Rain Prediction using LR, Tree, XGBoost, NN, CNN, KNN

By: Mridul Jain, Lynne Wang, Deepak Kumar Srivastava, and Naresh Kumar Chinnathambi Kailasam

Course: MIDS Spring 2025, W207 Final Project

Objective:

The goal is to predict rainfall using historical weather data with different machine learning models: Logistic Regression, Decision Tree, Random Forest, XGBoost, and Neural Networks.

Dataset:

- Source: Kaggle Playground Series - Season 5, Episode 3. https://www.kaggle.com/competitions/playground-series-s5e3
- Nature: Tabular weather dataset with features like temperature, humidity, wind speed, etc.
- Target: Binary classification (`0` : No Rain, `1` : Rain).

Data Preprocessing & EDA

- **Dataset**: Contains 2,190 training records with features like pressure, temperature (max, min, average), humidity, dewpoint, wind direction/speed, sunshine, and cloud cover.
- **Transformations**:
  - Cyclical encoding for `day` and `wind direction` .
  - Added synthetic `date` , `year` , and `month` .
  - Visualized rainy days per month/year.
  - Correlation heatmap constructed.

Models Evaluated:

1. **Logistic Regression (LR)**
2. **K-Nearest Neighbors (KNN)**
3. **Decision Tree**
4. **Random Forest Tree**
5. **XGBoost**
6. **Neural Network (NN)**
7. **1D Convolutional Neural Network (CNN)**
8. **2D Convolutional Neural Network (CNN)**

Model Performance:

Model	Validation Accuracy	Test Score
Logistic Regression	0.8837	~0.89610
KNN	0.8744	~0.87163
Decision Tree	0.8676	~0.85561
Random Forest	0.8744	~0.89578
XGBoost	0.8721	~0.89975
Neural Network	0.8790	~0.90145
1D CNN	0.8833	~0.89162
2D CNN	0.8787	~0.88792
Ensemble		~0.90376

Outcome:

- The top leaderboard score was 0.90654.
- Our ensemble achieved a score of 0.90376, which would have ranked 21st out of 4,382 participants.

Conclusion:

All models achieved relatively high validation accuracy, with Neural Networks and Random Forest slightly outperforming others. The ensemble-style or deep learning-based approaches seem most promising for future refinement.

In [257…

```
import numpy as np
from matplotlib import pyplot as plt
import pandas as pd
import seaborn as sns  # for nicer plots
sns.set(style="darkgrid")  # default style

from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
import matplotlib.pyplot as plt

from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import mean_squared_error

import tensorflow as tf
from tensorflow import keras
from keras import metrics
from keras.datasets import fashion_mnist
from tensorflow.keras.models import load_model
import pandas as pd
from sklearn.preprocessing import StandardScaler
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score

from xgboost import XGBClassifier
from sklearn import metrics

import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers

import glob

from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.
mount("/content/drive", force_remount=True).
```

EDA & Data Preprocessing

In [258…

```
# Load the training data
PATH = '/content/drive/My Drive/Colab Notebooks/207 Final/'
train_df = pd.read_csv(PATH+'train.csv')
test_df = pd.read_csv(PATH+'test.csv')
test_df2 = pd.read_csv(PATH+'test_extra7.csv')
# Show basic info and first few rows
train_info = train_df.info()
train_head = train_df.head()

train_info, train_head
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2190 entries, 0 to 2189
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   id             2190 non-null   int64  
 1   day            2190 non-null   int64  
 2   pressure       2190 non-null   float64
 3   maxtemp        2190 non-null   float64
 4   temparature    2190 non-null   float64
 5   mintemp        2190 non-null   float64
 6   dewpoint       2190 non-null   float64
 7   humidity       2190 non-null   float64
 8   cloud          2190 non-null   float64
 9   sunshine       2190 non-null   float64
 10  winddirection  2190 non-null   float64
 11  windspeed      2190 non-null   float64
 12  rainfall       2190 non-null   int64  
dtypes: float64(10), int64(3)
memory usage: 222.6 KB
```

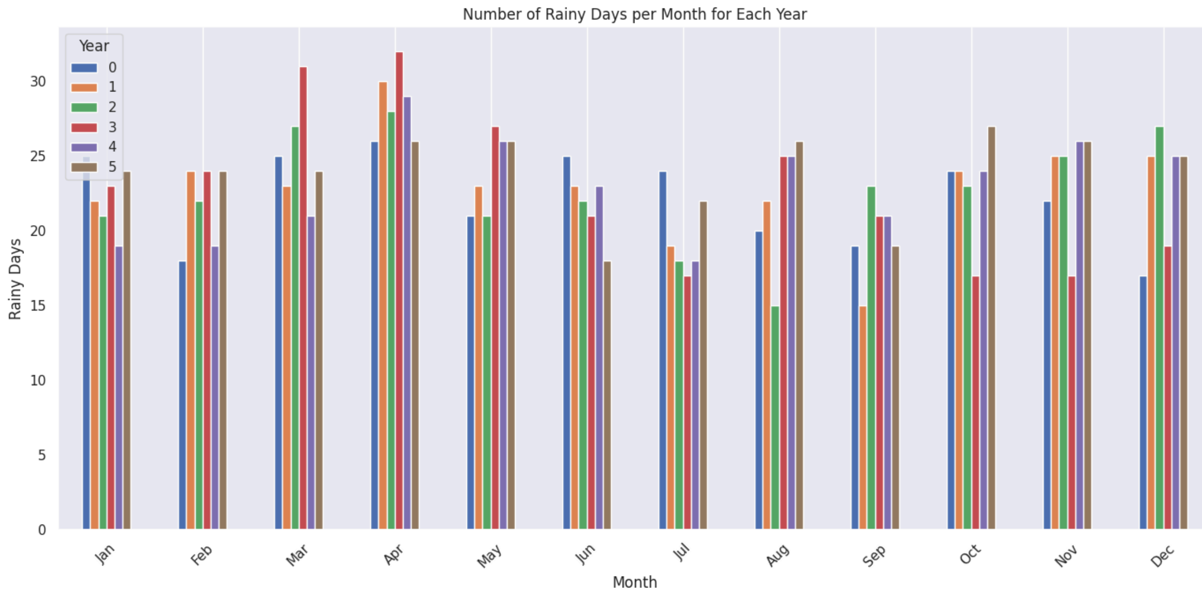
Out[258…

```
(None,
    id  day  pressure  maxtemp  temparature  mintemp  dewpoint  humidity  \
 0   0    1    1017.4     21.2         20.6     19.9      19.4      87.0
 1   1    2    1019.5     16.2         16.9     15.8      15.4      95.0
 2   2    3    1024.1     19.4         16.1     14.6       9.3      75.0
 3   3    4    1013.4     18.1         17.8     16.9      16.8      95.0
 4   4    5    1021.8     21.3         18.4     15.2       9.6      52.0

    cloud  sunshine  winddirection  windspeed  rainfall
 0   88.0       1.1           60.0       17.2         1
 1   91.0       0.0           50.0       21.9         1
 2   47.0       8.3           70.0       18.1         1
 3   95.0       0.0           60.0       35.6         1
 4   45.0       3.6           40.0       24.8         0  )
```

In [259…

```
# Create a synthetic date using day of year assuming year 2000 as base (for day to
train_df['date'] = pd.to_datetime(train_df['day'], format='%j', errors='coerce')

# Add synthetic year for splitting into multiple years (e.g., assume 6 years total
num_years = 6
train_df['year'] = (train_df.index // 365)
train_df['month'] = train_df['date'].dt.month

# Group by year and month to count rainy days
rainy_days = train_df[train_df['rainfall'] == 1].groupby(['year', 'month']).size().

# Plot
plt.figure(figsize=(12, 7))
rainy_days.T.plot(kind='bar', figsize=(14, 7))
plt.title('Number of Rainy Days per Month for Each Year')
plt.xlabel('Month')
plt.ylabel('Rainy Days')
plt.xticks(ticks=range(0, 12), labels=[
    'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
    'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'], rotation=45)
plt.legend(title='Year')
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```

```
<Figure size 1200x700 with 0 Axes>
```



In [260…

```
# Transform 'day' feature
day_frac = (train_df['day'] - 1) / 365  # range: 0 to ~1
day_radians = 2 * np.pi * day_frac
train_df['day_sin'] = np.sin(day_radians)

# Transform 'winddirection' feature
wind_radians = 2 * np.pi * train_df['winddirection'] / 360
train_df['wind_sin'] = np.sin(wind_radians)

train_info = train_df.info()
train_head = train_df.head()

train_info, train_head
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2190 entries, 0 to 2189
Data columns (total 18 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   id             2190 non-null   int64
 1   day            2190 non-null   int64
 2   pressure       2190 non-null   float64
 3   maxtemp        2190 non-null   float64
 4   temparature    2190 non-null   float64
 5   mintemp        2190 non-null   float64
 6   dewpoint       2190 non-null   float64
 7   humidity       2190 non-null   float64
 8   cloud          2190 non-null   float64
 9   sunshine       2190 non-null   float64
 10  winddirection  2190 non-null   float64
 11  windspeed      2190 non-null   float64
 12  rainfall       2190 non-null   int64
 13  date           2190 non-null   datetime64[ns]
 14  year           2190 non-null   int64
 15  month          2190 non-null   int32
 16  day_sin        2190 non-null   float64
 17  wind_sin       2190 non-null   float64
dtypes: datetime64[ns](1), float64(12), int32(1), int64(4)
memory usage: 299.5 KB
```

Out[260…

```
(None,
    id  day  pressure  maxtemp  temparature  mintemp  dewpoint  humidity  \
 0   0    1    1017.4     21.2         20.6     19.9      19.4      87.0
 1   1    2    1019.5     16.2         16.9     15.8      15.4      95.0
 2   2    3    1024.1     19.4         16.1     14.6       9.3      75.0
 3   3    4    1013.4     18.1         17.8     16.9      16.8      95.0
 4   4    5    1021.8     21.3         18.4     15.2       9.6      52.0

    cloud  sunshine  winddirection  windspeed  rainfall       date  year  \
 0   88.0       1.1           60.0       17.2         1 1900-01-01     0
 1   91.0       0.0           50.0       21.9         1 1900-01-02     0
 2   47.0       8.3           70.0       18.1         1 1900-01-03     0
 3   95.0       0.0           60.0       35.6         1 1900-01-04     0
 4   45.0       3.6           40.0       24.8         0 1900-01-05     0

    month   day_sin  wind_sin
 0      1  0.000000  0.866025
 1      1  0.017213  0.766044
 2      1  0.034422  0.939693
 3      1  0.051620  0.866025
 4      1  0.068802  0.642788  )
```

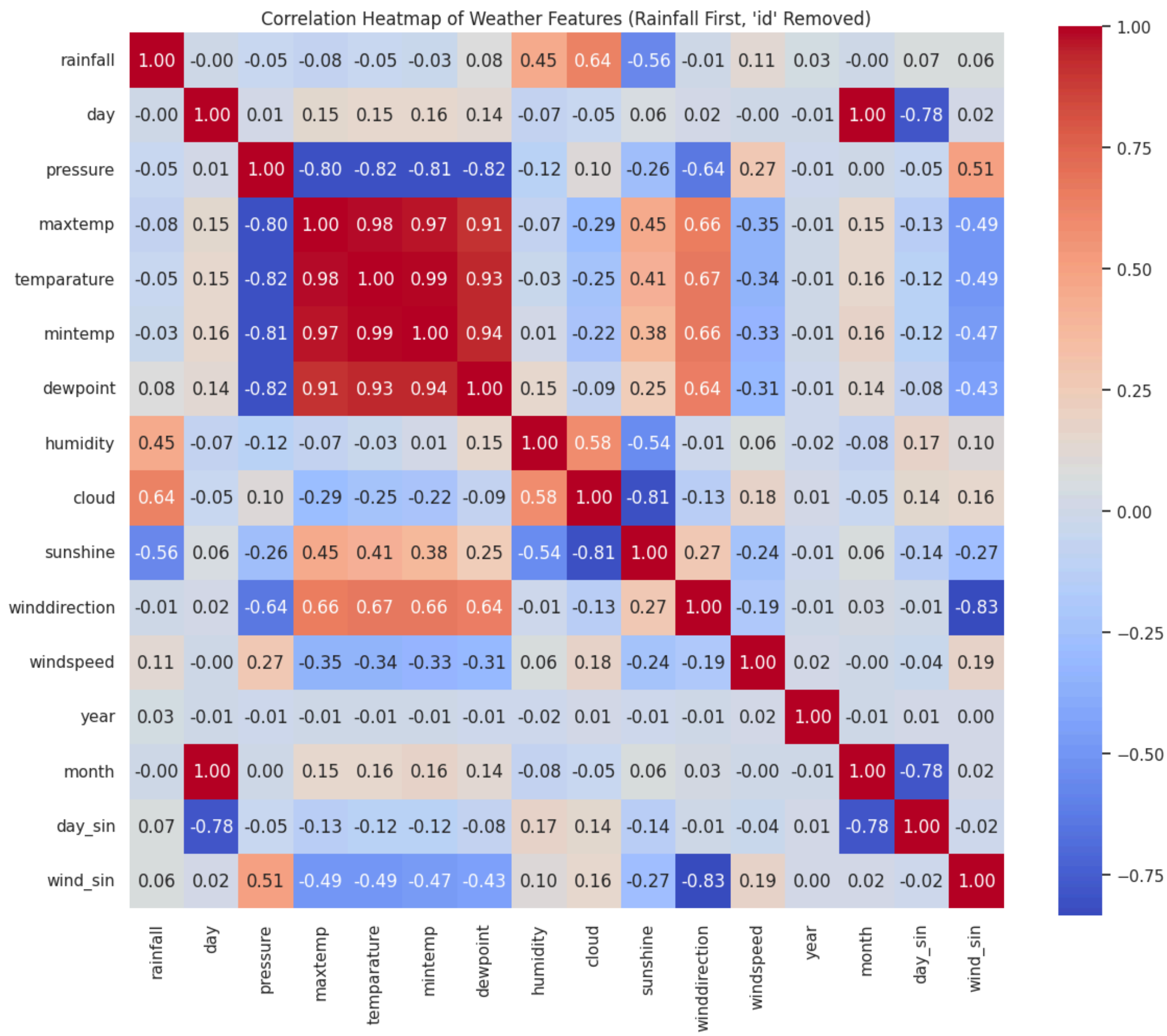
In [261…

```
# # Add a previous day's feature

# # Create lag features for the previous 2 days for selected columns
# lag_features = [
#     "pressure", "maxtemp", "temparature", "mintemp", "dewpoint",
#     "humidity", "cloud", "sunshine", "windspeed", "day_sin", "wind_sin"
# ]

# # Generate lag features for day -1 and day -2
```

```
# for lag in [1]:
#     for col in lag_features:
#         train_df[f"{col}_prev_{lag}"] = train_df[col].shift(lag)

# # Drop rows with NaNs introduced by shifting
# train_df = train_df.dropna().reset_index(drop=True)

# train_df
```

In [262…

```
import seaborn as sns
import matplotlib.pyplot as plt

# Calculate correlation matrix without 'id'
correlation_matrix = train_df.drop(columns='id').corr(numeric_only=True)

# Move 'rainfall' to the first row/column
cols = correlation_matrix.columns.tolist()
cols.insert(0, cols.pop(cols.index('rainfall')))
correlation_matrix = correlation_matrix.loc[cols, cols]

# Plot heatmap
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap="coolwarm", square=True
plt.title("Correlation Heatmap of Weather Features (Rainfall First, 'id' Removed)")
plt.tight_layout()
plt.show()
```



In [263…

```
# Drop non-feature columns and isolate target

# List of columns to keep
columns_to_keep = [
    'pressure', 'maxtemp', 'temparature', 'mintemp', 'humidity',
    'cloud', 'sunshine', 'winddirection', 'windspeed',
    'year', 'day_sin', 'wind_sin'
]

# Show the resulting DataFrame columns
train_df.columns.tolist()

X = train_df[columns_to_keep]
y = train_df['rainfall']

X.hist(bins=50, figsize=(20,15))
plt.show()
```

In [264…

```
# Scale features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Time-based split: use first 80% for training, remaining 20% for validation
split_index = int(len(X_scaled) * 0.8)
X_train, X_val = X_scaled[:split_index], X_scaled[split_index:]
y_train, y_val = y[:split_index], y[split_index:]
```

Training Models

Logistic Regression

In [265…

```
# Train logistic regression model
model_lr = LogisticRegression()
model_lr.fit(X_train, y_train)
y_train_pred = model_lr.predict(X_train)
train_acc = accuracy_score(y_train, y_train_pred)
print(f"Logistic Regression Training Accuracy: {train_acc:.4f}")

y_val_pred = model_lr.predict(X_val)
val_acc = accuracy_score(y_val, y_val_pred)
print(f"Logistic Regression validation Accuracy: {val_acc:.4f}")
#print(classification_report(y_val, y_pred))
```

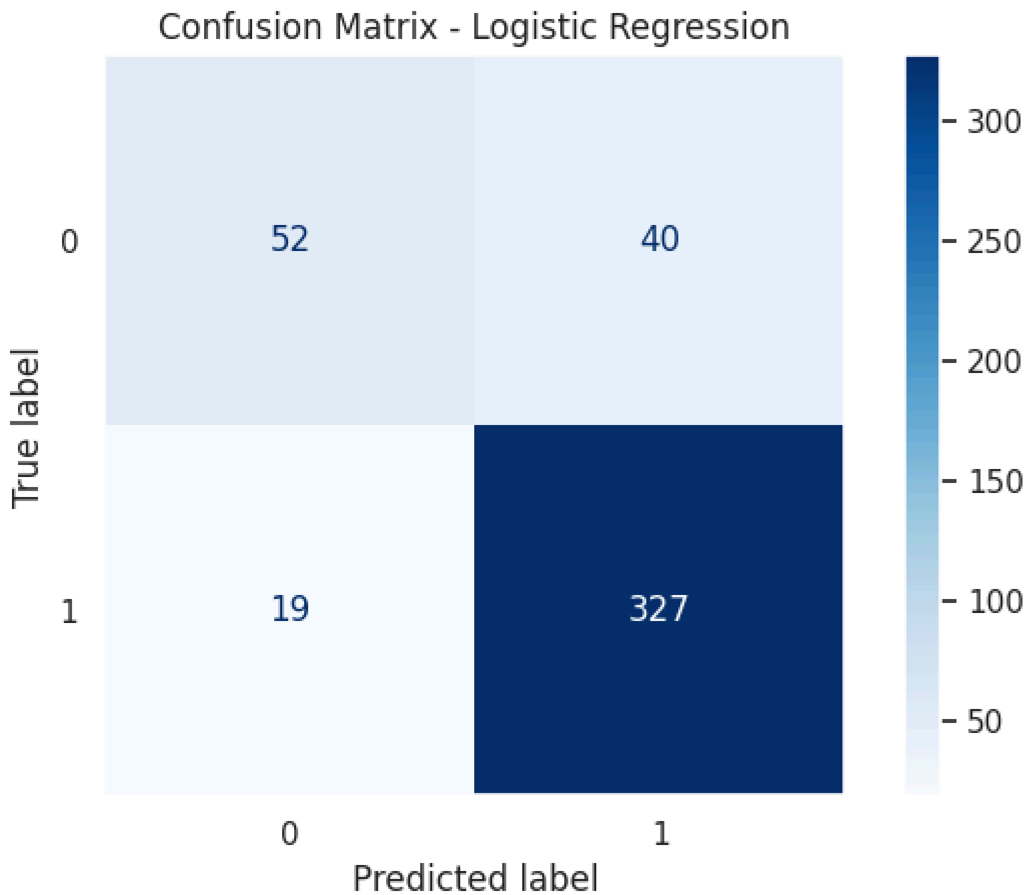
```
train_preds = model_lr.predict_proba(X_train)
print('Training Accuracy : ', metrics.roc_auc_score(y_train, train_preds[:,1]))

val_preds = model_lr.predict_proba(X_val)
print('Validation Accuracy : ', metrics.roc_auc_score(y_val, val_preds[:,1]))
print()

# Evaluate model
y_pred = model_lr.predict(X_val)
accuracy = accuracy_score(y_val, y_pred)
report = classification_report(y_val, y_pred)

print("Accuracy:", accuracy_score(y_val, y_pred))
print("\nClassification Report:\n", classification_report(y_val, y_pred))
```

```
Logistic Regression Training Accuracy: 0.8670
Logistic Regression validation Accuracy: 0.8653
Training Accuracy :  0.8972769226555654
Validation Accuracy :  0.8837961799447097

Accuracy: 0.865296803652968

Classification Report:
               precision    recall  f1-score   support

           0       0.73      0.57      0.64        92
           1       0.89      0.95      0.92       346

    accuracy                           0.87       438
   macro avg       0.81      0.76      0.78       438
weighted avg       0.86      0.87      0.86       438
```

In [266…

```
# Generate confusion matrix
cm = confusion_matrix(y_val, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=model_lr.classes_

# Plot confusion matrix
plt.figure(figsize=(6, 6))
disp.plot(cmap='Blues', values_format='d')
plt.title("Confusion Matrix - Logistic Regression")
plt.tight_layout()
plt.grid(False)
plt.show()
```

```
<Figure size 600x600 with 0 Axes>
```



In [267…

```
# Applying the trained model to test set
# Create synthetic date from 'day'
test_df['date'] = pd.to_datetime(test_df['day'], format='%j', errors='coerce')

# Simulate year assignment just like train_df (e.g., assume up to 6 years of data)
test_df['year'] = (test_df.index // 365)

# Extract month from synthetic date
test_df['month'] = test_df['date'].dt.month

# Create cyclical features
test_df['day_sin'] = np.sin(2 * np.pi * (test_df['day'] - 1) / 365)
test_df['wind_sin'] = np.sin(2 * np.pi * test_df['winddirection'] / 360)


# Select the same feature columns
X_test = test_df[columns_to_keep]

# Scale using the same scaler
X_test_scaled = scaler.transform(X_test)
print(X_test_scaled[0])

# Predict probabilities
test_probs = model_lr.predict_proba(X_test_scaled)[:, 1]  # Probability of rainfall

# Create submission DataFrame
submission = pd.DataFrame({
    'id': test_df['id'],
```

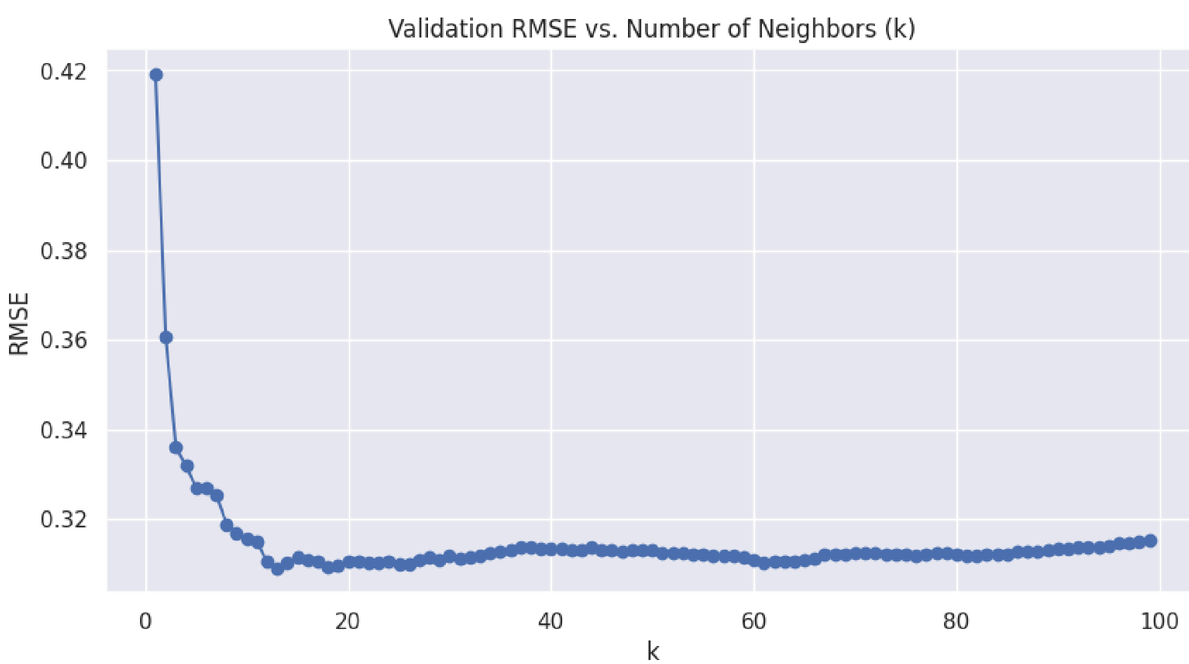
```
    'rainfall': test_probs
})

# Save to CSV
submission.to_csv(PATH + "submission_lr.csv", index=False)
```

```
[ 1.04311572 -1.56832434 -1.56152458 -1.87231256  1.79044735  1.29162077
 -1.03280391 -0.6859253   0.25214191 -1.46385011 -0.02004369  0.66623427]
```

K-Nearest Neighbor (KNN)

In [268…

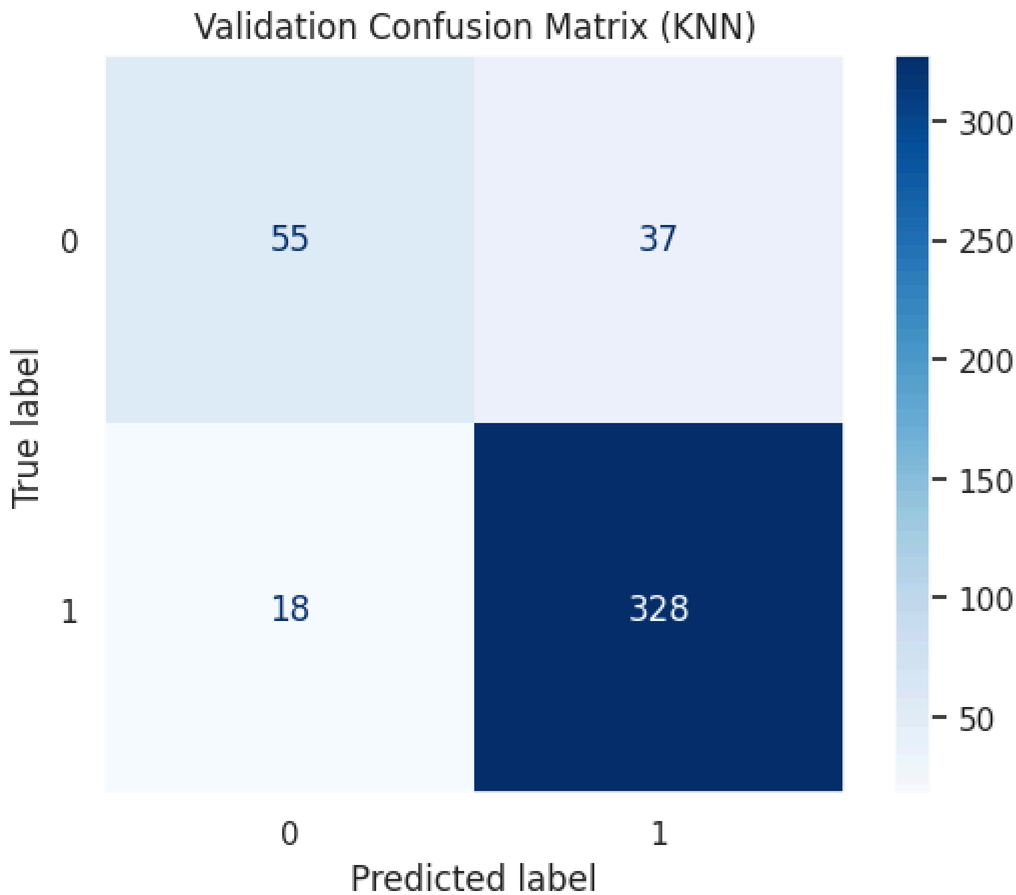
```
X_train_flat = X_train.reshape((X_train.shape[0], -1))
X_val_flat = X_val.reshape((X_val.shape[0], -1))

import matplotlib.pyplot as plt
rmse_list = []
for k in range(1, 100):
    model = KNeighborsRegressor(n_neighbors=k)
    model.fit(X_train, y_train)
    y_pred = model.predict(X_val)
    rmse = np.sqrt(mean_squared_error(y_val, y_pred))
    rmse_list.append(rmse)
    #print(f"k = {k}, RMSE = {rmse:.4f}")

# Plot results
plt.figure(figsize=(10, 5))
plt.plot(range(1, 100), rmse_list, marker='o')
plt.title('Validation RMSE vs. Number of Neighbors (k)')
plt.xlabel('k')
plt.ylabel('RMSE')
plt.grid(True)
plt.show()
```



In [269…

```
best_k = np.argmin(rmse_list) + 1  # +1 because range starts from 1
best_rmse = rmse_list[best_k - 1]
print(f"Best k: {best_k}, Lowest RMSE: {best_rmse:.4f}")
neigh = KNeighborsRegressor(n_neighbors=13)
neigh.fit(X_train_flat, y_train)

# Predict and round for classification
y_train_pred = np.round(neigh.predict(X_train_flat)).astype(int)
y_val_pred = np.round(neigh.predict(X_val_flat)).astype(int)

# Round true labels too (just in case)
y_train_true = np.round(y_train).astype(int)
y_val_true = np.round(y_val).astype(int)

# Compute accuracy
train_accuracy = accuracy_score(y_train_true, y_train_pred)
val_accuracy = accuracy_score(y_val_true, y_val_pred)

print(f"Train Accuracy: {train_accuracy:.4f}")
print(f"Validation Accuracy: {val_accuracy:.4f}")

# Generate and display confusion matrix for validation set
cm = confusion_matrix(y_val_true, y_val_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot(cmap='Blues', values_format='d')
plt.title("Validation Confusion Matrix (KNN)")
plt.grid(False)
plt.show()
```

```
Best k: 13, Lowest RMSE: 0.3091
Train Accuracy: 0.8716
Validation Accuracy: 0.8744
```



In [270…

```
# Predict regression values (no rounding)
y_test_pred = neigh.predict(X_test_scaled)

# Prepare submission file with continuous predictions
submission = pd.DataFrame({
    'id': test_df['id'],
    'rainfall': y_test_pred
})

submission.to_csv(PATH + 'submission_knn.csv', index=False)
```

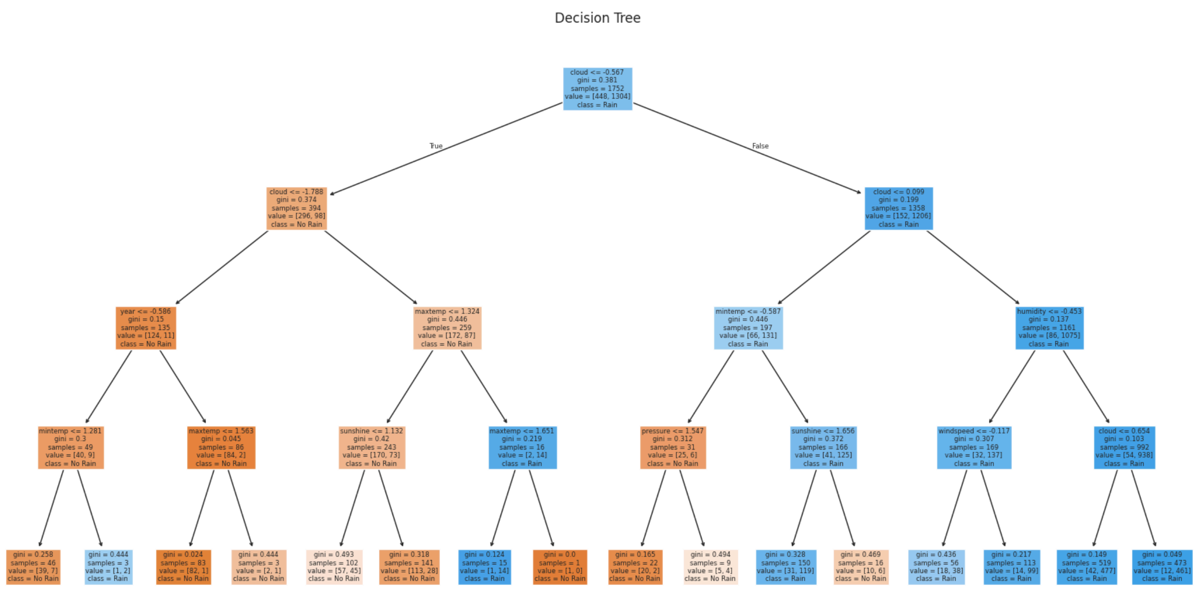
Decision Tree Model

In [271…

