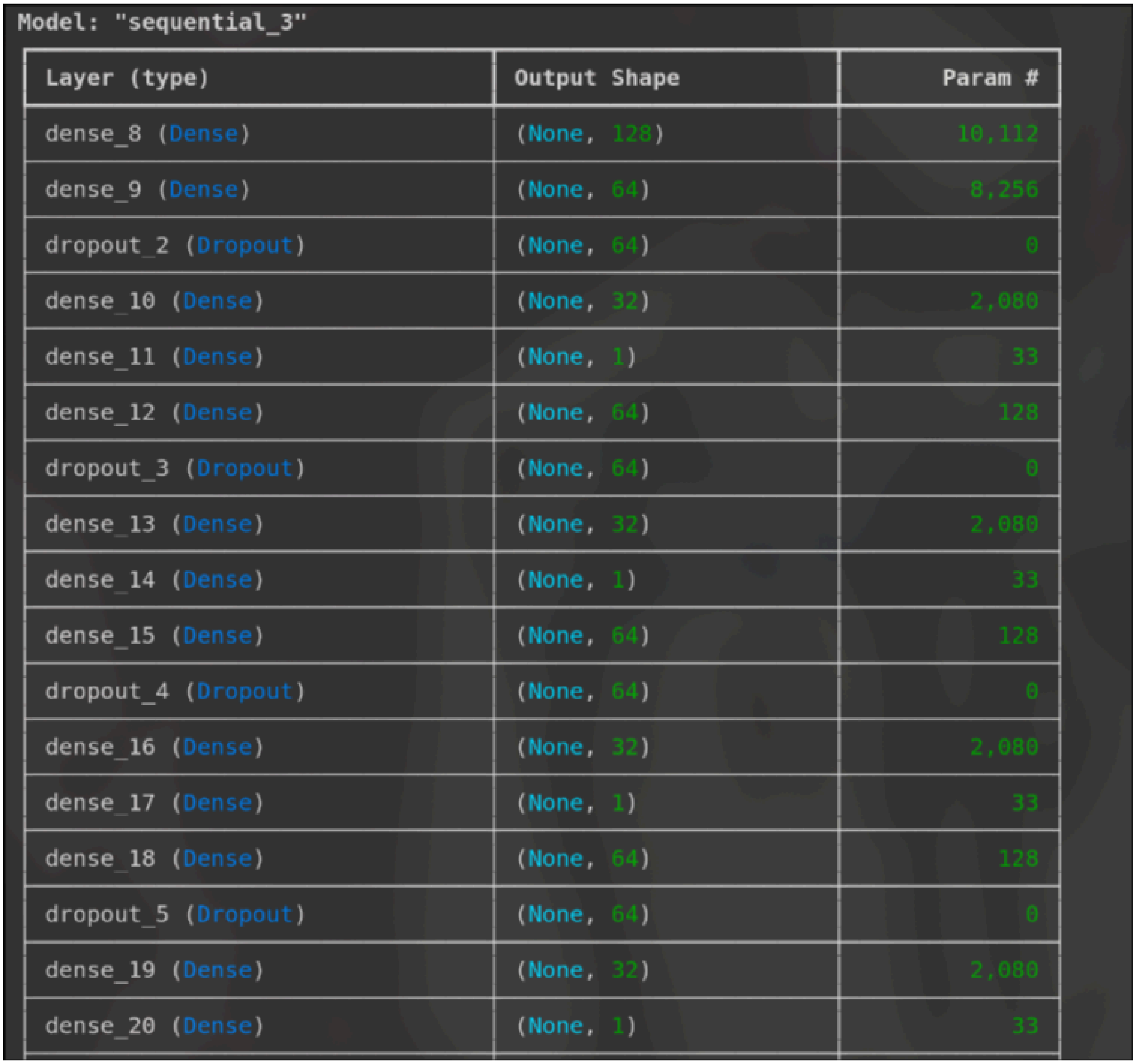
Praneesh R V
CB.SC.U4CYS23036

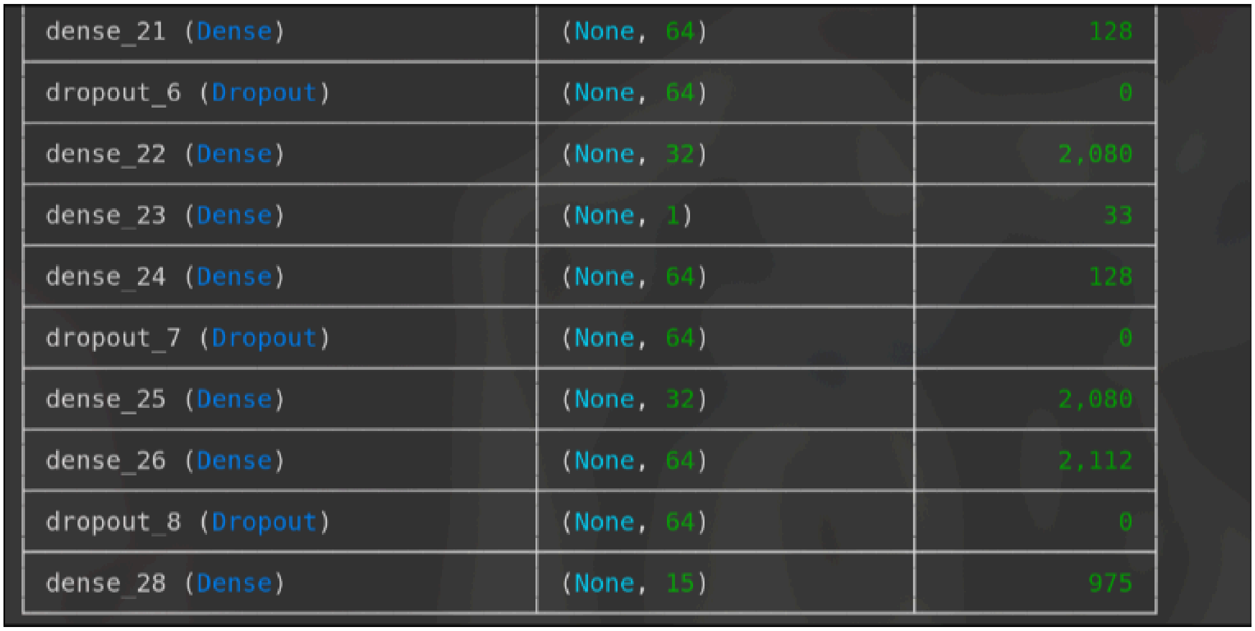
Lab - 3 - Feed Forward Neural Network to classify traffic data

```
import kagglehub
path = kagglehub.dataset_download("chethuhn/network-intrusion-dataset")
print("Path to dataset files:", path)
from google.colab import drive
drive.mount('/content/drive')
import os
kaggle_path =
'/root/.cache/kagglehub/datasets/chethuhn/network-intrusion-dataset/versio
ns/1'
for file in os.listdir(kaggle_path):
 print(file)
 import os
import shutil
kaggle_folder =
'/root/.cache/kagglehub/datasets/chethuhn/network-intrusion-dataset/versio
ns/1'
drive_folder = '/content/drive/MyDrive/CICIDS2017_Parts'
os.makedirs(drive_folder, exist_ok=True)
for file in os.listdir(kaggle_folder):
 if file.endswith('.csv'):
   shutil.copy(os.path.join(kaggle_folder, file),
os.path.join(drive_folder, file))
   import pandas as pd
import glob
csv_files = glob.glob('/content/drive/MyDrive/CICIDS2017_Parts/*.csv')
df = pd.concat((pd.read_csv(f) for f in csv_files), ignore_index=True)
df.head()
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.model_selection import train_test_split
data = df.dropna()
import numpy as np
from sklearn.preprocessing import StandardScaler
X = df.drop(' Label', axis=1)
y = df[' Label']
```

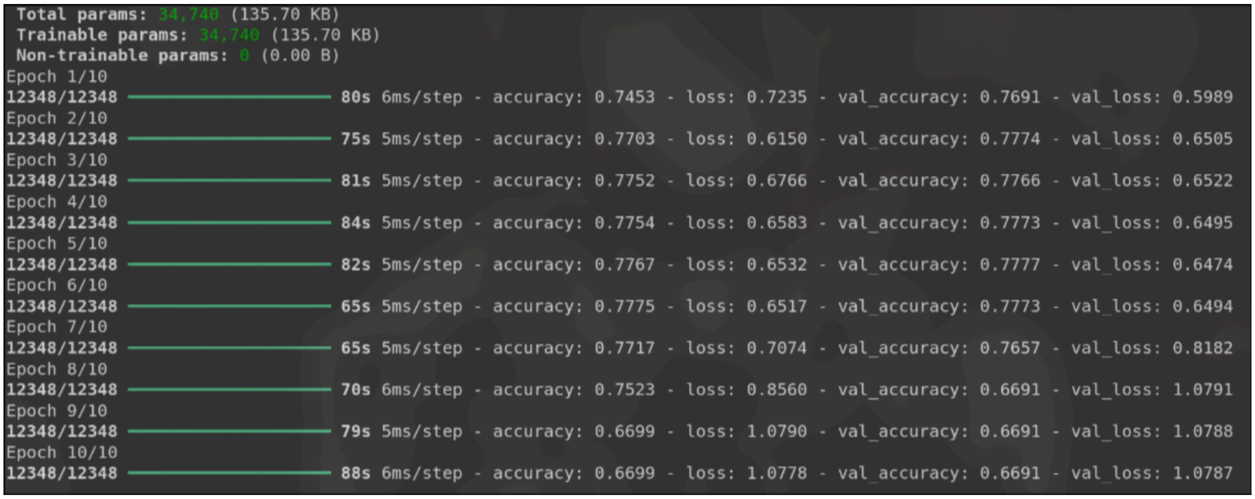
```
X = X[(X >= 0).all(axis=1)]
threshold = 1e8
