# Generative-Text-Model

Create a model (GPT or LSTM-based) that generates coherent paragraphs based on user-provided prompts.

LSTM-based Text Generator (Custom Training)

□ Jupyter Notebook Outline (text_generation_lstm.ipynb)

```
# Step 1: Import libraries
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Embedding
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.preprocessing.sequence import pad_sequences


# Step 2: Sample corpus
data = """Machine learning is transforming industries. AI-powered systems automate complex tasks..."""


# Step 3: Tokenization
tokenizer = Tokenizer()
tokenizer.fit_on_texts([data])
total_words = len(tokenizer.word_index) + 1


# Generate sequences
input_sequences = []
for line in data.split("."):
  tokens = tokenizer.texts_to_sequences([line])[0]
  for i in range(1, len(tokens)):
      n_gram = tokens[:i+1]
      input_sequences.append(n_gram)


# Pad and split
max_seq_len = max([len(x) for x in input_sequences])
input_sequences = np.array(pad_sequences(input_sequences, maxlen=max_seq_len, padding='pre'))
X, y = input_sequences[:, :-1], input_sequences[:, -1]
y = to_categorical(y, num_classes=total_words)


# Step 4: LSTM Model
model = Sequential()
model.add(Embedding(total_words, 10, input_length=max_seq_len-1))
model.add(LSTM(100))
model.add(Dense(total_words, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(X, y, epochs=100, verbose=1)


# Step 5: Generate text
seed_text = "Machine learning"
```

```
next_words = 20


for _ in range(next_words):
  token_list = tokenizer.texts_to_sequences([seed_text])[0]
  token_list = pad_sequences([token_list], maxlen=max_seq_len-1, padding='pre')
  predicted = model.predict(token_list, verbose=0)
  predicted_word = tokenizer.index_word[np.argmax(predicted)]
  seed_text += " " + predicted_word


print("Generated Text:\n", seed_text)
```

□ Project Structure

```
text_generation_model/
□ □ □text_generation_gpt2.ipynb     # GPT-based notebook
□ □ □text_generation_lstm.ipynb     # LSTM-based notebook
□ □ □sample_output.txt
□ □ □README.md
```