Commands | + Code | + Text | Run all

RAM Disk

Files

- ..
- drive
  - MyDrive
    - AI ML Notes
    - AI Research Papers
    - Colab Notebooks
    - DataSet
    - ICTAK_Project_Data
    - ICT_Project_IntelligentGOSe...
    - Misc
    - Resume
    - exit_exam
      - Reviews.csv
      - Reviews.csv.zip
    - Prototype.pdf
    - Python Quiz Web App
    - Python Quiz Web App for C...
    - Python Quiz Web App for C...
    - Rohit S_Resume (1).pdf

Disk 67.65 GB available

[26] ✓ 0s

```
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout
from tensorflow.keras.utils import to_categorical
from sklearn.preprocessing import LabelEncoder
```

[28] ✓ 0s

```
# Parameters (limitted for speedy processing)
max_words = 5000
max_len = 100
embedding_dim = 64
lstm_units = 64
```

[29] ✓ 57s

```
# Tokenize and pad sequences
tokenizer = Tokenizer(num_words=max_words)
tokenizer.fit_on_texts(df['Clean_Text'])
X_seq = tokenizer.texts_to_sequences(df['Clean_Text'])
X_pad = pad_sequences(X_seq, maxlen=max_len)
```

[30] ✓ 0s

```
# Encode labels
le = LabelEncoder()
y_enc = le.fit_transform(df['Sentiment'])
```

[31] ✓ 0s

```
# Train-test split
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X_pad, y_enc, test_size=0.2, random_state=42)
```

[35] ✓ 0s

```
# Build model
model = Sequential([
    Embedding(input_dim=max_words, output_dim=embedding_dim),
    LSTM(lstm_units, dropout=0.2, recurrent_dropout=0.2),
    Dense(1, activation='sigmoid')
])

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

Variables | Terminal

✓ 3:31 PM Python 3

AIML_Exit_Exam.ipynb

File Edit View Insert Runtime Tools Help

Share

Commands + Code + Text Run all

RAM Disk

Files

- ..
- drive
  - MyDrive
    - AI ML Notes
    - AI Research Papers
    - Colab Notebooks
    - DataSet
    - ICTAK_Project_Data
    - ICT_Project_IntelligentGOSe...
    - Misc
    - Resume
    - exit_exam
      - Reviews.csv
      - Reviews.csv.zip
    - Prototype.pdf
    - Python Quiz Web App
    - Python Quiz Web App for C...
    - Python Quiz Web App for C...
    - Rohit S_Resume (1).pdf

Disk 67.65 GB available

[35] ✓ 0s

```
    Dense(1, activation='sigmoid')
])

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

[37] ✓ 24m

```
# Train model (used fewer epochs and larger batch for speed)
history = model.fit(X_train, y_train, epochs=2, batch_size=512, validation_data=(X_test, y_test))
```

```
Epoch 1/2
822/822 ━━━━━━━━━━━━━━━━━━━━ 752s 906ms/step - accuracy: 0.8917 - loss: 0.2828 - val_accuracy: 0.9319 - val_loss: 0.1748
Epoch 2/2
822/822 ━━━━━━━━━━━━━━━━━━━━ 733s 892ms/step - accuracy: 0.9336 - loss: 0.1712 - val_accuracy: 0.9368 - val_loss: 0.1634
```

[38] ✓ 0s

```
# Model Summary
model.summary()
```

Model: "sequential_1"

Layer (type)	Output Shape	Param #
embedding_1 (Embedding)	(None, 100, 64)	320,000
lstm_1 (LSTM)	(None, 64)	33,024
dense_1 (Dense)	(None, 1)	65

Total params: 1,059,269 (4.04 MB)
Trainable params: 353,089 (1.35 MB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 706,180 (2.69 MB)

LSTMs are preferred over RNNs because LSTMs can effectively learn long-range dependencies in sequences. Simple RNNs svanishing gradient problem makes it difficult for the model to learn relationships between distant words in a sentence. LSTM networks, through their gated mechanisms, mitigate this. Resulting in better performance on tasks where understanding context and word order is important, such as sentiment analysis.

Variables Terminal

3:31PM Python 3

AFG - BAN Game score

Search

ENG IN 15:32 15-10-2025