X = X[(X < threshold).all(axis=1)]
y = y.loc[X.index]
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
X_scaled, y, test_size=0.2, random_state=42
)
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
model = Sequential()
model.add(Dense(128, input_dim=X_train.shape[1], activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.2)) model.add(Dense(32, activation='relu'))

from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.utils import to_categorical
import numpy as np
import pandas as pd

label_encoder = LabelEncoder()
label_encoder.fit(pd.concat([y_train, y_test]).unique())
y_train_encoded = label_encoder.transform(y_train)
y_test_encoded = label_encoder.transform(y_test)

num_classes = len(label_encoder.classes_)
y_train_categorical = to_categorical(y_train_encoded,
num_classes=num_classes)
y_test_categorical = to_categorical(y_test_encoded,
num_classes=num_classes)

model.pop() model.add(Dense(num_classes, activation='softmax'))
model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
model.summary()
history = model.fit(X_train, y_train_categorical, epochs=10,
batch_size=64, validation_split=0.2)
```



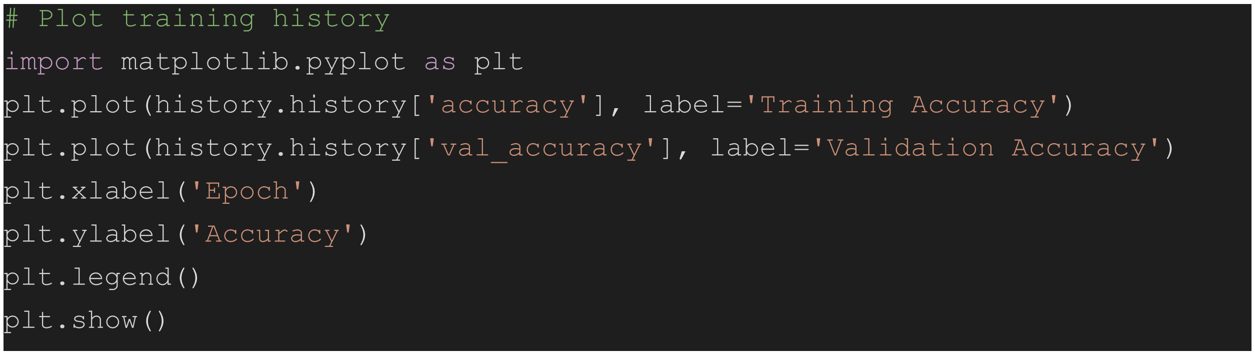
```
Model: "sequential_3"
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━┓
┃ Layer (type)                    ┃ Output Shape           ┃       Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━┩
