ICP7 REPORT

+ Code + Text Connect ✦ Gemini

```
!pip install tensorflow==2.12.0
!pip install keras==2.12.0
!pip install keras.utils
```

```
Requirement already satisfied: tensorflow==2.12.0 in /usr/local/lib/python3.10/dist-packages (2.12.0)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (1.6.3)
Requirement already satisfied: flatbuffers>=2.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (24.3.25)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (0.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (0.2.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (1.64.1)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (3.11.0)
Requirement already satisfied: jax>=0.3.15 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (0.4.30)
Requirement already satisfied: keras<2.13,>=2.12.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (2.12.0)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (18.1.1)
Requirement already satisfied: numpy<1.24,>=1.22 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (1.23.5)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (3.4.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (24.1)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (75.1.0)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (1.16.0)
Requirement already satisfied: tensorboard<2.13,>=2.12 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (2.12.3)
Requirement already satisfied: tensorflow-estimator<2.13,>=2.12.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (2.12.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (2.5.0)
```

+ Code + Text Connect ✦ Gemini

```
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.10/dist-packages (from tensorflow==
Requirement already satisfied: wrapt<1.15,>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (0.37.1)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0->tensorflow==2.12.0) (0.44.0)
Requirement already satisfied: jaxlib<=0.4.30,>=0.4.27 in /usr/local/lib/python3.10/dist-packages (from jax>=0.3.15->tensorflow==2.12.0) (0.4.30)
Requirement already satisfied: ml-dtypes>=0.2.0 in /usr/local/lib/python3.10/dist-packages (from jax>=0.3.15->tensorflow==2.12.0) (0.4.1)
Requirement already satisfied: scipy>=1.9 in /usr/local/lib/python3.10/dist-packages (from jax>=0.3.15->tensorflow==2.12.0) (1.13.1)
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.13,>=2.12->tensorflow==2.12.0) (2.
Requirement already satisfied: google-auth-oauthlib<1.1,>=0.5 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.13,>=2.12->tensorflow==2.
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.13,>=2.12->tensorflow==2.12.0) (3.7)
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.13,>=2.12->tensorflow==2.12.0) (2.32
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.13,>=2.12->tensorf
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.13,>=2.12->tensorflow==2.12.0) (3.0.6)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.13,>=2.12-
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.13,>=2.12->
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.13,>=2.12->tensorfl
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from google-auth-oauthlib<1.1,>=0.5->tensorboard<2
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorboard<2.13,>=2.12-
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorboard<2.13,>=2.12->tensorflow=
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorboard<2.13,>=2.12->tenso
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2.21.0->tensorboard<2.13,>=2.12->tenso
Requirement already satisfied: MarkupSafe>=2.1.1 in /usr/local/lib/python3.10/dist-packages (from werkzeug>=1.0.1->tensorboard<2.13,>=2.12->tensorflow=
Requirement already satisfied: pyasn1<0.7.0,>=0.4.6 in /usr/local/lib/python3.10/dist-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->ten
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<1.1,>=0
Requirement already satisfied: keras==2.12.0 in /usr/local/lib/python3.10/dist-packages (2.12.0)
Requirement already satisfied: keras.utils in /usr/local/lib/python3.10/dist-packages (1.0.13)
```

+ Code + Text Connect

```
[ ] # Mounting Google Drive
    from google.colab import drive
    drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
[ ] import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
    from keras.preprocessing.text import Tokenizer
    from tensorflow.keras.preprocessing.sequence import pad_sequences
    from keras.models import Sequential
    from keras.layers import Dense, Embedding, LSTM, SpatialDropout1D
    from matplotlib import pyplot
    from sklearn.model_selection import train_test_split
    from keras.utils.np_utils import to_categorical
    import re

    from sklearn.preprocessing import LabelEncoder

    data = pd.read_csv('/content/drive/My Drive/Sentiment.csv')
    # Keeping only the neccessary columns
    data = data[['text','sentiment']]
```