```
# Decision Tree Model
from sklearn.tree import DecisionTreeClassifier, plot_tree
tree_model = DecisionTreeClassifier(max_depth=4, random_state=42)
tree_model.fit(X_train, y_train)

# Evaluate model
y_pred = tree_model.predict(X_val)
print("Validation Accuracy:", accuracy_score(y_val, y_pred))
print("\nClassification Report:\n", classification_report(y_val, y_pred))

# Visualize the tree
plt.figure(figsize=(20,10))
plot_tree(tree_model, feature_names=X.columns, class_names=['No Rain', 'Rain'], fil
```

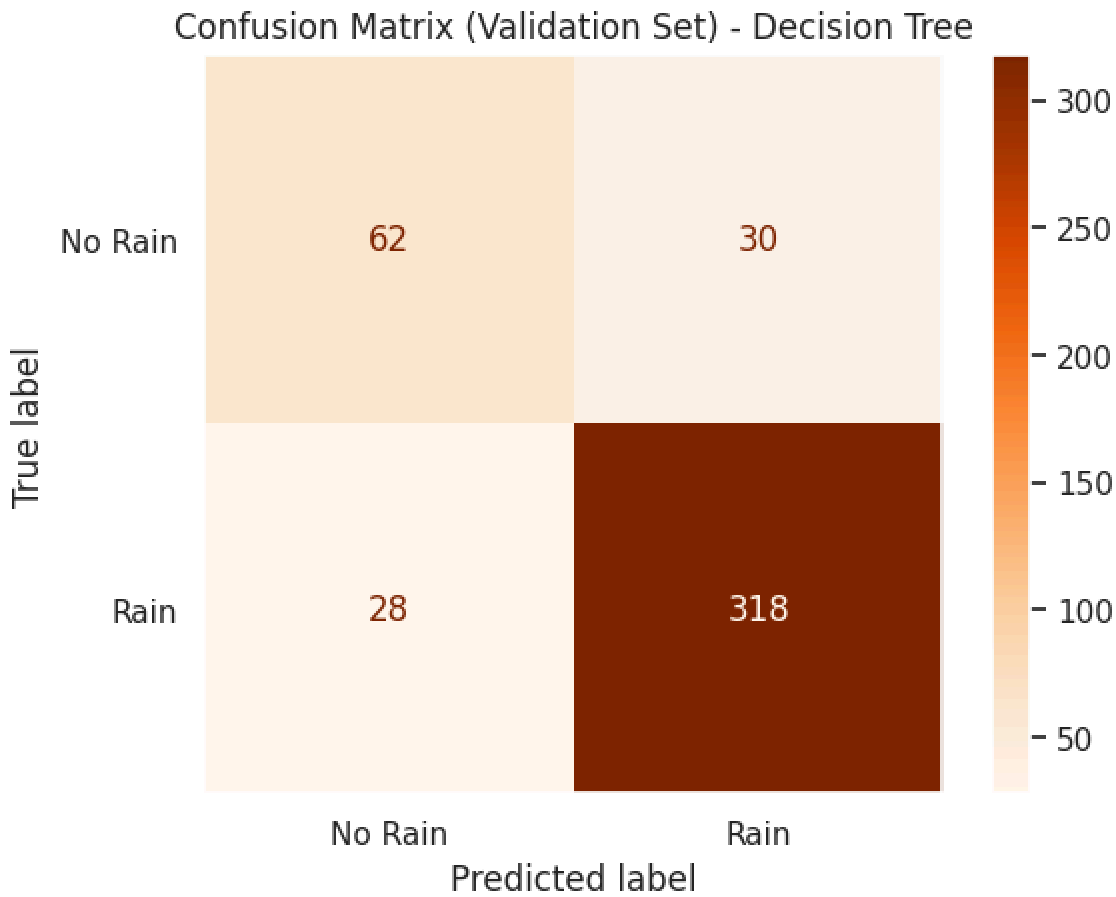
```
plt.title("Decision Tree")
plt.show()
```

```
Validation Accuracy: 0.867579908675799

Classification Report:
              precision    recall  f1-score   support

           0       0.69      0.67      0.68        92
           1       0.91      0.92      0.92       346

    accuracy                           0.87       438
   macro avg       0.80      0.80      0.80       438
weighted avg       0.87      0.87      0.87       438
```



In [272…

```
# Generate confusion matrix
cm = confusion_matrix(y_val, y_pred)

# Display the matrix
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["No Rain", "Rain
disp.plot(cmap=plt.cm.Oranges)
plt.title("Confusion Matrix (Validation Set) - Decision Tree")
plt.grid(False)
plt.show()
```



In [273…

```
# Get predicted probabilities for the positive class
test_probs = tree_model.predict_proba(X_test_scaled)[:, 1]  # probability of class

# If there's an 'id' column in test_df
submission = pd.DataFrame({
    "id": test_df["id"],
    "rainfall": test_probs
})

# Save to CSV
submission.to_csv(PATH + "submission_tree.csv", index=False)
```

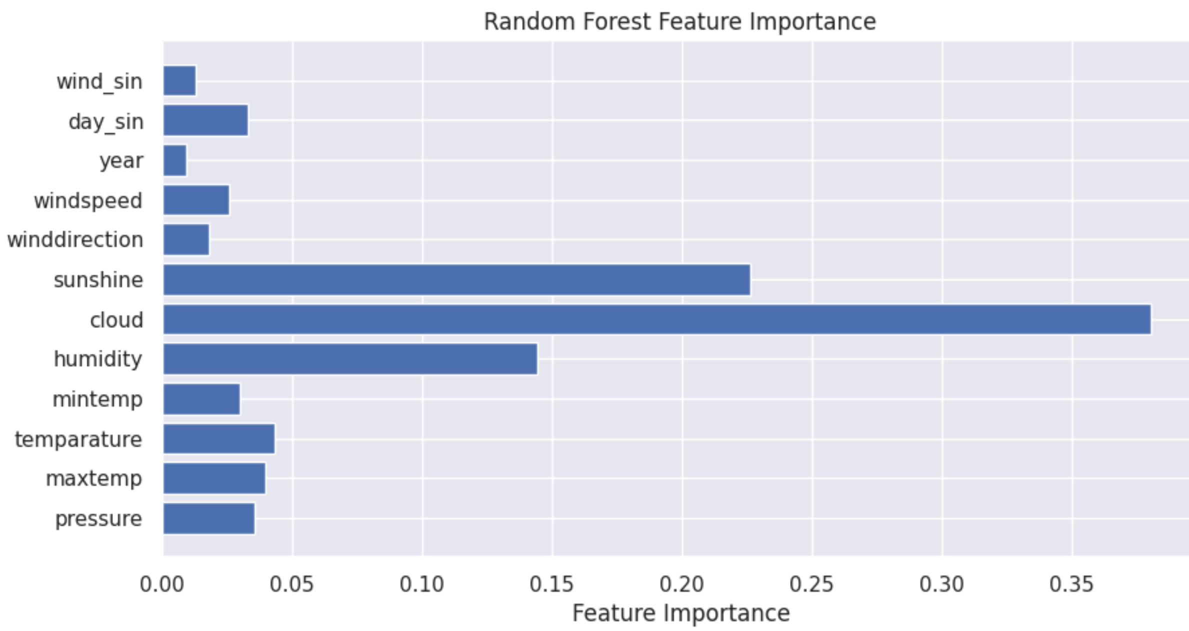
Decision Forest Model

In [274…

```
# Decision Forest
from sklearn.ensemble import RandomForestClassifier
forest_model = RandomForestClassifier(
    n_estimators=300,
    max_depth=6,
    random_state=42)
forest_model.fit(X_train, y_train)

# Evaluate on validation data
y_pred = forest_model.predict(X_val)
print("Validation Accuracy:", accuracy_score(y_val, y_pred))
print("\nClassification Report:\n", classification_report(y_val, y_pred))
```

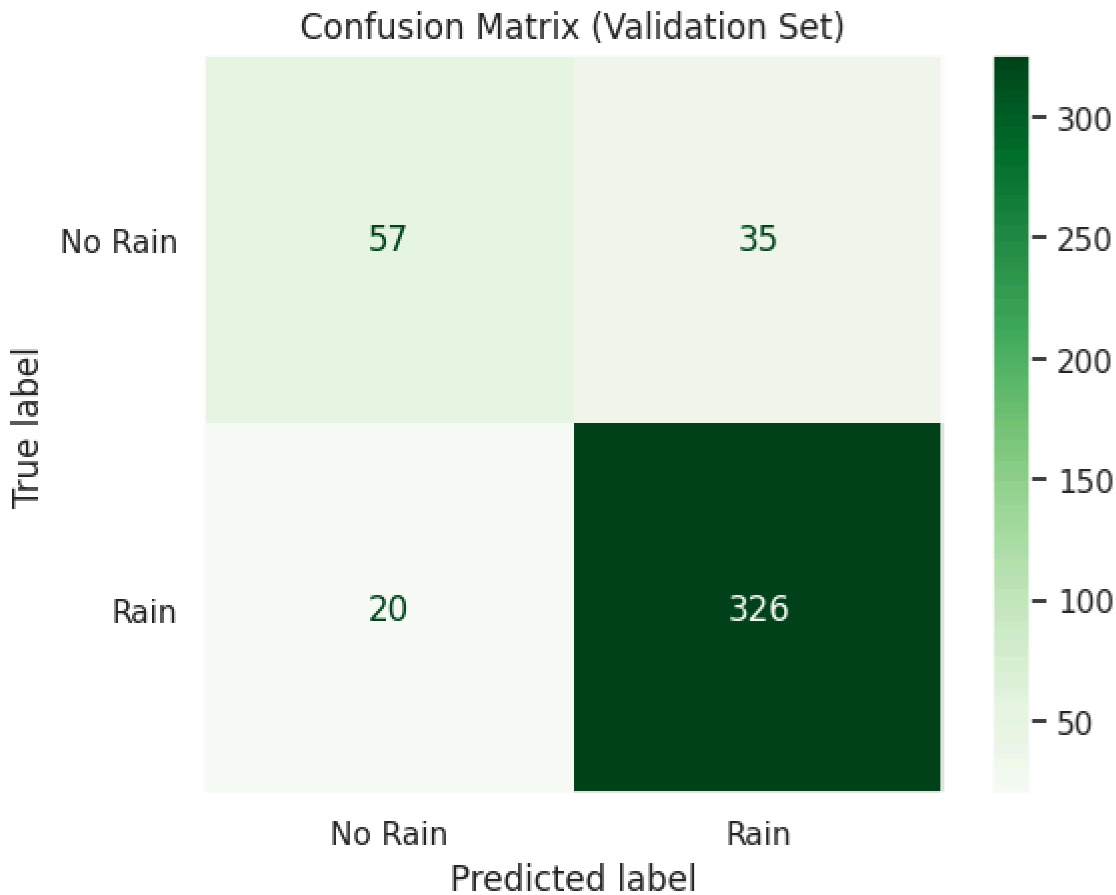
```
# Feature Importance Visualization
importances = forest_model.feature_importances_
features = X.columns
plt.figure(figsize=(10, 5))
plt.barh(features, importances)
plt.xlabel("Feature Importance")
plt.title("Random Forest Feature Importance")
plt.show()
```

```
Validation Accuracy: 0.8744292237442922

Classification Report:
              precision    recall  f1-score   support

           0       0.74      0.62      0.67        92
           1       0.90      0.94      0.92       346

    accuracy                           0.87       438
   macro avg       0.82      0.78      0.80       438
weighted avg       0.87      0.87      0.87       438
```



In [275…

```
# Confusion matrix
cm = confusion_matrix(y_val, y_pred)

# Display the matrix
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["No Rain", "Rain
disp.plot(cmap=plt.cm.Greens)
plt.title("Confusion Matrix (Validation Set)")
plt.grid(False)
plt.show()
```



In [276…

```
# Get predicted probabilities for the positive class
test_probs = forest_model.predict_proba(X_test_scaled)[:, 1]  # probability of clas

# If there's an 'id' column in test_df
submission = pd.DataFrame({
    "id": test_df["id"],
    "rainfall": test_probs
})

# Save to CSV
submission.to_csv(PATH + "submission_forest.csv", index=False)
```

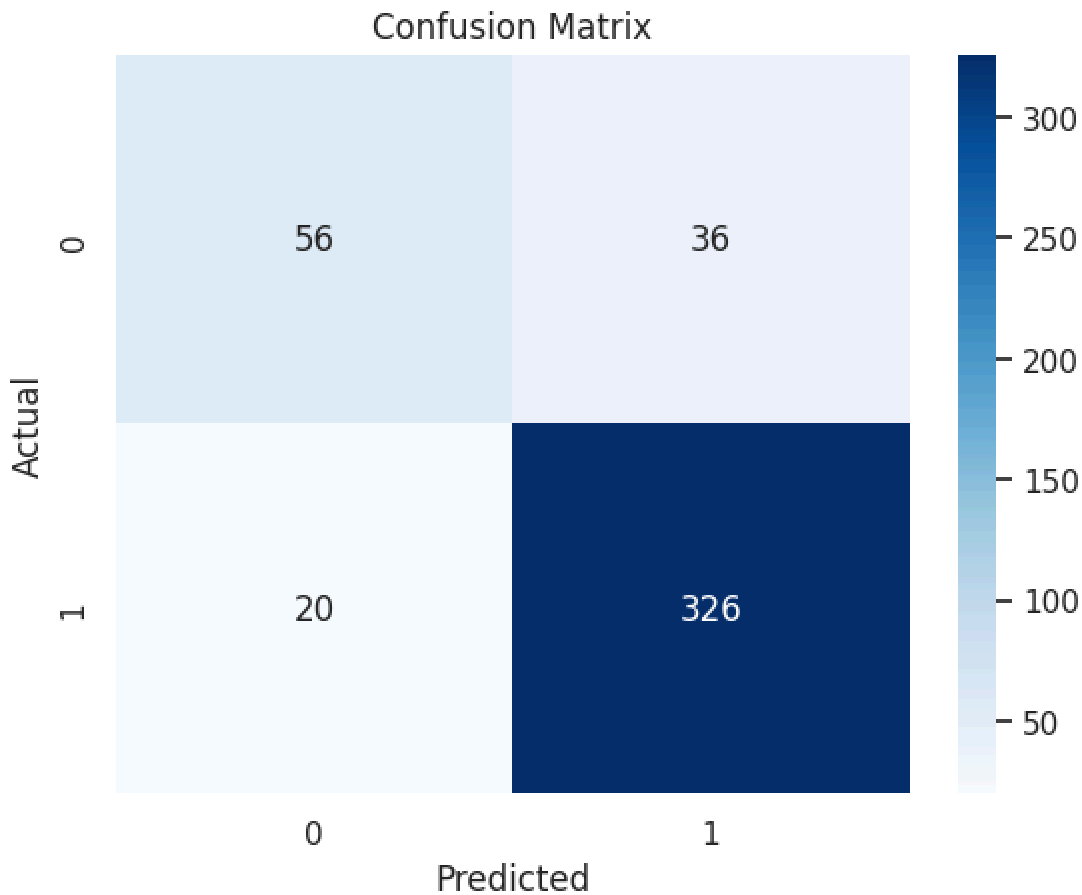
XGBoost Classifier

In [277…

```
model_xgb = XGBClassifier(
    n_estimators=300,
    learning_rate=0.05,
    max_depth=6,
    subsample=0.8,
    colsample_bytree=0.8,
    scale_pos_weight=3,  # if rainfall is imbalanced
    random_state=42
)

model_xgb.fit(X_train, y_train)
```

```
y_train_pred = model_xgb.predict(X_train)
train_acc = accuracy_score(y_train, y_train_pred)
print(f"Training Accuracy: {train_acc:.4f}")


y_pred = model_xgb.predict(X_val)
acc = accuracy_score(y_val, y_pred)
print(f"Validation Accuracy: {acc:.4f}")
#print(classification_report(y_val, y_pred))


cm = confusion_matrix(y_val, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()

# Get predicted probabilities for the positive class
test_probs = model_xgb.predict_proba(X_test_scaled)[:, 1]  # probability of class 1

# If there's an 'id' column in test_df
submission = pd.DataFrame({
    "id": test_df["id"],
    "rainfall": test_probs
})

# Save to CSV
submission.to_csv(PATH + "submission_XGboost.csv", index=False)
```

```
Training Accuracy: 0.9886
Validation Accuracy: 0.8721
```

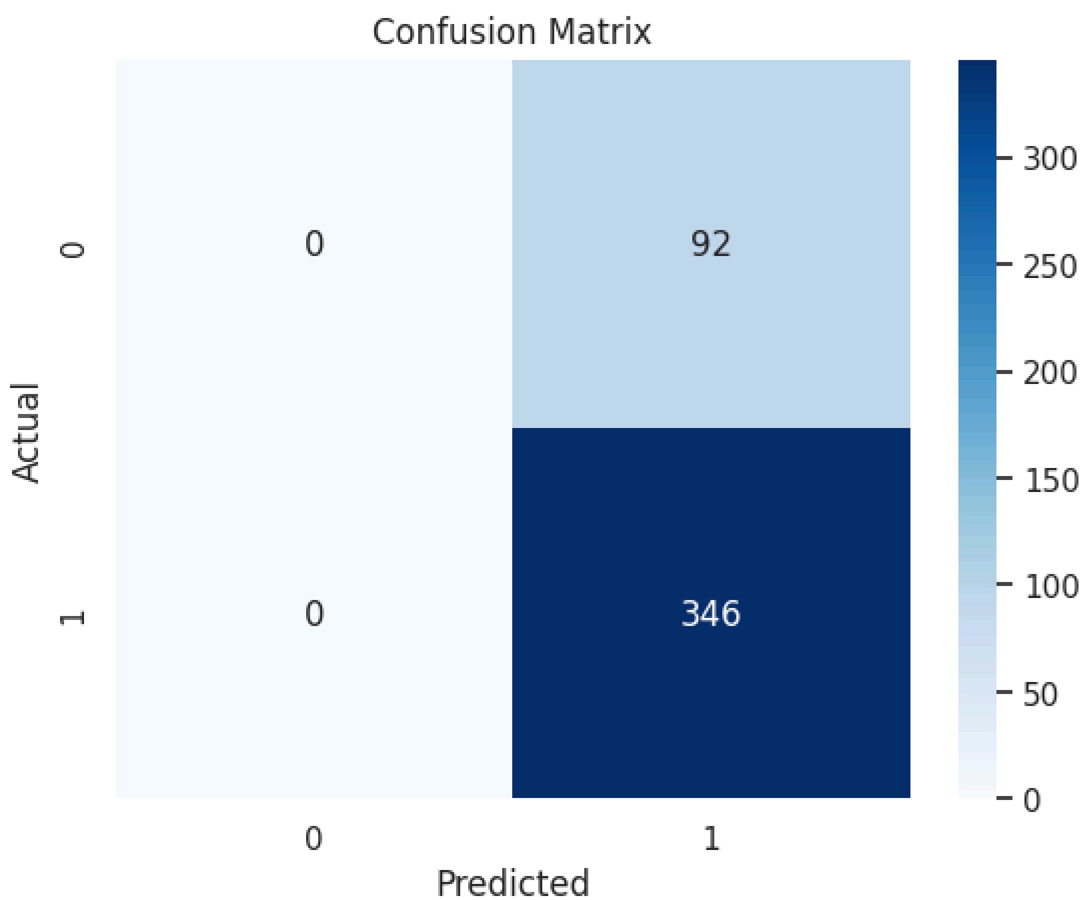


In [278…

```
# Another XGB model
model_xgb2 = XGBClassifier(
    max_depth=6,
    colsample_bytree=0.9,
    subsample=0.9,
    n_estimators=10_000,
    learning_rate=0.1,
    eval_metric="auc",
    early_stopping_rounds=100,
    alpha=1,
    random_state=42
)

# Train the model
model_xgb2.fit(
    X_train, y_train,
    eval_set=[(X_val, y_val)],
    verbose=100
)

# Predict probabilities
oof_xgb = model_xgb2.predict_proba(X_val)[:, 1]

# Optionally evaluate
print("Validation ROC AUC:", roc_auc_score(y_val, oof_xgb))

y_pred = model_xgb2.predict(X_val)
acc2 = accuracy_score(y_val, y_pred)
```

```
print(f"Validation Accuracy: {acc2:.4f}")
#print(classification_report(y_val, y_pred))


cm = confusion_matrix(y_val, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()

# Get predicted probabilities for the positive class
test_probs = model_xgb2.predict_proba(X_test_scaled)[:, 1]  # probability of class


submission = pd.DataFrame({
    "id": test_df["id"],
    "rainfall": test_probs
})

# Save to CSV
submission.to_csv(PATH + "submission_XGboost2.csv", index=False)
```

```
[0]	validation_0-auc:0.85844
[100]	validation_0-auc:0.86887
[101]	validation_0-auc:0.86884
Validation ROC AUC: 0.8783142749434532
Validation Accuracy: 0.7900
```



Neural Network

In [279…

```
# NN
# Clear session
tf.keras.backend.clear_session()
tf.random.set_seed(0)

# Build final model
nn_model = keras.Sequential([
    layers.Input(shape=(X_train.shape[1],)),
    layers.Dense(112, activation='relu'),  # Layer 1
    layers.Dense(224, activation='relu'),  # Layer 2
    layers.Dense(160, activation='relu'),  # Layer 3
    # No dropout
    layers.Dense(1, activation='sigmoid')  # Output layer for binary classification
])

# Compile with SGD and lr = 0.02
nn_model.compile(
    optimizer=keras.optimizers.SGD(learning_rate=0.005),
    loss='binary_crossentropy',
    metrics=['accuracy']
)

# Train the model
history = nn_model.fit(
    X_train, y_train,
    validation_data=(X_val, y_val),
    epochs=500,
    batch_size=32,
    callbacks=[tf.keras.callbacks.EarlyStopping(patience=20, restore_best_weights=T
    verbose=1
)

# Plot losses
plt.figure(figsize=(8, 5))
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Val Loss')
plt.title("Training & Validation Loss")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.grid(True)
plt.legend()
plt.show()

# Evaluate on training and validation sets
train_loss, train_acc = nn_model.evaluate(X_train, y_train, verbose=0)
val_loss, val_acc = nn_model.evaluate(X_val, y_val, verbose=0)

print(f"Training Accuracy: {train_acc:.4f}")
print(f"Validation Accuracy: {val_acc:.4f}")
```

```
Epoch 1/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - accuracy: 0.7408 - loss: 0.6475 - val_accur
acy: 0.7900 - val_loss: 0.5825
Epoch 2/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7433 - loss: 0.5969 - val_accur
acy: 0.7900 - val_loss: 0.5396
Epoch 3/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7433 - loss: 0.5615 - val_accur
acy: 0.7900 - val_loss: 0.5077
Epoch 4/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7433 - loss: 0.5329 - val_accur
acy: 0.7900 - val_loss: 0.4815
Epoch 5/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7438 - loss: 0.5079 - val_accur
acy: 0.7900 - val_loss: 0.4586
Epoch 6/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7496 - loss: 0.4851 - val_accur
acy: 0.7991 - val_loss: 0.4381
Epoch 7/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7646 - loss: 0.4640 - val_accur
acy: 0.8105 - val_loss: 0.4197
Epoch 8/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8072 - loss: 0.4447 - val_accur
acy: 0.8311 - val_loss: 0.4035
Epoch 9/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8275 - loss: 0.4272 - val_accur
acy: 0.8379 - val_loss: 0.3896
Epoch 10/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8400 - loss: 0.4119 - val_accur
acy: 0.8493 - val_loss: 0.3779
Epoch 11/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8441 - loss: 0.3989 - val_accur
acy: 0.8539 - val_loss: 0.3683
Epoch 12/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8508 - loss: 0.3879 - val_accur
acy: 0.8584 - val_loss: 0.3607
Epoch 13/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8579 - loss: 0.3789 - val_accur
acy: 0.8630 - val_loss: 0.3548
Epoch 14/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8634 - loss: 0.3716 - val_accur
acy: 0.8607 - val_loss: 0.3503
Epoch 15/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8629 - loss: 0.3657 - val_accur
acy: 0.8630 - val_loss: 0.3469
Epoch 16/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8611 - loss: 0.3611 - val_accur
acy: 0.8630 - val_loss: 0.3445
Epoch 17/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8620 - loss: 0.3574 - val_accur
acy: 0.8630 - val_loss: 0.3427
Epoch 18/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8616 - loss: 0.3545 - val_accur
acy: 0.8630 - val_loss: 0.3414
Epoch 19/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8614 - loss: 0.3522 - val_accur
```

```
acy: 0.8653 - val_loss: 0.3405
Epoch 20/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8617 - loss: 0.3503 - val_accur
acy: 0.8653 - val_loss: 0.3398
Epoch 21/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8605 - loss: 0.3487 - val_accur
acy: 0.8630 - val_loss: 0.3392
Epoch 22/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8631 - loss: 0.3474 - val_accur
acy: 0.8630 - val_loss: 0.3388
Epoch 23/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8656 - loss: 0.3464 - val_accur
acy: 0.8630 - val_loss: 0.3383
Epoch 24/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8654 - loss: 0.3454 - val_accur
acy: 0.8630 - val_loss: 0.3380
Epoch 25/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8667 - loss: 0.3446 - val_accur
acy: 0.8653 - val_loss: 0.3377
Epoch 26/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8663 - loss: 0.3439 - val_accur
acy: 0.8653 - val_loss: 0.3374
Epoch 27/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8664 - loss: 0.3432 - val_accur
acy: 0.8653 - val_loss: 0.3371
Epoch 28/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8674 - loss: 0.3426 - val_accur
acy: 0.8653 - val_loss: 0.3369
Epoch 29/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8675 - loss: 0.3421 - val_accur
acy: 0.8653 - val_loss: 0.3366
Epoch 30/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8676 - loss: 0.3415 - val_accur
acy: 0.8653 - val_loss: 0.3363
Epoch 31/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8676 - loss: 0.3411 - val_accur
acy: 0.8653 - val_loss: 0.3361
Epoch 32/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8678 - loss: 0.3406 - val_accur
acy: 0.8653 - val_loss: 0.3358
Epoch 33/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8678 - loss: 0.3401 - val_accur
acy: 0.8653 - val_loss: 0.3355
Epoch 34/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8678 - loss: 0.3397 - val_accur
acy: 0.8653 - val_loss: 0.3352
Epoch 35/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8676 - loss: 0.3393 - val_accur
acy: 0.8653 - val_loss: 0.3350
Epoch 36/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8692 - loss: 0.3389 - val_accur
acy: 0.8676 - val_loss: 0.3347
Epoch 37/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8692 - loss: 0.3386 - val_accur
acy: 0.8676 - val_loss: 0.3344
Epoch 38/500
```

```
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8690 - loss: 0.3382 - val_accur
acy: 0.8676 - val_loss: 0.3342
Epoch 39/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8690 - loss: 0.3379 - val_accur
acy: 0.8676 - val_loss: 0.3339
Epoch 40/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8700 - loss: 0.3375 - val_accur
acy: 0.8676 - val_loss: 0.3336
Epoch 41/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8700 - loss: 0.3372 - val_accur
acy: 0.8676 - val_loss: 0.3334
Epoch 42/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8699 - loss: 0.3369 - val_accur
acy: 0.8676 - val_loss: 0.3331
Epoch 43/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8699 - loss: 0.3366 - val_accur
acy: 0.8653 - val_loss: 0.3329
Epoch 44/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8699 - loss: 0.3362 - val_accur
acy: 0.8653 - val_loss: 0.3326
Epoch 45/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8699 - loss: 0.3359 - val_accur
acy: 0.8653 - val_loss: 0.3323
Epoch 46/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8699 - loss: 0.3357 - val_accur
acy: 0.8653 - val_loss: 0.3321
Epoch 47/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8700 - loss: 0.3354 - val_accur
acy: 0.8653 - val_loss: 0.3319
Epoch 48/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8703 - loss: 0.3351 - val_accur
acy: 0.8653 - val_loss: 0.3316
Epoch 49/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8713 - loss: 0.3348 - val_accur
acy: 0.8630 - val_loss: 0.3314
Epoch 50/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8713 - loss: 0.3345 - val_accur
acy: 0.8630 - val_loss: 0.3311
Epoch 51/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8715 - loss: 0.3343 - val_accur
acy: 0.8630 - val_loss: 0.3309
Epoch 52/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8715 - loss: 0.3340 - val_accur
acy: 0.8630 - val_loss: 0.3307
Epoch 53/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8717 - loss: 0.3337 - val_accur
acy: 0.8630 - val_loss: 0.3305
Epoch 54/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8717 - loss: 0.3335 - val_accur
acy: 0.8630 - val_loss: 0.3302
Epoch 55/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8717 - loss: 0.3332 - val_accur
acy: 0.8630 - val_loss: 0.3300
Epoch 56/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8717 - loss: 0.3330 - val_accur
acy: 0.8630 - val_loss: 0.3298
```

```
Epoch 57/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8717 - loss: 0.3327 - val_accur
acy: 0.8630 - val_loss: 0.3296
Epoch 58/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8717 - loss: 0.3325 - val_accur
acy: 0.8630 - val_loss: 0.3294
Epoch 59/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8717 - loss: 0.3322 - val_accur
acy: 0.8630 - val_loss: 0.3291
Epoch 60/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8717 - loss: 0.3320 - val_accur
acy: 0.8630 - val_loss: 0.3289
Epoch 61/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8725 - loss: 0.3317 - val_accur
acy: 0.8630 - val_loss: 0.3287
Epoch 62/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8725 - loss: 0.3315 - val_accur
acy: 0.8630 - val_loss: 0.3285
Epoch 63/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8731 - loss: 0.3313 - val_accur
acy: 0.8630 - val_loss: 0.3283
Epoch 64/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8731 - loss: 0.3310 - val_accur
acy: 0.8630 - val_loss: 0.3281
Epoch 65/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8731 - loss: 0.3308 - val_accur
acy: 0.8630 - val_loss: 0.3279
Epoch 66/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8731 - loss: 0.3306 - val_accur
acy: 0.8630 - val_loss: 0.3278
Epoch 67/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8731 - loss: 0.3303 - val_accur
acy: 0.8630 - val_loss: 0.3276
Epoch 68/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8731 - loss: 0.3301 - val_accur
acy: 0.8653 - val_loss: 0.3274
Epoch 69/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8731 - loss: 0.3299 - val_accur
acy: 0.8653 - val_loss: 0.3272
Epoch 70/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8729 - loss: 0.3297 - val_accur
acy: 0.8653 - val_loss: 0.3270
Epoch 71/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8729 - loss: 0.3295 - val_accur
acy: 0.8653 - val_loss: 0.3269
Epoch 72/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8729 - loss: 0.3293 - val_accur
acy: 0.8653 - val_loss: 0.3267
Epoch 73/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8729 - loss: 0.3291 - val_accur
acy: 0.8653 - val_loss: 0.3265
Epoch 74/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8729 - loss: 0.3288 - val_accur
acy: 0.8653 - val_loss: 0.3263
Epoch 75/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8729 - loss: 0.3286 - val_accur
```

