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

# قراءة الملفات
fake_df = pd.read_csv(r"C:\Users\Abdulrahman\Desktop\machine\Fake.csv")
real_df = pd.read_csv(r"C:\Users\Abdulrahman\Desktop\machine\True.csv")

# إضافة label
fake_df['label'] = 0  # Fake
real_df['label'] = 1  # Real

# دمج البيانات
df = pd.concat([fake_df, real_df], ignore_index=True)

# عرض أول صفوف
print(df.head())
print(df['label'].value_counts())
```

```
                                               title  \
0   Donald Trump Sends Out Embarrassing New Year’...
1   Drunk Bragging Trump Staffer Started Russian ...
2   Sheriff David Clarke Becomes An Internet Joke...
3   Trump Is So Obsessed He Even Has Obama’s Name...
4   Pope Francis Just Called Out Donald Trump Dur...

                                                text subject  \
0  Donald Trump just couldn t wish all Americans ...    News
1  House Intelligence Committee Chairman Devin Nu...    News
2  On Friday, it was revealed that former Milwauk...    News
3  On Christmas day, Donald Trump announced that ...    News
4  Pope Francis used his annual Christmas Day mes...    News

                date  label
0  December 31, 2017      0
1  December 31, 2017      0
2  December 30, 2017      0
3  December 29, 2017      0
4  December 25, 2017      0
label
0    23481
1    21417
Name: count, dtype: int64
```

In [3]:
```
import re
import string
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer

# تحميل stopwords
import nltk
nltk.download('stopwords')
stop_words = set(stopwords.words('english'))

stemmer = PorterStemmer()

def clean_text(text):
    # حذف الرموز والأرقام
    text = re.sub(r'\d+', '', text)
    text = text.translate(str.maketrans('', '', string.punctuation))
    text = text.lower()
```

```
    text = text.split()
    text = [stemmer.stem(word) for word in text if word not in stop_words]
    return ' '.join(text)

# دمج العنوان والمحتوى في عمود جديد
df['text'] = df['title'] + " " + df['text']
df = df.drop(['title', 'subject', 'date'], axis=1)

# تنظيف النصوص
df['text'] = df['text'].apply(clean_text)

# عرض عينة
print(df.head())
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Abdulrahman\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
```

	text	label
0	donald trump send embarrass new year’ eve mess...	0
1	drunk brag trump staffer start russian collus ...	0
2	sheriff david clark becom internet joke threat...	0
3	trump obsess even obama’ name code websit imag...	0
4	pope franci call donald trump christma speech ...	0

In [5]:
```
# التحقق من وجود قيم ناقصة
print(df.isnull().sum())
```

```
text     0
label    0
dtype: int64
```

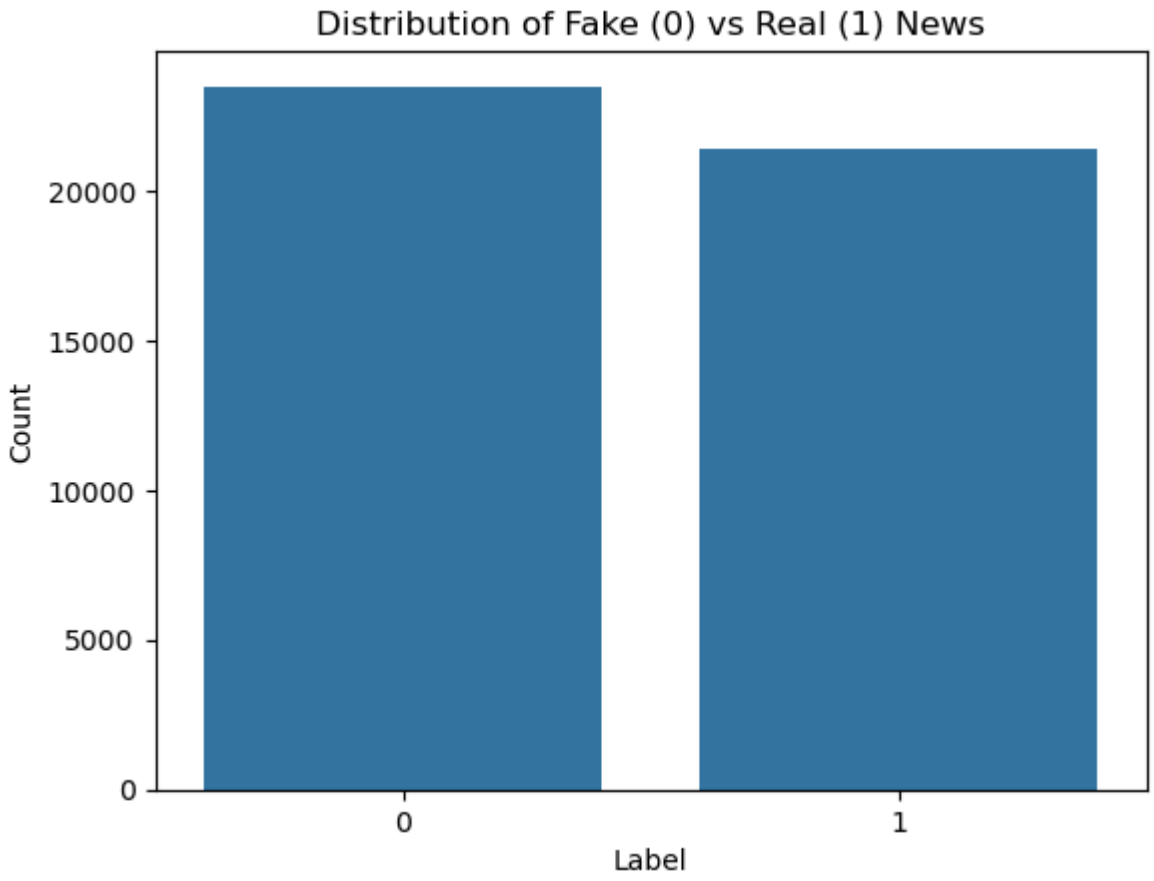
In [7]:
```
# إزالة الصفوف اللي فيها قيم ناقصة
df.dropna(inplace=True)
```

In [9]:
```
from sklearn.model_selection import train_test_split

X = df['text']
y = df['label']

