1. Problem statement: Restate the initial project that you proposed in deliverable one in 2 – 3 sentences. Be sure to refer back to this problem statement in the following questions.
   * The aim of our project is to analyze text from news media to identify political leaning. We will use a logistic regression model and data drawn from a wide array of political articles to accurately model this bias.
2. Data Preprocessing: Confirm the dataset you are working with. State any changes from the initial dataset you chose. Discuss the content of the dataset (number of samples, labels, etc.). Describe and justify your data preprocessing methods (did you delete or modify any data? If so, why?).
   * We will be working with [this](https://www.kaggle.com/datasets/mayobanexsantana/political-bias/data) dataset, with minor tweaks made for formatting and to remove empty entries or extra characters. The final dataset after preprocessing is made up of about 3.5 thousand entries consisting of article titles, content, bias, tokenized text, and BERT embeddings. The tokenization and BERT feature extraction will make working with this dataset easier and more fruitful.
3. Machine learning model: In the first deliverable, you proposed a model for your project. If you decided to change your model, explain why. Restate your chosen model and elaborate on the design decisions.
   * A): We installed necessary libraries such as nltk, transformers, torch, sentence-transformers to use for the original text. We also imported pandas and word2vec to initializes NLP tools like tokenization, lemmatization, and stop word filtering. The Bert Encoder contains 12 Transformer layers where the input is a tokenized text, and the output is a 768-dimensionalembedding for each token. The logistic Regression classifier contains a 768-dimensional BERT embedding with 2 layers. There is a linear layer and a sigmoid activation for binary classification. The output is the probability of political bias.
     + Architecture graph:
       - Input text -> Tokenization -> BERT Encoder -> Extract embedding -> Logistic Regression -> Sigmoid activation -> Output probability
   * B): The choice split is 70/15/15 for training/validation/test splits respectively. Since, BERT embeddings are high-dimensional, a large training set is better for the model to learn decision boundaries. Then, 15% for validation set allows fine-tune to avoid data leakage. Finally, 15% for test set ensures the prediction of the model reflects the correct answer. We used L2regularization because logistic regression is prone to overfitting when dealing with high-dimensional data. For the optimization strategy, we used a standard optimization algorithm with a learning rate = 0.01, momentum=0.9. The hyperparameter tuning is perform for grid search with SGD and we will adjust the regularization strength, batch size and learning rate.
   * C): To be done by Justin Cyr
   * D): When doing the testing, we found a KeyError with bert\_embedding, to fix it, we need to ensure that the column is the same as in the csv file. We have to write a line of code to strip it.
4. Our preliminary results show that the Logistic Regression model achieves  
   an accuracy of 23%, which is well below the 80% target. The classification   
   report reveals that Center bias is the most accurately classified category,   
   while Lean Left and Lean Right are frequently misclassified. The confusion   
   matrix further confirms that many predictions are incorrect, especially   
   between adjacent bias categories. These results indicate that Logistic   
   Regression struggles with the complexity of the task. As we are facing an   
   issue in model\_test.py code, we used the help of a generative AI to execute   
   the code and give an estimation of the accuracy of our model.
5. To improve our model and fix our issues, we are looking to refine our   
   dataset by transforming the texts into a correct list form where the list will   
   contains the words that will help us to use BERT embedding and then test   
   again our model and find how to improve the accuracy to make it reach   
   80%.