: 1.5955 - val_a
Epoch 37/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2118 - loss: 1.6092 - val_a
Epoch 38/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2401 - loss: 1.6050 - val_a
Epoch 39/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.2035 - loss: 1.6018 - val_a
Epoch 40/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2285 - loss: 1.5945 - val_a
Epoch 41/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.1918 - loss: 1.6135 - val_a
Epoch 42/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.1951 - loss: 1.6016 - val_a
Epoch 43/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2468 - loss: 1.5891 - val_a
Epoch 44/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2251 - loss: 1.6081 - val_a
Epoch 45/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.1901 - loss: 1.6093 - val_a
Epoch 46/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.2468 - loss: 1.5885 - val_a
Epoch 47/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2335 - loss: 1.5989 - val_a
```

```
Epoch 48/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2135 - loss: 1.6022 - val_a
Epoch 49/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2135 - loss: 1.6096 - val_a
Epoch 50/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2351 - loss: 1.6020 - val_a
Epoch 51/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.2268 - loss: 1.5939 - val_a
Epoch 52/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2368 - loss: 1.6095 - val_a
Epoch 53/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.1968 - loss: 1.6121 - val_a
Epoch 54/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2118 - loss: 1.6110 - val_a
Epoch 55/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.2568 - loss: 1.6107 - val_a
Epoch 56/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.1851 - loss: 1.6179 - val_a
Epoch 57/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2268 - loss: 1.6067 - val_a
Epoch 58/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2135 - loss: 1.6067 - val_a
Epoch 59/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.2468 - loss: 1.6019 - val_a
Epoch 60/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2335 - loss: 1.5861 - val_a
Epoch 61/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2401 - loss: 1.6089 - val_a
Epoch 62/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2851 - loss: 1.5934 - val_a
Epoch 63/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.1851 - loss: 1.6062 - val_a
Epoch 64/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.2235 - loss: 1.5971 - val_a
Epoch 65/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2301 - loss: 1.6076 - val_a
Epoch 66/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2068 - loss: 1.6148 - val_a
Epoch 67/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2485 - loss: 1.6016 - val_a
Epoch 68/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2051 - loss: 1.6015 - val_a
Epoch 69/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step - accuracy: 0.2468 - loss: 1.5933 - val_a
Epoch 70/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2401 - loss: 1.6113 - val_a
Epoch 71/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2001 - loss: 1.6181 - val_a
Epoch 72/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2468 - loss: 1.6002 - val_a
Epoch 73/300
```

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.2168 - loss: 1.6056 - val_a
Epoch 74/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2368 - loss: 1.6039 - val_a
Epoch 75/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 98ms/step - accuracy: 0.2118 - loss: 1.6027 - val_a
Epoch 76/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 83ms/step - accuracy: 0.2351 - loss: 1.6060 - val_a
Epoch 77/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 118ms/step - accuracy: 0.1818 - loss: 1.6089 - val_
Epoch 78/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 89ms/step - accuracy: 0.2218 - loss: 1.6166 - val_a
Epoch 79/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 110ms/step - accuracy: 0.2218 - loss: 1.5978 - val_
Epoch 80/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 101ms/step - accuracy: 0.2101 - loss: 1.6041 - val_
Epoch 81/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 94ms/step - accuracy: 0.2335 - loss: 1.6004 - val_a
Epoch 82/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 106ms/step - accuracy: 0.2235 - loss: 1.5968 - val_
Epoch 83/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 116ms/step - accuracy: 0.2651 - loss: 1.5931 - val_
Epoch 84/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 94ms/step - accuracy: 0.2301 - loss: 1.6023 - val_a
Epoch 85/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 104ms/step - accuracy: 0.2451 - loss: 1.6099 - val_
Epoch 86/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step - accuracy: 0.2101 - loss: 1.5970 - val_a
Epoch 87/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.2368 - loss: 1.5995 - val_a
Epoch 88/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2885 - loss: 1.5966 - val_a
Epoch 89/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2118 - loss: 1.6033 - val_a
Epoch 90/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2135 - loss: 1.6072 - val_a
Epoch 91/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.2351 - loss: 1.5994 - val_a
Epoch 92/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.2385 - loss: 1.6047 - val_a
Epoch 93/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2485 - loss: 1.6051 - val_a
Epoch 94/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.1786 - loss: 1.5928 - val_a
Epoch 95/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.2301 - loss: 1.5982 - val_a
Epoch 96/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2135 - loss: 1.6102 - val_a
Epoch 97/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.2035 - loss: 1.6158 - val_a
Epoch 98/300
```

```
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2268 - loss: 1.5969 - val_a
:poch 99/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.2368 - loss: 1.5999 - val_a
:poch 100/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.1968 - loss: 1.6056 - val_a
:poch 101/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2801 - loss: 1.5934 - val_a
:poch 102/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.2301 - loss: 1.5910 - val_a
:poch 103/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.2001 - loss: 1.6184 - val_a
:poch 104/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2368 - loss: 1.6001 - val_a
:poch 105/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2001 - loss: 1.6032 - val_a
:poch 106/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2251 - loss: 1.6016 - val_a
:poch 107/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2685 - loss: 1.5961 - val_a
:poch 108/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2085 - loss: 1.6224 - val_a
:poch 109/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2168 - loss: 1.6013 - val_a
:poch 110/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.1901 - loss: 1.6086 - val_a
:poch 111/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.1885 - loss: 1.6030 - val_a
:poch 112/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2351 - loss: 1.6025 - val_a
:poch 113/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.2301 - loss: 1.6056 - val_a
:poch 114/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.2268 - loss: 1.6084 - val_a
:poch 115/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2168 - loss: 1.5987 - val_a
:poch 116/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.1751 - loss: 1.6028 - val_a
:poch 117/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.2351 - loss: 1.6022 - val_a
:poch 118/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2035 - loss: 1.6082 - val_a
:poch 119/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2151 - loss: 1.6011 - val_a
:poch 120/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2435 - loss: 1.6096 - val_a
:poch 121/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2168 - loss: 1.6083 - val_a
:poch 122/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2118 - loss: 1.5995 - val_a
:poch 123/300
ı/4 ━━━━━━━━━━━━━━━━━━━━ 0s 96ms/step - accuracy: 0.2551 - loss: 1.5967 - val_a
```

```
Epoch 124/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 68ms/step - accuracy: 0.2085 - loss: 1.6099 - val_a
Epoch 125/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 91ms/step - accuracy: 0.2435 - loss: 1.5924 - val_a
Epoch 126/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 72ms/step - accuracy: 0.2268 - loss: 1.6005 - val_a
Epoch 127/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 72ms/step - accuracy: 0.2501 - loss: 1.5928 - val_a
Epoch 128/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 92ms/step - accuracy: 0.2268 - loss: 1.6037 - val_a
Epoch 129/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 94ms/step - accuracy: 0.2568 - loss: 1.6017 - val_a
Epoch 130/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 100ms/step - accuracy: 0.2401 - loss: 1.5887 - val_
Epoch 131/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 94ms/step - accuracy: 0.2451 - loss: 1.6028 - val_a
Epoch 132/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2501 - loss: 1.6099 - val_a
Epoch 133/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2551 - loss: 1.6091 - val_a
Epoch 134/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.2335 - loss: 1.6014 - val_a
Epoch 135/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.2168 - loss: 1.6119 - val_a
Epoch 136/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.1430 - loss: 1.6018 - val_a
Epoch 137/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2368 - loss: 1.6143 - val_a
Epoch 138/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.1130 - loss: 1.5988 - val_a
Epoch 139/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.2418 - loss: 1.6030 - val_a
Epoch 140/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.2301 - loss: 1.5984 - val_a
Epoch 141/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2651 - loss: 1.5939 - val_a
Epoch 142/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2501 - loss: 1.5976 - val_a
Epoch 143/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2468 - loss: 1.6107 - val_a
Epoch 144/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.1889 - loss: 1.6066 - val_a
Epoch 145/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - accuracy: 0.1939 - loss: 1.6103 - val_a
Epoch 146/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.1782 - loss: 1.6019 - val_a
Epoch 147/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2401 - loss: 1.5955 - val_a
Epoch 148/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.2435 - loss: 1.5987 - val_a
```

```
Epoch 149/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.1893 - loss: 1.6054 - val_a
Epoch 150/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.2568 - loss: 1.6010 - val_a
Epoch 151/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2318 - loss: 1.6063 - val_a
Epoch 152/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - accuracy: 0.2135 - loss: 1.6086 - val_a
Epoch 153/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.1985 - loss: 1.6023 - val_a
Epoch 154/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.2585 - loss: 1.5948 - val_a
Epoch 155/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2285 - loss: 1.6021 - val_a
Epoch 156/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2168 - loss: 1.6141 - val_a
Epoch 157/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.1851 - loss: 1.6143 - val_a
Epoch 158/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2068 - loss: 1.5940 - val_a
Epoch 159/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2318 - loss: 1.6074 - val_a
Epoch 160/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2051 - loss: 1.6131 - val_a
Epoch 161/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2085 - loss: 1.6148 - val_a
Epoch 162/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.2268 - loss: 1.5960 - val_a
Epoch 163/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2301 - loss: 1.6003 - val_a
Epoch 164/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.2235 - loss: 1.6043 - val_a
Epoch 165/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2185 - loss: 1.5959 - val_a
Epoch 166/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.2085 - loss: 1.6163 - val_a
Epoch 167/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2618 - loss: 1.5984 - val_a
Epoch 168/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2435 - loss: 1.5901 - val_a
Epoch 169/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2001 - loss: 1.6082 - val_a
Epoch 170/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step - accuracy: 0.2285 - loss: 1.6123 - val_a
Epoch 171/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2435 - loss: 1.5909 - val_a
Epoch 172/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.1885 - loss: 1.6072 - val_a
Epoch 173/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2168 - loss: 1.6040 - val_a
```

```
Epoch 174/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 99ms/step - accuracy: 0.2235 - loss: 1.6028 - val_a
Epoch 175/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 91ms/step - accuracy: 0.2085 - loss: 1.6032 - val_a
Epoch 176/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 93ms/step - accuracy: 0.1668 - loss: 1.6080 - val_a
Epoch 177/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 90ms/step - accuracy: 0.2235 - loss: 1.6005 - val_a
Epoch 178/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 73ms/step - accuracy: 0.2018 - loss: 1.5921 - val_a
Epoch 179/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 99ms/step - accuracy: 0.1918 - loss: 1.6007 - val_a
Epoch 180/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 92ms/step - accuracy: 0.2468 - loss: 1.6065 - val_a
Epoch 181/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.2451 - loss: 1.6060 - val_a
Epoch 182/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2301 - loss: 1.5870 - val_a
Epoch 183/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2001 - loss: 1.5972 - val_a
Epoch 184/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.2435 - loss: 1.5915 - val_a
Epoch 185/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2468 - loss: 1.5973 - val_a
Epoch 186/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2368 - loss: 1.6086 - val_a
Epoch 187/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2468 - loss: 1.6034 - val_a
Epoch 188/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2401 - loss: 1.6158 - val_a
Epoch 189/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2068 - loss: 1.6031 - val_a
Epoch 190/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.1622 - loss: 1.5989 - val_a
Epoch 191/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.2535 - loss: 1.5907 - val_a
Epoch 192/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.2535 - loss: 1.5973 - val_a
Epoch 193/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2285 - loss: 1.5946 - val_a
Epoch 194/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2101 - loss: 1.6065 - val_a
Epoch 195/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2301 - loss: 1.6063 - val_a
Epoch 196/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.2535 - loss: 1.6094 - val_a
Epoch 197/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.2568 - loss: 1.5935 - val_a
Epoch 198/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.1885 - loss: 1.5988 - val_a
Epoch 199/300
```