# تقسيم البيانات 70% تدريب / 30% اختبار
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42, stratify=y)
```

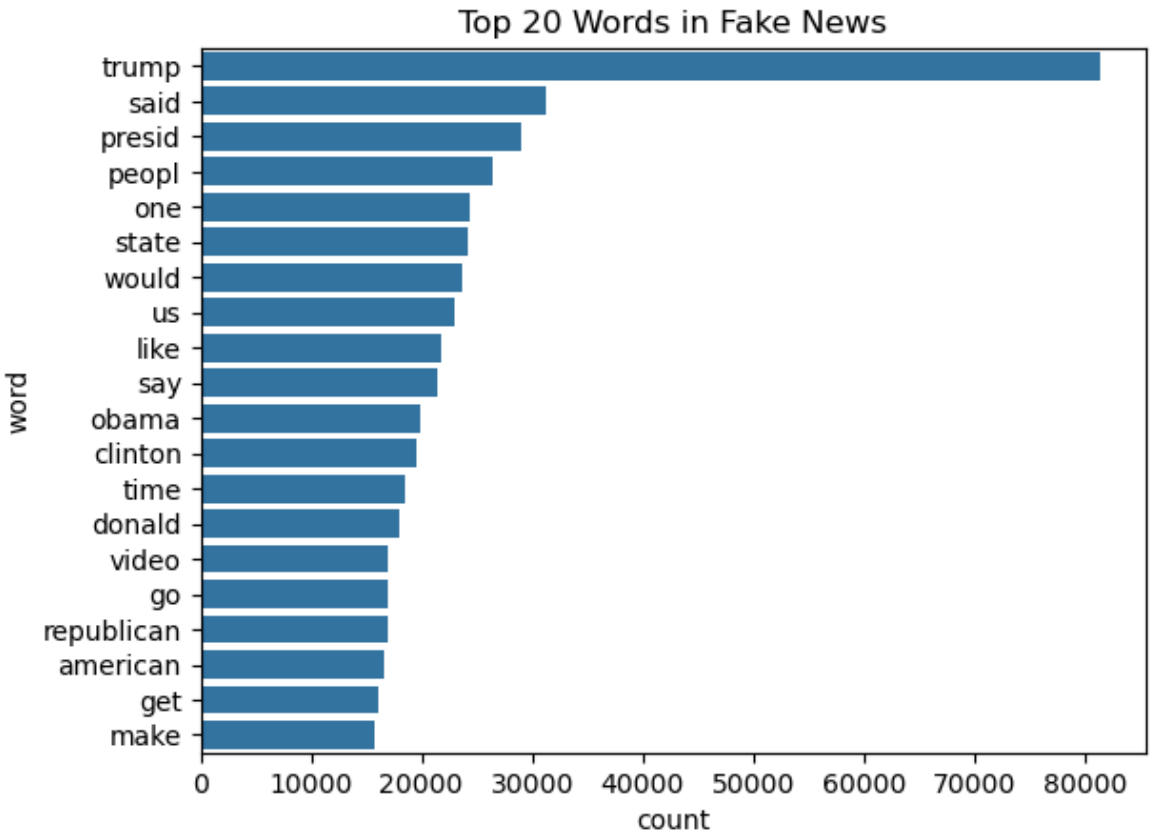
In [11]:
```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer(max_features=5000)

# تدريب المحول على الداتا وتحويل النصوص
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)
```

In [13]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# توزيع الأخبار المزيفة مقابل الحقيقية
sns.countplot(x='label', data=df)
plt.title('Distribution of Fake (0) vs Real (1) News')
plt.xlabel('Label')
```

```
plt.ylabel('Count')
plt.show()
```



In [15]:
```
from collections import Counter

fake_words = ' '.join(df[df['label'] == 0]['text']).split()
common_fake = Counter(fake_words).most_common(20)

fake_df_vis = pd.DataFrame(common_fake, columns=['word', 'count'])
sns.barplot(x='count', y='word', data=fake_df_vis)
plt.title('Top 20 Words in Fake News')
plt.show()
```



In [17]:
```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix, accuracy_sc
import matplotlib.pyplot as plt
import seaborn as sns
```

In [19]:
```
vectorizer = TfidfVectorizer(stop_words='english', max_df=0.7)
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)
```

In [21]:
```
kernels = ['linear', 'poly', 'rbf']
svm_models = {}
results = {}

for kernel in kernels:
    print(f"\nTraining SVM with {kernel} kernel...")
    svm = SVC(kernel=kernel, C=1.0)
    svm.fit(X_train_vec, y_train)
    y_pred = svm.predict(X_test_vec)

    acc = accuracy_score(y_test, y_pred)
    print(f"Accuracy ({kernel}):", acc)
    print(classification_report(y_test, y_pred))

    # Save results
    svm_models[kernel] = svm
    results[kernel] = {
        "accuracy": acc,
        "conf_matrix": confusion_matrix(y_test, y_pred),
        "report": classification_report(y_test, y_pred, output_dict=True)
    }
```

```
Training SVM with linear kernel...
Accuracy (linear): 0.9936896807720861
              precision    recall  f1-score   support

           0       1.00      0.99      0.99      7045
           1       0.99      1.00      0.99      6425

    accuracy                           0.99     13470
   macro avg       0.99      0.99      0.99     13470
weighted avg       0.99      0.99      0.99     13470


Training SVM with poly kernel...
Accuracy (poly): 0.9441722345953972
              precision    recall  f1-score   support

           0       0.92      0.98      0.95      7045
           1       0.97      0.91      0.94      6425

    accuracy                           0.94     13470
   macro avg       0.95      0.94      0.94     13470
weighted avg       0.95      0.94      0.94     13470


Training SVM with rbf kernel...
Accuracy (rbf): 0.9928730512249443
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      7045
           1       0.99      0.99      0.99      6425

    accuracy                           0.99     13470
   macro avg       0.99      0.99      0.99     13470
weighted avg       0.99      0.99      0.99     13470
```