│ dense_8 (Dense)                 │ (None, 128)            │        10,112 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_9 (Dense)                 │ (None, 64)             │         8,256 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dropout_2 (Dropout)             │ (None, 64)             │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_10 (Dense)                │ (None, 32)             │         2,080 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_11 (Dense)                │ (None, 1)              │            33 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_12 (Dense)                │ (None, 64)             │           128 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dropout_3 (Dropout)             │ (None, 64)             │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_13 (Dense)                │ (None, 32)             │         2,080 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_14 (Dense)                │ (None, 1)              │            33 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_15 (Dense)                │ (None, 64)             │           128 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dropout_4 (Dropout)             │ (None, 64)             │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_16 (Dense)                │ (None, 32)             │         2,080 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_17 (Dense)                │ (None, 1)              │            33 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_18 (Dense)                │ (None, 64)             │           128 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dropout_5 (Dropout)             │ (None, 64)             │             0 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_19 (Dense)                │ (None, 32)             │         2,080 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
│ dense_20 (Dense)                │ (None, 1)              │            33 │
├─────────────────────────────────┼────────────────────────┼───────────────┤
```



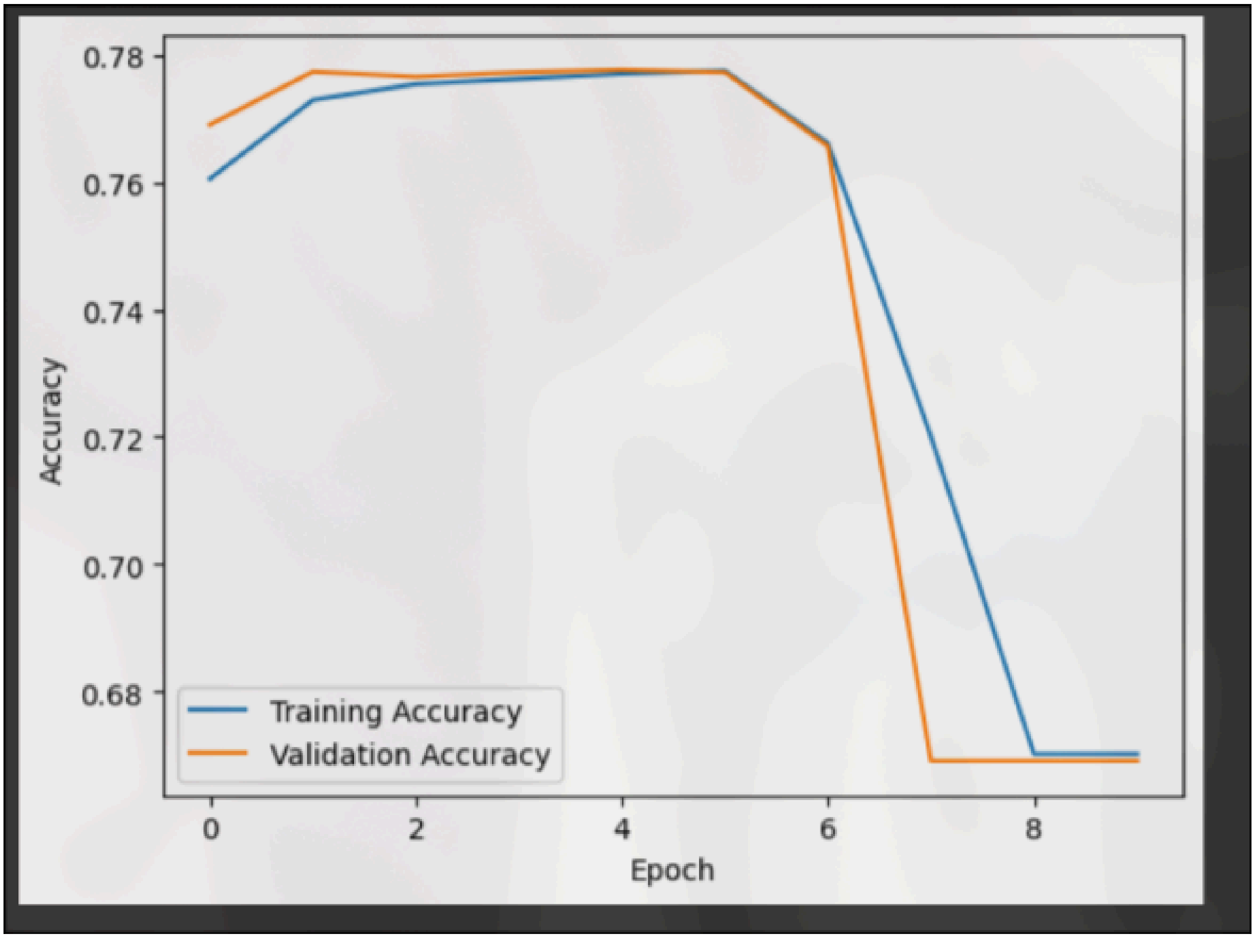
```
│ dense_21 (Dense)          │ (None, 64)            │           128 │
├───────────────────────────┼───────────────────────┼───────────────┤
│ dropout_6 (Dropout)       │ (None, 64)            │             0 │
├───────────────────────────┼───────────────────────┼───────────────┤
│ dense_22 (Dense)          │ (None, 32)            │         2,080 │
├───────────────────────────┼───────────────────────┼───────────────┤
│ dense_23 (Dense)          │ (None, 1)             │            33 │
├───────────────────────────┼───────────────────────┼───────────────┤
│ dense_24 (Dense)          │ (None, 64)            │           128 │
├───────────────────────────┼───────────────────────┼───────────────┤
│ dropout_7 (Dropout)       │ (None, 64)            │             0 │