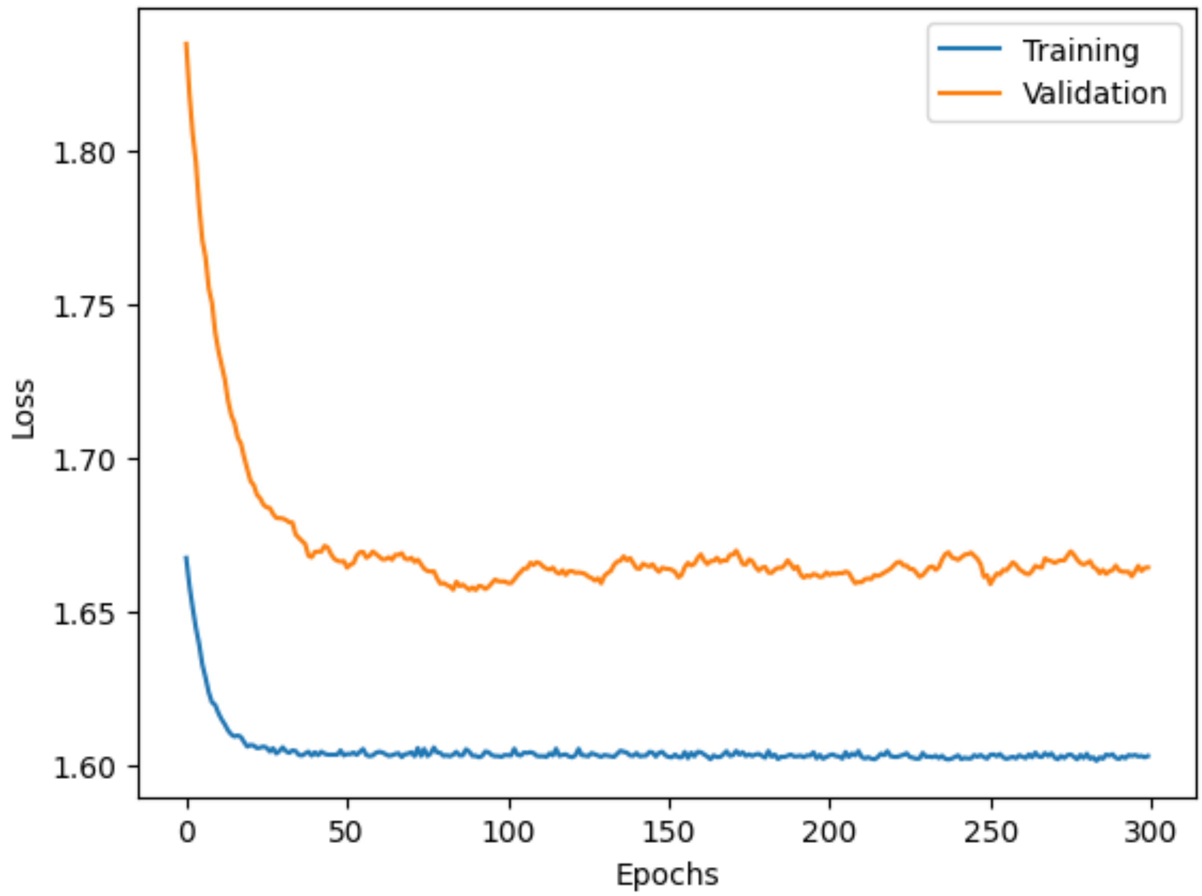
```
-
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2035 - loss: 1.6069 - val_a
Epoch 200/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.1868 - loss: 1.6028 - val_a
Epoch 201/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2201 - loss: 1.5950 - val_a
Epoch 202/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.1868 - loss: 1.6169 - val_a
Epoch 203/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2301 - loss: 1.5996 - val_a
Epoch 204/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2701 - loss: 1.5948 - val_a
Epoch 205/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.2235 - loss: 1.5953 - val_a
Epoch 206/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2651 - loss: 1.6035 - val_a
Epoch 207/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2018 - loss: 1.6096 - val_a
Epoch 208/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.1985 - loss: 1.6027 - val_a
Epoch 209/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step - accuracy: 0.2335 - loss: 1.5983 - val_a
Epoch 210/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2068 - loss: 1.6005 - val_a
Epoch 211/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.2518 - loss: 1.5976 - val_a
Epoch 212/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2118 - loss: 1.6096 - val_a
Epoch 213/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.2318 - loss: 1.6120 - val_a
Epoch 214/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2418 - loss: 1.5978 - val_a
Epoch 215/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2335 - loss: 1.6014 - val_a
Epoch 216/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2151 - loss: 1.6073 - val_a
Epoch 217/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.2218 - loss: 1.6034 - val_a
Epoch 218/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.2468 - loss: 1.6060 - val_a
Epoch 219/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.3001 - loss: 1.5961 - val_a
Epoch 220/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.2937 - loss: 1.6001 - val_a
Epoch 221/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 69ms/step - accuracy: 0.2418 - loss: 1.6000 - val_a
Epoch 222/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 92ms/step - accuracy: 0.2251 - loss: 1.6014 - val_a
Epoch 223/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 93ms/step - accuracy: 0.2651 - loss: 1.5996 - val_a
Epoch 224/300
```