In [30]:
```
!pip install tensorflow
```

```
Requirement already satisfied: tensorflow in c:\users\abdulrahman\anaconda3\lib\s
ite-packages (2.19.0)
Requirement already satisfied: absl-py>=1.0.0 in c:\users\abdulrahman\anaconda3\l
ib\site-packages (from tensorflow) (2.2.2)
Requirement already satisfied: astunparse>=1.6.0 in c:\users\abdulrahman\anaconda
3\lib\site-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in c:\users\abdulrahman\anaco
nda3\lib\site-packages (from tensorflow) (25.2.10)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in c:\users\ab
dulrahman\anaconda3\lib\site-packages (from tensorflow) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in c:\users\abdulrahman\anacon
da3\lib\site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: libclang>=13.0.0 in c:\users\abdulrahman\anaconda3
\lib\site-packages (from tensorflow) (18.1.1)
Requirement already satisfied: opt-einsum>=2.3.2 in c:\users\abdulrahman\anaconda
3\lib\site-packages (from tensorflow) (3.4.0)
Requirement already satisfied: packaging in c:\users\abdulrahman\anaconda3\lib\si
te-packages (from tensorflow) (24.1)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.2
1.4,!=4.21.5,<6.0.0dev,>=3.20.3 in c:\users\abdulrahman\anaconda3\lib\site-packag
es (from tensorflow) (4.25.3)
Requirement already satisfied: requests<3,>=2.21.0 in c:\users\abdulrahman\anacon
da3\lib\site-packages (from tensorflow) (2.32.3)
Requirement already satisfied: setuptools in c:\users\abdulrahman\anaconda3\lib\s
ite-packages (from tensorflow) (75.1.0)
Requirement already satisfied: six>=1.12.0 in c:\users\abdulrahman\anaconda3\lib
\site-packages (from tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\abdulrahman\anaconda3
\lib\site-packages (from tensorflow) (3.1.0)
Requirement already satisfied: typing-extensions>=3.6.6 in c:\users\abdulrahman\a
naconda3\lib\site-packages (from tensorflow) (4.11.0)
Requirement already satisfied: wrapt>=1.11.0 in c:\users\abdulrahman\anaconda3\li
b\site-packages (from tensorflow) (1.14.1)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in c:\users\abdulrahman\anacon
da3\lib\site-packages (from tensorflow) (1.71.0)
Requirement already satisfied: tensorboard~=2.19.0 in c:\users\abdulrahman\anacon
da3\lib\site-packages (from tensorflow) (2.19.0)
Requirement already satisfied: keras>=3.5.0 in c:\users\abdulrahman\anaconda3\lib
\site-packages (from tensorflow) (3.9.2)
Requirement already satisfied: numpy<2.2.0,>=1.26.0 in c:\users\abdulrahman\anaco
nda3\lib\site-packages (from tensorflow) (1.26.4)
Requirement already satisfied: h5py>=3.11.0 in c:\users\abdulrahman\anaconda3\lib
\site-packages (from tensorflow) (3.11.0)
Requirement already satisfied: ml-dtypes<1.0.0,>=0.5.1 in c:\users\abdulrahman\an
aconda3\lib\site-packages (from tensorflow) (0.5.1)
Requirement already satisfied: wheel<1.0,>=0.23.0 in c:\users\abdulrahman\anacond
a3\lib\site-packages (from astunparse>=1.6.0->tensorflow) (0.44.0)
Requirement already satisfied: rich in c:\users\abdulrahman\anaconda3\lib\site-pa
ckages (from keras>=3.5.0->tensorflow) (13.7.1)
Requirement already satisfied: namex in c:\users\abdulrahman\anaconda3\lib\site-p
ackages (from keras>=3.5.0->tensorflow) (0.0.9)
Requirement already satisfied: optree in c:\users\abdulrahman\anaconda3\lib\site-
packages (from keras>=3.5.0->tensorflow) (0.15.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\abdulrahman\a
naconda3\lib\site-packages (from requests<3,>=2.21.0->tensorflow) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\users\abdulrahman\anaconda3\lib
\site-packages (from requests<3,>=2.21.0->tensorflow) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\abdulrahman\anacond
a3\lib\site-packages (from requests<3,>=2.21.0->tensorflow) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\abdulrahman\anacond
```

```
a3\lib\site-packages (from requests<3,>=2.21.0->tensorflow) (2025.1.31)
Requirement already satisfied: markdown>=2.6.8 in c:\users\abdulrahman\anaconda3
\lib\site-packages (from tensorboard~=2.19.0->tensorflow) (3.4.1)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in c:\users
\abdulrahman\anaconda3\lib\site-packages (from tensorboard~=2.19.0->tensorflow)
(0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in c:\users\abdulrahman\anaconda3
\lib\site-packages (from tensorboard~=2.19.0->tensorflow) (3.0.3)
Requirement already satisfied: MarkupSafe>=2.1.1 in c:\users\abdulrahman\anaconda
3\lib\site-packages (from werkzeug>=1.0.1->tensorboard~=2.19.0->tensorflow) (2.1.
3)
Requirement already satisfied: markdown-it-py>=2.2.0 in c:\users\abdulrahman\anac
onda3\lib\site-packages (from rich->keras>=3.5.0->tensorflow) (2.2.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\users\abdulrahman\an
aconda3\lib\site-packages (from rich->keras>=3.5.0->tensorflow) (2.15.1)
Requirement already satisfied: mdurl~=0.1 in c:\users\abdulrahman\anaconda3\lib\s
ite-packages (from markdown-it-py>=2.2.0->rich->keras>=3.5.0->tensorflow) (0.1.0)
```

In [32]:
```
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.metrics import classification_report, confusion_matrix, accuracy_sc
import matplotlib.pyplot as plt
import seaborn as sns
```

In [36]:
```
vectorizer = TfidfVectorizer(stop_words='english', max_df=0.7, max_features=1000

X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)

X_train_dense = X_train_vec.astype('float32').toarray()
X_test_dense = X_test_vec.astype('float32').toarray()
```