├───────────────────────────┼───────────────────────┼───────────────┤
│ dense_25 (Dense)          │ (None, 32)            │         2,080 │
├───────────────────────────┼───────────────────────┼───────────────┤
│ dense_26 (Dense)          │ (None, 64)            │         2,112 │
├───────────────────────────┼───────────────────────┼───────────────┤
│ dropout_8 (Dropout)       │ (None, 64)            │             0 │
├───────────────────────────┼───────────────────────┼───────────────┤
│ dense_28 (Dense)          │ (None, 15)            │           975 │
└───────────────────────────┴───────────────────────┴───────────────┘
```

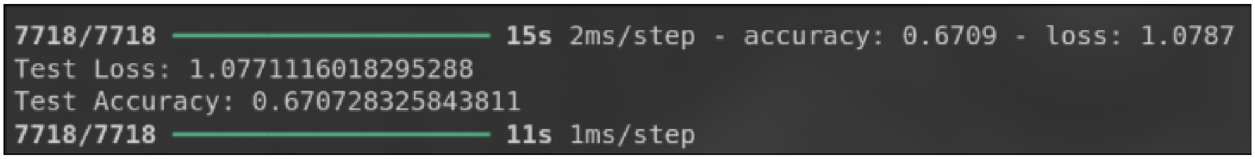


```
 Total params: 34,740 (135.70 KB)
 Trainable params: 34,740 (135.70 KB)
 Non-trainable params: 0 (0.00 B)
Epoch 1/10
12348/12348 ━━━━━━━━━━━━━━━━━━━━ 80s 6ms/step - accuracy: 0.7453 - loss: 0.7235 - val_accuracy: 0.7691 - val_loss: 0.5989
Epoch 2/10
12348/12348 ━━━━━━━━━━━━━━━━━━━━ 75s 5ms/step - accuracy: 0.7703 - loss: 0.6150 - val_accuracy: 0.7774 - val_loss: 0.6505
Epoch 3/10
12348/12348 ━━━━━━━━━━━━━━━━━━━━ 81s 5ms/step - accuracy: 0.7752 - loss: 0.6766 - val_accuracy: 0.7766 - val_loss: 0.6522
Epoch 4/10
12348/12348 ━━━━━━━━━━━━━━━━━━━━ 84s 5ms/step - accuracy: 0.7754 - loss: 0.6583 - val_accuracy: 0.7773 - val_loss: 0.6495
Epoch 5/10
12348/12348 ━━━━━━━━━━━━━━━━━━━━ 82s 5ms/step - accuracy: 0.7767 - loss: 0.6532 - val_accuracy: 0.7777 - val_loss: 0.6474