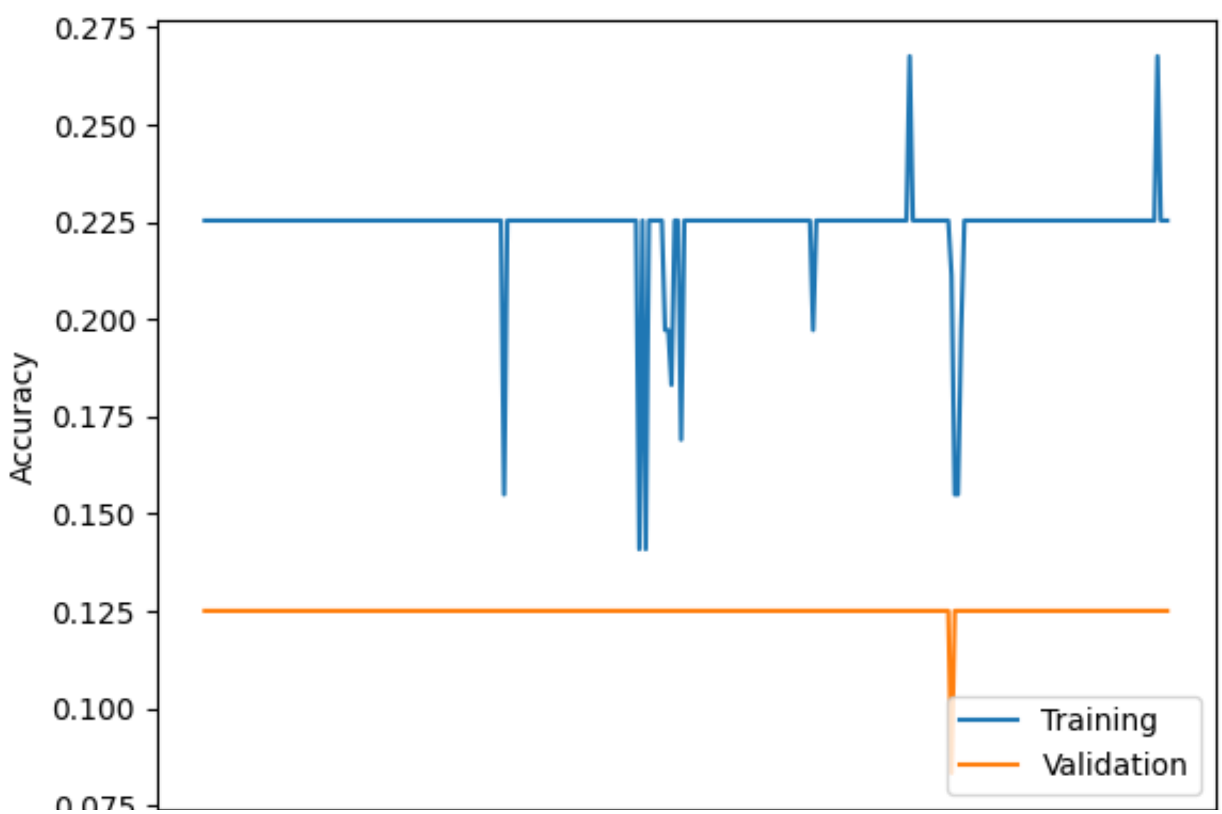
```
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 97ms/step - accuracy: 0.1935 - loss: 1.5946 - val_a
Epoch 225/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 1s 102ms/step - accuracy: 0.1868 - loss: 1.6179 - val_
Epoch 226/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 1s 67ms/step - accuracy: 0.2435 - loss: 1.5935 - val_a
Epoch 227/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2201 - loss: 1.5933 - val_a
Epoch 228/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2218 - loss: 1.6009 - val_a
Epoch 229/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2418 - loss: 1.5943 - val_a
Epoch 230/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2335 - loss: 1.6160 - val_a
Epoch 231/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.2385 - loss: 1.6044 - val_a
Epoch 232/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2368 - loss: 1.6081 - val_a
Epoch 233/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2112 - loss: 1.6072 - val_a
Epoch 234/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.1936 - loss: 1.6079 - val_a
Epoch 235/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.1720 - loss: 1.6045 - val_a
Epoch 236/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.1472 - loss: 1.6085 - val_a
Epoch 237/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.1985 - loss: 1.6081 - val_a
Epoch 238/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2368 - loss: 1.6019 - val_a
Epoch 239/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.2135 - loss: 1.6015 - val_a
Epoch 240/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.1885 - loss: 1.6001 - val_a
Epoch 241/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2468 - loss: 1.5937 - val_a
Epoch 242/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2401 - loss: 1.5979 - val_a
Epoch 243/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 0.2101 - loss: 1.6084 - val_a
Epoch 244/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2085 - loss: 1.6092 - val_a
Epoch 245/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2385 - loss: 1.5943 - val_a
Epoch 246/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.1801 - loss: 1.6172 - val_a
Epoch 247/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.2135 - loss: 1.6011 - val_a
Epoch 248/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.1968 - loss: 1.5970 - val_a
Epoch 249/300
l/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.2635 - loss: 1.5853 - val_a
```

