HANGMAN

Hangman is a classic word-guessing game where the player attempts to guess a hidden word by suggesting letters one at a time. The game presents unique challenges for both human players and computational models, as it requires predicting the most likely letters in a partially revealed word.
This report details the strategies used to build such a model, the n-gram approach, and its application to the Hangman problem.

STRATEGIES USED

Initially decision tree classifier was used wherein decision tree model could be used to predict the most likely next letter in a word during a game of Hangman, based on the patterns observed in the training data. During gameplay, the current word pattern (with some letters already guessed) can be transformed into a similar feature vector and passed to the decision tree to predict the next most probable letter.

The dictionary of words from file (`words_250000_train.txt`) was used to build n-gram models. The models are built using up to 6-grams, which are sequences of up to 6 letters. Each n-gram model stores the frequency of each letter sequence found in the dictionary.

1-gram to 6-gram: Each n-gram model is built by iterating through the dictionary and counting the occurrences of sequences of letters. For example, the 6-gram model counts sequences of 6 letters and stores how often each sequence appears.

The input word (which includes underscores representing unknown letters) is processed to create a regular expression pattern.The current dictionary is filtered based on incorrect guesses, and the n-gram models are updated accordingly.

The next letter to guess is determined by calculating probabilities using the n-gram models, starting with the 6-gram model and falling back to lower n-grams if needed.

For each possible position in the word (considering known and unknown letters), the code checks for matching patterns in the n-gram models and calculates the probability of each letter being the correct guess.

The probabilities derived from different n-gram models are combined to give a final score for each letter. This score determines which letter is the most likely to be correct, guiding the next guess in the game.

ADVANTAGES

1) N-grams capture the natural order and dependencies between letters in words, which is crucial for making accurate predictions in Hangman.
2) By using a range of n-gram sizes, the model can adapt to different lengths and patterns of words, providing more robust predictions.
3) Compared to the initial decision tree approach, the n-gram model offers higher accuracy by directly leveraging the letter sequences within the dictionary.