In [38]:
```
model = Sequential([
    Dense(128, activation='relu', input_shape=(X_train_dense.shape[1],)),  # Hia
    Dense(64, activation='relu'),  # طبقة إضافية للإبداع
    Dense(1, activation='sigmoid')  # Output binary classification
])
```

```
C:\Users\Abdulrahman\anaconda3\Lib\site-packages\keras\src\layers\core\dense.py:8
7: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. Whe
n using Sequential models, prefer using an `Input(shape)` object as the first lay
er in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

In [40]:
```
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy']
```

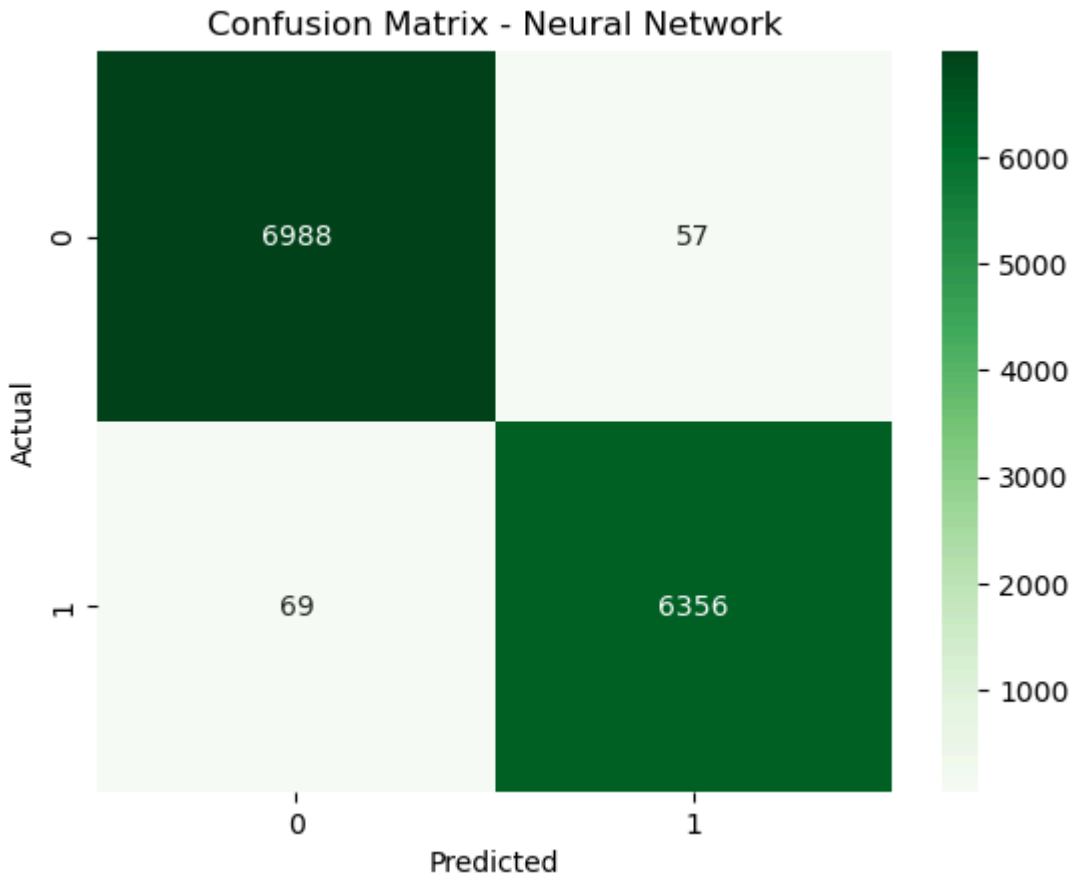
In [42]:
```
early_stop = EarlyStopping(monitor='val_loss', patience=3, restore_best_weights=

history = model.fit(
    X_train_dense, y_train,
    epochs=15,
    batch_size=32,
    validation_split=0.2,
    callbacks=[early_stop]
)
```

```
Epoch 1/15
786/786 ━━━━━━━━━━━━━━━━━━━━ 6s 6ms/step - accuracy: 0.9252 - loss: 0.1931 - val_
accuracy: 0.9870 - val_loss: 0.0380
Epoch 2/15
786/786 ━━━━━━━━━━━━━━━━━━━━ 5s 6ms/step - accuracy: 0.9974 - loss: 0.0107 - val_
accuracy: 0.9895 - val_loss: 0.0346
Epoch 3/15
786/786 ━━━━━━━━━━━━━━━━━━━━ 4s 6ms/step - accuracy: 0.9998 - loss: 0.0013 - val_
accuracy: 0.9905 - val_loss: 0.0357
Epoch 4/15
786/786 ━━━━━━━━━━━━━━━━━━━━ 4s 6ms/step - accuracy: 1.0000 - loss: 2.0852e-04 -
val_accuracy: 0.9909 - val_loss: 0.0382
Epoch 5/15
786/786 ━━━━━━━━━━━━━━━━━━━━ 5s 6ms/step - accuracy: 1.0000 - loss: 7.1856e-05 -
val_accuracy: 0.9909 - val_loss: 0.0400
```

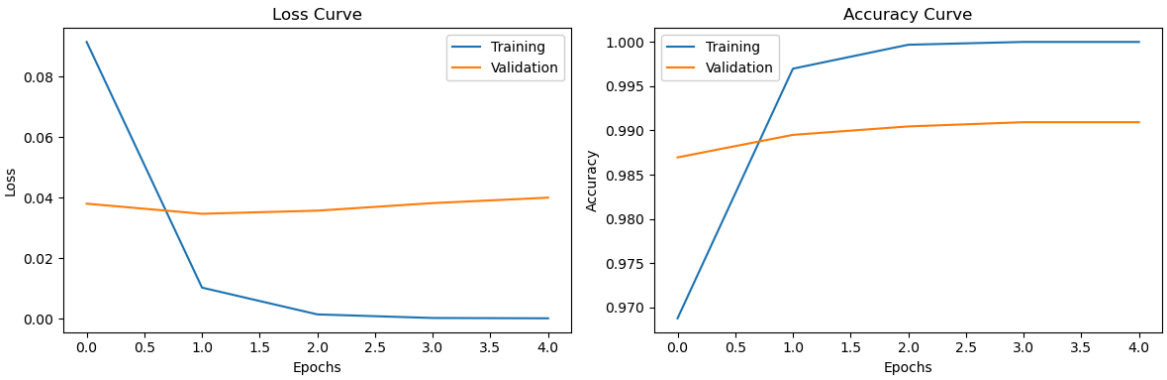
In [44]:
```
y_pred_prob = model.predict(X_test_dense)
y_pred = (y_pred_prob > 0.5).astype("int32")

print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
421/421 ━━━━━━━━━━━━━━━━━━━━ 0s 924us/step
Accuracy: 0.9906458797327394
Classification Report:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      7045
           1       0.99      0.99      0.99      6425

    accuracy                           0.99     13470
   macro avg       0.99      0.99      0.99     13470
weighted avg       0.99      0.99      0.99     13470
```

In [46]:
```
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Greens')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix - Neural Network')
plt.show()
```



In [48]:
```
plt.figure(figsize=(12, 4))