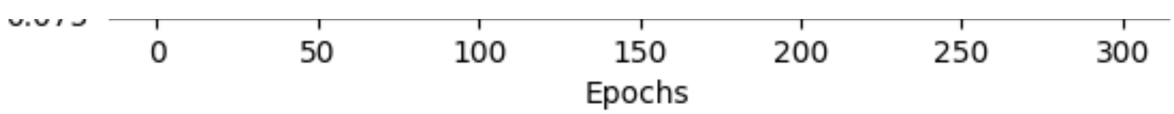
```
Epoch 250/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 0.2418 - loss: 1.5964 - val_a
Epoch 251/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2518 - loss: 1.5915 - val_a
Epoch 252/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.2418 - loss: 1.6001 - val_a
Epoch 253/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.1801 - loss: 1.6126 - val_a
Epoch 254/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.2235 - loss: 1.6002 - val_a
Epoch 255/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.2351 - loss: 1.6067 - val_a
Epoch 256/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2285 - loss: 1.5974 - val_a
Epoch 257/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2485 - loss: 1.6025 - val_a
Epoch 258/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.2235 - loss: 1.6051 - val_a
Epoch 259/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2318 - loss: 1.5937 - val_a
Epoch 260/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2018 - loss: 1.6051 - val_a
Epoch 261/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2068 - loss: 1.5946 - val_a
Epoch 262/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.2535 - loss: 1.6024 - val_a
Epoch 263/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2018 - loss: 1.6067 - val_a
Epoch 264/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2135 - loss: 1.6095 - val_a
Epoch 265/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2135 - loss: 1.6001 - val_a
Epoch 266/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 65ms/step - accuracy: 0.2518 - loss: 1.6020 - val_a
Epoch 267/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 98ms/step - accuracy: 0.2218 - loss: 1.6032 - val_a
Epoch 268/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 91ms/step - accuracy: 0.1851 - loss: 1.6123 - val_a
Epoch 269/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 96ms/step - accuracy: 0.2568 - loss: 1.6009 - val_a
Epoch 270/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 70ms/step - accuracy: 0.2135 - loss: 1.6087 - val_a
Epoch 271/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 67ms/step - accuracy: 0.1785 - loss: 1.6047 - val_a
Epoch 272/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 96ms/step - accuracy: 0.2285 - loss: 1.6033 - val_a
Epoch 273/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.2501 - loss: 1.6026 - val_a
Epoch 274/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.2668 - loss: 1.5983 - val_a
```

```
Epoch 275/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 63ms/step - accuracy: 0.1685 - loss: 1.6062 - val_a
Epoch 276/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2368 - loss: 1.6070 - val_a
Epoch 277/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2218 - loss: 1.6073 - val_a
Epoch 278/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step - accuracy: 0.2268 - loss: 1.6003 - val_a
Epoch 279/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.2951 - loss: 1.5898 - val_a
Epoch 280/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2418 - loss: 1.6068 - val_a
Epoch 281/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2168 - loss: 1.6067 - val_a
Epoch 282/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2485 - loss: 1.5978 - val_a
Epoch 283/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.2618 - loss: 1.5861 - val_a
Epoch 284/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2168 - loss: 1.6013 - val_a
Epoch 285/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2468 - loss: 1.5880 - val_a
Epoch 286/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 57ms/step - accuracy: 0.2385 - loss: 1.6000 - val_a
Epoch 287/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2618 - loss: 1.5970 - val_a
Epoch 288/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2535 - loss: 1.5946 - val_a
Epoch 289/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2401 - loss: 1.6030 - val_a
Epoch 290/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.2385 - loss: 1.5978 - val_a
Epoch 291/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2435 - loss: 1.5947 - val_a
Epoch 292/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2368 - loss: 1.5941 - val_a
Epoch 293/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2151 - loss: 1.6094 - val_a
Epoch 294/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2068 - loss: 1.6057 - val_a
Epoch 295/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.2051 - loss: 1.5998 - val_a
Epoch 296/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - accuracy: 0.2368 - loss: 1.6014 - val_a
Epoch 297/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.2754 - loss: 1.5941 - val_a
Epoch 298/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.2251 - loss: 1.5998 - val_a
Epoch 299/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.2235 - loss: 1.5992 - val_a
Epoch 300/300
```

```
poch 300/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.1951 - loss: 1.6124 - val_a
<matplotlib.legend.Legend at 0x7c757ddca890>
```







8. Explainability - SHAP Feature Importance

To better understand our model's predictions, we will use **SHAP (SHapley Additive exPlanations)** to analyze feature importance.

🔷 How SHAP Works?

- SHAP assigns each feature a **contribution score** for every prediction.
- Uses **Shapley values** (from game theory) to fairly distribute importance across features.

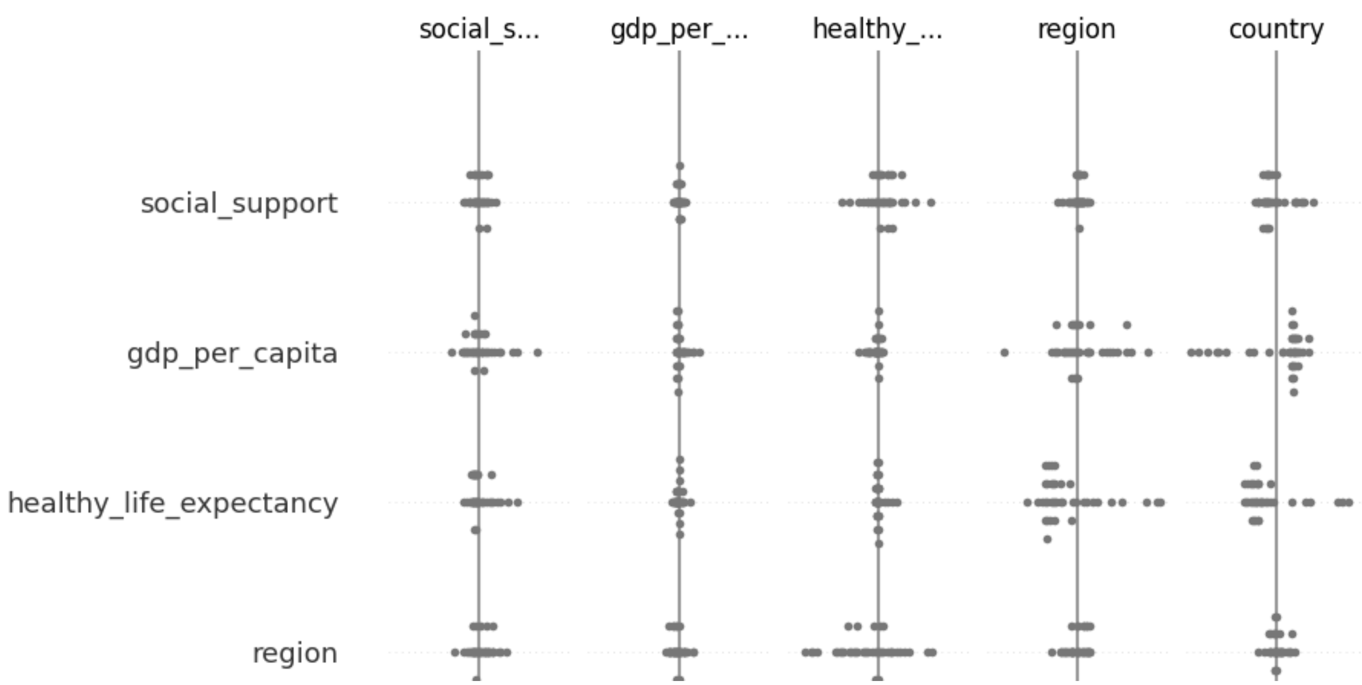
We will now apply SHAP to visualize and interpret our model's feature contributions.

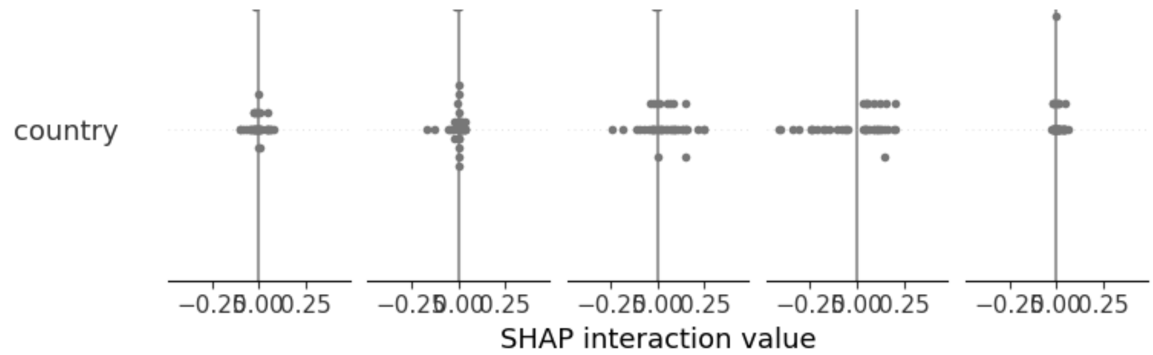
```
# Import libraries
import shap
from sklearn.impute import SimpleImputer

# SHAP Analysis:
# Create a wrapper function to handle NaN values during prediction:
def predict_wrapper(X):
    predictions = leaky_relu.predict(X)
    # Handle potential NaN values in predictions (replace with a default value, e
    predictions = np.nan_to_num(predictions)
    return predictions

# Initialize SHAP explainer using the wrapper function
explainer = shap.KernelExplainer(predict_wrapper, preprocessor(X_train))

# Compute SHAP values for X_test
shap_values = explainer.shap_values(preprocess.transform(X_test))

# Generate SHAP summary plot
shap.summary_plot(shap_values, preprocess.transform(X_test), feature_names=X_trai
```

```
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 23ms/step
100% 42/42 [30:51<00:00, 44.45s/it]
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 85ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 18s 2ms/step
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 134ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 18s 2ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 89ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 19s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 91ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 19s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 90ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 18s 2ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 146ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 18s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 89ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 18s 2ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 91ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 18s 2ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 89ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 19s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 92ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 19s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 130ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 19s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 155ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 18s 2ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 89ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 20s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 132ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 18s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 140ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 22s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 92ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 18s 2ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 92ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 18s 2ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 106ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 18s 2ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 90ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 18s 2ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 90ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 18s 2ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 167ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 19s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 90ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 18s 2ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 90ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 18s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 91ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 19s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 96ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 19s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 134ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 20s 3ms/step
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 90ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 18s 2ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 89ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 19s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 92ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 19s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 129ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 19s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 93ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 18s 2ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 140ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 18s 2ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 92ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 19s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 92ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 19s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 130ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 20s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 91ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 18s 2ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 90ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 19s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 91ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 19s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 115ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 19s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 90ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 19s 3ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 95ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 18s 2ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 89ms/step
7333/7333 ━━━━━━━━━━━━━━━━━━━━ 18s 2ms/step
```