Epoch 6/10
12348/12348 ━━━━━━━━━━━━━━━━━━━━ 65s 5ms/step - accuracy: 0.7775 - loss: 0.6517 - val_accuracy: 0.7773 - val_loss: 0.6494
Epoch 7/10
12348/12348 ━━━━━━━━━━━━━━━━━━━━ 65s 5ms/step - accuracy: 0.7717 - loss: 0.7074 - val_accuracy: 0.7657 - val_loss: 0.8182
Epoch 8/10
12348/12348 ━━━━━━━━━━━━━━━━━━━━ 70s 6ms/step - accuracy: 0.7523 - loss: 0.8560 - val_accuracy: 0.6691 - val_loss: 1.0791
Epoch 9/10
12348/12348 ━━━━━━━━━━━━━━━━━━━━ 79s 5ms/step - accuracy: 0.6699 - loss: 1.0790 - val_accuracy: 0.6691 - val_loss: 1.0788
Epoch 10/10
12348/12348 ━━━━━━━━━━━━━━━━━━━━ 88s 6ms/step - accuracy: 0.6699 - loss: 1.0778 - val_accuracy: 0.6691 - val_loss: 1.0787
```

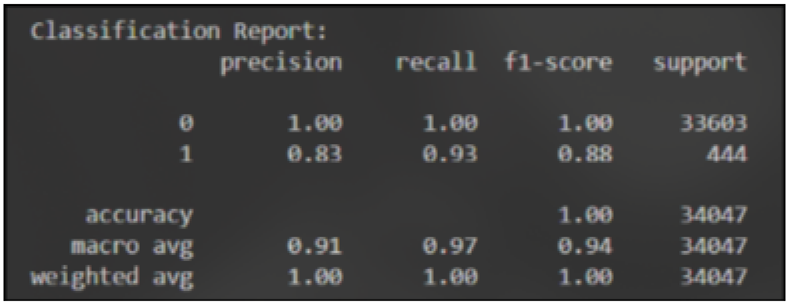


```python
# Plot training history
import matplotlib.pyplot as plt
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
```





```
7718/7718 ━━━━━━━━━━━━━━━━━━━━ 15s 2ms/step - accuracy: 0.6709 - loss: 1.0787
Test Loss: 1.0771116018295288
Test Accuracy: 0.670728325843811
7718/7718 ━━━━━━━━━━━━━━━━━━━━ 11s 1ms/step
```



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

           0       1.00      1.00      1.00     33603
           1       0.83      0.93      0.88       444

    accuracy                           1.00     34047
   macro avg       0.91      0.97      0.94     34047
weighted avg       1.00      1.00      1.00     34047
```