```
acy: 0.8653 - val_loss: 0.3262
Epoch 76/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8734 - loss: 0.3284 - val_accur
acy: 0.8653 - val_loss: 0.3260
Epoch 77/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8734 - loss: 0.3282 - val_accur
acy: 0.8676 - val_loss: 0.3259
Epoch 78/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8738 - loss: 0.3280 - val_accur
acy: 0.8676 - val_loss: 0.3257
Epoch 79/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8724 - loss: 0.3279 - val_accur
acy: 0.8676 - val_loss: 0.3256
Epoch 80/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8724 - loss: 0.3277 - val_accur
acy: 0.8699 - val_loss: 0.3255
Epoch 81/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8724 - loss: 0.3275 - val_accur
acy: 0.8699 - val_loss: 0.3253
Epoch 82/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8724 - loss: 0.3273 - val_accur
acy: 0.8699 - val_loss: 0.3252
Epoch 83/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8734 - loss: 0.3271 - val_accur
acy: 0.8699 - val_loss: 0.3250
Epoch 84/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8734 - loss: 0.3269 - val_accur
acy: 0.8699 - val_loss: 0.3249
Epoch 85/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8734 - loss: 0.3267 - val_accur
acy: 0.8699 - val_loss: 0.3248
Epoch 86/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8734 - loss: 0.3265 - val_accur
acy: 0.8699 - val_loss: 0.3246
Epoch 87/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8733 - loss: 0.3264 - val_accur
acy: 0.8699 - val_loss: 0.3245
Epoch 88/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8733 - loss: 0.3262 - val_accur
acy: 0.8699 - val_loss: 0.3244
Epoch 89/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8733 - loss: 0.3260 - val_accur
acy: 0.8699 - val_loss: 0.3243
Epoch 90/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8741 - loss: 0.3258 - val_accur
acy: 0.8699 - val_loss: 0.3241
Epoch 91/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8741 - loss: 0.3257 - val_accur
acy: 0.8699 - val_loss: 0.3240
Epoch 92/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8738 - loss: 0.3255 - val_accur
acy: 0.8699 - val_loss: 0.3239
Epoch 93/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8738 - loss: 0.3253 - val_accur
acy: 0.8699 - val_loss: 0.3238
Epoch 94/500
```

```
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8738 - loss: 0.3251 - val_accur
acy: 0.8699 - val_loss: 0.3237
Epoch 95/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8738 - loss: 0.3250 - val_accur
acy: 0.8699 - val_loss: 0.3235
Epoch 96/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8738 - loss: 0.3248 - val_accur
acy: 0.8699 - val_loss: 0.3234
Epoch 97/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8738 - loss: 0.3246 - val_accur
acy: 0.8699 - val_loss: 0.3233
Epoch 98/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8740 - loss: 0.3244 - val_accur
acy: 0.8699 - val_loss: 0.3232
Epoch 99/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8740 - loss: 0.3243 - val_accur
acy: 0.8699 - val_loss: 0.3231
Epoch 100/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8740 - loss: 0.3241 - val_accur
acy: 0.8699 - val_loss: 0.3230
Epoch 101/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8740 - loss: 0.3239 - val_accur
acy: 0.8699 - val_loss: 0.3229
Epoch 102/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8740 - loss: 0.3238 - val_accur
acy: 0.8699 - val_loss: 0.3228
Epoch 103/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8740 - loss: 0.3236 - val_accur
acy: 0.8699 - val_loss: 0.3227
Epoch 104/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8740 - loss: 0.3234 - val_accur
acy: 0.8699 - val_loss: 0.3226
Epoch 105/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8740 - loss: 0.3233 - val_accur
acy: 0.8699 - val_loss: 0.3225
Epoch 106/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8740 - loss: 0.3231 - val_accur
acy: 0.8699 - val_loss: 0.3224
Epoch 107/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8740 - loss: 0.3229 - val_accur
acy: 0.8699 - val_loss: 0.3223
Epoch 108/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8740 - loss: 0.3227 - val_accur
acy: 0.8699 - val_loss: 0.3222
Epoch 109/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8740 - loss: 0.3226 - val_accur
acy: 0.8699 - val_loss: 0.3221
Epoch 110/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8740 - loss: 0.3224 - val_accur
acy: 0.8699 - val_loss: 0.3220
Epoch 111/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3222 - val_accur
acy: 0.8699 - val_loss: 0.3219
Epoch 112/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3221 - val_accur
acy: 0.8699 - val_loss: 0.3218
```

```
Epoch 113/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3219 - val_accur
acy: 0.8699 - val_loss: 0.3217
Epoch 114/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3217 - val_accur
acy: 0.8699 - val_loss: 0.3216
Epoch 115/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3216 - val_accur
acy: 0.8699 - val_loss: 0.3216
Epoch 116/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.8751 - loss: 0.3214 - val_accur
acy: 0.8699 - val_loss: 0.3215
Epoch 117/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3212 - val_accur
acy: 0.8699 - val_loss: 0.3214
Epoch 118/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3211 - val_accur
acy: 0.8699 - val_loss: 0.3213
Epoch 119/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3209 - val_accur
acy: 0.8699 - val_loss: 0.3213
Epoch 120/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3207 - val_accur
acy: 0.8699 - val_loss: 0.3212
Epoch 121/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3206 - val_accur
acy: 0.8699 - val_loss: 0.3211
Epoch 122/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3204 - val_accur
acy: 0.8699 - val_loss: 0.3210
Epoch 123/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3202 - val_accur
acy: 0.8699 - val_loss: 0.3210
Epoch 124/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3201 - val_accur
acy: 0.8699 - val_loss: 0.3209
Epoch 125/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3199 - val_accur
acy: 0.8699 - val_loss: 0.3208
Epoch 126/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3197 - val_accur
acy: 0.8699 - val_loss: 0.3207
Epoch 127/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3196 - val_accur
acy: 0.8699 - val_loss: 0.3207
Epoch 128/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3194 - val_accur
acy: 0.8699 - val_loss: 0.3206
Epoch 129/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3193 - val_accur
acy: 0.8699 - val_loss: 0.3205
Epoch 130/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3191 - val_accur
acy: 0.8699 - val_loss: 0.3205
Epoch 131/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3189 - val_accur
```

```
acy: 0.8699 - val_loss: 0.3204
Epoch 132/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3188 - val_accur
acy: 0.8699 - val_loss: 0.3204
Epoch 133/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3186 - val_accur
acy: 0.8699 - val_loss: 0.3203
Epoch 134/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3184 - val_accur
acy: 0.8699 - val_loss: 0.3202
Epoch 135/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3183 - val_accur
acy: 0.8699 - val_loss: 0.3202
Epoch 136/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3181 - val_accur
acy: 0.8699 - val_loss: 0.3201
Epoch 137/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8751 - loss: 0.3179 - val_accur
acy: 0.8699 - val_loss: 0.3201
Epoch 138/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8753 - loss: 0.3178 - val_accur
acy: 0.8699 - val_loss: 0.3200
Epoch 139/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8763 - loss: 0.3176 - val_accur
acy: 0.8699 - val_loss: 0.3200
Epoch 140/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8766 - loss: 0.3175 - val_accur
acy: 0.8699 - val_loss: 0.3199
Epoch 141/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8766 - loss: 0.3173 - val_accur
acy: 0.8721 - val_loss: 0.3199
Epoch 142/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8766 - loss: 0.3172 - val_accur
acy: 0.8721 - val_loss: 0.3198
Epoch 143/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8766 - loss: 0.3170 - val_accur
acy: 0.8721 - val_loss: 0.3198
Epoch 144/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8766 - loss: 0.3168 - val_accur
acy: 0.8721 - val_loss: 0.3197
Epoch 145/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8766 - loss: 0.3167 - val_accur
acy: 0.8721 - val_loss: 0.3197
Epoch 146/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8766 - loss: 0.3165 - val_accur
acy: 0.8721 - val_loss: 0.3196
Epoch 147/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8766 - loss: 0.3164 - val_accur
acy: 0.8721 - val_loss: 0.3196
Epoch 148/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8766 - loss: 0.3162 - val_accur
acy: 0.8721 - val_loss: 0.3196
Epoch 149/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8766 - loss: 0.3160 - val_accur
acy: 0.8721 - val_loss: 0.3195
Epoch 150/500
```

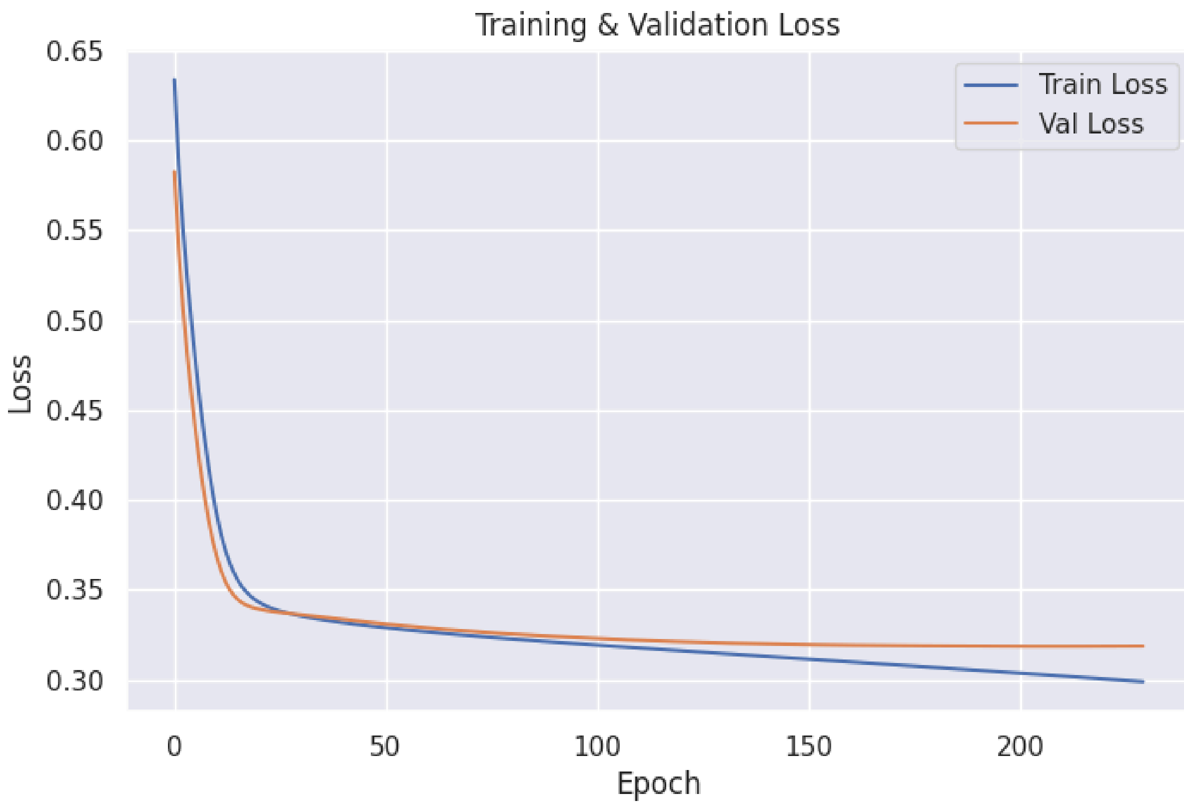
```
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8766 - loss: 0.3159 - val_accur
acy: 0.8721 - val_loss: 0.3195
Epoch 151/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8766 - loss: 0.3157 - val_accur
acy: 0.8721 - val_loss: 0.3195
Epoch 152/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8766 - loss: 0.3155 - val_accur
acy: 0.8721 - val_loss: 0.3194
Epoch 153/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8768 - loss: 0.3154 - val_accur
acy: 0.8721 - val_loss: 0.3194
Epoch 154/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8768 - loss: 0.3152 - val_accur
acy: 0.8721 - val_loss: 0.3194
Epoch 155/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8768 - loss: 0.3150 - val_accur
acy: 0.8721 - val_loss: 0.3193
Epoch 156/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8768 - loss: 0.3149 - val_accur
acy: 0.8721 - val_loss: 0.3193
Epoch 157/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8768 - loss: 0.3147 - val_accur
acy: 0.8721 - val_loss: 0.3193
Epoch 158/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8768 - loss: 0.3146 - val_accur
acy: 0.8721 - val_loss: 0.3192
Epoch 159/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8771 - loss: 0.3144 - val_accur
acy: 0.8721 - val_loss: 0.3192
Epoch 160/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8771 - loss: 0.3142 - val_accur
acy: 0.8721 - val_loss: 0.3192
Epoch 161/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8771 - loss: 0.3141 - val_accur
acy: 0.8721 - val_loss: 0.3192
Epoch 162/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8771 - loss: 0.3139 - val_accur
acy: 0.8721 - val_loss: 0.3191
Epoch 163/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8772 - loss: 0.3138 - val_accur
acy: 0.8721 - val_loss: 0.3191
Epoch 164/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8772 - loss: 0.3136 - val_accur
acy: 0.8744 - val_loss: 0.3191
Epoch 165/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8772 - loss: 0.3135 - val_accur
acy: 0.8744 - val_loss: 0.3191
Epoch 166/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8772 - loss: 0.3133 - val_accur
acy: 0.8744 - val_loss: 0.3191
Epoch 167/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8772 - loss: 0.3131 - val_accur
acy: 0.8744 - val_loss: 0.3190
Epoch 168/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8772 - loss: 0.3130 - val_accur
acy: 0.8744 - val_loss: 0.3190
```

```
Epoch 169/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8772 - loss: 0.3128 - val_accur
acy: 0.8744 - val_loss: 0.3190
Epoch 170/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8780 - loss: 0.3127 - val_accur
acy: 0.8744 - val_loss: 0.3190
Epoch 171/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8780 - loss: 0.3125 - val_accur
acy: 0.8744 - val_loss: 0.3190
Epoch 172/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8780 - loss: 0.3124 - val_accur
acy: 0.8744 - val_loss: 0.3190
Epoch 173/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8780 - loss: 0.3122 - val_accur
acy: 0.8744 - val_loss: 0.3189
Epoch 174/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8780 - loss: 0.3120 - val_accur
acy: 0.8744 - val_loss: 0.3189
Epoch 175/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8780 - loss: 0.3119 - val_accur
acy: 0.8744 - val_loss: 0.3189
Epoch 176/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8780 - loss: 0.3117 - val_accur
acy: 0.8744 - val_loss: 0.3189
Epoch 177/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8780 - loss: 0.3116 - val_accur
acy: 0.8744 - val_loss: 0.3189
Epoch 178/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8780 - loss: 0.3114 - val_accur
acy: 0.8744 - val_loss: 0.3188
Epoch 179/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8780 - loss: 0.3112 - val_accur
acy: 0.8744 - val_loss: 0.3188
Epoch 180/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8781 - loss: 0.3111 - val_accur
acy: 0.8744 - val_loss: 0.3188
Epoch 181/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8781 - loss: 0.3109 - val_accur
acy: 0.8744 - val_loss: 0.3188
Epoch 182/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8781 - loss: 0.3107 - val_accur
acy: 0.8744 - val_loss: 0.3188
Epoch 183/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8781 - loss: 0.3106 - val_accur
acy: 0.8744 - val_loss: 0.3187
Epoch 184/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8781 - loss: 0.3104 - val_accur
acy: 0.8744 - val_loss: 0.3187
Epoch 185/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8781 - loss: 0.3103 - val_accur
acy: 0.8744 - val_loss: 0.3187
Epoch 186/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8781 - loss: 0.3101 - val_accur
acy: 0.8744 - val_loss: 0.3187
Epoch 187/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8781 - loss: 0.3099 - val_accur
```

```
acy: 0.8744 - val_loss: 0.3187
Epoch 188/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8782 - loss: 0.3098 - val_accur
acy: 0.8744 - val_loss: 0.3187
Epoch 189/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8779 - loss: 0.3096 - val_accur
acy: 0.8744 - val_loss: 0.3186
Epoch 190/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8779 - loss: 0.3095 - val_accur
acy: 0.8744 - val_loss: 0.3186
Epoch 191/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8779 - loss: 0.3093 - val_accur
acy: 0.8744 - val_loss: 0.3186
Epoch 192/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8779 - loss: 0.3091 - val_accur
acy: 0.8744 - val_loss: 0.3186
Epoch 193/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8779 - loss: 0.3090 - val_accur
acy: 0.8744 - val_loss: 0.3186
Epoch 194/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8782 - loss: 0.3088 - val_accur
acy: 0.8744 - val_loss: 0.3186
Epoch 195/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8782 - loss: 0.3087 - val_accur
acy: 0.8744 - val_loss: 0.3186
Epoch 196/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8782 - loss: 0.3085 - val_accur
acy: 0.8744 - val_loss: 0.3186
Epoch 197/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8782 - loss: 0.3084 - val_accur
acy: 0.8744 - val_loss: 0.3186
Epoch 198/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8782 - loss: 0.3082 - val_accur
acy: 0.8744 - val_loss: 0.3186
Epoch 199/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8782 - loss: 0.3081 - val_accur
acy: 0.8744 - val_loss: 0.3185
Epoch 200/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8782 - loss: 0.3079 - val_accur
acy: 0.8744 - val_loss: 0.3186
Epoch 201/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8792 - loss: 0.3077 - val_accur
acy: 0.8744 - val_loss: 0.3186
Epoch 202/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8792 - loss: 0.3076 - val_accur
acy: 0.8744 - val_loss: 0.3186
Epoch 203/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8792 - loss: 0.3074 - val_accur
acy: 0.8744 - val_loss: 0.3186
Epoch 204/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8792 - loss: 0.3073 - val_accur
acy: 0.8744 - val_loss: 0.3186
Epoch 205/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8792 - loss: 0.3071 - val_accur
acy: 0.8744 - val_loss: 0.3186
Epoch 206/500
```

```
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8807 - loss: 0.3070 - val_accuracy: 0.8744 - val_loss: 0.3186
Epoch 207/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8807 - loss: 0.3068 - val_accuracy: 0.8744 - val_loss: 0.3186
Epoch 208/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8811 - loss: 0.3066 - val_accuracy: 0.8744 - val_loss: 0.3185
Epoch 209/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8811 - loss: 0.3065 - val_accuracy: 0.8744 - val_loss: 0.3185
Epoch 210/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8811 - loss: 0.3063 - val_accuracy: 0.8744 - val_loss: 0.3185
Epoch 211/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8817 - loss: 0.3062 - val_accuracy: 0.8744 - val_loss: 0.3185
Epoch 212/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8817 - loss: 0.3060 - val_accuracy: 0.8744 - val_loss: 0.3186
Epoch 213/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8817 - loss: 0.3058 - val_accuracy: 0.8744 - val_loss: 0.3186
Epoch 214/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8817 - loss: 0.3057 - val_accuracy: 0.8744 - val_loss: 0.3186
Epoch 215/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8817 - loss: 0.3055 - val_accuracy: 0.8744 - val_loss: 0.3186
Epoch 216/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8817 - loss: 0.3054 - val_accuracy: 0.8744 - val_loss: 0.3186
Epoch 217/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8833 - loss: 0.3052 - val_accuracy: 0.8744 - val_loss: 0.3186
Epoch 218/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8851 - loss: 0.3050 - val_accuracy: 0.8744 - val_loss: 0.3186
Epoch 219/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8851 - loss: 0.3049 - val_accuracy: 0.8744 - val_loss: 0.3186
Epoch 220/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8851 - loss: 0.3047 - val_accuracy: 0.8744 - val_loss: 0.3186
Epoch 221/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8851 - loss: 0.3045 - val_accuracy: 0.8744 - val_loss: 0.3186
Epoch 222/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8851 - loss: 0.3044 - val_accuracy: 0.8744 - val_loss: 0.3186
Epoch 223/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8851 - loss: 0.3042 - val_accuracy: 0.8744 - val_loss: 0.3186
Epoch 224/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8851 - loss: 0.3040 - val_accuracy: 0.8744 - val_loss: 0.3186
```

```
Epoch 225/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8851 - loss: 0.3039 - val_accur
acy: 0.8744 - val_loss: 0.3187
Epoch 226/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8851 - loss: 0.3037 - val_accur
acy: 0.8744 - val_loss: 0.3187
Epoch 227/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8851 - loss: 0.3035 - val_accur
acy: 0.8744 - val_loss: 0.3187
Epoch 228/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8851 - loss: 0.3034 - val_accur
acy: 0.8744 - val_loss: 0.3187
Epoch 229/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8851 - loss: 0.3032 - val_accur
acy: 0.8744 - val_loss: 0.3187
Epoch 230/500
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8851 - loss: 0.3030 - val_accur
acy: 0.8744 - val_loss: 0.3187
```



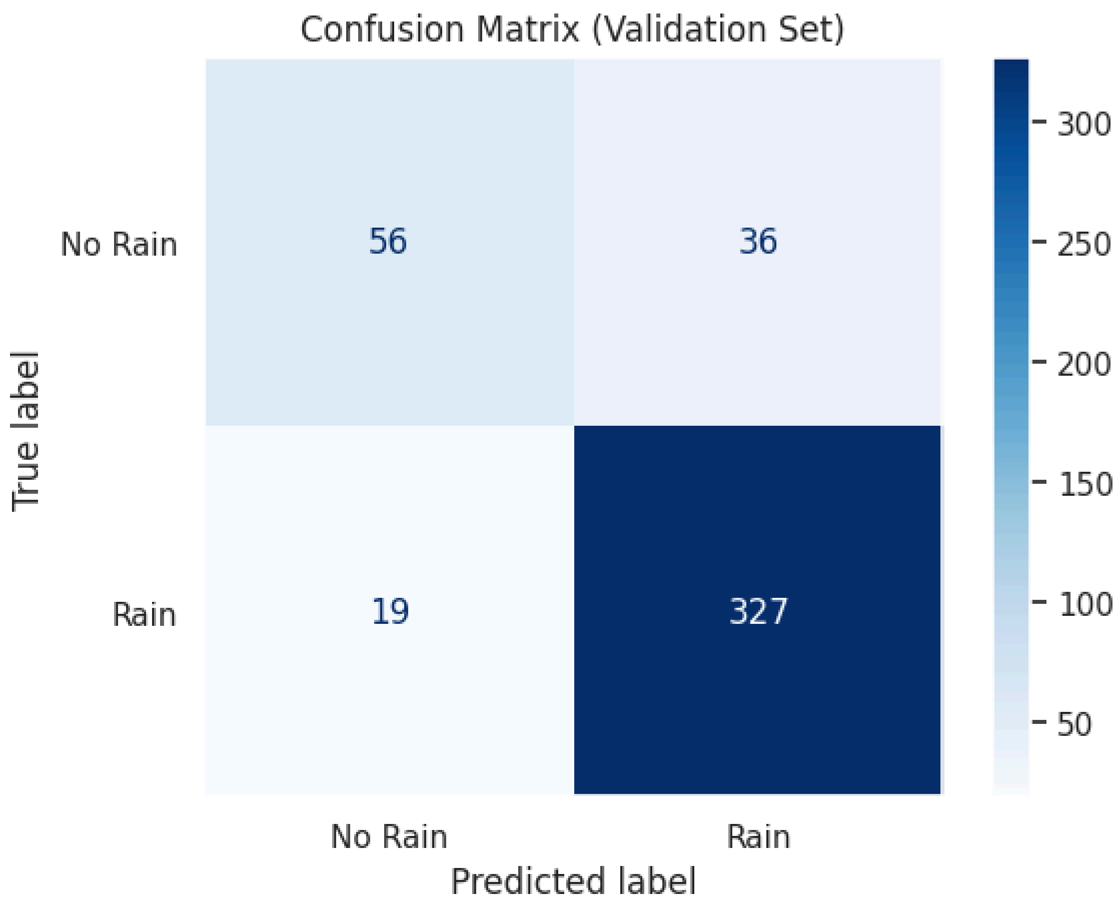

```
Training Accuracy: 0.8807
Validation Accuracy: 0.8744
```

In [280…

```
# Get predicted probabilities on validation set
y_val_probs = nn_model.predict(X_val)

# Convert probabilities to class labels (0 or 1)
y_val_preds = (y_val_probs > 0.5).astype("int32")

# Generate confusion matrix
cm = confusion_matrix(y_val, y_val_preds)

# Display confusion matrix
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["No Rain", "Rain
```

```
disp.plot(cmap=plt.cm.Blues)
plt.title("Confusion Matrix (Validation Set)")
plt.grid(False)
plt.show()

# Print classification report
print("\nClassification Report:")
print(classification_report(y_val, y_val_preds, target_names=["No Rain", "Rain"]))
print("Validation ROC AUC:", roc_auc_score(y_val, y_val_probs))
```

```
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
```




```
Classification Report:
              precision    recall  f1-score   support

     No Rain       0.75      0.61      0.67        92
        Rain       0.90      0.95      0.92       346

    accuracy                           0.87       438
   macro avg       0.82      0.78      0.80       438
weighted avg       0.87      0.87      0.87       438

Validation ROC AUC: 0.8847700427243026
```

In [281…

```
# Get predicted probabilities for the positive class
test_probs = nn_model.predict(X_test_scaled)


submission = pd.DataFrame({
    "id": test_df["id"],
```

```
    "rainfall": test_probs.flatten()
})

# Save to CSV
submission.to_csv(PATH + "submission_nn.csv", index=False)
```

```
23/23 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
```

1D CNN

In [282…

```
# CNN
window_size = 8

# Function to create sequences
def create_sequences(X, y, window_size):
    X_seq = []
    y_seq = []
    for i in range(len(X) - window_size + 1):
        X_seq.append(X[i : i + window_size])  # 8-day window
        y_seq.append(y[i + window_size - 1])  # label of last day in window
    return np.array(X_seq), np.array(y_seq)

# Generate sequences
X_seq, y_seq = create_sequences(X.values, y.values, window_size=window_size)

# Reshape X_seq for scaling: (samples * window_size, num_features)
num_samples, num_days, num_features = X_seq.shape
X_seq_2d = X_seq.reshape(-1, num_features)

# Scale the 2D version
scaler = StandardScaler()
X_seq_scaled_2d = scaler.fit_transform(X_seq_2d)

# Reshape back to (samples, window_size, num_features)
X_seq_scaled = X_seq_scaled_2d.reshape(num_samples, num_days, num_features)

# Train-validation split
# Time-based train-validation split (80% train, 20% validation)
split_index = int(len(X_seq_scaled) * 0.8)
X_train_seq = X_seq_scaled[:split_index]
X_val_seq = X_seq_scaled[split_index:]
y_train_seq = y_seq[:split_index]
y_val_seq = y_seq[split_index:]
```

In [283…

```
print(X_train_seq[0])
print("X_seq shape:", X_train_seq.shape)
print("X_seq shape:", X_val_seq.shape)
```

```
[[ 0.67397196 -0.91887634 -0.64644794 -0.45274503  0.63892102  0.68217836
  -0.73087155 -0.56263476 -0.46379094 -1.46653978 -0.02011487  0.82078886]
 [ 1.0453886  -1.80436923 -1.35564335 -1.26390571  1.66670316  0.84860842
  -1.03418993 -0.68755941  0.01144836 -1.46653978  0.00415576  0.66770424]
 [ 1.8589679  -1.23765378 -1.5089829  -1.50131859 -0.90275218 -1.59236577
   1.25448511 -0.43771011 -0.37278767 -1.46653978  0.0284192   0.93358352]
 [-0.0334883  -1.46788193 -1.18313636 -1.04627724  1.66670316  1.07051516
  -1.03418993 -0.56263476  1.39672037 -1.46653978  0.05266826  0.82078886]
 [ 1.45217825 -0.90116648 -1.06813169 -1.38261215 -3.85762582 -1.70331915
  -0.0415116  -0.81248406  0.30468112 -1.46653978  0.07689575  0.47898105]
 [ 1.6113568  -1.02513549 -1.02979681 -1.12541486 -0.38886111  0.29384156
  -1.03418993 -1.06233336 -0.61546306 -1.46653978  0.1010945   0.01846462]
 [ 1.62904331 -1.21994392 -1.06813169 -1.36282774 -3.34373475 -1.64784246
   1.06146433 -1.06233336  0.6686942  -1.46653978  0.12525733  0.01846462]
 [ 1.08076161 -1.87520867 -1.98816898 -1.87722232  1.79517593  1.3478986
  -1.03418993 -0.68755941  3.13589398 -1.46653978  0.14937708  0.66770424]]
X_seq shape: (1746, 8, 12)
X_seq shape: (437, 8, 12)
```

In [284…

```python
print(y_train_seq.shape)
y_val_seq.shape
print("X_train_seq shape:", X_seq.shape)
print("y_train shape:", y_seq.shape)
```

```
(1746,)
X_train_seq shape: (2183, 8, 12)
y_train shape: (2183,)
```

In [285…

```python
# Clear any existing model
tf.keras.backend.clear_session()

# Reshape input for Conv1D: (samples, timesteps, features)
# If you already have X_train_seq and X_val_seq shaped appropriately:
# (e.g., X_train_seq.shape = (num_samples, time_steps, num_features))

# Build Conv1D model
model_1d_cnn = tf.keras.Sequential()

# Add 1D convolutional layer
model_1d_cnn.add(tf.keras.layers.Conv1D(
    filters=128,
    kernel_size=4,
    strides=1,
    padding='same',
    activation='relu',
    input_shape=(X_train_seq.shape[1], X_train_seq.shape[2]),  # (timesteps, featur
    name='conv1d_1'
))

# Add 1D max pooling
model_1d_cnn.add(tf.keras.layers.MaxPooling1D(pool_size=2))

# Dropout for regularization
model_1d_cnn.add(tf.keras.layers.Dropout(0.5))

# Flatten and output
```

```
model_1d_cnn.add(tf.keras.layers.Flatten())
model_1d_cnn.add(tf.keras.layers.Dense(1, activation='sigmoid'))

# Compile model
model_1d_cnn.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=0.0005),
    loss='binary_crossentropy',
    metrics=['accuracy']
)

# Print model summary
model_1d_cnn.summary()
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

Model: "sequential"

Layer (type)	Output Shape	
conv1d_1 (Conv1D)	(None, 8, 128)	
max_pooling1d (MaxPooling1D)	(None, 4, 128)	
dropout (Dropout)	(None, 4, 128)	
flatten (Flatten)	(None, 512)	
dense (Dense)	(None, 1)	

Total params: 6,785 (26.50 KB)

Trainable params: 6,785 (26.50 KB)

Non-trainable params: 0 (0.00 B)