Experimentation

```
## You are encouraged to try more experimentation and any other models by adding 

## You can also try to import any new dataset pertaining to countries, merge it, 
## If it does not, try to explain why it wasn't helpful by exploring variable rel
```

Deep learning models are often considered 'black boxes' due to their complexity. Explore methods such as SHAP (SHapley Additive exPlanations) to explain your model's predictions. After applying one of these methods, do you feel it provides a clear and sufficient explanation of how your model makes decisions? How easy or difficult is it to justify your model's predictions using these techniques?

```
## Your Code and Answer:

# Going back to keras model.
keras_model = Sequential([
    Dense(128, input_dim=feature_count, activation='relu'),
    Dense(64, activation='relu'),
    Dense(64, activation='relu'),
```

```
    Dense(32, activation='relu'),
    Dense(5, activation='softmax')
])

# Compile model
keras_model.compile(loss='categorical_crossentropy', optimizer='sgd', metrics=['a

# Convert y_train to one-hot encoding
lb = LabelBinarizer()
y_train_encoded = lb.fit_transform(y_train)

# Fitting the model to the Training set
history = keras_model.fit(preprocessor(X_train), y_train_encoded,
               batch_size = 20,
               epochs = 300, validation_split=0.25)

from sklearn.base import BaseEstimator, ClassifierMixin
from sklearn.inspection import permutation_importance

# Permutation Feature Importance Analysis with NaN Handling:

class KerasClassifierWrapper(BaseEstimator, ClassifierMixin):
    def __init__(self, keras_model, preprocessor):
        self.keras_model = keras_model
        self.preprocessor = preprocessor

    def fit(self, X, y):
        return self

# Get predicted class labels
    def predict(self, X):
        preprocessed_X = self.preprocessor(X)  # Call the function directly
        predictions = self.keras_model.predict(preprocessed_X)
        predictions = np.nan_to_num(predictions, nan=0.0)
        return predictions.argmax(axis=1)  # Return class labels

# Create an instance of the wrapper class
wrapper = KerasClassifierWrapper(keras_model, preprocessor)

# Calculate baseline accuracy
y_test_labels = y_test.astype('category').cat.codes
baseline_accuracy = accuracy_score(y_test_labels, wrapper.predict(X_test))  # Use

# Perform permutation importance using the wrapper instance
result = permutation_importance(
```

```
    estimator=wrapper,  # Use the wrapper instance
    X=X_test,  # Pass the original X_test
    y=y_test_labels,
    n_repeats=30,
    random_state=42,
    scoring='accuracy'
)

# Process and Visualize Results:

# 1. Get Feature Importances and Sort
importances = result.importances_mean
sorted_idx = importances.argsort()

# 2. Create a DataFrame for easier handling
df_importances = pd.DataFrame({
    "Feature": X_train.columns[sorted_idx],
    "Importance": importances[sorted_idx]
})

# 3. Plotting
fig, ax = plt.subplots()
ax.barh(df_importances["Feature"], df_importances["Importance"])
ax.set_title("Permutation Feature Importance")
ax.set_xlabel("Importance")
plt.show()

# 4. Display the DataFrame
print(df_importances)
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87: Use
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Epoch 1/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 2s 252ms/step - accuracy: 0.2251 - loss: 1.6046 - val
Epoch 2/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 137ms/step - accuracy: 0.2118 - loss: 1.6004 - val
Epoch 3/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 84ms/step - accuracy: 0.2335 - loss: 1.5757 - val_
Epoch 4/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 169ms/step - accuracy: 0.2151 - loss: 1.5849 - val
Epoch 5/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 44ms/step - accuracy: 0.2185 - loss: 1.5699 - val_
Epoch 6/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2551 - loss: 1.5494 - val_
Epoch 7/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.1618 - loss: 1.5834 - val_
Epoch 8/300
```