# Loss
plt.subplot(1, 2, 1)
plt.plot(history.history['loss'], label='Training')
plt.plot(history.history['val_loss'], label='Validation')
plt.title('Loss Curve')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

# Accuracy
plt.subplot(1, 2, 2)
plt.plot(history.history['accuracy'], label='Training')
plt.plot(history.history['val_accuracy'], label='Validation')
plt.title('Accuracy Curve')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()

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

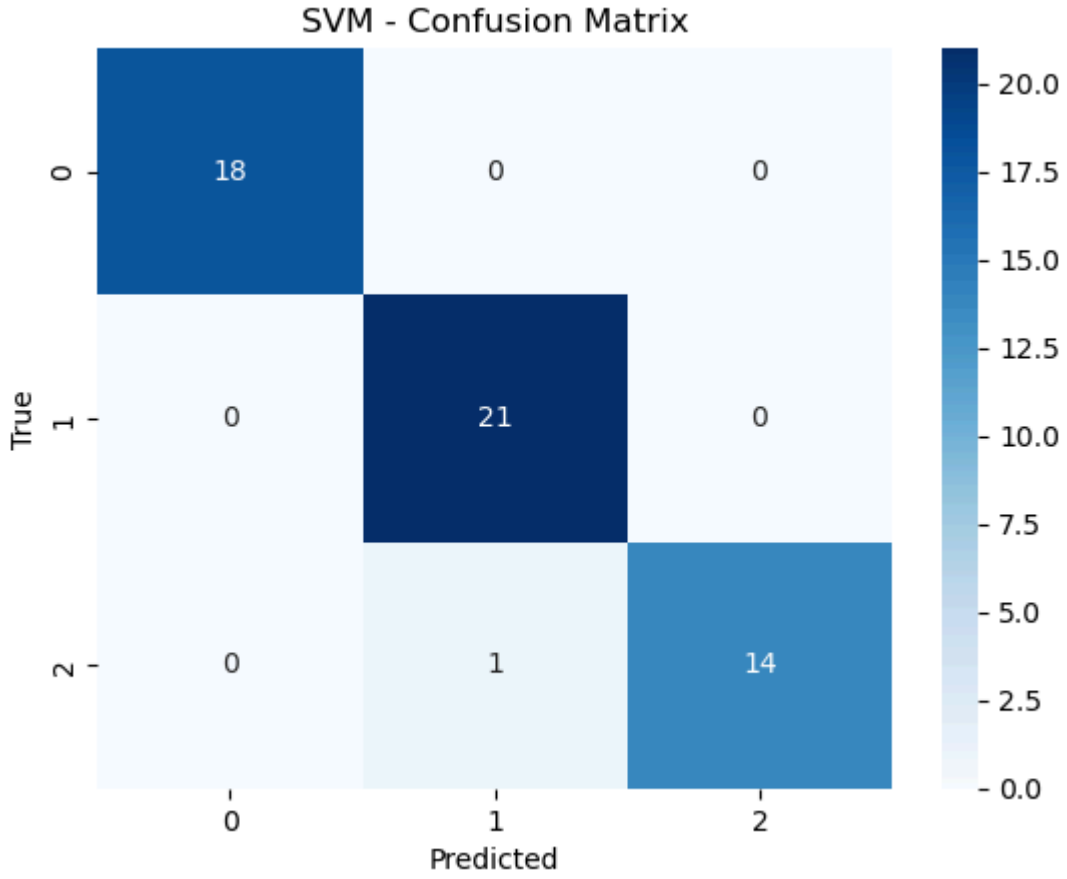


In [54]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_wine
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, label_binarize
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_sc

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.utils import to_categorical
import tensorflow as tf

# === Load Dataset ===
data = load_wine()
X = data.data
y = data.target
classes = np.unique(y)

# === Preprocessing ===
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# === Train SVM ===
svm_model = SVC(kernel='rbf', probability=True)
svm_model.fit(X_train, y_train)
svm_preds = svm_model.predict(X_test)
svm_probs = svm_model.predict_proba(X_test)

# === Train Neural Network ===
y_train_cat = to_categorical(y_train, num_classes=len(classes))

nn_model = Sequential([
    Dense(16, input_shape=(X_train.shape[1],), activation='relu'),
    Dense(len(classes), activation='softmax')
])

nn_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['ac
nn_model.fit(X_train, y_train_cat, epochs=50, batch_size=8, verbose=0)

nn_probs = nn_model.predict(X_test)
nn_preds = np.argmax(nn_probs, axis=1)

