Model Selection:

Model Name	Test accuracy
Logistic Regression	62.77%
Support Vector Machine	58.14%
Naive Bayes	55.73%
Decision Tree	50.30%
Multi-layer perceptron (sklearn)	58.55%
Multi-layer perceptron	49.50%
Transformer model	45.87%
Attention Based model	44.86%

Logistic Regression appears to be the best choice based on the provided metrics. It has the highest test accuracy (62.77%) among all the models listed. This indicates that Logistic Regression is better at generalizing to unseen data and making accurate predictions on new resumes.

Rationale for Choosing Logistic Regression:

1. Highest Accuracy: Logistic Regression achieves the highest accuracy among all the models listed, making it the most effective at correctly classifying the data in this case.
2. Simplicity and Interpretability: Logistic Regression is relatively simple compared to more complex models like Multi-layer Perceptron (MLP) and Transformer models. This simplicity often translates to easier interpretability, which can be beneficial for understanding how predictions are made.
3. Fewer Hyperparameters: Logistic Regression generally has fewer hyperparameters to tune compared to models like MLP or Transformer models, which can simplify the modeling process and reduce the risk of overfitting.
4. Performance vs. Complexity: While more complex models (such as MLP and Transformers) might offer the potential for higher accuracy in some cases, their performance here is not significantly better than Logistic Regression. Moreover, they require more computational resources and can be more challenging to fine-tune.
5. Practical Considerations: If computational resources or model deployment complexity is a concern, Logistic Regression is often more practical due to its efficiency and ease of use.

Details on preprocessing:

1. Initial Text Preprocessing (preprocess_text function)
    - This function performs basic text cleaning operations:
    - Lowercasing: Converts all characters in the text to lowercase to ensure uniformity (e.g., "Data Scientist" becomes "data scientist").
    - Special Character Removal: Removes all special characters, including punctuation marks, to focus on alphanumeric content only (e.g., "Hello, World!" becomes "Hello World").
    - Whitespace Trimming: Removes any leading or trailing whitespace from the text (e.g., " Data Scientist " becomes "Data Scientist").

2. Advanced Text Preprocessing (further_preprocess_text function)
    - This function involves more advanced NLP techniques for refining the text data:
    - Tokenization: Splits the text into individual words (tokens) based on whitespace.
    - Stop Words Removal: Filters out common stop words (e.g., "and", "the", "is") using the NLTK stopwords corpus. This reduces noise in the text data.
    - Lemmatization: Converts words to their base or root form (e.g., "running" becomes "run") using NLTK's WordNetLemmatizer.
    - Stemming: Further reduces words to their stem form (e.g., "running" becomes "run") using NLTK's PorterStemmer. This step may overlap with lemmatization but focuses on a more aggressive reduction.
    - Reconstruction: Joins the processed tokens back into a single string.

3. Number Removal (remove_numbers function)

This function removes all numeric characters from the text to eliminate irrelevant information that may not contribute meaningfully to text analysis.

Instructions on Running the script:

Please follow the README.md file.