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.2168 - loss: 1.5449 - val_
Epoch 9/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.2251 - loss: 1.5346 - val_
Epoch 10/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.2401 - loss: 1.5288 - val_
Epoch 11/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.1918 - loss: 1.5349 - val_
Epoch 12/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.2345 - loss: 1.5040 - val_
Epoch 13/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.2051 - loss: 1.5085 - val_
Epoch 14/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 0.2201 - loss: 1.5002 - val_
Epoch 15/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.2018 - loss: 1.4959 - val_
Epoch 16/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2018 - loss: 1.4996 - val_
Epoch 17/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.1974 - loss: 1.4904 - val_
Epoch 18/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 0.2291 - loss: 1.4642 - val_
Epoch 19/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 0.2037 - loss: 1.5011 - val_
Epoch 20/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.1960 - loss: 1.4844 - val_
Epoch 21/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2377 - loss: 1.4582 - val_
Epoch 22/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.2777 - loss: 1.4199 - val_
Epoch 23/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.3316 - loss: 1.3956 - val_
Epoch 24/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 0.3250 - loss: 1.3862 - val_
Epoch 25/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.3029 - loss: 1.4168 - val_
Epoch 26/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.3442 - loss: 1.4174 - val_
Epoch 27/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.3258 - loss: 1.4073 - val_
Epoch 28/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 0.3175 - loss: 1.4044 - val_
Epoch 29/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.3631 - loss: 1.3796 - val_
Epoch 30/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.3738 - loss: 1.3737 - val_
Epoch 31/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 79ms/step - accuracy: 0.3654 - loss: 1.3764 - val_
Epoch 32/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 91ms/step - accuracy: 0.4400 - loss: 1.3491 - val_
Epoch 33/300
```

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 96ms/step - accuracy: 0.4034 - loss: 1.3352 - val_
Epoch 34/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 91ms/step - accuracy: 0.4007 - loss: 1.3374 - val_
Epoch 35/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step - accuracy: 0.4390 - loss: 1.3037 - val_
Epoch 36/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 70ms/step - accuracy: 0.4163 - loss: 1.3374 - val_
Epoch 37/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 67ms/step - accuracy: 0.4513 - loss: 1.3027 - val_
Epoch 38/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 93ms/step - accuracy: 0.4069 - loss: 1.3533 - val_
Epoch 39/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 92ms/step - accuracy: 0.4280 - loss: 1.2871 - val_
Epoch 40/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.4042 - loss: 1.2911 - val_
Epoch 41/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.4715 - loss: 1.3040 - val_
Epoch 42/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.4415 - loss: 1.3103 - val_
Epoch 43/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.4999 - loss: 1.2996 - val_
Epoch 44/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.5049 - loss: 1.2439 - val_
Epoch 45/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.5618 - loss: 1.2547 - val_
Epoch 46/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.5760 - loss: 1.2311 - val_
Epoch 47/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.5241 - loss: 1.2141 - val_
Epoch 48/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.5251 - loss: 1.2456 - val_
Epoch 49/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.5178 - loss: 1.2761 - val_
Epoch 50/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.4878 - loss: 1.2137 - val_
Epoch 51/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.4945 - loss: 1.2496 - val_
Epoch 52/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.6446 - loss: 1.1715 - val_
Epoch 53/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.5420 - loss: 1.2274 - val_
Epoch 54/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.5391 - loss: 1.1929 - val_
Epoch 55/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.5241 - loss: 1.2006 - val_
Epoch 56/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.6200 - loss: 1.1461 - val_
Epoch 57/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.5520 - loss: 1.1680 - val_
Epoch 58/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.6016 - loss: 1.1517 - val_
```

```
Epoch 59/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.5723 - loss: 1.1884 - val_
Epoch 60/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.5920 - loss: 1.1315 - val_
Epoch 61/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.6042 - loss: 1.1632 - val_
Epoch 62/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.5956 - loss: 1.1719 - val_
Epoch 63/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.6058 - loss: 1.1325 - val_
Epoch 64/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.6139 - loss: 1.1527 - val_
Epoch 65/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.6448 - loss: 1.1453 - val_
Epoch 66/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.5823 - loss: 1.1409 - val_
Epoch 67/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.6742 - loss: 1.1051 - val_
Epoch 68/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.6865 - loss: 1.0979 - val_
Epoch 69/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.5866 - loss: 1.0982 - val_
Epoch 70/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.6625 - loss: 1.0654 - val_
Epoch 71/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.6998 - loss: 1.0596 - val_
Epoch 72/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 101ms/step - accuracy: 0.6573 - loss: 1.0710 - val
Epoch 73/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 65ms/step - accuracy: 0.6631 - loss: 1.0722 - val_
Epoch 74/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 64ms/step - accuracy: 0.6911 - loss: 1.0876 - val_
Epoch 75/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 93ms/step - accuracy: 0.6973 - loss: 1.0757 - val_
Epoch 76/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 92ms/step - accuracy: 0.7350 - loss: 1.0674 - val_
Epoch 77/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 65ms/step - accuracy: 0.7267 - loss: 1.0431 - val_
Epoch 78/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 107ms/step - accuracy: 0.7523 - loss: 0.9867 - val
Epoch 79/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 96ms/step - accuracy: 0.6911 - loss: 1.0287 - val_
Epoch 80/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 95ms/step - accuracy: 0.6771 - loss: 1.0222 - val_
Epoch 81/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 107ms/step - accuracy: 0.6884 - loss: 1.0398 - val
Epoch 82/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 95ms/step - accuracy: 0.6923 - loss: 1.0351 - val_
Epoch 83/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 65ms/step - accuracy: 0.6834 - loss: 1.0144 - val_
```

```
Epoch 84/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 93ms/step - accuracy: 0.7457 - loss: 0.9894 - val_
Epoch 85/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 91ms/step - accuracy: 0.7267 - loss: 0.9605 - val_
Epoch 86/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 94ms/step - accuracy: 0.6767 - loss: 1.0360 - val_
Epoch 87/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 72ms/step - accuracy: 0.7117 - loss: 0.9743 - val_
Epoch 88/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 104ms/step - accuracy: 0.7413 - loss: 0.9925 - val
Epoch 89/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.6723 - loss: 0.9904 - val_
Epoch 90/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 0.7396 - loss: 0.9379 - val_
Epoch 91/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7723 - loss: 0.9072 - val_
Epoch 92/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7503 - loss: 0.9272 - val_
Epoch 93/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.7396 - loss: 0.9313 - val_
Epoch 94/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7486 - loss: 0.9097 - val_
Epoch 95/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 0.7019 - loss: 0.9826 - val_
Epoch 96/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step - accuracy: 0.7140 - loss: 0.9215 - val_
Epoch 97/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.7369 - loss: 0.8976 - val_
Epoch 98/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7653 - loss: 0.8890 - val_
Epoch 99/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7507 - loss: 0.8960 - val_
Epoch 100/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.7663 - loss: 0.8719 - val_
Epoch 101/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.7513 - loss: 0.8697 - val_
Epoch 102/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.7403 - loss: 0.8710 - val_
Epoch 103/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.7280 - loss: 0.9002 - val_
Epoch 104/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.7553 - loss: 0.8565 - val_
Epoch 105/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.7519 - loss: 0.8346 - val_
Epoch 106/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7653 - loss: 0.8421 - val_
Epoch 107/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.7419 - loss: 0.8789 - val_
Epoch 108/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.7436 - loss: 0.8362 - val_
Epoch 109/300
```

```
Epoch 109/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.7769 - loss: 0.8341 - val_
Epoch 110/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.7296 - loss: 0.8503 - val_
Epoch 111/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.7213 - loss: 0.8328 - val_
Epoch 112/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.7253 - loss: 0.8524 - val_
Epoch 113/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7513 - loss: 0.8025 - val_
Epoch 114/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.7323 - loss: 0.8159 - val_
Epoch 115/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7553 - loss: 0.7808 - val_
Epoch 116/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7219 - loss: 0.8104 - val_
Epoch 117/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7742 - loss: 0.7642 - val_
Epoch 118/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.7786 - loss: 0.7668 - val_
Epoch 119/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.7726 - loss: 0.7668 - val_
Epoch 120/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.7619 - loss: 0.7363 - val_
Epoch 121/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.7265 - loss: 0.8135 - val_
Epoch 122/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7742 - loss: 0.7426 - val_
Epoch 123/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.7742 - loss: 0.7298 - val_
Epoch 124/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.7576 - loss: 0.7637 - val_
Epoch 125/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.7426 - loss: 0.7583 - val_
Epoch 126/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.7453 - loss: 0.7162 - val_
Epoch 127/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7869 - loss: 0.6902 - val_
Epoch 128/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.7632 - loss: 0.7262 - val_
Epoch 129/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 65ms/step - accuracy: 0.7849 - loss: 0.7055 - val_
Epoch 130/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 64ms/step - accuracy: 0.7815 - loss: 0.7057 - val_
Epoch 131/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 92ms/step - accuracy: 0.7945 - loss: 0.7233 - val_
Epoch 132/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 94ms/step - accuracy: 0.8145 - loss: 0.6603 - val_
Epoch 133/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 97ms/step - accuracy: 0.8218 - loss: 0.7168 - val_
Epoch 134/300
```