In [286…

```
early_stopping = tf.keras.callbacks.EarlyStopping(
    monitor='val_loss',
    patience=100,
    restore_best_weights=True,
    verbose=1
)

history = model_1d_cnn.fit(
    X_train_seq, y_train_seq,
    validation_data=(X_val_seq, y_val_seq),
    epochs=300,
    batch_size=1024,
    callbacks=[early_stopping]
)

# Plot losses
plt.figure(figsize=(8, 5))
```

```
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Val Loss')
plt.title("Training & Validation Loss")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.grid(True)
plt.legend()
plt.show()
```

```
Epoch 1/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 1s 269ms/step - accuracy: 0.4746 - loss: 0.7532 - val_accur
acy: 0.4027 - val_loss: 0.7406
Epoch 2/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 78ms/step - accuracy: 0.5548 - loss: 0.6897 - val_accura
cy: 0.5492 - val_loss: 0.6866
Epoch 3/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.6228 - loss: 0.6594 - val_accura
cy: 0.6796 - val_loss: 0.6416
Epoch 4/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.6701 - loss: 0.6222 - val_accura
cy: 0.7346 - val_loss: 0.6048
Epoch 5/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.7115 - loss: 0.5906 - val_accura
cy: 0.7826 - val_loss: 0.5753
Epoch 6/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.7133 - loss: 0.5908 - val_accura
cy: 0.7826 - val_loss: 0.5521
Epoch 7/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.7248 - loss: 0.5735 - val_accura
cy: 0.7918 - val_loss: 0.5337
Epoch 8/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.7484 - loss: 0.5523 - val_accura
cy: 0.7941 - val_loss: 0.5193
Epoch 9/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.7526 - loss: 0.5546 - val_accura
cy: 0.7941 - val_loss: 0.5078
Epoch 10/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.7516 - loss: 0.5392 - val_accura
cy: 0.7941 - val_loss: 0.4985
Epoch 11/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.7540 - loss: 0.5341 - val_accura
cy: 0.7963 - val_loss: 0.4907
Epoch 12/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.7527 - loss: 0.5469 - val_accura
cy: 0.7963 - val_loss: 0.4841
Epoch 13/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.7551 - loss: 0.5164 - val_accura
cy: 0.7963 - val_loss: 0.4784
Epoch 14/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.7552 - loss: 0.5143 - val_accura
cy: 0.7918 - val_loss: 0.4734
Epoch 15/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.7548 - loss: 0.5184 - val_accura
cy: 0.8032 - val_loss: 0.4689
Epoch 16/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.7565 - loss: 0.5078 - val_accura
cy: 0.8032 - val_loss: 0.4650
Epoch 17/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.7628 - loss: 0.5068 - val_accura
cy: 0.8078 - val_loss: 0.4615
Epoch 18/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.7608 - loss: 0.4881 - val_accura
cy: 0.8101 - val_loss: 0.4581
Epoch 19/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.7651 - loss: 0.4840 - val_accura
```

```
cy: 0.8078 - val_loss: 0.4544
Epoch 20/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.7681 - loss: 0.4878 - val_accura
cy: 0.8146 - val_loss: 0.4505
Epoch 21/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.7698 - loss: 0.4789 - val_accura
cy: 0.8146 - val_loss: 0.4463
Epoch 22/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.7838 - loss: 0.4718 - val_accura
cy: 0.8146 - val_loss: 0.4420
Epoch 23/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.7933 - loss: 0.4688 - val_accura
cy: 0.8146 - val_loss: 0.4379
Epoch 24/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.7833 - loss: 0.4661 - val_accura
cy: 0.8146 - val_loss: 0.4337
Epoch 25/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.7949 - loss: 0.4597 - val_accura
cy: 0.8169 - val_loss: 0.4298
Epoch 26/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 72ms/step - accuracy: 0.7824 - loss: 0.4710 - val_accura
cy: 0.8169 - val_loss: 0.4259
Epoch 27/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 72ms/step - accuracy: 0.7969 - loss: 0.4624 - val_accura
cy: 0.8169 - val_loss: 0.4221
Epoch 28/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.7870 - loss: 0.4595 - val_accura
cy: 0.8169 - val_loss: 0.4185
Epoch 29/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.7920 - loss: 0.4488 - val_accura
cy: 0.8169 - val_loss: 0.4153
Epoch 30/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 78ms/step - accuracy: 0.7773 - loss: 0.4494 - val_accura
cy: 0.8169 - val_loss: 0.4120
Epoch 31/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.7950 - loss: 0.4452 - val_accura
cy: 0.8192 - val_loss: 0.4089
Epoch 32/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.7935 - loss: 0.4390 - val_accura
cy: 0.8192 - val_loss: 0.4061
Epoch 33/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 79ms/step - accuracy: 0.8041 - loss: 0.4390 - val_accura
cy: 0.8192 - val_loss: 0.4035
Epoch 34/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8062 - loss: 0.4373 - val_accura
cy: 0.8192 - val_loss: 0.4011
Epoch 35/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8082 - loss: 0.4413 - val_accura
cy: 0.8192 - val_loss: 0.3988
Epoch 36/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8059 - loss: 0.4275 - val_accura
cy: 0.8192 - val_loss: 0.3967
Epoch 37/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8059 - loss: 0.4273 - val_accura
cy: 0.8238 - val_loss: 0.3946
Epoch 38/300
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.8135 - loss: 0.4377 - val_accura
cy: 0.8238 - val_loss: 0.3924
Epoch 39/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8128 - loss: 0.4197 - val_accura
cy: 0.8238 - val_loss: 0.3904
Epoch 40/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8187 - loss: 0.4166 - val_accura
cy: 0.8261 - val_loss: 0.3882
Epoch 41/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8138 - loss: 0.4143 - val_accura
cy: 0.8261 - val_loss: 0.3861
Epoch 42/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 78ms/step - accuracy: 0.8269 - loss: 0.4094 - val_accura
cy: 0.8261 - val_loss: 0.3840
Epoch 43/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 84ms/step - accuracy: 0.8155 - loss: 0.4166 - val_accura
cy: 0.8261 - val_loss: 0.3819
Epoch 44/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8263 - loss: 0.4080 - val_accura
cy: 0.8284 - val_loss: 0.3801
Epoch 45/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.8223 - loss: 0.4107 - val_accura
cy: 0.8307 - val_loss: 0.3785
Epoch 46/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 80ms/step - accuracy: 0.8207 - loss: 0.4119 - val_accura
cy: 0.8330 - val_loss: 0.3772
Epoch 47/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8219 - loss: 0.4064 - val_accura
cy: 0.8375 - val_loss: 0.3760
Epoch 48/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.8277 - loss: 0.4066 - val_accura
cy: 0.8398 - val_loss: 0.3746
Epoch 49/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 79ms/step - accuracy: 0.8262 - loss: 0.4089 - val_accura
cy: 0.8398 - val_loss: 0.3734
Epoch 50/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 83ms/step - accuracy: 0.8294 - loss: 0.4066 - val_accura
cy: 0.8398 - val_loss: 0.3720
Epoch 51/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 84ms/step - accuracy: 0.8311 - loss: 0.3908 - val_accura
cy: 0.8398 - val_loss: 0.3707
Epoch 52/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8292 - loss: 0.3876 - val_accura
cy: 0.8398 - val_loss: 0.3695
Epoch 53/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8216 - loss: 0.3982 - val_accura
cy: 0.8444 - val_loss: 0.3684
Epoch 54/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 83ms/step - accuracy: 0.8335 - loss: 0.3924 - val_accura
cy: 0.8444 - val_loss: 0.3671
Epoch 55/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8340 - loss: 0.3879 - val_accura
cy: 0.8467 - val_loss: 0.3654
Epoch 56/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.8255 - loss: 0.3903 - val_accura
cy: 0.8467 - val_loss: 0.3638
```

```
Epoch 57/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8337 - loss: 0.3904 - val_accura
cy: 0.8513 - val_loss: 0.3622
Epoch 58/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 81ms/step - accuracy: 0.8348 - loss: 0.3878 - val_accura
cy: 0.8535 - val_loss: 0.3609
Epoch 59/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 78ms/step - accuracy: 0.8360 - loss: 0.3792 - val_accura
cy: 0.8558 - val_loss: 0.3595
Epoch 60/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.8400 - loss: 0.3754 - val_accura
cy: 0.8513 - val_loss: 0.3583
Epoch 61/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 86ms/step - accuracy: 0.8408 - loss: 0.3771 - val_accura
cy: 0.8535 - val_loss: 0.3572
Epoch 62/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 79ms/step - accuracy: 0.8434 - loss: 0.3781 - val_accura
cy: 0.8513 - val_loss: 0.3562
Epoch 63/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 87ms/step - accuracy: 0.8473 - loss: 0.3758 - val_accura
cy: 0.8513 - val_loss: 0.3553
Epoch 64/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8429 - loss: 0.3786 - val_accura
cy: 0.8513 - val_loss: 0.3545
Epoch 65/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8478 - loss: 0.3680 - val_accura
cy: 0.8513 - val_loss: 0.3538
Epoch 66/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8480 - loss: 0.3730 - val_accura
cy: 0.8513 - val_loss: 0.3532
Epoch 67/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8472 - loss: 0.3666 - val_accura
cy: 0.8513 - val_loss: 0.3525
Epoch 68/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8614 - loss: 0.3628 - val_accura
cy: 0.8467 - val_loss: 0.3518
Epoch 69/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8493 - loss: 0.3575 - val_accura
cy: 0.8467 - val_loss: 0.3508
Epoch 70/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8487 - loss: 0.3707 - val_accura
cy: 0.8490 - val_loss: 0.3498
Epoch 71/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.8467 - loss: 0.3609 - val_accura
cy: 0.8490 - val_loss: 0.3488
Epoch 72/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8459 - loss: 0.3701 - val_accura
cy: 0.8513 - val_loss: 0.3478
Epoch 73/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8561 - loss: 0.3548 - val_accura
cy: 0.8558 - val_loss: 0.3468
Epoch 74/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8542 - loss: 0.3575 - val_accura
cy: 0.8558 - val_loss: 0.3459
Epoch 75/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 78ms/step - accuracy: 0.8562 - loss: 0.3660 - val_accura
```

```
cy: 0.8604 - val_loss: 0.3450
Epoch 76/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8474 - loss: 0.3595 - val_accura
cy: 0.8650 - val_loss: 0.3443
Epoch 77/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8553 - loss: 0.3530 - val_accura
cy: 0.8627 - val_loss: 0.3439
Epoch 78/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8567 - loss: 0.3503 - val_accura
cy: 0.8627 - val_loss: 0.3438
Epoch 79/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8516 - loss: 0.3519 - val_accura
cy: 0.8604 - val_loss: 0.3438
Epoch 80/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.8538 - loss: 0.3441 - val_accura
cy: 0.8627 - val_loss: 0.3438
Epoch 81/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8546 - loss: 0.3443 - val_accura
cy: 0.8627 - val_loss: 0.3434
Epoch 82/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8619 - loss: 0.3452 - val_accura
cy: 0.8627 - val_loss: 0.3429
Epoch 83/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8547 - loss: 0.3498 - val_accura
cy: 0.8627 - val_loss: 0.3421
Epoch 84/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8561 - loss: 0.3424 - val_accura
cy: 0.8673 - val_loss: 0.3412
Epoch 85/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8582 - loss: 0.3401 - val_accura
cy: 0.8696 - val_loss: 0.3403
Epoch 86/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8588 - loss: 0.3417 - val_accura
cy: 0.8696 - val_loss: 0.3396
Epoch 87/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.8632 - loss: 0.3367 - val_accura
cy: 0.8696 - val_loss: 0.3392
Epoch 88/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8602 - loss: 0.3399 - val_accura
cy: 0.8696 - val_loss: 0.3391
Epoch 89/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 78ms/step - accuracy: 0.8617 - loss: 0.3445 - val_accura
cy: 0.8696 - val_loss: 0.3390
Epoch 90/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.8650 - loss: 0.3375 - val_accura
cy: 0.8696 - val_loss: 0.3389
Epoch 91/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8618 - loss: 0.3409 - val_accura
cy: 0.8696 - val_loss: 0.3386
Epoch 92/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.8589 - loss: 0.3361 - val_accura
cy: 0.8696 - val_loss: 0.3383
Epoch 93/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8590 - loss: 0.3389 - val_accura
cy: 0.8696 - val_loss: 0.3379
Epoch 94/300
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8617 - loss: 0.3458 - val_accura
cy: 0.8696 - val_loss: 0.3374
Epoch 95/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8538 - loss: 0.3483 - val_accura
cy: 0.8719 - val_loss: 0.3371
Epoch 96/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8754 - loss: 0.3269 - val_accura
cy: 0.8719 - val_loss: 0.3366
Epoch 97/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8538 - loss: 0.3366 - val_accura
cy: 0.8673 - val_loss: 0.3362
Epoch 98/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8654 - loss: 0.3337 - val_accura
cy: 0.8719 - val_loss: 0.3359
Epoch 99/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 78ms/step - accuracy: 0.8682 - loss: 0.3382 - val_accura
cy: 0.8719 - val_loss: 0.3356
Epoch 100/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8623 - loss: 0.3324 - val_accura
cy: 0.8719 - val_loss: 0.3355
Epoch 101/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8697 - loss: 0.3293 - val_accura
cy: 0.8719 - val_loss: 0.3353
Epoch 102/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8612 - loss: 0.3467 - val_accura
cy: 0.8719 - val_loss: 0.3351
Epoch 103/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8711 - loss: 0.3276 - val_accura
cy: 0.8719 - val_loss: 0.3350
Epoch 104/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8648 - loss: 0.3312 - val_accura
cy: 0.8741 - val_loss: 0.3348
Epoch 105/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8640 - loss: 0.3273 - val_accura
cy: 0.8741 - val_loss: 0.3345
Epoch 106/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8634 - loss: 0.3277 - val_accura
cy: 0.8741 - val_loss: 0.3341
Epoch 107/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8579 - loss: 0.3298 - val_accura
cy: 0.8719 - val_loss: 0.3335
Epoch 108/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8697 - loss: 0.3248 - val_accura
cy: 0.8719 - val_loss: 0.3328
Epoch 109/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8638 - loss: 0.3267 - val_accura
cy: 0.8719 - val_loss: 0.3323
Epoch 110/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8682 - loss: 0.3264 - val_accura
cy: 0.8741 - val_loss: 0.3318
Epoch 111/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8651 - loss: 0.3325 - val_accura
cy: 0.8741 - val_loss: 0.3317
Epoch 112/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8766 - loss: 0.3235 - val_accura
cy: 0.8719 - val_loss: 0.3318
```

```
Epoch 113/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8671 - loss: 0.3215 - val_accura
cy: 0.8741 - val_loss: 0.3322
Epoch 114/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8687 - loss: 0.3257 - val_accura
cy: 0.8764 - val_loss: 0.3327
Epoch 115/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.8599 - loss: 0.3231 - val_accura
cy: 0.8787 - val_loss: 0.3333
Epoch 116/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8721 - loss: 0.3204 - val_accura
cy: 0.8810 - val_loss: 0.3335
Epoch 117/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.8655 - loss: 0.3180 - val_accura
cy: 0.8810 - val_loss: 0.3331
Epoch 118/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.8791 - loss: 0.3183 - val_accura
cy: 0.8787 - val_loss: 0.3323
Epoch 119/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.8720 - loss: 0.3189 - val_accura
cy: 0.8787 - val_loss: 0.3316
Epoch 120/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8715 - loss: 0.3189 - val_accura
cy: 0.8787 - val_loss: 0.3313
Epoch 121/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8664 - loss: 0.3235 - val_accura
cy: 0.8764 - val_loss: 0.3311
Epoch 122/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.8792 - loss: 0.3139 - val_accura
cy: 0.8764 - val_loss: 0.3309
Epoch 123/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8763 - loss: 0.3111 - val_accura
cy: 0.8764 - val_loss: 0.3307
Epoch 124/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.8720 - loss: 0.3112 - val_accura
cy: 0.8764 - val_loss: 0.3306
Epoch 125/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 79ms/step - accuracy: 0.8754 - loss: 0.3156 - val_accura
cy: 0.8764 - val_loss: 0.3306
Epoch 126/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 71ms/step - accuracy: 0.8754 - loss: 0.3160 - val_accura
cy: 0.8810 - val_loss: 0.3306
Epoch 127/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8774 - loss: 0.3187 - val_accura
cy: 0.8833 - val_loss: 0.3309
Epoch 128/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8798 - loss: 0.3112 - val_accura
cy: 0.8833 - val_loss: 0.3312
Epoch 129/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8787 - loss: 0.3105 - val_accura
cy: 0.8833 - val_loss: 0.3312
Epoch 130/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8742 - loss: 0.3079 - val_accura
cy: 0.8833 - val_loss: 0.3308
Epoch 131/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8721 - loss: 0.3136 - val_accura
```

```
cy: 0.8833 - val_loss: 0.3304
Epoch 132/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.8698 - loss: 0.3157 - val_accura
cy: 0.8833 - val_loss: 0.3300
Epoch 133/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8725 - loss: 0.3242 - val_accura
cy: 0.8833 - val_loss: 0.3296
Epoch 134/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.8741 - loss: 0.3087 - val_accura
cy: 0.8833 - val_loss: 0.3292
Epoch 135/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8676 - loss: 0.3154 - val_accura
cy: 0.8833 - val_loss: 0.3290
Epoch 136/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 80ms/step - accuracy: 0.8787 - loss: 0.3038 - val_accura
cy: 0.8833 - val_loss: 0.3291
Epoch 137/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8746 - loss: 0.3145 - val_accura
cy: 0.8833 - val_loss: 0.3294
Epoch 138/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8773 - loss: 0.3062 - val_accura
cy: 0.8833 - val_loss: 0.3297
Epoch 139/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8803 - loss: 0.3048 - val_accura
cy: 0.8833 - val_loss: 0.3304
Epoch 140/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 72ms/step - accuracy: 0.8696 - loss: 0.3058 - val_accura
cy: 0.8833 - val_loss: 0.3311
Epoch 141/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8742 - loss: 0.3030 - val_accura
cy: 0.8810 - val_loss: 0.3318
Epoch 142/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8778 - loss: 0.3103 - val_accura
cy: 0.8810 - val_loss: 0.3320
Epoch 143/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8786 - loss: 0.3065 - val_accura
cy: 0.8810 - val_loss: 0.3319
Epoch 144/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8747 - loss: 0.3092 - val_accura
cy: 0.8810 - val_loss: 0.3315
Epoch 145/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8793 - loss: 0.3063 - val_accura
cy: 0.8810 - val_loss: 0.3310
Epoch 146/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 72ms/step - accuracy: 0.8782 - loss: 0.3022 - val_accura
cy: 0.8810 - val_loss: 0.3305
Epoch 147/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.8730 - loss: 0.3166 - val_accura
cy: 0.8810 - val_loss: 0.3302
Epoch 148/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8752 - loss: 0.3169 - val_accura
cy: 0.8810 - val_loss: 0.3302
Epoch 149/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8786 - loss: 0.3058 - val_accura
cy: 0.8810 - val_loss: 0.3303
Epoch 150/300
```

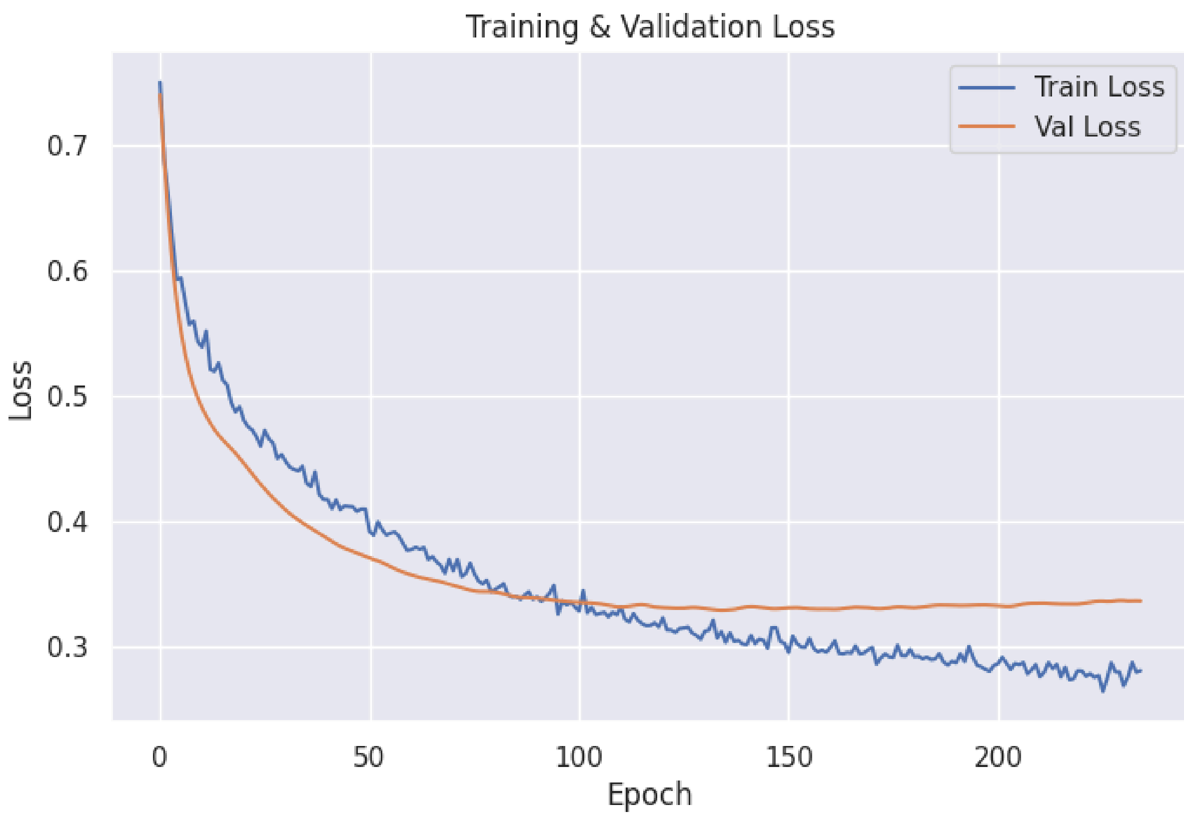
```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8691 - loss: 0.3062 - val_accura
cy: 0.8810 - val_loss: 0.3306
Epoch 151/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8783 - loss: 0.2981 - val_accura
cy: 0.8810 - val_loss: 0.3309
Epoch 152/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.8743 - loss: 0.3090 - val_accura
cy: 0.8810 - val_loss: 0.3310
Epoch 153/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8784 - loss: 0.3040 - val_accura
cy: 0.8810 - val_loss: 0.3310
Epoch 154/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 84ms/step - accuracy: 0.8778 - loss: 0.3000 - val_accura
cy: 0.8810 - val_loss: 0.3307
Epoch 155/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 78ms/step - accuracy: 0.8759 - loss: 0.3029 - val_accura
cy: 0.8833 - val_loss: 0.3305
Epoch 156/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 79ms/step - accuracy: 0.8713 - loss: 0.3077 - val_accura
cy: 0.8833 - val_loss: 0.3302
Epoch 157/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.8848 - loss: 0.2992 - val_accura
cy: 0.8833 - val_loss: 0.3300
Epoch 158/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8836 - loss: 0.2984 - val_accura
cy: 0.8833 - val_loss: 0.3299
Epoch 159/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 78ms/step - accuracy: 0.8778 - loss: 0.2975 - val_accura
cy: 0.8833 - val_loss: 0.3300
Epoch 160/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 81ms/step - accuracy: 0.8792 - loss: 0.2965 - val_accura
cy: 0.8833 - val_loss: 0.3300
Epoch 161/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 80ms/step - accuracy: 0.8810 - loss: 0.3008 - val_accura
cy: 0.8833 - val_loss: 0.3299
Epoch 162/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8751 - loss: 0.3068 - val_accura
cy: 0.8833 - val_loss: 0.3298
Epoch 163/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8811 - loss: 0.2977 - val_accura
cy: 0.8810 - val_loss: 0.3300
Epoch 164/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8866 - loss: 0.2960 - val_accura
cy: 0.8810 - val_loss: 0.3304
Epoch 165/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 83ms/step - accuracy: 0.8802 - loss: 0.3000 - val_accura
cy: 0.8810 - val_loss: 0.3307
Epoch 166/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 82ms/step - accuracy: 0.8761 - loss: 0.2978 - val_accura
cy: 0.8810 - val_loss: 0.3312
Epoch 167/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8838 - loss: 0.3034 - val_accura
cy: 0.8810 - val_loss: 0.3315
Epoch 168/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.8844 - loss: 0.2983 - val_accura
cy: 0.8810 - val_loss: 0.3313
```

```
Epoch 169/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 80ms/step - accuracy: 0.8741 - loss: 0.2983 - val_accura
cy: 0.8810 - val_loss: 0.3313
Epoch 170/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 78ms/step - accuracy: 0.8812 - loss: 0.2994 - val_accura
cy: 0.8810 - val_loss: 0.3310
Epoch 171/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 81ms/step - accuracy: 0.8848 - loss: 0.3020 - val_accura
cy: 0.8810 - val_loss: 0.3306
Epoch 172/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 78ms/step - accuracy: 0.8808 - loss: 0.2887 - val_accura
cy: 0.8810 - val_loss: 0.3303
Epoch 173/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 81ms/step - accuracy: 0.8829 - loss: 0.2942 - val_accura
cy: 0.8833 - val_loss: 0.3302
Epoch 174/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8808 - loss: 0.2953 - val_accura
cy: 0.8833 - val_loss: 0.3305
Epoch 175/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8804 - loss: 0.2949 - val_accura
cy: 0.8833 - val_loss: 0.3308
Epoch 176/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8812 - loss: 0.2923 - val_accura
cy: 0.8833 - val_loss: 0.3314
Epoch 177/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8803 - loss: 0.3036 - val_accura
cy: 0.8833 - val_loss: 0.3317
Epoch 178/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8780 - loss: 0.2954 - val_accura
cy: 0.8833 - val_loss: 0.3317
Epoch 179/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8773 - loss: 0.2948 - val_accura
cy: 0.8833 - val_loss: 0.3314
Epoch 180/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8816 - loss: 0.2989 - val_accura
cy: 0.8833 - val_loss: 0.3312
Epoch 181/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8782 - loss: 0.2932 - val_accura
cy: 0.8833 - val_loss: 0.3310
Epoch 182/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.8794 - loss: 0.2958 - val_accura
cy: 0.8833 - val_loss: 0.3313
Epoch 183/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8830 - loss: 0.2932 - val_accura
cy: 0.8833 - val_loss: 0.3317
Epoch 184/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8750 - loss: 0.2943 - val_accura
cy: 0.8833 - val_loss: 0.3322
Epoch 185/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8774 - loss: 0.2920 - val_accura
cy: 0.8833 - val_loss: 0.3325
Epoch 186/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 78ms/step - accuracy: 0.8834 - loss: 0.2920 - val_accura
cy: 0.8833 - val_loss: 0.3329
Epoch 187/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 78ms/step - accuracy: 0.8750 - loss: 0.2986 - val_accura
```

```
cy: 0.8833 - val_loss: 0.3331
Epoch 188/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8837 - loss: 0.2918 - val_accura
cy: 0.8810 - val_loss: 0.3331
Epoch 189/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8887 - loss: 0.2892 - val_accura
cy: 0.8833 - val_loss: 0.3330
Epoch 190/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 80ms/step - accuracy: 0.8864 - loss: 0.2928 - val_accura
cy: 0.8833 - val_loss: 0.3329
Epoch 191/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 78ms/step - accuracy: 0.8824 - loss: 0.2902 - val_accura
cy: 0.8833 - val_loss: 0.3327
Epoch 192/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8771 - loss: 0.2955 - val_accura
cy: 0.8833 - val_loss: 0.3327
Epoch 193/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8867 - loss: 0.2894 - val_accura
cy: 0.8833 - val_loss: 0.3328
Epoch 194/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8741 - loss: 0.3047 - val_accura
cy: 0.8833 - val_loss: 0.3328
Epoch 195/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8789 - loss: 0.2925 - val_accura
cy: 0.8833 - val_loss: 0.3330
Epoch 196/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8841 - loss: 0.2874 - val_accura
cy: 0.8833 - val_loss: 0.3331
Epoch 197/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8885 - loss: 0.2876 - val_accura
cy: 0.8833 - val_loss: 0.3331
Epoch 198/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8834 - loss: 0.2840 - val_accura
cy: 0.8833 - val_loss: 0.3332
Epoch 199/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8894 - loss: 0.2856 - val_accura
cy: 0.8833 - val_loss: 0.3330
Epoch 200/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 79ms/step - accuracy: 0.8807 - loss: 0.2890 - val_accura
cy: 0.8833 - val_loss: 0.3328
Epoch 201/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.8821 - loss: 0.2890 - val_accura
cy: 0.8833 - val_loss: 0.3325
Epoch 202/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8771 - loss: 0.2955 - val_accura
cy: 0.8833 - val_loss: 0.3323
Epoch 203/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8832 - loss: 0.2905 - val_accura
cy: 0.8810 - val_loss: 0.3320
Epoch 204/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8829 - loss: 0.2881 - val_accura
cy: 0.8810 - val_loss: 0.3321
Epoch 205/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8824 - loss: 0.2879 - val_accura
cy: 0.8810 - val_loss: 0.3326
Epoch 206/300
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8842 - loss: 0.2893 - val_accura
cy: 0.8810 - val_loss: 0.3333
Epoch 207/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 78ms/step - accuracy: 0.8847 - loss: 0.2890 - val_accura
cy: 0.8833 - val_loss: 0.3339
Epoch 208/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.8821 - loss: 0.2803 - val_accura
cy: 0.8833 - val_loss: 0.3344
Epoch 209/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8791 - loss: 0.2850 - val_accura
cy: 0.8856 - val_loss: 0.3346
Epoch 210/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8829 - loss: 0.2853 - val_accura
cy: 0.8856 - val_loss: 0.3346
Epoch 211/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8865 - loss: 0.2788 - val_accura
cy: 0.8856 - val_loss: 0.3346
Epoch 212/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8842 - loss: 0.2825 - val_accura
cy: 0.8856 - val_loss: 0.3347
Epoch 213/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.8772 - loss: 0.2901 - val_accura
cy: 0.8856 - val_loss: 0.3344
Epoch 214/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8862 - loss: 0.2866 - val_accura
cy: 0.8856 - val_loss: 0.3343
Epoch 215/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8802 - loss: 0.2902 - val_accura
cy: 0.8856 - val_loss: 0.3341
Epoch 216/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8854 - loss: 0.2807 - val_accura
cy: 0.8833 - val_loss: 0.3340
Epoch 217/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8854 - loss: 0.2866 - val_accura
cy: 0.8810 - val_loss: 0.3340
Epoch 218/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8933 - loss: 0.2761 - val_accura
cy: 0.8810 - val_loss: 0.3339
Epoch 219/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.8883 - loss: 0.2762 - val_accura
cy: 0.8810 - val_loss: 0.3339
Epoch 220/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8925 - loss: 0.2856 - val_accura
cy: 0.8810 - val_loss: 0.3339
Epoch 221/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.8801 - loss: 0.2857 - val_accura
cy: 0.8810 - val_loss: 0.3344
Epoch 222/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8845 - loss: 0.2802 - val_accura
cy: 0.8810 - val_loss: 0.3347
Epoch 223/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8878 - loss: 0.2783 - val_accura
cy: 0.8833 - val_loss: 0.3353
Epoch 224/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8897 - loss: 0.2778 - val_accura
cy: 0.8833 - val_loss: 0.3359
```

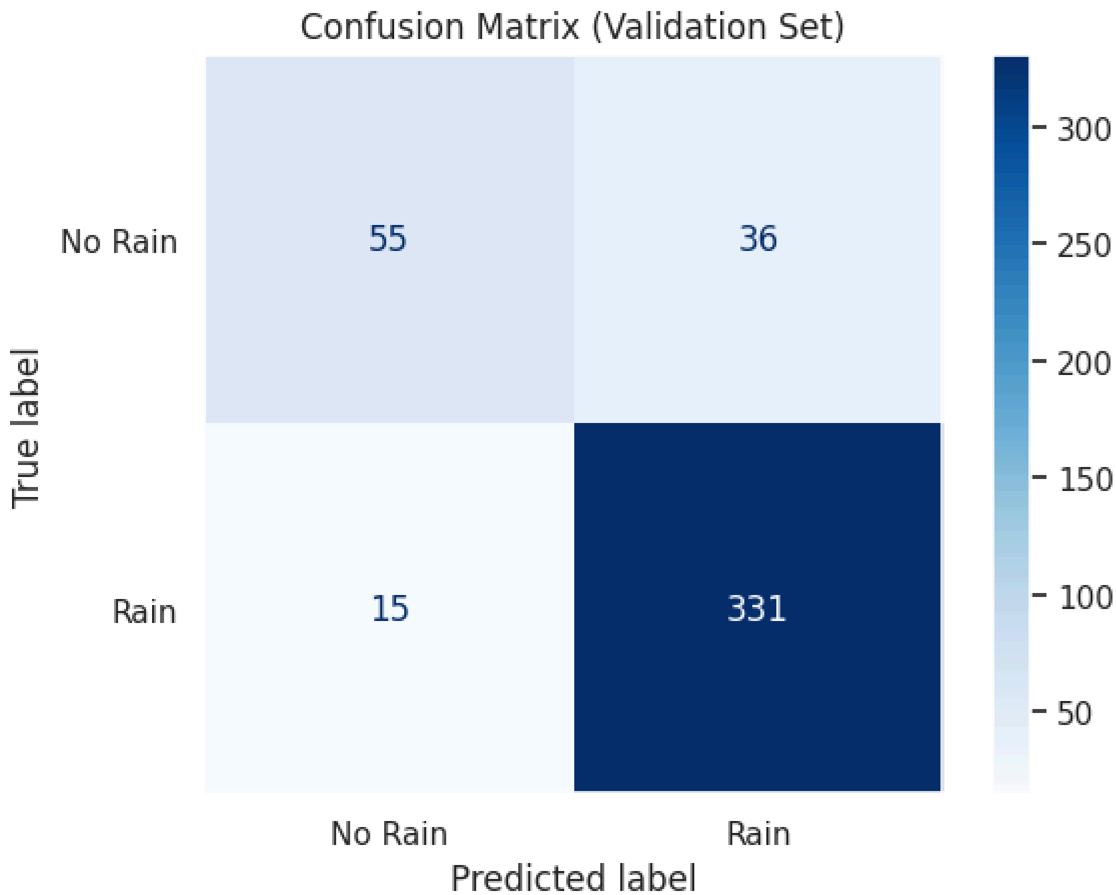
```
Epoch 225/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 73ms/step - accuracy: 0.8842 - loss: 0.2808 - val_accura
cy: 0.8856 - val_loss: 0.3362
Epoch 226/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8919 - loss: 0.2685 - val_accura
cy: 0.8856 - val_loss: 0.3362
Epoch 227/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8858 - loss: 0.2780 - val_accura
cy: 0.8833 - val_loss: 0.3360
Epoch 228/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8796 - loss: 0.2900 - val_accura
cy: 0.8856 - val_loss: 0.3360
Epoch 229/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8890 - loss: 0.2822 - val_accura
cy: 0.8879 - val_loss: 0.3365
Epoch 230/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8828 - loss: 0.2829 - val_accura
cy: 0.8879 - val_loss: 0.3367
Epoch 231/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 77ms/step - accuracy: 0.8788 - loss: 0.2725 - val_accura
cy: 0.8879 - val_loss: 0.3367
Epoch 232/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 76ms/step - accuracy: 0.8839 - loss: 0.2788 - val_accura
cy: 0.8879 - val_loss: 0.3363
Epoch 233/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 74ms/step - accuracy: 0.8742 - loss: 0.2928 - val_accura
cy: 0.8879 - val_loss: 0.3364
Epoch 234/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8858 - loss: 0.2819 - val_accura
cy: 0.8879 - val_loss: 0.3364
Epoch 235/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 75ms/step - accuracy: 0.8797 - loss: 0.2837 - val_accura
cy: 0.8879 - val_loss: 0.3363
Epoch 235: early stopping
Restoring model weights from the end of the best epoch: 135.
```