# === Evaluation Function ===
```

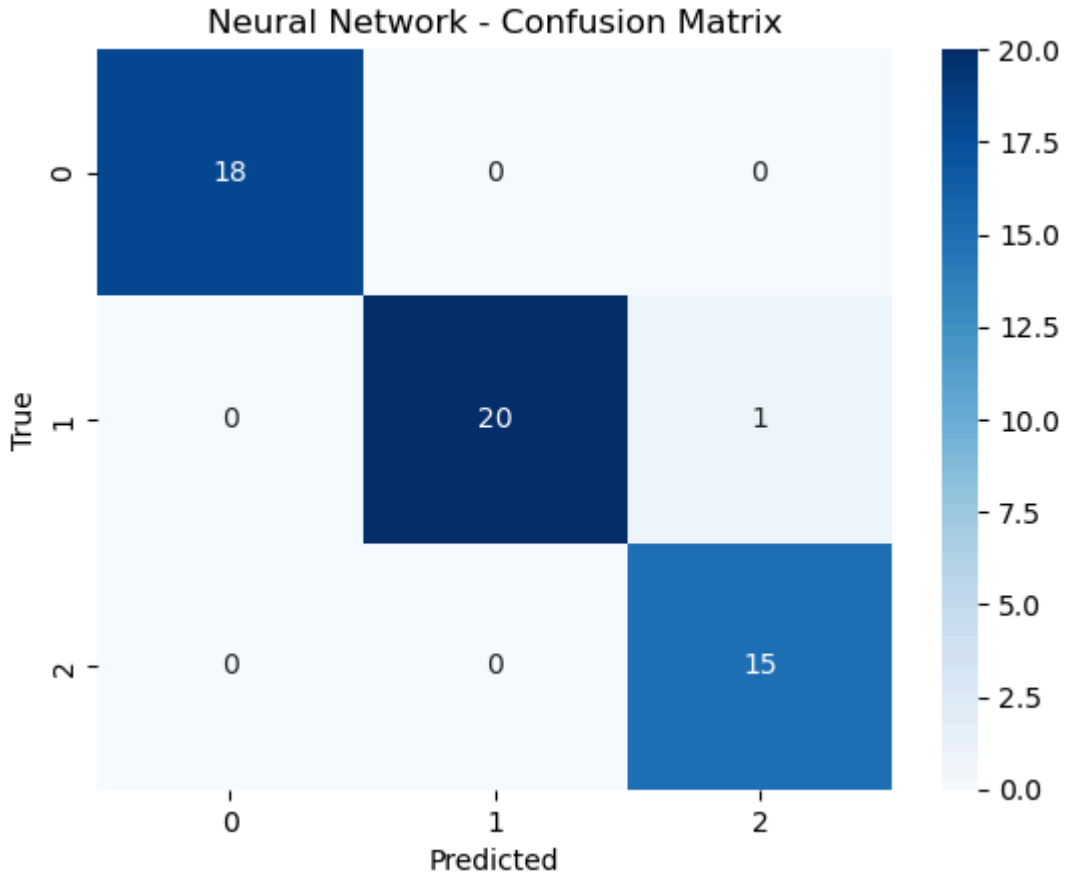
```
def evaluate_model(name, y_true, y_pred):
    print(f"\n--- {name} ---")
    acc = accuracy_score(y_true, y_pred)
    prec = precision_score(y_true, y_pred, average='weighted', zero_division=0)
    rec = recall_score(y_true, y_pred, average='weighted')
    f1 = f1_score(y_true, y_pred, average='weighted')
    print(f"Accuracy:  {acc:.4f}")
    print(f"Precision: {prec:.4f}")
    print(f"Recall:    {rec:.4f}")
    print(f"F1 Score:  {f1:.4f}")
    print("\nClassification Report:\n", classification_report(y_true, y_pred))

    cm = confusion_matrix(y_true, y_pred)
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
    plt.title(f'{name} - Confusion Matrix')
    plt.xlabel('Predicted')
    plt.ylabel('True')
    plt.show()

# === ROC Curve ===
def plot_roc_curve(y_true, probs, model_name, classes):
    y_bin = label_binarize(y_true, classes=classes)
    fpr = dict()
    tpr = dict()
    roc_auc = dict()
    for i in range(len(classes)):
        fpr[i], tpr[i], _ = roc_curve(y_bin[:, i], probs[:, i])
        roc_auc[i] = auc(fpr[i], tpr[i])

    plt.figure()
    for i in range(len(classes)):
        plt.plot(fpr[i], tpr[i], label=f'Class {classes[i]} (AUC = {roc_auc[i]:.
    plt.plot([0, 1], [0, 1], 'k--')
    plt.title(f'{model_name} - ROC Curve')
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.legend()
    plt.show()

# === Run evaluation ===
evaluate_model("SVM", y_test, svm_preds)
evaluate_model("Neural Network", y_test, nn_preds)

plot_roc_curve(y_test, svm_probs, "SVM", classes)
plot_roc_curve(y_test, nn_probs, "Neural Network", classes)

print("\n--- Results Analysis ---")
print("If training accuracy is much higher than test accuracy, the model may be
print("Check precision and recall to detect if any class is being missed.")
print("ROC curves help visualize how well each model separates the classes.")
print("To improve performance, consider hyperparameter tuning, regularization, o
```

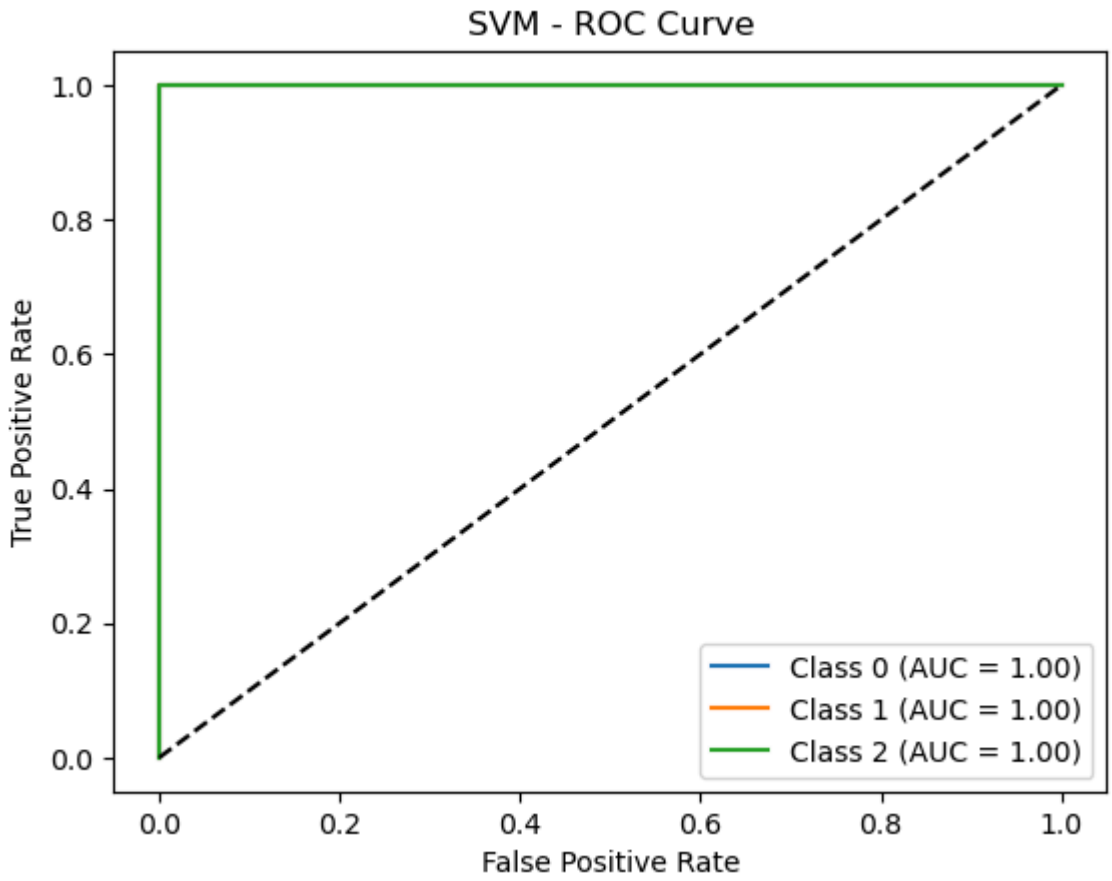
```
C:\Users\Abdulrahman\anaconda3\Lib\site-packages\keras\src\layers\core\dense.py:8
7: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. Whe
n using Sequential models, prefer using an `Input(shape)` object as the first lay
er in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