+ Code + Text Connect

```
[ ] data['text'] = data['text'].apply(lambda x: x.lower())
    data['text'] = data['text'].apply((lambda x: re.sub('[^a-zA-z0-9\s]', '', x)))

    for idx, row in data.iterrows():
        row[0] = row[0].replace('rt', ' ')

    max_fatures = 2000
    tokenizer = Tokenizer(num_words=max_fatures, split=' ')
    tokenizer.fit_on_texts(data['text'].values)
    X = tokenizer.texts_to_sequences(data['text'].values)

    X = pad_sequences(X)

    embed_dim = 128
    lstm_out = 196
    def createmodel():
        model = Sequential()
        model.add(Embedding(max_fatures, embed_dim,input_length = X.shape[1]))
        model.add(LSTM(lstm_out, dropout=0.2, recurrent_dropout=0.2))
        model.add(Dense(3,activation='softmax'))
        model.compile(loss = 'categorical_crossentropy', optimizer='adam',metrics = ['accuracy'])
        return model
    # print(model.summary())
```

+ Code + Text Connect ✦ Gemini

```
[ ] labelencoder = LabelEncoder()
    integer_encoded = labelencoder.fit_transform(data['sentiment'])
    y = to_categorical(integer_encoded)
    X_train, X_test, Y_train, Y_test = train_test_split(X,y, test_size = 0.33, random_state = 42)

    batch_size = 32
    model = createmodel()
    model.fit(X_train, Y_train, epochs = 1, batch_size=batch_size, verbose = 2)
    score,acc = model.evaluate(X_test,Y_test,verbose=2,batch_size=batch_size)
    print(score)
    print(acc)
    print(model.metrics_names)
```

```
<ipython-input-5-79347c4597c4>:21: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will
  row[0] = row[0].replace('rt', ' ')
<ipython-input-5-79347c4597c4>:21: FutureWarning: Series.__setitem__ treating keys as positions is deprecated. In a future version, integer keys will
  row[0] = row[0].replace('rt', ' ')
291/291 - 50s - loss: 0.8268 - accuracy: 0.6403 - 50s/epoch - 171ms/step
144/144 - 3s - loss: 0.7453 - accuracy: 0.6752 - 3s/epoch - 20ms/step
0.745284914970398
0.6751856803894043
['loss', 'accuracy']
```

+ Code + Text Connect ✦ Gemi

```
[ ] model.save('sentiment_Analy.h5')
```

```
[ ] from keras.models import load_model
    import numpy as np

    loaded_model = load_model('sentiment_Analy.h5')

    new_text = ["A lot of good things are happening. We are respected again throughout the world, and that's a great thing.@realDonaldTrump"]
    new_text = tokenizer.texts_to_sequences(new_text)
    new_text = pad_sequences(new_text, maxlen=X.shape[1], dtype='int32', value=0)
    sentiment_prob = loaded_model.predict(new_text, batch_size=1, verbose=2)[0]

    sentiment_classes = ['Positive', 'Neutral', 'Negative']
    sentiment_p = sentiment_classes[np.argmax(sentiment_prob)]

    print("Predicted sentiment: ", sentiment_p)
    print("Predicted probabilities: ", sentiment_prob)
```

```
1/1 - 1s - 809ms/epoch - 809ms/step
Predicted sentiment:  Positive
Predicted probabilities:  [0.44116956 0.16455497 0.39427555]
```

```
+ Code + Text                                                    Connect

[ ] from keras.wrappers.scikit_learn import KerasClassifier
    from sklearn.model_selection import GridSearchCV
    from keras.layers import LSTM

    # Function to create the model, as it's required by KerasClassifier
    def create_model(lstm_out=196, dropout=0.2):
        model = Sequential()
        model.add(Embedding(max_fatures, embed_dim, input_length=X.shape[1]))
        model.add(LSTM(lstm_out, dropout=dropout, recurrent_dropout=dropout))
        model.add(Dense(3, activation='softmax'))
        model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
        return model

    # Create the KerasClassifier
    model = KerasClassifier(build_fn=create_model, verbose=0)

    batch_s = [10, 20, 40]
    epochs_s = [1, 2, 3]

    # Define the grid of parameters to search
    param_grid = dict(batch_size=batch_s, epochs=epoch_s)
```



```
+ Code + Text                                    Connect  ✦ Gemini

[ ] # Create GridSearchCV
    grid = GridSearchCV(estimator=model, param_grid=param_grid, n_jobs=-1, cv=3)
    grid_result = grid.fit(X_train, Y_train)

    # Summarize results
    print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))

<ipython-input-13-3e27ad9c23bd>:15: DeprecationWarning: KerasClassifier is deprecated, use Sci-Keras (https://github.com/adriangb/scikeras) instead. S
  model = KerasClassifier(build_fn=create_model, verbose=0)
Best: 0.676638 using {'batch_size': 40, 'epochs': 2}
```

My Github Repository Link:-
https://github.com/PraveenDondapati/bda.git