In [287…

```
# Evaluate
train_loss, train_acc = model_1d_cnn.evaluate(X_train_seq, y_train_seq)
val_loss, val_acc = model_1d_cnn.evaluate(X_val_seq, y_val_seq)

print(f"Training Accuracy: {train_acc:.4f}")
print(f"Validation Accuracy: {val_acc:.4f}")

# Predict probabilities
y_val_probs = model_1d_cnn.predict(X_val_seq)

# Convert probabilities to binary predictions
y_val_preds = (y_val_probs > 0.5).astype("int32")

# Generate confusion matrix
cm = confusion_matrix(y_val_seq, y_val_preds)

# Display
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["No Rain", "Rain
disp.plot(cmap=plt.cm.Blues)
plt.title("Confusion Matrix (Validation Set)")
plt.grid(False)
plt.show()
```

```
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8809 - loss: 0.2992
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8846 - loss: 0.3343
Training Accuracy: 0.8877
Validation Accuracy: 0.8833
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
```



In [288…

```
# Load the test data
test_df = pd.read_csv(PATH + 'test_extra7.csv')


test_df['date'] = pd.to_datetime(test_df['day'], format='%j', errors='coerce')

# Simulate year assignment just like train_df (e.g., assume up to 6 years of data)
test_df['year'] = (test_df.index // 365)

# Extract month from synthetic date
test_df['month'] = test_df['date'].dt.month

# Create cyclical features
test_df['day_sin'] = np.sin(2 * np.pi * (test_df['day'] - 1) / 365)
test_df['wind_sin'] = np.sin(2 * np.pi * test_df['winddirection'] / 360)


# Step 2: Select the same feature columns
X_test = test_df[columns_to_keep]

# Step 3: Scale using the same scaler
X_test_scaled = scaler.transform(X_test)

# Create sequences
X_test_seq, _ = create_sequences(X_test.values, np.zeros(len(X_test)), window_size=

# Reshape for scaling
num_samples_test, num_days_test, num_features_test = X_test_seq.shape
```

```
X_test_2d = X_test_seq.reshape(-1, num_features_test)

# Apply the SAME scaler from training
X_test_scaled_2d = scaler.transform(X_test_2d)

# Reshape back to 3D for CNN
X_test_cnn = X_test_scaled_2d.reshape(num_samples_test, num_days_test, num_features

# Make predictions
y_test_pred = model_1d_cnn.predict(X_test_cnn).flatten()

# Align with correct IDs (assume ID starts from index 7 after 8-day sequences)
submission_ids = test_df['id'].iloc[window_size - 1:].reset_index(drop=True)

# Build submission DataFrame
submission = pd.DataFrame({
    'id': submission_ids,
    'rainfall': y_test_pred
})

# Save to CSV
submission.to_csv(PATH + 'submission_1d_cnn.csv', index=False)
```

```
23/23 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2732: UserWarnin
g: X has feature names, but StandardScaler was fitted without feature names
  warnings.warn(
```

2D CNN

In [289…

```
# One Layer CNN
tf.keras.backend.clear_session()

# Reshape input for Conv2D: (samples, height, width, channels)
X_train_2d = X_train_seq.reshape(-1, window_size, X.shape[1], 1)
X_val_2d = X_val_seq.reshape(-1, window_size, X.shape[1], 1)

# Build Conv2D model
tf.keras.backend.clear_session()
model_2d_cnn = tf.keras.Sequential()

# Add convolutional layer
model_2d_cnn.add(tf.keras.layers.Conv2D(
    filters=128,
    kernel_size=(4, 4),
    strides=(1, 1),
    padding='same',
    data_format='channels_last',
    activation='relu',
    name='conv_1',
    input_shape=(window_size, X.shape[1], 1)  # (height, width, channels)
))

# Add max pooling layer
model_2d_cnn.add(tf.keras.layers.MaxPooling2D(pool_size=(2, 2)))
```

```
# Add dropout layer
model_2d_cnn.add(tf.keras.layers.Dropout(rate=0.5))

# Add flattening layer
model_2d_cnn.add(tf.keras.layers.Flatten())

# Add classification layer
model_2d_cnn.add(tf.keras.layers.Dense(1, activation='sigmoid'))

# Compile model
model_2d_cnn.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=0.0005),
    loss=tf.keras.losses.BinaryCrossentropy(),
    metrics=['accuracy']
)

# Print summary
model_2d_cnn.summary()
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

Model: "sequential"

Layer (type)	Output Shape
conv_1 (Conv2D)	(None, 8, 12, 128)
max_pooling2d (MaxPooling2D)	(None, 4, 6, 128)
dropout (Dropout)	(None, 4, 6, 128)
flatten (Flatten)	(None, 3072)
dense (Dense)	(None, 1)

Total params: 5,249 (20.50 KB)

Trainable params: 5,249 (20.50 KB)

Non-trainable params: 0 (0.00 B)

In [290…

```
early_stopping = tf.keras.callbacks.EarlyStopping(
    monitor='val_loss',
    patience=100,
    restore_best_weights=True,
    verbose=1
)

history = model_2d_cnn.fit(
    X_train_seq, y_train_seq,
    validation_data=(X_val_seq, y_val_seq),
```

```
    epochs=300,
    batch_size=1024,
    callbacks=[early_stopping]
)

# Plot losses
plt.figure(figsize=(8, 5))
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Val Loss')
plt.title("Training & Validation Loss")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.grid(True)
plt.legend()
plt.show()
```

```
Epoch 1/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 1s 310ms/step - accuracy: 0.5480 - loss: 0.6880 - val_accur
acy: 0.7918 - val_loss: 0.6048
Epoch 2/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 140ms/step - accuracy: 0.7377 - loss: 0.6296 - val_accur
acy: 0.7918 - val_loss: 0.5565
Epoch 3/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 122ms/step - accuracy: 0.7483 - loss: 0.5916 - val_accur
acy: 0.7918 - val_loss: 0.5319
Epoch 4/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 133ms/step - accuracy: 0.7483 - loss: 0.5798 - val_accur
acy: 0.7918 - val_loss: 0.5212
Epoch 5/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 130ms/step - accuracy: 0.7483 - loss: 0.5816 - val_accur
acy: 0.7918 - val_loss: 0.5157
Epoch 6/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 120ms/step - accuracy: 0.7483 - loss: 0.5757 - val_accur
acy: 0.7918 - val_loss: 0.5105
Epoch 7/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 139ms/step - accuracy: 0.7483 - loss: 0.5689 - val_accur
acy: 0.7918 - val_loss: 0.5045
Epoch 8/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 130ms/step - accuracy: 0.7483 - loss: 0.5619 - val_accur
acy: 0.7918 - val_loss: 0.4985
Epoch 9/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 119ms/step - accuracy: 0.7483 - loss: 0.5536 - val_accur
acy: 0.7918 - val_loss: 0.4934
Epoch 10/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 130ms/step - accuracy: 0.7483 - loss: 0.5419 - val_accur
acy: 0.7918 - val_loss: 0.4898
Epoch 11/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 120ms/step - accuracy: 0.7483 - loss: 0.5341 - val_accur
acy: 0.7918 - val_loss: 0.4874
Epoch 12/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 120ms/step - accuracy: 0.7490 - loss: 0.5316 - val_accur
acy: 0.7918 - val_loss: 0.4852
Epoch 13/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 129ms/step - accuracy: 0.7551 - loss: 0.5268 - val_accur
acy: 0.7918 - val_loss: 0.4823
Epoch 14/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.7545 - loss: 0.5260 - val_accur
acy: 0.7918 - val_loss: 0.4783
Epoch 15/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 125ms/step - accuracy: 0.7562 - loss: 0.5174 - val_accur
acy: 0.7918 - val_loss: 0.4734
Epoch 16/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - accuracy: 0.7530 - loss: 0.5157 - val_accur
acy: 0.7918 - val_loss: 0.4685
Epoch 17/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.7563 - loss: 0.5085 - val_accur
acy: 0.7918 - val_loss: 0.4641
Epoch 18/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 118ms/step - accuracy: 0.7585 - loss: 0.5061 - val_accur
acy: 0.7918 - val_loss: 0.4604
Epoch 19/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.7544 - loss: 0.5037 - val_accur
```

```
acy: 0.7941 - val_loss: 0.4571
Epoch 20/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - accuracy: 0.7577 - loss: 0.4954 - val_accur
acy: 0.7941 - val_loss: 0.4541
Epoch 21/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 120ms/step - accuracy: 0.7585 - loss: 0.4928 - val_accur
acy: 0.7963 - val_loss: 0.4510
Epoch 22/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 118ms/step - accuracy: 0.7625 - loss: 0.4902 - val_accur
acy: 0.7963 - val_loss: 0.4480
Epoch 23/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.7633 - loss: 0.4877 - val_accur
acy: 0.7986 - val_loss: 0.4452
Epoch 24/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.7733 - loss: 0.4818 - val_accur
acy: 0.8032 - val_loss: 0.4424
Epoch 25/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 131ms/step - accuracy: 0.7723 - loss: 0.4801 - val_accur
acy: 0.8055 - val_loss: 0.4398
Epoch 26/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.7803 - loss: 0.4728 - val_accur
acy: 0.8055 - val_loss: 0.4371
Epoch 27/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.7802 - loss: 0.4759 - val_accur
acy: 0.8009 - val_loss: 0.4343
Epoch 28/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.7793 - loss: 0.4686 - val_accur
acy: 0.8009 - val_loss: 0.4316
Epoch 29/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.7829 - loss: 0.4666 - val_accur
acy: 0.8009 - val_loss: 0.4289
Epoch 30/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.7867 - loss: 0.4633 - val_accur
acy: 0.8032 - val_loss: 0.4263
Epoch 31/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 118ms/step - accuracy: 0.7866 - loss: 0.4585 - val_accur
acy: 0.8032 - val_loss: 0.4237
Epoch 32/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.7903 - loss: 0.4548 - val_accur
acy: 0.8032 - val_loss: 0.4211
Epoch 33/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.7947 - loss: 0.4507 - val_accur
acy: 0.8055 - val_loss: 0.4186
Epoch 34/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.7915 - loss: 0.4517 - val_accur
acy: 0.8124 - val_loss: 0.4160
Epoch 35/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 117ms/step - accuracy: 0.7977 - loss: 0.4458 - val_accur
acy: 0.8124 - val_loss: 0.4136
Epoch 36/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.8068 - loss: 0.4471 - val_accur
acy: 0.8124 - val_loss: 0.4112
Epoch 37/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.7996 - loss: 0.4420 - val_accur
acy: 0.8146 - val_loss: 0.4087
Epoch 38/300
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 117ms/step - accuracy: 0.8068 - loss: 0.4390 - val_accur
acy: 0.8146 - val_loss: 0.4063
Epoch 39/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.8091 - loss: 0.4392 - val_accur
acy: 0.8169 - val_loss: 0.4039
Epoch 40/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 118ms/step - accuracy: 0.8063 - loss: 0.4352 - val_accur
acy: 0.8192 - val_loss: 0.4016
Epoch 41/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 119ms/step - accuracy: 0.8115 - loss: 0.4297 - val_accur
acy: 0.8192 - val_loss: 0.3994
Epoch 42/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - accuracy: 0.8039 - loss: 0.4301 - val_accur
acy: 0.8192 - val_loss: 0.3971
Epoch 43/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 106ms/step - accuracy: 0.8121 - loss: 0.4306 - val_accur
acy: 0.8215 - val_loss: 0.3949
Epoch 44/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8105 - loss: 0.4242 - val_accur
acy: 0.8238 - val_loss: 0.3927
Epoch 45/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.8217 - loss: 0.4162 - val_accur
acy: 0.8284 - val_loss: 0.3904
Epoch 46/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 106ms/step - accuracy: 0.8175 - loss: 0.4206 - val_accur
acy: 0.8307 - val_loss: 0.3881
Epoch 47/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8172 - loss: 0.4145 - val_accur
acy: 0.8307 - val_loss: 0.3858
Epoch 48/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8314 - loss: 0.4079 - val_accur
acy: 0.8307 - val_loss: 0.3833
Epoch 49/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 115ms/step - accuracy: 0.8231 - loss: 0.4134 - val_accur
acy: 0.8330 - val_loss: 0.3810
Epoch 50/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8260 - loss: 0.4043 - val_accur
acy: 0.8330 - val_loss: 0.3786
Epoch 51/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8215 - loss: 0.4108 - val_accur
acy: 0.8375 - val_loss: 0.3763
Epoch 52/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8278 - loss: 0.4018 - val_accur
acy: 0.8398 - val_loss: 0.3743
Epoch 53/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8260 - loss: 0.4011 - val_accur
acy: 0.8352 - val_loss: 0.3726
Epoch 54/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8378 - loss: 0.3982 - val_accur
acy: 0.8398 - val_loss: 0.3710
Epoch 55/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 119ms/step - accuracy: 0.8387 - loss: 0.3980 - val_accur
acy: 0.8398 - val_loss: 0.3696
Epoch 56/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 118ms/step - accuracy: 0.8358 - loss: 0.3936 - val_accur
acy: 0.8421 - val_loss: 0.3682
```

```
Epoch 57/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8370 - loss: 0.3944 - val_accur
acy: 0.8444 - val_loss: 0.3669
Epoch 58/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8364 - loss: 0.3847 - val_accur
acy: 0.8467 - val_loss: 0.3655
Epoch 59/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8330 - loss: 0.3907 - val_accur
acy: 0.8467 - val_loss: 0.3640
Epoch 60/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8480 - loss: 0.3822 - val_accur
acy: 0.8513 - val_loss: 0.3624
Epoch 61/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 105ms/step - accuracy: 0.8451 - loss: 0.3859 - val_accur
acy: 0.8535 - val_loss: 0.3609
Epoch 62/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.8386 - loss: 0.3830 - val_accur
acy: 0.8535 - val_loss: 0.3594
Epoch 63/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.8375 - loss: 0.3823 - val_accur
acy: 0.8513 - val_loss: 0.3579
Epoch 64/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8473 - loss: 0.3783 - val_accur
acy: 0.8513 - val_loss: 0.3565
Epoch 65/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 126ms/step - accuracy: 0.8443 - loss: 0.3800 - val_accur
acy: 0.8513 - val_loss: 0.3551
Epoch 66/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 134ms/step - accuracy: 0.8433 - loss: 0.3750 - val_accur
acy: 0.8513 - val_loss: 0.3538
Epoch 67/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 122ms/step - accuracy: 0.8388 - loss: 0.3728 - val_accur
acy: 0.8513 - val_loss: 0.3525
Epoch 68/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.8553 - loss: 0.3716 - val_accur
acy: 0.8535 - val_loss: 0.3515
Epoch 69/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 123ms/step - accuracy: 0.8489 - loss: 0.3691 - val_accur
acy: 0.8513 - val_loss: 0.3506
Epoch 70/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 133ms/step - accuracy: 0.8459 - loss: 0.3727 - val_accur
acy: 0.8490 - val_loss: 0.3498
Epoch 71/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 117ms/step - accuracy: 0.8479 - loss: 0.3652 - val_accur
acy: 0.8490 - val_loss: 0.3491
Epoch 72/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8536 - loss: 0.3627 - val_accur
acy: 0.8513 - val_loss: 0.3485
Epoch 73/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8566 - loss: 0.3654 - val_accur
acy: 0.8513 - val_loss: 0.3479
Epoch 74/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 127ms/step - accuracy: 0.8534 - loss: 0.3650 - val_accur
acy: 0.8513 - val_loss: 0.3471
Epoch 75/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 124ms/step - accuracy: 0.8485 - loss: 0.3648 - val_accur
```

```
acy: 0.8513 - val_loss: 0.3465
Epoch 76/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.8532 - loss: 0.3594 - val_accur
acy: 0.8535 - val_loss: 0.3460
Epoch 77/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 118ms/step - accuracy: 0.8546 - loss: 0.3567 - val_accur
acy: 0.8558 - val_loss: 0.3453
Epoch 78/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8552 - loss: 0.3572 - val_accur
acy: 0.8558 - val_loss: 0.3446
Epoch 79/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.8478 - loss: 0.3561 - val_accur
acy: 0.8558 - val_loss: 0.3439
Epoch 80/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 117ms/step - accuracy: 0.8528 - loss: 0.3610 - val_accur
acy: 0.8581 - val_loss: 0.3431
Epoch 81/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8581 - loss: 0.3522 - val_accur
acy: 0.8581 - val_loss: 0.3424
Epoch 82/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 119ms/step - accuracy: 0.8556 - loss: 0.3522 - val_accur
acy: 0.8581 - val_loss: 0.3418
Epoch 83/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 103ms/step - accuracy: 0.8564 - loss: 0.3533 - val_accur
acy: 0.8581 - val_loss: 0.3413
Epoch 84/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 115ms/step - accuracy: 0.8592 - loss: 0.3481 - val_accur
acy: 0.8581 - val_loss: 0.3409
Epoch 85/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - accuracy: 0.8557 - loss: 0.3450 - val_accur
acy: 0.8581 - val_loss: 0.3405
Epoch 86/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 118ms/step - accuracy: 0.8607 - loss: 0.3474 - val_accur
acy: 0.8581 - val_loss: 0.3402
Epoch 87/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 106ms/step - accuracy: 0.8573 - loss: 0.3456 - val_accur
acy: 0.8627 - val_loss: 0.3400
Epoch 88/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8579 - loss: 0.3503 - val_accur
acy: 0.8650 - val_loss: 0.3397
Epoch 89/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 115ms/step - accuracy: 0.8618 - loss: 0.3457 - val_accur
acy: 0.8650 - val_loss: 0.3394
Epoch 90/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8527 - loss: 0.3477 - val_accur
acy: 0.8650 - val_loss: 0.3392
Epoch 91/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8544 - loss: 0.3504 - val_accur
acy: 0.8650 - val_loss: 0.3389
Epoch 92/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 103ms/step - accuracy: 0.8584 - loss: 0.3453 - val_accur
acy: 0.8650 - val_loss: 0.3386
Epoch 93/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 107ms/step - accuracy: 0.8621 - loss: 0.3461 - val_accur
acy: 0.8673 - val_loss: 0.3384
Epoch 94/300
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.8588 - loss: 0.3394 - val_accur
acy: 0.8673 - val_loss: 0.3381
Epoch 95/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 120ms/step - accuracy: 0.8588 - loss: 0.3428 - val_accur
acy: 0.8673 - val_loss: 0.3377
Epoch 96/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 123ms/step - accuracy: 0.8601 - loss: 0.3446 - val_accur
acy: 0.8673 - val_loss: 0.3373
Epoch 97/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.8663 - loss: 0.3355 - val_accur
acy: 0.8673 - val_loss: 0.3370
Epoch 98/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8527 - loss: 0.3411 - val_accur
acy: 0.8673 - val_loss: 0.3367
Epoch 99/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.8602 - loss: 0.3415 - val_accur
acy: 0.8673 - val_loss: 0.3365
Epoch 100/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 118ms/step - accuracy: 0.8666 - loss: 0.3363 - val_accur
acy: 0.8696 - val_loss: 0.3363
Epoch 101/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.8643 - loss: 0.3394 - val_accur
acy: 0.8719 - val_loss: 0.3362
Epoch 102/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 106ms/step - accuracy: 0.8643 - loss: 0.3354 - val_accur
acy: 0.8719 - val_loss: 0.3361
Epoch 103/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8557 - loss: 0.3391 - val_accur
acy: 0.8719 - val_loss: 0.3362
Epoch 104/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8610 - loss: 0.3425 - val_accur
acy: 0.8719 - val_loss: 0.3363
Epoch 105/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.8580 - loss: 0.3436 - val_accur
acy: 0.8719 - val_loss: 0.3363
Epoch 106/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 105ms/step - accuracy: 0.8711 - loss: 0.3365 - val_accur
acy: 0.8719 - val_loss: 0.3359
Epoch 107/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8588 - loss: 0.3324 - val_accur
acy: 0.8719 - val_loss: 0.3354
Epoch 108/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 120ms/step - accuracy: 0.8624 - loss: 0.3305 - val_accur
acy: 0.8719 - val_loss: 0.3349
Epoch 109/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 104ms/step - accuracy: 0.8669 - loss: 0.3344 - val_accur
acy: 0.8719 - val_loss: 0.3345
Epoch 110/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.8617 - loss: 0.3309 - val_accur
acy: 0.8696 - val_loss: 0.3344
Epoch 111/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 120ms/step - accuracy: 0.8628 - loss: 0.3338 - val_accur
acy: 0.8696 - val_loss: 0.3345
Epoch 112/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 117ms/step - accuracy: 0.8655 - loss: 0.3305 - val_accur
acy: 0.8696 - val_loss: 0.3348
```

```
Epoch 113/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8643 - loss: 0.3316 - val_accur
acy: 0.8696 - val_loss: 0.3349
Epoch 114/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8657 - loss: 0.3310 - val_accur
acy: 0.8696 - val_loss: 0.3351
Epoch 115/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 125ms/step - accuracy: 0.8607 - loss: 0.3303 - val_accur
acy: 0.8696 - val_loss: 0.3353
Epoch 116/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 104ms/step - accuracy: 0.8655 - loss: 0.3326 - val_accur
acy: 0.8696 - val_loss: 0.3353
Epoch 117/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 115ms/step - accuracy: 0.8651 - loss: 0.3337 - val_accur
acy: 0.8673 - val_loss: 0.3352
Epoch 118/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 103ms/step - accuracy: 0.8631 - loss: 0.3311 - val_accur
acy: 0.8673 - val_loss: 0.3351
Epoch 119/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8676 - loss: 0.3291 - val_accur
acy: 0.8673 - val_loss: 0.3350
Epoch 120/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8648 - loss: 0.3262 - val_accur
acy: 0.8673 - val_loss: 0.3346
Epoch 121/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8658 - loss: 0.3318 - val_accur
acy: 0.8673 - val_loss: 0.3342
Epoch 122/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.8647 - loss: 0.3317 - val_accur
acy: 0.8696 - val_loss: 0.3338
Epoch 123/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8709 - loss: 0.3207 - val_accur
acy: 0.8696 - val_loss: 0.3337
Epoch 124/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8687 - loss: 0.3315 - val_accur
acy: 0.8696 - val_loss: 0.3338
Epoch 125/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 125ms/step - accuracy: 0.8691 - loss: 0.3234 - val_accur
acy: 0.8696 - val_loss: 0.3341
Epoch 126/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 125ms/step - accuracy: 0.8695 - loss: 0.3287 - val_accur
acy: 0.8696 - val_loss: 0.3345
Epoch 127/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 119ms/step - accuracy: 0.8679 - loss: 0.3301 - val_accur
acy: 0.8696 - val_loss: 0.3349
Epoch 128/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 125ms/step - accuracy: 0.8694 - loss: 0.3242 - val_accur
acy: 0.8696 - val_loss: 0.3350
Epoch 129/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 129ms/step - accuracy: 0.8706 - loss: 0.3239 - val_accur
acy: 0.8696 - val_loss: 0.3350
Epoch 130/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 119ms/step - accuracy: 0.8712 - loss: 0.3280 - val_accur
acy: 0.8696 - val_loss: 0.3351
Epoch 131/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 120ms/step - accuracy: 0.8584 - loss: 0.3283 - val_accur
```

```
acy: 0.8696 - val_loss: 0.3351
Epoch 132/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 129ms/step - accuracy: 0.8680 - loss: 0.3290 - val_accur
acy: 0.8696 - val_loss: 0.3352
Epoch 133/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 124ms/step - accuracy: 0.8743 - loss: 0.3291 - val_accur
acy: 0.8696 - val_loss: 0.3350
Epoch 134/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - accuracy: 0.8734 - loss: 0.3214 - val_accur
acy: 0.8696 - val_loss: 0.3347
Epoch 135/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 125ms/step - accuracy: 0.8726 - loss: 0.3248 - val_accur
acy: 0.8696 - val_loss: 0.3343
Epoch 136/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 123ms/step - accuracy: 0.8730 - loss: 0.3166 - val_accur
acy: 0.8719 - val_loss: 0.3340
Epoch 137/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 120ms/step - accuracy: 0.8718 - loss: 0.3172 - val_accur
acy: 0.8719 - val_loss: 0.3341
Epoch 138/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 120ms/step - accuracy: 0.8715 - loss: 0.3180 - val_accur
acy: 0.8696 - val_loss: 0.3342
Epoch 139/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 119ms/step - accuracy: 0.8717 - loss: 0.3201 - val_accur
acy: 0.8719 - val_loss: 0.3342
Epoch 140/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8678 - loss: 0.3291 - val_accur
acy: 0.8719 - val_loss: 0.3341
Epoch 141/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8690 - loss: 0.3259 - val_accur
acy: 0.8719 - val_loss: 0.3339
Epoch 142/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8679 - loss: 0.3234 - val_accur
acy: 0.8719 - val_loss: 0.3338
Epoch 143/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8708 - loss: 0.3254 - val_accur
acy: 0.8719 - val_loss: 0.3340
Epoch 144/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8675 - loss: 0.3196 - val_accur
acy: 0.8719 - val_loss: 0.3341
Epoch 145/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.8779 - loss: 0.3092 - val_accur
acy: 0.8719 - val_loss: 0.3342
Epoch 146/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 123ms/step - accuracy: 0.8744 - loss: 0.3172 - val_accur
acy: 0.8719 - val_loss: 0.3344
Epoch 147/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8687 - loss: 0.3180 - val_accur
acy: 0.8719 - val_loss: 0.3341
Epoch 148/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.8677 - loss: 0.3209 - val_accur
acy: 0.8719 - val_loss: 0.3338
Epoch 149/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 123ms/step - accuracy: 0.8827 - loss: 0.3120 - val_accur
acy: 0.8719 - val_loss: 0.3334
Epoch 150/300
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 120ms/step - accuracy: 0.8679 - loss: 0.3245 - val_accur
acy: 0.8719 - val_loss: 0.3330
Epoch 151/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 117ms/step - accuracy: 0.8712 - loss: 0.3164 - val_accur
acy: 0.8719 - val_loss: 0.3328
Epoch 152/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 117ms/step - accuracy: 0.8753 - loss: 0.3120 - val_accur
acy: 0.8719 - val_loss: 0.3327
Epoch 153/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8666 - loss: 0.3190 - val_accur
acy: 0.8696 - val_loss: 0.3329
Epoch 154/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.8670 - loss: 0.3123 - val_accur
acy: 0.8719 - val_loss: 0.3331
Epoch 155/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 122ms/step - accuracy: 0.8747 - loss: 0.3101 - val_accur
acy: 0.8719 - val_loss: 0.3334
Epoch 156/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8669 - loss: 0.3172 - val_accur
acy: 0.8719 - val_loss: 0.3339
Epoch 157/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 118ms/step - accuracy: 0.8726 - loss: 0.3141 - val_accur
acy: 0.8719 - val_loss: 0.3342
Epoch 158/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 123ms/step - accuracy: 0.8746 - loss: 0.3152 - val_accur
acy: 0.8719 - val_loss: 0.3344
Epoch 159/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8683 - loss: 0.3174 - val_accur
acy: 0.8719 - val_loss: 0.3345
Epoch 160/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 123ms/step - accuracy: 0.8652 - loss: 0.3192 - val_accur
acy: 0.8719 - val_loss: 0.3345
Epoch 161/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.8688 - loss: 0.3142 - val_accur
acy: 0.8719 - val_loss: 0.3345
Epoch 162/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8680 - loss: 0.3231 - val_accur
acy: 0.8719 - val_loss: 0.3343
Epoch 163/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.8745 - loss: 0.3104 - val_accur
acy: 0.8719 - val_loss: 0.3341
Epoch 164/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8716 - loss: 0.3165 - val_accur
acy: 0.8787 - val_loss: 0.3337
Epoch 165/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8689 - loss: 0.3170 - val_accur
acy: 0.8764 - val_loss: 0.3336
Epoch 166/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8752 - loss: 0.3094 - val_accur
acy: 0.8764 - val_loss: 0.3335
Epoch 167/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - accuracy: 0.8661 - loss: 0.3219 - val_accur
acy: 0.8764 - val_loss: 0.3336
Epoch 168/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 120ms/step - accuracy: 0.8672 - loss: 0.3138 - val_accur
acy: 0.8787 - val_loss: 0.3340
```

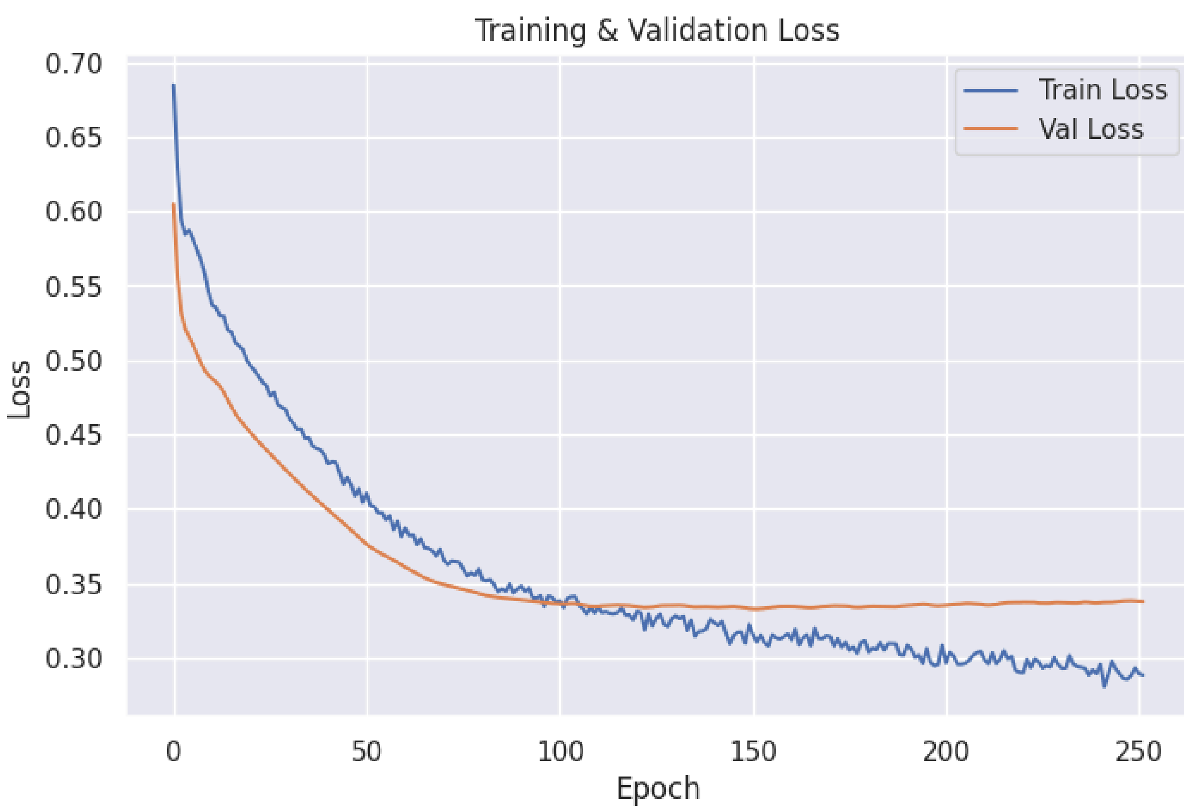
```
Epoch 169/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8733 - loss: 0.3129 - val_accur
acy: 0.8787 - val_loss: 0.3343
Epoch 170/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8718 - loss: 0.3177 - val_accur
acy: 0.8764 - val_loss: 0.3345
Epoch 171/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8720 - loss: 0.3161 - val_accur
acy: 0.8764 - val_loss: 0.3347
Epoch 172/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8667 - loss: 0.3100 - val_accur
acy: 0.8764 - val_loss: 0.3349
Epoch 173/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8667 - loss: 0.3145 - val_accur
acy: 0.8764 - val_loss: 0.3348
Epoch 174/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - accuracy: 0.8746 - loss: 0.3098 - val_accur
acy: 0.8787 - val_loss: 0.3347
Epoch 175/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8662 - loss: 0.3140 - val_accur
acy: 0.8719 - val_loss: 0.3347
Epoch 176/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8741 - loss: 0.3074 - val_accur
acy: 0.8719 - val_loss: 0.3345
Epoch 177/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 107ms/step - accuracy: 0.8731 - loss: 0.3115 - val_accur
acy: 0.8764 - val_loss: 0.3340
Epoch 178/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 120ms/step - accuracy: 0.8845 - loss: 0.3037 - val_accur
acy: 0.8764 - val_loss: 0.3338
Epoch 179/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 119ms/step - accuracy: 0.8721 - loss: 0.3125 - val_accur
acy: 0.8764 - val_loss: 0.3339
Epoch 180/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 122ms/step - accuracy: 0.8783 - loss: 0.3135 - val_accur
acy: 0.8764 - val_loss: 0.3342
Epoch 181/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 117ms/step - accuracy: 0.8744 - loss: 0.3071 - val_accur
acy: 0.8741 - val_loss: 0.3344
Epoch 182/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8675 - loss: 0.3097 - val_accur
acy: 0.8764 - val_loss: 0.3345
Epoch 183/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8738 - loss: 0.3081 - val_accur
acy: 0.8741 - val_loss: 0.3345
Epoch 184/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 119ms/step - accuracy: 0.8714 - loss: 0.3108 - val_accur
acy: 0.8741 - val_loss: 0.3344
Epoch 185/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8708 - loss: 0.3067 - val_accur
acy: 0.8741 - val_loss: 0.3344
Epoch 186/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 115ms/step - accuracy: 0.8779 - loss: 0.3122 - val_accur
acy: 0.8741 - val_loss: 0.3343
Epoch 187/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 130ms/step - accuracy: 0.8700 - loss: 0.3119 - val_accur
```