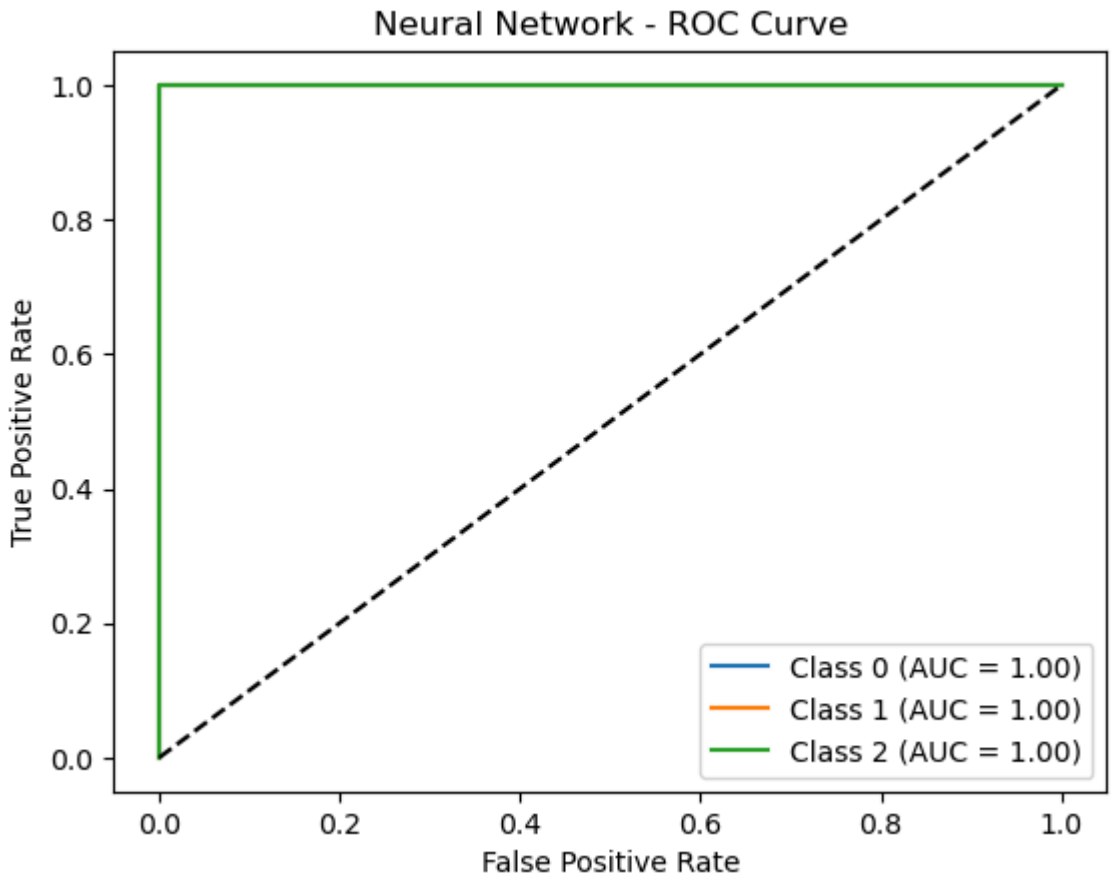
```
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 27ms/step

--- SVM ---
Accuracy:  0.9815
Precision: 0.9823
Recall:    0.9815
F1 Score:  0.9814

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        18
           1       0.95      1.00      0.98        21
           2       1.00      0.93      0.97        15

    accuracy                           0.98        54
   macro avg       0.98      0.98      0.98        54
weighted avg       0.98      0.98      0.98        54
```



```
--- Neural Network ---
Accuracy:  0.9815
Precision: 0.9826
Recall:    0.9815
F1 Score:  0.9816

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        18
           1       1.00      0.95      0.98        21
           2       0.94      1.00      0.97        15

    accuracy                           0.98        54
   macro avg       0.98      0.98      0.98        54
weighted avg       0.98      0.98      0.98        54
```




SVM - ROC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Class 0 (AUC = 1.00)
Class 1 (AUC = 1.00)
Class 2 (AUC = 1.00)


Neural Network - ROC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Class 0 (AUC = 1.00)
Class 1 (AUC = 1.00)
Class 2 (AUC = 1.00)

```
--- Results Analysis ---
If training accuracy is much higher than test accuracy, the model may be overfitt
ing.
Check precision and recall to detect if any class is being missed.
ROC curves help visualize how well each model separates the classes.
To improve performance, consider hyperparameter tuning, regularization, or data a
ugmentation.
```

In [56]:
```
import joblib

# حفظ نموذج SVM في مسار معين
joblib.dump(svm_model, r"C:\Users\Abdulrahman\Desktop\machine\svm_model.pkl")  #
```

Out[56]:
```
['C:\\Users\\Abdulrahman\\Desktop\\machine\\svm_model.pkl']
```

In [58]:
```
# حفظ محول TF-IDF في مسار معين
joblib.dump(vectorizer, r"C:\Users\Abdulrahman\Desktop\machine\vectorizer.pkl")
```