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 95ms/step - accuracy: 0.7628 - loss: 0.7069 - val_
Epoch 135/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 68ms/step - accuracy: 0.8178 - loss: 0.6447 - val_
Epoch 136/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.7895 - loss: 0.7222 - val_
Epoch 137/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.7928 - loss: 0.6891 - val_
Epoch 138/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.8191 - loss: 0.6790 - val_
Epoch 139/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.8334 - loss: 0.6791 - val_
Epoch 140/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.8480 - loss: 0.6517 - val_
Epoch 141/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.8341 - loss: 0.6733 - val_
Epoch 142/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.8607 - loss: 0.6437 - val_
Epoch 143/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.8247 - loss: 0.6695 - val_
Epoch 144/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.8797 - loss: 0.6038 - val_
Epoch 145/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.8364 - loss: 0.6313 - val_
Epoch 146/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.8887 - loss: 0.6023 - val_
Epoch 147/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.8737 - loss: 0.6195 - val_
Epoch 148/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.8214 - loss: 0.6260 - val_
Epoch 149/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.8520 - loss: 0.6557 - val_
Epoch 150/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.8437 - loss: 0.6346 - val_
Epoch 151/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.8687 - loss: 0.6217 - val_
Epoch 152/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.8687 - loss: 0.6184 - val_
Epoch 153/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.8653 - loss: 0.6133 - val_
Epoch 154/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.8970 - loss: 0.5785 - val_
Epoch 155/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.8643 - loss: 0.6066 - val_
Epoch 156/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.8610 - loss: 0.5841 - val_
Epoch 157/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.8876 - loss: 0.6011 - val_
Epoch 158/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.8420 - loss: 0.5878 - val_
Epoch 159/300
```

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.9116 - loss: 0.5458 - val_
Epoch 160/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.8860 - loss: 0.5264 - val_
Epoch 161/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.8876 - loss: 0.5608 - val_
Epoch 162/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9066 - loss: 0.5506 - val_
Epoch 163/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.8783 - loss: 0.5356 - val_
Epoch 164/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.8999 - loss: 0.5347 - val_
Epoch 165/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9256 - loss: 0.5262 - val_
Epoch 166/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.8899 - loss: 0.5385 - val_
Epoch 167/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.8916 - loss: 0.5418 - val_
Epoch 168/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step - accuracy: 0.8916 - loss: 0.5335 - val_
Epoch 169/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9022 - loss: 0.5198 - val_
Epoch 170/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.9049 - loss: 0.5127 - val_
Epoch 171/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.9372 - loss: 0.4734 - val_
Epoch 172/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.8989 - loss: 0.4933 - val_
Epoch 173/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9016 - loss: 0.5484 - val_
Epoch 174/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.8822 - loss: 0.4920 - val_
Epoch 175/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 100ms/step - accuracy: 0.9006 - loss: 0.4911 - val
Epoch 176/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 91ms/step - accuracy: 0.8799 - loss: 0.4844 - val_
Epoch 177/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 80ms/step - accuracy: 0.8333 - loss: 0.5262 - val_
Epoch 178/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 67ms/step - accuracy: 0.9056 - loss: 0.4845 - val_
Epoch 179/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 66ms/step - accuracy: 0.9106 - loss: 0.5037 - val_
Epoch 180/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 101ms/step - accuracy: 0.8822 - loss: 0.4926 - val
Epoch 181/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 94ms/step - accuracy: 0.9066 - loss: 0.4852 - val_
Epoch 182/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.8906 - loss: 0.5142 - val_
Epoch 183/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9089 - loss: 0.4771 - val_
Epoch 184/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9145 - loss: 0.4873 - val_
```

```
Epoch 185/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9056 - loss: 0.4625 - val_
Epoch 186/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.8922 - loss: 0.4902 - val_
Epoch 187/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.8772 - loss: 0.4510 - val_
Epoch 188/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.9295 - loss: 0.4735 - val_
Epoch 189/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9189 - loss: 0.4296 - val_
Epoch 190/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step - accuracy: 0.9145 - loss: 0.4407 - val_
Epoch 191/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.9145 - loss: 0.4443 - val_
Epoch 192/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9279 - loss: 0.4359 - val_
Epoch 193/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.8845 - loss: 0.4632 - val_
Epoch 194/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.8956 - loss: 0.4262 - val_
Epoch 195/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.8722 - loss: 0.4437 - val_
Epoch 196/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.8995 - loss: 0.4096 - val_
Epoch 197/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.9095 - loss: 0.4112 - val_
Epoch 198/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 0.9045 - loss: 0.4428 - val_
Epoch 199/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.9012 - loss: 0.4175 - val_
Epoch 200/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9112 - loss: 0.4347 - val_
Epoch 201/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.9095 - loss: 0.3877 - val_
Epoch 202/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.8912 - loss: 0.4108 - val_
Epoch 203/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9145 - loss: 0.4277 - val_
Epoch 204/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9362 - loss: 0.3375 - val_
Epoch 205/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9012 - loss: 0.3875 - val_
Epoch 206/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.9179 - loss: 0.3740 - val_
Epoch 207/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.8962 - loss: 0.3788 - val_
Epoch 208/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9245 - loss: 0.3780 - val_
Epoch 209/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9095 - loss: 0.3834 - val_
```

```
Epoch 210/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 0.9379 - loss: 0.3699 - val_
Epoch 211/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9395 - loss: 0.3044 - val_
Epoch 212/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.9412 - loss: 0.3339 - val_
Epoch 213/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9329 - loss: 0.3663 - val_
Epoch 214/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.9395 - loss: 0.3248 - val_
Epoch 215/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.9385 - loss: 0.3570 - val_
Epoch 216/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step - accuracy: 0.9491 - loss: 0.3250 - val_
Epoch 217/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9475 - loss: 0.3400 - val_
Epoch 218/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9291 - loss: 0.3568 - val_
Epoch 219/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9368 - loss: 0.3206 - val_
Epoch 220/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step - accuracy: 0.9475 - loss: 0.3475 - val_
Epoch 221/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9564 - loss: 0.3215 - val_
Epoch 222/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9681 - loss: 0.2974 - val_
Epoch 223/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 86ms/step - accuracy: 0.9614 - loss: 0.3066 - val_
Epoch 224/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 71ms/step - accuracy: 0.9341 - loss: 0.3184 - val_
Epoch 225/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 72ms/step - accuracy: 0.9581 - loss: 0.2848 - val_
Epoch 226/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.9764 - loss: 0.2660 - val_
Epoch 227/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 93ms/step - accuracy: 0.9481 - loss: 0.3278 - val_
Epoch 228/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 68ms/step - accuracy: 0.9748 - loss: 0.2707 - val_
Epoch 229/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 64ms/step - accuracy: 0.9481 - loss: 0.2980 - val_
Epoch 230/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 67ms/step - accuracy: 0.9581 - loss: 0.2784 - val_
Epoch 231/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 67ms/step - accuracy: 0.9531 - loss: 0.2812 - val_
Epoch 232/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 70ms/step - accuracy: 0.9431 - loss: 0.3019 - val_
Epoch 233/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.9564 - loss: 0.2660 - val_
Epoch 234/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 91ms/step - accuracy: 0.9631 - loss: 0.2838 - val_
Epoch 235/300
```

```
Epoch 235/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 53ms/step - accuracy: 0.9281 - loss: 0.3071 - val_
Epoch 236/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9714 - loss: 0.2379 - val_
Epoch 237/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.9664 - loss: 0.2478 - val_
Epoch 238/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9431 - loss: 0.2807 - val_
Epoch 239/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9714 - loss: 0.2374 - val_
Epoch 240/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9581 - loss: 0.2550 - val_
Epoch 241/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9531 - loss: 0.2569 - val_
Epoch 242/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 0.9531 - loss: 0.2566 - val_
Epoch 243/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9771 - loss: 0.2335 - val_
Epoch 244/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.9621 - loss: 0.2854 - val_
Epoch 245/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9804 - loss: 0.2021 - val_
Epoch 246/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.9910 - loss: 0.2394 - val_
Epoch 247/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9721 - loss: 0.2464 - val_
Epoch 248/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 0.9854 - loss: 0.2245 - val_
Epoch 249/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.9944 - loss: 0.2519 - val_
Epoch 250/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step - accuracy: 0.9910 - loss: 0.2027 - val_
Epoch 251/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 0.9910 - loss: 0.2251 - val_
Epoch 252/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.9760 - loss: 0.2213 - val_
Epoch 253/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9860 - loss: 0.2067 - val_
Epoch 254/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 0.9910 - loss: 0.1986 - val_
Epoch 255/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9760 - loss: 0.2176 - val_
Epoch 256/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9860 - loss: 0.2049 - val_
Epoch 257/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 0.9910 - loss: 0.1922 - val_
Epoch 258/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 0.9910 - loss: 0.2004 - val_
Epoch 259/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.1889 - val_
Epoch 260/300
```

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 1.0000 - loss: 0.2016 - val_
Epoch 261/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.1900 - val_
Epoch 262/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 1.0000 - loss: 0.1941 - val_
Epoch 263/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.1650 - val_
Epoch 264/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.2026 - val_
Epoch 265/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.1834 - val_
Epoch 266/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.1822 - val_
Epoch 267/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.1825 - val_
Epoch 268/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.1883 - val_
Epoch 269/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 1.0000 - loss: 0.1696 - val_
Epoch 270/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.1758 - val_
Epoch 271/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.1741 - val_
Epoch 272/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.1726 - val_
Epoch 273/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 1.0000 - loss: 0.1679 - val_
Epoch 274/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step - accuracy: 1.0000 - loss: 0.1669 - val_
Epoch 275/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 97ms/step - accuracy: 1.0000 - loss: 0.1730 - val_
Epoch 276/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 92ms/step - accuracy: 1.0000 - loss: 0.1529 - val_
Epoch 277/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 93ms/step - accuracy: 1.0000 - loss: 0.1673 - val_
Epoch 278/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 95ms/step - accuracy: 1.0000 - loss: 0.1640 - val_
Epoch 279/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 94ms/step - accuracy: 1.0000 - loss: 0.1459 - val_
Epoch 280/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 89ms/step - accuracy: 1.0000 - loss: 0.1587 - val_
Epoch 281/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 75ms/step - accuracy: 1.0000 - loss: 0.1576 - val_
Epoch 282/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 46ms/step - accuracy: 1.0000 - loss: 0.1305 - val_
Epoch 283/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 1.0000 - loss: 0.1329 - val_
Epoch 284/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 1.0000 - loss: 0.1328 - val_
Epoch 285/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 1.0000 - loss: 0.1429 - val_
```

```
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 1.0000 - loss: 0.1439 - val_
Epoch 286/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.1342 - val_
Epoch 287/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step - accuracy: 1.0000 - loss: 0.1397 - val_
Epoch 288/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 1.0000 - loss: 0.1366 - val_
Epoch 289/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step - accuracy: 1.0000 - loss: 0.1186 - val_
Epoch 290/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.1198 - val_
Epoch 291/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step - accuracy: 1.0000 - loss: 0.1194 - val_
Epoch 292/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.1197 - val_
Epoch 293/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step - accuracy: 1.0000 - loss: 0.1309 - val_
Epoch 294/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.1184 - val_
Epoch 295/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step - accuracy: 1.0000 - loss: 0.1175 - val_
Epoch 296/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 1.0000 - loss: 0.1238 - val_
Epoch 297/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.1155 - val_
Epoch 298/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step - accuracy: 1.0000 - loss: 0.1082 - val_
Epoch 299/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step - accuracy: 1.0000 - loss: 0.1262 - val_
Epoch 300/300
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step - accuracy: 1.0000 - loss: 0.1102 - val_
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 70ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 58ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 67ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 66ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 70ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 71ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 70ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 62ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 69ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 68ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 72ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 96ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 69ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 1s 217ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 183ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 122ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 84ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 124ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 140ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 138ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 71ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 64ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 95ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 137ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 1s 319ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 81ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 69ms/step
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 70ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 80ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 67ms/step
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 82ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 71ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 65ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 82ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 78ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 42ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 71ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 66ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 85ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 70ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 71ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 80ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 62ms/step
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 62ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 67ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 68ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 72ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 57ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 62ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 62ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 70ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 62ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 80ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 83ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 53ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 65ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 78ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 71ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 72ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 64ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 65ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 72ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 64ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 58ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
```

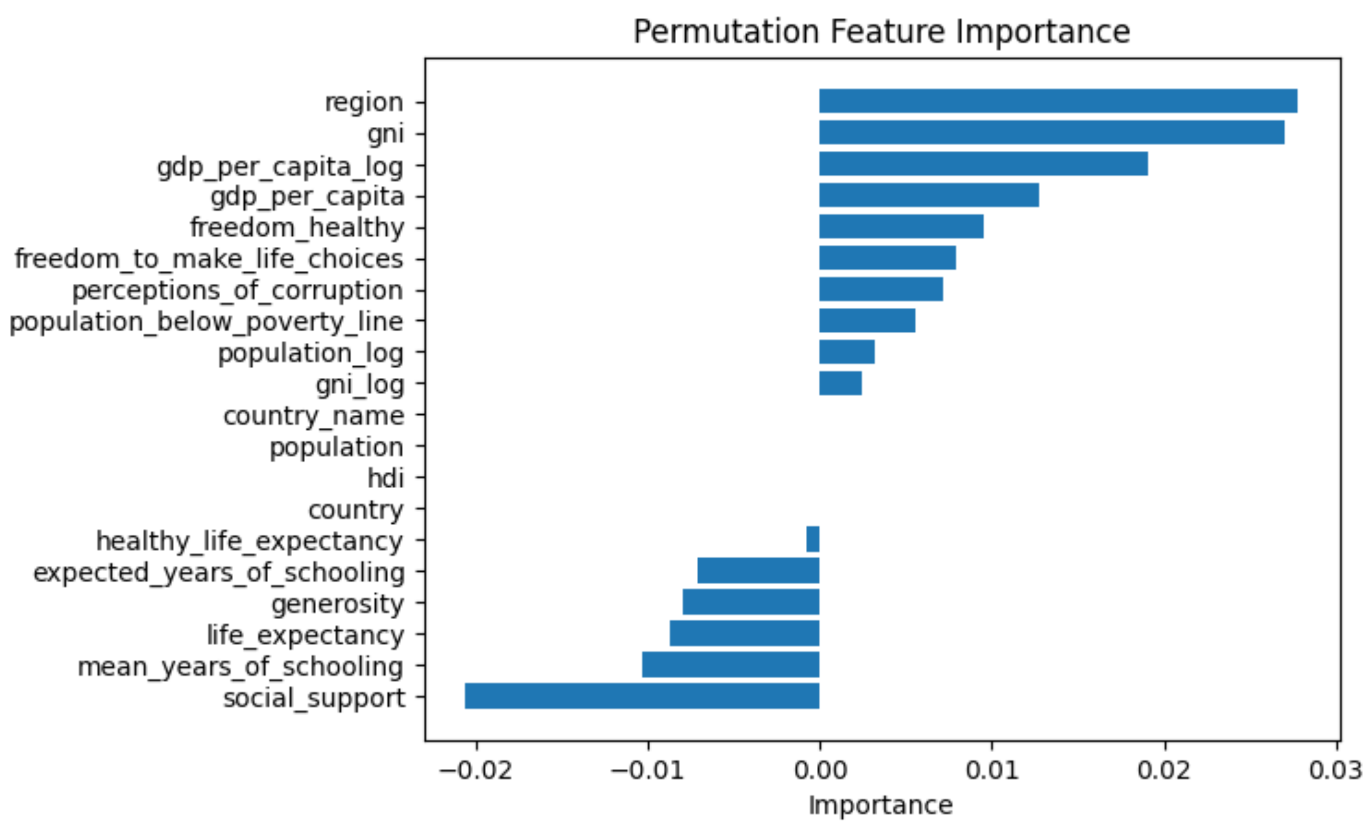
```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 87ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 83ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 72ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 68ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 66ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 81ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 80ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 69ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 71ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 61ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 70ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 58ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 142ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 84ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 68ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 62ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 68ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 71ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 71ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 80ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 79ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 43ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 54ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 201ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 72ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 71ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 80ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 79ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 67ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 82ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 66ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 85ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 68ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 58ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 65ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 57ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 51ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 47ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 45ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 52ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 46ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 60ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 59ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 63ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 67ms/step
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 65ms/step
```



	Feature	Importance
0	social_support	-0.020635
1	mean_years_of_schooling	-0.010317
2	life_expectancy	-0.008730
3	generosity	-0.007937
4	expected_years_of_schooling	-0.007143
5	healthy_life_expectancy	-0.000794
6	country	0.000000
7	hdi	0.000000
8	population	0.000000
9	country_name	0.000000
10	gni_log	0.002381
11	population_log	0.003175
12	population_below_poverty_line	0.005556
13	perceptions_of_corruption	0.007143
14	freedom_to_make_life_choices	0.007937
15	freedom_healthy	0.009524
16	gdp_per_capita	0.012698
17	gdp_per_capita_log	0.019048
18	gni	0.026984
19	region	0.027778

SHAP interaction provides shows how each variable contributes to predictions for individual instances. However, running the model took a long time, showing that it is extremely resource-intensive. We can visualize variable attributions but the the conceptual reasoning the model uses remains vague. In combination with permutation importance, we were able to identify relevant features globally to compare with the detailed per-prediction explanation from SHAP.