```
acy: 0.8741 - val_loss: 0.3341
Epoch 188/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - accuracy: 0.8705 - loss: 0.3100 - val_accur
acy: 0.8741 - val_loss: 0.3342
Epoch 189/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 125ms/step - accuracy: 0.8794 - loss: 0.3048 - val_accur
acy: 0.8741 - val_loss: 0.3346
Epoch 190/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 115ms/step - accuracy: 0.8760 - loss: 0.3028 - val_accur
acy: 0.8741 - val_loss: 0.3348
Epoch 191/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 123ms/step - accuracy: 0.8774 - loss: 0.3111 - val_accur
acy: 0.8741 - val_loss: 0.3350
Epoch 192/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8737 - loss: 0.3084 - val_accur
acy: 0.8764 - val_loss: 0.3353
Epoch 193/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - accuracy: 0.8785 - loss: 0.3030 - val_accur
acy: 0.8764 - val_loss: 0.3355
Epoch 194/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 123ms/step - accuracy: 0.8738 - loss: 0.3035 - val_accur
acy: 0.8741 - val_loss: 0.3356
Epoch 195/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 130ms/step - accuracy: 0.8793 - loss: 0.2976 - val_accur
acy: 0.8741 - val_loss: 0.3359
Epoch 196/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 126ms/step - accuracy: 0.8700 - loss: 0.3075 - val_accur
acy: 0.8741 - val_loss: 0.3358
Epoch 197/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 117ms/step - accuracy: 0.8734 - loss: 0.2980 - val_accur
acy: 0.8741 - val_loss: 0.3354
Epoch 198/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 143ms/step - accuracy: 0.8887 - loss: 0.2968 - val_accur
acy: 0.8741 - val_loss: 0.3350
Epoch 199/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 119ms/step - accuracy: 0.8752 - loss: 0.2976 - val_accur
acy: 0.8719 - val_loss: 0.3348
Epoch 200/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 119ms/step - accuracy: 0.8706 - loss: 0.3103 - val_accur
acy: 0.8741 - val_loss: 0.3350
Epoch 201/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 119ms/step - accuracy: 0.8792 - loss: 0.3014 - val_accur
acy: 0.8719 - val_loss: 0.3352
Epoch 202/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 119ms/step - accuracy: 0.8753 - loss: 0.3065 - val_accur
acy: 0.8719 - val_loss: 0.3355
Epoch 203/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8769 - loss: 0.3038 - val_accur
acy: 0.8719 - val_loss: 0.3358
Epoch 204/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8825 - loss: 0.2968 - val_accur
acy: 0.8741 - val_loss: 0.3359
Epoch 205/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - accuracy: 0.8863 - loss: 0.2989 - val_accur
acy: 0.8741 - val_loss: 0.3361
Epoch 206/300
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - accuracy: 0.8792 - loss: 0.2993 - val_accur
acy: 0.8741 - val_loss: 0.3364
Epoch 207/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - accuracy: 0.8748 - loss: 0.3034 - val_accur
acy: 0.8741 - val_loss: 0.3364
Epoch 208/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - accuracy: 0.8765 - loss: 0.3030 - val_accur
acy: 0.8741 - val_loss: 0.3362
Epoch 209/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8758 - loss: 0.3048 - val_accur
acy: 0.8741 - val_loss: 0.3360
Epoch 210/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8820 - loss: 0.3062 - val_accur
acy: 0.8741 - val_loss: 0.3358
Epoch 211/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8769 - loss: 0.3013 - val_accur
acy: 0.8741 - val_loss: 0.3354
Epoch 212/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8755 - loss: 0.2984 - val_accur
acy: 0.8741 - val_loss: 0.3353
Epoch 213/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8752 - loss: 0.3062 - val_accur
acy: 0.8741 - val_loss: 0.3354
Epoch 214/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 119ms/step - accuracy: 0.8708 - loss: 0.2982 - val_accur
acy: 0.8741 - val_loss: 0.3357
Epoch 215/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 117ms/step - accuracy: 0.8780 - loss: 0.3015 - val_accur
acy: 0.8741 - val_loss: 0.3363
Epoch 216/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.8737 - loss: 0.3096 - val_accur
acy: 0.8741 - val_loss: 0.3367
Epoch 217/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.8805 - loss: 0.3019 - val_accur
acy: 0.8741 - val_loss: 0.3371
Epoch 218/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 118ms/step - accuracy: 0.8791 - loss: 0.3040 - val_accur
acy: 0.8741 - val_loss: 0.3372
Epoch 219/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 106ms/step - accuracy: 0.8835 - loss: 0.2941 - val_accur
acy: 0.8764 - val_loss: 0.3372
Epoch 220/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.8784 - loss: 0.2916 - val_accur
acy: 0.8764 - val_loss: 0.3372
Epoch 221/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 124ms/step - accuracy: 0.8807 - loss: 0.2930 - val_accur
acy: 0.8764 - val_loss: 0.3373
Epoch 222/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8785 - loss: 0.3022 - val_accur
acy: 0.8741 - val_loss: 0.3373
Epoch 223/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8784 - loss: 0.2948 - val_accur
acy: 0.8741 - val_loss: 0.3373
Epoch 224/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 123ms/step - accuracy: 0.8794 - loss: 0.3023 - val_accur
acy: 0.8741 - val_loss: 0.3372
```

```
Epoch 225/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8773 - loss: 0.3010 - val_accur
acy: 0.8741 - val_loss: 0.3367
Epoch 226/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8784 - loss: 0.2987 - val_accur
acy: 0.8741 - val_loss: 0.3366
Epoch 227/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 105ms/step - accuracy: 0.8766 - loss: 0.2969 - val_accur
acy: 0.8764 - val_loss: 0.3366
Epoch 228/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 123ms/step - accuracy: 0.8818 - loss: 0.2958 - val_accur
acy: 0.8764 - val_loss: 0.3367
Epoch 229/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8758 - loss: 0.3063 - val_accur
acy: 0.8741 - val_loss: 0.3370
Epoch 230/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8758 - loss: 0.3002 - val_accur
acy: 0.8741 - val_loss: 0.3371
Epoch 231/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8825 - loss: 0.2951 - val_accur
acy: 0.8741 - val_loss: 0.3370
Epoch 232/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 122ms/step - accuracy: 0.8752 - loss: 0.2971 - val_accur
acy: 0.8741 - val_loss: 0.3370
Epoch 233/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8810 - loss: 0.3056 - val_accur
acy: 0.8741 - val_loss: 0.3368
Epoch 234/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8851 - loss: 0.3010 - val_accur
acy: 0.8741 - val_loss: 0.3367
Epoch 235/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8772 - loss: 0.2975 - val_accur
acy: 0.8764 - val_loss: 0.3367
Epoch 236/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 122ms/step - accuracy: 0.8797 - loss: 0.2971 - val_accur
acy: 0.8741 - val_loss: 0.3371
Epoch 237/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 122ms/step - accuracy: 0.8814 - loss: 0.2969 - val_accur
acy: 0.8741 - val_loss: 0.3374
Epoch 238/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 105ms/step - accuracy: 0.8834 - loss: 0.2911 - val_accur
acy: 0.8741 - val_loss: 0.3373
Epoch 239/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 119ms/step - accuracy: 0.8803 - loss: 0.2947 - val_accur
acy: 0.8741 - val_loss: 0.3369
Epoch 240/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 122ms/step - accuracy: 0.8804 - loss: 0.2940 - val_accur
acy: 0.8741 - val_loss: 0.3368
Epoch 241/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8820 - loss: 0.2979 - val_accur
acy: 0.8741 - val_loss: 0.3370
Epoch 242/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8860 - loss: 0.2821 - val_accur
acy: 0.8741 - val_loss: 0.3372
Epoch 243/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8790 - loss: 0.2940 - val_accur
```

```
acy: 0.8741 - val_loss: 0.3372
Epoch 244/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 120ms/step - accuracy: 0.8803 - loss: 0.3004 - val_accur
acy: 0.8741 - val_loss: 0.3373
Epoch 245/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 122ms/step - accuracy: 0.8757 - loss: 0.2933 - val_accur
acy: 0.8741 - val_loss: 0.3376
Epoch 246/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 120ms/step - accuracy: 0.8813 - loss: 0.2931 - val_accur
acy: 0.8741 - val_loss: 0.3380
Epoch 247/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 104ms/step - accuracy: 0.8805 - loss: 0.2890 - val_accur
acy: 0.8741 - val_loss: 0.3382
Epoch 248/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 134ms/step - accuracy: 0.8778 - loss: 0.2910 - val_accur
acy: 0.8741 - val_loss: 0.3383
Epoch 249/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 123ms/step - accuracy: 0.8771 - loss: 0.2893 - val_accur
acy: 0.8741 - val_loss: 0.3383
Epoch 250/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 130ms/step - accuracy: 0.8867 - loss: 0.2980 - val_accur
acy: 0.8741 - val_loss: 0.3382
Epoch 251/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 131ms/step - accuracy: 0.8833 - loss: 0.2933 - val_accur
acy: 0.8741 - val_loss: 0.3380
Epoch 252/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 124ms/step - accuracy: 0.8847 - loss: 0.2908 - val_accur
acy: 0.8741 - val_loss: 0.3378
Epoch 252: early stopping
Restoring model weights from the end of the best epoch: 152.
```



In [291…

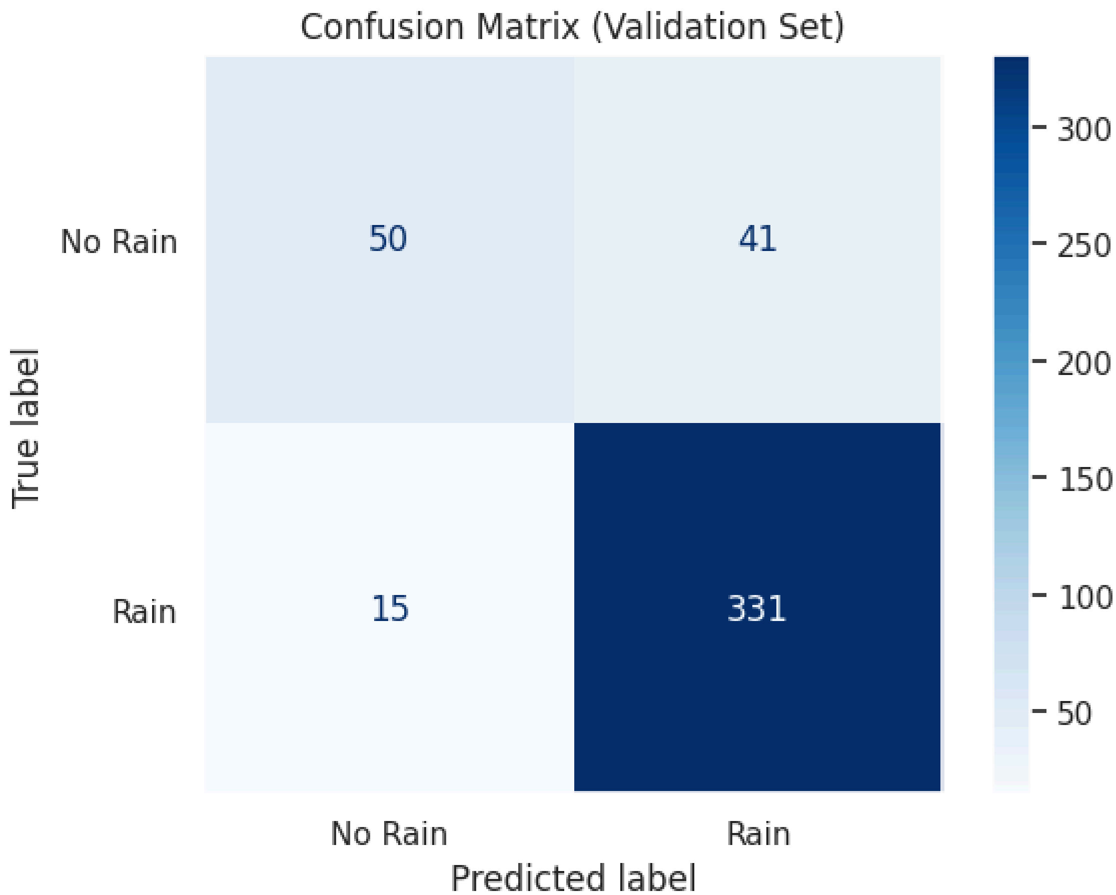
```
# Evaluate
train_loss, train_acc = model_2d_cnn.evaluate(X_train_seq, y_train_seq)
val_loss, val_acc = model_2d_cnn.evaluate(X_val_seq, y_val_seq)

print(f"Training Accuracy: {train_acc:.4f}")
print(f"Validation Accuracy: {val_acc:.4f}")
```

```
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8754 - loss: 0.3133
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8715 - loss: 0.3419
Training Accuracy: 0.8792
Validation Accuracy: 0.8719
```

In [292…

```
# Predict probabilities
y_val_probs = model_2d_cnn.predict(X_val_seq)

# Convert probabilities to binary predictions
y_val_preds = (y_val_probs > 0.5).astype("int32")

# Generate confusion matrix
cm = confusion_matrix(y_val_seq, y_val_preds)

# Display
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["No Rain", "Rain
disp.plot(cmap=plt.cm.Blues)
plt.title("Confusion Matrix (Validation Set)")
plt.grid(False)
plt.show()
```

```
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
```



In [293…

```
# Apply to test set
y_test_pred = model_2d_cnn.predict(X_test_cnn).flatten()

submission = pd.DataFrame({
    'id': submission_ids,
    'rainfall': y_test_pred
})

# Save to CSV
submission.to_csv(PATH + 'submission_2d_cnn.csv', index=False)
```

```
23/23 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
```

In [294…

```
# Two-Layer CNN model
tf.keras.backend.clear_session()
model_2d_cnn2 = tf.keras.Sequential()

# Add convolutional layer
model_2d_cnn2.add(tf.keras.layers.Conv2D(
    filters=128,
    kernel_size=(4, 8),
    strides=(1, 1),
    padding='same',
    data_format='channels_last',
    activation='relu',
    name='conv_1',
    input_shape=(window_size, X.shape[1], 1)  # (height, width, channels)
```

```
))

# Add max pooling layer
model_2d_cnn2.add(tf.keras.layers.MaxPooling2D(pool_size=(2, 2)))

# Add dropout layer
model_2d_cnn2.add(tf.keras.layers.Dropout(rate=0.6))

model_2d_cnn2.add(tf.keras.layers.Conv2D(
    filters=32,
    kernel_size=(2, 4),
    strides=(1, 1),
    padding='same',
    data_format='channels_last',
    activation='relu',
    name='conv_2')  # (height, width, channels)
)

# Add max pooling layer
model_2d_cnn2.add(tf.keras.layers.MaxPooling2D(pool_size=(2, 2)))

# Add dropout layer
model_2d_cnn2.add(tf.keras.layers.Dropout(rate=0.6))

# Add flattening layer
model_2d_cnn2.add(tf.keras.layers.Flatten())

# Add classification layer
model_2d_cnn2.add(tf.keras.layers.Dense(1, activation='sigmoid'))

# Compile model
model_2d_cnn2.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=0.002),
    loss=tf.keras.losses.BinaryCrossentropy(),
    metrics=['accuracy']
)

# Print summary
model_2d_cnn2.summary()
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base_conv.py:
107: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When
using Sequential models, prefer using an `Input(shape)` object as the first layer in
the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

Model: "sequential"

Layer (type)	Output Shape	
conv_1 (Conv2D)	(None, 8, 12, 128)	
max_pooling2d (MaxPooling2D)	(None, 4, 6, 128)	
dropout (Dropout)	(None, 4, 6, 128)	
conv_2 (Conv2D)	(None, 4, 6, 32)	
max_pooling2d_1 (MaxPooling2D)	(None, 2, 3, 32)	
dropout_1 (Dropout)	(None, 2, 3, 32)	
flatten (Flatten)	(None, 192)	
dense (Dense)	(None, 1)	

Total params: 37,217 (145.38 KB)

Trainable params: 37,217 (145.38 KB)

Non-trainable params: 0 (0.00 B)

In [295…

```
# Fit the model
history = model_2d_cnn2.fit(
    X_train_2d, y_train_seq,
    validation_data=(X_val_2d, y_val_seq),
    epochs=300,
    batch_size=1024,
    callbacks=[early_stopping]
)
# Plot losses
plt.figure(figsize=(8, 5))
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Val Loss')
plt.title("Training & Validation Loss")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.grid(True)
plt.legend()
plt.show()
```

```
Epoch 1/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 2s 374ms/step - accuracy: 0.6485 - loss: 0.6575 - val_accur
acy: 0.7918 - val_loss: 0.5231
Epoch 2/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 161ms/step - accuracy: 0.7483 - loss: 0.6020 - val_accur
acy: 0.7918 - val_loss: 0.5444
Epoch 3/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 165ms/step - accuracy: 0.7500 - loss: 0.5726 - val_accur
acy: 0.7918 - val_loss: 0.5914
Epoch 4/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 162ms/step - accuracy: 0.7469 - loss: 0.5745 - val_accur
acy: 0.7941 - val_loss: 0.5844
Epoch 5/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 164ms/step - accuracy: 0.7557 - loss: 0.5622 - val_accur
acy: 0.7918 - val_loss: 0.5438
Epoch 6/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 155ms/step - accuracy: 0.7466 - loss: 0.5452 - val_accur
acy: 0.7918 - val_loss: 0.5023
Epoch 7/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 166ms/step - accuracy: 0.7559 - loss: 0.5254 - val_accur
acy: 0.7918 - val_loss: 0.4822
Epoch 8/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 157ms/step - accuracy: 0.7585 - loss: 0.5255 - val_accur
acy: 0.8055 - val_loss: 0.4808
Epoch 9/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 160ms/step - accuracy: 0.7681 - loss: 0.5118 - val_accur
acy: 0.8055 - val_loss: 0.4914
Epoch 10/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 154ms/step - accuracy: 0.7624 - loss: 0.5093 - val_accur
acy: 0.8146 - val_loss: 0.5012
Epoch 11/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 166ms/step - accuracy: 0.7743 - loss: 0.5033 - val_accur
acy: 0.8124 - val_loss: 0.4996
Epoch 12/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 156ms/step - accuracy: 0.7712 - loss: 0.4935 - val_accur
acy: 0.7941 - val_loss: 0.4862
Epoch 13/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 161ms/step - accuracy: 0.7717 - loss: 0.4964 - val_accur
acy: 0.8055 - val_loss: 0.4708
Epoch 14/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 160ms/step - accuracy: 0.7728 - loss: 0.4875 - val_accur
acy: 0.8101 - val_loss: 0.4608
Epoch 15/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 156ms/step - accuracy: 0.7754 - loss: 0.4770 - val_accur
acy: 0.8192 - val_loss: 0.4552
Epoch 16/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 160ms/step - accuracy: 0.7889 - loss: 0.4722 - val_accur
acy: 0.8261 - val_loss: 0.4503
Epoch 17/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 162ms/step - accuracy: 0.7877 - loss: 0.4673 - val_accur
acy: 0.8307 - val_loss: 0.4391
Epoch 18/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 157ms/step - accuracy: 0.7982 - loss: 0.4503 - val_accur
acy: 0.8307 - val_loss: 0.4232
Epoch 19/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 155ms/step - accuracy: 0.8027 - loss: 0.4429 - val_accur
```

```
acy: 0.8330 - val_loss: 0.4106
Epoch 20/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 162ms/step - accuracy: 0.8040 - loss: 0.4439 - val_accur
acy: 0.8421 - val_loss: 0.4056
Epoch 21/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 155ms/step - accuracy: 0.8089 - loss: 0.4308 - val_accur
acy: 0.8535 - val_loss: 0.4075
Epoch 22/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 158ms/step - accuracy: 0.8269 - loss: 0.4237 - val_accur
acy: 0.8650 - val_loss: 0.4052
Epoch 23/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 158ms/step - accuracy: 0.8252 - loss: 0.4146 - val_accur
acy: 0.8673 - val_loss: 0.3981
Epoch 24/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 158ms/step - accuracy: 0.8307 - loss: 0.4076 - val_accur
acy: 0.8719 - val_loss: 0.3919
Epoch 25/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 157ms/step - accuracy: 0.8358 - loss: 0.4085 - val_accur
acy: 0.8741 - val_loss: 0.3830
Epoch 26/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 157ms/step - accuracy: 0.8231 - loss: 0.4067 - val_accur
acy: 0.8719 - val_loss: 0.3791
Epoch 27/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 162ms/step - accuracy: 0.8350 - loss: 0.3948 - val_accur
acy: 0.8787 - val_loss: 0.3791
Epoch 28/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 172ms/step - accuracy: 0.8468 - loss: 0.3845 - val_accur
acy: 0.8810 - val_loss: 0.3729
Epoch 29/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 163ms/step - accuracy: 0.8438 - loss: 0.3859 - val_accur
acy: 0.8810 - val_loss: 0.3652
Epoch 30/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 166ms/step - accuracy: 0.8490 - loss: 0.3811 - val_accur
acy: 0.8787 - val_loss: 0.3611
Epoch 31/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 180ms/step - accuracy: 0.8473 - loss: 0.3746 - val_accur
acy: 0.8787 - val_loss: 0.3610
Epoch 32/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 168ms/step - accuracy: 0.8473 - loss: 0.3741 - val_accur
acy: 0.8764 - val_loss: 0.3629
Epoch 33/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 174ms/step - accuracy: 0.8432 - loss: 0.3855 - val_accur
acy: 0.8696 - val_loss: 0.3685
Epoch 34/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 173ms/step - accuracy: 0.8476 - loss: 0.3671 - val_accur
acy: 0.8673 - val_loss: 0.3702
Epoch 35/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 175ms/step - accuracy: 0.8480 - loss: 0.3743 - val_accur
acy: 0.8650 - val_loss: 0.3723
Epoch 36/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 172ms/step - accuracy: 0.8628 - loss: 0.3674 - val_accur
acy: 0.8627 - val_loss: 0.3736
Epoch 37/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 162ms/step - accuracy: 0.8542 - loss: 0.3626 - val_accur
acy: 0.8627 - val_loss: 0.3698
Epoch 38/300
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 163ms/step - accuracy: 0.8635 - loss: 0.3551 - val_accur
acy: 0.8650 - val_loss: 0.3656
Epoch 39/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 155ms/step - accuracy: 0.8594 - loss: 0.3565 - val_accur
acy: 0.8719 - val_loss: 0.3652
Epoch 40/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 165ms/step - accuracy: 0.8538 - loss: 0.3695 - val_accur
acy: 0.8696 - val_loss: 0.3668
Epoch 41/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 163ms/step - accuracy: 0.8518 - loss: 0.3710 - val_accur
acy: 0.8719 - val_loss: 0.3688
Epoch 42/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 154ms/step - accuracy: 0.8540 - loss: 0.3668 - val_accur
acy: 0.8696 - val_loss: 0.3690
Epoch 43/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 160ms/step - accuracy: 0.8614 - loss: 0.3583 - val_accur
acy: 0.8673 - val_loss: 0.3661
Epoch 44/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 158ms/step - accuracy: 0.8523 - loss: 0.3688 - val_accur
acy: 0.8696 - val_loss: 0.3596
Epoch 45/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 164ms/step - accuracy: 0.8582 - loss: 0.3527 - val_accur
acy: 0.8673 - val_loss: 0.3577
Epoch 46/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 163ms/step - accuracy: 0.8607 - loss: 0.3526 - val_accur
acy: 0.8673 - val_loss: 0.3569
Epoch 47/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 158ms/step - accuracy: 0.8544 - loss: 0.3558 - val_accur
acy: 0.8696 - val_loss: 0.3544
Epoch 48/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 158ms/step - accuracy: 0.8585 - loss: 0.3529 - val_accur
acy: 0.8696 - val_loss: 0.3554
Epoch 49/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 157ms/step - accuracy: 0.8575 - loss: 0.3580 - val_accur
acy: 0.8696 - val_loss: 0.3594
Epoch 50/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 162ms/step - accuracy: 0.8681 - loss: 0.3431 - val_accur
acy: 0.8696 - val_loss: 0.3595
Epoch 51/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 164ms/step - accuracy: 0.8600 - loss: 0.3423 - val_accur
acy: 0.8673 - val_loss: 0.3565
Epoch 52/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 160ms/step - accuracy: 0.8560 - loss: 0.3539 - val_accur
acy: 0.8696 - val_loss: 0.3538
Epoch 53/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 157ms/step - accuracy: 0.8581 - loss: 0.3471 - val_accur
acy: 0.8719 - val_loss: 0.3576
Epoch 54/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 158ms/step - accuracy: 0.8597 - loss: 0.3412 - val_accur
acy: 0.8650 - val_loss: 0.3631
Epoch 55/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 161ms/step - accuracy: 0.8593 - loss: 0.3514 - val_accur
acy: 0.8696 - val_loss: 0.3613
Epoch 56/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 165ms/step - accuracy: 0.8671 - loss: 0.3387 - val_accur
acy: 0.8719 - val_loss: 0.3566
```

```
Epoch 57/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 158ms/step - accuracy: 0.8670 - loss: 0.3374 - val_accur
acy: 0.8696 - val_loss: 0.3501
Epoch 58/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 160ms/step - accuracy: 0.8717 - loss: 0.3341 - val_accur
acy: 0.8696 - val_loss: 0.3496
Epoch 59/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 157ms/step - accuracy: 0.8595 - loss: 0.3423 - val_accur
acy: 0.8650 - val_loss: 0.3532
Epoch 60/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 152ms/step - accuracy: 0.8619 - loss: 0.3459 - val_accur
acy: 0.8673 - val_loss: 0.3576
Epoch 61/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 157ms/step - accuracy: 0.8674 - loss: 0.3316 - val_accur
acy: 0.8673 - val_loss: 0.3575
Epoch 62/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 157ms/step - accuracy: 0.8552 - loss: 0.3498 - val_accur
acy: 0.8719 - val_loss: 0.3536
Epoch 63/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 157ms/step - accuracy: 0.8620 - loss: 0.3342 - val_accur
acy: 0.8741 - val_loss: 0.3505
Epoch 64/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 165ms/step - accuracy: 0.8620 - loss: 0.3445 - val_accur
acy: 0.8719 - val_loss: 0.3490
Epoch 65/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 159ms/step - accuracy: 0.8616 - loss: 0.3402 - val_accur
acy: 0.8741 - val_loss: 0.3489
Epoch 66/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 162ms/step - accuracy: 0.8658 - loss: 0.3381 - val_accur
acy: 0.8741 - val_loss: 0.3523
Epoch 67/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 156ms/step - accuracy: 0.8597 - loss: 0.3328 - val_accur
acy: 0.8696 - val_loss: 0.3572
Epoch 68/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 170ms/step - accuracy: 0.8664 - loss: 0.3338 - val_accur
acy: 0.8627 - val_loss: 0.3583
Epoch 69/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 163ms/step - accuracy: 0.8589 - loss: 0.3435 - val_accur
acy: 0.8627 - val_loss: 0.3549
Epoch 70/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 160ms/step - accuracy: 0.8729 - loss: 0.3272 - val_accur
acy: 0.8673 - val_loss: 0.3484
Epoch 71/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 171ms/step - accuracy: 0.8672 - loss: 0.3296 - val_accur
acy: 0.8741 - val_loss: 0.3449
Epoch 72/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 166ms/step - accuracy: 0.8689 - loss: 0.3247 - val_accur
acy: 0.8650 - val_loss: 0.3516
Epoch 73/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 166ms/step - accuracy: 0.8653 - loss: 0.3348 - val_accur
acy: 0.8604 - val_loss: 0.3600
Epoch 74/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 175ms/step - accuracy: 0.8724 - loss: 0.3323 - val_accur
acy: 0.8627 - val_loss: 0.3569
Epoch 75/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 169ms/step - accuracy: 0.8694 - loss: 0.3286 - val_accur
```

```
acy: 0.8673 - val_loss: 0.3523
Epoch 76/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 169ms/step - accuracy: 0.8566 - loss: 0.3364 - val_accur
acy: 0.8741 - val_loss: 0.3497
Epoch 77/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 174ms/step - accuracy: 0.8559 - loss: 0.3385 - val_accur
acy: 0.8719 - val_loss: 0.3538
Epoch 78/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 172ms/step - accuracy: 0.8689 - loss: 0.3237 - val_accur
acy: 0.8673 - val_loss: 0.3573
Epoch 79/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 169ms/step - accuracy: 0.8701 - loss: 0.3272 - val_accur
acy: 0.8719 - val_loss: 0.3544
Epoch 80/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 170ms/step - accuracy: 0.8706 - loss: 0.3260 - val_accur
acy: 0.8696 - val_loss: 0.3482
Epoch 81/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 173ms/step - accuracy: 0.8738 - loss: 0.3239 - val_accur
acy: 0.8696 - val_loss: 0.3495
Epoch 82/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 165ms/step - accuracy: 0.8680 - loss: 0.3224 - val_accur
acy: 0.8627 - val_loss: 0.3551
Epoch 83/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 164ms/step - accuracy: 0.8648 - loss: 0.3161 - val_accur
acy: 0.8650 - val_loss: 0.3579
Epoch 84/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 168ms/step - accuracy: 0.8708 - loss: 0.3208 - val_accur
acy: 0.8650 - val_loss: 0.3548
Epoch 85/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 159ms/step - accuracy: 0.8701 - loss: 0.3270 - val_accur
acy: 0.8696 - val_loss: 0.3520
Epoch 86/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 157ms/step - accuracy: 0.8677 - loss: 0.3216 - val_accur
acy: 0.8673 - val_loss: 0.3530
Epoch 87/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 166ms/step - accuracy: 0.8676 - loss: 0.3250 - val_accur
acy: 0.8627 - val_loss: 0.3543
Epoch 88/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 166ms/step - accuracy: 0.8702 - loss: 0.3153 - val_accur
acy: 0.8627 - val_loss: 0.3525
Epoch 89/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 163ms/step - accuracy: 0.8672 - loss: 0.3240 - val_accur
acy: 0.8650 - val_loss: 0.3523
Epoch 90/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 158ms/step - accuracy: 0.8734 - loss: 0.3213 - val_accur
acy: 0.8650 - val_loss: 0.3565
Epoch 91/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 159ms/step - accuracy: 0.8741 - loss: 0.3088 - val_accur
acy: 0.8627 - val_loss: 0.3599
Epoch 92/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 158ms/step - accuracy: 0.8639 - loss: 0.3217 - val_accur
acy: 0.8604 - val_loss: 0.3576
Epoch 93/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 158ms/step - accuracy: 0.8710 - loss: 0.3186 - val_accur
acy: 0.8627 - val_loss: 0.3560
Epoch 94/300
```

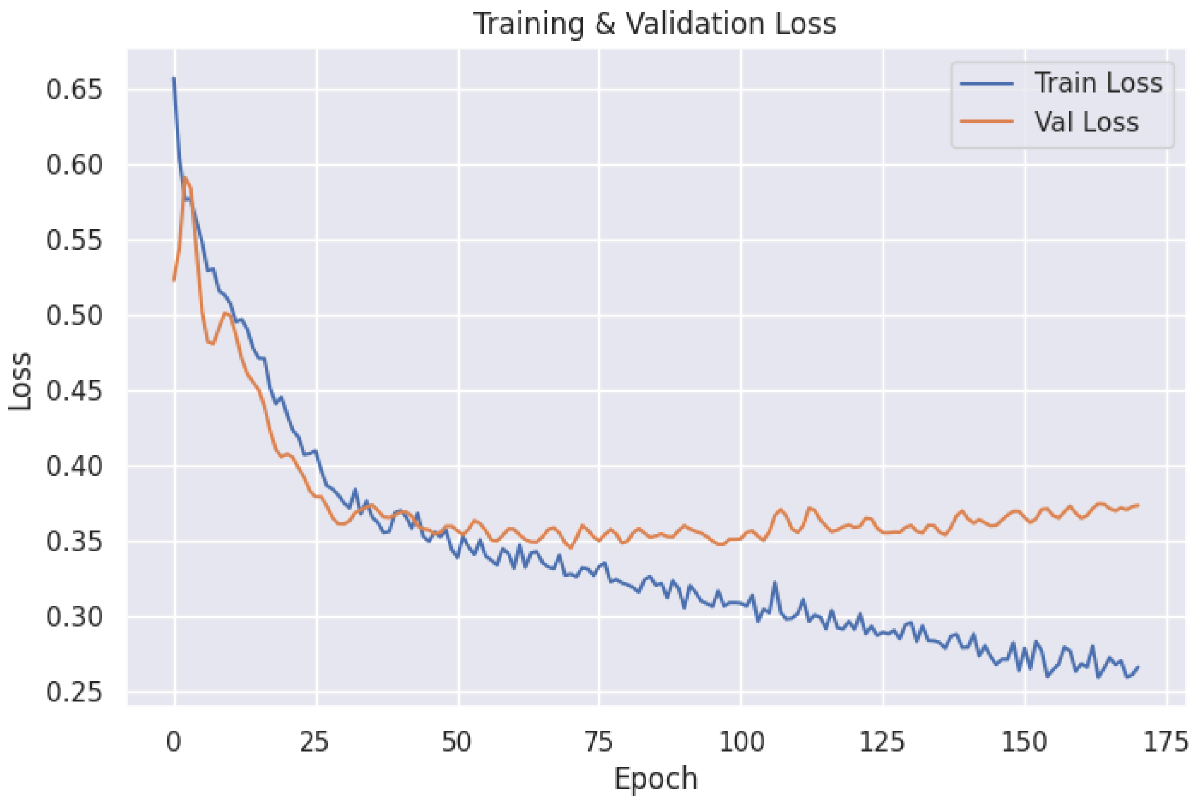
```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 159ms/step - accuracy: 0.8748 - loss: 0.3121 - val_accur
acy: 0.8650 - val_loss: 0.3549
Epoch 95/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 156ms/step - accuracy: 0.8762 - loss: 0.3094 - val_accur
acy: 0.8673 - val_loss: 0.3525
Epoch 96/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 159ms/step - accuracy: 0.8784 - loss: 0.3111 - val_accur
acy: 0.8696 - val_loss: 0.3496
Epoch 97/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 157ms/step - accuracy: 0.8721 - loss: 0.3216 - val_accur
acy: 0.8719 - val_loss: 0.3475
Epoch 98/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 165ms/step - accuracy: 0.8731 - loss: 0.3100 - val_accur
acy: 0.8719 - val_loss: 0.3476
Epoch 99/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 158ms/step - accuracy: 0.8755 - loss: 0.3121 - val_accur
acy: 0.8719 - val_loss: 0.3508
Epoch 100/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 167ms/step - accuracy: 0.8785 - loss: 0.3107 - val_accur
acy: 0.8696 - val_loss: 0.3507
Epoch 101/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 164ms/step - accuracy: 0.8691 - loss: 0.3103 - val_accur
acy: 0.8673 - val_loss: 0.3511
Epoch 102/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 157ms/step - accuracy: 0.8763 - loss: 0.3111 - val_accur
acy: 0.8650 - val_loss: 0.3554
Epoch 103/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 165ms/step - accuracy: 0.8726 - loss: 0.3140 - val_accur
acy: 0.8673 - val_loss: 0.3564
Epoch 104/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 155ms/step - accuracy: 0.8860 - loss: 0.3008 - val_accur
acy: 0.8696 - val_loss: 0.3527
Epoch 105/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 164ms/step - accuracy: 0.8779 - loss: 0.3050 - val_accur
acy: 0.8673 - val_loss: 0.3500
Epoch 106/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 153ms/step - accuracy: 0.8734 - loss: 0.3016 - val_accur
acy: 0.8673 - val_loss: 0.3558
Epoch 107/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 155ms/step - accuracy: 0.8700 - loss: 0.3271 - val_accur
acy: 0.8673 - val_loss: 0.3666
Epoch 108/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 156ms/step - accuracy: 0.8764 - loss: 0.3071 - val_accur
acy: 0.8650 - val_loss: 0.3706
Epoch 109/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 158ms/step - accuracy: 0.8797 - loss: 0.3014 - val_accur
acy: 0.8627 - val_loss: 0.3662
Epoch 110/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 155ms/step - accuracy: 0.8883 - loss: 0.2993 - val_accur
acy: 0.8673 - val_loss: 0.3578
Epoch 111/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 160ms/step - accuracy: 0.8821 - loss: 0.3036 - val_accur
acy: 0.8627 - val_loss: 0.3553
Epoch 112/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 153ms/step - accuracy: 0.8758 - loss: 0.3158 - val_accur
acy: 0.8627 - val_loss: 0.3599
```

```
Epoch 113/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 156ms/step - accuracy: 0.8799 - loss: 0.2983 - val_accur
acy: 0.8650 - val_loss: 0.3716
Epoch 114/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 189ms/step - accuracy: 0.8674 - loss: 0.3049 - val_accur
acy: 0.8650 - val_loss: 0.3700
Epoch 115/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 176ms/step - accuracy: 0.8790 - loss: 0.3007 - val_accur
acy: 0.8581 - val_loss: 0.3637
Epoch 116/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 175ms/step - accuracy: 0.8836 - loss: 0.2955 - val_accur
acy: 0.8627 - val_loss: 0.3595
Epoch 117/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 172ms/step - accuracy: 0.8784 - loss: 0.3062 - val_accur
acy: 0.8650 - val_loss: 0.3558
Epoch 118/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 174ms/step - accuracy: 0.8786 - loss: 0.2945 - val_accur
acy: 0.8627 - val_loss: 0.3570
Epoch 119/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 173ms/step - accuracy: 0.8804 - loss: 0.2946 - val_accur
acy: 0.8627 - val_loss: 0.3591
Epoch 120/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 169ms/step - accuracy: 0.8739 - loss: 0.3023 - val_accur
acy: 0.8604 - val_loss: 0.3604
Epoch 121/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 164ms/step - accuracy: 0.8748 - loss: 0.2949 - val_accur
acy: 0.8650 - val_loss: 0.3587
Epoch 122/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 173ms/step - accuracy: 0.8820 - loss: 0.3026 - val_accur
acy: 0.8627 - val_loss: 0.3594
Epoch 123/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 170ms/step - accuracy: 0.8841 - loss: 0.2924 - val_accur
acy: 0.8604 - val_loss: 0.3649
Epoch 124/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 176ms/step - accuracy: 0.8809 - loss: 0.2972 - val_accur
acy: 0.8581 - val_loss: 0.3644
Epoch 125/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 160ms/step - accuracy: 0.8903 - loss: 0.2902 - val_accur
acy: 0.8627 - val_loss: 0.3582
Epoch 126/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 162ms/step - accuracy: 0.8843 - loss: 0.2915 - val_accur
acy: 0.8650 - val_loss: 0.3552
Epoch 127/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 159ms/step - accuracy: 0.8768 - loss: 0.2927 - val_accur
acy: 0.8650 - val_loss: 0.3551
Epoch 128/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 152ms/step - accuracy: 0.8751 - loss: 0.2917 - val_accur
acy: 0.8650 - val_loss: 0.3556
Epoch 129/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 158ms/step - accuracy: 0.8808 - loss: 0.2839 - val_accur
acy: 0.8650 - val_loss: 0.3554
Epoch 130/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 156ms/step - accuracy: 0.8791 - loss: 0.2958 - val_accur
acy: 0.8650 - val_loss: 0.3587
Epoch 131/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 161ms/step - accuracy: 0.8864 - loss: 0.3000 - val_accur
```

```
acy: 0.8650 - val_loss: 0.3600
Epoch 132/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 164ms/step - accuracy: 0.8898 - loss: 0.2900 - val_accur
acy: 0.8650 - val_loss: 0.3561
Epoch 133/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 159ms/step - accuracy: 0.8805 - loss: 0.2947 - val_accur
acy: 0.8650 - val_loss: 0.3550
Epoch 134/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 166ms/step - accuracy: 0.8918 - loss: 0.2886 - val_accur
acy: 0.8627 - val_loss: 0.3601
Epoch 135/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 155ms/step - accuracy: 0.8755 - loss: 0.2885 - val_accur
acy: 0.8650 - val_loss: 0.3599
Epoch 136/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 164ms/step - accuracy: 0.8872 - loss: 0.2850 - val_accur
acy: 0.8650 - val_loss: 0.3556
Epoch 137/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 157ms/step - accuracy: 0.8838 - loss: 0.2855 - val_accur
acy: 0.8650 - val_loss: 0.3537
Epoch 138/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 155ms/step - accuracy: 0.8797 - loss: 0.2909 - val_accur
acy: 0.8673 - val_loss: 0.3585
Epoch 139/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 158ms/step - accuracy: 0.8818 - loss: 0.2921 - val_accur
acy: 0.8650 - val_loss: 0.3664
Epoch 140/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 159ms/step - accuracy: 0.8797 - loss: 0.2828 - val_accur
acy: 0.8581 - val_loss: 0.3698
Epoch 141/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 166ms/step - accuracy: 0.8898 - loss: 0.2814 - val_accur
acy: 0.8627 - val_loss: 0.3645
Epoch 142/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 164ms/step - accuracy: 0.8824 - loss: 0.2938 - val_accur
acy: 0.8627 - val_loss: 0.3615
Epoch 143/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 165ms/step - accuracy: 0.8894 - loss: 0.2736 - val_accur
acy: 0.8673 - val_loss: 0.3639
Epoch 144/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 165ms/step - accuracy: 0.8939 - loss: 0.2829 - val_accur
acy: 0.8650 - val_loss: 0.3617
Epoch 145/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 159ms/step - accuracy: 0.8918 - loss: 0.2772 - val_accur
acy: 0.8650 - val_loss: 0.3597
Epoch 146/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 166ms/step - accuracy: 0.8860 - loss: 0.2699 - val_accur
acy: 0.8650 - val_loss: 0.3600
Epoch 147/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 156ms/step - accuracy: 0.8915 - loss: 0.2733 - val_accur
acy: 0.8673 - val_loss: 0.3636
Epoch 148/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 155ms/step - accuracy: 0.8945 - loss: 0.2696 - val_accur
acy: 0.8650 - val_loss: 0.3668
Epoch 149/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 162ms/step - accuracy: 0.8788 - loss: 0.2843 - val_accur
acy: 0.8604 - val_loss: 0.3694
Epoch 150/300
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 166ms/step - accuracy: 0.8885 - loss: 0.2700 - val_accur
acy: 0.8650 - val_loss: 0.3694
Epoch 151/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 167ms/step - accuracy: 0.8814 - loss: 0.2819 - val_accur
acy: 0.8650 - val_loss: 0.3653
Epoch 152/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 162ms/step - accuracy: 0.8869 - loss: 0.2721 - val_accur
acy: 0.8673 - val_loss: 0.3619
Epoch 153/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 159ms/step - accuracy: 0.8910 - loss: 0.2848 - val_accur
acy: 0.8673 - val_loss: 0.3644
Epoch 154/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 162ms/step - accuracy: 0.8842 - loss: 0.2790 - val_accur
acy: 0.8627 - val_loss: 0.3706
Epoch 155/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 160ms/step - accuracy: 0.8931 - loss: 0.2650 - val_accur
acy: 0.8650 - val_loss: 0.3711
Epoch 156/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 157ms/step - accuracy: 0.8880 - loss: 0.2698 - val_accur
acy: 0.8673 - val_loss: 0.3667
Epoch 157/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 160ms/step - accuracy: 0.8817 - loss: 0.2744 - val_accur
acy: 0.8673 - val_loss: 0.3649
Epoch 158/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 170ms/step - accuracy: 0.8798 - loss: 0.2809 - val_accur
acy: 0.8627 - val_loss: 0.3690
Epoch 159/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 172ms/step - accuracy: 0.8916 - loss: 0.2820 - val_accur
acy: 0.8604 - val_loss: 0.3729
Epoch 160/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 167ms/step - accuracy: 0.8795 - loss: 0.2674 - val_accur
acy: 0.8627 - val_loss: 0.3682
Epoch 161/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 176ms/step - accuracy: 0.8804 - loss: 0.2715 - val_accur
acy: 0.8673 - val_loss: 0.3648
Epoch 162/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 166ms/step - accuracy: 0.8820 - loss: 0.2712 - val_accur
acy: 0.8673 - val_loss: 0.3666
Epoch 163/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 168ms/step - accuracy: 0.8822 - loss: 0.2844 - val_accur
acy: 0.8627 - val_loss: 0.3713
Epoch 164/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 172ms/step - accuracy: 0.8893 - loss: 0.2615 - val_accur
acy: 0.8627 - val_loss: 0.3744
Epoch 165/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 169ms/step - accuracy: 0.8919 - loss: 0.2696 - val_accur
acy: 0.8604 - val_loss: 0.3742
Epoch 166/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 171ms/step - accuracy: 0.8896 - loss: 0.2781 - val_accur
acy: 0.8627 - val_loss: 0.3712
Epoch 167/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 176ms/step - accuracy: 0.8908 - loss: 0.2707 - val_accur
acy: 0.8627 - val_loss: 0.3697
Epoch 168/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 164ms/step - accuracy: 0.8878 - loss: 0.2734 - val_accur
acy: 0.8627 - val_loss: 0.3717
```

```
Epoch 169/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 157ms/step - accuracy: 0.8939 - loss: 0.2609 - val_accur
acy: 0.8650 - val_loss: 0.3705
Epoch 170/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 158ms/step - accuracy: 0.8896 - loss: 0.2639 - val_accur
acy: 0.8650 - val_loss: 0.3726
Epoch 171/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 162ms/step - accuracy: 0.8892 - loss: 0.2710 - val_accur
acy: 0.8627 - val_loss: 0.3733
Epoch 171: early stopping
Restoring model weights from the end of the best epoch: 71.
```



In [296…

```
# Evaluate
train_loss, train_acc = model_2d_cnn2.evaluate(X_train_2d, y_train_seq)
val_loss, val_acc = model_2d_cnn2.evaluate(X_val_2d, y_val_seq)

print(f"Training Accuracy: {train_acc:.4f}")
print(f"Validation Accuracy: {val_acc:.4f}")
```

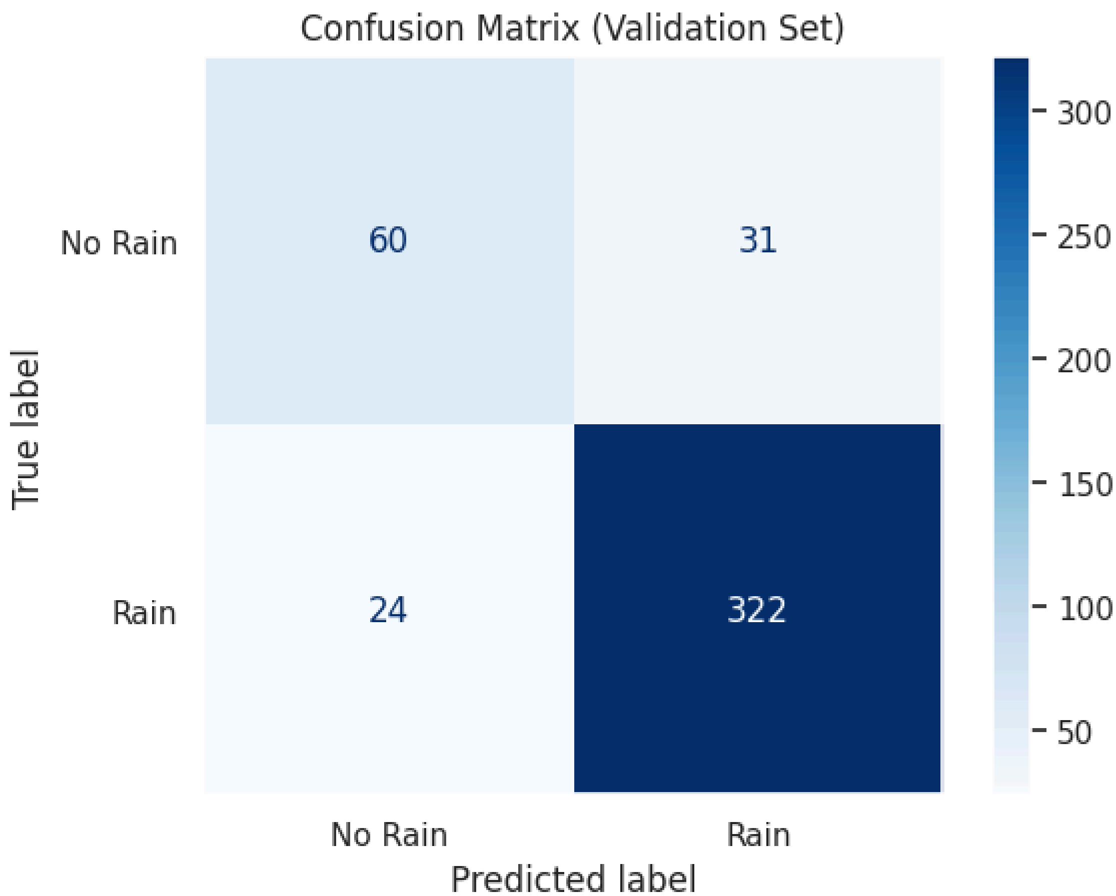
```
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8765 - loss: 0.3000
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8755 - loss: 0.3363
Training Accuracy: 0.8826
Validation Accuracy: 0.8741
```

In [297…

```
# Predict probabilities
y_val_probs = model_2d_cnn2.predict(X_val_2d)

# Convert probabilities to binary predictions
y_val_preds = (y_val_probs > 0.5).astype("int32")

# Generate confusion matrix
cm2 = confusion_matrix(y_val_seq, y_val_preds)
```

```
# Display
disp = ConfusionMatrixDisplay(confusion_matrix=cm2, display_labels=["No Rain", "Rai
disp.plot(cmap=plt.cm.Blues)
plt.title("Confusion Matrix (Validation Set)")
plt.grid(False)
plt.show()
```

```
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step
```



In [298…

```
# Apply to test set

X_test_2d = X_test_cnn.reshape(-1, window_size, X.shape[1], 1)
y_test_pred = model_2d_cnn2.predict(X_test_2d).flatten()

submission = pd.DataFrame({
    'id': submission_ids,
    'rainfall': y_test_pred
})

# Save to CSV
submission.to_csv(PATH + 'submission_2d_cnn2.csv', index=False)
```

```
23/23 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
```

In [299…

```
model_tf3 = tf.keras.Sequential([
    tf.keras.layers.Conv2D(32, (4, 8), padding='same', activation='relu', input_sha
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling2D((2, 2)),
```

```
    tf.keras.layers.SpatialDropout2D(0.5),

    tf.keras.layers.Conv2D(16, (2, 4), padding='same', activation='relu'),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.SpatialDropout2D(0.5),

    tf.keras.layers.Conv2D(16, (2, 4), padding='same', activation='relu'),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.SpatialDropout2D(0.5),

    tf.keras.layers.GlobalAveragePooling2D(),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

model_tf3.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=0.005),
    loss=tf.keras.losses.BinaryCrossentropy(),
    metrics=['accuracy']
)

# Print summary
model_tf3.summary()
# Fit the model
history = model_tf3.fit(
    X_train_2d, y_train_seq,
    validation_data=(X_val_2d, y_val_seq),
    epochs=300,
    batch_size=1024,
    callbacks=[early_stopping]
)
# Plot losses
plt.figure(figsize=(8, 5))
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Val Loss')
plt.title("Training & Validation Loss")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.grid(True)
plt.legend()
plt.show()
```

```
/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

Model: "sequential_1"

Layer (type)	Output Shape	
conv2d (Conv2D)	(None, 8, 12, 32)	
batch_normalization (BatchNormalization)	(None, 8, 12, 32)	
max_pooling2d_2 (MaxPooling2D)	(None, 4, 6, 32)	
spatial_dropout2d (SpatialDropout2D)	(None, 4, 6, 32)	
conv2d_1 (Conv2D)	(None, 4, 6, 16)	
batch_normalization_1 (BatchNormalization)	(None, 4, 6, 16)	
max_pooling2d_3 (MaxPooling2D)	(None, 2, 3, 16)	
spatial_dropout2d_1 (SpatialDropout2D)	(None, 2, 3, 16)	
conv2d_2 (Conv2D)	(None, 2, 3, 16)	
batch_normalization_2 (BatchNormalization)	(None, 2, 3, 16)	
max_pooling2d_4 (MaxPooling2D)	(None, 1, 1, 16)	
spatial_dropout2d_2 (SpatialDropout2D)	(None, 1, 1, 16)	
global_average_pooling2d (GlobalAveragePooling2D)	(None, 16)	
dense_1 (Dense)	(None, 1)	

Total params: 7,505 (29.32 KB)

Trainable params: 7,377 (28.82 KB)

Non-trainable params: 128 (512.00 B)

```
Epoch 1/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 3s 400ms/step - accuracy: 0.4807 - loss: 1.3247 - val_accur
acy: 0.6018 - val_loss: 0.6828
Epoch 2/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.5853 - loss: 1.0343 - val_accur
acy: 0.7872 - val_loss: 0.6283
Epoch 3/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 115ms/step - accuracy: 0.6359 - loss: 0.9087 - val_accur
acy: 0.7918 - val_loss: 0.5992
Epoch 4/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.6867 - loss: 0.8377 - val_accur
acy: 0.7849 - val_loss: 0.6046
Epoch 5/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.7011 - loss: 0.7681 - val_accur
acy: 0.7735 - val_loss: 0.6199
Epoch 6/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.7011 - loss: 0.7697 - val_accur
acy: 0.7620 - val_loss: 0.6352
Epoch 7/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.7128 - loss: 0.6685 - val_accur
acy: 0.7574 - val_loss: 0.6401
Epoch 8/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.7024 - loss: 0.6696 - val_accur
acy: 0.7529 - val_loss: 0.6317
Epoch 9/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.7208 - loss: 0.6433 - val_accur
acy: 0.7666 - val_loss: 0.6186
Epoch 10/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.6901 - loss: 0.6362 - val_accur
acy: 0.7735 - val_loss: 0.6065
Epoch 11/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 118ms/step - accuracy: 0.7238 - loss: 0.6026 - val_accur
acy: 0.7872 - val_loss: 0.5951
Epoch 12/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.7169 - loss: 0.5972 - val_accur
acy: 0.7918 - val_loss: 0.5844
Epoch 13/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 119ms/step - accuracy: 0.7321 - loss: 0.5849 - val_accur
acy: 0.7941 - val_loss: 0.5750
Epoch 14/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.7518 - loss: 0.5343 - val_accur
acy: 0.8009 - val_loss: 0.5681
Epoch 15/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.7469 - loss: 0.5551 - val_accur
acy: 0.8055 - val_loss: 0.5634
Epoch 16/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.7559 - loss: 0.5319 - val_accur
acy: 0.8055 - val_loss: 0.5593
Epoch 17/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.7502 - loss: 0.5322 - val_accur
acy: 0.8032 - val_loss: 0.5543
Epoch 18/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.7562 - loss: 0.5297 - val_accur
acy: 0.8055 - val_loss: 0.5479
Epoch 19/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.7619 - loss: 0.5093 - val_accur
```

```
acy: 0.8055 - val_loss: 0.5396
Epoch 20/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.7560 - loss: 0.5028 - val_accur
acy: 0.8055 - val_loss: 0.5290
Epoch 21/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.7670 - loss: 0.4972 - val_accur
acy: 0.8009 - val_loss: 0.5178
Epoch 22/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.7757 - loss: 0.4924 - val_accur
acy: 0.8055 - val_loss: 0.5074
Epoch 23/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - accuracy: 0.7769 - loss: 0.4948 - val_accur
acy: 0.8055 - val_loss: 0.4992
Epoch 24/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 117ms/step - accuracy: 0.7843 - loss: 0.4952 - val_accur
acy: 0.7986 - val_loss: 0.4915
Epoch 25/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 117ms/step - accuracy: 0.7948 - loss: 0.4705 - val_accur
acy: 0.7963 - val_loss: 0.4853
Epoch 26/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 122ms/step - accuracy: 0.7855 - loss: 0.4676 - val_accur
acy: 0.8009 - val_loss: 0.4795
Epoch 27/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 122ms/step - accuracy: 0.7885 - loss: 0.4735 - val_accur
acy: 0.8009 - val_loss: 0.4751
Epoch 28/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 117ms/step - accuracy: 0.8164 - loss: 0.4487 - val_accur
acy: 0.8009 - val_loss: 0.4718
Epoch 29/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.8004 - loss: 0.4665 - val_accur
acy: 0.8009 - val_loss: 0.4688
Epoch 30/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.7987 - loss: 0.4536 - val_accur
acy: 0.8009 - val_loss: 0.4660
Epoch 31/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.8129 - loss: 0.4438 - val_accur
acy: 0.7986 - val_loss: 0.4625
Epoch 32/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 115ms/step - accuracy: 0.8189 - loss: 0.4442 - val_accur
acy: 0.8009 - val_loss: 0.4572
Epoch 33/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.8204 - loss: 0.4385 - val_accur
acy: 0.8078 - val_loss: 0.4522
Epoch 34/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 118ms/step - accuracy: 0.8147 - loss: 0.4289 - val_accur
acy: 0.8101 - val_loss: 0.4475
Epoch 35/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - accuracy: 0.8205 - loss: 0.4262 - val_accur
acy: 0.8124 - val_loss: 0.4435
Epoch 36/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 117ms/step - accuracy: 0.8233 - loss: 0.4249 - val_accur
acy: 0.8146 - val_loss: 0.4402
Epoch 37/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.8247 - loss: 0.4468 - val_accur
acy: 0.8146 - val_loss: 0.4371
Epoch 38/300
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8256 - loss: 0.4260 - val_accur
acy: 0.8146 - val_loss: 0.4346
Epoch 39/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 115ms/step - accuracy: 0.8234 - loss: 0.4188 - val_accur
acy: 0.8169 - val_loss: 0.4326
Epoch 40/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8311 - loss: 0.4271 - val_accur
acy: 0.8192 - val_loss: 0.4312
Epoch 41/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8374 - loss: 0.4058 - val_accur
acy: 0.8169 - val_loss: 0.4296
Epoch 42/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8255 - loss: 0.4011 - val_accur
acy: 0.8169 - val_loss: 0.4278
Epoch 43/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8351 - loss: 0.4087 - val_accur
acy: 0.8215 - val_loss: 0.4251
Epoch 44/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8349 - loss: 0.3948 - val_accur
acy: 0.8215 - val_loss: 0.4225
Epoch 45/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8408 - loss: 0.3991 - val_accur
acy: 0.8192 - val_loss: 0.4200
Epoch 46/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8351 - loss: 0.4144 - val_accur
acy: 0.8238 - val_loss: 0.4173
Epoch 47/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8407 - loss: 0.4053 - val_accur
acy: 0.8238 - val_loss: 0.4154
Epoch 48/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.8441 - loss: 0.3907 - val_accur
acy: 0.8238 - val_loss: 0.4129
Epoch 49/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8527 - loss: 0.3891 - val_accur
acy: 0.8261 - val_loss: 0.4103
Epoch 50/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.8485 - loss: 0.3900 - val_accur
acy: 0.8284 - val_loss: 0.4082
Epoch 51/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 118ms/step - accuracy: 0.8411 - loss: 0.3926 - val_accur
acy: 0.8284 - val_loss: 0.4072
Epoch 52/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8461 - loss: 0.3950 - val_accur
acy: 0.8307 - val_loss: 0.4067
Epoch 53/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8514 - loss: 0.3816 - val_accur
acy: 0.8330 - val_loss: 0.4050
Epoch 54/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8408 - loss: 0.3930 - val_accur
acy: 0.8307 - val_loss: 0.4026
Epoch 55/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8338 - loss: 0.3841 - val_accur
acy: 0.8352 - val_loss: 0.3996
Epoch 56/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8430 - loss: 0.3853 - val_accur
acy: 0.8352 - val_loss: 0.3974
```