Out[58]:
```
['C:\\Users\\Abdulrahman\\Desktop\\machine\\vectorizer.pkl']
```

In [60]:
```
from tensorflow.keras.models import save_model

# حفظ الشبكة العصبية في مسار معين
save_model(nn_model, r"C:\Users\Abdulrahman\Desktop\machine\nn_model.h5")
```

```
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `ker
as.saving.save_model(model)`. This file format is considered legacy. We recommend
using instead the native Keras format, e.g. `model.save('my_model.keras')` or `ke
ras.saving.save_model(model, 'my_model.keras')`.
```

In [66]:
```
!pip install tensorflow
```

```
Requirement already satisfied: tensorflow in c:\users\abdulrahman\anaconda3\lib\s
ite-packages (2.19.0)
Requirement already satisfied: absl-py>=1.0.0 in c:\users\abdulrahman\anaconda3\l
ib\site-packages (from tensorflow) (2.2.2)
Requirement already satisfied: astunparse>=1.6.0 in c:\users\abdulrahman\anaconda
3\lib\site-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in c:\users\abdulrahman\anaco
nda3\lib\site-packages (from tensorflow) (25.2.10)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in c:\users\ab
dulrahman\anaconda3\lib\site-packages (from tensorflow) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in c:\users\abdulrahman\anacon
da3\lib\site-packages (from tensorflow) (0.2.0)
Requirement already satisfied: libclang>=13.0.0 in c:\users\abdulrahman\anaconda3
\lib\site-packages (from tensorflow) (18.1.1)
Requirement already satisfied: opt-einsum>=2.3.2 in c:\users\abdulrahman\anaconda
3\lib\site-packages (from tensorflow) (3.4.0)
Requirement already satisfied: packaging in c:\users\abdulrahman\anaconda3\lib\si
te-packages (from tensorflow) (24.1)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.2
1.4,!=4.21.5,<6.0.0dev,>=3.20.3 in c:\users\abdulrahman\anaconda3\lib\site-packag
es (from tensorflow) (4.25.3)
Requirement already satisfied: requests<3,>=2.21.0 in c:\users\abdulrahman\anacon
da3\lib\site-packages (from tensorflow) (2.32.3)
Requirement already satisfied: setuptools in c:\users\abdulrahman\anaconda3\lib\s
ite-packages (from tensorflow) (75.1.0)
Requirement already satisfied: six>=1.12.0 in c:\users\abdulrahman\anaconda3\lib
\site-packages (from tensorflow) (1.16.0)
Requirement already satisfied: termcolor>=1.1.0 in c:\users\abdulrahman\anaconda3
\lib\site-packages (from tensorflow) (3.1.0)
Requirement already satisfied: typing-extensions>=3.6.6 in c:\users\abdulrahman\a
naconda3\lib\site-packages (from tensorflow) (4.11.0)
Requirement already satisfied: wrapt>=1.11.0 in c:\users\abdulrahman\anaconda3\li
b\site-packages (from tensorflow) (1.14.1)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in c:\users\abdulrahman\anacon
da3\lib\site-packages (from tensorflow) (1.71.0)
Requirement already satisfied: tensorboard~=2.19.0 in c:\users\abdulrahman\anacon
da3\lib\site-packages (from tensorflow) (2.19.0)
Requirement already satisfied: keras>=3.5.0 in c:\users\abdulrahman\anaconda3\lib
\site-packages (from tensorflow) (3.9.2)
Requirement already satisfied: numpy<2.2.0,>=1.26.0 in c:\users\abdulrahman\anaco
nda3\lib\site-packages (from tensorflow) (1.26.4)
Requirement already satisfied: h5py>=3.11.0 in c:\users\abdulrahman\anaconda3\lib
\site-packages (from tensorflow) (3.11.0)
Requirement already satisfied: ml-dtypes<1.0.0,>=0.5.1 in c:\users\abdulrahman\an
aconda3\lib\site-packages (from tensorflow) (0.5.1)
Requirement already satisfied: wheel<1.0,>=0.23.0 in c:\users\abdulrahman\anacond
a3\lib\site-packages (from astunparse>=1.6.0->tensorflow) (0.44.0)
Requirement already satisfied: rich in c:\users\abdulrahman\anaconda3\lib\site-pa
ckages (from keras>=3.5.0->tensorflow) (13.7.1)
Requirement already satisfied: namex in c:\users\abdulrahman\anaconda3\lib\site-p
ackages (from keras>=3.5.0->tensorflow) (0.0.9)
Requirement already satisfied: optree in c:\users\abdulrahman\anaconda3\lib\site-
packages (from keras>=3.5.0->tensorflow) (0.15.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\abdulrahman\a
naconda3\lib\site-packages (from requests<3,>=2.21.0->tensorflow) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\users\abdulrahman\anaconda3\lib
\site-packages (from requests<3,>=2.21.0->tensorflow) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\abdulrahman\anacond
a3\lib\site-packages (from requests<3,>=2.21.0->tensorflow) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\abdulrahman\anacond
```

```
a3\lib\site-packages (from requests<3,>=2.21.0->tensorflow) (2025.1.31)
Requirement already satisfied: markdown>=2.6.8 in c:\users\abdulrahman\anaconda3
\lib\site-packages (from tensorboard~=2.19.0->tensorflow) (3.4.1)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in c:\users
\abdulrahman\anaconda3\lib\site-packages (from tensorboard~=2.19.0->tensorflow)
(0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in c:\users\abdulrahman\anaconda3
\lib\site-packages (from tensorboard~=2.19.0->tensorflow) (3.0.3)
Requirement already satisfied: MarkupSafe>=2.1.1 in c:\users\abdulrahman\anaconda
3\lib\site-packages (from werkzeug>=1.0.1->tensorboard~=2.19.0->tensorflow) (2.1.
3)
Requirement already satisfied: markdown-it-py>=2.2.0 in c:\users\abdulrahman\anac
onda3\lib\site-packages (from rich->keras>=3.5.0->tensorflow) (2.2.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\users\abdulrahman\an
aconda3\lib\site-packages (from rich->keras>=3.5.0->tensorflow) (2.15.1)
Requirement already satisfied: mdurl~=0.1 in c:\users\abdulrahman\anaconda3\lib\s
ite-packages (from markdown-it-py>=2.2.0->rich->keras>=3.5.0->tensorflow) (0.1.0)
```

In [ ]: