Text Generation with Markov Chains

Project Goals: The main goal of this project is to build a text generator that can produce sentences or paragraphs of text that resemble the style and content of the input dataset. This involves creating a model that learns the statistical relationships between words in the training data and then uses these relationships to generate new text.

Skills to Develop:

1. **Text Preprocessing:** Clean and preprocess the text data, including tasks like removing punctuation, lowercasing, and tokenizing the text into words.
2. **Building Markov Chains:** Construct a Markov chain by calculating the transition probabilities between words in the training data.
3. **Text Generation:** Use the Markov chain to generate new text by selecting the next word based on the probabilities of occurrence in the training data.
4. **Model Evaluation:** Develop a method to evaluate the generated text in terms of coherence and similarity to the original text.

Data Source: For this project, you can use a variety of text sources, such as books, articles, or speeches. Project Gutenberg offers a wide collection of free eBooks that you can use. Choose a dataset that aligns with your interests and provides a diverse range of language usage.

Steps to Follow:

1. **Data Collection:** Obtain the text data from a suitable source. You might need to preprocess the data to remove any irrelevant information, such as headers or footers.
2. **Text Preprocessing:** Clean the text data by removing punctuation, converting text to lowercase, and tokenizing it into words. You may also want to remove stopwords (common words like "the," "and," etc.) to improve the quality of the Markov chain.
3. **Building Markov Chains:** Create a dictionary where the keys are words from the dataset, and the values are lists of words that often follow the key word in the dataset. The length of the list can be determined by the order of the Markov chain (first order, second order, etc.).
4. **Text Generation:** To generate text, start with a seed word and use the Markov chain to predict the next word based on the probability distribution associated with that seed word. Repeat this process to generate a sequence of words.
5. **Evaluation:** Evaluating generated text can be subjective. You could evaluate it based on coherence, grammaticality, and whether it captures the style of the input dataset. Human judgment is valuable for this step.

Example Evaluation Metric: One way to evaluate the generated text is by using perplexity. Perplexity measures how well a probability distribution predicts a sample. In the context of text generation, a lower perplexity indicates better performance. You could calculate the perplexity of the generated text compared to the original text data.

Note: This project is a simplified approach to text generation and might not produce highly sophisticated results. However, it's an excellent opportunity to gain hands-on experience with language modeling concepts and Markov chains.

Remember that while the project's end product is important, the learning experience you gain while working on it is equally valuable. Document your process, challenges faced, and solutions found in your portfolio or GitHub repository to showcase your skills to potential employers.