```
Epoch 57/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8474 - loss: 0.3822 - val_accur
acy: 0.8352 - val_loss: 0.3969
Epoch 58/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8437 - loss: 0.3879 - val_accur
acy: 0.8352 - val_loss: 0.3960
Epoch 59/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8607 - loss: 0.3758 - val_accur
acy: 0.8352 - val_loss: 0.3947
Epoch 60/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8520 - loss: 0.3790 - val_accur
acy: 0.8375 - val_loss: 0.3937
Epoch 61/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.8477 - loss: 0.3751 - val_accur
acy: 0.8375 - val_loss: 0.3922
Epoch 62/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8489 - loss: 0.3727 - val_accur
acy: 0.8375 - val_loss: 0.3910
Epoch 63/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8500 - loss: 0.3697 - val_accur
acy: 0.8421 - val_loss: 0.3916
Epoch 64/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8502 - loss: 0.3985 - val_accur
acy: 0.8375 - val_loss: 0.3908
Epoch 65/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8459 - loss: 0.3834 - val_accur
acy: 0.8375 - val_loss: 0.3890
Epoch 66/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8466 - loss: 0.3713 - val_accur
acy: 0.8398 - val_loss: 0.3863
Epoch 67/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8510 - loss: 0.3637 - val_accur
acy: 0.8421 - val_loss: 0.3846
Epoch 68/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8422 - loss: 0.3786 - val_accur
acy: 0.8421 - val_loss: 0.3839
Epoch 69/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 107ms/step - accuracy: 0.8444 - loss: 0.3676 - val_accur
acy: 0.8421 - val_loss: 0.3841
Epoch 70/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8493 - loss: 0.3643 - val_accur
acy: 0.8467 - val_loss: 0.3837
Epoch 71/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8402 - loss: 0.3718 - val_accur
acy: 0.8467 - val_loss: 0.3818
Epoch 72/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8419 - loss: 0.3737 - val_accur
acy: 0.8490 - val_loss: 0.3800
Epoch 73/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8508 - loss: 0.3805 - val_accur
acy: 0.8490 - val_loss: 0.3781
Epoch 74/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8532 - loss: 0.3624 - val_accur
acy: 0.8513 - val_loss: 0.3771
Epoch 75/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8520 - loss: 0.3640 - val_accur
```

```
acy: 0.8513 - val_loss: 0.3779
Epoch 76/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8625 - loss: 0.3533 - val_accur
acy: 0.8490 - val_loss: 0.3785
Epoch 77/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 123ms/step - accuracy: 0.8490 - loss: 0.3713 - val_accur
acy: 0.8490 - val_loss: 0.3787
Epoch 78/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 106ms/step - accuracy: 0.8608 - loss: 0.3513 - val_accur
acy: 0.8513 - val_loss: 0.3779
Epoch 79/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8503 - loss: 0.3551 - val_accur
acy: 0.8490 - val_loss: 0.3760
Epoch 80/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8522 - loss: 0.3620 - val_accur
acy: 0.8467 - val_loss: 0.3743
Epoch 81/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8517 - loss: 0.3565 - val_accur
acy: 0.8490 - val_loss: 0.3730
Epoch 82/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8623 - loss: 0.3666 - val_accur
acy: 0.8490 - val_loss: 0.3716
Epoch 83/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8488 - loss: 0.3584 - val_accur
acy: 0.8490 - val_loss: 0.3700
Epoch 84/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8610 - loss: 0.3508 - val_accur
acy: 0.8490 - val_loss: 0.3687
Epoch 85/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8521 - loss: 0.3617 - val_accur
acy: 0.8513 - val_loss: 0.3673
Epoch 86/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 107ms/step - accuracy: 0.8624 - loss: 0.3626 - val_accur
acy: 0.8535 - val_loss: 0.3679
Epoch 87/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 107ms/step - accuracy: 0.8468 - loss: 0.3649 - val_accur
acy: 0.8558 - val_loss: 0.3689
Epoch 88/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8611 - loss: 0.3567 - val_accur
acy: 0.8581 - val_loss: 0.3680
Epoch 89/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8569 - loss: 0.3529 - val_accur
acy: 0.8558 - val_loss: 0.3642
Epoch 90/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 117ms/step - accuracy: 0.8652 - loss: 0.3324 - val_accur
acy: 0.8581 - val_loss: 0.3599
Epoch 91/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 118ms/step - accuracy: 0.8618 - loss: 0.3583 - val_accur
acy: 0.8581 - val_loss: 0.3574
Epoch 92/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 119ms/step - accuracy: 0.8570 - loss: 0.3580 - val_accur
acy: 0.8558 - val_loss: 0.3574
Epoch 93/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 130ms/step - accuracy: 0.8618 - loss: 0.3415 - val_accur
acy: 0.8558 - val_loss: 0.3568
Epoch 94/300
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8577 - loss: 0.3474 - val_accur
acy: 0.8558 - val_loss: 0.3581
Epoch 95/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - accuracy: 0.8593 - loss: 0.3464 - val_accur
acy: 0.8535 - val_loss: 0.3591
Epoch 96/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8595 - loss: 0.3377 - val_accur
acy: 0.8535 - val_loss: 0.3603
Epoch 97/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 123ms/step - accuracy: 0.8693 - loss: 0.3442 - val_accur
acy: 0.8535 - val_loss: 0.3591
Epoch 98/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 115ms/step - accuracy: 0.8659 - loss: 0.3508 - val_accur
acy: 0.8604 - val_loss: 0.3576
Epoch 99/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.8615 - loss: 0.3410 - val_accur
acy: 0.8627 - val_loss: 0.3531
Epoch 100/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.8599 - loss: 0.3405 - val_accur
acy: 0.8581 - val_loss: 0.3498
Epoch 101/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 126ms/step - accuracy: 0.8584 - loss: 0.3443 - val_accur
acy: 0.8604 - val_loss: 0.3492
Epoch 102/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 119ms/step - accuracy: 0.8605 - loss: 0.3408 - val_accur
acy: 0.8558 - val_loss: 0.3520
Epoch 103/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.8652 - loss: 0.3459 - val_accur
acy: 0.8535 - val_loss: 0.3551
Epoch 104/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - accuracy: 0.8564 - loss: 0.3476 - val_accur
acy: 0.8558 - val_loss: 0.3551
Epoch 105/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8563 - loss: 0.3499 - val_accur
acy: 0.8627 - val_loss: 0.3521
Epoch 106/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.8528 - loss: 0.3437 - val_accur
acy: 0.8673 - val_loss: 0.3481
Epoch 107/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8627 - loss: 0.3433 - val_accur
acy: 0.8673 - val_loss: 0.3460
Epoch 108/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8625 - loss: 0.3324 - val_accur
acy: 0.8696 - val_loss: 0.3443
Epoch 109/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 107ms/step - accuracy: 0.8529 - loss: 0.3477 - val_accur
acy: 0.8650 - val_loss: 0.3454
Epoch 110/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8639 - loss: 0.3276 - val_accur
acy: 0.8650 - val_loss: 0.3454
Epoch 111/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8646 - loss: 0.3364 - val_accur
acy: 0.8650 - val_loss: 0.3457
Epoch 112/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8608 - loss: 0.3311 - val_accur
acy: 0.8650 - val_loss: 0.3455
```

```
Epoch 113/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8678 - loss: 0.3229 - val_accur
acy: 0.8650 - val_loss: 0.3421
Epoch 114/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8618 - loss: 0.3390 - val_accur
acy: 0.8696 - val_loss: 0.3375
Epoch 115/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8684 - loss: 0.3356 - val_accur
acy: 0.8673 - val_loss: 0.3326
Epoch 116/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8626 - loss: 0.3357 - val_accur
acy: 0.8673 - val_loss: 0.3333
Epoch 117/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8617 - loss: 0.3383 - val_accur
acy: 0.8650 - val_loss: 0.3367
Epoch 118/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 107ms/step - accuracy: 0.8616 - loss: 0.3376 - val_accur
acy: 0.8673 - val_loss: 0.3400
Epoch 119/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8629 - loss: 0.3371 - val_accur
acy: 0.8673 - val_loss: 0.3416
Epoch 120/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8600 - loss: 0.3348 - val_accur
acy: 0.8650 - val_loss: 0.3416
Epoch 121/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8645 - loss: 0.3427 - val_accur
acy: 0.8673 - val_loss: 0.3398
Epoch 122/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8564 - loss: 0.3437 - val_accur
acy: 0.8719 - val_loss: 0.3392
Epoch 123/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8678 - loss: 0.3255 - val_accur
acy: 0.8719 - val_loss: 0.3394
Epoch 124/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 106ms/step - accuracy: 0.8617 - loss: 0.3285 - val_accur
acy: 0.8696 - val_loss: 0.3389
Epoch 125/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8692 - loss: 0.3342 - val_accur
acy: 0.8696 - val_loss: 0.3371
Epoch 126/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8657 - loss: 0.3238 - val_accur
acy: 0.8719 - val_loss: 0.3360
Epoch 127/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8680 - loss: 0.3282 - val_accur
acy: 0.8764 - val_loss: 0.3360
Epoch 128/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8680 - loss: 0.3316 - val_accur
acy: 0.8696 - val_loss: 0.3383
Epoch 129/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8674 - loss: 0.3200 - val_accur
acy: 0.8650 - val_loss: 0.3408
Epoch 130/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8731 - loss: 0.3252 - val_accur
acy: 0.8673 - val_loss: 0.3412
Epoch 131/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8798 - loss: 0.3161 - val_accur
```

```
acy: 0.8719 - val_loss: 0.3400
Epoch 132/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 107ms/step - accuracy: 0.8707 - loss: 0.3156 - val_accur
acy: 0.8696 - val_loss: 0.3400
Epoch 133/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8656 - loss: 0.3282 - val_accur
acy: 0.8696 - val_loss: 0.3411
Epoch 134/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8691 - loss: 0.3275 - val_accur
acy: 0.8673 - val_loss: 0.3435
Epoch 135/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 107ms/step - accuracy: 0.8689 - loss: 0.3233 - val_accur
acy: 0.8673 - val_loss: 0.3471
Epoch 136/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8729 - loss: 0.3299 - val_accur
acy: 0.8650 - val_loss: 0.3451
Epoch 137/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8722 - loss: 0.3088 - val_accur
acy: 0.8604 - val_loss: 0.3458
Epoch 138/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 106ms/step - accuracy: 0.8710 - loss: 0.3129 - val_accur
acy: 0.8581 - val_loss: 0.3453
Epoch 139/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 107ms/step - accuracy: 0.8754 - loss: 0.3158 - val_accur
acy: 0.8696 - val_loss: 0.3382
Epoch 140/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8654 - loss: 0.3223 - val_accur
acy: 0.8719 - val_loss: 0.3310
Epoch 141/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8690 - loss: 0.3161 - val_accur
acy: 0.8696 - val_loss: 0.3278
Epoch 142/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8741 - loss: 0.3207 - val_accur
acy: 0.8696 - val_loss: 0.3302
Epoch 143/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8740 - loss: 0.3065 - val_accur
acy: 0.8604 - val_loss: 0.3374
Epoch 144/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 107ms/step - accuracy: 0.8661 - loss: 0.3143 - val_accur
acy: 0.8581 - val_loss: 0.3425
Epoch 145/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8691 - loss: 0.3204 - val_accur
acy: 0.8627 - val_loss: 0.3412
Epoch 146/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8661 - loss: 0.3229 - val_accur
acy: 0.8696 - val_loss: 0.3348
Epoch 147/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8721 - loss: 0.3247 - val_accur
acy: 0.8696 - val_loss: 0.3328
Epoch 148/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 106ms/step - accuracy: 0.8749 - loss: 0.2995 - val_accur
acy: 0.8696 - val_loss: 0.3350
Epoch 149/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8609 - loss: 0.3436 - val_accur
acy: 0.8696 - val_loss: 0.3386
Epoch 150/300
```

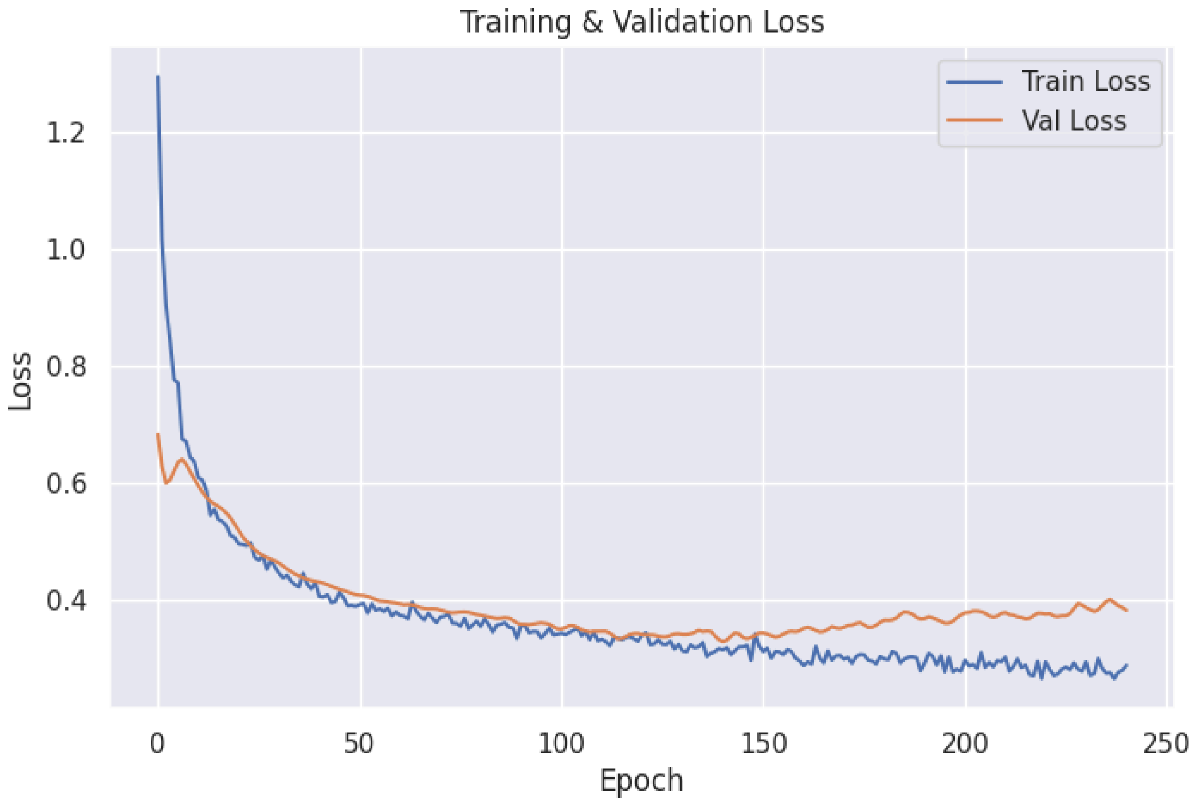
```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8714 - loss: 0.3243 - val_accur
acy: 0.8696 - val_loss: 0.3424
Epoch 151/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8720 - loss: 0.3159 - val_accur
acy: 0.8673 - val_loss: 0.3425
Epoch 152/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 107ms/step - accuracy: 0.8701 - loss: 0.3180 - val_accur
acy: 0.8719 - val_loss: 0.3408
Epoch 153/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8739 - loss: 0.3042 - val_accur
acy: 0.8719 - val_loss: 0.3379
Epoch 154/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8720 - loss: 0.3159 - val_accur
acy: 0.8764 - val_loss: 0.3355
Epoch 155/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8747 - loss: 0.3099 - val_accur
acy: 0.8741 - val_loss: 0.3370
Epoch 156/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8718 - loss: 0.3092 - val_accur
acy: 0.8696 - val_loss: 0.3408
Epoch 157/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8670 - loss: 0.3165 - val_accur
acy: 0.8696 - val_loss: 0.3442
Epoch 158/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 123ms/step - accuracy: 0.8698 - loss: 0.3154 - val_accur
acy: 0.8696 - val_loss: 0.3456
Epoch 159/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8729 - loss: 0.3046 - val_accur
acy: 0.8627 - val_loss: 0.3447
Epoch 160/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8712 - loss: 0.3016 - val_accur
acy: 0.8581 - val_loss: 0.3463
Epoch 161/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.8792 - loss: 0.2898 - val_accur
acy: 0.8558 - val_loss: 0.3506
Epoch 162/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 115ms/step - accuracy: 0.8772 - loss: 0.2961 - val_accur
acy: 0.8558 - val_loss: 0.3524
Epoch 163/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 115ms/step - accuracy: 0.8800 - loss: 0.2941 - val_accur
acy: 0.8581 - val_loss: 0.3502
Epoch 164/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.8769 - loss: 0.3210 - val_accur
acy: 0.8650 - val_loss: 0.3469
Epoch 165/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 121ms/step - accuracy: 0.8830 - loss: 0.3039 - val_accur
acy: 0.8627 - val_loss: 0.3444
Epoch 166/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.8718 - loss: 0.2941 - val_accur
acy: 0.8650 - val_loss: 0.3451
Epoch 167/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8726 - loss: 0.3170 - val_accur
acy: 0.8627 - val_loss: 0.3486
Epoch 168/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.8700 - loss: 0.3016 - val_accur
acy: 0.8604 - val_loss: 0.3534
```

```
Epoch 169/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 120ms/step - accuracy: 0.8674 - loss: 0.3118 - val_accur
acy: 0.8604 - val_loss: 0.3514
Epoch 170/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 122ms/step - accuracy: 0.8761 - loss: 0.3127 - val_accur
acy: 0.8650 - val_loss: 0.3503
Epoch 171/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 123ms/step - accuracy: 0.8810 - loss: 0.2969 - val_accur
acy: 0.8650 - val_loss: 0.3524
Epoch 172/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8757 - loss: 0.3031 - val_accur
acy: 0.8627 - val_loss: 0.3554
Epoch 173/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8811 - loss: 0.2913 - val_accur
acy: 0.8650 - val_loss: 0.3560
Epoch 174/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8776 - loss: 0.3023 - val_accur
acy: 0.8604 - val_loss: 0.3589
Epoch 175/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8692 - loss: 0.3119 - val_accur
acy: 0.8604 - val_loss: 0.3604
Epoch 176/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8791 - loss: 0.3096 - val_accur
acy: 0.8650 - val_loss: 0.3563
Epoch 177/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8631 - loss: 0.3023 - val_accur
acy: 0.8719 - val_loss: 0.3523
Epoch 178/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 107ms/step - accuracy: 0.8734 - loss: 0.2980 - val_accur
acy: 0.8696 - val_loss: 0.3526
Epoch 179/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8831 - loss: 0.2895 - val_accur
acy: 0.8673 - val_loss: 0.3558
Epoch 180/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8704 - loss: 0.3028 - val_accur
acy: 0.8558 - val_loss: 0.3609
Epoch 181/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8843 - loss: 0.3002 - val_accur
acy: 0.8490 - val_loss: 0.3647
Epoch 182/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 106ms/step - accuracy: 0.8785 - loss: 0.3026 - val_accur
acy: 0.8535 - val_loss: 0.3643
Epoch 183/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 105ms/step - accuracy: 0.8722 - loss: 0.3118 - val_accur
acy: 0.8535 - val_loss: 0.3650
Epoch 184/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8749 - loss: 0.3182 - val_accur
acy: 0.8535 - val_loss: 0.3682
Epoch 185/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8777 - loss: 0.2946 - val_accur
acy: 0.8513 - val_loss: 0.3746
Epoch 186/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.8710 - loss: 0.3023 - val_accur
acy: 0.8535 - val_loss: 0.3786
Epoch 187/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8791 - loss: 0.3013 - val_accur
```

```
acy: 0.8535 - val_loss: 0.3776
Epoch 188/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8665 - loss: 0.3075 - val_accur
acy: 0.8535 - val_loss: 0.3745
Epoch 189/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8732 - loss: 0.3090 - val_accur
acy: 0.8558 - val_loss: 0.3692
Epoch 190/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8884 - loss: 0.2804 - val_accur
acy: 0.8627 - val_loss: 0.3664
Epoch 191/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8863 - loss: 0.2976 - val_accur
acy: 0.8581 - val_loss: 0.3671
Epoch 192/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8731 - loss: 0.3133 - val_accur
acy: 0.8581 - val_loss: 0.3706
Epoch 193/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8811 - loss: 0.3058 - val_accur
acy: 0.8558 - val_loss: 0.3685
Epoch 194/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8798 - loss: 0.2929 - val_accur
acy: 0.8627 - val_loss: 0.3657
Epoch 195/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8821 - loss: 0.3104 - val_accur
acy: 0.8650 - val_loss: 0.3627
Epoch 196/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 107ms/step - accuracy: 0.8841 - loss: 0.2777 - val_accur
acy: 0.8673 - val_loss: 0.3594
Epoch 197/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8765 - loss: 0.3095 - val_accur
acy: 0.8673 - val_loss: 0.3595
Epoch 198/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8791 - loss: 0.2800 - val_accur
acy: 0.8696 - val_loss: 0.3629
Epoch 199/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8763 - loss: 0.2831 - val_accur
acy: 0.8627 - val_loss: 0.3686
Epoch 200/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8770 - loss: 0.2814 - val_accur
acy: 0.8558 - val_loss: 0.3735
Epoch 201/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8750 - loss: 0.2995 - val_accur
acy: 0.8558 - val_loss: 0.3771
Epoch 202/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8756 - loss: 0.2919 - val_accur
acy: 0.8535 - val_loss: 0.3778
Epoch 203/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8898 - loss: 0.2848 - val_accur
acy: 0.8535 - val_loss: 0.3801
Epoch 204/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8756 - loss: 0.2858 - val_accur
acy: 0.8513 - val_loss: 0.3803
Epoch 205/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8797 - loss: 0.3164 - val_accur
acy: 0.8535 - val_loss: 0.3789
Epoch 206/300
```

```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8795 - loss: 0.2881 - val_accur
acy: 0.8558 - val_loss: 0.3765
Epoch 207/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8754 - loss: 0.2923 - val_accur
acy: 0.8558 - val_loss: 0.3719
Epoch 208/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 115ms/step - accuracy: 0.8851 - loss: 0.2933 - val_accur
acy: 0.8604 - val_loss: 0.3698
Epoch 209/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8800 - loss: 0.3006 - val_accur
acy: 0.8558 - val_loss: 0.3715
Epoch 210/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8733 - loss: 0.3010 - val_accur
acy: 0.8467 - val_loss: 0.3760
Epoch 211/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8793 - loss: 0.3027 - val_accur
acy: 0.8421 - val_loss: 0.3776
Epoch 212/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 113ms/step - accuracy: 0.8823 - loss: 0.2818 - val_accur
acy: 0.8535 - val_loss: 0.3741
Epoch 213/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8792 - loss: 0.2880 - val_accur
acy: 0.8535 - val_loss: 0.3730
Epoch 214/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 106ms/step - accuracy: 0.8859 - loss: 0.3040 - val_accur
acy: 0.8604 - val_loss: 0.3703
Epoch 215/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 107ms/step - accuracy: 0.8773 - loss: 0.2854 - val_accur
acy: 0.8650 - val_loss: 0.3691
Epoch 216/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8822 - loss: 0.2938 - val_accur
acy: 0.8673 - val_loss: 0.3666
Epoch 217/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8828 - loss: 0.2753 - val_accur
acy: 0.8627 - val_loss: 0.3677
Epoch 218/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8818 - loss: 0.2758 - val_accur
acy: 0.8604 - val_loss: 0.3719
Epoch 219/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8835 - loss: 0.2973 - val_accur
acy: 0.8535 - val_loss: 0.3765
Epoch 220/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8846 - loss: 0.2662 - val_accur
acy: 0.8558 - val_loss: 0.3761
Epoch 221/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 110ms/step - accuracy: 0.8751 - loss: 0.2942 - val_accur
acy: 0.8581 - val_loss: 0.3750
Epoch 222/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 108ms/step - accuracy: 0.8813 - loss: 0.2829 - val_accur
acy: 0.8535 - val_loss: 0.3756
Epoch 223/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8860 - loss: 0.2778 - val_accur
acy: 0.8581 - val_loss: 0.3718
Epoch 224/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 109ms/step - accuracy: 0.8884 - loss: 0.2811 - val_accur
acy: 0.8627 - val_loss: 0.3706
```

```
Epoch 225/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step - accuracy: 0.8861 - loss: 0.2811 - val_accur
acy: 0.8604 - val_loss: 0.3721
Epoch 226/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 130ms/step - accuracy: 0.8759 - loss: 0.2898 - val_accur
acy: 0.8581 - val_loss: 0.3719
Epoch 227/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 115ms/step - accuracy: 0.8908 - loss: 0.2796 - val_accur
acy: 0.8581 - val_loss: 0.3758
Epoch 228/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 122ms/step - accuracy: 0.8730 - loss: 0.2982 - val_accur
acy: 0.8513 - val_loss: 0.3843
Epoch 229/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.8830 - loss: 0.2875 - val_accur
acy: 0.8375 - val_loss: 0.3930
Epoch 230/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.8764 - loss: 0.2836 - val_accur
acy: 0.8513 - val_loss: 0.3905
Epoch 231/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 117ms/step - accuracy: 0.8753 - loss: 0.3000 - val_accur
acy: 0.8535 - val_loss: 0.3861
Epoch 232/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 115ms/step - accuracy: 0.8795 - loss: 0.2727 - val_accur
acy: 0.8558 - val_loss: 0.3826
Epoch 233/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 122ms/step - accuracy: 0.8854 - loss: 0.2771 - val_accur
acy: 0.8604 - val_loss: 0.3794
Epoch 234/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 115ms/step - accuracy: 0.8763 - loss: 0.3006 - val_accur
acy: 0.8604 - val_loss: 0.3825
Epoch 235/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 114ms/step - accuracy: 0.8837 - loss: 0.2842 - val_accur
acy: 0.8535 - val_loss: 0.3893
Epoch 236/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 115ms/step - accuracy: 0.8912 - loss: 0.2766 - val_accur
acy: 0.8490 - val_loss: 0.3965
Epoch 237/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 117ms/step - accuracy: 0.8794 - loss: 0.2777 - val_accur
acy: 0.8490 - val_loss: 0.3996
Epoch 238/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 116ms/step - accuracy: 0.8818 - loss: 0.2692 - val_accur
acy: 0.8490 - val_loss: 0.3945
Epoch 239/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 118ms/step - accuracy: 0.8795 - loss: 0.2817 - val_accur
acy: 0.8558 - val_loss: 0.3893
Epoch 240/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 123ms/step - accuracy: 0.8881 - loss: 0.2823 - val_accur
acy: 0.8558 - val_loss: 0.3861
Epoch 241/300
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 111ms/step - accuracy: 0.8779 - loss: 0.2954 - val_accur
acy: 0.8627 - val_loss: 0.3814
Epoch 241: early stopping
Restoring model weights from the end of the best epoch: 141.
```



In [300…

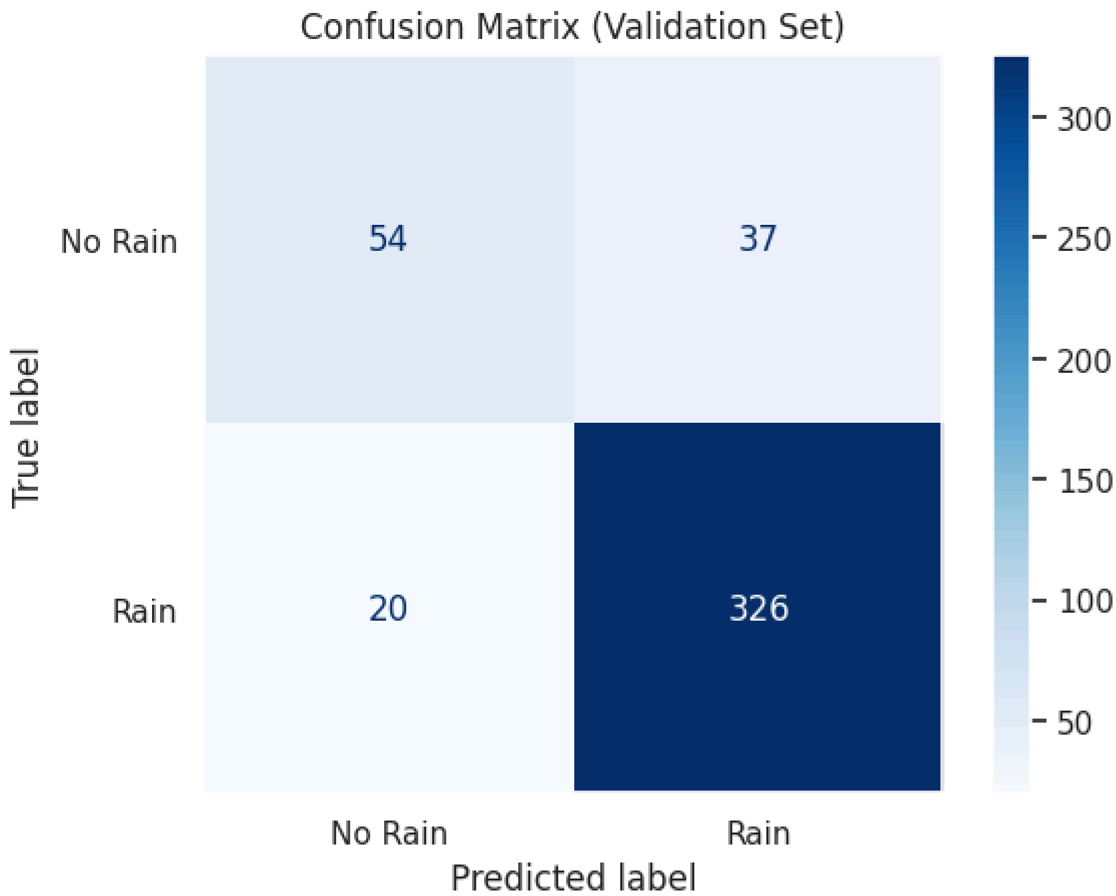
```
# Evaluate
train_loss, train_acc = model_tf3.evaluate(X_train_2d, y_train_seq)
val_loss, val_acc = model_tf3.evaluate(X_val_2d, y_val_seq)

print(f"Training Accuracy: {train_acc:.4f}")
print(f"Validation Accuracy: {val_acc:.4f}")
```

```
55/55 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.8847 - loss: 0.2746
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8704 - loss: 0.3248
Training Accuracy: 0.8889
Validation Accuracy: 0.8696
```

In [301…

```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
import numpy as np
import matplotlib.pyplot as plt

# Predict probabilities
y_val_probs = model_tf3.predict(X_val_2d)

# Convert probabilities to binary predictions
y_val_preds = (y_val_probs > 0.5).astype("int32")

# Generate confusion matrix
cm2 = confusion_matrix(y_val_seq, y_val_preds)

# Display
disp = ConfusionMatrixDisplay(confusion_matrix=cm2, display_labels=["No Rain", "Rai
disp.plot(cmap=plt.cm.Blues)
plt.title("Confusion Matrix (Validation Set)")
plt.grid(False)
plt.show()
```

```
14/14 ━━━━━━━━━━━━━━━━━━━━ 0s 15ms/step
```



In [302…

```
# Apply to test set

y_test_pred = model_tf3.predict(X_test_2d).flatten()

submission = pd.DataFrame({
    'id': submission_ids,
    'rainfall': y_test_pred
})

# Save to CSV
submission.to_csv(PATH + 'submission_2d_cnn3.csv', index=False)
```

```
23/23 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
```

Ensamble

In [303…

```
# List of our submission files
submission_files = [
    "submission_lr.csv",
    "submission_XGboost.csv",
    "submission_XGboost2.csv",
    "submission_nn.csv",
    "submission_2d_cnn.csv",
    "submission_2d_cnn2.csv",
    "submission_2d_cnn3.csv",
    #"submission_knn.csv",
```

```
]

dfs = [pd.read_csv(PATH + f) for f in submission_files]

# Stack all probability columns and compute the mean
probs = pd.concat([df["rainfall"] for df in dfs], axis=1)
avg_probs = probs.mean(axis=1)

# Create final submission
ensemble_submission = pd.DataFrame({
    "id": dfs[0]["id"],
    "probability": avg_probs
})

# Save to CSV
ensemble_submission.to_csv(PATH + "submission_ensemble.csv", index=False)
```